

АКАДЕМИЯ НАУК УКРАИНСКОЙ ССР
ИНСТИТУТ КИБЕРНЕТИКИ им. В. М. ГЛУШКОВА

З.М. Асельдеров
Г.А. Донец

**ПРЕДСТАВЛЕНИЕ
И
ВОССТАНОВЛЕНИЕ
ГРАФОВ**

КИЕВ НАУКОВА ДУМКА 1991

УДК 510 (022) — 512.54 + 519.1—519.9

Представление и восстановление графов / Асельдеров З. М., Донец Г. А.;
Отв. ред. Андон Ф. И.; АН УССР. Ин-т кибернетики им. В. М. Глушкова.
— Киев : Наук. думка, 1991. — 192 с. — ISBN 5-12-002332-0

Монография посвящена теоретическим и прикладным вопросам теории графов. Наряду с известными и общепринятыми способами представления графов предлагается способ задания графа с помощью некоторой квадратичной формы. Изложены элементы теории сложности алгоритмов для задач на графах. Освещены проблемы оптимального представления графов. Рассмотрены операции над графами, заданными как традиционными способами, так и своими формальными квадратичными формами. Дается некоторый подход к решению одной из классических проблем теории графов — проблеме восстановления графа по его полному допустимому набору подграфов, известной как гипотеза Улама.

Для студентов вузов по специальности математика и прикладная математика, а также для научных работников и инженеров.
Табл. 17.

Ил. 80. Библиогр.; С. 185—186.

Ответственный редактор
Ф. И. Андон

Утверждено к печати ученым советом
Института кибернетики им. В. М. Глушкова АН УССР

Редакция физики и кибернетики

Редактор И. Г. Васинюк

А 1602100000-188
M221(04)-91 180-91

ISBN 5-12-002332-0

© З. М. Асельдеров, Г. А. Донец, 1991

ПРЕДИСЛОВИЕ

Предлагаемая книга написана на основе цикла лекций по теории графов в рамках общего курса по дискретной математике, который читался студентам факультета вычислительной математики и информатики Киевского политехнического института и факультета автоматизации строительного процесса Киевского инженерно-строительного института.

Теория графов — один из важных разделов современной дискретной математики. Рост проблем, связанных с проектированием и созданием интегрированных АСУ, САПР и СУБД (например, задачи календарного планирования, сетевого планирования и управления на промышленных предприятиях и др.), обусловил повышенный интерес математиков и инженеров к самой теории графов. Наряду с решением прикладных задач теории графов получены существенные результаты на стыке теории графов с современной алгеброй, математической логикой, комбинаторной топологией и общей теорией систем. Для систематического изучения теории графов рекомендуется книга А. А. Зыкова «Основы теории графов».

Настоящая монография посвящена узкому кругу вопросов, дополняющих друг друга, а именно, представлению и восстановлению графов. Книга состоит из пяти глав. В гл. 1 наряду с известными способами представления графов уделено большое внимание аналитическим представлениям графов. Гл. 2 посвящена проблемам оптимального представления графов. Рассмотрено представление графов с помощью структур данных. Дается оценка числа операций алгоритмов. В гл. 3 освещены элементы теории сложности алгоритмов для задач на графах. В ней вводятся основные понятия теории сложности. Подробно описан класс NP -полных задач и его применение для анализа задач. В гл. 4 наряду с операциями над графами, заданными известными в литературе способами, достаточно полно освещаются эти же операции над графами, которые заданы в виде некоторой формальной квадратичной формы. Гл. 5 в основном посвящена проблеме восстановления графов. В ней предлагается некоторый подход к решению одной из проблем в теории графов известной в литературе как гипотеза С. Улама. Дается алгоритм восстановления графа по его полному допустимому набору подграфов, а также дается нижняя и верхняя оценки минимального набора подграфов достаточного для восстановления искомого графа.

Гл. 1—3 (за исключением 1,5) написаны Г. А. Донцом, гл. 4, 5 — З. М. Асельдеровым.

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ

$x \in M$	— элемент x принадлежит множеству M ;
$x \notin M$	— элемент x не принадлежит множеству M ;
$A \subseteq B$	— A есть подмножество множества B ;
$A \subset B$	— $A \subseteq B$ и $A \neq B$;
\emptyset	— пустое множество;
$ A $	— количество элементов множества A ;
$A \cup B$	— объединение множеств A и B ;
$\bigcup_{i \in I} A_i$ или $\bigcup_{i=1}^n A_i$	— объединение множеств A_i , где i пробегает индексное множество I или $i = 1, 2, \dots, n$;
$A \cap B$	— пересечение множеств A и B ;
$\bigcap_{i \in I} A_i$ или $\bigcap_{i=1}^n A_i$	— пересечение множеств A_i , где i пробегает индексное множество I или $i = 1, 2, \dots, n$;
\bar{A} , $\neg A$	— не A , логическое отрицание высказывания A ;
$A \vee B$	— A или B , дизъюнкция высказываний A и B (неразделительная), т. е. допускающая одновременную их истинность;
$A \wedge B$, $A \& B$	— A и B конъюнкция высказываний A и B ;
$A \Rightarrow B$	— если A , то B , логическая импликация;
$A \Leftrightarrow B$	— A равнозначна B , логическая эквивалентность;
$\forall x ((x \in A) \& P(x))$	— для любого элемента x из множества A истинно высказывание P об этом элементе;
$\exists x ((x \in A) \& (P(x)))$	— множество A имеет хотя бы один такой элемент x , о котором истинно высказывание P ;
$\{x/P(x)\}$	— множество всех тех элементов x , для которых истинно высказывание P ;
$\{x \in A/P(x)\} \doteq \{x/x \in A \& P(x)\}$	— равно по определению;
\Leftrightarrow	— равнозначно по определению;
xy	— упорядоченная пара элементов x, y ;
\overline{xy}	— неупорядоченная пара элементов x, y ;
$A \times B \doteq \{xy/x \in A \& y \in B\}$	— декартово произведение множества A на множество B ;
$r = (r_1, r_2, \dots, r_n)$	— n -мерный вектор, упорядоченная система n чисел;
$\ a_{ij}\ _m^n$	— матрица с n строками и m столбцами;
$[x]$	— наибольшее целое число, не превосходящее x .

СПОСОБЫ ПРЕДСТАВЛЕНИЯ ГРАФОВ

1.1. ОБЩЕЕ ПРЕДСТАВЛЕНИЕ ПРОИЗВОЛЬНЫХ ГРАФОВ

Понятие графа служит для математического изучения ситуации, когда имеются две совокупности объектов, при этом объекты второй группы играют роль связей, соединяющих пары предметов первой группы друг с другом (речь может идти, например, об отдельных деталях электрической сети и соединяющих проводниках; о городах и соединяющих их авиалиниях; о людях и отношениях любви, дружбы или родства между ними, о целых числах и отношениях делимости; т. д.). Для одной и той же пары предметов допускается одновременное наличие нескольких связей, среди которых могут быть и односторонние и двусторонние; возможны связи, соединяющие предмет с самим с собой.

Можно упростить понятие связи, считая достаточным знание о ее наличии или отсутствии. Можно потребовать, чтобы между каждой парой допускалась одна связь. Всякие упрощения или требования приводят к какому-то подклассу из общего класса. Чтобы не утратить при этом многих приложений, надо определить граф так, чтобы это определение охватывало все случаи. Этим свойством обладает следующее определение. Дан граф $G = (X, U, P)$, если даны два множества X, U и трехместный предикат P , удовлетворяющий следующим двум условиям: а) P определен на всех таких упорядоченных тройках элементов x, u, y , для которых $x, y \in X$ и $u \in U$; б) элемент u , соответствующий паре x и y , может указывать (но необязательно) на их упорядоченность.

Очевидно, это определение предполагает, что $X \neq \emptyset$ и $X \cap U = \emptyset$. Элементы множества X называются *вершинами*, элементы множества U (которое может быть и пустым) — *ребрами*, а предикат P — *инцидентором* графа G . Условие «б» позволяет нам разбить множество ребер U на непересекающиеся подмножества \bar{U}, U^o, \bar{U} . Элементы первого множества \bar{U} соединяют только упорядоченные пары вершин и называются *дугами*, элементы второго множества U^o соединяют вер-

шину саму с собой и называются *петлями*, элементы третьего множества \bar{U} соединяют неупорядоченные пары и называются *звеньями*.

Если истинно высказывание $P(x, u, y)$ и $u \in U$, то говорят, что дуга u выходит из вершины x и заходит в вершину y . Две вершины x и y называются *смежными*, если они связаны ребром. В этом (и только этом) случае вершина x (также как и вершина y) инцидентна с ребром. Два ребра называются смежными, если у них есть общая инцидентная вершина. Пусть $X' \subseteq X$ и $U' \subseteq U$ — подмножества вершин и ребер графа $G = (X, U, P)$, а P' — предикат, индуцированный инцидентором P на этих подмножествах. Если X' и U' выбраны так, что они вместе с P' удовлетворяют всем пунктам определения графа, то граф $G' = (X', U', P')$ называется *частью графа G* , порожденной подмножествами X' и U' . Ясно, что произвольно заданная пара подмножеств порождает часть тогда и только тогда, когда включив в U' некоторые ребра графа G , необходимо также включать в X' те вершины G , которые соединяются этим ребром. К числу частей графа G относится и весь G ; остальные его части называются *правильными* или *собственными*. Предикат P' , индуцированный предикатом P , можно писать без штриха.

Особо важную роль в теории графов играют следующие два типа частей. Часть $G' = (X', U', P)$ называется *подграфом* графа $G = (X, U, P)$, порожденным подмножеством вершин $X' \subseteq X$, если подмножество ребер $U' \subseteq U$ выбирается так, что при образовании части G' сохраняется все те же ребра G , которые соединяют между собой сохраняемые вершины. Это означает, что после произвольного выбора непустого подмножества $X' \subset X$ подмножество $U' \subset U$ определяется однозначно. Часть $G' = (X, U', P)$, в которой $X' = X$, называется *суграфом* графа G , порожденным подмножеством ребер U' . Два графа $G = (X, U, P)$ и $G' = (X', U', P')$ называются *изоморфными*, если между их вершинами, а также между их ребрами можно установить взаимно однозначное соответствие $X \leftrightarrow X', U \leftrightarrow U'$, сохраняющее инцидентор, т. е. такое, что для всякой пары $(x, y) \in X$ ребра $u \in U$, соединяющего их, обязательно существуют соответствующие пара $(x', y') \in X'$ и ребро $u' \in U'$, соединяющие их. Отношение изоморфизма графов рефлексивно, симметрично и транзитивно. т. е. представляет собой эквивалентность.

В дальнейшем нас будет интересовать только такие свойства графов, которые сохраняются при замене графа любым ему изоморфным; тем самым оказываются несущественными как природа элементов, составляющих множества X и U , так и конкретный смысл их отношения (предикат P). Однако всег-

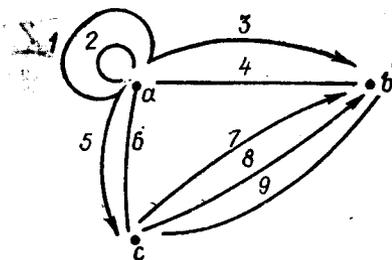


Рис. 1.1.1

да предполагаем, что вершины и ребра графа либо сами являются определенными объектами (буквами, числами и т. д.), либо помечены различными индексами из каких-либо конкретных индексных множеств (обозначены буквами или пронумерованы) и два графа, различающиеся лишь индексацией элементов, рассматриваем как изоморфные, но не как тождественные.

Все введенные понятия и обозначения поясним на примере. Пусть $X = \{a, b, c, d\}$, $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Посчитаем число различных высказываний для инцидентора P . Число всевозможных пар для X равно $4 \times 4 = 16$, вместе со всевозможными комбинациями ребер получим $16 \times 9 = 144$. Пусть инцидентор P определен так: $P(a, 1, a)$; $P(a, 2, a)$; $P(a, 3, b)$; $P(a, 4, b)$; $P(b, 4, a)$; $P(a, 5, c)$; $P(a, 6, c)$; $P(c, 6, a)$; $P(c, 7, b)$; $P(c, 8, b)$; $P(b, 9, c)$ истинны, остальные 133 ложны.

По этим значениям можно разбить множество U на три составляющих: $\bar{U} = \{3, 5, 7, 8, 9\}$, $U^o = \{1, 2\}$, $\bar{U} = \{4, 6\}$. На рис. 1.1.1 ребра графа $G = (X, U, P)$ изображены отрезками, соединяющими инцидентные им вершины, причем стрелка на каждой дуге показывает из какой вершины в какую идет эта дуга. Видно, что вершина a смежна сама с собой; вершина d не смежна ни с какой вершиной; ребра 1 и 2, 1 и 3, 1 и 4, 3 и 4, 7 и 8, 8 и 9 смежны, а ребра 1 и 8 не смежны; дуга 7 идет из вершины c в вершину b , звено 4 соединяет вершины d и b , ребро 2 является петлей при вершине a . Подграф $G' = (X', U', P)$, для которого $X' = \{a, b\}$, изоморфен подграфу $G'' = (X'', U'', P)$, для которого $X'' = \{a, c\}$, но не изоморфен подграфу G''' , порождаемому множеством вершин $X''' = \{b, c\}$ (рис. 1.1.2). Изоморфизм между G' и G'' можно установить двумя способами:

$a \leftrightarrow a; b \leftrightarrow c; 1 \leftrightarrow 1; 2 \leftrightarrow 2; 3 \leftrightarrow 5; 4 \leftrightarrow 6$
или
 $a \leftrightarrow a; b \leftrightarrow c; 1 \leftrightarrow 2; 2 \leftrightarrow 1; 3 \leftrightarrow 5; 4 \leftrightarrow 6$

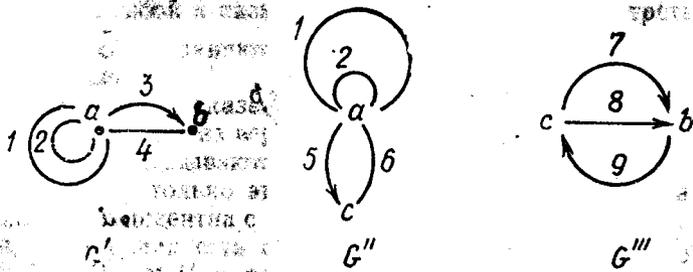


Рис. 1.1.2

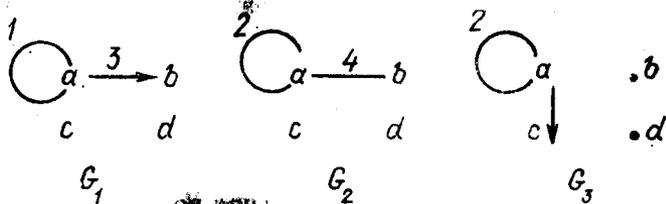


Рис. 1.1.3

Рассмотрим суграфы G_1 , G_2 и G_3 , для которых соответственно $U_1 = \{1, 3\}$, $U_2 = \{2, 4\}$ и $U_3 = \{2, 5\}$, изображенные на рис. 1.1.3.

Здесь G_1 изоморфен G_3 , но на не изоморфны G_2 . Часть G_0 , для которой $X_0 = \{a\}$, $U_0 = \emptyset$, изоморфна подграфу с множеством вершин $\{d\}$, но не изоморфна подграфу с множеством вершин $\{a\}$; последний же не только изоморфен, но и совпадает с частью G_0 , для которой $X^0 = \{a\}$, $U^0 = \{1, 2\}$. Пара подмножеств $Y = \{a, b, d\}$, $V = \{1, 5, 7\}$ не определяет никакой части графа G . В дальнейшем будем рассматривать только конечные графы, т. е. такие, у которых множества X и U конечны. Количество $|X|$ вершин графа G будем обозначать через $n(G)$ либо $p(G)$, а количество $|U|$ его ребер — через $m(G)$ либо $q(G)$.

1.2. ЗАДАНИЕ ГРАФОВ С ПОМОЩЬЮ МАТРИЦ

Графы, в которых на самом деле есть и дуги, и звенья, встречаются сравнительно редко, хотя полностью их из рассмотрения исключить нельзя. Поэтому общее представление графов, предложенное в 1.1, в более простых случаях громоздко. Рассмотрим случаи, когда можно общее определение заменить более конкретным. Граф G (общего вида) называется *неориентированным*, если $\vec{U} = \emptyset$, и *ориентированным*, если

$\vec{U} = \emptyset$, иначе, орграфом; в случае $U^0 = \emptyset$ к наименованию графа добавляется слово «без петель».

Особую важную роль играют *обыкновенные* графы, которые характеризуются следующими свойствами: конечны; являются неориентированными; не содержат петель; не содержат кратных параллельных ребер.

Обыкновенный граф можно записать в виде $G = (X, U)$. Среди ориентированных графов важную роль играют *графы Бержа*, характеризующиеся тем, что из каждой вершины в каждую другую может идти не более одной дуги, а при каждой вершине может быть не более одной петли. Следуя книгам Бержа [1, 2], будем записывать такой граф в виде $G = (X, \Gamma)$, где $X \neq \emptyset$ — множество вершин, а $\Gamma: X \leftrightarrow 2^X$ — отображение, относящее каждой $x \in X$ подмножество $\Gamma x \subset X$ тех вершин, в которые из x идет дуга или петля. Любой граф можно представить с помощью матрицы, однако для общего графа потребуется использование многозначной логики. Посмотрим, как это представление выполняется для графов Бержа и обыкновенных графов (рис. 1.2.1). Пронумеруем вершины этих графов натуральными числами $1, 2, \dots, n$. Обозначим \overline{xy} упорядоченную пару, т. е. дугу, выходящую из вершины x и заходящую в вершину y , а xy обозначает ребро, соединяющее вершины x и y .

Составим для этих графов квадратные матрицы $A(G) = \|a_{ij}\|_n^n$ с n строками и n столбцами, элементы которой определяются следующим образом:

$$a_{ij} \begin{cases} 1, & \text{если из } i\text{-й вершины заходит дуга в } j\text{-ю вершину;} \\ 0 & \text{в противном случае.} \end{cases}$$

Видно, что матрицы $A(G_1)$ и $A(G_2)$ представляют граф Бержа G_1 и обыкновенный граф G_2 однозначно:

$$A(G_1) = \begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix}, \quad A(G_2) = \begin{vmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{vmatrix}$$

Заметим, что в матрице $A(G_1)$ элемент $a_{3,3}$ равен 1, так как при вершине 3 есть петля, а матрица $A(G_2)$ симметрична, так как ребро приравнивается двум дугам, соединяющим две вершины, но противоположно направленным. Матрица $A(G)$ называется *матрицей смежностей* графа G . Существует еще

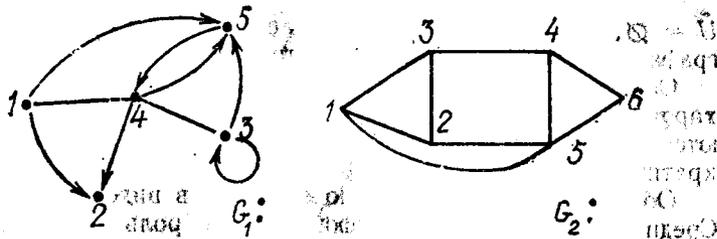


Рис. 1.2.1

несколько способов представления указанных графов с помощью матриц. Пусть задана матрица $S(G)$, у которой число строк равно числу вершин, а число столбцов — числу дуг. Пусть дуги занумерованы в порядке $1, 2, \dots, m$. Положим

$$s_{ij} = \begin{cases} +1, & \text{если в столбце, соответствующем дуге } \overrightarrow{xy}, \\ & y = 1; \\ -1, & \text{если в столбце, соответствующем дуге } \overleftarrow{xy}, \\ & x = 1; \\ 0, & \text{если вершина } i \text{ не инцидентна дуге;} \\ 2, & \text{если } j\text{-й столбец соответствует петле } (i, j). \end{cases}$$

Такая матрица называется *матрицей инциденций* для дуг:

$$S(G_1) = \begin{vmatrix} \overrightarrow{12} & \overrightarrow{14} & \overrightarrow{15} & \overrightarrow{33} & \overrightarrow{35} & \overrightarrow{42} & \overrightarrow{43} & \overrightarrow{45} & \overrightarrow{54} \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & -1 \end{vmatrix}$$

В случае неориентированного графа составляем *матрицу инциденций* для ребер

$$R(G_2) = \begin{vmatrix} \tilde{12} & \tilde{13} & \tilde{15} & \tilde{23} & \tilde{25} & \tilde{34} & \tilde{45} & \tilde{46} & \tilde{56} \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{vmatrix}$$

Здесь

$$r_{i,j} = \begin{cases} +1, & \text{если ребро, соответствующее } j\text{-му столбцу,} \\ & \text{инцидентно вершине } i; \\ 0 & \text{в противном случае.} \end{cases}$$

Наконец, для обыкновенных графов можно построить *матрицу смежностей ребер* графа (G) . Матрица состоит из m строк и m столбцов.

$$m_{i,j} = \begin{cases} +1, & \text{если ребра, соответствующие } i\text{-й строке и } j\text{-му} \\ & \text{столбцу, имеют общую вершину;} \\ 0 & \text{в противном случае.} \end{cases}$$

$$M(G_2) = \begin{vmatrix} \tilde{12} & \tilde{13} & \tilde{15} & \tilde{23} & \tilde{25} & \tilde{34} & \tilde{45} & \tilde{46} & \tilde{56} \\ \tilde{12} & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \tilde{13} & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ \tilde{15} & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \tilde{23} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \tilde{25} & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ \tilde{34} & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ \tilde{45} & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \tilde{46} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ \tilde{56} & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$$

Эту матрицу можно трактовать и как матрицу смежностей для нового графа, вершинами которого являются ребра графа G_2 , а ребрами — пара ребер, имеющих общую вершину. Такой граф называется *графом смежностей ребер* для графа G_2 или *смежностным графом* G_2 .

1.3. ДВОИЧНОЕ ПРЕДСТАВЛЕНИЕ ГРАФОВ

Используя матрицу смежностей графа $A(G)$, можно представить граф в виде двоичного кода. Очевидно, что если запомнить последовательность нулей и единиц матрицы $A(G)$, то по ней можно восстановить весь граф. Так как матрица $A(G)$ симметрична, достаточно запомнить лишь те элементы, которые расположены над главной диагональю, т. е. последовательность $(a_{1,2}, a_{1,3}, a_{1,4}, a_{2,4}, a_{3,4}, \dots, a_{n-1,n})$. Длина этой последовательности равна $C_n^2 = \frac{n(n-1)}{2}$.

Число, которое получается после выполнения операций

$$a_{1,2} \cdot 2^0 + a_{1,3} \cdot 2^1 + a_{2,3} \cdot 2^2 + a_{1,4} \cdot 2^3 + \dots + a_{n-1,n} \cdot 2^{n-1},$$

назовем *двоичным кодом матрицы* $A(G)$. Так как разным нумерациям вершин графа соответствуют разные матрицы смежностей, то существует $n!$ двоичных кодов. Наименьший из этих кодов будем называть *мини-кодом* $\mu(G)$, а наибольший — *макси-кодом* $\mu(G)$ графа G .

По любому из этих кодов однозначно восстанавливается одна из матриц смежностей графа, а значит и сам граф. Рассмотрим пример. Граф из четырех вершин допускает $4! = 24$ различных нумераций вершин (рис. 1.3.1). Из-за очевидной симметрии в структуре графа перестановка друг с другом номеров вершин степени два не изменяет матрицу смежностей и поэтому различных матриц будет не 24, а только 12. Построим соответствующие матрицы для нумераций верхнего ряда, нижний ряд имеет те же матрицы:

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}; \quad \mu(G) = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + \\ + 0 \cdot 2^4 + 1 \cdot 2^5 = 39$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}; \quad \mu(G) = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + \\ + 1 \cdot 2^4 = 57$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}; \quad \mu(G) = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + \\ + 1 \cdot 2^4 + 1 \cdot 2^5 = 53$$

Мини-код $\mu(G) = 15$ отвечает седьмой нумерации (и расположенной под ней 19-й), а *макси-код* $\mu(G) = 60$ соответствует восьмой нумерации (и расположенной под ней 20-й). Наоборот, если о графе известно, что он имеет четыре вершины, а двоичный код одной из его матриц смежностей равен 38, то представляя это число в двоичной системе до разряда 2^5 (так как $C_4^2 - 1 = \frac{4 \cdot 3}{1 \cdot 2} - 1 = 5$, получаем $38 = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$).

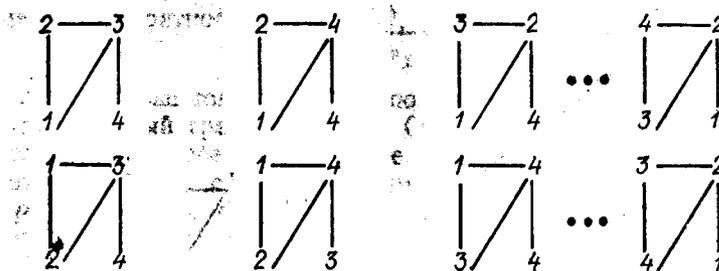


Рис. 1.3.1

Из этого разложения находим $a_{1,2} = 0$, $a_{1,3} = 1$, $a_{2,3} = 1$, $a_{1,4} = 0$, $a_{2,4} = 0$, $a_{3,4} = 1$. Это дает нам матрицу смежностей и соответствующий граф (рис. 1.3.2):

$$A(G) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

В случае пятивершинного графа с тем же кодом надо записать разложение до разряда 2^9 (так как $C_5^2 - 1 = \frac{5 \cdot 4}{1 \cdot 2} = 9$): $38 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + \dots + 0 \cdot 2^9$.

Поскольку число вершин нового графа больше на единицу, то очевидно, что пятая вершина будет изолированной (рис. 1.3.3), а в матрице смежностей $A(G)$ пятая строка и пятый столбец будут состоять из нулей

$$A(G) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Количество вершин можно не сообщать, если заранее предполагать, что у графа нет изолированных вершин. Если всегда задавать граф с помощью макси-кода, то станет ясно, что он всегда содержит единицу старшего разряда 2^k . Таким образом, макси-код $\mu(G)$ соответствует такой нумерации вершин графа, при которой вершины с номерами $n-1$ и n смежны, в силу чего число вершин однозначно определяется из условия $n = \frac{1 + \sqrt{1 + 8k}}{2}$. Лишь в случае $\mu(G) = 0$, т. е. когда G — безреберный граф, число его вершин может быть другим.

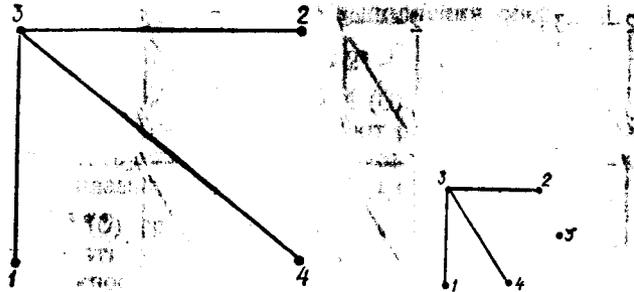


Рис. 1.3.2

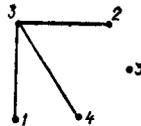


Рис. 1.3.3

1.4. БИНАРНЫЕ ОТНОШЕНИЯ ДЛЯ ГРАФОВ

Бинарное отношение R определяется как соотношение

$$xRy, \quad (1.4.1)$$

которое выполняется для некоторых пар элементов заданного множества X . В соответствии с этим определением бинарное отношение может быть представлено в виде графа с множеством вершин X

$$G(R) = G(X, U), \quad (1.4.2)$$

так что (ориентированное) ребро (x, y) принадлежит U тогда и только тогда, когда в R выполняется соотношение (1.4.1). Обратное, любой граф определяет бинарное отношение R на своем множестве вершин, если положить xRy для каждого его ребра $(x, y) \in U$, где $x, y \in X$.

Имеется, однако, небольшое различие между этими двумя понятиями. Приписывать отношению кратность обычно нет надобности. Поэтому можно говорить о взаимно однозначном соответствии между бинарными отношениями на множестве X , с одной стороны, и графами с однократными ребрами на множестве вершин X — с другой. Остановимся кратко на некоторых связях между бинарными отношениями и графами.

Граф, не содержащий ребер, отвечает нулевому отношению $x0y$, которое не выполняется ни для какой пары элементов. Полный граф отвечает универсальному отношению xUy , которое выполняется для любой пары элементов. Каждое отношение R имеет дополнительное отношение, как отрицание, \bar{R} , так что xRy тогда и только тогда, когда (1.4.1) не выполняется. Например, для отношения $x = y$ (x и y одинаковы) дополнительное отношение есть $x \neq y$ (x отлично от y). Граф $G(\bar{R})$ является *дополнительным* графом для $G(R)$, т.е. дополнением $G(\bar{R}) = U - G(R)$ по отношению к полному графу U , определенному на множестве вершин X . Для любого отношения R су-

ществует обратное отношение R^* , так что

$$yR^*x \quad (1.4.3)$$

тогда и только тогда, когда выполняется (1.4.1). Граф $G(R^*)$ есть обратный граф для $G(R)$. Отношение $R = R^*$, которое является само себе обратным, т.е. из xRy следует yRx и наоборот, называется симметрическим. В этом случае вершины x и y должны быть соединены ориентированным ребром в каждом направлении, но проще заменить их одним неориентированным ребром. Таким образом, неориентированные графы отвечают симметрическим отношениям.

Говорят, что из отношения R следует отношение R' , или что R' содержит R , $R' \supset R$, если из xRy всегда вытекает $xR'y$. Очевидно, для соответствующих графов $G(R') \supset G(R)$.

Для любых двух отношений R_1 и R_2 пересечение $R_1 \cap R_2$ есть отношение $x(R_1 \cap R_2)y$, которое выполняется тогда и только тогда, когда одновременно

$$xR_1y \text{ и } xR_2y. \quad (1.4.4)$$

Сумма есть отношение $R_1 \cup R_2$, которое выполняется тогда и только тогда, когда выполняется хотя бы одно из соотношений (1.4.4). Таким образом, для соответствующих графов имеем

$$G(R_1 \cap R_2) = G(R_1) \cap G(R_2),$$

$$G(R_1 \cup R_2) = G(R_1) \cup G(R_2).$$

До сих пор рассматривались отношения между элементами одного множества. Можно также вводить отношения $xR'x'$; $x \in X$; $x' \in X'$ между элементами двух различных множеств X и X' . В качестве иллюстрации можно взять отображение τ элементов из X на элементы из X' ; здесь соотношение $x\tau x'$ выполняется тогда и только тогда, когда $x' = \tau(x)$ есть образ x при отображении τ . Другой важный пример получим, взяв в качестве X некоторое множество, а в качестве X' — семейство всех подмножеств A множества X ; здесь отношение $x \in A$ означает, что x есть элемент A . В графах для таких отношений, связывающих одно множество с другим, все ребра будут соединять X с X' . Такого рода графы называются *двудольными*.

Кроме симметрии приходится иметь дело и с другими свойствами отношений.

Отношение R называется *рефлексивным*, если xRx для любого $x \in X$. Соответствующий граф имеет петлю в каждой вершине. Обратное отношение также будет рефлексивным. Отношение *антирефлексивно*, если xRx никогда не выполняется, т.е. граф не имеет петель. Отношение $x \neq y$ антирефлексивно; другим примером может служить свойство ортогональности для двух векторов. Отношение R *транзитивно*, если из xRy

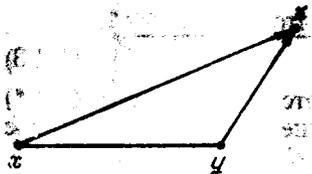


Рис. 1.4.1

и yRz следует xRz . Для графа это означает, что, если $G(R)$ содержит ребра (x, y) и (y, z) , то он также содержит и замыкающее ребро (x, z) (рис. 1.4.1).

В дальнейшем будет проведено более детальное изучение бинарных отношений с точки зрения теории

графов. Однако есть два вида отношений, настолько важных и общеупотребительных, что имеет смысл рассмотреть их здесь.

Отношение эквивалентности. Отношение R , определяемое на множестве X , называется *отношением эквивалентности*, если оно: 1) рефлексивно; 2) симметрично; 3) транзитивно.

Все элементы из X , эквивалентные данному элементу x , образуют множество $R(x)$, которое называется *классом эквивалентности элемента x* . Из рефлексивности R следует, что $x \in R(x)$. Если xRy и yRz , то из транзитивности следует xRz , так что $R(x) \supseteq R(y)$. Поэтому из симметричности отношения получаем, что $R(x) = R(y)$ при xRy . Наконец, два различных класса эквивалентности $R(x)$ и $R(z)$ не могут иметь какого-либо общего элемента y , так как иначе было бы $R(x) = R(y) = R(z)$.

Таким образом, классы эквивалентности образуют разбиение X , т. е. разложение X на непересекающиеся подмножества. Класс эквивалентности $R(x) = x$, состоящий из единственного элемента, называется *особым*, в противном случае — *неособым*.

Предположим теперь, что, наоборот, задано разбиение

$$X = \bigcup B_k \quad (1.4.5)$$

множества X на непересекающиеся подмножества B . Тогда можно определить отношение эквивалентности с этими классами B , полагая xRy тогда и только тогда, когда x и y принадлежат одному подмножеству B_k . В соответствующем графе $G(R)$ любые две вершины из одного множества B_k будут соединены ребром, и никакое ребро не соединяет вершины из различных множеств. Следовательно,

$$G(R) = \bigcup G_0(B) \quad (1.4.6)$$

является прямой суммой полных графов, определенных на различных множествах B_k .

Частичное упорядочение. Отношение

$$x \geq y \quad (1.4.7)$$

называется *частично упорядоченным* или *отношением включения*, если оно обладает следующими свойствами: 1) $x \geq x$ (рефлексивность); 2) из $x \geq y$ и $y \geq z$ следует $x \geq z$ (транзитивность); 3) транзитивность.

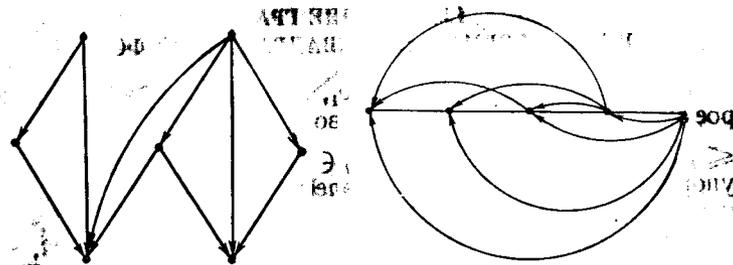


Рис. 1.4.2

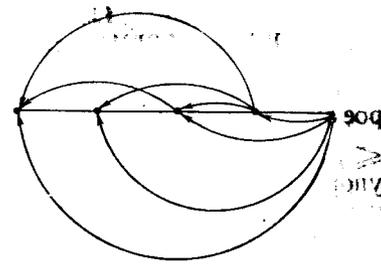


Рис. 1.4.3

Соответствующий граф транзитивен, имеет петли и любые две вершины в нем соединены не более, чем одним ребром. Граф на рис. 1.4.2 дает пример частичного упорядочения. *Частично упорядоченное множество есть множество, в котором определено частичное упорядочение.*

Отношение включения (1.4.7.) называется *отношением упорядочения*, а соответствующее множество X — упорядоченным, если помимо перечисленных, выполняется также условие: 4) для любой пары элементов $x, y \in X$ справедливо одно из соотношений $x \geq y$ или $y \geq x$. На рис. 1.4.3 изображен граф некоторого упорядоченного множества. Для семейства подмножеств $\{A\}$ множества X имеет место отношение включения множеств $A \supseteq B$, которое означает, что A содержит все элементы множества B . Очевидно, это отношение является частичным упорядочением. Можно показать, что каждое частичное упорядочение P изоморфно такому включению множеств. Для этого каждому элементу x из P соотнесем множество $P(x)$ всех таких элементов y , что $x \geq y$. Из аксиомы частичного упорядочения следует, что $P(x) \supseteq P(y)$ только для $x = y$. Далее, если $x \geq z$, то по транзитивности $P(x) \supseteq P(z)$; обратно, очевидно, что из последнего соотношения следует $x \geq z$. В связи с частичным упорядочением заметим, что наряду с ним используется также и несколько другое понятие.

Строгое частичное упорядочение. Отношение $x > y$ называется *строгим частичным упорядочением*, или *строгим включением*, если оно удовлетворяет двум условиям: 1) $x > y$ и $y > x$ одновременно не имеет места; 2) транзитивности.

Легко увидеть, что строгое частичное упорядочение можно рассматривать как пересечение частичного упорядочения и отношения $x = y$. Таким образом, получаем граф строгого частичного упорядочения, удаляя из графа частичного упорядочения все петли. Вместо выражения «строгое включение» иногда используется выражение «собственное включение».

Отметим, что свойство быть частичным упорядочением сохраняется при переходе к обратному отношению.

1.5. ЗАДАНИЕ ГРАФА В ВИДЕ ФОРМАЛЬНОЙ КВАДРАТИЧНОЙ ФОРМЫ

Пусть $V = \{v_0, v_1, v_2, \dots, v_1, \dots, v_p\}$ ($p \geq 0$) — некоторое конечное непустое множество, причем $v_i \neq v_j$ ($0 \leq i, j \leq p$), $W = \vec{V}^{[2]} = \{\vec{v}_i v_j / v_i, v_j \in V\}$, т. е. W — множество упорядоченных пар (v_i, v_j) элементов множества V . Пару $v_i v_j = (v_i, v_j) \in W$ будем иногда обозначать $v_i v_j$.

Рассмотрим коммутативную полугруппу $\mathfrak{F} = \{\mathfrak{M}, +\}$ с основным множеством \mathfrak{M} , порожденную множеством W , и с аддитивной операцией «+» (сложение), определенной на множестве следующими тождественными соотношениями: 1) $u + v = v + u$; 2) $u + (v + w) = (u + v) + w$; 3) $u + w_0 = w_0 + u = u$, где $u, v, w, w_0 \in W$, причем $w_0 = (v_0, v_0) = v_0 v_0$ — «нуль» полугруппы \mathfrak{F} .

Пусть $u_1, u_2, \dots, u_1, \dots, u_n$ — элементы из \mathfrak{M} . Тогда, как принято в аддитивных полугруппах, положим: 4) $\sum_{i=1}^n u_i = u_1 + u_2 + \dots + u_i + \dots + u_n$.

В частности, при $u = u_1 = \dots = u_2 = \dots = u_i = \dots = u_n$ имеем 5) $nu = u + u + \dots + u$ (n «слагаемых»).

Таким образом, к соотношениям «1»—«5» можно добавить еще соотношения 6) $n(u + v) = nu + nv$; 7) $n(mu) = (nm)u$; 8) $(m + n)u = mu + nu$, где m, n — натуральные числа.

Так как m, n в нашем случае не являются элементами полугруппы \mathfrak{F} , а представляют собой нечто внешнее — натуральные числа, то при $n = 0$ полагаем 9) $0u = u0 = w_0$.

Пусть теперь G — произвольный $(p - q)$ граф. Если i -й ($1 \leq i \leq p$) вершине этого графа поставить в соответствие элемент вида $v_0 v_i \in W$ (т. е. i -ю вершину графа будем рассматривать как фиктивное ребро (дуга)), инцидентный самой вершине v_i и некоторой фиктивной вершине v_0 (аналог бесконечно удаленной точки в геометрии) и, вообще говоря, в нашем представлении вершины и ребра графа формально не различимы, а ребру (дуге), инцидентному вершинам v_i, v_j ($i, j = 0$), — элемент $v_i v_j \in W$, то легко показать справедливость следующего утверждения.

Лемма 1.5.1. Для каждого произвольного $(p - q)$ графа G можно построить формальную квадратичную форму следующего вида:

$$g(v_0, v_1, v_2, \dots, v_p) = \sum_{i,j} \alpha_{i,j} v_i v_j,$$

где $\alpha_{0,i} = 1$, а $\alpha_{i,j}$ ($i, j = \overline{1, p}$) — целые неотрицательные числа, которая определяет данный граф G , с точностью до

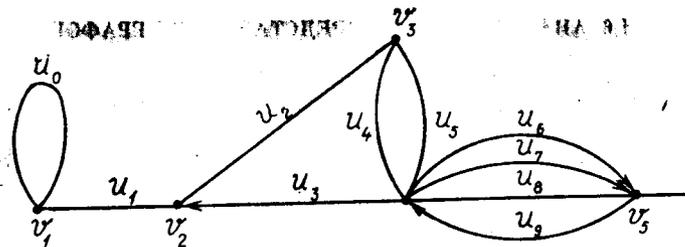


Рис. 1.5.1

замены любого неориентированного ребра, парой противоположно направленных дуг между теми же вершинами.

Очевидно, справедливо и обратное утверждение. Однако для графов без кратных ребер задание графа квадратичной формой однозначно.

Пример. На рис. 1.5.1 изображен граф с шестью вершинами, девятью ребрами и одной петлей u_0 , причем ребра u_1, u_3, u_6, u_7, u_9 являются дугами; u_4, u_5 — кратные ребра; u_8, u_7, u_8, u_9 — также кратные ребра; u_0 — изолированная вершина.

Формальная квадратичная форма, соответствующая графу на рис. 1.5.1, имеет вид

$$g(v_0, v_1, v_2, \dots, v_6) = \sum_{i,j=0}^6 \alpha_{i,j} v_i v_j = v_0 v_1 + v_0 v_2 + \dots + v_0 v_6 + v_1 v_1 + v_2 v_1 + v_2 v_3 + v_3 v_2 + v_4 v_2 + \dots + 2v_3 v_4 + 2v_4 v_3 + 3v_4 v_5 + 2v_5 v_4.$$

Ниже будем рассматривать неориентированные обыкновенные графы, т. е. графы, квадратичные формы которых имеют вид

$$g(v_0, v_1, v_2, \dots, v_p) = \sum_{i,j=0}^p \alpha_{i,j} v_i v_j \quad (\alpha_{i,j} \in \{0, 1\}),$$

и являются элементами полугруппы $\mathfrak{F} = \{\mathfrak{M}, +\}$, у которой соотношение «5» заменено соотношением $a + a = a$.

Для краткости при записи формальных квадратичных форм будем пользоваться некоторыми метасокращениями и метаобозначениями, сущность которых легко будет понята по контексту. Например,

$$\sum_{i=1}^p v_m v_i = v_m \sum_{i=1}^p v_i.$$

Формальная квадратичная форма, соответствующая графу, изображенному на рис. 1.5.1, будет записана в виде

$$g(v_0, v_1, v_2, \dots, v_6) = v_0 \sum_{i=1}^6 v_i + (v_1 + v_2) v_1 + (v_3 + v_4) v_2 + v_2 v_3 + 2v_3 v_4 + v_4 (2v_3 + 3v_5) + 2v_5 v_4.$$

1.6. АНАЛИТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ГРАФОВ

В последнее время с связи с увеличением размерности задач, решаемых на ЭВМ, все большее внимание привлекают вопросы эффективности алгоритмов. Для оценки сложности алгоритмов существует много критериев, основным из которых является продолжительность времени решения задачи, выраженная как функция от объема необходимой машинной памяти для исходных данных.

Многие сложные задачи удается сформулировать на языке теории графов, поэтому естественным является вопрос о способах представления исходных данных о графах. Для графов существует несколько способов представления их в машинной памяти. Если обыкновенный граф состоит из n вершин и m ребер, то представление его в виде матрицы смежностей требует памяти порядка n^2 единиц, а в виде списка смежностей — памяти порядка $(n + m)$ единиц [6]. Списком смежностей часто пользуются, когда m значительно меньше n^2 , поэтому на практике матрицами смежностей пользуются редко.

Списками смежностей удобно пользоваться, если в задаче определяют различные пути в графах или представляют сети большого объема. Однако существует ряд задач, где бывает важно только определить смежность двух любых вершин данного графа. Возникает вопрос, нельзя ли получить такого рода информацию о графе с помощью определенных вычислений, не загружая память машины исходными данными о структуре графа? В данной работе предлагается новый способ представления графов, который позволяет так закодировать вершины и ребра графа, что использование исходных данных о графах водится к определенным операциям над кодами вершин графа.

В последнее время теория графов стала интенсивно применяться в различных информационных моделях или управляющих системах, важнейшим свойством которых является структура или совокупность бинарных отношений на наборах элементарных единиц данных и действий. В зависимости от конкретной задачи можно использовать то или иное представление графа. В частности, можно определить граф как бинарное отношение на множестве своих вершин [3]. Очевидно, что любое определение удовлетворяет универсальному определению графа, данному в [4], как и приводимое ниже.

Определение 1. Графом G называется тройка $G = (X, U, F)$, где $X = \{x_1, x_2, \dots, x_n\}$ — множество вершин, $U = \{u_1, u_2, \dots, u_p\}$ — множество образующих, F — некоторая однозначная функция двух аргументов (порождающая функция) такая, что

$$\forall u \exists i \exists j (u \in U) \& (x_i, x_j \in X) \& [F(x_i, x_j) \in U]. \quad (1.6.1)$$

Относительно функции F будем предполагать, что она в какой-то мере «легко вычислима» и в подавляющем большинстве имеет явный аналитический вид. Отсюда название для таких графов, которые задаются аналитическим способом, — A -графы. Граф называется конечным, если n конечно.

Будем говорить, что в графе G из вершины x_i в вершину x_j заходит дуга, если $F(x_i, x_j) \in U$. Если $F(x_i, x_j) \in U$, то в вершине x_i существует петля.

Определение 2. Величина

$$\rho^+(x_i) = |\{x_j / F(x_i, x_j) \in U\}| \quad (1.6.2)$$

называется степенью *полуисхода*, а

$$\rho^-(x_i) = |\{x_j / F(x_j, x_i) \in U\}| \quad (1.6.3)$$

степенью *полузахода* вершины x_i .

Эти числа определены для ориентированных графов. A -граф, у которого $F(x_i, x_j) = F(x_j, x_i) = u$ для всех $u \in U$, называется неориентированным. Для него $\rho^+(x_i) = \rho^-(x_i) = \rho(x_i)$ есть степень вершины.

Определение 3. Графы $G = (X, U, F)$ и $G' = (X', U', F')$, называются *изоморфными*, если существует взаимно однозначное соответствие φ между множествами X и X' , сохраняющее смежность, т. е. $G \cong G'$, если

$$\forall i \forall j (x_i, x_j \in X) \& [F(x_i, x_j) \in U] \Leftrightarrow [F'(\varphi x_i, \varphi x_j) \in U']. \quad (1.6.4)$$

Определение 4. Представление графа $G = (X, U, F)$ называется *минимальным относительно функции F* , если для любого $G' = (X', U', F') = G(X, U, F)$ справедливо

$$|U| \leq |U'|. \quad (1.6.5)$$

Следует заметить, что мощность множества U в общем случае не равна количеству ребер графа.

Элементами множества X могут быть произвольные объекты (числа, множества, векторы, матрицы и т. д.). Наибольший интерес представляет случай, когда элементами множества X являются числа (целые, рациональные, действительные, комплексные). Такие графы будем называть числовыми.

Определение 5. Представление числового графа $G = (X, U, F)$ называется *совершенным относительно функции F* , если оно минимальное, и для любого минимального $G' = (X', U', F') = G(X, U, F)$ справедливо

$$\max_{x_i, x_j \in X} |x_j - x_i| \max_{x_i, x_k \in X} |x_i - x_k|. \quad (1.6.6)$$

Особый интерес представляют собой A -графы, у которых множество X состоит из целых положительных чисел.

Определение 6. Граф $G = (X, U, F)$ называется *примитивным*, если у него $X = \{1, 2, 3, \dots, n\}$.

Для числовых графов одной из основных является следующая задача.

Задача 1. Для данного графа G и функции F построить совершенный граф.

Другие задачи возникают как производные от этой задачи. В зависимости от вида F основная задача может быть решена не для всяких графов. Рассмотрим некоторые частные виды порождающих функций.

Квадратичная порождающая функция в общем виде запишется так:

$$F(x_i, x_j) = ax_i^2 + 2bx_ix_j^2 + cx_j^2 + 2dx_i + 2ex_j + f. \quad (1.6.7)$$

В качестве иллюстрации для дальнейшего изложения рассмотрим функцию

$$F(x_i, x_j) = 2x_j^2 - 3x_ix_j + x_i^2 - x_i + 1. \quad (1.6.8)$$

Зададим $X = \{1, 2, 3, 4, 5\}$ и $U = \{2\}$. По определению этот граф является примитивным. Ему соответствует граф на рис. 1.6.1. Действительно, легко проверить, что

$$F(2,1) = 2 \cdot 2^2 - 3 \cdot 2 \cdot 1 + 1^2 - 2 + 1 = 2;$$

$$F(1,3) = 2 \cdot 1^2 - 3 \cdot 1 \cdot 3 + 3^2 - 1 + 1 = 2;$$

$$F(3,2) = 2 \cdot 3^2 - 3 \cdot 3 \cdot 2 + 2^2 - 3 + 1 = 2;$$

$$F(2,5) = 2 \cdot 2^2 - 3 \cdot 2 \cdot 5 + 5^2 - 2 + 1 = 2; \quad (1.6.9)$$

$$F(4,3) = 2 \cdot 4^2 - 3 \cdot 4 \cdot 3 + 3^2 - 4 + 1 = 2;$$

$$F(5,4) = 2 \cdot 5^2 - 3 \cdot 5 \cdot 4 + 4^2 - 5 + 1 = 2.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A -графа состоит из одного элемента u_1 , то его можно приравнять нулю, а саму порождающую функцию заменить $F = (x_i, x_j) - u_1 = F'(x_i, x_j)$. Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0. \quad (1.6.10)$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}. \quad (1.6.11)$$

Определение 6. Граф G = (X, U, F) называется примитивным, если у него X = {1, 2, 3, ..., n}.

Для числовых графов одной из основных является следующая задача.

Задача 1. Для данного графа G и функции F построить совершенный граф.

Другие задачи возникают как производные от этой задачи.

В зависимости от вида F основная задача может быть решена не для всяких графов.

Рассмотрим некоторые частные виды порождающих функций.

Квадратичная порождающая функция в общем виде запишется так:

$$F(x_i, x_j) = ax_i^2 + 2bx_ix_j^2 + cx_j^2 + 2dx_i + 2ex_j + f.$$

В качестве иллюстрации для дальнейшего изложения рассмотрим функцию

$$F(x_i, x_j) = 2x_j^2 - 3x_ix_j + x_i^2 - x_i + 1.$$

Зададим X = {1, 2, 3, 4, 5} и U = {2}. По определению этот граф является примитивным.

Ему соответствует граф на рис. 1.6.1. Действительно, легко проверить, что

$$F(2,1) = 2 \cdot 2^2 - 3 \cdot 2 \cdot 1 + 1^2 - 2 + 1 = 2;$$

$$F(1,3) = 2 \cdot 1^2 - 3 \cdot 1 \cdot 3 + 3^2 - 1 + 1 = 2;$$

$$F(3,2) = 2 \cdot 3^2 - 3 \cdot 3 \cdot 2 + 2^2 - 3 + 1 = 2;$$

$$F(2,5) = 2 \cdot 2^2 - 3 \cdot 2 \cdot 5 + 5^2 - 2 + 1 = 2;$$

$$F(4,3) = 2 \cdot 4^2 - 3 \cdot 4 \cdot 3 + 3^2 - 4 + 1 = 2;$$

$$F(5,4) = 2 \cdot 5^2 - 3 \cdot 5 \cdot 4 + 4^2 - 5 + 1 = 2.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Из аналитической геометрии известно, что это уравнение кривой второго порядка. Важные значения для этих уравнений имеют инварианты, которые в соответствии с обозначениями (1.6.7) равны

$$I = a + c; \quad D = \begin{vmatrix} a & b \\ b & d \end{vmatrix}; \quad A = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f - u_k \end{vmatrix}.$$

Здесь граф имеет вид двух склеенных контуров. Очевидно, если множество образующих A-графа состоит из одного элемента u_1, то его можно приравнять нулю, а саму порождающую функцию заменить F = (x_i, x_j) - u_1 = F'(x_i, x_j). Тогда вопрос о смежности вершин x_i и x_j сводится к решению уравнений двух переменных

$$F'(x, y) = 0.$$

Рис. 1.6.1

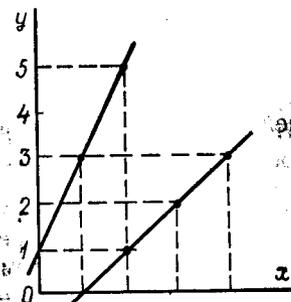


Рис. 1.6.2

Найдем их значения для нашего уравнения

$$2x^2 - 3xy + y^2 - x + 1 = 0; \quad (1.6.12)$$

$$A = \begin{vmatrix} 2 & -\frac{3}{2} & -\frac{1}{2} \\ -\frac{3}{2} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{vmatrix} = 0; \quad D = \begin{vmatrix} 2 & -\frac{3}{2} \\ -\frac{3}{2} & 1 \end{vmatrix} = -\frac{1}{4} < 0;$$

$$1 = 2 + 1 = 3. \quad (1.6.13)$$

Это означает, что уравнение (1.6.12) является вырожденным гиперболического типа, т. е. представляет собой уравнение пары пересекающихся прямых. Действительно, его можно записать в виде

$$(y - x + 1)(y - 2x - 1) = 0. \quad (1.6.14)$$

На графике ему соответствуют две прямые (рис. 1.6.2):

$$y_1 = x - 1; \quad y_2 = 2x + 1. \quad (1.6.15)$$

Как видно из графика, для множества вершин $X = \{1, 2, 3, 4, 5\}$ осуществляется смежность, которая соответствует ориентированному графу на рис. 1.6.1. Если множество образующих содержит более одного элемента, то для каждого u_k в (1.6.11) будут разными значения A . Это означает, что тип порождающей функции будет всегда один и тот же (эллиптический, параболический или гиперболический), но для некоторых u_k уравнение может быть вырожденным и представлять на графике либо одну точку, либо пару прямых (пересекающихся, параллельных или совпадающих).

Кроме квадратичной функции интерес представляют функции вида $F(x_i, x_j) = |x_j - x_i|$, $F(x_i, x_j) = \sin(x_i + x_j)$, а также алгебраические функции выше второго порядка. Самой простой порождающей функцией является линейная функция.

Линейная порождающая функция имеет вид

$$F(x_i, x_j) = ax_i + bx_j + c. \quad (1.6.16)$$

Пусть $\alpha \neq 0$. Обозначим $\alpha U = \{\alpha u_1, \alpha u_2, \dots, \alpha u_p\}$ и $\alpha X = \{\alpha x_1, \alpha x_2, \dots, \alpha x_n\}$, а через $\bar{\alpha} + X$ множество $\{\alpha + x_1, \alpha + x_2, \dots, \alpha + x_n\}$. Докажем ряд утверждений.

Лемма 1.6.1. *A-графы $G = (X, U, F)$ и $G' = (\alpha X', \alpha U', f)$ изоморфны, если выбрать $f(x_i, x_j) = F(x_i, x_j) + (\alpha - 1)c$.*

Действительно, пусть $F(x_i, x_j) = ax_i + bx_j + c = u_1 \in U$. Тогда $f(\alpha x_i, \alpha x_j) = F(\alpha x_i, \alpha x_j) + (\alpha - 1)c = \alpha(ax_i + bx_j) + c + (\alpha - 1)c = \alpha(ax_i + bx_j + c) = \alpha u_1$.

Очевидно, что $\alpha u_1 \in \alpha U$.

Лемма 1.6.2. *A-графы $G = (X, U, F)$ и $G' = (\bar{\alpha} + X, \alpha(a + b) + u, F)$ изоморфны.*

Проверяется непосредственно:

$$\begin{aligned} F(\alpha + x_i, \alpha + x_j) &= a(\alpha + x_i) + b(\alpha + x_j) + c = \\ &= ax_i + bx_j + \overline{\alpha(a + b)} + c. \end{aligned}$$

Очевидно, что если $ax_i + bx_j + c = u_1 \in U$, то $F(\alpha + x_i, \alpha + x_j) \in \overline{\alpha(a + b)} + u_1 \in U'$.

Лемма 1.6.3. *A-графы $G = (X, U, F)$ и $G' = (X, \bar{\alpha} + U, f)$ изоморфны, если $f(x_i, x_j) = F(x_i, x_j) + \alpha$.*

Проверяется непосредственно, так же как и следующая

Лемма 1.6.4. *A-графы $G = (X, U, F)$ и $G' = (X, \alpha U, \alpha F)$ изоморфны.*

A-графы с линейной порождающей функцией будут неориентированными, если $a = b$, т. е.

$$F(x, x) = a(x_i + x_j) + c. \quad (1.6.17)$$

В частности, если $a = 1$, $ac = 0$, получаем числовой граф, который носит название арифметического [5, 6].

Лемма 1.6.5. *Всякий неориентированный A-граф с линейной порождающей функцией изоморфен арифметическому графу.*

Действительно, возьмем в графе $G' = (X', U', F)$, $X' = X$,

$$U' = \frac{U - \bar{c}}{a}, \quad aF'(x_i, x_j) = \frac{F(x_i, x_j)}{a} - c = x_i + x_j.$$

Непосредственно проверяем, что если $F(x_i, x_j) \in U$, то $(x_i + x_j) \in U'$. Таким образом, изучение неориентированных A-графов с линейной порождающей функцией можно сводить к изучению арифметических графов, которые в некотором роде

являются более простыми. Однако для них существуют задачи, которые еще не решены.

Задача 1 решена для простейших арифметических графов, таких как цепь, цикл, $K_{3,3}$ и полный граф. В работах [5, 6] показано, что эти графы можно представить в виде примитивных совершенных графов, причем для цепи достаточно двух образующих, для циклов — трех (при n четном) или четырех (при n нечетном) образующих, для графа $K_{3,3}$ — пяти образующих и для полного графа — $(2n - 3)$ образующих. Однако уже для дерева эта задача пока не решена. В данной работе будет решена задача 1 для однородных деревьев степени p и ранга 2 и доказано, что при этом достаточно p образующих.

Рассмотрим линейную функцию $F(x_i, x_j) = x_i - x_j + 3$ и $U = \{0, 1, 4\}$. В качестве множества вершин выберем $X = \{1, 2, 3, 4, 5\}$. Легко проверить, что этот примитивный граф (см. рис. 1.6.1) состоит из шести дуг: $(5, 4)$, $(4, 3)$, $(3, 2)$, $(2, 1)$, $(1, 3)$, $(2, 5)$ и задан с помощью квадратичной функции.

Если сравнить «легко вычисляемые» квадратичную и линейную функции, то очевидно, что последняя намного проще. В первой число умножений равно 5, а число сложений — 4, тогда как во второй число операций сложения равно 2. Зато в первой функции только одно сравнение, а во второй каждый раз нужно производить поиск числа среди трех образующих. Если различным операциям дать веса, которые соответствовали бы реальной трудоемкости вычисления на ЭВМ, например, λ_1 — вес операции умножения, λ_2 — вес операции сравнения и λ_3 — вес операции сложения, то возникает следующая задача.

Задача 2. Для данного графа G найти такую порождающую функцию F , в которой используется c_1 умножений, c_2 сложений и c_3 сравнений, чтобы величина

$$T(F) = \sum_{i=1}^3 \lambda_i c_i \quad (1.6.18)$$

была минимальной.

Когда рассматриваем линейную функцию (1.6.16), то для проверки смежности вершин решаем уравнения

$$ax_i + bx_j + c = u_k \quad (k = 1, 2, \dots, n). \quad (1.6.19)$$

Изобразим на координатной плоскости параллельные прямые

$$ax + by + c - u_k = 0. \quad (1.6.20)$$

Из вершины x_i в x_j будет выходить дуга в том и только в том случае, когда точка с координатами $x = x_i$, $y = x_j$ принадлежит одной из прямых (1.6.20). Для арифметических графов уравнения (1.6.20) превращаются в следующие:

$$x + y = u_k. \quad (1.6.21)$$

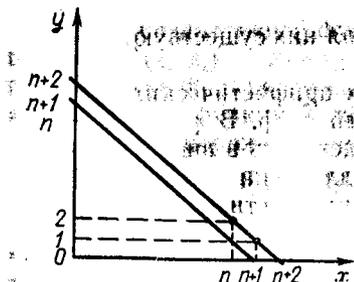


Рис. 1.6.3

$$x + y = n + 1; \quad x + y = n + 2. \quad (1.6.22)$$

Из графика следует, что первая прямая связывает вершины $(1, n)$, $(2, n - 1)$, $(3, n - 2)$ и т. д., что соответствует паросочетанию для цепи. Вторая прямая связывает вершины $(2, n)$, $(3, n - 1)$, $(4, n - 2)$ и т. д., что соответствует паросочетанию, сопряженному с первым паросочетанием. Это и дает нам способ нумерации вершин цепи при двух образующих. Следует заметить, что при этом для нечетных n в вершине $(n + 1)/2$ образуется петля.

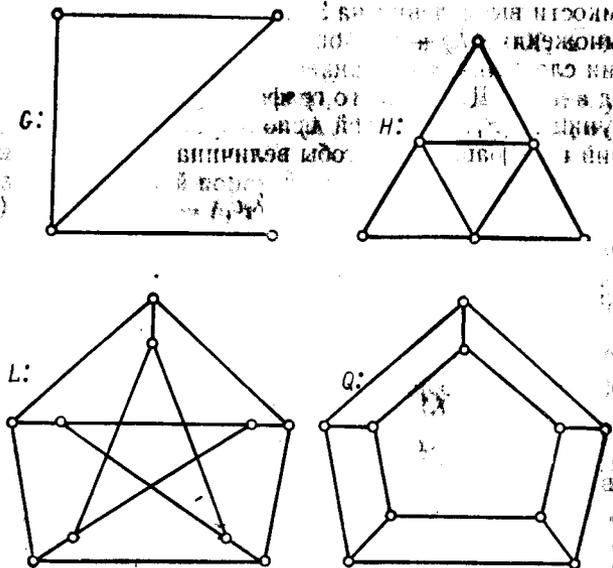


Рис. 1.6.4

Рассмотрим вопрос, как представить, например, цепь в виде совершенного примитивного А-графа. Пусть необходимо для цепи из n вершин найти минимальное число образующих. Очевидно, что их число $\rho \geq 2$. Покажем, что $\rho = 2$. Для этого проведем на координатной плоскости две прямые (рис. 1.6.3)

Упражнения.

- Нарисовать n -вершинный граф, у которого матрица смежностей A обладает тем свойством, что $a_{ij} = 1$, если $i + j = n + 1$, равна 0 для остальных a_{ij} .
- Нарисовать граф Берга для $X = \{x_1, x_2, x_3, x_4, x_5\}$: $Gx_1 = \{x_2, x_4\}$, $Gx_2 = \emptyset$, $Gx_3 = \{x_1, x_3, x_5\}$, $Gx_4 = \{x_5\}$, $Gx_5 = \{x_2, x_3, x_4\}$.
- Доказать, что в графе из 6 вершин найдутся три вершины, которые либо все не смежны, либо составляют треугольник.
- Двоичный код графа из 6 вершин равен 33. Найти его минимальный и максимальный двоичные коды.
- Доказать, что если в n -вершинном графе степень каждой вершины не меньше $(n - 1)/2$, то он связан.
- Построить матрицы смежности и инцидентности для правильных многогранников.
- Какие из графов правильных многогранников имеют гамильтоны цепи и циклы?
- Разбить квадрат на n^2 мелких квадратов прямыми, параллельными его сторонам. Имеют ли такие графы гамильтоны цепи и циклы?
- Имеется набор из четырех цепей длины $l_1 = 6$, $l_2 = 4$, $l_3 = 4$, $l_4 = 2$. Представить их в виде арифметического примитивного графа с помощью двух образующих.
- Задать в виде формальных квадратичных форм:
 - E_5 — безреберный (вполне несвязный или регулярный степени 0) граф;
 - P_5 — простую цепь с 5-ю вершинами;
 - C_5 — простой цикл с 5-ю вершинами;
 - F_5 — полный граф с 5-ю вершинами;
 - $F_{2,3}$ — полный двудольный с $2 + 3$ вершинами и его дополнение $\bar{F}_{2,3}$.
- Для каждого из графов, изображенных на рис. 1.6.4:
 - дать представление их в виде формальных квадратичных форм, предварительно пометив их вершины символами x_i ($i = \overline{1, p}$);
 - нарисовать дополнительные графы и представить их в виде формальных квадратичных форм.
- Нарисовать и задать все пары самодополнительных графов с 8-ю вершинами с помощью формальных квадратичных форм.

ПРОБЛЕМЫ ОПТИМАЛЬНОГО ПРЕДСТАВЛЕНИЯ ГРАФОВ

2.1. ПРЕДСТАВЛЕНИЕ ГРАФОВ С ПОМОЩЬЮ СТРУКТУР ДАННЫХ

При составлении программ для реализации различных алгоритмов пользуются структурами данных, среди которых наиболее простыми являются массивы. Для дальнейших целей необходимо ввести в рассмотрение другие виды структур данных, такие как списки и стеки, которые часто используются в эффективных алгоритмах.

С математической точки зрения список — это конечная последовательность элементов, взятых из некоторого подходящего множества. Описание алгоритма часто будет включать в себя некоторый список, к которому добавляются и из которого удаляются элементы. В частности, можем добавить или удалить элемент где-то в середине списка. По этой причине надо разработать структуры данных, позволяющие реализовать списки, в которых разрешается удалять и добавлять элементы так, как нам захочется.

Рассмотрим список ЭЛЕМ 1, ЭЛЕМ 2, ЭЛЕМ 3, ЭЛЕМ 4, ЭЛЕМ 5.

Простейшей его реализацией будет структура последовательно связанных компонент, изображенная на рис. 2.1.1. Каждая компонента в этой структуре состоит из двух ячеек памяти. Первая ячейка содержит сам элемент, вторая — указатель следующего элемента. Если элемент сам является сложной структурой, то первая ячейка может содержать указатель его местоположения.

Этот список можно реализовать в виде двух массивов, которые на рис. 2.1.2 названы ИМЯ и СЛЕДУЮЩАЯ. Пусть номер строки массива КОМПОНЕНТА, тогда ИМЯ [КОМПОНЕНТА] — сам элемент, хранящийся там, а СЛЕДУЮЩАЯ [КОМПОНЕНТА] — номер строки массива, где хранится следующий элемент (если такой элемент существует). Первый элемент характерен тем, что для него всегда КОМПОНЕНТА = 0, а последний тем, что для него СЛЕДУЮЩАЯ [КОМПОНЕНТА] = 0. Заметим, что порядок элементов в массиве ИМЯ

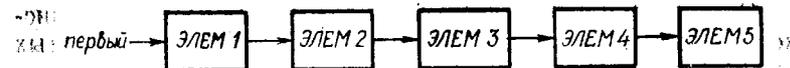


Рис. 2.1.1

	Имя	Следующая
0		2
1	ЭЛЕМ 2	5
2	ЭЛЕМ 1	6
3	ЭЛЕМ 4	4
4	ЭЛЕМ 5	0
5	ЭЛЕМ 3	3
	НОВЭЛЕМ	1

Рис. 2.1.2

	Имя	Следующая
0		2
1	ЭЛЕМ 2	5
2	ЭЛЕМ 1	1
3	ЭЛЕМ 4	4
4	ЭЛЕМ 5	0
5	ЭЛЕМ 3	3

Рис. 2.1.3

не совпадает с их порядком в списке. Тем не менее рис. 2.1.2 дает верное представление списка, изображенного на рис. 2.1.1, так как массив СЛЕДУЮЩАЯ располагает элементы в том же порядке, как они расположены в списке.

Опишем процедуру, вставляющую новую компоненту в список. В ней предполагается, что СВОБОДНАЯ — номер незанятой ячейки в массивах ИМЯ и СЛЕДУЮЩАЯ, а ПОЗИЦИЯ — номер той ячейки в списке, после которой надлежит вставить НОВЭЛЕМ (новый элемент).

ПРОЦЕДУРА ВСТАВИТЬ (НОВЭЛЕМ, СВОБОДНАЯ, ПОЗИЦИЯ)

НАЧАЛО

ИМЯ [СВОБОДНАЯ] = НОВЭЛЕМ;
СЛЕДУЮЩАЯ [СВОБОДНАЯ] = СЛЕДУЮЩАЯ [ПОЗИЦИЯ];
СЛЕДУЮЩАЯ [ПОЗИЦИЯ] = СВОБОДНАЯ

КОНЕЦ.

Вставим в список новый элемент после ЭЛЕМ 1, т. е. получим новый список: ЭЛЕМ 1, НОВЭЛЕМ, ЭЛЕМ 2, ЭЛЕМ 3, ЭЛЕМ 4, ЭЛЕМ 5.

Если шестая ячейка в каждом массиве на рис. 2.1.2 пуста, можно вставить НОВЭЛЕМ после ЭЛЕМ 1 (позиция 2), вызвав процедуру ВСТАВИТЬ (НОВЭЛЕМ, 6, 2). В результате выполнения трех операторов получим

ИМЯ [6] = НОВЭЛЕМ;
СЛЕДУЮЩАЯ [6] = 1;
СЛЕДУЮЩАЯ [3] = 6.

В результате получим список, показанный на рис. 2.1.3. Для того чтобы удалить компоненту, следующую за компонентой 1, можно положить

СЛЕДУЮЩАЯ [1] = СЛЕДУЮЩАЯ [СЛЕДУЮЩАЯ[1]].

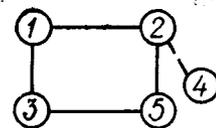
Часто в один и тот же массив вкладывается несколько списков. Обычно один из этих списков, состоящий из незанятых ячеек, называется свободным списком.

Эти две операции над списками — не единственные, которыми обычно пользуются для организации памяти. Есть еще несколько операций над списками — конкатенация (сцепление) двух списков, в результате которой образуется единственный список, и обратная к ней операция расщепления списка, стоящего после некоторого элемента, результатом которой будут два списка. Списки можно сделать проходимыми в обоих направлениях, если добавить еще один массив, называемый ПЕРЕДЫДУЩАЯ. Значение ПЕРЕДЫДУЩАЯ i равно ячейке, в которой помещается тот элемент списка, который стоит непосредственно перед элементом из ячейки i . Список такого рода называется дважды связанным. Из дважды связанного списка можно удалить элемент или вставить в него элемент, не зная ячейку, где находится предыдущий элемент.

Обычно списки применяются для представления множеств. При этом объем памяти, необходимой для представления множества, пропорционален числу его элементов. Время, требуемое для выполнения операции над множествами, зависит от природы операции.

Другой способ представления множества, отличный от представления его в виде списка — представление в виде двоичного (битового) вектора. Пусть U — универсальное множество (т. е. все рассматриваемые множества являются его подмножествами), состоящее из n элементов. Линейно упорядочим его. Подмножество $S \subseteq U$ представляется в виде вектора V_S из n битов, такого, что i -й разряд в V_S равен 1 тогда и только тогда, когда i -й элемент множества U принадлежит S . Будем называть V_S характеристическим вектором для S .

Для представления графов существует несколько способов. Один из них — матрица смежностей, т. е. для заданного графа $G = (X, U)$ составляется матрица $A = (a_{i,j})$ размером $|X| \times |X|$, состоящая из 0 и 1, в которой $a_{i,j} = 1$ тогда и только тогда, когда из вершины i в вершину j заходит дуга. Представление в виде матрицы смежностей удобно для тех алгоритмов на графах, которым часто нужно знать, есть ли в графе данное ребро, ибо время, необходимое для определения наличия ребра, фиксировано и не зависит от $|X|$ и $|U|$. Основным недостатком применения матрицы смежностей заключается в том, что она занимает память объема $|X|^2$ даже тогда, когда граф содержит число ребер порядка числа вершин. Уже начальное заполнение матрицы смежностей посредством естественной процедуры требует времени порядка $|X|^2$, что сводит на нет алгоритмы сложности порядка $|X|$ при работе с графами, со-



а

Вершины

	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	1	1
3	0	1	0	0	1
4	0	0	0	0	0
5	0	0	0	0	0

б

Дуги

	Конец	Следующий
1	6	
2	8	
3	10	
4	0	
5	0	
6	2	7
7	3	0
8	4	9
9	5	0
10	2	11
11	5	0

в

Рис. 2.1.4

держащими число вершин порядка числа ребер. Хотя разработаны методы для преодоления этой трудности, почти неизбежно возникают другие проблемы, которые приводят к тому, что алгоритмы сложности порядка $|X|$, основанные на работе с матрицей смежностей, встречаются редко.

Интересной альтернативой является представление строк и столбцов матрицы смежностей в виде двоичных векторов. Такое представление может способствовать значительной эффективности алгоритмов на графах.

Еще одно возможное представление графа с помощью списков. Списком смежностей для вершины x называется список всех вершин y , смежных с x . Граф можно представить с помощью списков смежности из каждой вершины.

На рис. 2.1.4 изображен ориентированный граф, содержащий пять вершин (а), его представление в виде матрицы смежностей (б) в табличное представление списков смежностей (в).

Каждая из первых пяти ячеек в массиве СЛЕДУЮЩИЙ содержит указатель на первую вершину списка смежностей, а именно СЛЕДУЮЩИЙ $[i]$ указывает на первую вершину списка смежностей для вершины i . Заметим, что СЛЕДУЮЩИЙ $[4] =$ СЛЕДУЮЩИЙ $[5] = 0$, поскольку из вершин 4 и 5 дуги не выходят. Остальные составляющие массива СЛЕДУЮЩИЙ представляют дуги графа. Массив КОНЕЦ содержит вершины из списков смежностей. Таким образом, список смеж-

ностей вершины 1 начинается в ячейке 6 , ибо СЛЕДУЮЩИЙ $[1] = 6$, КОНЕЦ $[6] = 2$; это показывает, что есть дуга $(1, 2)$. Равенства СЛЕДУЮЩИЙ $[6] = 7$ и КОНЕЦ $[7] = 3$ означают, что есть дуга $(1, 3)$, а СЛЕДУЮЩИЙ $[7] = 0$ — что больше нет дуг, исходящих из вершины 1 .

2.2. ПРЕДСТАВЛЕНИЕ ДЕРЕВЬЕВ

Рассмотрим очень важный вид графов — деревья и изучим те структуры данных, пригодные для их представления, которые часто встречаются в вычислительных алгоритмах.

Определим формально дерево как конечное множество T , состоящее из одной или более вершин, таких, что а) имеется одна специально обозначенная вершина, называемая *корнем дерева*; б) остальные вершины (исключая корень) содержатся в $m \geq 0$ попарно непересекающихся множествах T_1, T_2, \dots, T_m , каждое из которых в свою очередь является деревом. Деревья T_1, T_2, \dots, T_m называются *поддеревьями* данного корня.

Процедуру, которая прямо или косвенно обращается к себе, называют *рекурсивной*. Данное определение дерева является рекурсивным, так как мы определили дерево в терминах самих же деревьев. Разумеется, никакого порочного круга здесь не возникает, поскольку деревья с $n > 1$ вершинами определяются с использованием понятия дерева с количеством вершин, меньшим, чем n ; следовательно, определение дает понятия деревьев с двумя вершинами, тремя вершинами, в конечном итоге с любым числом вершин. Можно дать и нерекурсивное определение дерева [1, 2], но рекурсивное определение кажется наиболее подходящим, так как рекурсивность является естественной характеристикой структур типа дерева. Основываясь на рекурсивном определении, легко дать строгие доказательства многих важных фактов относительно деревьев, используя индукцию по числу вершин в дереве. Граф, состоящий из нескольких деревьев, называется *лесом*. Леса и деревья — столь часто встречающиеся частные случаи графов, что для описания их свойств стоит ввести специальную терминологию.

Из определения следует, что каждая вершина дерева является корнем некоторого поддерева, которое содержится в этом дереве. Число поддеревьев данной вершины называется *степенью* этой вершины. Вершина с нулевой степенью называется *концевой вершиной*, или *листом*. Глубина вершины x в дереве — это длина пути из корня в x . Высота вершины x в дереве — это длина самого длинного пути из x в какой-нибудь лист. *Высотой* дерева называется высота его корня. *Упорядоченным деревом* называется дерево, в котором множество поддеревьев каждой вершины упорядочено. При изображении упорядо-

ченного дерева будем считать, что множество поддеревьев каждой вершины упорядочено слева направо.

Двоичным (бинарным) деревом называется такое упорядоченное дерево, что 1) каждое поддерево произвольной вершины идентифицируется либо как левое поддерево, либо как правое поддерево; 2) каждая вершина имеет не более одного левого поддерева и не более одного правого поддерева.

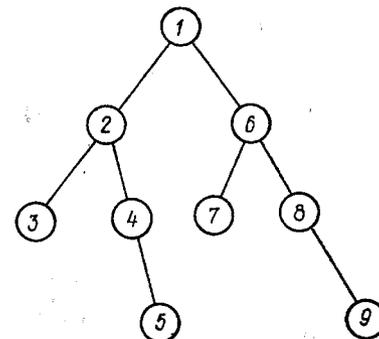


Рис. 2.2.1

Двоичное дерево называется *полным*, если для некоторого целого числа k каждая вершина глубины меньшей k имеет как левое, так и правое поддерево и каждая вершина глубины k является листом. Полное двоичное дерево глубины k имеет ровно $2^{k+1} - 1$ вершин.

Полное двоичное дерево высоты k часто представляют одним массивом (рис. 2.2.1). В позиции 1 этого массива находится корень. Корень левого поддерева вершины i расположен в позиции $2i$, а корень его правого поддерева — в позиции $2i + 1$.

Многие алгоритмы, использующие деревья, часто проходят дерево (посещают каждую его вершину) в некотором порядке. Известно несколько систематических способов сделать это. Рассмотрим три способа прохождения дерева: в прямом, обратном и внутреннем порядках.

Обозначим корень дерева T через r , а корни его поддеревьев через v_1, v_2, \dots, v_m ($m \geq 0$). При $m = 0$ дерево состоит из единственной вершины.

Прохождение дерева T в прямом порядке определяется следующей рекурсией: 1) посетить корень r ; 2) посетить в прямом порядке поддерева с корнями v_1, v_2, \dots, v_m в указанной последовательности.

Прохождение дерева T в обратном порядке определяется следующей рекурсией: 1) посетить в обратном порядке поддерева с корнями v_1, v_2, \dots, v_m в указанной последовательности;

2) посетить корень r .

Прохождение двоичного дерева во внутреннем порядке определяется следующей рекурсией: 1) посетить во внутреннем порядке левое поддерево корня (если оно существует);

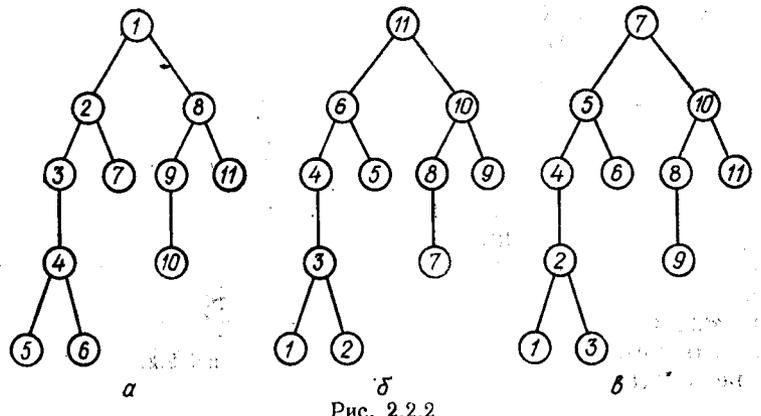


Рис. 2.2.2

2) посетить корень; 3) посетить во внутреннем порядке правое поддерево корня (если оно существует).

На рис. 2.2.2 изображено двоичное дерево, вершины которого пронумерованы в соответствии с прохождением его в прямом (а), обратном (б) и внутреннем (в) порядках.

Если при некотором прохождении дерева его вершинам были присвоены какие-то номера, то на вершины удобно ссылаться по этим номерам. Так, v будет означать вершину, которой был присвоен номер v .

Если все вершины занумерованы в порядке посещения, то рассмотренные нумерации обладают рядом интересных свойств.

При нумерации в прямом порядке все вершины поддерева с корнем r имеют номера не меньше r . Точнее, если D_r — множество вершин поддерева вершины r , то v будет номером некоторой вершины из D_r , тогда и только тогда, когда $r \leq v < r + |D_r|$.

Поставив в соответствие каждой вершине v ее номер в прямом порядке и количество вершин ее поддеревьев, легко определить, является ли некоторая вершина w частью поддеревьев вершины v .

После того, как номера присвоены (в соответствии с прямым порядком) и вычислено число вершин поддеревьев каждой вершины, на вопрос, является ли w потолком для v , можно ответить за фиксированное время, не зависящее от размера дерева. Номера, соответствующие обратному порядку, обладают аналогичным свойством.

Номера вершин двоичного дерева, соответствующие внутреннему порядку, обладают тем свойством, что и номера вершин в левом поддереве для w меньше v , а в правом поддереве больше v .

При разработке эффективных алгоритмов необходимо учитывать время, которое используется для реализации этих алгоритмов. Иногда ведется подсчет числа операций, которые применяются при этом. Эти операции связаны с обработкой различных структур данных, а также вызовов и возвращений различных процедур. Время, требуемое для вызова процедуры, пропорционально времени, требуемому для вычисления значений фактических параметров и запоминания указателей их значений в массиве. Время возвращения не превосходит этого времени. При подсчете времени, затрачиваемого несколькими рекурсивными процедурами, обычно легче всего оценить вес вызова процедуры, осуществляющей этот вызов. Тогда можно оценить сверху, как функцию от размера входа, разность времени, затрачиваемого на вызов каждой процедуры, и времени, затрачиваемого теми процедурами, которые она вызывает. Суммируя эти оценки по всем вызовам процедур, получаем верхнюю границу общего затраченного времени.

Для подсчета времени работы алгоритма с рекурсией применяются рекуррентные уравнения. С i -й процедурой связывается функция $T_i(n)$, обозначающая время выполнения i -й процедуры как функцию некоторого параметра n рассматриваемого входа. Обычно рекуррентное уравнение для $T_i(n)$ можно записать в терминах времен выполнения процедур, вызываемых процедурой i . Затем полученная система рекуррентных уравнений решается. Часто в алгоритме фигурирует только одна процедура, и $T_i(n)$ зависит только от $T_i(m)$ для конечного числа аргументов m меньших n .

Для решения той или иной задачи ее часто разбивают на части, находят их решения и затем из них получают решение всей задачи. Этот прием, особенно если его применять рекурсивно, часто приводит к эффективному решению задачи, подзадачи которой представляют собой ее меньшие версии.

Для иллюстрации этого способа рассмотрим задачу о нахождении наибольшего и наименьшего элементов множества S , содержащего n элементов. Для простоты будем считать, что n есть степень числа 2. Очевидный путь поиска наибольшего и наименьшего элементов состоит в том, чтобы искать их по отдельности. Например, процедура поиска наибольшего элемента состоит в поочередном сравнении всех элементов из S с оставлением большего. При этом будет произведено $n - 1$ сравнение. Аналогично можно найти наименьший элемент среди $n - 1$ оставшихся, произведя $n - 2$ сравнения. Итак, для нахождения наибольшего и наименьшего элементов при $n \gg 2$ потребуется $2n - 3$ уравнений.

Очевидно, что это неэффективный алгоритм. Применяя прием разбиения, можно разделить множество S на две равные части S_1 и S_2 . Тогда описанный выше алгоритм нашел бы наибольший и наименьший элементы в каждой из двух половин с помощью рекурсии.

Наибольший и наименьший элементы множества S можно было бы определить, произведя еще два сравнения — наибольших элементов в S_1 и S_2 и наименьших элементов в них. Теперь можно подсчитать число сравнений в этом алгоритме:

$$T(n) = \begin{cases} 1 & \text{при } n = 2; \\ 2T(n/2) + 2 & \text{при } n > 2. \end{cases}$$

Решением этих рекуррентных уравнений служит функция $T(n) = 3/2 n - 2$.

Легко проверить, что эта функция справедлива при $n = 2$. Если она справедлива для $n = m$, то

$$T(2m) = 2\left(\frac{3}{2}m - 2\right) + 2 = \frac{3}{2}(2m) - 2.$$

Таким образом, индукцией по n доказано, что $T(n) = 3/2 n - 2$ удовлетворяет рекуррентным состояниям, если n есть степень числа 2.

Рассмотрим умножение двух n -разрядных двоичных чисел. Традиционный метод требует порядка n^2 битовых операций. Пусть x и y — два n -разрядных двоичных числа; n есть степень числа 2. Разобьем числа x и y на две равные части $x = (a, b)$, $y = (c, d)$. Если рассматривать каждую из этих частей как $(n/2)$ -разрядное число, то можно представить произведение чисел x и y в виде

$$xy = (a2^{n/2} + b)(c2^{n/2} + d) = ac2^n + (ad + bc)2^{n/2} + bd.$$

На время забудем, что из-за переноса $a + b$ и $c + d$ могут иметь $n/2 + 1$ разрядов, и предположим, что они состоят лишь из $n/2$ разрядов. Наша схема для умножения двух n -разрядных чисел требует только трех умножений $(n/2)$ -разрядных чисел и нескольких сложений и сдвигов. Для вычисления произведений ac , ad и bc можно применить рекурсию. Сложения и сдвиги занимают время линейное относительно n . А временная сложность умножения двух n -разрядных чисел ограничена сверху функцией

$$T(n) = \begin{cases} k & \text{при } n = 1; \\ 3T(n/2) + kn & \text{при } n > 1, \end{cases}$$

где k — постоянная, отражающая сложение и сдвиги. Решением рекуррентных уравнений служит функция

$$T(n) = 3kn^{\log_3 3} - 2kn.$$

Доказательство проведем индукцией по n , где n — степень числа 2. Случай $n = 1$ тривиален. Если функция верна для $n = m$, то $T(2m) = 3T(m) + 2km = 3[3km^{\log_3 3} - 2km] + 2km = 3k(2m)^{\log_3 3} - 2k(2m)$.

Этот асимптотически быстрый алгоритм умножения целых чисел можно применять не только к двоичным, но и к десятичным числам. Например,

$$\begin{aligned} x &= 3141; & a &= 31; & b &= 41; \\ y &= 5927; & c &= 59; & d &= 27; \\ a + b &= 72; & c + d &= 86. \end{aligned}$$

Произведем необходимые вычисления:

$$\begin{aligned} u &= (a + b)(c + d) = 72 \cdot 86 = 6192; \\ v &= ac = 31 \cdot 59 = 1829; \\ w &= bd = 41 \cdot 27 = 1107; \\ xy &= 1829 \cdot 10^4 + (6129 - 1829 - 1107) \cdot 100 + 1107 = \\ &= 18616707. \end{aligned}$$

Заметим, что здесь одно умножение заменено тремя сложениями и вычитанием. Это основано на том, что умножение выполнять труднее, чем сложение, и для достаточно больших n любое число n -разрядных сложений требует меньше времени, чем n -разрядное умножение, независимо от того, какой из алгоритмов применяется.

2.4. ОБ ОПТИМАЛЬНОЙ КОДИРОВКЕ АРИФМЕТИЧЕСКИХ ГРАФОВ

Рассмотрим примитивные арифметические графы и решим для них задачу 1. Если вершинам графа произвольно присвоить номера от 1 до n , то число $i + j$, полученное для смежных вершин $i + j$, не всегда может служить элементом множества образующих, так как при этом автоматически возникают ребра $(1, i + j - 1)$, $(2, i + j - 2)$ и т. д., среди которых некоторые окажутся лишними. Поэтому полагаем, что вершины графа пронумерованы произвольно от 1 до n , а для решения задачи 1 нам необходимо присвоить каждой вершине i числа (коды) x_i , которые и определяют оптимальное решение. Таким образом, оптимальное кодирование в классе примитивных графов есть некоторое отображение множества $X = \{1, 2, \dots, n\}$ в себя. Следует заметить, что минимальный примитивный граф, который автоматически является совершенным, может не быть минимальным в классе графов с натуральным множеством X , и что не всякий граф, как бы мы ни задавали U , может быть примитивным.

Определение 1. Для примитивного графа $G = (X, U)$ отображение $g: X \rightarrow X$ называется эквивалентным преобразованием, если при этом не изменяется мощность множества образующих.

Рассмотрим n -вершинную цепь, вершины которой занумерованы последовательно.

Теорема 2.4.1. Для того чтобы цепь была закодирована с помощью двух образующих, необходимо и достаточно, чтобы коды ее четных вершин образовали возрастающую арифметическую прогрессию, а коды нечетных вершин — убывающую арифметическую прогрессию, причем обе прогрессии с одним и тем же знаменателем.

Необходимость. Пусть цепь закодирована множеством чисел $\{x_i\}$ ($i = 1, 2, \dots, n$), а ее образующие равны u_1 и u_2 . Не нарушая общности, положим $u_1 < u_2$. Составим отношения для смежных вершин

$$\begin{aligned} x_1 + x_2 &= u_1; \\ x_2 + x_3 &= u_2; \end{aligned} \quad (2.4.1)$$

$$x_3 + x_4 = u_1;$$

$$\dots$$

Отсюда находим

$$\begin{aligned} u_2 - u_1 &= x_3 - x_1; \\ u_2 - u_1 &= x_2 - x_4; \end{aligned} \quad (2.4.2)$$

$$u_2 - u_1 = x_5 - x_3;$$

$$\dots$$

Для вершин с нечетными номерами получаем зависимость

$$\begin{aligned} x_3 - x_1 &= x_5 - x_3 = x_7 - x_5; \\ 2x_3 &= x_1 + x_5; \\ x_3 &= \frac{x_1 + x_5}{2}. \end{aligned} \quad (2.4.3)$$

Или в общем случае

$$x_{2k-1} = \frac{x_{2k-3} + x_{2k+1}}{2} \quad (k = 2, 3, \dots). \quad (2.4.4)$$

Аналогично для вершин с четными номерами получаем общую зависимость

$$x_{2k} = \frac{x_{2k-2} + x_{2k+2}}{2} \quad (k = 2, 3, \dots). \quad (2.4.5)$$

Соотношения (2.4.4), (2.4.5) характеризуют арифметическую прогрессию со знаменателем $d = u_2 - u_1$. Учитывая то, что

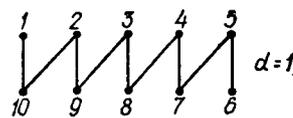


Рис. 2.4.1

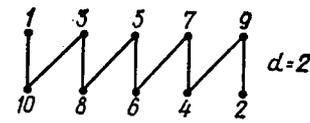


Рис. 2.4.2

$x_{2k+1} > x_{2k-1}$, а $x_{2k+2} < x_{2k}$, приходим к необходимому утверждению.

Достаточность. Пусть $\{x_i\}$ удовлетворяет требованию арифметической прогрессии

$$\begin{cases} x_{2k+1} = x_{2k-1} + d; \\ x_{2k+2} = x_{2k} - d \end{cases} \text{ или } \begin{cases} x_{2k+1} = x_1 + kd; \\ x_{2k+2} = x_2 - kd. \end{cases} \quad (k > 1) \quad (2.4.6)$$

Покажем, что эти соотношения задают кодировку вершин цепи с помощью двух образующих. Зафиксируем сумму кодов первой и второй вершины

$$x_1 + x_2 = u_1. \quad (2.4.7)$$

Отсюда $x_2 + x_3 = x_2 + (x_1 + d) = u_1 + d$. В общем случае

$$\begin{cases} x_{2k+1} + x_{2k+2} = x_1 + kd + x_2 - kd = x_1 + x_2 = u_1; \\ x_{2k+2} + x_{2k+3} = x_2 + kd + x_1 - kd + d = u_1 + d. \end{cases} \quad (2.4.8)$$

Таким образом, образующие цепи равны u_1 и $u_1 + d$, что и требовалось доказать.

Покажем теперь, что для цепей существует несколько кодировок с помощью двух образующих в классе примитивных графов.

Присвоим первой вершине цепи код $x_1 = 1$. Поскольку количество четных и нечетных вершин цепи отличается не более чем на единицу, то и соответствующие им арифметические прогрессии должны быть равновеликими по длине. Множество чисел от 1 до n можно разбить на две равновеликие арифметические прогрессии только двумя способами:

$$A. 1, 2, 3, \dots, \left\lfloor \frac{n}{2} \right\rfloor \text{ и } \left\lfloor \frac{n}{2} \right\rfloor + 1, \left\lfloor \frac{n}{2} \right\rfloor + 2, \dots, n, \quad (2.4.9)$$

$$B. 1, 3, 5, \dots, 2 \left\lfloor \frac{n-1}{2} \right\rfloor + 1 \text{ и } 2, 4, 6, \dots, 2 \left\lfloor \frac{n}{2} \right\rfloor.$$

Получаем две арифметические прогрессии со знаменателями $d = 1$ и $d = 2$. Это дает нам два способа кодирования цепи (рис. 2.4.1 и 2.4.2 для $n = 10$). При этом получаются образующие для вертикальных ребер $u_1 = n + 1$ и для наклонных ребер $u_2 = n + 2$.

Здесь для вертикальных ребер получаем образующие $u_1 = n + 1$, а для наклонных — $u_2 = n + 3$.

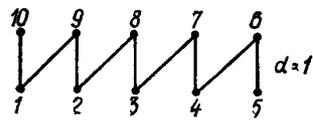


Рис. 2.4.3

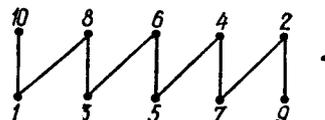


Рис. 2.4.4

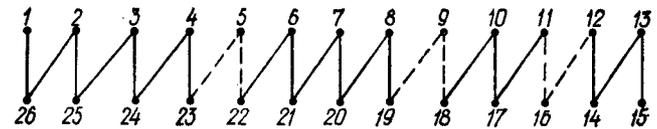


Рис. 2.4.5

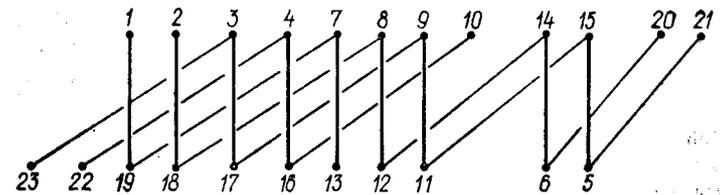


Рис. 2.4.6

Лемма 2.4.1. Преобразование множества вершин для цепи $x_i \leftrightarrow n + 1 - x_i$ является эквивалентным.

Действительно, пусть $x_i + x_j = u$. Тогда в преобразованном множестве получим

$$(n + 1 - x_i) + (n + 1 - x_j) = 2n + 2 - (x_i + x_j) = 2n + 2 - u.$$

Следовательно, при данном преобразовании множество образующих изменится по правилу

$$u'_i = 2n + 2 - u_i. \quad (2.4.10)$$

При этом мощности этих множеств одинаковы. Применяя это преобразование к цепи, получим еще две эквивалентные кодировки (рис. 2.4.3 и 2.4.4). Вертикальным ребрам соответствует образующая $u_1 = n + 1$, а наклонным ребрам — образующая $u_2 = n$. Образующие равны $u_1 = n + 1$, $u_2 = n - 1$.

Остальные кодировки повторяют полученные с помощью перестановок начала и конца цепи. Эти четыре кодировки получены для четного n . В случае нечетного n получим только три кодировки. Первые два случая, как для четного n , будут ($d = 1$; $u_1 = n + 1$, $u_2 = n + 2$) и ($d = 2$; $u_1 = n$, $u_2 = n + 1$), затем эквивалентное преобразование даст вариант ($d = 1$; $u_1 = n + 1$, $u_2 = n$), а второй вариант перейдет сам в себя ($d = 2$; $u'_1 = 2n + 2 - u_1 = n + 2$, $u'_2 = 2n + 2 - u_2 = n$).

При кодировании вершин нескольких цепей можно использовать тот же способ, что и при кодировании одной цепи. Для этого достаточно добавить к l цепям ($l > 1$, в каждой цепи n_i вершин) $l - 1$ фиктивных вершины, которые склеют все цепи в одну, и применить изложенный выше метод, как показано на рис. 2.4.5. Здесь имеются четыре цепи ($l = 4$; $n_1 = 8$; $n_2 = 7$; $n_3 = 4$; $n_4 = 4$). При решении добавлены 3 фиктивные вершины; они закодированы числами 5, 9 и 16. Такая кодировка задает минимальный граф с двумя образующими $u_1 = n + 1$ и $u_2 = n + 2$. Однако этот граф не является совершенным. Очевидно, если будет получен примитивный граф, то он будет совершенным. В данном случае это удастся сделать (рис. 2.4.6). Вертикальные ребра соответствуют образующей $u_1 = 20$, а наклонные ребра — образующей $u_2 = 26$. Цепи полу-

чают кодировку (3, 5, 9, 11, 15, 17, 21, 23), (2, 6, 8, 12, 14, 18, 20), (1, 7, 13, 19) и (4, 10, 16, 22).

Нетрудно заметить, что для каждой отдельно взятой цепи выполняются условия теоремы 2.4.1, т. е. коды, соответствующие четным (или нечетным) номерам вершин, составляют возрастающую (или убывающую) арифметическую прогрессию. Общий знаменатель этих прогрессий равен $d = 6$. Возникает вопрос, при каких значениях d можно закодировать цепи с помощью двух образующих. Для одной цепи, как указывает теорема, $d = 1$; 2.

Лемма 2.4.2. Для $d = 1$ не существует примитивного графа с двумя образующими, представляющего несколько цепей.

Прежде всего заметим, что для двух образующих справедливо

$$u_1 \leq n + 1 \leq u_2. \quad (2.4.11)$$

Левое неравенство достаточно для того, чтобы вершина с кодом 1 не была изолированной, а правое — чтобы не была изолированной вершина с кодом n . Допустим, от противного, что нам удалось закодировать больше одной цепи для $d = 1$. Возьмем три последовательные вершины x_k , x_{k+1} и x_{k+2} . Для них справедливо

$$x_k + x_{k+1} = u_1, \quad x_{k+1} + x_{k+2} = u_2.$$

Так как $\|x_{k+2} - x_k\| = d$, то $|u_2 - u_1| = 1$. При условии (2.4.11) это может быть только в двух случаях: а) $u_1 = n + 1$; $u_2 = n + 2$; б) $u_1 = n$; $u_2 = n + 1$.

Случай «б» получается из «а» при эквивалентном преобразовании (лемма 2.4.1) и его можно не рассматривать. Подсчитаем, скольким ребрам соответствуют образующие из случая «а», u_1 соответствует ребрам (1, n), (2, $n - 1$), (3, $n - 2$) и т. д.

При n четном число таких ребер равно $(n/2)$, при n нечетном оно равно $(n - 1)/2$. Аналогично для u_2 получаем, что 2 при n четных число соответствующих ребер равно $n/2 - 1$, при n нечетных — $(n - 1)/2$. В каждом случае образующие задают в графе $n - 1$ ребро. Но для l цепей ($l > 1$) число ребер должно быть равно $n - l$. Это подтверждает справедливость леммы.

Для заданного d множество чисел от 1 до n однозначно разбивается на d классов с представителем $i \pmod{d}$ ($i = 0, 1, \dots, d - 1$). Чтобы закодировать одну цепь с помощью двух образующих, надо четным вершинам присвоить числа $i \pmod{d}$ в возрастающем порядке, а нечетным — $j \pmod{d}$ ($i \neq j$) в убывающем порядке. Отсюда можно сделать вывод, что $d \leq 2l$.

Рассмотрим вопрос, когда можно закодировать l цепей с помощью двух образующих при $d = 2l$. Ответ на него дает следующая теорема.

Теорема 2.4.2. С помощью двух образующих при $d = 2l$ можно закодировать ровно l цепей k способами, где $k = \left[\frac{n \pmod{d}}{2} \right] + 1$. Для каждого способа образующие имеют вид

$$u_1 = \left[\frac{n}{d} \right] d + 2i - 1; \quad u_2 = \left(\left[\frac{n}{d} \right] d + 1 \right) d + 2i - 1$$

$(i = 1, 2, \dots, k).$

Доказательство. Так как цепь составляется из чисел, принадлежащих двум разным классам вычетов по модулю d , то задача сводится к разбиению $2l$ классов на l пар, суммы которых одинаковы. Расположим числа от 1 до d в порядке возрастания

$$1, 2, \dots, d - 1; \quad (2.4.12)$$

равноудаленные от середины числа составляют l пар, сумма каждой из которых равна $d + 1 \equiv 1 \pmod{d}$. Поэтому первая образующая будет равна сумме двух чисел, первое из которых равно наименьшему представителю класса $1 \pmod{d}$, т. е. 1, а второе — наибольшему представителю класса $0 \pmod{d}$; это число равно $[n/d]d$.

Таким образом,

$$u_1 = \left[\frac{n}{d} \right] d + 1 \text{ и } u_2 = \left(\left[\frac{n}{d} \right] + 1 \right) d + 1.$$

Сделаем циклический сдвиг последовательности (2.4.12):

$$2, 3, 4, \dots, d - 1, d, 1. \quad (2.4.13)$$

Здесь одинаковые суммы равноудаленных от середины чисел равны $3 \pmod{d}$. Выбирая наименьшее число из класса $2 \pmod{d}$, т. е. 2, и наибольшее из класса $1 \pmod{d}$, т. е.

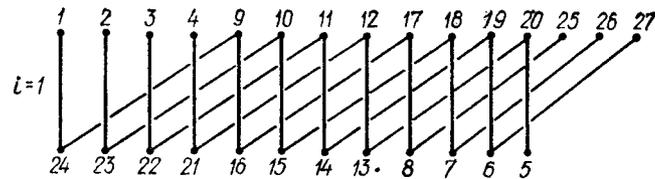


Рис. 2.4.7

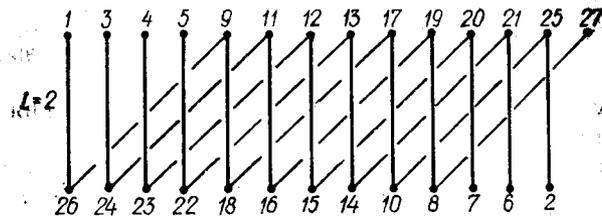


Рис. 2.4.8

$[n/d]d + 1$, получаем образующие

$$u_1 = \left[\frac{n}{d} \right] d + 3 \text{ и } u_2 = \left(\left[\frac{n}{d} \right] + 1 \right) d + 3.$$

При этом образующие увеличились на 2. Следовательно, после циклического сдвига последовательности (2.4.13) образующие еще увеличатся на 2. После k сдвигов исходной последовательности (2.4.12) образующие будут равны

$$u_1 = \left[\frac{n}{d} \right] d + 2k - 1; \quad u_2 = \left(\left[\frac{n}{d} \right] + 1 \right) d + 2k - 1 \quad (k > 1).$$

Очевидно, что эти образующие для любого k должны удовлетворять условию (2.4.11). Правое условие выполняется явно. Для u_1 необходимо $[n/d]d + 2k - 1 < n + 1$. Отсюда $2k < n - [n/d]d + 2$, или $\max k = \left[\frac{n \pmod{d}}{2} \right] + 1$, что и требовалось доказать.

В качестве иллюстрации рассмотрим способы кодировки четырех цепей для $n = 27$. Здесь $d = 8$. Разобьем числа от 1 до 27 на классы вычетов по модулю 8. В соответствии с (2.4.12) получим следующие разбиения: (1, 9, 17, 25), (2, 10, 18, 26), (3, 11, 19, 27), (4, 12, 20), (5, 13, 21), (6, 14, 22), (7, 15, 2), (8, 16, 24). Поскольку $27 \pmod{8} = 3$, то $k = [3/2] + 1 = 2$, т. е. существует 2 способа кодировки, которые приведены на рис. 2.4.7, где $u_1 = [27/8]8 + 2 \cdot 1 - 1 = 25$, $u_2 = 33$, $n_1 = n_2 = n_3 = 7$, $n_4 = 6$, и на рис. 2.4.8 — $u_1 = [27/8]8 + 22 - 1 = 27$, $u_2 = 35$, $n_1 = 8$, $n_2 = 7$, $n_3 = n_4 = 6$.

Очевидно, что для $d < 2l$ также возможна кодировка l цепей, как показано на рис. 2.4.6. Однако вопрос о том, какие

при этом должны быть ограничения на их размеры, остается открытым.

В примитивных графах любая образующая u задает столько ребер, сколько решений имеет уравнение

$$x + y = u \quad (2.4.14)$$

на множестве чисел $\{1, 2, \dots, n\}$.

Определение 2. Функция $r_n(u)$, которая равна числу решений уравнения (2.4.14) на множестве $\{1, 2, \dots, n\}$, называется частотой образующей u .

Нетрудно убедиться, что максимального значения $r_n(u)$ достигает при $u = \lfloor n + 1 \rfloor$. При этом для n четного $r_n(n + 1) = n/2$, а для n нечетного $r_n(n + 1) = (n - 1)/2$. В общем случае $r_n(n + 1) = \lfloor n/2 \rfloor$. Для u , меньших чем $n + 1$, легко видеть, что $r_n(u)$ не зависит от n . Тогда

$$r_n(u) = r_{n-1}(u) = \left\lfloor \frac{u-1}{2} \right\rfloor. \quad (2.4.15)$$

Для $u > n + 1$ можно воспользоваться эквивалентным преобразованием (2.4.10), т. е.

$$r_n(u) = r_n(2n + 2 - u) \quad (u > n + 1). \quad (2.4.16)$$

Тогда

$$r_n(u) = \left\lfloor \frac{2n + 1 - u}{2} \right\rfloor = n - \left\lfloor \frac{u}{2} \right\rfloor. \quad (2.4.17)$$

В общем случае

$$r_n(u) = \begin{cases} \left\lfloor \frac{u-1}{2} \right\rfloor, & u \leq n + 1; \\ n - \left\lfloor \frac{u}{2} \right\rfloor, & u > n + 1. \end{cases} \quad (2.4.18)$$

Очевидно, что в примитивном графе $3 \leq u \leq 2n + 1$. Пользуясь функцией $r_n(u)$, можно теперь для произвольного примитивного графа с m ребрами и n вершинами подбирать некоторые образующие так, чтобы выполнялось условие

$$\sum_{i=1}^p r_n(u_i) = m. \quad (2.4.19)$$

В этом смысле можно проверить правильность формул для образующих в теореме 2.4.2. Для $u_1 \leq n + 1$ имеем

$$r_n\left(\left\lfloor \frac{n}{d} \right\rfloor d + 2k - 1\right) = \left\lfloor \frac{\left\lfloor \frac{n}{d} \right\rfloor d + 2k - 2}{2} \right\rfloor = \left\lfloor \frac{n}{d} \right\rfloor 1 + k - 1.$$

Для $u_2 > n + 1$ имеем

$$r_n\left(\left\lfloor \frac{n}{d} \right\rfloor d + d + 2k - 1\right) = n - \left\lfloor \frac{\left\lfloor \frac{n}{d} \right\rfloor d + d + 2k - 1}{2} \right\rfloor = n - \left(\left\lfloor \frac{n}{d} \right\rfloor 1 + 1 + k - 1\right).$$

Число ребер в l цепях равно $n - l$. Проверая (2.4.19), получаем

$$\left\lfloor \frac{n}{d} \right\rfloor l + k - 1 + n - \left(\left\lfloor \frac{n}{d} \right\rfloor l + l + k - 1\right) = n - l.$$

Рассмотрим вопрос о кодировке циклов в классе примитивных графов. Так как в цикле n ребер, то закодировать граф с помощью двух образующих невозможно. Действительно, если бы существовали две такие образующие u_1 и u_2 , то тогда согласно (2.4.19) $r_n(u_1) + r_n(u_2) = n$. Используя (2.4.11) и выражение из (2.4.18), получаем

$$\left\lfloor \frac{u_1 - 1}{2} \right\rfloor + n - \left\lfloor \frac{u_2}{2} \right\rfloor = n;$$

$$\left\lfloor \frac{u_1 - 1}{2} \right\rfloor = \left\lfloor \frac{u_2}{2} \right\rfloor.$$

Но это невозможно, так как $u_2 > u_1$.

Теорема 2.4.3. Для кодирования циклов в классе примитивных графов достаточно: а) при нечетных n — четырех образующих; б) при четных n — трех образующих, которые обязательно имеют вид $(u_1, n + 1, n + u_1)$, где u_1 — некоторое нечетное число.

Доказательство. Рассмотрим возможность кодирования цикла тремя образующими независимо от значения n . Пусть эти образующие в порядке возрастания u_1, u_2 и u_3 . Вершины $1, 2, \dots, u_1 - 1$ соединены ребрами благодаря u_1 так, что образуют паросочетание. Так как в цикле степень каждой вершины равна 2, то эти вершины в отдельности должны соединяться еще с одной вершиной с помощью только одной образующей u_2 , но не u_3 . Это возможно только тогда, когда $u_3 \geq n + u_1$. Вершина u_1 смежна с двумя вершинами только благодаря образующим u_2 и u_3 . Отсюда $u_3 = n + u_1$.

Рассмотрим вершины $u_1 + 1, u_1 + 2, \dots, u_2 - 1$. Они также должны иметь степень 2 благодаря образующим u_2 и u_3 , и в то же время должны исчерпывать собой все вершины графа. Это может быть только тогда, когда $u_2 - 1 = n$, т. е. $u_2 = n + 1$. Так доказано, что вершины должны иметь вид $(u_1, n + 1, u_1 +$

+ n). Условие (2.4.19) запишем в виде

$$\left[\frac{u_1 - 1}{2} \right] + \left[\frac{n}{2} \right] + n - \left[\frac{n + u_1}{2} \right] = n. \quad (2.4.20)$$

Пусть n — нечетное. Тогда (2.4.20) переписывается так:

$$\left[\frac{u_1 - 1}{2} \right] + \left[\frac{n}{2} \right] - \left[\frac{n - 1}{2} \right] - \left[\frac{u_1 + 1}{2} \right] = 0$$

или

$$\left[\frac{u_1 - 1}{2} \right] = \left[\frac{u_1 + 1}{2} \right].$$

Это неверно для любого u_1 , поэтому при n нечетных трех образующих недостаточно.

Пусть n — четное. Тогда

$$\left[\frac{u_1 - 1}{2} \right] + \left[\frac{n}{2} \right] - \left[\frac{n}{2} \right] - \left[\frac{u_1}{2} \right] = 0$$

или $\left[\frac{u_1 - 1}{2} \right] = \left[\frac{u_1}{2} \right]$. Это возможно только при u_1 нечетном.

При n четном не всякое u_1 нечетное подходит в качестве образующей. Укажем образующие для четных и нечетных n , которые не зависят от n . Для n четных это эквивалентные наборы

$$(3, n + 1, n + 3) \text{ и } (n - 1, n + 1, 2n - 1). \quad (2.4.21)$$

Для n нечетных это эквивалентные наборы

$$(3, 4, n + 2, n + 4) \text{ и } (n - 2, n, 2n - 1, 2n - 2). \quad (2.4.22)$$

Надо сказать, что для некоторых значений n могут существовать и другие наборы. Так, для $n = 20$, кроме (2.4.21), существует пара эквивалентных наборов (7, 21, 27) и (15, 21, 35), а для $n = 17$, кроме (2.4.22), еще наборы (5, 8, 21, 35) и (11, 15, 28, 31).

В каждом отдельном случае задача оптимальной кодировки вершин того или иного графа сводится к поиску его совершенного представления. Для цепей и циклов эти задачи решены в [2, 3]. В настоящей работе решается аналогичная задача для однородных деревьев второго ранга.

Как известно, рангом дерева называется длина максимального пути от центра к периферийной (висячей) вершине. Рассмотрим однородные деревья степени p и ранга k . В качестве корня дерева выберем его-центр. Для $k = 1$ получаем звезду. Оптимальная кодировка для нее находится просто: все вершины кодируются от 1 до $p + 1$, это и будет совершенное представление. Пусть $k = 2$. Обозначим корень дерева через y_n ,

где n — число всех вершин (нетрудно подсчитать, что $n = p^2 + 1$), вершины первого яруса — y_i ($i = 1, 2, \dots, p$), остальные вершины второго (верхнего) яруса — y_j ($j = p + 1, p + 2, \dots, n - 1$).

Известно, что набор цепей можно закодировать с помощью двух образующих.

Сделаем правильную раскраску ребер графа. Эта раскраска требует p цветов и делается однозначно с точностью до их перестановки. Если поставить в соответствие каждой краске одну образующую, то возникает вопрос о возможности построения соответствующей оптимальной кодировки. Прежде, чем дать на него положительный ответ, рассмотрим одну проблему из теории чисел.

Задача 1. Для каждого $p \geq 2$ найти такие целые числа $x_p > x_{p-1} > \dots > x_1 = 1$, чтобы 1) для любых $1 \leq i, j \leq p$ ни одна разность $\|x_i - x_j\|$ не повторялась; 2) x_p минимальное.

В общем виде эта задача не решена, хотя для $p \leq 11$ с помощью ЭВМ удалось найти все такие числа. Для $2 \leq p \leq 4$ получается единственное решение, для больших значений p решение не единственное. В табл. 2.4.1 [4] перечислены все эти решения. Для $p \leq 7$ приводятся все решения, а дальше — только по одному варианту. В третьем столбце вычислены разности между соседними значениями последовательности $\{x_i\}$.

Таблица 2.4.1 Решения задачи 1

p	x_p	Разность	Последовательность $\{x_i\}$
2	2	1	1, 2
3	4	1, 2	1, 2, 4
4	7	1, 3, 2	1, 2, 5, 7
5	11	1, 3, 5, 2	1, 2, 5, 10, 12
6	18	2, 5, 1, 3	1, 3, 8, 9, 12
6	18	1, 3, 6, 2, 5	1, 2, 5, 11, 13, 18
		1, 3, 6, 5, 2	1, 2, 5, 11, 16, 18
		1, 7, 4, 2, 3	1, 2, 9, 12, 14, 18
		1, 7, 3, 2, 4	1, 2, 9, 13, 15, 18
		1, 3, 6, 8, 5, 2	1, 2, 5, 11, 19, 24, 26
7	26	1, 6, 4, 9, 3, 2	1, 2, 8, 12, 21, 24, 26
		1, 10, 5, 3, 4, 2	1, 2, 12, 17, 20, 24, 26
		2, 1, 7, 6, 5, 4	1, 3, 4, 11, 17, 22, 26
		2, 5, 6, 8, 1, 3	1, 3, 8, 14, 22, 23, 26
		1, 3, 5, 6, 7, 10, 2	1, 2, 5, 10, 16, 23, 33, 35
8	35	1, 4, 7, 13, 2, 8, 6, 3	1, 2, 6, 13, 26, 28, 36, 42, 45
		1, 5, 4, 13, 3, 8, 7, 12, 2	1, 2, 7, 11, 24, 27, 35, 42, 54, 56
11	73		56
		1, 3, 9, 15, 5, 14, 7, 10, 6, 2	1, 2, 5, 14, 29, 34, 48, 55, 65, 71, 73

На примере для $p = 6$ (первый вариант) будем строить дальнейшие рассуждения. Рассмотрим матрицу A размером $p \times p$, у которой все элементы вычисляются по формуле из табл. 2.4.1:

$$a_{i,j} = x_j - x_i. \quad (2.4.23)$$

По самому определению последовательности $\{x_i\}$ все элементы матрицы A , за исключением главной диагонали, различны. Над главной диагональю стоят элементы из третьего столбца таблицы.

Все остальные элементы в верхней части матрицы равны различным последовательным суммам элементов из этого столбца:

$$A = \begin{bmatrix} 0 & 1 & 4 & 10 & 12 & 17 \\ -1 & 0 & 3 & 9 & 11 & 16 \\ -4 & -3 & 0 & 6 & 8 & 13 \\ -10 & -9 & -6 & 0 & 2 & 7 \\ -12 & -11 & -8 & -2 & 0 & 5 \\ -17 & -16 & -13 & -7 & -5 & 0 \end{bmatrix}. \quad (2.4.24)$$

Рассмотрим теперь матрицу B :

$$B = \begin{bmatrix} 1 & 2 & 5 & 11 & 13 & 18 \\ 6 & 7 & 10 & 16 & 18 & 23 \\ 8 & 9 & 12 & 18 & 20 & 25 \\ 14 & 15 & 18 & 24 & 26 & 31 \\ 17 & 18 & 21 & 27 & 29 & 34 \\ 18 & 19 & 22 & 28 & 30 & 35 \end{bmatrix}. \quad (2.4.25)$$

В этой матрице в первой строке записана последовательность $\{x_i\}$. Во второй строке записаны те же элементы, увеличенные на последнее значение из строки разностей, т. е. на 5. Третья строка получена из второй путем увеличения ее на предпоследнее значение из строки разностей, т. е. на 2 и т. д. Как видно, в матрице все элементы, за исключением подчеркнутых, различны. Подчеркнутые элементы образуют диагональ и равны $\{x_p\}$. Относительно этой диагонали можно заметить закономерность

$$a_{i,j} + a_{n+1-j, n+1-i} = 2x_p. \quad (2.4.26)$$

Очевидно, что последовательность $\{x_i\}$, построенную для вектора разностей (1, 3, 6, 2, 5), (см. табл. 2.4.1), можно построить и для обратного вектора (5, 2, 6, 3, 1). Будем называть

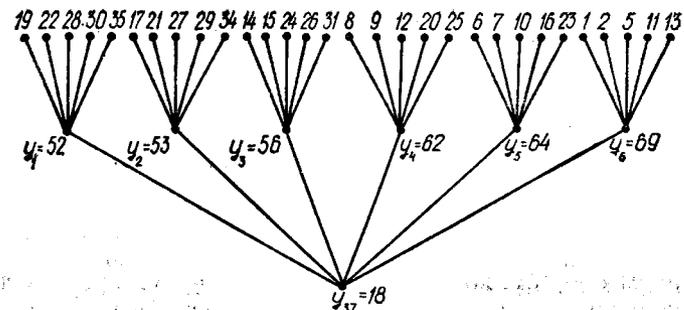


Рис. 2.4.9

такую последовательность двойственной. В матрице B ей соответствует первый столбец. Возьмем два произвольных элемента матрицы, одновременно не принадлежащие подчеркнутой диагонали, и покажем, что они не совпадают. Это очевидно, если они взяты из одной строки или столбца. Пусть их номера строк и столбцов не совпадают, т. е. сравниваем $a_{i,j}$ и $a_{k,l}$, где $l > j$. Возможны два случая:

$$1. i > k, \text{ тогда } a_{i,j} = a_{k,j} + d_1; \quad a_{k,l} = a_{k,j} + d_2.$$

Здесь d_1 и d_2 являются суммами последовательных элементов разностей, т. е. элементами матрицы A , что равносильно $a_{i,j} \neq a_{k,l}$.

$$2. i < k, \text{ тогда } a_{i,j} = a_{k,j} - d_1; \quad a_{k,l} = a_{k,j} + d_2.$$

Числа d_1 и d_2 обладают теми же свойствами, что и в случае 1. Здесь явно $a_{k,l} > a_{i,j}$, следовательно, $a_{i,j} \neq a_{k,l}$.

Вернемся к однородному дереву второго ранга для $p = 6$ (рис. 2.4.9).

Теорема 2.4.4. Для совершенного представления однородного дерева второго ранга степени p справедливо

$$S(G) = \max_{i,j} |y_i - y_j| \leq 4(x_p - 1). \quad (2.4.27)$$

Доказательство. Представим, что вершины дерева закодированы в соответствии с совершенным представлением. Сделаем перекодировку по правилу $y_i = y_i - y_n$ для всех вершин. Тогда $y_n = 0$ и образующие ребер, смежных с корнем, будут иметь вид $u_i = y_i$ ($i = 1, 2, \dots, p$). Те же образующие должны быть и для ребер, соединяющих вершины первого и второго ярусов. Коды висячих вершин, связанных с первой вершиной, имеют вид $y_k = y_i - y_1$; коды висячих вершин, связанных с j -й вершиной первого яруса ($j = 1, 2, \dots, p$), представлены так:

$$y_k = y_i - y_j \quad (j = i, k = p + 1, p + 2, \dots, n - 1).$$

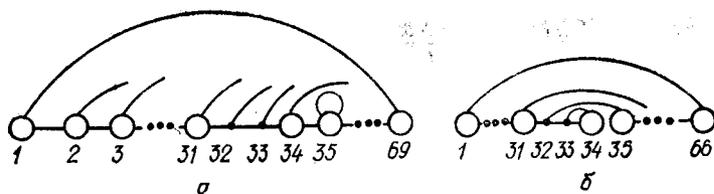


Рис. 2.4.10

Таким образом, коды висячих вершин будут представлять всевозможные разности между кодами первых p вершин, связанных с корнем дерева, в том числе и отрицательные. Набор кодов висячих вершин, связанных с j -й вершиной первого яруса, составляет j -ю строку матрицы A . Чтобы сделать все коды положительными, прибавим к ним число, равное $S(G) + 1$. Из табл. 2.4.1 известно, что оно равно 18. Таким образом, корень дерева приобретет код $y_n = 18$, а висячие вершины, связанные с j -й вершиной первого яруса, приобретут коды, соответствующие элементам строки матрицы B под номером $(7 - j)$, исключая подчеркнутые элементы, относящиеся к коду корня дерева.

Рассмотрим положительную числовую ось и нанесем на ней все значения из матрицы B . На рис. 2.4.10, *a* они обозначены кружками. Максимальный элемент матрицы B равен 35, из соотношения находим, что он равен $2x - 1$. Очевидно, что все y_i ($i = 1, 2, \dots, p$) должны быть больше 35, и y_p нужно выбрать так, чтобы $u_1 = y_p + 1$ и при этом чтобы не существовали два разных числа из матрицы B , сумма которых была бы равна u_1 . Если отметить на рис. 2.4.10 дугами все паросочетания, соответствующие образующей u_1 , то эти дуги не должны соединять два кружка. Самым крайним кружком должен быть кружок под номером 35. В качестве наименьшего u_1 можно выбрать число 70, т. е. дуга соединяет кружок под номером 35 с самим собою, затем числа 34 и 36, 33 и 37, и наконец, 1 и 69. Это означает, что в качестве y_p выбрано число 69, т. е. $y_p = 2(x_p - 1) - 1$. Так как наименьшее число $u_1 = 1$, то

$$S(G) \leq y_p - 1 = 4(x_p - 1), \quad (2.4.28)$$

что и требовалось доказать.

Зная y_p , вычитая последовательные разности, можно найти y_i и y_j :

$$\begin{aligned} y_1 &= 52, & y_2 &= 53, & y_3 &= 56, & y_4 &= 62, & y_5 &= 64, & y_6 &= 69, \\ u_1 &= 70, & u_2 &= 71, & u_3 &= 74, & u_4 &= 80, & u_5 &= 82, & u_6 &= 87. \end{aligned}$$

Эта кодировка реализована на рис. 2.4.9.

Благодаря теореме 2.4.4 получили верхнюю оценку для совершенного представления деревьев этого класса. Однако она не является точной. Иногда удается построить на основе табл. 2.4.1 представление, которое является совершенным.

На рис. 2.4.10, *б* слева от максимального, отмеченного кружком элемента, находятся две неотмеченные точки 32 и 33. Если соединить дугой элементы 32 и 35, 33 и 34, то появится возможность построить паросочетание, соответствующее образующей $u_1 = 67$. Для следующей образующей $u_2 = 68$ паросочетание также получится непротиворечивым — 33 и 35, 34 сама с собой. Тем самым построено совершенное представление:

$$\begin{aligned} y_1 &= 49, & y_2 &= 50, & y_3 &= 53, & y_4 &= 59, & y_5 &= 61, & y_6 &= 66, \\ u_1 &= 67, & u_2 &= 68, & u_3 &= 71, & u_4 &= 77, & u_5 &= 79, & u_6 &= 84. \end{aligned}$$

Здесь $S(G) = 65$. Это лучше того результата, который приводится на рис. 2.4.9, где $S(G) = 68$.

Теперь можно сделать общее описание процедуры построения совершенного представления. Сначала строим матрицу B в соответствии с табл. 2.4.1. На числовой оси отмечаем все элементы этой матрицы. Находим наименьшую неотмеченную точку в интервале $(x_p, 2x_p - 1)$ и строим паросочетание, соединяя эту точку с максимальным отмеченным элементом. Если оно не противоречиво, т. е. не соединяет двух отмеченных точек, то получаем значение u_1 , и если оно не противоречит остальным образующим, то получаем совершенное представление. Если на каком-то шаге получаем противоречивое паросочетание, то переходим к следующей неотмеченной точке. В конце концов нам гарантирован спектр графа, определяемый теоремой 2.4.4. Иногда, когда уже получено значение u_1 , а u_2 не удовлетворительно, можно перейти к двойственной последовательности, где u_2 может быть удовлетворительным. Продемонстрируем процедуру построения совершенных представлений для всей табл. 2.4.1:

$$p = 3$$

$$B = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 4 & 6 \\ 4 & 5 & 7 \end{pmatrix}. \quad (2.4.29)$$

Здесь отмеченные точки занимают весь интервал, поэтому

$$y_p = 4x_p - 3 = 4 \cdot 4 - 3 = 13,$$

$$y_1 = 10, \quad y_2 = 11, \quad y_3 = 13,$$

$$u_1 = 14, \quad u_2 = 15, \quad u_3 = 17;$$

$$p = 4$$

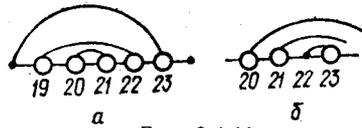


Рис. 2.4.11

$$B = \begin{bmatrix} 1 & 2 & 5 & 7 \\ 3 & 4 & 7 & 9 \\ 6 & 7 & 10 & 12 \\ 7 & 8 & 11 & 13 \end{bmatrix}. \quad (2.4.30)$$

Решение получаем на основании теоремы 2.4.4:

$$y_p = 4x_p - 3 = 4 \cdot 7 - 3 = 25, \\ y_1 = 19, \quad y_2 = 20, \quad y_3 = 23, \quad y_4 = 25, \\ u_1 = 26, \quad u_2 = 27, \quad u_3 = 30, \quad u_4 = 32.$$

$p = 5$. Здесь возможны два варианта:

$$B^{(1)} = \begin{bmatrix} 1 & 2 & 5 & 10 & 12 \\ 3 & 4 & 7 & 12 & 14 \\ 8 & 9 & 12 & 17 & 19 \\ 11 & 12 & 15 & 20 & 22 \\ 12 & 13 & 16 & 21 & 23 \end{bmatrix}. \quad (2.4.31)$$

Неотмеченным элементом, который нужно проверить, является 18. Если его соединить с максимальным — 23, то получится противоречивое паросочетание (рис. 2.4.11, а). Здесь кодировку улучшить по сравнению с той, которую нам гарантирует теорема 2.4.4, нельзя.

Рассмотрим второй вариант:

$$B^{(2)} = \begin{bmatrix} 1 & 3 & 8 & 9 & 12 \\ 4 & 6 & 11 & 12 & 15 \\ 5 & 7 & 12 & 13 & 16 \\ 10 & 12 & 17 & 18 & 21 \\ 12 & 14 & 19 & 20 & 23 \end{bmatrix}. \quad (2.4.32)$$

Неотмеченным элементом, который нужно проверить, является 22. Соединим его с максимальным — 23. В результате получаем непротиворечивое паросочетание (рис. 2.4.11, б).

что дает $u_1 = 45$ и $y_p = 44$. Окончательно получаем совершенное паросочетание

$$y_1 = 33, \quad y_2 = 35, \quad y_3 = 40, \quad y_4 = 41, \quad y_5 = 44, \\ u_1 = 45, \quad u_2 = 47, \quad u_3 = 52, \quad u_4 = 53, \quad u_5 = 56.$$

Для $p = 6$ построили совершенное представление для первого варианта. Еще такое же представление дает третий вариант. Второй и четвертый дают спектр по формуле (2.4.28).

Для $p = 7$ первые три варианта дают спектр по (2.4.28), $S(G) = 4(26 - 1) = 100$, но четвертый и пятый варианты позволяют улучшить спектр до $S(G) = 99$.

Для $p = 8$ получаем стандартное выражение для спектра $S(G) = 4(35 - 1) = 136$. Для $p = 9$ удается улучшить спектр $S(G) = 174$ по сравнению с формулой (2.4.28), где $S(G) = 4(45 - 1) = 176$. Для $p = 10, 11$ лучшее значение, чем по формуле (2.4.28), получить нельзя.

Все эти расчеты велись для $p = 8, 9, 10, 11$ по табличным данным. Но нельзя забывать, что в ней приведены единственные варианты последовательности $\{x_i\}$. Нет сомнения, что если проанализировать все варианты, то среди них нашлись бы такие, которые существенно улучшили бы спектры соответствующих деревьев до уровня совершенного представления.

Таким образом, для однородных деревьев второго ранга степени p показано, что для их представления достаточно p образующих, и с помощью табл. 2.4.1 можно находить совершенное представление. Однако для деревьев более высокого ранга вопрос о количестве образующих остается открытым.

Теорема 2.4.5. Для однородных деревьев третьего ранга степени представление с помощью образующих невозможно.

Доказательство будем вести от противного. Допустим, что такое представление с помощью p образующих получено. Пусть изображение и обозначение вершин дерева будет таким же, что и для деревьев второго ранга. Преобразуем коды: $y_k = y_k - y_n$. Тогда корень дерева приобретет код $y_n = 0$, а вершины первого яруса соответственно коды y_i ($i = 1, 2, \dots, p$), которые будут равны u , а вершины второго яруса — коды $y_i - y_j$ ($i \neq j$). Зафиксируем три вершины первого яруса y_k, y_l, y_m . Очевидно, среди вершин второго яруса, связанных с k -й вершиной, есть такая, код которой равен $y_i - y_k$, а среди вершин второго яруса, связанных с m -й вершиной, есть вершина с кодом $y_i - y_m$. Очевидно, среди вершин третьего яруса найдется вершина с кодом y^* , которая связана с первой указанной выше при помощи образующей, равной y_m , т. е. $y^* = y_m - (y_i - y_k)$.

Аналогично среди вершин третьего яруса найдется вершина с кодом y^{**} , которая связана со второй указанной выше с помощью образующей, равной y'_k , т. е. $y^{**} = y'_k - (y'_i - y'_m)$. Но тогда $y^* = y^{**}$, это должно противоречить условию представления графа. Следовательно, это дерево невозможно представить с помощью p образующих, что и требовалось доказать.

Упражнения.

1. Сделать табличное представление списков смежностей для полных 5-вершинного и 6-вершинного графов.

2. Выделить все максимальные поддеревья полного 6-вершинного графа и сделать прохождение их в прямом и обратном порядках.

3. Напишите алгоритмы для построения списков смежностей для неориентированного графа. Каждое ребро (u, v) надо представить дважды: в списках смежностей для u и для v . Оба экземпляра каждого ребра должны связываться между собой так, что если один из них вычеркивается, другой также легко вычеркнуть.

4. Рассмотрим двоичное дерево, вершинам которого приписаны имена. Напишите алгоритм, печатающий эти имена в (а) прямом, (б) внутреннем и (в) обратном порядках.

5. Подсчитайте число операций, необходимых для выполнения алгоритмов, описанных в п. 3, 4.

6. Напишите алгоритм для кодирования трех цепей длины n , $n + 1$, $[n/2]$ в виде арифметического графа с двумя образующими. Оцените временную сложность алгоритма.

ЭЛЕМЕНТЫ ТЕОРИИ СЛОЖНОСТИ ДЛЯ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧ НА ГРАФАХ

3.1. ОСНОВНЫЕ ПОНЯТИЯ

Математике всегда было присуще стремление разрабатывать эффективные методы решения как можно более широких классов задач. Многолетний опыт развития теории дискретных и комбинаторных задач и практика их решения показали, что эти две стороны — общность метода и его эффективность — находятся в известном противоречии. Вместе с тем очень важно знать (особенно это касается задач дискретной математики), можно ли в принципе надеяться на создание достаточно общих и эффективных методов или надо сознательно идти по пути разбиения задач на все более узкие классы и, пользуясь их спецификой, разрабатывать для них эффективные («хорошие») алгоритмы.

Неудачи, постигшие исследователей на этом пути, привели к необходимости анализа сложности задач. Анализ трудностей, возникших в процессе создания эффективных методов решения дискретных задач, обусловил постановку центральной теоретико-методологической проблемы всей дискретной математики — можно ли исключить перебор при решении дискретных задач? Другими словами, речь идет о принципиальной возможности найти нужное решение (оптимальное), не перебирая все или почти все варианты в задаче. Эта проблема имеет не только чисто математическое, но глубокое познавательное значение. Оно заключается в том, что при поиске эффективных точных методов решения широкого класса дискретных задач надо учитывать возможность отсутствия таких методов и, следовательно, уметь вовремя ограничить поиск, признав, что существуют «труднорешаемые задачи». До настоящего времени эта проблема остается открытой.

В переборных задачах, как правило, имеется конечное множество вариантов, среди которых нужно найти решение. Так, в задаче о рюкзаке решение отыскивается среди 2^n булевских векторов длины n , и перебирая это экспоненциальное множество векторов, обязательно решим задачу. Но с ростом n число векторов быстро растет и задача становится «труднорешаемой»,

т. е. практически неразрешимой. Поэтому в конечной области аналогом алгоритмической неразрешимости является перебор экспоненциального числа вариантов, а аналогом алгоритмической разрешимости — существование алгоритма более экономичного, чем перебор. Стало общепринятым считать переборную задачу решаемой эффективно, если имеется алгоритм, решающий ее за время, ограниченное полиномом от исходного параметра, называемого размером задачи.

В настоящее время разработана систематическая концепция эффективной (полиномиальной) сводимости переборных задач, в основу которой положены работы [1, 2, 3].

Для того чтобы в дальнейшем изучать такие понятия, как «труднорешаемые задачи» и «эквивалентные по сложности задачи», необходимо сначала договориться о значении нескольких основных терминов. Под массовой задачей (или просто задачей) будем понимать некоторый общий вопрос, на который следует дать ответ. Обычно задача содержит несколько параметров или свободных переменных, конкретные значения которых не определены. Задача P определяется следующей информацией: общим списком всех ее параметров; формулировкой тех свойств, которым должен удовлетворять ответ, или, другими словами, решение задачи. Индивидуальная задача I получается из массовой задачи P , если всем параметрам задачи P присвоить конкретные значения.

В качестве примера рассмотрим классическую задачу о коммивояжере. Параметры этой массовой задачи состоят из конечного набора городов $C = \{c_1, c_2, \dots, c_m\}$ и расстояний $d(c_i, c_j)$ между каждой парой городов c_i, c_j из C . Решением задачи является такой упорядоченный набор $\langle c_{\pi(1)}, \dots, c_{\pi(m)} \rangle$ заданных городов, который минимизирует величину

$$\sum_{i=1}^{m-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(m)}, c_{\pi(1)}).$$

Это выражение дает длину маршрута, начинающегося в городе $c_{\pi(1)}$, проходящего последовательно через все города и возвращающегося в $c_{\pi(1)}$ непосредственно из последнего города $c_{\pi(m)}$. Индивидуальная задача о коммивояжере, показанная на рис. 3.1.1, задается следующим образом: $C = \{c_1, c_2, c_3, c_4\}$; $d(c_1, c_2) = 10$; $d(c_1, c_3) = 5$; $d(c_1, c_4) = 9$; $d(c_2, c_3) = 6$; $d(c_2, c_4) = 9$; $d(c_3, c_4) = 3$.

Последовательность $\langle c_1, c_2, c_4, c_3 \rangle$ представляет собой решение задачи, поскольку соответствующий маршрут имеет минимальную возможную длину, равную 27.

Под алгоритмом понимаем общую, выполняемую шаг за шагом процедуру решения задачи. Для определенности можем считать ее программой для ЭВМ, написанной на формальном

машинном языке. Будем говорить, что алгоритм решает массовую задачу P , если он применим к любой индивидуальной задаче I из P и обязательно дает решение задачи I . Термин «решение» понимается здесь строго в соответствии с данным выше определением. Поэтому нельзя сказать, что алгоритм «решает» задачу о коммивояжере, если он не выдает маршрут минимальной длины хотя бы для какой-то одной индивидуальной задачи.

Вообще говоря, нам нужен наиболее «эффективный алгоритм» для решения задачи. В самом широком смысле понятие эффективности связано со всеми вычислительными ресурсами, необходимыми для работы алгоритма. Однако обычно под «самым эффективным» алгоритмом понимается самый быстрый. Поскольку ограничения по времени часто являются доминирующим фактором, определяющим пригодность конкретного алгоритма для практики, основное внимание сосредоточим главным образом на этом виде ресурсов.

Время работы алгоритма удобно выражать в виде функции от одной переменной, характеризующей «размер» индивидуальной задачи, т. е. объем входных данных, требуемых для описания этой задачи. Такой подход удобен, поскольку в дальнейшем сравнительная сложность задач будет оцениваться через их размеры. Часто размер задачи измеряется неформально. В задаче о коммивояжере, например, для этой цели обычно используется число городов. Однако в задаче с m городами кроме номеров этих городов на объем входной информации влияют также $m(m-1)/2$ величины, определяющие расстояние между городами, и значения этих величин. Если нам предстоит иметь дело с временными характеристиками в точной математической постановке, то необходимо так определить размер задачи, чтобы все эти факторы были учтены.

Для этого обратим внимание на то, что описание индивидуальной задачи, которое дается в терминах входа для ЭВМ, можно рассматривать как одну конечную цепочку (или слово) символов, выбранных из конечного входного алфавита. Несмотря на то что существуют различные пути описания данной индивидуальной задачи, предположим, что заранее выбран некоторый определенный способ и что с каждой массовой задачей связана некоторая фиксированная схема кодирования, которая отображает индивидуальные задачи в соответствующие цепочки символов. Входная длина индивидуальной задачи I из P определяется как число символов в цепочке, полу-

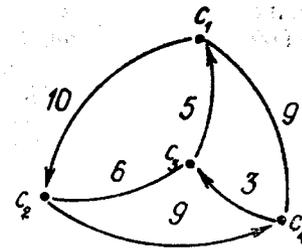


Рис. 3.1.1

ченной применением к задаче *I* схемы кодирования для массовой задачи *P*. Именно это число, т. е. входная длина, и используется в качестве формальной характеристики размера индивидуальной задачи.

Например, различные конкретные задачи о коммивояжере можно описать с помощью алфавита {*c*, 1,], /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, при этом предыдущий пример будет закодирован в виде такой цепочки символов: *c* [1] *c* [2] *c* [3] *c* [4] // 10/5/9//6/9//3. Более сложные индивидуальные задачи кодируются аналогичным образом. При такой кодирующей схеме для задачи о коммивояжере входная длина в нашем примере будет равна 32.

Временная сложность алгоритма отражает требующиеся для его работы затраты времени. Это функция, которая каждой входной длине *n* ставит в соответствие максимальное (по всем индивидуальным задачам длины *n*) время, затрачиваемое алгоритмом на решение индивидуальных задач этой длины. Естественно, что эта функция не будет полностью определена до тех пор, пока не зафиксирована схема кодирования, определяющая входную длину индивидуальной задачи, и не выбрано вычислительное устройство (или его модель), определяющее время работы. Однако подобные детали окажут незначительное влияние на различие между существующими классами задач. Поэтому в дальнейшем предполагаем, что схема кодирования для каждой задачи, а также вычислительное устройство или его модель фиксированы, и рассматриваем временную сложность алгоритмов в соответствии с получающимися входными длинами и соответствующими затратами времени.

Разные алгоритмы имеют различную временную сложность, и выяснение того, какие алгоритмы «достаточно эффективны», а какие «совершенно неэффективны», всегда будет зависеть от конкретной ситуации. Однако теоретики, занимающиеся разработкой и анализом алгоритмов, предлагают для сравнения эффективности алгоритмов один простой подход, позволяющий существенно прояснить ситуацию. Речь идет о различии между полиномиальными и экспоненциальными алгоритмами.

Будем говорить, что функция $f(n)$ есть $O(g(n))$, если существует константа *c* такая, что $\|f(n)\| < c \|g(n)\|$ для всех значений $n > 0$.

Полиномиальным алгоритмом (или алгоритмом полиномиальной временной сложности) называется алгоритм, у которого временная сложность равна $O(p(n))$, где $p(n)$ — некоторая полиномиальная функция, а *n* — входная длина. Алгоритмы, временная сложность которых не поддается подобной оценке, называются экспоненциальными.

Различие между двумя указанными типами алгоритмов становится особенно заметным при решении задач большого размера. Табл. 3.1.1 позволяет сравнить скорости роста нескольких типичных полиномиальных и экспоненциальных функций.

Таблица 3.1.1. Сравнение нескольких полиномиальных и экспоненциальных функций временной сложности

Функция временной сложности	Размер <i>n</i>					
	10	20	30	40	50	60
n	0,00001 с	0,00002 с	0,00003 с	0,00004 с	0,00005 с	0,00006 с
n^2	0,0001 с	0,0004 с	0,0009 с	0,0016 с	0,0025 с	0,0036 с
n^3	0,001 с	0,008 с	0,027 с	0,064 с	0,125 с	0,216 с
n^5	0,5 с	3,2 с	24,3 с	1,7 мин	5,2 мин	13,0 мин
2^n	0,001 с	1,0 с	17,9 мин	12,7 дней	35,7 лет	336 столетий
3^n	0,059 с	58 мин	6,5 лет	3855 столетий	$2 \cdot 10^8$ столетий	$1,3 \cdot 10^{13}$ столетий

Различие между полиномиальными и экспоненциальными алгоритмами проявляется еще более убедительно, если проанализировать влияние увеличения быстродействия ЭВМ на время работы алгоритмов. Табл. 3.1.2 показывает, на сколько увеличатся размеры задач, решаемых за один час машинного времени, если благодаря совершенствованию технологии быстродействие ЭВМ возрастает в 100 или 1000 раз по сравнению с современными машинами.

Таблица 3.1.2. Влияние технического совершенствования ЭВМ на полиномиальные и экспоненциальные алгоритмы. Размеры наибольшей задачи, решаемой за один час

Функция временной сложности	На современных ЭВМ	На ЭВМ, в 100 раз более быстрых	На ЭВМ, в 1000 раз более быстрых
n	N_1	$100N_1$	$1000N_1$
n^2	N_2	$10N_2$	$31,6N_2$
n^3	N_3	$4,64N_3$	$10N_3$
n^5	N_4	$2,6N_4$	$3,98N_4$
2^n	N_5	$+6,64 + N_5$	$N_5 + 9,97$
3^n	N_6	$+4,19 + N_6$	$N_6 + 6,29$

Заметим, что для функции $f(n) = 2^n$ увеличение скорости вычислений в 1000 раз приводит лишь к тому, что размер наибольшей задачи, разрешимой за один час, возрастает только на 10 единиц, в то время как для функции $f(n) = n^5$ этот размер возрастает почти в 4 раза. Эти таблицы демонстрируют некоторые причины, по которым полиномиальные алгоритмы обычно считаются более предпочтительными по сравнению с экспоненциальными. Эта точка зрения, различающая, с одной стороны, полиномиальные алгоритмы, а с другой — экспоненциальные, является открытым пунктом в нашем определении труднорешаемых задач.

Большинство экспоненциальных алгоритмов — это просто варианты полного перебора, в то время как полиномиальные алгоритмы можно построить лишь тогда, когда удается более глубоко проникнуть в суть решаемой задачи. Имеется широко распространенное соглашение, согласно которому задача не считается «хорошо решаемой» до тех пор, пока для нее не получен полиномиальный алгоритм. Поэтому задачу называют труднорешаемой, если для ее решения не существует полиномиального алгоритма.

Конечно, это формальное определение следует рассматривать только как одну из возможных трактовок понятия «труднорешаемая задача». Различие между эффективными (полиномиальными) алгоритмами и неэффективными (экспоненциальными) алгоритмами может принять совсем иной характер, когда размеры решаемых задач невелики. Даже на табл. 3.1.1. функция $f(n) = 2^n$ ведет себя лучше, чем $f(n) = n^5$ при $n < 20$. Можно легко построить еще более яркие примеры.

Известны некоторые экспоненциальные алгоритмы, хорошо зарекомендовавшие себя на практике. Дело в том, что временная сложность определена нами как мера поведения алгоритма в наихудшем случае, и тот факт, что какой-то алгоритм имеет временную сложность порядка 2^n , означает только, что решение по крайней мере одной индивидуальной задачи размера требует времени порядка 2^n . На самом деле может оказаться, что большинство индивидуальных задач требует для своего решения значительно меньших затрат времени, и такого рода ситуация действительно существует для нескольких хорошо известных алгоритмов. Так, симплекс-метод для решения задач линейного программирования имеет экспоненциальную временную сложность, но в то же время есть подтверждения того, что этот метод хорошо работает на практике. Другой пример: алгоритмы ветвей и границ столь успешно решают задачу о рюкзаке, что многие исследователи считают эту

задачу «хорошо решаемой», хотя алгоритмы ветвей и границ имеют экспоненциальную временную сложность.

К сожалению, подобные примеры очень редки. Хотя экспоненциальные алгоритмы известны для многих задач, лишь некоторые из них считаются приемлемыми для практических целей. Даже при наличии успешно работающих экспоненциальных алгоритмов, приведенных выше, исследователи не отказались от попыток найти для соответствующих задач полиномиальные алгоритмы. В действительности, сам факт успешного применения экспоненциальных алгоритмов давал основание предположить, что они каким-то образом выявляют некоторые существенные особенности решаемых задач и что более глубокое их исследование может привести к дальнейшему улучшению методов. В настоящее время пока не получены удовлетворительные объяснения, почему эти алгоритмы работают успешно, и не известны методы, позволяющие заранее прогнозировать хорошую работу того или иного экспоненциального алгоритма в практической ситуации.

Однако полиномиальные алгоритмы часто позволяют делать такого рода прогнозы, поскольку полиномиальные функции значительно более адекватно оценивают время работы алгоритмов. Хотя алгоритмы, имеющие временную сложность типа n^{100} или $n^2 \cdot 10^{99}$, не могут считаться эффективными с практической точки зрения, естественно возникающие полиномиальные задачи обычно требуют для своего решения (в самом худшем случае) времени порядка n^2 или n^3 , причем коэффициенты при старших членах полиномов не слишком велики. Алгоритмы, обладающие такими оценками, можно считать эффективными, и именно это весьма желательное свойство заставляет отдавать предпочтение полиномиальным алгоритмам как средству решения задач.

3.2. КЛАССЫ P И NP

Определение труднорешаемой задачи создает базу для теории, обладающей значительной общностью и большими возможностями. Понятие труднорешаемой задачи оказывается, по существу, независимым от конкретной схемы кодирования и модели ЭВМ, используемых при определении временной сложности.

Рассмотрим схемы кодирования. Например, решаемая задача определена на графе $G = (X, U)$, где X — множество вершин, U — множество ребер, и каждое ребро понимается как неупорядоченная пара вершин. Условия этой задачи могут быть описаны (табл. 3.2.1) либо просто списками всех вершин и ребер, либо с помощью матрицы инцидентий графа, либо со-

Таблица 3.2.1. Описание графа $G = (X, U)$, где $X = \{x_1, x_2, x_3, x_4\}$, $U = \{(x_1, x_2), (x_2, x_3)\}$ при трех схемах кодирования

Схема кодирования	Цепочка символов	Длина
Списки вершин и ребер	$x [1] x [2] x [3] x [4] (x [1] x [2]) (x [2] x [3])$	36
Список соседей	$(x [2]) (x [1] x [3]) (x [2]) ()$	24
Строки матрицы инцидентий	0100/1010/0010/0000	19

ставлением для каждой вершины списка всех вершин, имеющих с данными общее ребро («списка соседей»). Эти схемы кодирования для одного и того же графа могут дать входы разной длины. Однако легко проверить (табл. 3.2.2), что получаемые входы отличаются друг от друга не более чем полиномиальным образом, т. е. любой алгоритм, имеющий полиномиальную временную сложность при одной из этих схем кодирования, будет также обладать полиномиальной временной сложностью при всех остальных схемах.

Таблица 3.2.2. Общие оценки длины входа для трех схем кодирования графа $G = (X, U)$, представленных в табл. 3.2.1, где $|X| = v$, $|U| = u$

Схема кодирования	Нижняя оценка	Верхняя оценка
Списки вершин и ребер	$4v + 10u$	$4v + 10u + (v + 2u) \lceil \log_{10} v \rceil$
Список соседей	$2v + 8u$	$2v + 8u + 2u \lceil \log_{10} v \rceil$
Матрица инцидентий	$v + v - 1$	$v + v - 1$

Поскольку $u < v^2$, приведенные оценки показывают, что длины входа отличаются друг от друга не более, чем полиномиальным образом. В действительности стандартные схемы кодирования, используемые на практике для любой конкретной задачи, повидимому, всегда будут отличаться друг от друга не более чем полиномиальным образом. Было бы трудно представить себе разумную схему кодирования для какой-либо задачи, которая отличалась бы более чем полиномиальным образом от стандартных схем. Хотя мы не можем формально выразить такое понятие, как «разумная схема кодирования», следующие условия охватывают важные требования, связанные с этим понятием: код любой индивидуальной задачи I должен быть «сжатым», т. е. не содержать избыточной информа-

ции или символов; числа, входящие в условия задачи I , должны быть представлены в двоичной системе счисления (или десятичной, или восьмеричной, или иметь другое основание, отличное от 1).

Аналогичные замечания можно сделать относительно выбора модели ЭВМ. Все известные в настоящее время реалистичские модели ЭВМ (например, одноленточные машины и многоленточные машины Тьюринга, машины с произвольным доступом к памяти) эквивалентны относительно полиномиальной временной сложности. Можно ожидать, что и любая иная «разумная» модель ЭВМ будет эквивалентна по сложности всем перечисленным моделям.

Под словом «разумная модель», главным образом, имеется в виду то, что объем работы, выполняемый машиной в единицу времени, ограничен (сверху) полиномом. Так, модель, обладающая способностью выполнять параллельно произвольно много операций, не будет считаться «разумной», и в действительности ни одна из существующих (или проектируемых) ЭВМ не обладает подобным свойством. Во всяком случае, если ограничиться рассмотрением стандартных моделей реальных ЭВМ, то класс труднорешаемых задач не будет зависеть от выбора конкретной модели, и такой выбор можно осуществлять, исходя из интересов дела и не уменьшая при этом сферы применимости наших результатов.

Охарактеризуем теперь различия между двумя аспектами труднорешаемости, которые отражены в нашем определении. Первый аспект, который обычно имеется в виду, состоит в том, что искомое решение настолько велико, что не может быть представлено в виде выражения, длина которого была бы ограничена полиномом от длины входа.

Вторая ситуация возникает, например, если в задаче о коммивояжере в качестве дополнительного параметра фигурирует число B и требуется найти все маршруты длины, не превосходящие B . Легко построить индивидуальную задачу о коммивояжере, в которой имеется экспоненциальное число маршрутов длины, не превосходящих B , поэтому не существует алгоритма с полиномиальной временной сложностью, который все их перечисляет.

Труднорешаемостью этого вида ни в коем случае нельзя пренебрегать, и очень важно ее своевременно обнаружить. В большинстве случаев, однако, ее наличие ясно из постановки задачи. Этот аспект труднорешаемости обычно можно рассматривать как указание на то, что постановка задачи не реалистична, поскольку запрашивается больше информации, чем когда-либо можно использовать. В связи с этим с настоящего момента будем рассматривать только первый тип труднореша-

емости. А именно, изучать только такие задачи, длина решения которых ограничена полиномиальной функцией от длины входной информации.

Задачи распознавания свойств. Задача распознавания P состоит из двух множеств: множества D_P всех возможных индивидуальных задач и множества Y_P ($Y_P \subset D_P$) индивидуальных задач с ответом «да». Стандартная форма, которую будем применять для описания задач, состоит из двух частей. В первой части дается описание условия задачи в терминах различных компонент-множеств, графов, функций, чисел и т. д. Во второй части в терминах условия формулируется вопрос, предполагающий один из двух ответов — «да» или «нет». Это описание определяет D_P и Y_P очевидным образом. Индивидуальная задача принадлежит D_P в том случае, когда она может быть получена из стандартной формы описания подстановкой конкретных значений во все компоненты условия. Индивидуальная задача принадлежит Y_P в том и только том случае, когда ответом на вопрос задачи будет «да».

В качестве примера приведем переформулировку задачи о коммивояжере в терминах задачи распознавания.

КОММИВОЯЖЕР

УСЛОВИЕ. Заданы конечное множество $C = \{c_1, c_2, \dots, c_m\}$ городов, расстояние $d(c, c') \in Z^+$ для каждой пары городов $c_i, c_j \in C$ и граница $B \in Z^+$ (здесь Z^+ означает положительные целые числа).

ВОПРОС. Существует ли маршрут, проходящий через все города C , длина которого не превосходит B ? Другими словами, существует ли последовательность $\langle c_{(1)}, c_{(2)}, \dots, c_{(m)} \rangle$ элемен-

тов C такая, что $\sum_{i=1}^{m-1} d(c_{(i)}, c_{(i+1)}) + d(c_{(m)}, c_{(1)}) \leq B$?

Это иллюстрация важного приема — получения из оптимизационной задачи соответствующей задачи распознавания. Если в оптимизационной задаче среди всех структур данного типа ищется структура, имеющая минимальную «стоимость» (маршрут минимальной длины), то этой задаче можно сопоставить задачу распознавания, в которой в качестве дополнительного параметра фигурирует числовая граница B , а вопрос ставится о существовании структуры данного типа, стоимость которой не превосходит B (маршрут длины не более B).

Относительно подобного соответствия между задачами распознавания и задачами оптимизации важно отметить, что поскольку значение функции легко оценить, задача распознавания не может быть сложнее соответствующей задачи оптимизации.

Выбрав какую-нибудь схему кодирования для произвольной задачи распознавания и зная алгоритм ее решения, можно для фиксированной ЭВМ (зависящей от схемы кодирования) составить программу решения. Время, требуемое программой M для вычисления при входе задачи, есть число шагов, выполняемых до момента остановки. Временная сложность $T_m(n)$ программы M определяется как максимальное время, необходимое программе M для вычисления массовой задачи P с длиной входа n . Программа M называется полиномиальной программой, если существует такой полином p , что для всех $n \in Z^+$, $T_m(n) \leq p(n)$.

Будем говорить, что задача распознавания P принадлежит классу P при заданном кодировании, если существует полиномиальная программа, которая решает задачу P при заданном кодировании.

Очевидно, что формальным эквивалентом полиномиального алгоритма является полиномиальная программа. Таким образом, при доказательстве того, что определенные задачи могут быть решены полиномиальным алгоритмом, нет необходимости вдаваться в детали соответствующих программ. Следуя общепринятой практике, будем обсуждать алгоритмы в машинно-независимом стиле, как будто они работают непосредственно с компонентами индивидуальной задачи (множествами, графами, числами и т. д.), а не с их кодами. Здесь подразумевается, что при наличии желания и терпения для любого полиномиального алгоритма можно построить соответствующую полиномиальную программу.

Перейдем к определению второго класса задач распознавания свойств — классу NP . Для этого опять рассмотрим задачу КОММИВОЯЖЕР, описание которой дано выше. В условии даны: множества городов, расстояние между ними и граница B . При этом спрашивается, существует ли проходящий через все города маршрут длины, не превосходящий B . Полиномиальный алгоритм решения этой задачи не известен. Предположим, однако, что относительно некоторой индивидуальной задачи кто-то получил ответ «да». Если сомневаемся, то можно потребовать доказательства этого утверждения — предъявление маршрута, обладающего необходимыми свойствами. Имея предъявленное решение, нетрудно проверить, является ли оно на самом деле маршрутом, и если это так, то вычислить его длину, сравнить ее с границей B и тем самым проверить соответствующее утверждение. Более того, эту «процедуру проверки» можно представить в виде алгоритма, временная сложность которого ограничена полиномом от длины входа индивидуальной задачи.

Именно понятие полиномиальной «проверяемости» позволяет выделить задачи класса NP . Отметим, что проверяемость за полиномиальное время решения задачи не влечет разрешимости задачи за полиномиальное время. А именно, что за полиномиальное время можно проверить ответ «да» для задачи КОММИВОЯЖЕР, не учитывая время, которое может понадобиться на поиск нужного маршрута среди экспоненциального числа всех возможных маршрутов. Утверждаем, что по любому заданному маршруту для индивидуальной задачи I можно за полиномиальное время проверить, доказывает ли этот маршрут, что ответ на вопрос относительно индивидуальной задачи I есть «да».

Неформально класс NP можно определить с помощью понятия, которое будем называть *недетерминированным алгоритмом*.

Такой алгоритм состоит из двух различных стадий — стадии угадывания и стадии проверки. По заданной индивидуальной задаче I на первой стадии происходит просто «угадывание» некоторой структуры S . Затем I и S вместе подаются в качестве входа на стадию проверки, которая выполняется обычным детерминированным образом и либо заканчивается ответом «да», либо заканчивается ответом «нет», либо продолжается без остановки (последние две возможности различать не обязательно). Недетерминированный алгоритм «решает» задачу распознавания P , если для любой индивидуальной задачи $I \in D_P$ выполнены следующие два свойства: 1) если $I \in Y_P$, то существует такая структура S , угадывание которой для входа I приведет к тому, что стадия проверки, начиная работу на входе (I, S) , закончится ответом «да»; 2) если $I \notin Y_P$, то не существует такой структуры S , угадывание которой для входа I обеспечило бы окончание стадии проверки на ходе (I, S) ответом «да».

Например, недетерминированный алгоритм решения задачи КОММИВОЯЖЕР можно было бы построить, используя в качестве стадии угадывания просто выбор произвольной последовательности городов, а в качестве стадии проверки — упомянутую выше полиномиальную процедуру «проверки доказательства» для задачи КОММИВОЯЖЕР. Очевидно, для любой индивидуальной задачи I найдется такая догадка S , что результатом работы стадии проверки на входе (I, S) будет «да» в том и только том случае, если для индивидуальной задачи I существует маршрут искомой длины.

Говорят, что недетерминированный алгоритм, решающий задачу распознавания P , работает в течение «полиномиального времени», если найдется полином p такой, что для любого $I \in Y_P$ найдется некоторая догадка S , приводящая на стадии

детерминированной проверки на входе (I, S) к ответу «да» за время $p(n)$, где n — длина входа задачи I .

Класс NP , определяемый неформально, — это класс всех задач распознавания P , которые при разумном кодировании могут быть решены недетерминированными алгоритмами за полиномиальное время. Вопрос о взаимоотношении классов NP и P имеет фундаментальное значение. Одно соответствие, которое неявно присутствовало во всех прежних рассуждениях, заключается в том, что $P \subseteq NP$. Всякая задача распознавания, разрешимая за полиномиальное время детерминированным алгоритмом, разрешима также за полиномиальное время недетерминированным алгоритмом. Если $P \in P$ и A — произвольный детерминированный полиномиальный алгоритм решения P , то полиномиальный недетерминированный алгоритм для P можно получить, воспользовавшись A в качестве стадии проверки и игнорируя стадию угадывания. Таким образом, из $P \in P$ следует, что $P \in NP$.

3.3. ПОЛИНОМИАЛЬНАЯ СВОДИМОСТЬ И NP -ПОЛНЫЕ ЗАДАЧИ

Из приводившихся ранее рассуждений можно понять, что есть много причин считать, что P не совпадает с NP . Полиномиальные недетерминированные алгоритмы определенно оказываются более мощными, чем полиномиальные детерминированные алгоритмы, и не известны общие методы их превращения в детерминированные полиномиальные алгоритмы. В действительности самый сильный из известных в настоящее время результатов состоит в следующем.

Теорема 3.3.1. *Если $P \in NP$, то существует такой полином P , что P может быть решена детерминированным алгоритмом с временной сложностью порядка $2^{p(n)}$.*

Способность недетерминированного алгоритма проверить за полиномиальное время экспоненциальное число возможностей может навести на мысль, что полиномиальные недетерминированные алгоритмы являются более мощным средством, чем полиномиальные детерминированные алгоритмы. В самом деле, для многих задач класса NP , таких как КОММИВОЯЖЕР, ИЗОМОРФИЗМ ПОДГРАФУ, и большого числа других задач не найдено полиномиального детерминированного алгоритма. Поэтому не удивляет широко распространенное мнение, что $P = NP$, хотя пока доказательство этой гипотезы отсутствует. При существующем уровне знаний более разумно работать, соглашаясь, что $P \neq NP$, чем пытаться доказать противное.

Если P не совпадает с NP , то различие между P и NP/P очень существенно. Все задачи из P могут быть решены полиномиальными алгоритмами, а все задачи из NP/P труднорешаемы.

До тех пор, пока не доказано, что $P \neq NP$, нет никакой надежды показать, что некоторая конкретная задача принадлежит классу NP/P . По этой причине цель создаваемой теории заключается в доказательстве более слабых результатов вида «если $P \neq NP$, то $P \in NP/P$ ». Хотя доказательство таких условий результатов может показаться столь же трудным, как и безусловных, однако, как увидим в дальнейшем, имеются несложные методы доказательства. Основная идея подобного условного перехода основана на понятии полиномиальной сводимости.

Будем говорить, что имеет место полиномиальная сводимость схемы кодирования L_1 к схеме кодирования L_2 , если существует функция $f: L_1 \rightarrow L_2$, удовлетворяющая двум условиям:

1. Существует детерминированная программа, вычисляющая f с временной сложностью ограниченной полиномом.

2. Для любой индивидуальной задачи $I \in L_1$ тогда и только тогда, когда $f(I) \in L_2$.

Если L_1 полиномиально сводится к L_2 , то запишем $L_1 \propto L_2$. Если P_1 и P_2 — задачи распознавания, а L_1 и L_2 — их схемы кодирования, то пишем $P_1 \propto P_2$, если $L_1 \propto L_2$.

Пример. Пусть $G(X, U)$ — граф с множеством вершин X и множеством ребер U . Простым циклом в G называется такая последовательность (x_1, x_2, \dots, x_k) различных вершин из X , что $(x_1, x_{i+1}) \in U$, $1 < i < k$ и $(x_k, x_1) \in U$.

Гамильтоновым циклом в G называется простой цикл, содержащий все вершины графа G . Задача ГАМИЛЬТОНОВ ЦИКЛ определяется следующим образом.

ГАМИЛЬТОНОВ ЦИКЛ

УСЛОВИЕ. Задан граф $G = (X, U)$.

ВОПРОС. Верно ли, что G содержит гамильтонов цикл?

Покажем, что задача распознавания ГАМИЛЬТОНОВ ЦИКЛ (ГЦ) сводится к задаче КОММИВОЯЖЕР (КМ). Для этого требуется указать функцию f , которая отображает каждую индивидуальную задачу из ГЦ в соответствующую индивидуальную задачу из КМ и удовлетворяет двум условиям, которые требуются от полиномиальной сводимости.

Функция f определяется очень просто. Пусть $G(X, U)$, $|X| = m$ означает фиксированную индивидуальную задачу из ГЦ. Соответствующая задача из КМ строится так: множество городов S совпадает с X ; для любых двух городов $X_i, X_j \in S$ расстояние $d(x_i, x_j)$ между ними полагаем равным 1, если

$(x_i, x_j) \in U$, и 2 в противном случае. Граница B для длины искомого маршрута берется равной m .

Легко показать, что эта функция f осуществляет сводимость и может быть вычислена за полиномиальное время. Для вычисления $m(m-1)/2$ расстояний $d(x_i, x_j)$ необходимо лишь выяснить, принадлежит ли (x_i, x_j) множеству U или нет. Поэтому первое условие полиномиальной сводимости выполнено. Для проверки второго требования необходимо показать, что G содержит гамильтонов цикл тогда и только тогда, когда в $f(G)$ имеется проходящий через все города маршрут длины, не превосходящий B . Вначале допустим, что $\langle x_1, x_2, \dots, x_m \rangle$ — гамильтонов цикл в G . Тогда $\langle x_1, x_2, \dots, x_m \rangle$ — маршрут $f(G)$, а его длина равна $m(m=B)$, так как расстояние между любыми соседними городами маршрута равно 1, поскольку оно соответствует ребру в G . Наоборот, предположим, что найден маршрут $f(G)$, длина которого не превосходит B . Поскольку расстояние между любыми двумя городами в $f(G)$ равно 1, либо 2 и при вычислении длины маршрута суммируется ровно m таких расстояний, то из равенства $B = m$ следует, что расстояние между каждой парой соседних городов в маршруте равно 1. По определению $f(G)$ следует, что (x_i, x_{i+1}) , $1 \leq i \leq m$, и (x_m, x_1) являются ребрами графа G , следовательно, $\langle x_1, x_2, \dots, x_m \rangle$ — гамильтонов цикл в G . Тем самым доказали, что ГЦ \propto КМ. Для дальнейших рассуждений нам понадобится несколько лемм, которые приводятся без доказательств.

Лемма 3.3.1. Если $P_1 \propto P_2$, то из $P_2 \in P$ следует, что $P_1 \in P$. Эквивалентное утверждение: из $P_1 \in P$ следует, что $P_2 \notin P$.

На примере сводимости ГЦ \propto КМ и на основании леммы 3.3.1 пришли к выводу: если задача КОММИВОЯЖЕР может быть решена полиномиальным алгоритмом, то этот факт верен и для задачи ГАМИЛЬТОНОВ ЦИКЛ, а если ГЦ — труднорешаемая задача, то и КМ — также труднорешаемая задача. Таким образом, лемма 3.3.1 позволяет интерпретировать сводимость $P_1 \propto P_2$ как утверждение, что задача P_2 «не проще» задачи P_1 .

Отношение полиномиальной сводимости особенно удобно, поскольку оно является транзитивным, что устанавливает следующая лемма.

Лемма 3.3.2. Если $P_1 \propto P_2$ и $P_2 \propto P_3$, то $P_1 \propto P_3$.

Теперь можно сказать, что задачи распознавания P_1 и P_2 полиномиально эквивалентны, если они сводятся друг к другу, т. е. $P_1 \propto P_2$ и $P_2 \propto P_1$. Лемма 3.3.2 определяет частичное упорядочение возникающих классов эквивалентности задач распознавания. В действительности класс P — это «наименьший» относительно этого частичного порядка класс эквивалент-

ности и с вычислительной точки зрения его можно рассматривать как класс самых легких задач распознавания. Класс NP -полных задач дает нам другой класс эквивалентности, который характеризуется тем, что он содержит «самые трудные» задачи распознавания из NP .

Задача распознавания P называется NP -полной, если $P \in NP$, и любая другая задача распознавания $P' \in NP$ полиномиально сводится к P . Тем самым лемма 3.3.1 позволяет отождествлять NP -полные задачи с самыми трудными задачами из NP . Если хотя бы одна NP -полная задача может быть решена за полиномиальное время, то и все задачи из NP также могут быть решены за полиномиальное время. Если хотя бы одна задача из NP труднорешаемая, то и все NP -полные задачи труднорешаемы. На основании лемм 3.3.1 и 3.3.2 можно сделать вывод, что если задачи распознавания P_1 и P_2 принадлежат классу NP , P_1 — NP -полная задача, то P_2 также NP -полная задача. На уровне задач распознавания это позволяет указать простой путь доказательства NP -полноты новой задачи P , если известна хотя бы одна NP -полная задача. Для этого достаточно показать, что $P \in NP$; какая-то одна известная NP -полная задача P' сводится к P . Остается только найти некоторую исходную NP -полную задачу, которая является в некотором смысле «первой» в классе NP .

Пусть $U = \{u_1, u_2, \dots, u_m\}$ — множество булевских переменных. Под набором значений истинности на множестве U будем понимать функцию $t: u \rightarrow \{T, F\}$. Если $t(u) = T$, то говорим, что u принимает значение «истина» относительно t ; если $t(u) = F$, то u принимает значение «ложь». Если u — переменная из U , то \bar{u} и u назовем литералами из U . Литерал \bar{u} принимает значение «истина» относительно t в том и только в том случае, если переменная u принимает значение «ложь».

Дизъюнкцией над U назовем множество литералов над U , например, $\{u_1, u_4, u_8\}$. Она представляет дизъюнкцию этих литералов и называется выполненной при некотором наборе значений истинности тогда и только тогда, когда при рассматриваемом наборе значений истинности хотя бы один из ее членов принимает значение «истина». В нашем примере дизъюнкция будет выполнена относительно t , если одновременно не окажется, что $t(u_1) = F$, $t(u_4) = T$, $t(u_8) = F$. Набор S дизъюнкций над U называется выполнимым в том и только том случае, если найдется некоторый набор значений истинности на множестве U , такой, что одновременно выполнены все

дизъюнкции из S . Такой набор значений истинности называется выполняющим набором значений истинности для S . Задача ВЫПОЛНИМОСТЬ формулируется следующим образом:
ВЫПОЛНИМОСТЬ

УСЛОВИЕ. Заданы множество переменных U и набор S дизъюнкций над U .

ВОПРОС. Существует ли выполняющий набор значений истинности для S ?

Честь быть «первой» NP -полной задачей выпала на долю задачи распознавания ВЫПОЛНИМОСТЬ (ВЫП), что устанавливает фундаментальная теорема Кука.

Теорема 3.3.2. (теорема Кука). *Задача ВЫПОЛНИМОСТЬ есть NP -полная задача.*

3.4. ДОКАЗАТЕЛЬСТВО РЕЗУЛЬТАТОВ ОБ NP -ПОЛНОТЕ

Доказательство теоремы Кука о NP -полноте задачи ВЫПОЛНИМОСТЬ неэлементарно, с ним можно подробно ознакомиться в работах [1, 3]. Если бы все доказательства NP -полноты для других задач были так же сложны, то очень сомнительно, чтобы список NP -полных задач смог стать таким обширным, как в настоящее время. Однако, как указывалось раньше, если известна хотя бы одна NP -полная задача, то процедура доказательства NP -полноты других задач значительно упрощается. Для доказательства NP -полноты задачи $P \in NP$ достаточно показать, что какая-нибудь из известных NP -полных задач P' может быть сведена к P . Таким образом, в дальнейшем процесс доказательства NP -полноты задачи распознавания P будет состоять из следующих четырех шагов:

1) доказательства того, что P лежит в NP ; 2) выбора известной NP -задачи P' ; 3) построения функции f , сводящей задачу P' к задаче P ; 4) доказательства того, что функция f осуществляет полиномиальное сведение.

Когда опытным практикам встречается задача P , NP -полноту которой требуется доказать, их преимущество заключается в том, что они имеют богатый выбор путей такого доказательства. Вполне может случиться, что в прошлом они уже доказывали или видели доказательство NP -полноты похожей задачи P' . Поэтому возникнут мысли попытаться построить доказательство NP -полноты задачи P , копируя доказательство NP -полноты задачи P' , или просто сводя задачу P' к задаче P . Во многих случаях это может привести к достаточно простому доказательству NP -полноты задачи P .

Очень часто не удается найти NP -полную задачу, похожую на P . В таких случаях практику может быть не ясно, какая из

известных NP -полных задач лучше всего подходит в качестве основы искомого доказательства. Однако, предыдущий опыт может оказаться полезным для ограничения области выбора некоторым ядром из базовых задач, использовавшихся ранее. Хотя теоретически любую из известных NP -полных задач можно наравне с другими выбрать для доказательства NP -полноты новой задачи, на практике оказывается, что некоторые задачи подходят для этой цели гораздо лучше других. Следующие шесть задач входят в число тех, которые используются наиболее часто и для начинающего они могут служить основным ядром списка известных NP -полных задач.

1. 3-ВЫПОЛНИМОСТЬ (3-ВЫП)

УСЛОВИЕ. Дан набор $C = \{c_1, c_2, \dots, c_m\}$ дизъюнкций на конечном множестве переменных U , таких, что $|c_i| = 3, 1, \leq i \leq m$.

ВОПРОС. Существует ли на U набор значений истинности, при котором выполняются все дизъюнкции из C ?

2. ТРЕХМЕРНОЕ СОЧЕТАНИЕ (3 — С)

УСЛОВИЕ. Дано множество $M \subset W \times X \times Y$, W, X и Y — непересекающиеся множества, содержащие одинаковое число элементов q .

ВОПРОС. Верно ли, что M содержит трехмерное сочетание, т. е. подмножество $M' \subseteq M$ такое, что $|M'| = q$ и никакие два разных элемента M' не имеют ни одной равной координаты?

3. ВЕРШИННОЕ ПОКРЫТИЕ (ВП)

УСЛОВИЕ. Дан граф $G = (X, U)$ и положительное целое число $k, k < |X|$.

ВОПРОС. Имеется ли в графе G вершинное покрытие не более, чем из k элементов, т. е. такое подмножество $X' \subseteq X$, что $|X'| \leq k$ и для каждого ребра $(u, v) \in U$ хотя бы одна из вершин u или v принадлежит X' ?

4. КЛИКА

УСЛОВИЕ. Дан граф $G = (X, U)$ и положительное число $J \leq |X|$.

ВОПРОС. Верно, что G содержит некоторую клику мощности не менее J , т. е. такое подмножество $X' \subseteq X$, что $|X'| \geq J$ и любые две вершины из X' соединены ребром из U ?

5. ГАМИЛЬТОНОВ ЦИКЛ

УСЛОВИЕ. Дан граф $G = (X, U)$.

ВОПРОС. Верно ли, что G содержит гамильтонов цикл, т. е. такую последовательность $\langle x_1, x_2, \dots, x_n \rangle$ вершин графа G , что $n = |X|$, $(x_n, x_1) \in U$ и для всех $i, 1 \leq i \leq n$?

6. РАЗБИЕНИЕ

УСЛОВИЕ. Заданы конечное множество A и вес $s(a) \in \mathbb{Z}^+$ для каждого $a \in A$.

ВОПРОС. Существует ли подмножество $A' \subseteq A$ такое, что $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$?

Одна из причин популярности этих шести задач заключается в том, что все они содержались в исходном списке из 21 NP -полной задачи, приведенном в работе [3]. Первая сводимость будет продемонстрирована с помощью задачи ВЫПОЛНИМОСТЬ, поскольку пока это единственная задача, NP -полнота которой считается установленной с помощью теоремы Кука. На рис. 3.4.1 указана последовательность сведения задач, которая может быть применена в доказательстве NP -полноты шести основных задач, причем если стрелка ведет от одной задачи к другой, то первая сводится ко второй.

Проиллюстрируем некоторые приемы сведения, последовательно выполняя п. 1—4.

Теорема 3.4.1. Задача 3-ВЫПОЛНИМОСТЬ является NP -полной.

Доказательство рассмотрим на примере. Пусть $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$ — множество переменных и $C = \{c_1, c_2, c_3, c_4\}$ набор дизъюнкций, а именно $c_1 = \{u_1\}$, $c_2 = \{u_1, u_2\}$, $c_3 = \{u_1, u_2, u_3\}$, $c_4 = \{u_2, u_3, u_4, u_5, u_6, u_7\}$. Эта индивидуальная задача ВЫП, которую нужно свести в 3-ВЫП.

1. Нетрудно видеть, что 3-ВЫП $\subseteq NP$. Это следует из того, что если недетерминированный алгоритм угадает набор истинности переменных задачи, то для k наборов дизъюнкций в задаче 3-ВЫП необходимо провести $3k$ подстановок и сделать k проверок выполнения каждой дизъюнкции.

2. В качестве P' выбирается NP -полная задача ВЫП.

3. Построим набор C' трехлитеральных дизъюнкций на некотором множестве переменных U' такой, что C' выполним тогда и только тогда, когда выполним C .

Набор C' будет строиться путем замены каждой отдельной дизъюнкции $c_j \in C$ эквивалентным в некотором роде набором C'_j трехлитеральных дизъюнкций на множестве U исходных переменных и множестве некоторых дополнительных переменных, причем последние переменные будут использоваться только в дизъюнкциях из C'_j .

Пусть C_j задается множеством $\{z_1, z_2, \dots, z_k\}$, где z_i — литералы на множестве U .

С л у ч а й 1. $k = 1$. Сюда относится c_1 . Тогда добавляются две переменные y_1, y_2 :

$$C'_1 = \{u_1, y_1, y_2\}, \{u_1, y_1, \bar{y}_2\}, \{u_1, \bar{y}_1, y_2\}, \{u_1, \bar{y}_1, \bar{y}_2\}.$$

Очевидно, что если C_1 выполнима, то независимо от значений $t(y_1)$ или $t(y_2)$ выполним и весь набор C'_1 . Наоборот,

если набор C_1 выполним, то найдется такая дизъюнкция, где литералы от y_1 и y_2 принимают значение «ложь» и для выполнимости этой дизъюнкции необходима выполнимость u_1 .

С л у ч а й 2. $k = 2$. Сюда относится C_2 . Тогда добавляется одна переменная y_3 .

$$C_2' = \{u_1, u_2, y_3\}, \{\bar{u}_1, u_2, \bar{y}_3\}.$$

Очевидно, что при выполнимости C_2 выполним и набор C_2' . Наоборот, при выполнимости C_2' возьмем ту дизъюнкцию, где литерал от y_3 принимает значение «ложь». Тогда для выполнимости C_2' необходима выполнимость C_2 .

С л у ч а й 3. $k = 3$. В этом случае никаких изменений не надо.

С л у ч а й 4. $k > 3$. Сюда относится C_4 и тогда добавляются переменные y_4, y_5, y_6 .

$$C_4' = \{u_2, u_3, y_4\}, \{\bar{y}_4, \bar{u}_4, y_5\}, \{\bar{y}_5, \bar{u}_5, y_6\}, \{\bar{y}_6, u_6, u_7\}.$$

В данном случае конъюнкция C_4 разбивается на $(k - 2)$ компоненты (в нашем случае на 4). $A: (u_2, u_3), (\bar{u}_4), (\bar{u}_5), (u_6, u_7)$. Соответственно из новых переменных также образуются компоненты.

$$B: (y_4), (\bar{y}_4, y_5), (\bar{y}_5, y_6), (\bar{y}_6).$$

Очевидно, при любых значениях $t(y_i)$ среди компонент B может быть невыполнимой ровно одна компонента. Для выполнения C_4 достаточно выполнения одной из компонент A . Таким образом, если C_4 выполнима, то выполним и набор C_4' . Для этого необходимо дать переменным y_i такие значения, чтобы номер компоненты невыполнимости из B совпадал с номером компоненты выполнимости из A . Наоборот, пусть C_4' — выполнимый набор для каких-то значений переменных $\{u_2, u_3, u_4, u_5, u_6, u_7, y_4, y_5, y_6\}$. Выделим компоненту невыполнимости из B . Так как C_4' выполним, то соответствующая компонента из A будет выполняемой, что влечет за собой выполнимость C_4 . Этим и завершается доказательство сводимости задачи ВВП к задаче 3-ВВП.

4. Для того чтобы убедиться, что эта сводимость полиномиальна, достаточно заметить, что число трехлитеральных дизъюнкций в C' , ограничено полиномом от mn , где m — число дизъюнкций в C , а n — число переменных в U .

Действительно, пусть в C существуют p дизъюнкций длины 1, q дизъюнкций длины 2. Остальные $m - p - q$ дизъюнкции имеют длину r_i , $1 < r_i < m - p - q$. Тогда в результате добавления переменных получим трехлитеральные дизъюнкции в

количестве

$$4p + 2q + \sum_{i=1}^{m-p-q} (r_i - 2) = 6p + 4q - 2m + \\ + \sum_{i=1}^{m-p-q} r_i < 6p + 4q - 2m + n(m - p - q) \leq mn.$$

Следовательно, размер индивидуальной задачи 3-ВВП ограничен сверху некоторым полиномом от размера соответствующей задачи ВВП, а так как все детали построения сводимости очевидны, то нетрудно проверить, что эта сводимость является полиномиальной.

Ограниченная структура задачи 3-ВВП делает ее гораздо более полезной при доказательстве результатов об NP -полноте по сравнению с задачей ВВП. Любое доказательство, основанное на задаче ВВП (кроме только что приведенного), может быть легко перестроено в доказательство, основанное на 3-ВВП, даже без изменения функции, осуществляющей сводимость. Действительно, дополнительное условие о том, что все дизъюнкции имеют одинаковую длину, часто может упростить сводимость, которую нужно построить, и тем самым облегчить ее отыскание. Более того, очень малая длина дизъюнкций позволяет использовать сводимости, которые вообще не могли бы быть построены для индивидуальных задач с дизъюнкциями большей длины.

Рассмотрим еще один способ сводимости одной задачи к другой из схемы на рис. 3.4.1.

Теорема 3.4.2. Задача ВЕРШИННОЕ ПОКРЫТИЕ NP -полна.

1. Для доказательства $VP \in NP$ заметим, что если недетерминированный алгоритм укажет решение, т. е. угадает подмножество вершин, то можно за полиномиальное время проверить, что это подмножество содержит хотя бы один конец любого ребра и имеет указанное число элементов.

2. В качестве известной NP -полной задачи выбираем задачу 3-ВВП.

3. Сведем задачу 3-ВВП к задаче ВП. Пусть $U = \{u_1, u_2, \dots, u_n\}$ и $C = \{c_1, c_2, \dots, c_m\}$ определяют произвольную индивидуальную задачу из 3-ВВП.

Укажем граф $G = (X, U)$ и положительное число $k \leq |X|$ такие, что G имеет вершинное покрытие с числом элементов не более k тогда и только тогда, когда выполним набор дизъюнкций C . Проиллюстрируем наши рассуждения. Пусть $Y = \{y_1, y_2, y_3, y_4\}$ и $C = \{c_1, c_2\}$, где $c_1 = \{y_1, \bar{y}_3, \bar{y}_4\}$, $c_2 = \{\bar{y}_1, y_2, y_4\}$. Для каждой переменной $y_i \in Y$ имеется компонента $T_i = (y_i, \bar{y}_i)$ набора значений истинности, т. е. пара вершин, соединен-

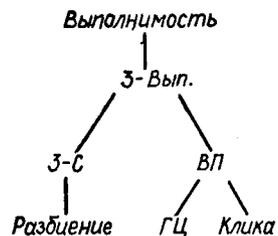


Рис. 3.4.1

ных одним ребром. Таким образом, получаем множество вершин X_i и ребер U_i :

$X_i = \{y_i, \bar{y}_i\}$; $U_i = (y_i, \bar{y}_i)$. Для каждой дизъюнкции $c_j \in C$ имеется компонента, состоящая из треугольника, т. е. из трех вершин и связывающих их ребер:

$$X'_j = \{a_1[j], a_2[j], a_3[j]\};$$

$$U'_j = \{(a_1[j], a_2[j]), (a_1[j], a_3[j]), (a_2[j], a_3[j])\}.$$

Для каждой дизъюнкции $c_j \in C$ обозначим через p_j, q_j, r_j три входящих в нее литерала. Тогда связывающие литералы и треугольники ребра задаются следующим образом:

$$U'_j = \{(a_1[j], p_j), (a_2[j], q_j), (a_3[j], r_j)\}.$$

Построение индивидуальной задачи из ВП будет закончено, если положить $k = n + 2m$ и $G = (x, u)$, где

$$X = \left(\bigcup_{i=1}^n X_i \right) \cup \left(\bigcup_{j=1}^m X'_j \right);$$

$$U = \left(\bigcup_{i=1}^n U_i \right) \cup \left(\bigcup_{j=1}^m U'_j \right) \cup \left(\bigcup_{j=1}^m U''_j \right).$$

На рис. 3.4.2 приведен граф, соответствующий индивидуальной задаче из 3-ВЫП, которая задана выше.

Предположим, что $X' \subseteq X$ — вершинное покрытие G и $|X'| \leq k$. В силу сделанных выше замечаний X' должно содержать по крайней мере одну вершину каждого T_i и хотя бы две вершины каждого треугольника. В совокупности это дает

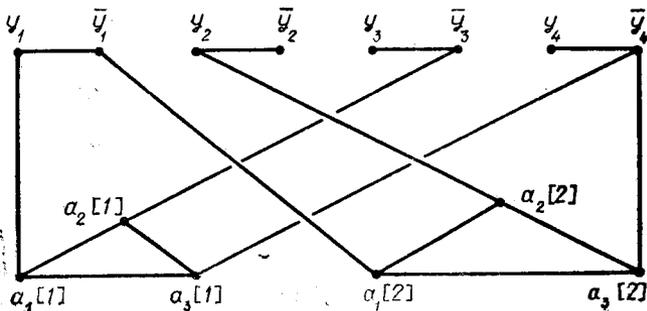


Рис. 3.4.2

по крайней мере $n + 2m = k$ вершин, поэтому X' должно содержать ровно одну вершину каждого T_i и ровно две вершины каждого треугольника.

Таким образом, можно использовать пересечение X' с компонентами наборов значений истинности, чтобы получить отображение $t: y \rightarrow \{T, F\}$. Для этого положим $t(y)_i = T$, если $y_i \in X'$, и $t(y)_i = F$, если $\bar{y}_i \in X'$. Для доказательства того, что набор значений истинности выполняет каждую дизъюнкцию $c_j \in C$, рассмотрим три ребра подграфа U'_j . Только два из них могут быть инцидентны вершинам из $X'_j \cap X'$, поэтому одно из них должно быть инцидентно вершине некоторого подграфа X_i , принадлежащей также и X' . Отсюда следует, что соответствующий литерал y_i или \bar{y}_i дизъюнкции C_j принимает значение «истина» при рассматриваемом наборе значений истинности t , и следовательно, t выполняет дизъюнкцию C_j . Поскольку это утверждение имеет место для каждой дизъюнкции $c_j \in C$, то отсюда следует, что t — выполняющий набор значений истинности для C .

Обратно предположим, что $t: y \rightarrow \{T, F\}$ — выполняющий набор значений истинности для C . Тогда соответствующее вершинное покрытие X' содержит одну вершину из каждого T_i и две вершины из каждого треугольника. При этом вершиной из T_i , принадлежащей X' , будет y_i (если $t(y)_i = T$) или \bar{y}_i (если $t(y)_i = F$). Поскольку t выполняет каждую дизъюнкцию C_j , то это обеспечивает покрытие по крайней мере одного из трех ребер, принадлежащих множеству U'_j . Следовательно, осталось включить в X' концы оставшихся двух ребер из U'_j , принадлежащие соответствующему треугольнику (которые могут быть либо инцидентны, либо не инцидентны вершинам из компонент наборов значений истинности); при этом получится искомого вершинное покрытие.

4. Нетрудно видеть, что эту конструкцию можно осуществить за полиномиальное время, так же как в предыдущем случае сведения задачи ВЫП к задаче 3-ВЫП.

3.5. ПРИМЕНЕНИЕ ТЕОРИИ NP-ПОЛНОТЫ ДЛЯ АНАЛИЗА ЗАДАЧ

Овладев основными элементами теории NP-полноты, можно использовать эту теорию для анализа задач.

Встретившись с новой задачей, естественно задать вопрос: «Можно ли рассматриваемую задачу решить полиномиальным алгоритмом?» Если ответ на этот вопрос оказывается положительным, то с точки зрения NP-полноты ничего большего о задаче сказать нельзя. Дальнейшие усилия нужно сконцентри-

ровать на поиске как можно более эффективных полиномиальных алгоритмов. Однако, если для решения задачи не известно полиномиального алгоритма, возникает второй вопрос: «Верно ли, что рассматриваемая задача NP -полна?»

Для того чтобы поставленный вопрос имел смысл, предположим, что рассматриваемая задача сформулирована как задача распознавания и что она принадлежит классу NP . В некоторых случаях легко устанавливается полиномиальная разрешимость интересующей нас задачи, в других же оказывается очевидной ее NP -полнота. Если задача оказалась NP -полной, то это является сильным аргументом для доказательства того, что ее нельзя решить за полиномиальное время.

В большинстве случаев найти ответ на каждый из поставленных вопросов довольно трудно. Обычно неизвестно, разрешима задача за полиномиальное время или она NP -полна, а требуется некоторая работа для выяснения, какая из этих двух возможностей в действительности реализуется (в том случае, если $NP \neq P$, то в классе NP существуют задачи, которые неразрешимы за полиномиальное время и не NP -полны). Как в этом случае выяснить сложность рассматриваемой задачи?

Если интуитивно склоняемся в пользу одной из возможностей, то очень соблазнительно сконцентрировать усилия в этом направлении. Однако в подобных вопросах интуиция оказывается особенно обманчивой, ибо зачастую задачи, разрешимые за полиномиальное время, очень незначительно отличаются от NP -полных задач. Для этого сравним две задачи из теории графов.

КРАТЧАЙШИЙ ПУТЬ МЕЖДУ ДВУМЯ ВЕРШИНАМИ.

УСЛОВИЕ. Заданы: граф $G = (X, U)$, длины $l(u) \in \mathbb{Z}^+$ для всех $u \in U$, выделены вершины $a, b \in X$ и положительное число B .

ВОПРОС. Существует ли в G простой путь из a в b , имеющий длину не более B ?

Вторая задача по постановке почти не отличается от первой.

САМЫЙ ДЛИННЫЙ ПУТЬ МЕЖДУ ДВУМЯ ВЕРШИНАМИ.

УСЛОВИЕ. Заданы: граф $G = (X, U)$, длины $l(u) \in \mathbb{Z}^+$ для всех $u \in U$, выделенные вершины $a, b \in X$ и положительное целое число B .

ВОПРОС. Существует ли в G простой путь из a в b , имеющий длину не менее B ?

Отличаются эти задачи только одним словом: слово «более» заменено на слово «менее». Однако первая задача принадлежит к классу P , а другая NP -полна. Наша интуиция, как правило, основана на опыте работы с близкими задачами.

Таким образом, при анализе лучше пользоваться двусторонним подходом. Пытаясь с одной стороны доказать NP -полноту задачи, одновременно целесообразно искать полиномиальный алгоритм ее решения. Неудачи в доказательстве NP -полноты задачи могут породить идею алгоритма ее решения, а неудачи в построении алгоритма могут указать путь к доказательству NP -полноты задачи.

Предположим, что интересующая нас задача NP -полна. При этом получаем полные ответы на оба вопроса, с которых начинался анализ.

Однако это лишь первый шаг на пути решения задачи. Задача, которой мы занимаемся, часто получается как идеализация менее привлекательной прикладной задачи, и некоторые детали, опущенные в процессе идеализации, могут изменить задачу в такой степени, что она станет разрешимой за полиномиальное время. Даже если это не так, у задачи могут иметься некоторые важные частные подзадачи, которые могут быть решены за полиномиальное время. Возможно также, что индивидуальные задачи, плохо поддающиеся решению, встречаются относительно редко и имеют легко выявляемые особенности, позволяющие заблаговременно опознать такие задачи. Эти возможности могут быть изучены с помощью анализа подзадач интересующей нас задачи.

Согласно принятому описанию массовых задач распознавания каждая из них состоит из двух частей: множества D всех индивидуальных задач и множества Y индивидуальных задач из D с ответом «да». Подзадачей (или «частным случаем») задачи $P = (D, Y)$ называется такая задача $P' = (D', Y')$, что $D' \subseteq D$ и $Y' = Y \cap D'$. Другими словами, P' есть подзадача задачи P , если в ней сформулирован тот же вопрос, что и в P , но только на некотором подмножестве множества индивидуальных задач из P .

Таким образом, подзадачи возникают всякий раз, когда на множество индивидуальных задач налагаются дополнительные ограничения. В задачах из теории графов можно ограничиться индивидуальными задачами, в которых графы планарны, или двудольны, или ацикличны, или обладают некоторыми из этих свойств одновременно. В задачах, содержащих множества, можно ограничиться индивидуальными задачами, в которых мощность множеств не превосходит заданной величины и все элементы содержатся не более чем в ограниченном числе множеств. Можно потребовать, чтобы любые величины, сопоставляемые элементами множества, принадлежали некоторому ограниченному множеству значений. Из всех этих возможных задач для детального анализа обычно выбираются те, которые имеют известные приложения или в каком-то смысле являются

такие алгоритмы оцениваются и сравниваются на основе сочетания эмпирических данных и аргументов, опирающихся на здравый смысл.

Однако совсем недавно был получен ряд результатов, которые показали, что эвристические алгоритмы не всегда столь сложны для формального анализа. В некоторых случаях удается доказать, что решения, получаемые эвристическим алгоритмом, всегда будут отличаться в процентном отношении от оптимального решения не более чем на определенную величину. Подобные результаты можно рассматривать как «оценки погрешности» алгоритмов. Дадим формальное описание понятия «оптимизационная задача».

КОМБИНАТОРНАЯ ОПТИМИЗАЦИОННАЯ ЗАДАЧА
 P есть либо задача минимизации, либо задача максимизации и состоит из следующих трех частей: 1) множества D_P индивидуальных задач; 2) для каждой $I \in D_P$ конечного множества $S_P(I)$ допустимых решений индивидуальной задачи I ; 3) функции m_P , сопоставляющей каждой индивидуальной задаче $I \in D$ и каждому допустимому решению $\sigma \in S_P(I)$ некоторое положительное целое число $m_P(I, \sigma)$, называемое величиной решения I .

Если P — задача минимизации (соответственно максимизации), то **ОПТИМАЛЬНЫМ РЕШЕНИЕМ** индивидуальной задачи $I \in D_P$ является такое допустимое решение $\sigma^* \in S_P(I)$, что для всех $\sigma \in S_P(I)$ выполнено неравенство $m_P(I, \sigma^*) \leq m_P(I, \sigma)$ (соответственно $m_P(I, \sigma^*) \geq m_P(I, \sigma)$). Для обозначения величины $m_P(I, \sigma^*)$ оптимального решения индивидуальной задачи I будем использовать символ $ОПТ_P(I)$ или то же с опущением индекса P .

Алгоритм A называется **ПРИБЛИЖЕННЫМ АЛГОРИТМОМ** решения задачи P , если для любой индивидуальной задачи $I \in D_P$ алгоритм A отыскивает некоторое допустимое решение $\sigma \in S_P(I)$. Через $A(I)$ будем обозначать величину $m_P(I, \sigma)$ того возможного решения σ , которое A строит по I . Если $A(I) = ОПТ_P(I)$ для всех $I \in D_P$, то A называется точным алгоритмом решения задачи P .

Приведенные определения можно проиллюстрировать на примере уже известной задачи о коммивояжере. Это задача минимизации, множество ее индивидуальных задач состоит из всех конечных множеств городов и попарных расстояний между ними. Допустимые решения конкретной индивидуальной задачи представляют собой все возможные перестановки заданных городов. Величина решения для каждой такой перестановки равна длине соответствующего маршрута. Таким образом, приближенный алгоритм для данной задачи должен отыскать только некоторую перестановку городов, а точный

алгоритм должен всегда отыскать перестановку, соответствующую маршруту минимальной длины.

Если оптимизационная задача не принадлежит классу P , то нельзя построить точный полиномиальный алгоритм решения этой задачи. Более реалистично попытаться найти приближенный алгоритм A , время работы которого ограничено полиномом невысокой степени и который обладает тем свойством, что для всех индивидуальных задач I величина $A(I)$ «близка» к величине $ОПТ_P(I)$.

Рассмотрим задачу об «упаковке в контейнеры». Она формулируется следующим образом: задано конечное множество $U = \{u_1, u_2, \dots, u_n\}$ «предметов» и «размеры» $S(u) \in [0, 1]$ каждого предмета $u \in U$ (размер предмета представлен рациональным числом); требуется найти такое разбиение множества предметов на непересекающиеся подмножества U_1, U_2, \dots, U_k , чтобы сумма размеров предметов в каждом подмножестве не превосходила 1 и чтобы число k было наименьшим возможным. Можно сказать, что предметы, принадлежащие каждому множеству u_i , упаковываются в один контейнер размера 1, а наша цель — упаковать предметы множества U в как можно меньшее число контейнеров.

Рассматриваемая задача NP -полна, так что мало надежды на отыскание полиномиального алгоритма решения этой задачи. Однако для ее решения имеется несколько заслуживающих внимания простых алгоритмов. Один из них известен под названием «в первый подходящий». Предположим, что существует бесконечная последовательность контейнеров размера 1, каждый из которых в начальный момент времени пуст. Алгоритм заключается в том, чтобы помещать предметы в контейнеры по очереди, в порядке возрастания номеров. Предметы помещаются в контейнеры согласно следующему простому правилу: очередной предмет u_i помещается в контейнер с наименьшим номером, у которого сумма размеров помещенных в него предметов не превосходит 1 — $S(u_i)$. Другими словами, u_i помещается в первый из контейнеров, куда он может попасть.

С интуитивной точки зрения этот алгоритм кажется очень естественным и разумным. Он не начинает заполнять новый контейнер до тех пор, пока все непустые контейнеры не будут слишком заполнены. Вначале рассмотрим число контейнеров, требующихся для указанного алгоритма, как функцию от параметров задачи. Обозначим эту функцию через $FF(I)$. Имеют место неравенства $FF(I) \geq 1$; $FF(I) < \left[2 \sum_{i=1}^n S(u_i) \right]$.

Последнее неравенство вытекает из того, что в результате работы алгоритма не может получиться более одного контейнера, заполненного менее, чем наполовину. То, что указанная

граница — наилучшая из возможных, следует из рассмотрения индивидуальной задачи $U = \{u_1, u_2, \dots, u_n\}$, где $S(u_i) = \frac{1}{2} + \epsilon$, $1 \leq i \leq n$. Поскольку никакие два предмета не могут попасть в один контейнер, то $FF(I) = n$. Так как сумма размеров предметов равна $n/2 + n\epsilon$, она будет как угодно близка $kn/2$, если выбрать ϵ достаточно малым.

Сделанное замечание позволяет нам также оценить, во сколько раз плохое решение, получаемое алгоритмом «в первый подходящий», может отличаться от оптимального решения задачи.

Поскольку

$$OPT(I) \geq \left[\sum_{i=1}^n S(U_i) \right],$$

то для всех индивидуальных задач I выполнено неравенство $FF(I) \leq 2OPT(I)$. Однако, как следует из теоремы, существует лучшая оценка.

Теорема 3.5.1. Для всех индивидуальных задач I об упаковке имеет место неравенство $FF(I) \leq 17/10 OPT(I) + 2$.

Более того, существуют индивидуальные задачи I , для которых $OPT(I)$ сколь угодно велико и $FF(I) \geq 17/10 [OPT(I) - 1]$.

ОПЕРАЦИИ НАД ОБЫКНОВЕННЫМИ ГРАФАМИ

Все операции над обыкновенными графами, рассматриваемые в данной главе [1, 2], будут производиться над соответствующими этим графам формальными квадратичными формами [3].

4.1. ОПЕРАЦИИ НАД ВЕРШИНАМИ И РЕБРАМИ

Прежде, чем перейти к описанию операций вида удаление, склеивание вершин, а также стягивание ребер и склеивание ребер обыкновенных графов, предварительно введем и определим операцию «подстановка» над квадратичными формами.

Определение. Пусть $g(v_0, v_1, v_2, \dots, v_p) = \sum_{i,j=1}^p \alpha_{i,j} v_i v_j$ — квадратичная форма, соответствующая некоторому обыкновенному графу G с множеством вершин из $V = \{v_1, v_2, \dots, v_p\}$. Тогда подстановка y вместо символа v_i в форму $g(v_0, v_1, \dots, v_p)$ состоит в одновременной замене каждого вхождения символа v_i в $g(v_0, v_1, \dots, v_i, \dots, v_p)$ на вхождение y . Результат подстановки y вместо v_i в $g(v_0, v_1, v_2, \dots, v_p)$ запишем в виде

$$g(v_0, v_1, \dots, v_{i-1}, y, v_{i+1}, \dots, v_p) \hat{=} S_y^{v_i} [g(v_0, v_1, v_2, \dots, v_i, \dots, v_p)].$$

Из определения операции подстановки непосредственно вытекает:

Лемма 4.1.1. Если $g(v_0, v_1, v_2, \dots, v_p)$ — квадратичная форма графа G , то $S_y^{v_i} [g(v_0, v_1, v_2, \dots, v_p)]$ также является квадратичной формой некоторого графа H (возможно, отличного от G), каковы бы ни были v_i и y .

В частности, при $y = v_j$ ($i \neq j$) подстановка y вместо v_i в форму $g(v_0, v_1, v_2, \dots, v_p)$ соответствует операции склеивания вершины v_i с вершиной v_j , в том числе и операции стягивания ребра $v_i v_j$ при $\alpha_{i,j} = 1$; а при $y = v_0$ — операции удаления вершины v_i графа G .

Очевидно, справедливо также следующее утверждение (свойство подстановки).

Лемма 4.1.2.

$$\begin{aligned} & \forall i \forall j \forall k \forall l (i \neq j) \& (l \neq k) \& (k \neq j) \times \\ & \times \{S_{v_k}^{v_i} [S_{v_l}^{v_j} [g(v_0, v_1, v_2, \dots, v_p)]] = \\ & = S_{v_l}^{v_j} [S_{v_k}^{v_i} [g(v_0, v_1, v_2, \dots, v_p)]] = S_{v_k, v_l}^{v_i, v_j} [g(v_0, v_1, v_2, \dots, v_p)] \}, \end{aligned}$$

где $S_{v_k, v_l}^{v_i, v_j} [g(v_0, v_1, v_2, \dots, v_p)]$ — результат одновременной подстановки v_k вместо v_i и v_l вместо v_j в форму $g(v_0, v_1, v_2, \dots, v_p)$.

Следует заметить, что если квадратичная форма $g(v_0, v_1, v_2, \dots, v_p) = \sum_{i,j} \alpha_{i,j} v_i v_j$ графа G такая, что

$$(\alpha_{i,j} = 1) \& (\alpha_{k,l} = 1) \& (i \neq j) \& (k \neq l) \& [(i \neq k) \vee (j \neq l)],$$

то $S_{v_k, v_l}^{v_i, v_j} [g(v_0, v_1, v_2, \dots, v_p)]$ — квадратичная форма некоторого графа G' , полученного из графа G склеиванием ребра $v_i v_j$ с ребром $v_k v_l$.

Далее, обобщив операции подстановки естественным образом и на случаи, когда 1) $v_i = \alpha_{i,j}$, $y = 0$; 2) $y = x + z$, где $v_j \neq x$ и $v_l \neq y$ ($j = \overline{1, p}$), получим, что в первом случае операция подстановки будет соответствовать операции удаления ребра $v_i v_j$, а во втором — операции расщепления вершины v_l на две разные вершины x и z , отличные от всех остальных вершин графа G .

Композиция. Прежде, чем перейти к изложению операции композиции, предварительно введем и рассмотрим еще одну бинарную (мультипликативную) операцию « \times » (умножение), определенную на множестве \mathfrak{M} следующими тождественными соотношениями: 1) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$; 2) $(a + b) \cdot c = a \cdot c + b \cdot c$; 3) $c \cdot (a + b) = c \cdot a + c \cdot b$; 4) $a^n = a \cdot a \cdot \dots \cdot a$ (n — сомножителей); 5) $n(a \cdot b) = na \cdot b = a \cdot nb$.

В силу допущения в 1.5 имеем еще соотношения:

6) $a^n = a$ (вместо соотношения 4); 7) $n(a \cdot b) = a \cdot b$ (вместо соотношения 5); 8) $a \cdot b = b \cdot a$, причем если положить, что $a = (x, y)$, $b = (u, v)$, то

$$9) a \cdot b = (x, y) \cdot (u, v) \doteq (x, y) + (x, u) + (x, v) + (y, u) + (y, v) + (u, v).$$

Определение. Пусть $g(v_0, y_1, y_2, \dots, y_n)$, $f_1(v_0, x_1^{(1)}, x_2^{(1)}, \dots, x_{p_1}^{(1)})$, ..., $f_n(v_0, x_1^{(n)}, x_2^{(n)}, \dots, x_{p_n}^{(n)})$ — квадратичные формы соответственно графов G, F_1, F_2, \dots, F_n , удовлетворяющие условиям: а) $\forall i \forall j (y_i \neq y_j)$, ($i, j = \overline{1, n}$); б) $\forall i \forall j \forall k (y_i \neq x_j^{(k)})$, ($k,$

$t = \overline{1, n}$; $j = \overline{1, p_n}$); в) $\forall i \forall j \forall k \forall l (x_i^{(k)} \neq x_j^{(l)})$, ($k, l = \overline{1, n}$; $i, j = \overline{1, p_n}$).

Тогда квадратичную форму вида $\psi(v_0, x_1^{(1)}, x_2^{(1)}, \dots, x_{p_1}^{(1)}, x_1^{(2)}, x_2^{(2)}, \dots, x_{p_2}^{(2)}, \dots, x_1^{(n)}, x_2^{(n)}, \dots, x_{p_n}^{(n)}) = g(v_0, f_1, f_2, \dots, f_1, \dots, f_n) = S_{f_{i_n}}^{y_n} [S_{f_{i_{n-1}}}^{y_{n-1}} [\dots S_{f_{i_1}}^{y_1} [g(v_0, y_1, y_2, \dots, y_1, \dots, y_n)]] \dots]]$, где $f_i(v, x_1^{(i)}, \dots, x_{p_i}^{(i)}) y_i = y_i f_i(v_0, x_1^{(i)}, x_2^{(i)}, \dots, x_{p_i}^{(i)}) = f_i(v_0, x_1^{(i)}, \dots, x_{p_i}^{(i)}) v_0 y_i + x_1^{(i)} y_i + \dots + x_{p_i}^{(i)} y_i$ определим как квадратичную форму, полученную с помощью операции суперпозиции из квадратичных форм g, f_1, f_2, \dots, f_n .

Очевидно, если для квадратичных форм g, f_1, f_2, \dots, f_n выполняются условия «а», «б», «в», то порядок, в каком будем делать эти подстановки, безразличен и результаты подстановок будут совпадать с результатом одновременной замены «переменных» y_i соответственно квадратичными формами f_i , т. е. справедливо соотношение

$$\begin{aligned} & S_{f_{i_n}}^{y_n} [S_{f_{i_{n-1}}}^{y_{n-1}} [\dots S_{f_{i_1}}^{y_1} [g(v_0, y_1, \dots, y_1, \dots, y_n)]] \dots] = \\ & = S_{f_{i_1}, \dots, f_{i_n}}^{y_1, \dots, y_n} [g(v_0, y_1, \dots, y_n)]. \end{aligned}$$

Аналогично, граф Ψ , соответствующий квадратичной форме ψ , получается суперпозицией из графов G, F_1, F_2, \dots, F_n . Из формулировки операции суперпозиции непосредственно следует:

Лемма 4.1.3. Если $g(v_0, y_1, y_2, \dots, y_n)$, $f_1(v_0, x_1^{(1)}, \dots, x_{p_1}^{(1)})$, ..., $f_n(v_0, x_1^{(n)}, \dots, x_{p_n}^{(n)})$ — квадратичные формы соответственно графов

$$G, F_1, F_2, \dots, F_n, \text{ то } S_{f_{i_1}, \dots, f_{i_n}}^{y_1, \dots, y_n} [g(v_0, y_1, y_2, \dots, y_n)]$$

также является квадратичной формой некоторого графа Ψ , каковы бы ни были квадратичные формы, удовлетворяющие условиям «а» — «в».

Пример. Пусть $g(v_0, y_1, y_2, y_3) = v_0(y_1 + y_2 + y_3) + y_1 y_2$; $f_1(v_0, x_1, x_2, x_3) = v_0(x_1 + x_2 + x_3) + x_1(x_2 + x_3) + x_2 x_3$; $f_2(v_0, a_1, a_2) = v_0(a_1 + a_2) + a_1 a_2$; $f_3(v_0, b_1, b_2, b_3) = v_0(b_1 + b_2 + b_3) + b_1(b_2 + b_3) + b_2 b_3$ — квадратичные формы соответственно графов G, F_3, P_2, F_3 . Тогда квадратичная форма $\psi(v_0, x_1, x_2, x_3, a_1, a_2, b_1, b_2,$

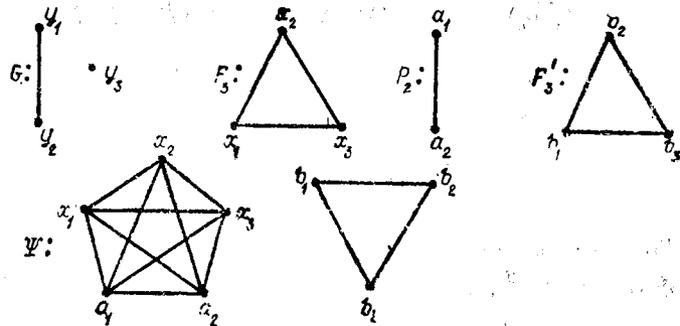


Рис. 4.1.1

b_3), соответствующая суперпозиции

$$S_{f_1, f_2, f_3}^{y_1, y_2, y_3} [g(v_0, y_1, y_2, y_3)],$$

имеет вид $\psi(v_0, x_1, x_2, x_3, a_1, a_2, b_1, b_2, b_3) = v_0(x_1 + x_2 + x_3 + a_1 + a_2 + b_1 + b_2 + b_3) + x_1(x_2 + x_3 + a_1 + a_2) + x_2 \times (x_3 + a_1 + a_2) + x_3(a_1 + a_2) + a_1 a_2 + b_1(b_2 + b_3) + b_2 b_3$.

На рис. 4.1.1 изображены графы G, F_3, P_2, F_3 и Ψ , соответствующие квадратичным формам g, f_1, f_2, f_3 и ψ . Пусть теперь $h(v_0, y_1, y_2, \dots, y_m)$ и $g(v_0, x_1, \dots, x_n)$ — квадратичные формы соответственно графов H и G , удовлетворяющие условиям: а) $\forall i \forall j (x_i \neq x_j), (i, j = \overline{1, n})$; б) $\forall i \forall j (y_i \neq y_j), (i, j = \overline{1, m})$; в) $\forall i \forall j (x_i \neq y_j), (i = \overline{1, n}; j = \overline{1, m})$.

Рассмотрим квадратичную форму вида

$$q(v_0, x_1^{(1)}, \dots, x_n^{(1)}, x_1^{(2)}, \dots, x_n^{(2)}, \dots, x_1^{(m)}, \dots, x_n^{(m)}) = S_{g^{(1)}, \dots, g^{(m)}}^{y_1, \dots, y_m} [h(v_0, y_1, \dots, y_i, \dots, y_m)],$$

где

$$g^{(i)}(v_0, x_1^{(i)}, \dots, x_n^{(i)}) = S_{x_1^{(i)}, \dots, x_j^{(i)}, \dots, x_n^{(i)}}^{x_1, \dots, x_j, \dots, x_n} \times [g(v_0, x_1, \dots, x_j, \dots, x_n)] \quad (i = \overline{1, m}),$$

причем $\forall i \forall j \forall k \forall l (x_i^{(k)} \neq x_j^{(l)}) (i, j = \overline{1, n}; k, l = \overline{1, m})$.

Будем говорить, что граф Q , соответствующий квадратичной форме q , получается с помощью операции композиции из графов H и G . Результат композиции графов H и G обозначим через $Q = G[H]$, причем граф $G[H]$ отличен от графа $H[G]$.

Пример. Пусть графы P_2, H_3, F_3 (рис. 4.1.2) задаются соответственно квадратичными формами:

$$\begin{aligned} p_2(v_0, x_1, x_2) &= v_0(x_1 + x_2) + x_1 x_2; \\ p_3(v_0, y_1, y_2, y_3) &= v_0(y_1 + y_2 + y_3) + y_2(y_1 + y_3); \\ f_3(v_0, z_1, z_2, z_3) &= v_0(z_1 + z_2 + z_3) + z_1(z_2 + z_3) + z_2 z_3. \end{aligned}$$

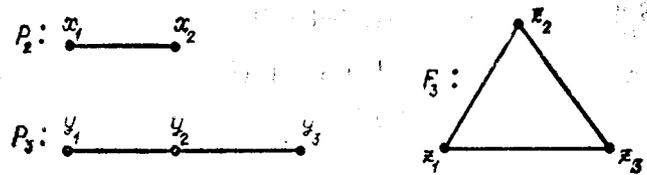


Рис. 4.1.2

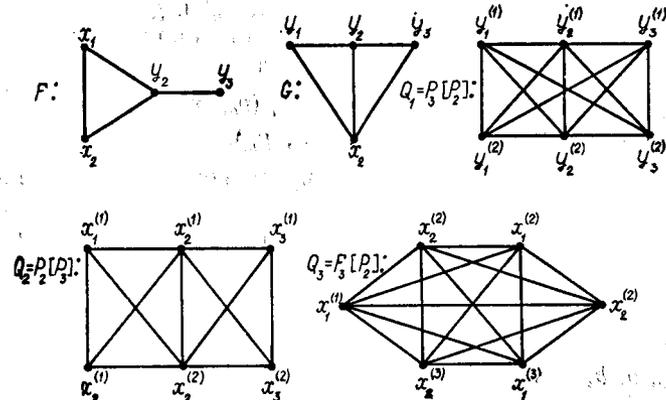


Рис. 4.1.3

Приведенные ниже квадратичные формы f, g, q_1, q_2, q_3 определяют соответственно графы F, G, Q_1, Q_2, Q_3 (рис. 4.1.3):

$$\begin{aligned} f &= S_{p_2}^{y_1} [p_3(v_0, y_1, y_2, y_3)] = v_0(x_1 + x_2 + y_2 + y_3) + x_1 x_2 + y_2(x_1 + x_2 + y_3); \\ g &= S_{p_2}^{x_1} [p_2(v_0, x_1, x_2)] = v_0(y_1 + y_2 + y_3 + x_2) + x_2(y_1 + y_2 + y_3) + y_2(y_1 + y_3); \\ q_1 &= S_{p_3, p_3}^{x_1, x_2} [p_2(v_0, x_1, x_2)] = v_0(y_1^{(1)} + y_2^{(1)} + y_3^{(1)} + y_1^{(2)} + y_2^{(2)} + y_3^{(2)}) + y_1^{(1)}(y_2^{(1)} + y_1^{(2)} + y_2^{(2)} + y_3^{(2)}) + y_2^{(1)}(y_3^{(1)} + y_1^{(2)} + y_2^{(2)} + y_3^{(2)}) + y_3^{(1)}(y_1^{(2)} + y_2^{(2)} + y_3^{(2)}) + y_2^{(2)}(y_1^{(2)} + y_3^{(2)}); \\ q_2 &= S_{p_2, p_2, p_2}^{y_1, y_2, y_3} [p_3(v_0, y_1, y_2, y_3)] = v_0(x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_1^{(1)}(x_2^{(1)} + x_2^{(2)} + x_1^{(2)} + x_1^{(3)} + x_2^{(3)} + x_2^{(2)} + x_2^{(3)}) + x_1^{(2)}(x_2^{(2)} + x_2^{(3)} + x_1^{(3)} + x_2^{(3)}); \end{aligned}$$

$$q_3 = S_{p_2, p_2, p_2}^{z_1, z_1, z_1} [f_3(v_0, z_1, z_2, z_3)] = (x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)})v_0 + x_1^{(1)}(x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_2^{(1)}(x_2^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_1^{(2)}(x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_2^{(2)}(x_1^{(3)} + x_2^{(3)}) + x_1^{(3)}x_2^{(3)}.$$

Следует заметить, что для произвольных m и n графы $F_m [F_n]$ и $F_m [F_n]$ изоморфны графу F_{mn} .

Объединение. Пусть $f(v_0, x_1, x_2, \dots, x_n)$ и $g(v_0, y_1, y_2, \dots, y_m)$ — квадратичные формы соответственно графов F и G . Рассмотрим квадратичную форму вида

$$h(v_0, x_1, \dots, x_n, y_1, \dots, y_m) = f(v_0, x_1, \dots, x_n) + g(v_0, y_1, \dots, y_m).$$

Будем говорить, что граф H , определяемый квадратичной формой $h(v_0, x_1, \dots, x_n, y_1, \dots, y_m)$, получается объединением графов F и G . Объединение графов F и G обозначим символом $F \cup G$.

Пример. Положив

$$f(v_0, a, b, c, d, e) = v_0(a + b + c + d + e) + a(b + c) + bc + ed,$$

$$g(v_0, a, b, c, x, y, z) = v_0(a + b + c + x + y + z) + a \times (x + c + z) + c(z + y) + b(x + y),$$

имеем

$$h(v_0, a, b, c, d, e, x, y, z, u, v, w) = f(v_0, a, b, c, d, e) + g(v_0, x, y, z, u, v, w).$$

Графы F , G и H , определяемые данными квадратичными формами f , g и h , иллюстрируются рис. 4.1.4.

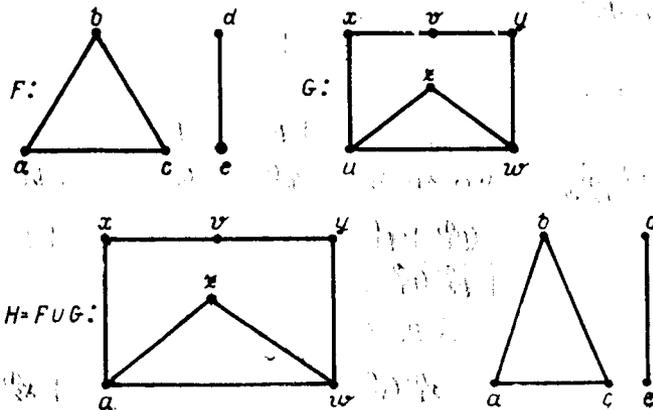


Рис. 4.1.4

Конъюнкция. Декартовое произведение.

Пусть $f = \sum_{i,j=0}^m \alpha_{i,j} x_i x_j$ и $g = \sum_{i,j=0}^p \beta_{i,j} y_i y_j$ — квадратичные формы соответственно графов F и G . Рассмотрим квадратичные формы

$$f^{(t)} = S_{x_1^{(t)}, x_2^{(t)}, \dots, x_m^{(t)}}^{x_1, x_2, \dots, x_m} [f]; \quad \bar{h} = S_{y_1^{(t)}, \dots, y_p^{(t)}}^{y_1, \dots, y_p} [g] =$$

$$= \sum_{\substack{i,j=0,m \\ k,l=0,p}} \delta_{i,j}^{k,l} x_i^{(k)} x_j^{(l)} = \sum_{\substack{i,j=0,m \\ k,l=0,p}} (\alpha_{i,j} \beta_{k,l}) x_i^{(k)} x_j^{(l)};$$

$$h = S_{y_1^{(t)}, \dots, y_p^{(t)}}^{y_1, \dots, y_p} [h],$$

где коэффициенты $\gamma_{i,j}^{k,l}$ удовлетворяют одному из следующих ниже условий:

- а) $\gamma_{i,j}^{k,l} = \begin{cases} 0, & \text{если } (\alpha_{i,j} = 0) \vee (\beta_{k,l} = 0); \\ 1, & \text{если } (\alpha_{i,j} = 1) \& (\beta_{k,l} = 1); \end{cases}$
- б) $\gamma_{i,j}^{k,l} = \begin{cases} 1, & \text{если } [(i = j) \& (\beta_{k,l} = 1) \vee (k = l) \& (\alpha_{i,j} = 1)]; \\ 0 & \text{в противном случае.} \end{cases}$

Говорят, что граф H , определяемый квадратичной формой $h(v_0, x_1^{(1)}, \dots, x_m^{(1)}, \dots, x_1^{(p)}, \dots, x_m^{(p)})$, является конъюнкцией или декартовым произведением графов F и G в зависимости от того, удовлетворяют ли коэффициенты $\gamma_{i,j}^{k,l}$ квадратичной формы h условию «а» или «б». Конъюнкцию графов F и G обозначают символом $F \& G$, декартовое произведение — $F \times G$.

Пример. Пусть графы P_2, P_3, F_3 определяются соответственно квадратичными формами $p_2(v_0, x_1, x_2), p_3(v_0, y_1, y_2, y_3), f_3(v_0, z_1, z_2, z_3)$. Тогда квадратичные формы

$$h_1(v_0, x_1^{(1)}, x_2^{(1)}, x_1^{(2)}, x_2^{(2)}, x_1^{(3)}, x_2^{(3)}) = v_0(x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_1^{(1)}x_2^{(2)} + x_2^{(2)}x_1^{(3)} + x_1^{(2)}x_2^{(1)} + x_1^{(2)}x_2^{(3)};$$

$$h_2(v_0, x_1^{(1)}, x_2^{(1)}, x_1^{(2)}, x_2^{(2)}, x_1^{(3)}, x_2^{(3)}) = v_0(x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_1^{(1)}x_2^{(2)} + x_1^{(1)}x_2^{(3)} + x_1^{(2)}x_2^{(1)} + x_1^{(2)}x_2^{(3)} + x_1^{(3)}x_2^{(1)} + x_1^{(3)}x_2^{(2)};$$

$$h_3(v_0, x_1^{(1)}, x_2^{(1)}, x_1^{(2)}, x_2^{(2)}, x_1^{(3)}, x_2^{(3)}) = v_0(x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_1^{(1)}x_2^{(2)} + x_1^{(1)}x_2^{(3)} + x_2^{(2)}x_1^{(1)} + x_2^{(2)}x_2^{(3)} + x_2^{(2)}x_1^{(2)} + x_1^{(3)}x_1^{(2)} + x_1^{(3)}x_2^{(3)};$$

$$h_4(v_0, x_1^{(1)}, x_2^{(1)}, x_1^{(2)}, x_2^{(2)}, x_1^{(3)}, x_2^{(3)}) = v_0(x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_1^{(3)} + x_2^{(3)}) + x_1^{(1)}x_1^{(2)} + x_1^{(1)}x_1^{(3)} + x_1^{(1)}x_2^{(1)} + x_2^{(3)}x_1^{(3)} + x_2^{(3)}x_2^{(1)} + x_2^{(3)}x_2^{(2)} + x_2^{(1)}x_2^{(2)} + x_1^{(2)}x_1^{(3)} + x_2^{(1)}x_2^{(2)}$$

определяют соответственно графы: $H_1 = P_2 \& P_3$, $H_2 = P_2 \& F_3$, $H_3 = P_2 \times P_3$, $H_4 = P_2 \times F_3$ (рис. 4.1.5).

Соединение. Разъединение. Пусть $p(v_0, x_1, x_2, \dots, x_n)$, $q(v_0, y_1, y_2, \dots, y_m)$ — квадратичные формы соответственно графов P и Q .

Соединением двух графов P и Q назовем граф S , определяемый квадратичной формой вида

$$S(v_0, x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) \doteq p(v_0, x_1, \dots, x_n) + g(v_0, y_1, \dots, y_m) + \sum_{\substack{i=1, \dots, n \\ j=1, \dots, m}} x_i y_j$$

Соединение графов P и Q обозначим через $S = P + Q$.

Разъединением двух графов P и Q , где граф Q подграф графа S , назовем граф P , определяемый квадратичной формой вида

$$p(v_0, x_1, \dots, x_n) \doteq s(v_0, x_1, \dots, x_n, y_1, \dots, y_m) \doteq g(v_0, y_1, \dots, y_m) = S_{v_0, \dots, v_0}^{y_1, \dots, y_m} [s(v_0, x_1, \dots, x_n, y_1, \dots, y_m)].$$

Разъединение графов S и Q обозначим через $P = S \div Q$.

Очевидно, что операции соединения и разъединения графов взаимобратны, т. е. если $S = P + Q$, то справедливы $P = S \div Q$; $Q = S \div P$.

Пример. Если в качестве графов P и Q взять соответственно графы P_2 и P_3 (см. рис. 4.2.2), их соединение $S = P_2 + P_3$ (рис. 4.1.6) определяется квадратичной формой

$$s(v_0, x_1, x_2, y_1, y_2, y_3) = v_0(x_1 + x_2 + y_1 + y_2 + y_3) + x_1(x_2 + y_1 + y_2 + y_3) + x_2(y_1 + y_2 + y_3) + y_2(y_1 + y_3).$$

ДОПОЛНЕНИЕ. Пусть граф G определяется квадратичной формой

$$g(v_0, x_1, x_2, \dots, x_p) = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$$

Дополнением графа G называется граф, определяемый квадратичной формой $\bar{g}(v_0, x_1, x_2, \dots, x_p)$ вида

$$\bar{g}(v_0, x_1, x_2, \dots, x_p) \doteq S_{1-\alpha_{i,j}}^{i,j} [g] = \sum_{i,j=0}^p \beta_{i,j} x_i x_j,$$

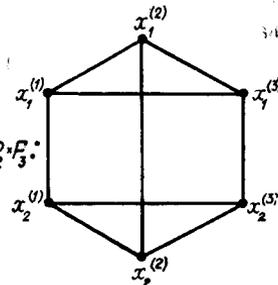
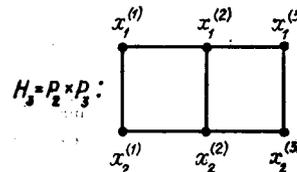
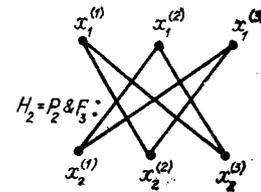
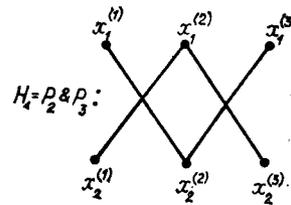


Рис. 4.1.5

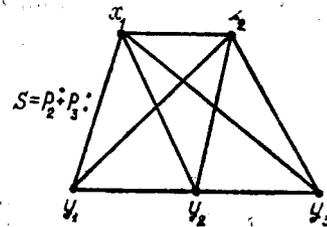


Рис. 4.1.6

где

$$\beta_{i,j} = \begin{cases} \alpha_{i,j} & \text{при } (i=0) \vee (j=0) \vee (i=j); \\ 1 - \alpha_{i,j} & \text{в противоположном случае.} \end{cases}$$

Дополнение графа G обозначают обычно символом \bar{G} . Если графы G и \bar{G} изоморфны, то граф G самодополнительный.

Пример. Пусть $G = G_5$, т. е. граф G_5 определяется квадратичной формой

$$c_5(v_0, x_1, x_2, x_3, x_4, x_5) = v_0(x_1 + x_2 + x_3 + x_4 + x_5) + x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_1.$$

Тогда дополнение графа G_5 , т. е. граф \bar{G}_5 (рис. 4.1.7) определяется квадратичной формой

$$c_5(v_0, x_1, x_2, x_3, x_4, x_5) = v_0(x_1 + x_2 + x_3 + x_4 + x_5) + x_1x_3 + x_3x_5 + x_5x_2 + x_2x_4 + x_4x_1.$$

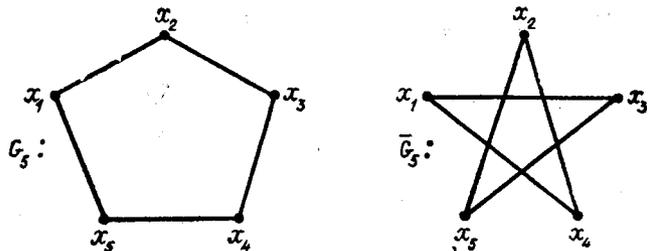


Рис. 4.1.7

Упражнения.

1. Выполнить следующие операции подстановки и нарисовать графы, соответствующие квадратичным формам — результатам подстановок:

- а) $S_{x_5}^{x_2} [f_5(x_0, x_1, x_2, \dots, x_5)]$ (удаление вершины x_5);
- б) $S_{x_5}^{x_{10}} [l(x_0, x_1, x_2, \dots, x_{10})]$ (склеивание вершин x_5 и x_{10});
- в) $S_{x_4}^{x_5} [l(x_0, x_1, x_2, \dots, x_{10})]$ (стягивание ребра x_4x_5);
- г) $S_{x_4x_5}^{x_7} [f_8(x_0, x_1, x_2, \dots, x_{10})]$ (склеивание ребер x_4x_5 и x_7x_8);
- д) $S_0^{\alpha_9, 10} [l(x_0, x_1, x_2, \dots, x_{10})]$ (удаление ребра x_9x_{10});
- е) $S_0^{\alpha_4, 5} [S_{x_4+x_5}^{x_4} [f_4(x_0, x_1, x_2, x_3, x_4)]]$ (расщепление вершины x_4 на вершины x_4 и x_5), где f_4, f_5, f_8 — квадратичные формы соответственных полных графов F_4, F_5 и F_8 , а l — квадратичная форма графа L , изображенного на рис. 4.1.8.

2. Выполнить операции суперпозиции и нарисовать графы соответствующие результатам операций:

$$S_{p_2, c_4}^{z_1, z_2} [p_2(z_0, z_1, z_2)]; S_{c_3, c_4}^{z_1, z_2} [p_3(z_0, z_1, z_2, z_3)],$$

где $p_2(z_0, z_1, z_2) = z_0 [(z_1 + z_2) + z_1 z_2]$; $p_2(x_0, x_1, x_2) = x_0 (x_1 + x_2) + x_1 x_2$;
 $c_5(x_0, x_1, x_2, x_3) = x_0 (x_1 + x_2 + x_3) + x_1 x_2 + x_2 x_3 + x_3 x_1$;
 $c_4(y_0, y_1, y_2, y_3, y_4) = y_0 (y_1 + y_2 + y_3 + y_4) + y_1 y_2 + y_2 y_3 + y_3 y_4 + y_4 y_1$.
 Для графов P_2, P_3, C_4 , сделав пометки, выполнить операции $P_2 [C_4]$, $C_4 [P_3]$, $P_2 \times P_3$, $P_2 \& P_3$, $P_2 + C_3$ и $P_2 + C_3$;

написать квадратичные формы и нарисовать графы, соответствующие результатам этих операций.

4. Определить группу автоморфизмов для графа L , изображенного на рис. 4.1.8.

5. Построить граф с четным числом ≥ 4 квазиподобных вершин.

6. Построить граф с нечетным числом квазиподобных вершин. (Построение такого графа нам представляется трудным.)

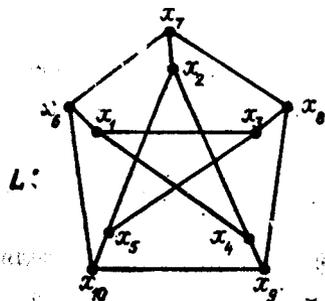


Рис. 4.1.8

Глава 5
ВОССТАНОВЛЕНИЕ ГРАФОВ

5.1. ИЗОМОРФИЗМ

Пусть даны квадратичные формы

$$g = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j \text{ и } h = \sum_{i,j=0}^p \beta_{i,j} x_i x_j,$$

соответствующие (p, q) графам G и H , вершины которых помечены произвольным образом соответственно символами $x_0, x_1, \dots, x_i, \dots, x_p$ и $y_0, y_1, \dots, y_i, \dots, y_p$ ($x_i = y_i$).

Графы G и H называются *изоморфными* $G \simeq H$, если между множествами $X = \{x_i\}$ и $Y = \{y_i\}$ их вершин можно установить взаимно однозначное соответствие, сохраняющее отношение смежности вершин, т. е. существует хотя бы одна биекция

$$\Phi = \begin{pmatrix} y_0, y_1, \dots, y_j, \dots, y_p \\ x_0, x_1, \dots, x_i, \dots, x_p \end{pmatrix}$$

такая, что имеет место $\Phi(h) = g = \forall j \{S_{x_i}^{y_j} [h]\}$. При этом само соответствие Φ называется *изоморфизмом графов*.

Не нарушая общности, можем полагать, что вершины (p, q) графов G и H помечены символами одного и того же множества, скажем $X = \{x_0, x_1, \dots, x_i, \dots, x_p\}$, причем метки вершин одного графа расставлены независимо от расстановки меток вершин другого графа. Тогда графы G и H назовем изоморфными, если существует хотя бы одна перестановка

$$\Phi = \begin{pmatrix} 1 & 2 & \dots & i & \dots & p \\ j_1 & j_2 & \dots & j_i & \dots & j_p \end{pmatrix}$$

такая, что $\Phi(h) = g$.

Например,

$$g_1 = \sum_{i=1}^6 x_0 x_i + x_1 (x_4 + x_5 + x_6) + x_2 (x_4 + x_5 + x_6) + x_3 (x_4 + x_5 + x_6);$$

$$g_2 = \sum_{i=1}^6 x_0 y_i + y_1 (y_4 + y_5 + y_6) + y_2 (y_4 + y_5 + y_6) + y_3 (y_4 + y_5 + y_6);$$

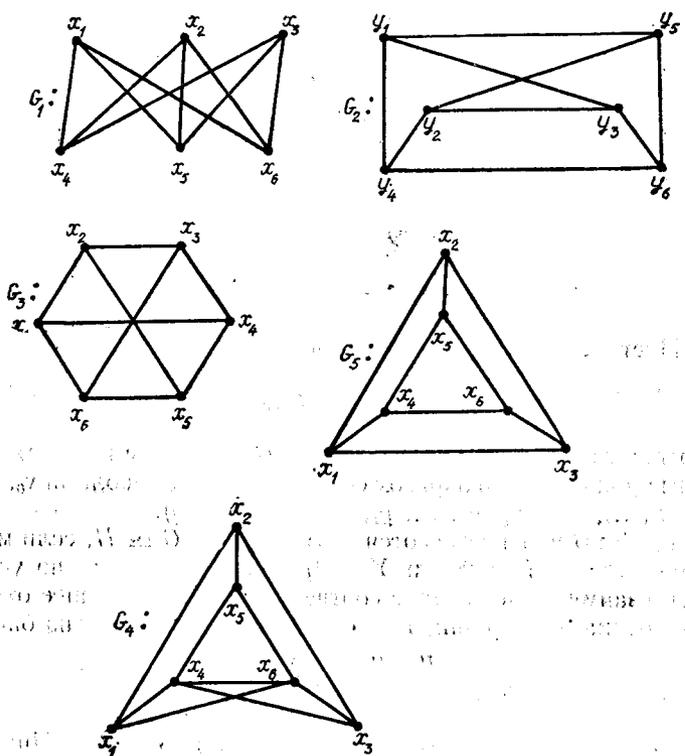


Рис. 5.1.1

$$g_3 = \sum_{i=1}^6 x_0 x_i + x_1(x_2 + x_4 + x_6) + x_3(x_2 + x_4 + x_6) + x_5(x_2 + x_4 + x_6);$$

$$g_4 = \sum_{i=1}^6 x_0 x_i + x_2(x_1 + x_3 + x_5) + x_4(x_1 + x_3 + x_5) + x_6(x_1 + x_3 + x_5);$$

$$g_5 = \sum_{i=1}^6 x_0 x_i + x_1(x_2 + x_3 + x_4) + x_5(x_2 + x_4 + x_6) + x_6(x_3 + x_4) + x_2 x_3$$

есть квадратичные формы, соответствующие пяти 3-однородным графам G_i ($i = 1, \dots, 5$) (рис. 5.1.1).

Очевидно, что $G_1 \simeq G_2 \simeq G_3 \simeq G_4 \not\simeq G_5$. Действительно, если положить

$$\Phi_1 = \begin{pmatrix} y_1 & \dots & y_1 & \dots & y_6 \\ x_1 & \dots & x_i & \dots & x_6 \end{pmatrix}; \quad \Phi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 6 & 3 & 5 \end{pmatrix},$$

$$\Phi_3 = \varepsilon = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix},$$

то легко можно убедиться в справедливости соотношений $g_1 = \Phi_1(g_2)$; $g_1 = \Phi_2(g_3)$; $g_1 = \Phi_3(g_4)$. Для доказательства, что $G_1 \simeq G_5$, потребуется перебрать все перестановки 6-й степени, т. е. к одной из квадратичных форм g_1 или g_5 последовательно применять все $6! = 720$ перестановок и каждый раз сравнивать результат со второй квадратичной формой.

В общем случае для определения изоморфизма двух p -вершинных графов необходимо просмотреть все $p!$ перестановок p -й степени. Неэффективность такой процедуры очевидна даже при не слишком больших значениях p : уже при $p = 20$ перебор всех $20!$ вариантов потребовал бы нескольких лет у современных ЭВМ. Поэтому, естественно, попытаться найти такие относительно легко вычисляемые характеристики графа (число, системы чисел, векторы, многочлены, матрицы и т. д.), по которым определялся бы граф однозначно с точностью до изоморфизма.

5.2. ИНВАРИАНТЫ

Определим понятие инварианта графа, следуя [1].

Определение. Пусть F — функция, относящаяся к каждому графу G ; $F(G)$ — некоторый элемент из множества \mathfrak{M} произвольной природы (элементами множества \mathfrak{M} чаще всего могут служить числа, системы чисел, векторы, многочлены, матрицы и т. д.). Эту функцию называют инвариантом, если на изоморфных графах ее значения совпадают, т. е. для любых графов G и H

$$[G \simeq H] \Rightarrow [F(G) = F(H)].$$

Примерами тривиальных инвариантов графа G могут служить число вершин $p(G)$, число ребер $q(G)$, число ребер и вектор степеней $s(G) = (s_1, s_2, \dots, s_i, \dots, s_p)$, который, в частности, дает числовые инварианты $s(G) = s_1 = \min_{x \in X} s(G, x)$ и $\bar{s}(G) = s_n = \max_{x \in X} s(G, x)$ (степень графа).

Наиболее известными инвариантами графа G являются: **плотность графа** $\phi(G)$ — число вершин наибольшего полного подграфа (наибольшей клики) в G , т. е. наибольшее количество попарно смежных вершин; $\chi(G)$;

плотность графа $\varepsilon(G)$ — наибольшее количество его попарно несмежных вершин, т. е. число вершин наибольшего безреберного подграфа графа G ;

число компонент $k(G)$ — число всех максимальных связанных подграфов графа G ;

хроматическое число $\gamma(G)$ — наименьшее число, при котором граф G допускает такую раскраску своих вершин, что каждая его вершина окрашена в один из $\gamma(G)$ цветов, причем все $\gamma(G)$ цвета использованы и никакие две смежные вершины не окрашены в один и тот же цвет;

число Харвицера $\eta(G)$ — такое наибольшее целое число, что граф G стягиваем к полному $\eta(G)$ — вершинному графу K_η .

В качестве инвариантов графа можно рассмотреть не одно число, а систему чисел, в частности вектор, например вектор $(p, g, \varphi, \varepsilon, k, \gamma, \eta)$.

Следуя [1], рассмотрим несколько инвариантов графа G , заданных в виде многочлена

$$P = P(x) = \sum_{i \geq 0} p_i x^i = p_0 + p_1 x + p_2 x^2 + \dots$$

формальной переменной x , где «суммирование» ведется до последнего отличного от нуля слагаемого:

$$F(G) = \sum_{i \geq 0} f_i(G) x^i = f_0(G) + f_1(G) x + f_2(G) x^2 + \dots + f(G) x^p,$$

где $\varphi = \varphi(G)$, $f(G)$ — количество полных i -вершинных подграфов (т. е. i -клик) графа G ; $f_1(G) = p(G)$; $f_2(G) = g(G)$; $\max\{i/f_i(G) \neq 0\} = \varphi(f_0(G))$ полагаем равным 1, так как под 0-вершинным подграфом понимается пустое множество;

$$E(G) = \sum_{i \geq 0} e_i(G) x^i = e_0(G) + e_1(G) x + e_2(G) x^2 + \dots + e_s(G) x^s,$$

где $\varepsilon = \varepsilon(G)$; $e_i(G)$ — количество безреберных i -вершинных подграфов графа G ; $e(G) = 1$; $e_1(G) = p(G)$; $e_2(G) = \binom{p(G)}{2} - g(G)$; $\max\{i/e_i(G) = 0\} = \varepsilon$;

$$\Gamma(G) = \sum_{i \geq 1} g_i(G) x^i = g(G) x^\gamma + g_{\gamma+1}(G) x^{\gamma+1} + \dots + g_p(G) x^p,$$

где $p = p(G)$; $\gamma = \gamma(G)$; $g_i(G)$ — количество i -раскрасок графа G , а далее $g_1(G) = g_2(G) = \dots = g_{\gamma-1}(G) = 0$; $g_p(G) = 1$,

$$g_{p+1}(G) = g_{p+2}(G) = \dots = 0, \quad \min\{i/g_i(G) \neq 0\} = \gamma;$$

$$H(G) = \sum_{i \geq 1} h_i(G) x^i = h_1(G) x + h_2(G) x^2 + \dots + h_\eta(G) x^\eta,$$

где $\eta = \eta(G)$; $h_i(G)$ — количество различных стягиваний связного графа G на i -клик, $h_1(G) = 1$, $\max\{i/h_i(G) \neq 0\} = \eta$. Аналогично можно записать в виде многочленов от нескольких формальных переменных x, y систему инвариантов графа, зависящих от двух или более параметров. Например, $A(G) = \sum_{i,j \geq 0} a_{i,j}(G) x^i y^j$, где $a_{i,j}(G)$ — количество тех подграфов графа G , которые имеют i вершину и j ребер; $a_{0,0}(G) = 1$, $a_{0,j}(G) = 0$ при $j > 0$. Конкретные примеры, поясняющие эти инварианты, а также способы их вычисления можно найти в монографии [1].

Рассмотрим числовые инварианты, связанные с матричным заданием графа (например, с матрицей смежностей графа с заданной нумерацией вершин $1, 2, \dots, p$). Пусть квадратичная матрица p -го порядка $A(G) = \{a_{i,j}\}$ есть матрица смежностей графа G с заданной нумерацией вершин. Очевидно, сама матрица смежностей не является инвариантом графа, ибо при переходе от одной нумерации его вершин к другой она претерпевает перестановку строк и соответствующих столбцов. Инвариантом графа G является любая функция от элементов, $a_{i,j}$, не изменяющаяся ни при какой перенумерации вершин графа G . К числу инвариантов, связанных с матрицей смежностей графа G , относятся

$$\sum_{i,j=1}^p a_{i,j}; \quad \sum_{i=1}^p a_{i,i}; \quad \sum_{i=1}^p a_{i,i}^2; \quad \det A = |a_{i,j}|,$$

а также характеристический многочлен матрицы A и корни этого многочлена.

Совокупность корней характеристического многочлена, называемая спектром графа, есть инвариант, который не определяет граф с точностью до изоморфизма [2].

Согласно определению [1] неизоморфные графы с одинаковым спектром называются коспектральными. В качестве примера пары коспектральных графов можно привести графы G и H , изображенные на рис. 5.2.1.

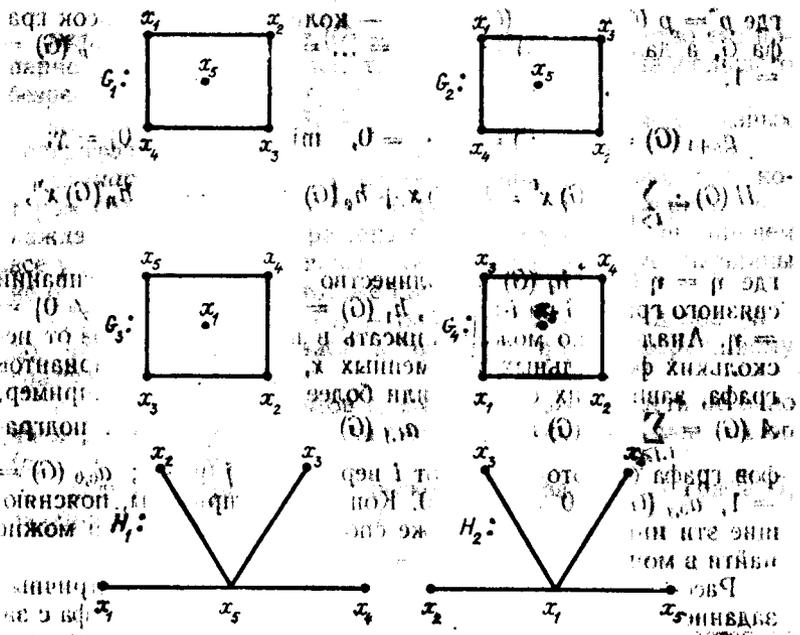


Рис. 5.2.1

Действительно, матрицы смежностей графов G и H имеют вид

$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad A(H) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Соответствующие графам G и H характеристические многочлены совпадают и имеют вид

$$A(G) - \lambda E = \begin{pmatrix} -\lambda & 1 & 0 & 1 & 0 \\ 1 & -\lambda & 1 & 0 & 0 \\ 0 & 1 & -\lambda & 1 & 0 \\ 1 & 0 & 1 & -\lambda & 0 \\ 0 & 0 & 0 & 0 & -\lambda \end{pmatrix} = -\lambda^3(\lambda^2 - 4)$$

$$A(H) - \lambda E = \begin{pmatrix} 0 & 0 & 0 & 0 & -\lambda \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = -\lambda^3(\lambda^2 - 4)$$

Следовательно, спектром графов G и H является множество $\{-2; 0, 2\}$.

Рассмотрим инварианты, по которым и количеству вершин графа легко восстанавливается одна из матриц смежностей графа, а значит и сам граф с точностью до изоморфизма. С этой целью совокупность всевозможных $p!$ матриц смежностей p -вершинного графа G обозначим через $A^{(k)}(G) = \|\alpha_{i,j}^{(k)}\|$, $k = \overline{1, p!}$

Запишем так называемые двоичные коды матриц $A^{(k)}(G)$ в виде

$$\mu^{(k)}(G) = \alpha_{1,2}^{(k)}2^0 + \alpha_{1,3}^{(k)}2^1 + \alpha_{2,3}^{(k)}2^2 + \alpha_{1,4}^{(k)}2^3 + \alpha_{2,4}^{(k)}2^4 + \dots + \alpha_{p-1,p}^{(k)}2^{p-1} \quad (k = \overline{1, p!})$$

Определяем

$$\mu(G) = \min_{k=1, p!} \{\mu^{(k)}(G)\} \text{ (мини-код)}, \quad \bar{\mu}(G) = \max_{k=1, p!} \{\mu^{(k)}(G)\} \text{ (максимум-код)},$$

т. е. для графов G и H , изображенных на рис. 5.2.1, имеем

$$\mu^{(1)}(G_1) = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 0 \cdot 2^8 + 0 \cdot 2^9 = 1 + 4 + 8 + 32 = 45,$$

$$\mu(G_2) = \mu^{(2)}(G_2) = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 0 \cdot 2^8 + 0 \cdot 2^9 = 2 + 4 + 8 + 16 = 30,$$

$$\bar{\mu}(G_2) = \mu^{(3)}(G_2) = 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 1 \cdot 2^8 + 1 \cdot 2^9 = 4 + 16 + 256 + 512 = 788,$$

$$\mu(H_1) = \mu^{(1)}(H_1) = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 1 \cdot 2^7 + 1 \cdot 2^8 + 1 \cdot 2^9 = 64 + 128 + 256 + 512 = 960,$$

$$\begin{aligned} \mu(H_2) &= \mu^{(2)}(H_2) = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + \\ &+ 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7 + 0 \cdot 2^8 + 0 \cdot 2^9 = \\ &= 1 + 2 + 8 + 64 = 75. \end{aligned}$$

Наоборот, если о графе известно, что он имеет четыре вершины, а значение некоторого его двоичного кода равно 51, то восстанавливая по этому значению двоичный код (до разряда $2^{\binom{p}{2}-1} = 2^{\binom{4}{2}-1} = 2^5$), получаем $51 = 1 + 2 \cdot 25 = 1 + 2 \times$

откуда находим матрицу

$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

и соответствующий граф G_4 (рис. 5.2.2).

В случае 5-вершинного графа с тем же кодом 51 необходимо этот код представить в виде

$$51 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 0 \cdot 2^8 + 0 \cdot 2^9$$

(так как $\binom{5}{2} - 1 = 9$), что дает матрицу

$$A(H) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

и соответствующий граф G_4 (см. рис. 5.2.1).

Очевидно, количество вершин можно не сообщать, если заранее известно, что у графа нет изолированных вершин.

Далее, для обыкновенного помеченного p -вершинного графа G рассмотрим неупорядоченную совокупность векторов $(p-1)$ -го порядка

$$R^{(k)}(G) = \{\bar{s}_i^{(k)}(G)\} = \{(s_{i,1}^{(k)}, s_{i,2}^{(k)}, \dots, s_{i,p-1}^{(k)})\},$$

$$i = \overline{1, p}; \quad j = \overline{1, p-1}; \quad 1 \leq k \leq p!$$

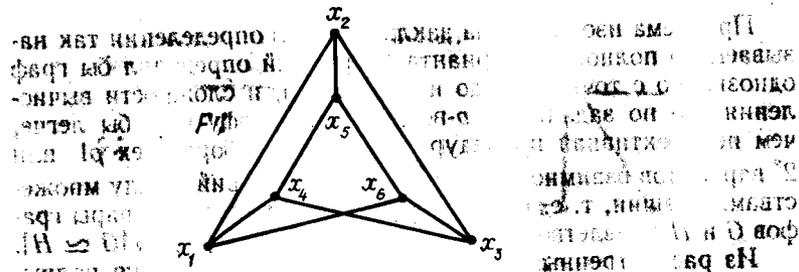
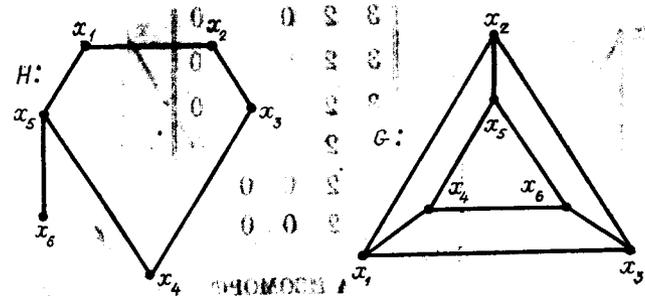


Рис. 5.2.3

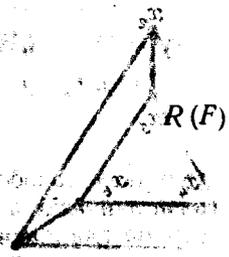
где $s_{i,j}^{(k)}$ — число различных вершин графа G , минимальное расстояние до которых от i -й вершины равно j , а k — порядок нумерации вершин графа (при всевозможных $p!$ нумерациях) ($s_{i,1}^{(k)}$ — совпадает со степенью i -й вершины ($i = \overline{1, p}$)).

Очевидно, матрица минимальных расстояний $\|\bar{s}_i^{(k)}(G)\|$ — не инвариант, ибо при переходе от k -й нумерации его вершин к l -й нумерации она претерпевает перестановку строк. Если же на множество $R^{(k)}(G)$ не накладывать ограничений на упорядоченность его элементов (векторов) $\bar{s}_i^{(k)}$, то ясно, что

$$\forall i \forall k \{R^{(k)}(G) = R^{(l)}(G) = R(G)\}.$$

В качестве примера рассмотрим графы, изображенные на рис. 5.2.3. Матрицы минимальных расстояний соответствующие графам G , H и F имеют вид

$$R(G) = \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \end{pmatrix}, \quad R(H) = \begin{pmatrix} 2 & 3 & 0 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \end{pmatrix}$$



$$R(F) = \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \end{pmatrix}$$


5.3. ПРОБЛЕМА ИЗОМОРФИЗМА $Z = 1 + 2 + 3$

Проблема изоморфизма заключается в определении так называемого полного инварианта f , который определял бы граф однозначно с точностью до изоморфизма, и сложности вычисления его по заданному p -вершинному графу был бы легче, чем неэффективная процедура полного перебора всех $p!$ или 2^p вариантов взаимно однозначных соответствий между множествами вершин, т. е. полный инвариант f для любой пары графов G и H удовлетворит отношению $[f(G) = f(H)] \Leftrightarrow [G \simeq H]$.

Из рассмотренных выше инвариантов не являются полными ни вектор степеней $\bar{s}(G)$, ни матрица минимальных расстояний $R = \|s_{i,j}\|$. В качестве конкретного примера могут служить графы G и F на рис. 5.2.3, так как $\bar{s}(G) = \bar{s}(F) = (3, 3, 3, 3, 3)$,

$$R(F) = \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \end{pmatrix}$$

Очевидность неполноты многочленного инварианта $F(G)$ обнаруживается на графах G и H , а инварианта $E(G)$ — на графах \bar{G} и \bar{H} , дополнительных к ним (рис. 5.3.1). Действительно, $F(G) = 1 + 5x + 6x^2 + 2x^3 = F(H)$, $E(\bar{G}) = 1 + 5x + 4x^2 = E(\bar{H})$.

Очевидно также, что графы G и H на рис. 5.3.2 выявляют неполноту инварианта $\Gamma(G)$.

Что касается полноты многочленного инварианта от двух формальных переменных (x, y) $A(G)$, то и она была опровергнута впервые А. А. Зыковым примерами графов. На рис. 5.3.3 представлены графы, построенные И. М. Горгосом, выявляющие неполноту инварианта $A(G)$. Этот пример опровергает

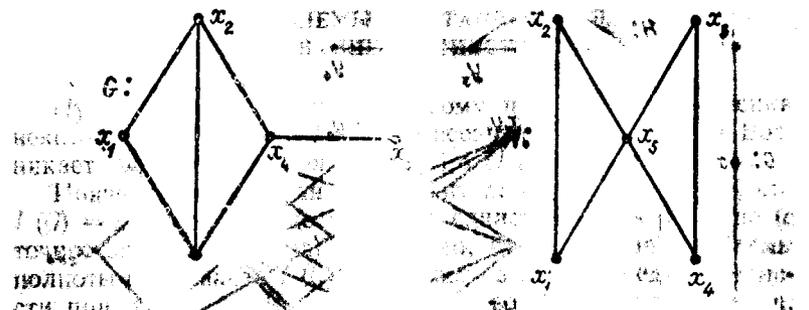


Рис. 5.3.1

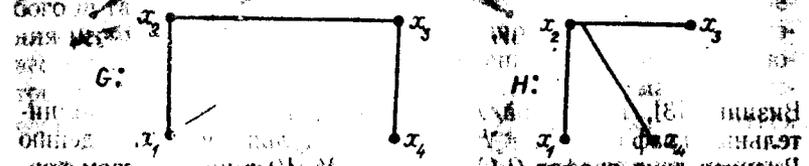


Рис. 5.3.2

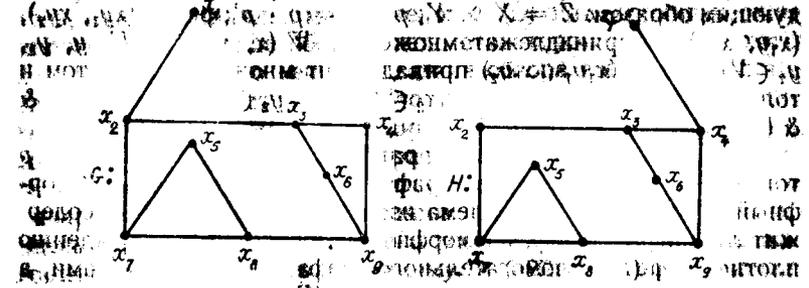


Рис. 5.3.3

полноту и ряда других многочленных инвариантов, рассмотренных в [1].

В предыдущем параграфе мы убедились в полноте инвариантов максикода $\mu(G)$ и миникода $\bar{\mu}(G)$ графа G с фиксированным числом вершин p . Хотя $[\mu(G) = \mu(H)] \Leftrightarrow [G \simeq H]$ и $[\bar{\mu}(G) = \bar{\mu}(H)] \Leftrightarrow [G \simeq H]$, к сожалению, эти инварианты проблему изоморфизма не решают, ибо процессы вычисления самих инвариантов μ и $\bar{\mu}$ для заданного p -вершинного графа так же неэффективны, как и полный $p!$ вариантов двух графов, так как надо определять наибольший и соответственно наименьший коды всех $p!$ матриц смежностей графа.

Поиск другого пути решения проблемы изоморфизма связан с дополнительным графом к оригинальной конструкции

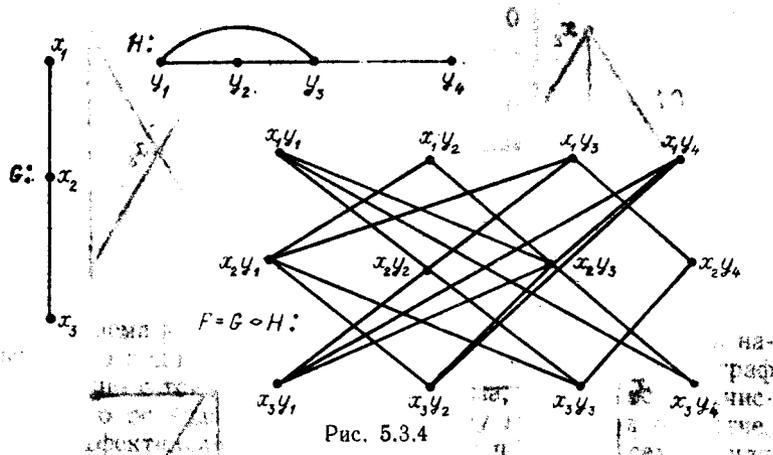


Рис. 5.3.4

Визинга [3], названной *модульным произведением*. Дополнительный граф $G \diamond H = F(Z, W)$ к модульному произведению Визинга двух графов $G(X, U)$ и $H(Y, V)$ с количеством вершин $p = p(G) = |X| \leq p' = p'(H) = |Y|$ строится, следующим образом: $Z = X \times Y$, $p(Z) = p \cdot p'$; ребра (x_1y_1, x_2y_2) , (x_1y_2, x_2y_1) не принадлежат множеству W ($x_1, x_2 \in U$; $y_1, y_2 \in V$); ребро (x_1y_1, x_2y_2) принадлежит множеству W в том и только том случае, если $(x_1x_2 \in U) \& (y_1y_2 \in V) \vee (x_1x_2 \notin U) \& (y_1y_2 \notin V)$ (рис. 5.3.4).

Для графа $F = G \diamond H$ справедливо равенство $\varphi(F) = p$ тогда и только тогда, когда граф H содержит подграф, изоморфный графу G , т. е. проблема изоморфного вхождения (содержит ли граф подграф, изоморфный G) сводится к нахождению плотности $\varphi(F)$ вспомогательного графа F с pp' вершинами, а точнее к определению соотношения $\varphi(F) \leq p$.

В частности, если для графов G и H $p(G) = p(H)$, то задача изоморфизма сводится к нахождению $\varphi(F)$ p^2 -вершинного графа F . Эта конструкция не решает проблему изоморфизма полностью, а лишь сводит ее к другой проблеме, не менее сложной. Такой подход в сочетании с другими идеями приводит на практике к эффективному способу идентификации графов. Следует отметить, что конструкция Визинга содержит в себе значительно больше, чем простое сведение проблемы изоморфного вхождения к задаче нахождения плотности. В частности, из нее следует, что все n -клики графа $F = G \diamond H$ взаимно однозначно соответствуют всевозможным изоморфным вхождениям G в качестве подграфа в H . Так, на рис. 5.3.4. граф F содержит четыре подграфа типа 3-вершинного полного графа, в то же время граф H содержит четыре подграфа изоморфных графу G .

5.4. ПРОБЛЕМЫ ВОССТАНОВЛЕНИЯ. СУЩЕСТВОВАНИЕ И ЕДИНСТВЕННОСТЬ

Пусть \mathfrak{M} — множество, которому принадлежат значения некоторого инварианта $I(G)$ при всевозможных графах G . Возникает вопрос: $\exists G \forall m \{(m \in \mathfrak{M}) \& (I(G) = m)\}$?

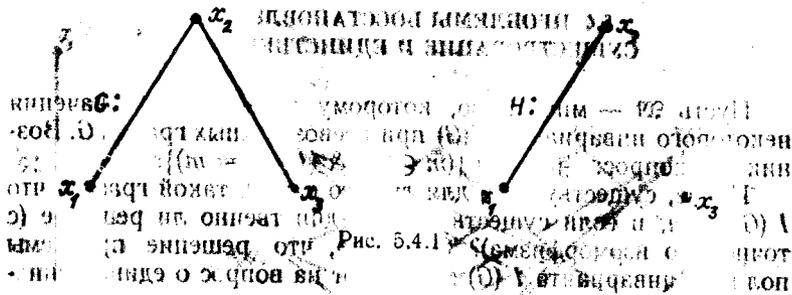
Иначе, существует ли для любого $m \in \mathfrak{M}$ такой граф G , что $I(G) = m$, и если существует, то единственно ли решение (с точностью изоморфизма)? Очевидно, что решение проблемы полноты инварианта $I(G)$ дает ответ на вопрос о единственности при любых m . Что же касается вопроса существования, т. е. определения эффективного алгоритма построения для любого m такого графа G , что $I(G) = m$, то до полного его решения необходимы интенсивные исследования почти для всех известных инвариантов. В этом направлении даже для часто употребляемых инвариантов, как F , E и Γ , пока мало что известно. Например, многочлен $P = \sum_{i \geq 1} s_i x^i$ может равняться

$F(G)$ для некоторого графа G лишь при соблюдении некоторых условий. Необходимы также дополнительные условия для того, чтобы P являлся многочленом $\Gamma(G)$ или хроматическим многочленом. Лучшее обстоит дело с вектором степеней, хотя и здесь нет полного решения задачи обзора всех графов с заданным вектором степеней. Что касается полноты инварианта, связанного с матрицей минимальных расстояний $R(G)$, то примеры графов G и F , изображенные на рис. 5.2.3, опровергают вообще существование алгоритма построения для любого инварианта $m \in \mathfrak{M}$ единственного графа G , что $R(G) = m$. Здесь речь может идти только о построении эффективного алгоритма построения по заданному $m \in \mathfrak{M}$ некоторого класса графов $\{G_i\}$, что $R(G_i) = m$.

Определение 1. Проблемы существования и единственности искомого графа G с заданным (искомым) значением некоторого инварианта $I(G) \in \mathfrak{M}$, где \mathfrak{M} — множество (система) графов, называют *проблемой восстановления графа G по инварианту $I(G)$* .

Классическим примером проблемы восстановления служит следующая проблема, известная как гипотеза С. М. Улама [4], подробное описание некоторого подхода к решению которой дано в последующих параграфах. Здесь сформулируем только ряд проблем восстановления, начиная с гипотезы Улама.

Пусть $\{G \setminus x_i\}$ ($i = \overline{1, p}$) — набор всех $(p-1)$ -вершинных подграфов некоторого p -вершинного графа $G(X, V)$, каждый из которых задан независимо от остальных с точностью до изоморфизма. Определяется ли по набору $\{G \setminus x_i\}$ исходный граф



G однозначно (с точностью до изоморфизма)? Очевидно, при $p = 2$ гипотеза неверна, а для $3 \leq p \leq 9$ положительный ответ получен непосредственной проверкой. Ясна также восстанавливаемость для графов несвязных или с несвязным дополнением, ибо удаление одной вершины делает неопределенной лишь одну из компонент, т. е. среди компонент всех подграфов G/x_i встречаются и все компоненты исходного графа G , достаточно отсортировать их от неопределенных компонент. Хотя случайный конечный граф восстанавливается с вероятностью единица [5], для бесконечных графов гипотеза Улама не справедлива [6]. Ф. Харари выдвинул более «сильную» гипотезу: при $p \geq 4$ для восстановления p -вершинного графа достаточно знать не все подграфы $G \setminus x_i$, а только неизоморфные. В том, что при $p = 3$ это неверно, видно на примере графов G и H на рис. 5.4.1.

В пользу справедливости гипотезы Улама говорят и другие известные инварианты графа G , определяемые набором $\{G \setminus x_i\}$ [7, 8]. Например, вектор степеней $\bar{s}(G)$ определяется по набору всех $(p-2)$ -вершинных подграфов $G/(xy)$ ($x = y$) [9]. Проблема Улама естественно обобщается на случай. Определяется ли однозначно исходный p -вершинный граф G с точностью

до изоморфизма по набору $\{G_1, G_2, \dots, G_m\}$ ($m = \binom{p}{k}$) всевозможных $(p-k)$ -вершинных подграфов графа G , где $1 \leq k \leq p-1$. При $k = p-1$ этот набор несет информацию только о количестве вершин графа G , а при $k = p-2$ — только о количестве вершин и ребер. Так что восстанавливаемости в этом случае нет.

Несомненный интерес вызывают и следующие вопросы, связанные с положительным решением гипотезы Улама.

1. Для произвольного набора $\{G_i\}$ ($i = \overline{1, p}$) $(p-1)$ -вершинных графов существует ли хоть один такой p -вершинный граф $G(X, V)$ ($X = \{x_i\}; i = \overline{1, p}$), что $G \setminus x_i \simeq G_i$?

2. При каком наибольшем значении k восстанавливаемость имеет место для всех графов?

3. Каков минимальный набор $\{G_i\}$ ($i = \overline{1, m}$) $(p-k)$ -вершинных графов, определяющих однозначно с точностью до изоморфизма исходный p -вершинный граф G ?

Частичные ответы на эти вопросы будут даны нами в последующих параграфах.

Можно выдвинуть еще одну проблему, очевидно, эквивалентную гипотезе Улама.

Пусть $\{G_i(X, V_i)\}$ ($i = \overline{1, p}$) — набор всех p -вершинных надграфов p -вершинного графа $G(X, V)$ таких, что $\forall j \{(x_i, x_j) \in V_i\}$ и каждый из которых задан независимо от остальных с точностью до изоморфизма. Определяется ли по набору $\{G_i\}$ исходный граф G однозначно с точностью до изоморфизма? Если первоначальную гипотезу Улама характеризовать как вершинную, то естественно сформулировать и аналогичную реберную: восстанавливается ли однозначно с точностью до изоморфизма граф $G(X, \{v_1, v_2, \dots, v_q\})$ по набору $\{G/v_i\}$ всех своих $(q-1)$ -реберных суграфов? Очевидно, для конечных графов из справедливости вершинной гипотезы Улама следует справедливость реберной гипотезы [10].

Рассмотрим еще одну проблему восстановления графов, а именно проблему окружений. Пусть G — произвольный граф; существует ли такой граф $L(X, V)$, что $O(L, x) \simeq G$ для любой вершины $x \in X$, где $O(L, x)$ — подграф, порожденный всеми вершинами, смежными с вершиной x ? В работе [11] показано, что для класса всех обыкновенных графов проблема окружений алгоритмически неразрешима.

5.5. ГИПОТЕЗА УЛАМА

Пусть G и H — группы порядка p ; $G_i^k \subset G$ и $H_j^k \subset H$ — i -е и j -е подмножества из k элементов соответственно множеств G и H ($1 < k < p; 1 \leq i, j \leq \binom{p}{k}$).

Определение 1. Два подмножества G_i^k и H_j^k условно изоморфны, если существует взаимно однозначное отображение φ множества G_i^k на H_j^k такое, что если элементы a, b и $c = ab$ принадлежат G_i^k , то $\varphi(c) = \varphi(a)\varphi(b)$.

Каково минимальное число $m(p, k)$, обладающее тем свойством, что из условного изоморфизма каждого G_i^k некоторому H_j^k следует изоморфизм G и H ? В силу того, что каждая конечная группа изоморфна некоторой подгруппе группы перестановок S_n , наша задача может быть сформулирована на языке групп перестановок следующим образом.

Пусть $G = \{\alpha_1, \alpha_2, \dots, \alpha_p\}$ и $H = \{\beta_1, \beta_2, \dots, \beta_p\}$ — группы перестановок порядка p степени n , т. е. действующие на

множестве объектов $N = \{1, 2, \dots, n\}$; $G_i^k = \{\alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_k}\}$ и $H_j^k = \{\beta_{j_1}, \beta_{j_2}, \dots, \beta_{j_k}\}$ — i -я и j -я подгруппы порядка k соответственно групп G и H , где $1 < k < p$; $1 \leq i \leq \binom{p}{k}$.

Определение 2. Две подгруппы G_i^k и H_j^k условно изоморфны ($G_i^k \simeq H_j^k$), если существует взаимно однозначное отображение φ подгруппы G_i^k на H_j^k такое, что если a, b и $c = ab$ принадлежат G_i^k , то $\varphi(a), \varphi(b)$ и $\varphi(c) = \varphi(a)\varphi(b)$ принадлежат H_j^k . Каково минимальное число $m(p, k)$, обладающее тем свойством, что $\forall i \exists j \exists \varphi [G_i^k \simeq H_j^k] = \Rightarrow \exists \psi [G \simeq H]$?

Пусть $T_n = \{(i, j)\}$ — множество всех транспозиций из S_n , т. е. перестановок, оставляющих на месте все элементы множества $N = \{1, \dots, i, \dots, j, \dots, n\}$, кроме чисел i и j , которые переставляются. Поскольку любую перестановку можно разложить в произведение транспозиций вида (i, j) , то любую группу $G_k = \{\alpha_1, \dots, \alpha_k\}$ перестановок можно представить некоторым множеством транспозиций $T_{G_k} = \{(i_1, j_1), \dots, (i_l, j_l)\} \subset T_n$, которое будет системой образующих данной группы G_k .

По множеству T_{G_k} строится неориентированный граф P , вершины которого обозначены символами v_1, v_2, \dots, v_n , причем вершины v_i и v_j смежны тогда и только тогда, когда транспозиция (i, j) принадлежит множеству T_{G_k} . Например, множеству транспозиций $(1, 2), (2, 3), (2, 5), (3, 5), (4, 5)$ соответствует граф P_1 , множеству $(1, 2), (1, 3), (2, 3), (4, 5)$ — граф P_2 , а множеству $(1, 2), (2, 4), (2, 5), (1, 3), (3, 6), (3, 7), (6, 8), (6, 9)$ — граф P_3 (рис. 5.5.1).

Граф P , построенный по множеству транспозиций T_G некоторой группы перестановок G , называют *графом Пойа* этого множества [12, 13].

Очевидно, в силу известных фактов [12, 13]:

а) множество транспозиций T будет системой образующих симметрической группы S_n , тогда и только тогда, когда граф Пойа этого множества связан;

б) система образующих симметрической группы будет неприводимой тогда и только тогда, когда граф Пойа этой системы является деревом;

с) для произвольной последовательности l_0, l_1, \dots, l_k различных натуральных чисел, таких, что $l_0 = i, l_k = j$ имеет место следующее разложение транспозиции (i, j) :

$$(i, j) = (i, l_1) \cdot (l_1, l_2) \cdot \dots \cdot (l_{k-1}, j) \cdot (l_k, l_{k-2}) \cdot \dots \cdot (l_1, i);$$

д) если граф Пойа множества транспозиций T деревом не является, то в последовательности $l_0, l_1, \dots, l_k, l_{k+1} = l_0$ ин-

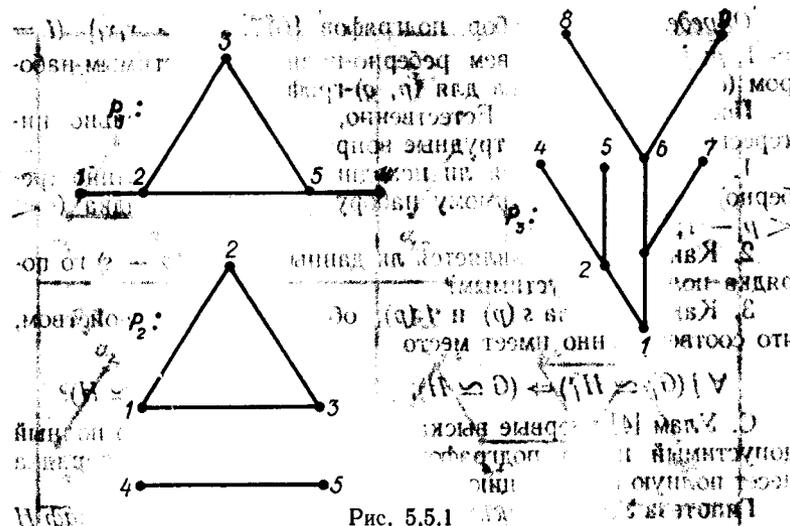


Рис. 5.5.1

дексов вершин, образующих замкнутый путь, транспозицию (l_0, l_k) убрать из множества T , так как эта транспозиция раскладывается в произведение остальных транспозиций этой последовательности.

Теперь задачу можно сформулировать на языке деревьев Пойа или на языке обыкновенных графов.

Допустимые наборы графов. Пусть $G = (p, q)$ — граф с помеченными p вершинами x_1, x_2, \dots, x_p и q ребрами. Удаление вершины x из графа G приводит к подграфу $G_j^{p-1} \triangleq G - x_j$, содержащему все вершины графа G , за исключением вершины x_j , и все ребра графа G , не инцидентные вершине x_j . Иначе G_j^{p-1} есть максимальный подграф графа G , не содержащий x_j .

Определение 3. Набор подграфов $\{G_j^{p-1} \triangleq G - x_j\}$ ($j = \overline{1, p}$) назовем вершинно-полным допустимым набором $(p-1)$ -го порядка для (p, q) -графа G .

Определение 4. Набор подграфов $\{G_j^{p-s} \triangleq ((\dots((G - x_{j_1}) - x_{j_2}) \dots) - x_{j_s})\}$ ($j = \overline{1, m}$) назовем вершинно-полным допустимым набором $(p-s)$ -го порядка для (p, q) -графа G , где m равно числу сочетаний из p по $(p-s)$, т. е. $m = \binom{p}{p-s}$.

Удаление ребра $x_i x_j$ графа G приводит к подграфу $G_{i,j}^{p-1} \triangleq G - x_i x_j$, содержащему все вершины и ребра графа G , за исключением ребра $x_i x_j$, т. е. $G_{i,j}^{p-1}$ есть частичный подграф графа G , не содержащий ребра $x_i x_j$.

Определение 5. Набор подграфов $\{G_{i,j}^{p-1} \doteq G \pm x_i x_j\}$ ($i = \overline{1, p}; j = \overline{1, p}$) назовем реберно-полным допустимым набором ($q - t$)-го порядка для (p, q) -графа G .

Постановка задачи. Естественно, возникают довольно интересные и достаточно трудные вопросы.

1. Восстанавливается ли исходный граф по вершинно (реберно)-полному допустимому набору $(p - s)$ -го порядка ($s < p - 1; p \geq 3$)?

2. Как выяснить, является ли данный набор $(p - s)$ -го порядка полным допустимым?

3. Каковы числа $s(p)$ и $t(p)$, обладающие тем свойством, что соответственно имеет место

$$\forall j (G_j^{s_1} \simeq H_j^{t_1}) \Rightarrow (G \simeq H); \quad \forall l (G_l^{s_2} \simeq H_l^{t_2}) \Rightarrow (G \simeq H)?$$

С. Улам [4] впервые высказал предположение, что полный допустимый набор подграфов $\{G \pm x_j\}$ $(p - 1)$ -го порядка несет полную информацию о всем графе G .

Гипотеза Улама. Пусть граф G имеет p вершин x_i , граф H имеет p вершин y_i и $p \geq 3$. Если для каждого j подграфы $G_j^{p-1} \doteq G \pm x_j$ и $H_j^{p-1} \doteq H \pm y_j$ изоморфны, то и графы G и H изоморфны, т. е.

$$\forall j [(G \pm x_j) \simeq (H \pm y_j)] \Rightarrow (G \simeq H).$$

Иными словами [14], в гипотезе С. Улама утверждается, что любые два графа с одним и тем же вершинно-полным допустимым набором $(p - 1)$ -го порядка изоморфны. Ниже даются ответы на первые два вопроса для случая $s = 1$.

5.6. АЛГОРИТМ ВОССТАНОВЛЕНИЯ ГРАФОВ ПО ДОПУСТИМОМУ НАБОРУ

Пусть $\Gamma^{p-1} = \{G_j^{p-1}\}$ ($i = \overline{1, p}; p \geq 3$) — вершинно-полный допустимый набор $(p - 1)$ -го порядка для некоторого p -вершинного графа G (рис. 5.6.1). Положим, что $G_j^{p-1} \doteq G \pm x_j$ ($j = \overline{1, p}$), где x_j — условные метки вершин искомого графа G , а $g(x_0, x_1, \dots, x_p) = \sum_{k,l=0}^p \beta_{k,l} x_k x_l$, $g_j(a_0, a_1, j, \dots$

$\dots, a_{p-1,j}) = \sum_{k,l=0}^{p-1} \alpha_{k,l} a_{k,j} a_{l,j}$ — соответствующие квадратичные формы графов G и G_j^{p-1} ($j = \overline{1, p}$), вершины которых предварительно помечены символами $a_{i,j}$ ($i, j = \overline{1, p}$).

Требуется восстановить граф G , т. е. необходимо определить коэффициенты $\beta_{i,j}$ ($i = \overline{1, p}; j = \overline{1, p}$) квадратичной фор-

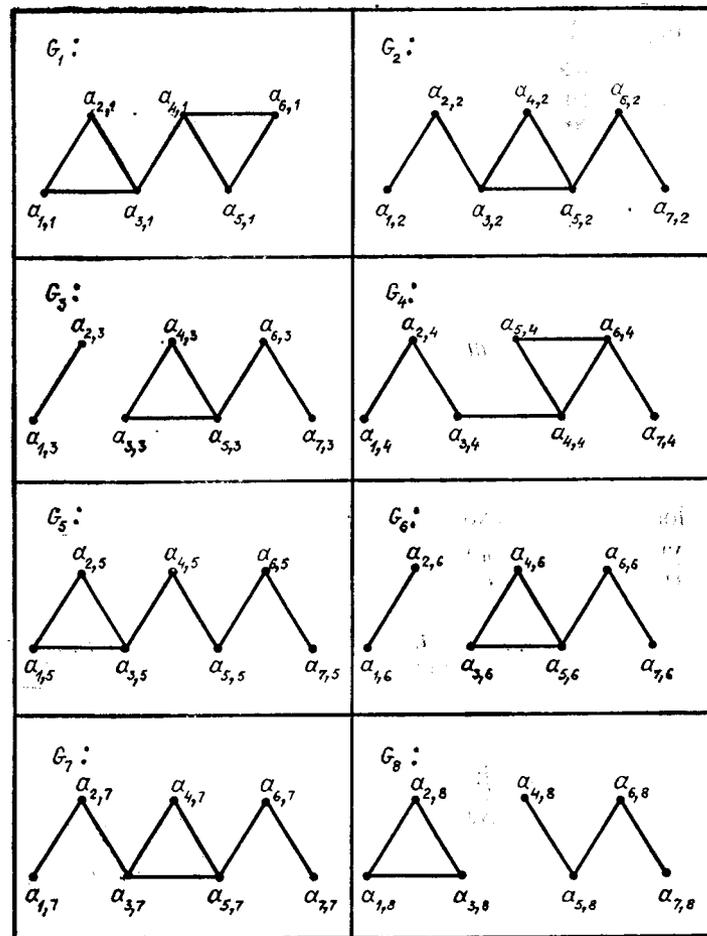


Рис. 5.6.1

мы $g(x_0, x_1, x_2, \dots, x_i, \dots, x_p)$ искомого графа G таким образом, чтобы $\forall j [\exists \omega [\omega(g_i) = S_0^j [g]]]$, где ω — некоторое отображение множества $\{a_{i,j}\}$ на множество $\{x_i\}$, а S -символ обозначает оператор подстановки [15]. С этой целью сначала выразим каждую неизвестную (метку вершины) x_i искомого графа G через метки вершин графов G_j ($i \neq j$), которые потенциально соответствуют вершине графа G с меткой x_i , т. е.

$$(x_i = a_{i_1, j}) \vee \dots \vee (x_i = a_{i_{\alpha}, j}) \vee \dots \vee (x_i = a_{i_s, j}),$$

$$(1 \leq i \leq p; j = \overline{1, p}; 1 \leq \alpha \leq s \leq p - 1),$$

где $\{a_{i,\alpha,j}\} = W_{i,j,k}$ ($1 \leq i \leq p$; $j = \overline{1, p}$; $1 \leq k \leq p$; $1 \leq \alpha \leq s - k$ -е подмножество множества $\{a_{i,j}, a_{i,j}, \dots, a_{i,p-1,j}\}$). Затем, используя эти соотношения, метки $a_{i,j}$ на графах G_j заменяем соответствующими метками x_i . Наконец, по вновь помеченным графам G_j определяем значения коэффициентов $\beta_{i,j}$.

Матрица сопряженностей. Очевидно, в силу свойства операций подстановки [15] справедливо

$$\forall i \forall j \{S_0^i [S_0^j [g]] = S_0^i [S_0^j [g]]\} \quad (i, j = \overline{1, p}).$$

Откуда непосредственно следует

$$\forall i \forall j \exists m \exists n \{S_{a_0}^{n,i} [g_i] \simeq S_{a_0}^{m,j} [g_j]\}, \quad (5.6.1)$$

где $i \neq j$; $1 \leq m, n \leq p - 1$.

Определение 1. Вершины $a_{n,i}$ и $a_{m,j}$, удовлетворяющие условию (5.6.1), назовем *взаимно сопряженными вершинами* графов G_i^{p-1} и G_j^{p-1} и обозначим $a_{n,i} \overset{\circ}{\sim} a_{m,j}$.

Построим для исходного допустимого набора Γ^{p-1} квадратичную матрицу $W_0 = \{W_{i,j}\}$ ($i, j = \overline{1, p}$) (табл. 5.6.1), элементы которой $W_{i,j} = \{W_{i,j,1}, \dots, W_{i,j,k}, \dots, W_{i,j,L_{i,j}}\}$ образованы из попарно непересекающихся упорядоченных классов $W_{i,j,k}$ вида $W_{i,j,k} = \{a_{i,\alpha,j}\}$ ($1 \leq \alpha \leq s_{i,j}^k \leq p - 1$), удовлетворяющих условию $a_{i,\alpha,j} \overset{\circ}{\sim} a_{j,\beta,i}$, т. е.

$$\forall i \forall j \forall k \forall \alpha \forall \beta \{S_{x_0}^{a_{i,\alpha,j}} [g_j] \simeq S_{x_0}^{a_{j,\beta,i}} [g_i]\} \quad (5.6.2)$$

$$(i \neq j; 1 \leq \alpha \leq s_{i,j}^k; 1 \leq \beta \leq s_{j,i}^k).$$

Определение 2. Классы $W_{i,j,k}$ и $W_{j,i,k}$, удовлетворяющие условию (5.6.2), назовем *взаимно сопряженными классами* подграфов соответственно G_i^{p-1} и G_j^{p-1} и обозначим $W_{i,j,k} \overset{\circ}{\sim} W_{j,i,k}$.

Определение 3. Матрицу W_0 , элементы которой удовлетворяют условию (5.6.1), назовем матрицей сопряженностей для допустимого набора $(p - 1)$ -го порядка Γ^{p-1} .

Относительно матрицы W_0 следует заметить:

1. $\forall i \forall j (L_{i,j} = L_{j,i})$.

2. Вершины x_i и x_j графа G , соответствующие взаимно сопряженным вершинам $a_{i,\alpha,j}$ и $a_{j,\beta,i}$ подграфов G_i^{p-1} и G_j^{p-1} , не обязательно должны быть подобны в графе G , т. е. из выполнения условия (5.6.2) для этих вершин не всегда следует

$$\exists \alpha \{[\alpha(x_i) = x_j] \& [S_0^{\alpha,i} [g] \simeq S_0^{\alpha,j} [g]]\},$$

где α — некоторый автоморфизм (отличный от тождественного автоморфизма e_0) графа G .

3. Элементы по главной диагонали суть пустые множества.

4. j -й столбец матрицы W_0 построен только из вершин $a_{i,\alpha,j}$ графа G_j^{p-1} , поэтому для удобства и наглядности j -й столбец отметим символом G_j .

5. i -я строка матрицы W_0 построена из тех вершин графов G_j^{p-1} ($j = 1, 2, \dots, p$), которые являются «потенциально» возможными значениями неизвестной x_i у искомого графа G , поэтому i -ю строку матрицы W_0 отметим символом x_i (см. табл. 5.6.1).

6. Если $x_i = a_{i,\alpha,i}$, где $a_{i,\alpha,i} \in W_{i,i,k}$, то в силу (5.6.2)

$$x_j = a_{j,\beta,i}, \text{ где } a_{j,\beta,i} \in W_{j,i,k} \quad (1 \leq \alpha \leq s_{i,j}^k; 1 \leq \beta \leq s_{j,i}^k)$$

и наоборот, т. е. имеет место

$$\forall i \forall j \forall k \exists \alpha \exists \beta \{(x_i = a_{i,\alpha,i}) \& (a_{i,\alpha,i} \in W_{i,i,k}) \Leftrightarrow \\ \Leftrightarrow (x_j = a_{j,\beta,i}) \& (a_{j,\beta,i} \in W_{j,i,k})\}.$$

7. Если k -й класс $W_{i,j,k}$ элементов $W_{i,j}$ матрицы W_0 содержит s элементов, то по крайней мере s элементов j -го столбца матрицы W_0 содержат в точности такой же класс, как $W_{i,j,k}$.

Построение матрицы поиска решений. Здесь описывается алгоритм A построения некоторой матрицы W_1 (матрицы поиска решений), основанной на матрице сопряженностей W_0 с учетом следующих замечаний.

1. В силу свойств элементов матрицы W_0 в роли искомого значения неизвестных x_i от графа G_j^{p-1} могут выступать каждый раз произвольные элементы $a_{i,\alpha,j}$, ($\alpha = \overline{1, s_{i,j}^k}$) какого-нибудь только одного класса $W_{i,j,k} \overset{\circ}{\sim} W_{i,j}$, а в силу того что $W_{i,j,k} \overset{\circ}{\sim} W_{j,i,k}$, то значения $a_{j,\beta,i}$ неизвестной x_j от графа G_i^{p-1} однозначно определяются как элементы из класса $W_{j,i,k}$, взаимно сопряженного с классом $W_{i,j,k}$.

2. Если для некоторых i, j, k, α ($1 \leq i \leq p$; $1 \leq j \leq p - 1$; $1 \leq k \leq L_{i,j}$; $1 \leq \alpha \leq s_{i,j}^k$) положить, что $x_i = a_{i,\alpha,i}$, то в силу (5.6.2) число вершин искомого графа G , которые могут принимать значения из $W_{i,j,k}$, равно числу вхождений класса $W_{i,j,k}$ в j -й столбец матрицы W_0 , обозначенному в дальнейшем символом $\tau_{i,j,k}$. Но так как одновременно могут принимать значения $a_{i,\alpha,i}$ только $s_{i,j}^k$ вершин графа G , то, очевидно, число возможных вариантов того, что $s_{i,j}^k$ вершин графа G примут эти значения, равно (τ) , т. е. числу сочетаний из $\tau_{i,j,k}$

элементов по $s_{i,j}^k$.

Прежде чем приступить к описанию алгоритма A , введем следующее определение и обозначения.

Таблица 5.6.1. Матрица

сопряженностей W_0

x_i/G_j	$G \perp x_1$	$G \perp x_2$	$G \perp x_3$	$G \perp x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{6,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{6,4} \\ a_{7,4} \end{Bmatrix}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \\ a_{6,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \\ a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{5,2} \\ a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\begin{Bmatrix} a_{6,4} \\ a_{3,4} \end{Bmatrix}$
x_4	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \\ a_{7,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \\ a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{7,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \\ a_{3,2} \\ a_{5,2} \\ a_{4,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \\ a_{7,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,4} \\ a_{4,4} \\ a_{5,4} \\ a_{6,4} \end{Bmatrix}$
x_6	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \\ a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \\ a_{4,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \\ a_{5,3} \\ a_{6,3} \\ a_{7,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,4} \\ a_{6,4} \end{Bmatrix}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \\ a_{2,2} \\ a_{6,2} \\ a_{3,2} \\ a_{5,2} \\ a_{4,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \\ a_{6,4} \end{Bmatrix}$

$G \perp x_1$	$G \perp x_2$	$G \perp x_3$	$G \perp x_4$
$\begin{Bmatrix} a_{6,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
$\begin{Bmatrix} a_{6,5} \\ a_{3,5} \\ a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7}, a_{7,7} \\ a_{2,7}, a_{6,7} \\ a_{3,7} \\ a_{5,7} \\ a_{4,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
$\begin{Bmatrix} a_{6,5} \\ a_{3,5} \\ a_{5,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6}, a_{2,6} \\ a_{3,6}, a_{4,6} \\ a_{5,6} \\ a_{6,6} \\ a_{7,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \\ a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \\ a_{5,8} \\ a_{6,8} \end{Bmatrix}$
$\begin{Bmatrix} a_{5,5} \\ a_{4,5} \\ a_{7,5} \\ a_{5,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{7,6} \\ a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \\ a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \\ a_{7,6} \\ a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,7} \\ a_{3,7} \\ a_{5,7} \\ a_{2,7}, a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \\ a_{4,8} \\ a_{7,8} \end{Bmatrix}$
$\begin{Bmatrix} a_{3,5} \\ a_{5,5} \\ a_{6,5} \end{Bmatrix}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \\ a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \\ a_{5,8} \\ a_{6,8} \end{Bmatrix}$
$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$

x_i/Q_j	$G \perp x_1$	$G \perp x_2$	$G \perp x_3$	$G \perp x_4$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \\ a_{6,3} \end{Bmatrix}$	$\{a_{4,4}\}$

Определение 4. Если $(\tau_{i,j,k} = s_{i,j}^k \vee (L_{i,j} = 1))$, то вхождение класса $W_{i,j,k}$ в j -й столбец матрицы W_0 назовем *фиксированным вхождением*, в противном случае — *свободным вхождением*.

Количество фиксированных вхождений класса $W_{i,j,k}$ в j -й столбец матрицы W_0 обозначим $f_{i,j,k}$, а количество свободных вхождений — $c_{i,j,k} = \tau_{i,j,k} - f_{i,j,k}$.

Описание алгоритма А. Просматривая классы элемента j -го столбца ($j = 1, p - 1$) i -й строки ($i = j + 1, p$) матрицы W_0 , осуществляем следующие преобразования:

Шаг 1. Если $W_{i,j,k}$ и $W_{i,j,k}$ — классы фиксированного вхождения, то $s_{i,j}^k := s_{i,j}^k - f_{i,j,k}$, $s_{i,l}^k := s_{i,l}^k - f_{i,j,k}$; в противном случае переходим к шагу 3.

Шаг 2. Для каждого $l \neq k$ $\tau_{i,j,l} := \tau_{i,j,l} - 1$; $\tau_{i,l,l} := \tau_{i,l,l} - 1$; классы $W_{i,j,k}$ и $W_{i,l,k}$ аннулируем; на количество аннулированных классов уменьшаем соответственно величины $L_{i,j}$ и $L_{i,l}$.

Шаг 3. Шаги 1—2 повторяем до тех пор, пока у матрицы W_0 будут появляться классы фиксированного вхождения.

Результатом работы данного алгоритма является матрица W_1 (матрица поиска решений (табл. 5.6.2)), построением устраниением таких классов $W_{i,j,l}$ у элементов $W_{i,j,l}$ матрицы сопряженностей W_0 , которые при предположении, что $x_i = a_{i\alpha,l}$, привели бы к очевидным противоречиям.

Алгоритм построения потенциальных решений.

Определение 5. Потенциальным решением для x_i по вершинно-полному допустимому набору $(p - 1)$ -го порядка $G^{p-1} = \{G_j^{p-1}\}$ p -вершинного графа G будем называть произвольное выражение вида

$$\& \left[\&_{i=1,p} [x_i = W_{i,j,k}^1] \right] \quad (i \neq j; 1 \leq k \leq L_{i,j} < p), \quad (5.6.3)$$

где $W_{i,j,k}^1$ — некоторые классы элементов j -го столбца i -й строки матрицы W_1 , а запись $x_i = W_{i,j,k}^1$ означает

$$\bigvee_{i\alpha} [x_i = a_{i\alpha,j}] \& (a_{i\alpha,j} \in W_{i,j,k}^1).$$

табл. 5.6.1

$G \perp x_3$	$G \perp x_4$	$G \perp x_7$	$G \perp x_8$
$\begin{Bmatrix} a_{3,5} \\ a_{4,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \\ a_{6,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Очевидно, выражение вида

$$\&_{i=1,p} \left[\&_{l=1,p} \left[\bigvee_{k=1}^{L_{i,j}} (x_l = W_{i,j,k}^1) \right] \right] \quad (i \neq j), \quad (5.6.4)$$

которое легко строится по матрице поиска решений W_1 и включает все потенциальные решения, ибо в противном случае, при предположении, что $x_i = a_{i\alpha,j}$ ($i \neq j; 1 \leq i_\alpha \leq p$), которому не соответствует ни один дизъюнктивный член выражения (5.6.4), придем к явному противоречию, так как вершины $a_{i\alpha,j}$ должны удовлетворять условию (5.6.2) и как возможные значения x_i должны учитываться при построении матрицы сопряженностей.

Опишем алгоритм B построения всех потенциальных решений для x_i по матрице поиска решений W_1 , избегая как повторных их построений, так и построений выражений, которые не могут быть решениями.

Прежде чем приступить к формальному описанию работы алгоритма B , введем некоторые обозначения и дадим к ним необходимые пояснения: обозначим символами W_n ($n = 1, \dots, N$) матрицы, которые служат исходными данными для алгоритма B при $n = 1$ и являются промежуточными результатами его работы при $1 < n \leq N$; $L_{i,j}^n$ и $W_{i,j,k}^n$ — соответственно количество всех классов и k -й класс элемента $W_{i,j}^n$ матрицы W_n ; $\tau_{i,j,k}^n$ и $f_{i,j,k}^n$ — количество соответственно всех и фиксированных вхождений класса $W_{i,j,k}$ в j -й столбец матрицы W_n ; $W_{i,j,k}^n := \emptyset$ — процесс аннулирования класса $W_{i,j,k}^n$ матрицы W_n ; $A [W_n]$ — результат работы алгоритма A над матрицей W_n .

Процесс порождения матрицы W_n алгоритмом B представим как процесс построения некоторого графа — графа потенциальных решений (рис. 5.6.2), а именно дерева, вершины которого обозначены через $V_n [m; M; \alpha; \beta; l; L_{\alpha,\beta}^m]$ ($1 \leq n \leq N$; $2 \leq m < N$; $1 \leq \alpha, \beta \leq p$; $1 \leq l \leq L_{\alpha,\beta}^m$), где аргумент M есть матрица W_n , порожденная алгоритмом A непосредствен-

Таблица 5.6.2. Матрица

x_i/G_j	$G \pm x_1$	$G \pm x_2$	$G \pm x_3$	$G \pm x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{6,3} \end{Bmatrix}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \\ a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$		$\{a_{6,4}\}$
		$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\{a_{3,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{7,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \\ a_{7,3} \end{Bmatrix}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \\ a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{3,4}\}$
		$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\{a_{6,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \\ a_{6,3} \end{Bmatrix}$	$\{a_{4,4}\}$

поиска решений W_1

	$G \pm x_5$	$G \pm x_6$	$G \pm x_7$	$G \pm x_8$
	$\{a_{6,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{6,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{5,5}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
		$\begin{Bmatrix} a_{7,6} \\ a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \\ a_{7,6} \end{Bmatrix}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
			$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\{a_{5,5}\}$			
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{3,6} \\ a_{4,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \\ a_{6,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

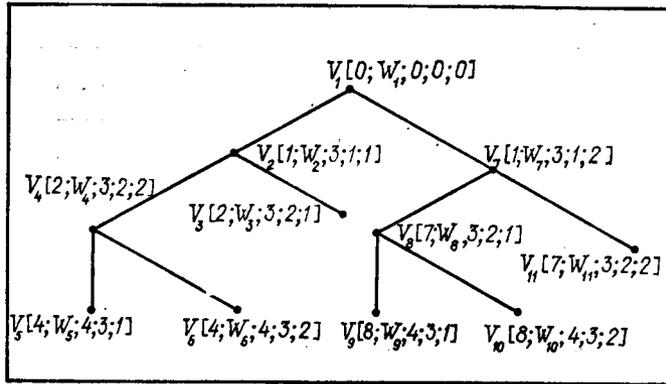


Рис. 5.6.2

но из матрицы W_m при предварительном аннулировании у ее элементов $W_{\alpha,\beta}^m$ всех классов $W_{\alpha,\beta,\gamma}^m$ и $W_{\beta,\alpha,\gamma}^m$, отличных от класса $W_{\alpha,\beta,\alpha}^m$, и при соответствующей корректировке величин $f_{\alpha,\beta,\gamma}^n$, $f_{\beta,\alpha,\gamma}^n$, $\tau_{\alpha,\beta,\gamma}^n$, $\tau_{\beta,\alpha,\gamma}^n$, $L_{\alpha,\beta}^n$, $L_{\beta,\alpha}^n$. Записи вида $m_n = m[V_n]$, $M_n = M[V_n]$, $\alpha_n = \alpha[V_n]$, $\beta_n = \beta[V_n]$, $l_n = l[V_n]$ обозначают выделение соответствующих компонент у вершины V_n .

Через x^r обозначим r -е потенциальное решение, полученное в результате работы i -го шага алгоритма B .

Описание алгоритма B .

Начало.

Шаг 0. $m_0 = m[V_0] := 0$; $M_0 = M[V_0] := W_1$; $\alpha_0 := 0$; $\beta_0 := 0$;

$L_{\alpha,\beta}^0 := 0$; $V_1 := 0$; $n := 1$; $r := 0$; $\delta := 0$; $\lambda := 0$.

Для элементов $W_{i,j}^0$ матрицы $M[V_0]$ выполнить

Шаг 1. $j := 0$.

Шаг 2. $j := j + 1$; $i := j$.

Шаг 3. Если $j = p$, то перейти к шагу 8, иначе шаг 4.

Шаг 4. $i := i + 1$; $k := 0$; если $i > p$, то перейти к шагу 2, иначе $L_{\alpha,\beta}^0 := L_{i,j}^0$; если $L_{\alpha,\beta}^0 = 1$, то перейти к шагу 4, иначе

Шаг 5. $k := k + 1$; если $k > L_{\alpha,\beta}^0$, то (если $m[V_0] = 0$, то перейти на ВЫХОД, иначе перейти к шагу 9), иначе

$\forall \gamma (\gamma \neq k) \{ W_{i,j,\gamma}^0 := \emptyset \} \& (W_{j,i,\gamma}^0 := \emptyset) \&$

$\& (\tau_{i,j,\gamma}^0 := \tau_{i,j,\gamma}^0 - 1) \& (\tau_{j,i,\gamma}^0 := \tau_{j,i,\gamma}^0 - 1)$;

$L_{\alpha,\beta}^0 := 1$; $L_{\beta,\alpha}^0 := 1$; $f_{i,j,k}^0 := f_{i,j,k}^0 + 1$;

$$f_{i,j,k}^0 := f_{i,j,k}^0 + 1;$$

$$\alpha_0 := i; \beta_0 := j; l_0 := k; \delta := 1;$$

если $\lambda = 0$, то перейти к шагу 6, иначе $\{\lambda := 0$; перейти к шагу 7).

Шаг 6. $m[V_0] := n$;

Шаг 7. $n := n + 1$; $W_n := A[M_0]$; $M_0 := W_n$; $V_n := V_0$; перейти к шагу 4.

Шаг 8. $x^r := [\& [\& (x_i = W_{i,j}^n)]]$ ($i \neq j$); $r := r + 1$;

если $\delta = 0$, то перейти на ВЫХОД, иначе

Шаг 9. $s_0 := m_0$; $M_0 := M_s$; $i := \alpha_0$; $j := \beta_0$; $k := l_0$; $\lambda := 1$; если $k = L_{i,j}^1$, то $\{V_0 := V_s$, перейти к шагу 9), иначе перейти к шагу 5.

ВЫХОД

КОНЕЦ

Алгоритм построения искомого графа по потенциальным решениям. Результатом работы алгоритма B является совокупность всех потенциальных решений (табл. 5.6.3—5.6.10) вида

$$\bigvee_{r=1}^m [\& [\& (x_i = W_{i,j}^n)]] \quad (i \neq j; 1 \leq n_r \leq N), \quad (5.6.5)$$

где метки вершин x_i искомого графа G выражены через $W_{i,j}^n = \{a_{i,j}, \dots, a_{s_{i,j},j}\}$, т. е. через метки вершин графов G_j^{n-1} .

Для построения искомого графа G необходимо определить: обратную зависимость, т. е. каждой метке вершин $a_{i,j}$ графов G_j^{n-1} ($j = \overline{1, p}$) поставить в соответствие некоторую совокупность меток x_i вершинам графа G ; все ребра $x_\alpha x_\beta$ инцидентные вершинам x_i ($i \neq j$) одновременно во всех графах G_j^{n-1} , за исключением случаев, когда $j = \alpha$ или $j = \beta$. С этой целью сначала произведем подстановки переменных x_i во все квадратичные формы g_j ($j = \overline{1, p}$) вместо меток вершин z $W_{i,j}^n$, найденных алгоритмом B .

Результаты применения оператора подстановки обозначим через $\bar{g}_j(x_0, x_1, \dots, x_i, \dots, x_p)$ ($i \neq j$).

Следует заметить, что в силу свойств элементов матрицы сопряженностей W_0 каждый из $s_{i,j}$ элементов класса $W_{i,j}^n = \{a_{i,j}, \dots, a_{s_{i,j},j}\}$ может быть значением одной и той же неизвестной x_i так же, как ровно $s_{i,j}$ различных неизвестных x_i могут одинаково определяться через элементы одного и того же класса $W_{i,j}^n$. Поэтому в квадратичной форме $\bar{g}_j(x_0, x_1, \dots, x_i, \dots, x_p)$ — результате применения оператора подстановки S к квадратичной форме g_j индексами неизвестных будут служить

Таблица 5.6.3. Матрица W_2 , построенная

по матрице W_1 при $i=3, j=1$ и $k=1$

x_i	$G \pm x_i$			
	$G \pm x_1$	$G \pm x_2$	$G \pm x_3$	$G \pm x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \\ a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\begin{Bmatrix} a_{6,4} \\ a_{3,4} \end{Bmatrix}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{7,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \\ a_{7,3} \end{Bmatrix}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \\ a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,3} \\ a_{6,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,4} \\ a_{6,4} \end{Bmatrix}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{4,4}\}$

a_j	$G \pm x_i$			
	$G \pm x_5$	$G \pm x_6$	$G \pm x_7$	$G \pm x_8$
	$\{a_{6,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{3,5} \\ a_{5,5} \end{Bmatrix}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \\ a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\begin{Bmatrix} a_{7,6} \\ a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \\ a_{7,6} \end{Bmatrix}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \\ a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{3,6} \\ a_{4,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.4. Матрица W_3 — 1-е потенциальное решение,

x_i	$G \pm x_i$			
	$G \pm x_1$	$G \pm x_2$	$G \pm x_3$	$G \pm x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$		$\{a_{3,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\{a_{7,3}\}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{6,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{5,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{4,4}\}$

полученное из матрицы W_2 при $l=3, j=2$ и $k=1$

G_i	$G \pm x_i$			
	$G \pm x_5$	$G \pm x_6$	$G \pm x_7$	$G \pm x_8$
	$\{a_{6,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{3,5}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\{a_{7,6}\}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{5,5}\}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.5. Матрица W_4 , построенная

x_i	G_i			
	$G \perp x_1$	$G \perp x_2$	$G \perp x_3$	$G \perp x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\begin{Bmatrix} a_{6,4} \\ a_{3,4} \end{Bmatrix}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{7,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \\ a_{7,3} \end{Bmatrix}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,4} \\ a_{6,4} \end{Bmatrix}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{4,4}\}$

из матрицы W_5 , при $i=3$, $j=2$ и $k=2$

G_j	G_j			
	$G \perp x_5$	$G \perp x_6$	$G \perp x_7$	$G \perp x_8$
	$\{a_{6,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\{a_{7,7}\}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{3,5} \\ a_{5,5} \end{Bmatrix}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \\ a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\begin{Bmatrix} a_{7,6} \\ a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \\ a_{7,6} \end{Bmatrix}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{3,5} \\ a_{5,5} \end{Bmatrix}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \\ a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \\ a_{3,6} \\ a_{4,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{6,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.6. Матрица W_5 —2-е потенциальное решение,

x_i	G_j			
	$G \perp x_1$	$G \perp x_2$	$G \perp x_3$	$G \perp x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\{a_{6,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\{a_{7,3}\}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{3,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{4,4}\}$

полученное из матрицы W_4 при $i=4, j=3$ и $k=2$

x_i	G_j			
	$G \perp x_5$	$G \perp x_6$	$G \perp x_7$	$G \perp x_8$
	$\{a_{6,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\{a_{7,7}\}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{6,6}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\{a_{7,6}\}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.7. Матрица W_6 — 3-е потенциальное решение,

x_i	$G \rightarrow x_j$			
	$G \rightarrow x_1$	$G \rightarrow x_2$	$G \rightarrow x_3$	$G \rightarrow x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\{a_{3,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\{a_{7,3}\}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{6,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{4,4}\}$

полученное из матрицы W_4 при $i=4, j=3$ и $k=2$

a_j	$G \rightarrow x_i$			
	$G \rightarrow x_5$	$G \rightarrow x_6$	$G \rightarrow x_7$	$G \rightarrow x_8$
	$\{a_{6,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{3,5}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\{a_{7,6}\}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.8. Матрица W_9 — 4-е потенциальное решение,

x_i	G_j			
	$G \rightarrow x_1$	$G \rightarrow x_2$	$G \rightarrow x_3$	$G \rightarrow x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$		$\{a_{6,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\{a_{7,3}\}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{3,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\{a_{4,4}\}$

полученное из матрицы W_9 при $i=4$, $j=3$ и $k=1$

G_j	G_j			
	$G \rightarrow x_5$	$G \rightarrow x_6$	$G \rightarrow x_7$	$G \rightarrow x_8$
	$\{a_{6,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{5,5}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\{a_{7,6}\}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.9. Матрица W_{10} — 5-е потенциальное решение,

x_i	G_j			
	$G \rightarrow x_1$	$G \rightarrow x_2$	$G \rightarrow x_3$	$G \rightarrow x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$		$\{a_{3,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\{a_{7,3}\}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{6,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\{a_{1,4}\}$

полученное из матрицы W_8 при $i=4$, $j=3$ и $k=2$

x_i	G_j			
	$G \rightarrow x_5$	$G \rightarrow x_6$	$G \rightarrow x_7$	$G \rightarrow x_8$
x_1	$\{a_{6,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
x_2	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
x_3	$\{a_{3,5}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
x_4	$\{a_{7,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
x_5		$\{a_{7,6}\}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
x_6	$\{a_{5,5}\}$		$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
x_7	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
x_8	$\{a_{4,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.10. Матрица W_{11} — 6-е потенциальное

x_i	$G \pm x_j$			
	$G \pm x_1$	$G \pm x_2$	$G \pm x_3$	$G \pm x_4$
x_1		$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{6,3}\}$	$\{a_{7,4}\}$
x_2	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$		$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_3	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$		$\{a_{6,4}\}$
x_4	$\{a_{7,1}\}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	
x_5	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\{a_{4,2}\}$	$\{a_{7,3}\}$	$\{a_{5,4}\}$
x_6	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,2} \\ a_{6,2} \end{Bmatrix}$	$\{a_{5,3}\}$	$\{a_{3,4}\}$
x_7	$\begin{Bmatrix} a_{1,1} \\ a_{2,1} \\ a_{5,1} \\ a_{6,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,2} \\ a_{7,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,3} \\ a_{2,3} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,4} \\ a_{2,4} \end{Bmatrix}$
x_8	$\begin{Bmatrix} a_{3,1} \\ a_{4,1} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,2} \\ a_{5,2} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,3} \\ a_{4,3} \end{Bmatrix}$	$\{a_{4,4}\}$

решение, полученное из матрицы W_7 при $l=3, j=2$ и $k=2$

a_j	$G \pm x_k$			
	$G \pm x_5$	$G \pm x_6$	$G \pm x_7$	$G \pm x_8$
	$\{a_{6,5}\}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,6} \\ a_{2,6} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{5,5}\}$	$\{a_{5,6}\}$	$\begin{Bmatrix} a_{2,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{7,5}\}$	$\{a_{7,6}\}$	$\begin{Bmatrix} a_{1,7} \\ a_{7,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
		$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$	$\{a_{4,7}\}$	$\begin{Bmatrix} a_{4,8} \\ a_{7,8} \end{Bmatrix}$
	$\{a_{3,5}\}$		$\begin{Bmatrix} a_{3,7} \\ a_{6,7} \end{Bmatrix}$	$\begin{Bmatrix} a_{5,8} \\ a_{6,8} \end{Bmatrix}$
	$\begin{Bmatrix} a_{1,5} \\ a_{2,5} \end{Bmatrix}$	$\begin{Bmatrix} a_{3,6} \\ a_{4,6} \end{Bmatrix}$		$\begin{Bmatrix} a_{1,8} \\ a_{2,8} \\ a_{3,8} \end{Bmatrix}$
	$\{a_{4,5}\}$	$\{a_{6,6}\}$	$\begin{Bmatrix} a_{3,7} \\ a_{5,7} \end{Bmatrix}$	

Таблица 5.6.11. Матрица

x_i	x_1	x_2	x_3	x_4
G_i				
	G_1 G_2 G_3 G_4			
	G_5 G_6 G_7 G_8			
x_1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0 1 0 1 1 1 1 1	1 1 0 0 0 1 1 1
x_2		0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
x_3			0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
x_4				0 0 0 0 0 0 0 0
x_5				
x_6				
x_7				
x_8				

выражения $I = \{i_1, \dots, i_{s_i}\}$, т. е. квадратичную форму \bar{g}_i можно представить в виде

$$\bar{g}_i = \sum_{t=0}^p x_0 x_t + \sum_{I, J} \alpha_{I, J} x_I x_J \quad (i \neq j; j = \overline{1, p}),$$

где I и J — множества индексов (см. рис. 5.6.3). Графы G_i , соответствующие квадратичным формам \bar{g}_i , представлены на рис. 5.6.3.

По квадратичным формам $g_i(x_0, x_1, \dots, x_i, \dots, x_p)$ ($i \neq j$) строим квадратичную матрицу смежностей $Z = \{z_{i,j}\}$ ($i, j =$

смежностей Z

	x_5	x_6	x_7	x_8
	G_1 G_2 G_3 G_4	G_1 G_2 G_3 G_4	G_2 G_2 G_3 G_4	G_2 G_2 G_3 G_4
	G_5 G_6 G_7 G_8			
	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0	0 1 0 0 0 0 1 0
	1 0 0 0 0 0 0 0	1 0 1 1 1 0 1 1	1 0 1 1 1 1 0 1	1 0 0 0 0 0 0 0
	1 1 0 1 0 1 1 1	1 1 0 0 0 0 1 0	1 0 0 0 0 0 0 0	1 1 0 1 1 1 1 0
	0 0 0 0 0 0 0 0	0 1 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 1 1 1 0 1 1 0
		0 0 0 0 0 0 0 0	1 1 1 1 1 0 0 1	1 1 1 1 1 0 1 0
			0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
				0 0 0 0 0 0 0 0

$= \overline{1, p}$), у которой каждый элемент $z_{i,j}$ представляет собой последовательность вида $(\gamma_1, \gamma_2, \dots, \gamma_k, \dots, \gamma_p)$, где $\gamma_i = 0$, $\gamma_j = 0$, при $(k \neq i) \& (k \neq j)$

$$\gamma_k = \begin{cases} 1, & \text{если в графе } \bar{G}_k \text{ вершины } x_i \text{ и } x_j \text{ смежны;} \\ 0 & \text{в противном случае.} \end{cases}$$

Относительно матрицы смежностей Z для обыкновенных графов следует заметить: $z_{i,j} = z_{j,i}$; у элементов $z_{i,j} = (\gamma_1, \dots, \gamma_k, \dots, \gamma_p)$ все γ_k равны нулю, т. е. матрица Z симметрична относительно главной диагонали (поэтому достаточно ограни-

читься рассмотрением только одной треугольной матрицы, расположенной либо выше, либо ниже главной диагонали (табл. 5.6.11)); выполнение для всех $k = \overline{1, p}$ равенства $\gamma_k = 1$, за исключением случаев, когда $k = 1$ или $k = j$, означает, что ребро $x_i x_j$ инцидентно вершинам x_i и x_j одновременно во всех графах \overline{G}_k ($k = \overline{1, p}$).

Затем по матрице смежностей Z легко строим квадратичную форму вида $g(x_0, x_1, \dots, x_i, \dots, x_p) = \sum_{i,j=0}^p \beta_{i,j} x_i x_j$, соответствующую некоторому потенциальному решению (5.6.5).

Пример. Пусть

$$g_1 = \sum_{i=1}^7 x_0 a_{i,1} + a_{1,1} a_{2,1} + a_{1,1} a_{3,1} + a_{2,1} a_{3,1} + a_{4,1} a_{3,1} + \\ + a_{4,1} a_{5,1} + a_{4,1} a_{6,1} + a_{5,1} a_{6,1};$$

$$g_2 = \sum_{i=1}^7 x_0 a_{i,2} + a_{2,2} a_{1,2} + a_{2,2} a_{3,2} + a_{4,2} a_{3,2} + a_{4,2} a_{5,2} + \\ + a_{6,2} a_{5,2} + a_{6,2} a_{7,2} + a_{3,2} a_{5,2};$$

$$g_3 = \sum_{i=1}^7 x_0 a_{i,3} + a_{1,3} a_{2,3} + a_{4,3} a_{3,3} + a_{4,3} a_{5,3} + a_{6,3} a_{5,3} + \\ + a_{6,3} a_{7,3} + a_{3,3} a_{5,3};$$

$$g_4 = \sum_{i=1}^7 x_0 a_{i,4} + a_{3,4} a_{1,4} + a_{3,4} a_{2,4} + a_{3,4} a_{4,4} + a_{6,4} a_{4,4} + \\ + a_{6,4} a_{5,4} + a_{6,4} a_{7,4} + a_{1,4} a_{2,4} + a_{4,4} a_{5,4};$$

$$g_5 = \sum_{i=1}^7 x_0 a_{i,5} + a_{1,5} a_{3,5} + a_{2,5} a_{1,5} + a_{2,5} a_{3,5} + a_{4,5} a_{3,5} + \\ + a_{4,5} a_{5,5} + a_{6,5} a_{5,5} + a_{6,5} a_{7,5};$$

$$g_6 = \sum_{i=1}^7 x_0 a_{i,6} + a_{1,6} a_{2,6} + a_{5,6} a_{3,6} + a_{5,6} a_{4,6} + a_{5,6} a_{6,6} + \\ + a_{3,6} a_{4,6} + a_{6,6} a_{7,6};$$

$$g_7 = \sum_{i=1}^7 x_0 a_{i,7} + a_{2,7} a_{1,7} + a_{2,7} a_{3,7} + a_{4,7} a_{3,7} + a_{4,7} a_{5,7} + \\ + a_{6,7} a_{5,7} + a_{6,7} a_{7,7} + a_{3,7} a_{5,7};$$

$$g_8 = \sum_{i=1}^7 x_0 a_{i,8} + a_{1,8} a_{2,8} + a_{1,8} a_{3,8} + a_{2,8} a_{3,8} + a_{3,8} a_{4,8} + \\ + a_{5,8} a_{6,8} + a_{6,8} a_{7,8};$$

— квадратичные формы соответственно графов $G_j^{p-1} = G - x_j$ ($j = \overline{1, 8}$) (см. рис. 5.6.1), которые образуют вершинно-полный допустимый набор 7-го порядка некоторого 8-ми вершинного графа G .

Требуется восстановить искомый граф G .

Решение.

Шаг 1. В соответствии с алгоритмом A строим матрицу сопряженностей (см. табл. 5.6.1).

Шаг 2. В соответствии с алгоритмом A строим матрицу поиска решений (табл. 5.6.2).

Шаг 3. В соответствии с алгоритмом B строим граф (дерево) процесса построения потенциальных решений (см. рис. 5.6.2).

Матрицы $W_3, W_5, W_6, W_9, W_{10}$ и W_{11} , соответствующие вершинам $V_3, V_5, V_6, V_9, V_{10}$ и V_{11} дерева потенциальных решений, определяют однозначно потенциальные решения по формуле

$$(x'_i = W_{i,1}^{n_r}) \& \dots \& (x'_i = W_{i,i}^{n_r}) \& \dots \& (x'_i = W_{i,p}^{n_r}),$$

где $i \neq j; 1 \leq r \leq 6; n_r \in \{3; 5; 6; 9; 10; 11\}$ (см. табл. 5.6.4, 5.6.6—5.6.10). Например, при $i = 3$ и $r = 1$, следовательно, при $n_r = 3$ имеем

$$\{(x_3^1 = a_{1,1}) \vee (x_3^1 = a_{2,1}) \vee (x_3^1 = a_{5,1}) \vee (x_3^1 = a_{6,1})\} \& \\ \& \{(x_3^1 = a_{2,2}) \vee (x_3^1 = a_{6,2})\} \& (x_3^1 = a_{3,4}) \& (x_3^1 = a_{3,5}) \& \\ \& (x_3^1 = a_{5,6}) \& \{(x_3^1 = a_{3,7}) \vee (x_3^1 = a_{5,7})\} \& \{(x_3^1 = a_{5,8}) \vee \\ \vee (x_3^1 = a_{3,8})\}. \quad (5.6.6)$$

С целью сокращения изложения в последующих шагах нашего алгоритма ограничимся рассмотрением случая $r = 2$, т. е. $n_r = 5$ (см. рис. 5.6.3), при этом для наглядности не будем изображать верхние индексы при x'_j .

Шаг 4. Производим подстановку переменных x_i вместо правых частей равенства (5.6.6) во все квадратичные формы $g_j(x_0, a_{1,j}, \dots, a_{i,j}, \dots, a_{7,j})$ ($j = \overline{1, 8}$). В результате получаем квадратичные формы

$$\overline{g}_1 = \sum_{i=2}^8 x_0 x_i + x_{2,3,5,7} x_{2,3,5,7} + x_{2,3,5,7} x_{6,8} + x_{6,8} x_{6,8};$$

$$\overline{g}_2 = \sum_{\substack{i=1; \\ i \neq 2}}^8 x_0 x_i + x_{1,6} (x_{4,7} + x_{3,8}) + x_{3,8} (x_{3,8} + x_5);$$

$$\overline{g}_3 = \sum_{\substack{i=1; \\ i \neq 3}}^8 x_0 x_i + x_1 x_4 + x_{2,7} (x_{2,7} + x_6) + x_5 (x_5 + x_8);$$

$$\bar{g}_4 = \sum_{\substack{i=1; \\ i \neq 4}}^8 x_0 x_i + x_{2,7}(x_{2,7} + x_8) + x_3(x_1 + x_5 + x_8) + x_8(x_5 + x_8);$$

$$\bar{g}_5 = \sum_{\substack{i=1; \\ i \neq 5}}^8 x_0 x_i + x_1(x_3 + x_4) + x_8(x_3 + x_8) + x_{2,7}(x_{2,7} + x_6);$$

$$\bar{g}_6 = \sum_{\substack{i=1; \\ i \neq 6}}^8 x_0 x_i + x_1(x_3 + x_4) + x_{5,8}(x_{5,8} + x_3) + x_{2,7}x_{2,7};$$

$$\bar{g}_7 = \sum_{\substack{i=1; \\ i \neq 7}}^8 x_0 x_i + x_{1,6}(x_{2,4} + x_{3,8}) + x_{3,8}(x_{3,8} + x_5);$$

$$\bar{g}_8 = \sum_{i=1}^8 x_0 x_i + x_{2,6,7}x_{2,6,7} + x_{1,3}(x_{1,3} + x_{4,5}),$$

соответствующие графам \bar{G}_j и представленные на рис. 5.6.3.

Шаг 5. По квадратичным формам \bar{g}_j ($x_0, x_1, \dots, x_i, \dots, x_8$) ($i \neq j; j = \overline{1, 8}$) строим матрицу смежностей $Z = \{z_{i,j}\}$ ($i, j = \overline{1, 8}$) в соответствии с алгоритмом, описанным в п. 5.6.4 (табл. 5.6.11).

Шаг 6. По матрице смежностей Z восстанавливаем квадратичную форму:

$$\begin{aligned} \tilde{g}(x_0, x_1, \dots, x_8) = & \sum_{i=1}^8 x_0 x_i + x_1(x_3 + x_4) + x_2(x_6 + x_7) + \\ & + x_3(x_5 + x_8) + x_5(x_6 + x_8) + x_6x_7, \end{aligned}$$

соответствующую некоторому графу \tilde{G} (рис. 5.6.4)

Шаг 7. Для данного потенциального решения ($n_r = 5, r = 2$) проверяем изоморфизм квадратичных форм g_j и $\tilde{g} \pm x_j$ для каждого $j = \overline{1, 8}$. В результате, действительно, $\forall j [g_j \cong (\tilde{g} \pm x_j)]$; следовательно, квадратичная форма $\tilde{g}(x_0, x_1, \dots, x_8)$ — квадратичная форма некоторого графа \tilde{G} , изоморфного искомому графу G .

Если на шаге 4 алгоритма восстановления рассматривать случай $r = 5$, т. е. $n_r = 10$, то квадратичная форма, восстановленная по матрице Z , имеет вид

$$\begin{aligned} h(x_0, x_1, \dots, x_8) = & \sum_{i=1}^8 x_0 x_i + x_8x_8 + x_1(x_4 + x_6) + \\ & + x_2(x_3 + x_7) + x_3(x_7 + x_8) + x_5(x_6 + x_8) \end{aligned}$$

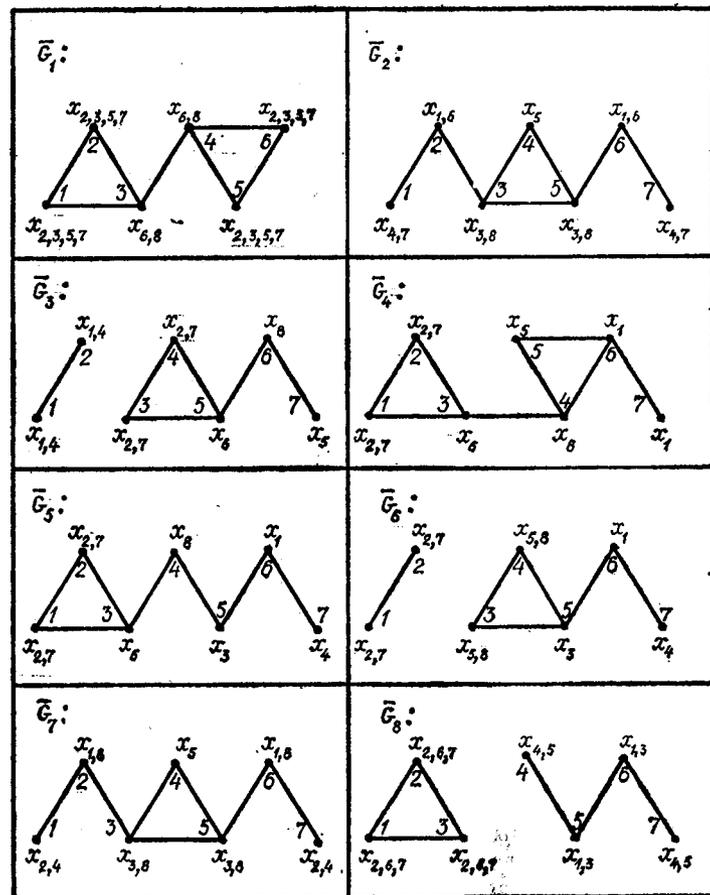


Рис. 5.6.3

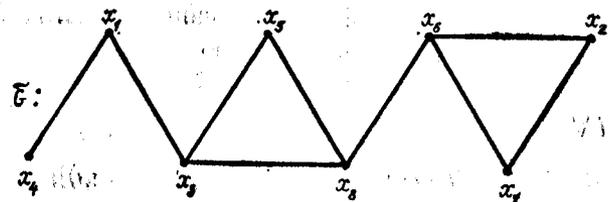


Рис. 5.6.4

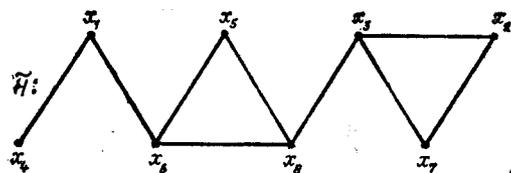


Рис. 5.6.5

и соответствует графу \bar{H} (рис. 5.6.5), изоморфному графу G .

Для случаев $r = 1, 3, 4, 6$ уже на шаге 5, а именно при построении первой строчки матрицы Z , становится очевидным противоречие: для вершины x_1 нет вершины x_i , которая с ней смежна во всех других графах G_j^{p-1} ($j = \overline{1, 8}; j \neq i; j \neq 1$).

5.7. ТЕОРЕМА О СУЩЕСТВОВАНИИ И ЕДИНСТВЕННОСТИ

Теорема 5.7.1. Пусть $\Gamma_G = \{G_k^{p-1}\}$ ($k = \overline{1, p}; p \geq 3$) — вершинно-полный допустимый набор $(p-1)$ -го порядка для некоторого p -вершинного графа G . Существует эффективная процедура, позволяющая за конечное число шагов восстановить однозначно с точностью до изоморфизма искомый граф по его вершинно-полному допустимому набору $(p-1)$ -го порядка.

Доказательство теоремы состоит из двух частей (А и В): А — доказательство существования решения, т. е. доказывається, что алгоритм восстановления В, описанный по вершинно-полному допустимому набору подграфа $(p-1)$ -го порядка $\{G_j^{p-1}\}$ ($j = \overline{1, p}$) некоторого (p, q) -графа, найдет хотя бы один искомый (p, q) -граф G с тем же самым исходным вершинно-полным допустимым набором подграфов; В — доказательство единственности решения (с точностью до изоморфизма).

А. Доказательство существования. Пусть

$$a) g_l(x_0, a_{1,l}, a_{2,l}, \dots, a_{p,l}) = \sum_{k=1}^{p-1} x_0 a_{k,l} + \sum_{k,l=1}^{p-1} \alpha_{k,l}^l a_{k,l} a_{l,l}$$

($l = \overline{1, p}$) — квадратичные формы вершинно-полного допустимого набора $(p-1)$ -го порядка подграфов $\Gamma_G = \{G_l^{p-1}\}$ ($l = \overline{1, p}$) некоторого (p, q) -графа G , причем

$$b) \forall i \forall j \exists k \exists l [S_{x_0}^{a_{k,l}} [g_l(x_0, a_{1,l}, \dots, a_{k,l}, \dots, a_{p-1,l})] = S_{x_0}^{a_{l,i}} [g_l(x_0, a_{1,l}, \dots, a_{l,l}, \dots, a_{p-1,l})]] \\ (i, j = \overline{1, p}; 1 \leq k, l \leq p-1).$$

Покажем, что алгоритм восстановления построит хотя бы один $(p-q)$ -граф G такой, что его квадратичная форма $g(x_0, x_1, \dots, x_l, \dots, x_p)$ удовлетворяет условию

$$c) \forall j [S_{x_0}^{x_j} [g(x_0, x_1, \dots, x_l, \dots, x_p)] \simeq g_l(x_0 a_{1,j}, \dots, a_{l,j}, \dots, a_{p-1,j})].$$

Действительно, так как удовлетворяются условия «а» и все элементы $W_{i,j}^0$ матрицы сопряженностей W_0 формируются с учетом выполнения условий (5.6.2), то на этапе построения алгоритмом А матрицы W_0 не будет упущено ни одно потенциально возможное значение $W_{i,j,k}^0$ ($1 \leq i \leq p; j = \overline{1, p}; 1 \leq k \leq p$), которое может принимать неизвестная x_i .

На этапе построения алгоритмом А матрицы W_1 у элементов $W_{i,j}^0$ матрицы W_0 аннулируются классы $W_{i,j,k}^0$, что при предположении $x = W_{i,j,k}^0$ приводит к противоречию, заключающемуся в том, что класс $W_{i,j,k}^0$ ($k \neq 1$) уже не сможет быть значением неизвестной x_i , которая входит в число $s_{i,j}^k$ неизвестных, обязанных (в силу условий 5.6.2) принимать значение $W_{i,j,k}^0$, т. е. приходим к явному противоречию: $\tau_{i,j,k} < s_{i,j}^k$.

Следовательно, и на этом этапе не будет упущено ни одно потенциально возможное значение неизвестной x_i . Что же касается алгоритма В, то достаточно заметить, что на этом этапе работа алгоритма восстановления заключается, по существу, в организации направленного перебора всех вариантов возможных значений неизвестной x_i (см. 5.6.4).

Наконец, поскольку на следующем этапе работы алгоритма для каждого потенциального решения (см. 5.6.5) строим соответствующую матрицу смежностей Z_m , то рано или поздно найдем такую матрицу смежностей, что построенная на ней квадратичная форма удовлетворяет условию «с», если, действительно, набор «а» удовлетворял условиям «b».

В. Единственность решения. Пусть

$g(x_0, x_1, \dots, x_l, \dots, x_p) = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$ — квадратичная форма $p > 3$ -вершинного графа G , восстановленного алгоритмом по соответствующим квадратичным формам

$$g_k(x_0, x_1, \dots, x_l, \dots, x_p) = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j \\ (i, j = k; k = \overline{1, p})$$

его вершинно-полного допустимого набора подграфов $(p-1)$ -го порядка $\Gamma_G = \{G_k^{p-1}\}$ ($k = \overline{1, p}$). (Здесь нумерация как

самих подграфов G_k^{p-1} , так и вершин этих подграфов произвольная и не зависит от нумерации вершин графа G .) Требуется доказать, что граф G единственный с точностью до изоморфизма.

Прежде чем приступить непосредственно к доказательству единственности, введем некоторые обозначения, дадим необходимые определения и следствия, вытекающие из них.

Определение 1. Квадратичные формы

$$g(x_0, x_1, \dots, x_i, \dots, x_p) = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j,$$

$$h(x_0, x_1, \dots, x_i, \dots, x_p) = \sum_{i,j=0}^p \beta_{i,j} x_i x_j$$

назовем тождественно равными ($g = h$), если $\forall i \forall j (\alpha_{i,j} = \beta_{i,j})$.

Определение 2. Пусть $f(x_0, x_1, \dots, x_i, \dots, x_p) = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$ —

некоторая квадратичная форма с p переменными; $N = \{1, 2, \dots, p\}$ — множество индексов при переменных; S_p — совокупность всех перестановок на множестве N_p . Для произвольной перестановки $\varphi \in S_p$ определим ее действие на квадратичную форму $f(x_0, x_1, \dots, x_i, \dots, x_p)$, положив

$$\varphi(f(x_0, x_1, \dots, x_i, \dots, x_p)) = f(x_0, x_{\varphi(1)}, \dots, x_{\varphi(i)}, \dots,$$

$$\dots, x_{\varphi(p)}) = \sum_{i,j=0}^p \alpha_{i,j} x_{\varphi(i)} x_{\varphi(j)} = \sum_{i,j=0}^p \gamma_{i,j} x_i x_j.$$

Определение 3. Квадратичные формы

$$g(x_0, x_1, \dots, x_i, \dots, x_p) = \sum_{i,j=1}^p \alpha_{i,j} x_i x_j,$$

$$h(v_0, v_1, \dots, v_i, \dots, v_p) = \sum_{i,j=0}^p \beta_{i,j} v_i v_j$$

назовем условно изоморфными, если существует хотя бы одна биекция (отображение)

$$\omega = \begin{pmatrix} v_0 v_1 \dots v_i \dots v_p \\ x_0 x_{j_1} \dots x_{j_i} \dots x_{j_p} \end{pmatrix},$$

удовлетворяющая соотношению

$$\omega(h(v_0, v_1, \dots, v_i, \dots, v_p)) = g(x_0, x_1, \dots, x_i, \dots, x_p).$$

Лемма 5.7.1. Пусть $f = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$ — некоторая квадратичная форма; $\varphi = \begin{pmatrix} 1 & 2 & \dots & i & \dots & p \\ k_1 & k_2 & \dots & k_i & \dots & k_p \end{pmatrix}$ — произвольная перестановка

p -й степени на множестве N_p . Тогда справедливо соотношение

$$\varphi(f) = \varphi\left(\sum_{i,j}^p \alpha_{i,j} x_i x_j\right) = \sum_{i,j=0}^p \alpha_{i,j} x_{\varphi(i)} x_{\varphi(j)} = \sum_{i,j=0}^p \alpha_{\varphi^{-1}(i), \varphi^{-1}(j)} x_i x_j.$$

Действительно, непосредственно из определения имеем

$$\varphi\left(\sum_{i,j=0}^p \alpha_{i,j} x_i x_j\right) = \sum_{i,j=0}^p \alpha_{i,j} x_{\varphi(i)} x_{\varphi(j)}.$$

Если положить $\varphi(i) = m$ и $\varphi(j) = n$, $i = \varphi^{-1}(m)$ и $j = \varphi^{-1}(n)$, то результат действия перестановки φ на выражение $\alpha_{i,j} x_i x_j$ можно представить в виде

$$\varphi(\alpha_{i,j} x_i x_j) = \alpha_{i,j} x_{\varphi(i)} x_{\varphi(j)} = \alpha_{i,j} x_m x_n = \alpha_{\varphi^{-1}(m), \varphi^{-1}(n)} x_m x_n.$$

Следовательно, результат действия перестановки на всю квадратичную форму f можем записать как

$$\begin{aligned} \varphi\left(\sum_{i,j=0}^p \alpha_{i,j} x_i x_j\right) &= \sum_{i,j=0}^p \alpha_{i,j} x_{\varphi(i)} x_{\varphi(j)} = \sum_{m,n=0}^p \alpha_{\varphi^{-1}(m), \varphi^{-1}(n)} x_m x_n = \\ &= \sum_{i,j=0}^p \alpha_{\varphi^{-1}(i), \varphi^{-1}(j)} x_i x_j. \end{aligned}$$

Вводя обозначение $S_0^k[f]$ для результата действия оператора подстановки S на квадратичную форму по правилу [15]

$$S_0^k[f] = S_0^k \left[\sum_{i,j=0}^p \alpha_{i,j} x_i x_j \right] = S_0^k \left[\sum_{\substack{i,j=0 \\ i,j \neq k}}^p \alpha_{i,j} x_i x_j + \right.$$

$$\left. + \sum_{\substack{l=0 \\ l \neq k}}^p \alpha_{k,l} x_k x_l \right] = \sum_{\substack{i,j=0 \\ i,j \neq k}}^p \alpha_{i,j} x_i x_j + \sum_{\substack{l=0 \\ l \neq k}}^p \alpha_{0,l} x_0 x_l = \sum_{\substack{i,j=0 \\ i,j \neq k}}^p \alpha_{i,j} x_i x_j$$

(т. е. переменная x_k заменяется на x_0 , выражение $x_0 x_0$ — на пустой символ, а выражение $x_i x_j + x_i x_j$ — на $x_i x_j$), дадим следующие определения.

Определение 1. Переменную x_n ($1 \leq n \leq p$) квадратичной формы $f(x_0, x_1, \dots, x_n, \dots, x_p)$ назовем асимметричной, если справедливо утверждение

$$\forall i ((i \neq n) \Rightarrow [S_0^i[f] \neq S_0^n[f]]) \Leftrightarrow \neg \{ \exists i [(i \neq n) \& [S_0^i[f] \simeq S_0^n[f]]] \}.$$

Определение 2. Некоторую перестановку ω p -й степени назовем автоморфизмом квадратичной формы $f(x_0, x_1, \dots, x_i, \dots, x_p)$, если имеет место соотношение $f = \omega(f)$.

Определение 3. Переменные x_k и x_l ($1 \leq k \neq l \leq p$) квадратичной формы $f(x_0, x_1, \dots, x_k, \dots, x_l, \dots, x_p)$ назовем подобны-

ми, если для некоторого автоморфизма этой квадратичной формы имеет место соотношение $\omega(k) = l$.

Определение 4. Переменные x_u и x_v ($1 \leq u \neq v \leq p$) квадратичной формы $f(x_0, x_1, \dots, x_u, \dots, x_v, \dots, x_p)$ назовем квазиподобными, если для этих переменных справедливо соотношение $S_0^u[f] \simeq S_0^v[f]$ и для любого автоморфизма ω этой квадратичной формы имеет место $\omega(u) \neq v$.

Обозначив символом M_f совокупность индексов асимметричных переменных, символами P_{ii}^α ($\alpha = 1, 2, \dots, \delta < p$) — совокупности индексов попарных переменных, а символами Q_j^β ($\beta = 1, 2, \dots, \rho < p$) — совокупности индексов квазиподобных переменных квадратичной формы $f(x_0, x_1, \dots, x_i, \dots, x_p)$, сформулируем некоторые следствия из только что приведенных определений. Для произвольной квадратичной формы $f(x_0, x_1, \dots, x_i, \dots, x_p)$ справедливы следующие утверждения.

Следствие 1. $\forall \alpha \forall k \forall l \{k \neq l\} \& (k \in P_{ii}^\alpha) \& (l \in P_{ii}^\alpha) \Rightarrow [S_0^k[f] \simeq S_0^l[f]] \& \exists \omega [(\omega \neq e_0) \& [f = \omega(f)] \& [l = \omega(k)]]$, где e_0 — тождественный автоморфизм.

Следствие 2.

$$\forall m \forall \omega \forall \beta \{(m \in P_{ii}^\beta) \& [f = \omega(f)] \Rightarrow [\omega(m) = m]\} \quad (\beta = \overline{1, \delta}),$$

где ω пробегает автоморфизмы квадратичной формы f .

Следствие 3.

$$\forall m \forall n \forall \beta \{(m \in P_{ii}^\beta) \& (n \in M_f) \Rightarrow \forall i [(i \in P_{ii}^\beta) \Rightarrow (\alpha_{n,i} = \alpha_{m,n})]\}.$$

Следствие 4.

$$\forall \omega \forall v \{(\omega \in Q_j^\beta) \& (v \in Q_j^\beta) \quad (\omega \neq v) \Leftrightarrow$$

$$\Leftrightarrow [S_0^\omega[f] \simeq S_0^v[f]] \& [\exists \omega [(\omega \neq e_0) \& (f = \omega(f)) \& (v = \omega(\omega))]]\}.$$

Следствие 5.

$$\forall m \forall n \forall \omega \forall \alpha \forall \beta \forall \varphi \{(m \in M_f) \& (n \in P_{ii}^\alpha) \& (\omega \in Q_j^\beta) \& \forall i [S_0^i[f] \simeq S_0^i[\omega(f)]] \Rightarrow (m = \varphi(m)) \& (\varphi(n) \in P_{ii}^\alpha) \& (\varphi(\omega) \in Q_j^\beta)\}.$$

Теперь утверждение основной теоремы о единственности решения можно переформулировать следующим образом.

Если квадратичные формы

$$g(x_0, x_1, \dots, x_k, \dots, x_p) = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j,$$

$$h(x_0, x_1, \dots, x_k, \dots, x_p) = \sum_{i,j=0}^p \beta_{i,j} x_i x_j,$$

$$g_k(x_0, x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_p) = \sum_{\substack{i,j=0 \\ i,j \neq k}}^p \alpha_{i,j}^k x_i x_j \quad (k = \overline{1, p}).$$

удовлетворяют условиям

$$\forall l \exists k [S_0^l[g] \simeq g_k] \& \forall k \exists l [g_k \simeq S_0^l[g]]; \quad (5.7.1)$$

$$\forall i \exists j [S_0^i[h] \simeq g_i] \& \forall i \exists j [g_i \simeq S_0^i[h]], \quad (5.7.2)$$

то $g(x_0, x_1, \dots, x_k, \dots, x_p) \simeq h(x_0, x_1, \dots, x_k, \dots, x_p)$, т. е. существует хотя бы одна перестановка ψ p -й степени на множестве N_p , что справедливо равенство

$$g(x_0, x_1, \dots, x_k, \dots, x_p) = \psi(h(x_0, x_1, \dots, x_k, \dots, x_p)).$$

Чтобы доказательство о единственности решения было понятным, изложим его план.

После (леммы 5.7.2) для квадратичных форм g и g_k ($k = \overline{1, p}$), удовлетворяющих условиям (5.7.1); покажем существование таких перестановок ω и δ_k , что $\forall k \{S_0^k[\overline{g}] = \overline{g}_k\}$, где $\overline{g} = \omega(g)$, $\overline{g}_k = \delta_k(g_k)$. Затем (лемма 5.7.3): предположив, что квадратичные формы h и g_k ($k = \overline{1, p}$) удовлетворяют условиям (5.7.2), а квадратичные формы g и g_k ($k = \overline{1, p}$) таковы, что $\forall k \{S_0^k[g] = g_k\}$, покажем существование такой перестановки φ , что $\forall k \{S_0^k[g] \simeq S_0^k[\overline{h}]\}$, где $\overline{h} = \varphi(h)$. Далее, (лемма 5.7.4): при предположении, что квадратичные формы g , h и g_k ($k = 1, 2, \dots, p$) таковы, что $\forall k \{g_k = S_0^k[g] \simeq S_0^k[h]\}$, покажем справедливость утверждения $\exists \psi \forall k \{S_0^k[g] = S_0^k[\psi(h)]\}$,

т. е. существует такая перестановка ψ , что $\forall k \{S_0^k[g] = S_0^k[\overline{h}]\}$, где $\overline{h} = \psi(h)$. Леммой 5.7.5 завершается доказательство теоремы единственности решения, ибо в ней будет показано, что квадратичные формы g и h , удовлетворяющие условию $\forall k \{g_k = S_0^k[g] = S_0^k[h]\}$, равны.

Лемма 5.7.2. Пусть квадратичные формы $g = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$ и

$$g_k = \sum_{\substack{i,j=0 \\ i,j \neq k}}^p \beta_{i,j} x_i x_j \text{ удовлетворяют условиям}$$

$$\forall k \exists l_k [g_k \simeq S_0^{l_k}[g]] \quad (5.7.3)$$

$$(1 \leq l_k \leq p; \quad k = 1, 2, \dots, i, \dots, j, \dots, p; \quad l_i \neq l_j);$$

$$\forall l \exists k_l [S_0^l[g] \simeq g_{k_l}] \quad (5.7.4)$$

$$(1 \leq k_l \leq p; \quad l = 1, 2, \dots, i, \dots, j, \dots, p; \quad k_l \neq k_j).$$

Тогда справедливо утверждение

$$\exists \omega \forall k \exists \delta_k \{S_0^k[\omega[g] = \delta_k[g_k]]\} \Leftrightarrow \forall k \exists l \exists \delta_{k,l} \{S_0^k[g] = \delta_{k,l}[g_l]\},$$

где ω и $\delta_{k,l}$, $\delta_{k,l}$ ($k = \overline{1, p}$) — некоторые перестановки p -й степени на множестве N_p .

Доказательство. Условие (5.7.3) означает, что для каждого k ($1 \leq k \leq p$) существует такая совокупность (непустая) различных индексов $L_k = \{\overline{l_k^s}\}$ (где $s = \overline{1, s_k}$; а s_k — число изоморфных с g_k квадратичных форм из $\{g_k\}$, что справедливо соотношение $g_k \simeq S_0^l[g]$ ($l \in L_k$), или $\forall k \exists L_k \forall s \{(\overline{l_k^s} \in L_k) \Rightarrow [g_k \simeq S_0^{\overline{l_k^s}}[g]]\}$.

Условие (5.7.4) означает, что для каждого l ($1 \leq l \leq p$) существует такая совокупность (непустая) различных индексов $K_l = \{\overline{k_l^r}\}$, где $r = \overline{1, r_l}$, а r_l — число переменных квадратичной формы g , подобных или квазиподобных переменной x_l , что справедливо соотношение $S_0^l[g] \simeq g_k$, ($k \in K_l$), или $\forall l \exists K_l \forall r \{(k_l^r \in K_l) \Rightarrow [S_0^l[g] \simeq g_{k_l^r}]\}$. Причем непосредственно из определений множеств L_k и K_l имеем

$$\forall k \forall s [k \in K_{\overline{l_k^s}}] \quad (k = \overline{1, p}; s = \overline{1, s_k});$$

$$\forall l \forall r [l \in L_{\overline{k_l^r}}] \quad (l = \overline{1, p}; r = \overline{1, r_l});$$

$$\forall k \forall s [s_k = r_{\overline{l_k^s}}] \quad (k = \overline{1, p}; s = \overline{1, s_k});$$

$$\forall l \forall r [r_l = s_{\overline{k_l^r}}] \quad (l = \overline{1, p}; r = \overline{1, r_l});$$

Отсюда следует существование биекций (отображений) $\omega_{l,k} = \binom{l}{k}$ ($l \in \{\overline{l_k^s}\}; k \in \{\overline{k_l^r}\}; 1 \leq l, k \leq p$), удовлетворяющих соотношению

$$\forall k \forall l \forall s \forall r \{(l \in \{\overline{l_k^s}\}) \& (k \in \{\overline{k_l^r}\}) \Rightarrow [S_0^k[\omega_{l,k}[g]] \simeq S_0^l[g]]\},$$

где

$$S_0^k[\omega_{l,k}[g(x_0, x_1, \dots, x_{l-1}, x_l, x_{l+1}, \dots, x_{k-1}, x_k, x_{k+1}, \dots,$$

$$\dots, x_p)] \doteq S_0^k[\overline{g}(x_0, x_1, \dots, x_{l-1}, x_k, x_{l+1}, x, \dots, x_{k-1},$$

$$x_l, x_{k+1}, \dots, x_p)] \doteq \overline{g}(x_0, x_1, \dots, x_{k-1}, x_0, x_{k+1}, \dots,$$

$$\dots, x_p) \simeq g_k.$$

Если теперь по биекциям $\omega_{l,k} = \binom{l}{k}$ ($l, k = \overline{1, p}$) построить перестановку вида

$$\omega = \begin{pmatrix} l_1^1 & l_2^1 & \dots & l_k^1 & \dots & l_p^1 \\ 1 & 2 & \dots & k & \dots & p \end{pmatrix} \quad (1 \leq t_k \leq s_k),$$

где $\forall i \forall j (l_i^i \neq l_j^j)$, то эта перестановка будет удовлетворять соотношениям $S_0^k[[\omega[g]] \simeq g_k]$ ($k = \overline{1, p}$), или $\exists \omega \forall k \{S_0^k[\omega[g]] \simeq g_k\}$ (5.75). Из приведенных соотношений определения изоморфизма квадратичных форм следует, что для каждого значения k ($1 \leq k \leq p$) должна существовать хотя бы одна перестановка δ_k вида

$$\delta_k = \begin{pmatrix} 1 & 2 & \dots & k-1 & k & k+1 & \dots & p \\ j_1 & j_2 & \dots & j_{k-1} & k & j_{k+1} & \dots & j_p \end{pmatrix} \quad (j_m \neq j_n; (1 \leq m, n \leq p)),$$

удовлетворяющая равенству $S_0^k[\omega[g]] = \delta_k[g_k]$, что делает очевидной справедливость утверждения леммы 5.7.2.

Пример А. Пусть g, g_i ($i = \overline{1, 8}$) — квадратичные формы соответственно графов G, G_i ($i = \overline{1, 8}$) (рис. 5.7.1), удовлетворяющих условиям леммы 5.7.2.

Из условий (5.7.3) и (5.7.4) для нашего примера имеем, что перестановка ω , удовлетворяющая соотношению $\forall k \{S_0^k[\omega(g) \simeq g_k]\}$, примет вид

$$\omega = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 6 & 2 & 7 & 8 & 5 & 4 & 1 \end{pmatrix},$$

а граф $\omega[G]$, соответствующий квадратичной форме $\omega(g)$ — (рис. 5.7.2).

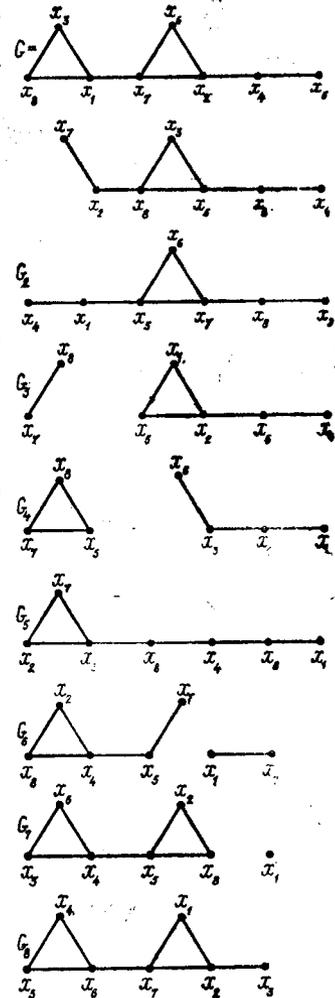


Рис. 5.7.1

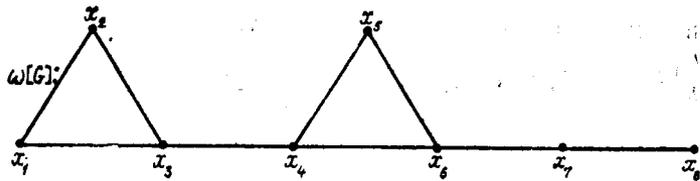


Рис. 5.7.2

Из определения изоморфизма квадратичных форм для изоморфных $S_0^k[\omega(g)]$ и g_k ($k = \overline{1, 8}$) имеем перестановки δ_k ($k = \overline{1, 8}$) вида

$$\delta_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 5 & 8 & 6 & 7 & 2 & 4 \end{pmatrix};$$

$$\delta_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 8 & 1 & 4 & 5 & 6 & 7 \end{pmatrix};$$

$$\delta_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 3 & 8 & 4 & 7 & 1 & 2 \end{pmatrix};$$

$$\delta_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 8 & 6 & 4 & 3 & 5 & 1 & 2 \end{pmatrix};$$

$$\delta_5 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 1 & 3 & 7 & 5 & 8 & 2 & 4 \end{pmatrix};$$

$$\delta_6 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 2 & 5 & 3 & 4 & 6 & 8 & 1 \end{pmatrix};$$

$$\delta_7 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 5 & 1 & 3 & 4 & 2 & 7 & 6 \end{pmatrix};$$

$$\delta_8 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 2 & 1 & 3 & 4 & 8 \end{pmatrix}.$$

откуда имеем, что $\forall k \{S_0^k[\omega(g)] = \bar{g}_k\}$, где $\bar{g}_k = \delta_k(g_k)$.

Лемма 5.7.3. Пусть квадратичные формы g , h и g_k ($k = \overline{1, p}$) удовлетворяют соотношениям

$$\forall k \exists l_k \{S_0^k[h] \simeq g_{l_k}\} \& \forall k \exists l_k \{g_k \simeq S_0^{l_k}[h]\};$$

$$(l \leq l_k = p; k = \overline{1, p}; l_i \neq l_j) \quad (5.7.6)$$

$$\forall k \{S_0^k[g] = g_k\}.$$

Тогда справедливо утверждение

$$\exists \varphi \forall i \{S_0^i[\varphi(h)] \simeq S_0^i[g]\},$$

или $\forall i \{S_0^i[\bar{h}] \simeq S_0^i[g]\}$, где $\bar{h} = \varphi(h)$; φ — некоторая перестановка p -й степени на множестве N_p .

Действительно, так как из условий (5.7.2) следует, что квадратичные формы h и g_k ($k = \overline{1, p}$) удовлетворяют условиям предыдущей леммы, то очевидно, что для них справедлив аналог соотношения (5.7.5), т. е. имеет место соотношение $\exists \varphi \forall i \{S_0^i[\varphi(h)] \simeq g_i\}$, где φ — некоторая перестановка p -й степени на множестве N_p ; откуда, если учесть условия (5.7.6), получим

$$\exists \varphi \forall i \{S_0^i[\varphi(h)] \simeq S_0^i[g]\}.$$

Лемма 5.7.4. Пусть $g = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$ и $h = \sum_{i,j=0}^p \beta_{i,j} x_i x_j$ — квадратичные формы, удовлетворяющие условию

$$\forall i \{g_i \simeq h_i\} \quad (i = \overline{1, p}), \quad (5.7.7)$$

где $g_i = S_0^i[g]$, $h_i = S_0^i[h]$.

Тогда справедливо утверждение

$$\exists \varphi \forall i \{S_0^i[\varphi(h)] = g_i\}, \quad (5.7.8)$$

где φ — некоторая перестановка p -й степени на множестве N_p .

Доказательство. Сначала покажем, что для квадратичных форм g и h , удовлетворяющих условию (5.7.7), справедливо следующее утверждение:

$$\exists i \exists \varphi_i \{[g_i = S_0^i[\varphi_i(h)]] \& \forall j [g_j \simeq S_0^j[\varphi_i(h)]]\}, \quad (5.7.9)$$

где φ_i — некоторая перестановка p -й степени на множестве N_p . Покажем, что каждая перестановка, φ_i , удовлетворяющая соотношению (5.7.9), будет удовлетворять утверждению (5.7.8).

Прежде чем приступить к доказательству справедливости утверждения (5.7.9), заметим, что из изоморфизмов (5.7.7) и следствий 1—5 следует:

$$M_g = M_h = M;$$

$$P_g = P_h = \bigcup_{\alpha} P = P \quad (1 \leq \alpha < p);$$

$$Q_g = Q_h = \bigcup_{\beta} Q = Q \quad (1 \leq \beta < p).$$

Для доказательства справедливости утверждения (5.7.9) рассмотрим следующие три случая: 1) $M \neq \emptyset$; 2) $Q \neq \emptyset$, $M = \emptyset$, $P = \emptyset$; 3) $M = \emptyset$, $P \neq \emptyset$.

В 1-ом случае $M \neq \emptyset$, т. е. квадратичные формы g и h содержат хотя бы по одной одноименной асимметричной переменной. Положив для определенности, что этой асимметричной

переменной является x_u , опишем для $u \in M$ процесс построения хотя бы одной перестановки φ_u p -й степени, удовлетворяющей утверждению (5.7.9).

Пусть $\Phi_u = \{\varphi_u^i\}$ — множество перестановок p -й степени на множестве N_p таких, что $\forall \tau [\varphi_u^i(u) = u]$. Действительно, в силу изоморфизмов (5.7.7) для $i = u$ существует такое непустое множество перестановок $\psi_u = \{\psi_u^t\}$ ($t = \overline{1, m}$; $m \geq 1$), что имеет место

$$\forall \tau \{[\psi_u^t(u) = u] \& [\psi_u^t(h_u) = S_0^u[\psi_u^t(h)] = g_u]\}. \quad (5.7.10)$$

Покажем, что $\Psi_u \cap \Phi_u \neq \emptyset$, т. е. $\exists \tau \forall j [S_0^j[\psi_u(h)] \simeq g_j]$. Предположим обратное, т. е.

$$\neg \{ \exists \tau \forall j [S_0^j[\psi_u(h)] \simeq g_j] \} \Leftrightarrow \forall \tau \exists j [S_0^j[\psi_u^t(h)] \not\simeq g_j]. \quad (5.7.11)$$

Положим для определенности, что $\psi_u^t \in \Psi_u$ ($1 \leq t \leq m$) — некоторая перестановка, имеющая вид

$$\psi_u^t = \begin{pmatrix} 1 & \dots & a = a_i & \dots & u & \dots & a_j & \dots & b_1 & \dots & b = b_k & \dots & p \\ \psi_u^t(1) & \dots & a_j & \dots & u & \dots & b_1 & \dots & b = b_k & \dots & a = a_i & \dots & \psi_u^t(p) \end{pmatrix},$$

для которой в силу нашего предположения справедливо утверждение

$$\begin{aligned} & \exists a \exists b \{ [g_a \not\simeq S_0^a[\psi_u^t(h)]] \& [a = \psi_u^t(b)] \& \\ & \& [u = \psi_u^t(u)] \& [g_u = \psi_u^t(h_u) = S_0^u[\psi_u^t(h)]] \}. \quad (5.7.12) \end{aligned}$$

Очевидно, что для нашей перестановки ψ_u^t так же, как и для любой другой перестановки из множества Ψ_u , удовлетворяющей предположению (5.7.11), имеет место

$$\begin{aligned} & [P_{\psi_u^t(h)}^{a_i} \neq P^{a_i}] \& [P^{a_i} = P^{b_k = \psi_u^{-t}(a_i)}] \vee [Q^{a_i} \neq Q_{\psi_u^t(h)}^{a_i}] \& [Q^{a_i} \neq \\ & \neq Q^{b_k = \psi_u^{-t}(a_i)}], \quad (5.7.13) \end{aligned}$$

ибо в противном случае из утверждений (5.7.12) следует противоречивость нашего предположения (5.7.11).

Итак, предполагаем справедливость утверждений (5.7.12) и (5.7.13). Здесь возможны следующие 3 подслучая

$$a) a_1 \in M_{\psi_u^t(h_u)}; \quad б) a_i \in P_{\psi_u^t(h_u)}^{a_i}; \quad в) a_i \in Q_{\psi_u^t(h_u)}^{a_i}.$$

В подслучае «а» из определения множеств M_{g_u} и M_{h_u} имеем

$$M_{g_u} = M_{h_u} = M_{\psi_u^t(h_u)} = M; \quad a_i, a_j = \psi_u^t(a_i) \in M, \quad (5.7.14)$$

откуда непосредственно следует

$$\psi_u^t(a_i) = a_j = a_i; \quad b_k = \psi_u^{-t}(a_i) = a_i,$$

ибо, в противном случае (при $a_i \neq a_j$), в силу предположения $g_u = \psi_u^t(h_u)$ из равенства $S_0^{a_i}[g_u] = S_0^{a_j}[\psi_u^t(h_u)]$, пришли бы к выводу: $a_i, a_j \notin M$. Это противоречит соотношениям (5.7.14). Следовательно, наше исходное предположение (5.7.11) ложно.

В подслучае «б», очевидно, вследствие нашего предположения (5.7.12) имеем

$$[P_{g_u}^{a_i} \neq P_{\psi_u^t(h_u)}^{a_i}] \& [a_i = \psi_u^t(b_k)],$$

откуда следует, что

$$P_{g_u}^{a_i} = P_{h_u}^{a_i} \text{ либо } P_{g_u}^{a_i} \neq P_{h_u}^{a_i}.$$

В случае, когда $P_{g_u}^{a_i} = P_{h_u}^{a_i}$, в силу тождества $\psi_u^t(h_u) = g_u$ имеем $P_{g_u}^{a_i} = P_{h_u}^{a_i} = P_{\psi_u^t(h_u)}^{a_i}$, т. е.

$$[a_i = \psi_u^t(b_k)] \& [a_i \in P_{g_u}^{a_i} = P_{h_u}^{a_i}] \Rightarrow [a_i, b_k \in P_{g_u}^a = P_{h_u}^a = P_{\psi_u^t(h_u)}^a],$$

откуда очевидна справедливость отношения

$$S_0^{a_i}[\psi_u^t(h)] \simeq g_{a_i},$$

что противоречит исходному предположению (5.7.12).

Наконец, с целью проведения анализа случая, когда $P_{g_u}^{a_i} \neq P_{h_u}^{a_i}$, рассмотрим множества P_{g_u} и P_{h_u} более детально. Пусть $P_{g_u}^{a_i}, P_{g_u}^{b_k} \in P_{g_u}$ и $P_{h_u}^{\alpha_m}, P_{h_u}^{\alpha_n} \in P_{h_u}$ — некоторые классы подобия, удовлетворяющие условиям

$$[P_{g_u}^{a_i} \subseteq P_{\psi_u^t(h_u)}^{\alpha_m}] \& [P_{g_u}^{a_i} \neq P_{h_u}^{\alpha_m}];$$

$$[P_{g_u}^{b_k} \subseteq P_{\psi_u^t(h_u)}^{\alpha_n}] \& [P_{g_u}^{b_k} \neq P_{h_u}^{\alpha_n}].$$

Представим эти классы в виде

$$P_{g_u}^{a_i} = \{a_1, \dots, a_i, \dots, a_j, \dots, a_L\};$$

$$P_{g_u}^{b_k} = \{b_1, \dots, b_k, \dots, b_l, \dots, b_R\};$$

$$P_{h_u}^{\alpha_m} = \{\alpha_1, \dots, \alpha_m = b_k, \dots, \alpha_n = a_i, \dots, \alpha_L\};$$

$$P_{h_u}^{\alpha_n} = \{\beta_1, \dots, \beta_r = b_l, \dots, \beta_s = a_j, \dots, \beta_R\},$$

где

$$[a_j \notin P_{h_u}^{\alpha_m}] \& [b_k \notin P_{h_u}^{\alpha_n}].$$

Тогда в силу тождества $g_u = \psi_u^t(h_u)$ имеем

$$P_{g_u}^{a_i} = \psi_u^t [P_{h_u}^{a_i}] = (a_1, \dots, a_i, \dots, a_j, \dots, a_L);$$

$$P_{g_u}^{b_k} = \psi_u^t [P_{h_u}^{b_k}] = (b_1, \dots, b_k, \dots, b_l, \dots, b_R);$$

где

$$a_i = \psi_u^t(\alpha_m = b_k), \quad a_j = \psi_u^t(\alpha_n = a_i),$$

$$b_k = \psi_u^t(\beta_r = b_l), \quad b_l = \psi_u^t(\beta_s = a_j),$$

причем в силу нашего предположения имеем

$$\{g_{a_i} \not\approx S_0^{a_i}[\psi_u^t(h)]\} \& \{g_{a_j} \not\approx S_0^{a_j}[\psi_u^t(h)]\}.$$

Пусть $E_{g_u} = \{e_k \mid (k = \bar{0}, s)\}$ — группа автоморфизмов квадратичной формы $g_u = \psi_u^t(h_u)$, e_0 — тождественный автоморфизм (перестановка) p -й степени на множество N_p , т. е. имеет место

$$\forall k [g_u = e_k(g_u = \psi_u^t(h_u))].$$

Следовательно, существует ровно $s + 1$ перестановка

$$\{\psi_u^k\} \quad (k = \bar{0}, s), \quad \text{что } \forall k [g_u = \psi_u^k(h_u)],$$

где $\psi_u^k = \psi_u^t e_k$ ($0 \leq k \leq s$). Отсюда в силу определения классов подобия

$P_{g_u}^{a_i}$ и $P_{g_u}^{b_k}$ следует существование автоморфизма e_p вида

$$e_p = \begin{pmatrix} \dots a_i \dots a_j \dots u \dots b_k \dots a_i \\ \dots a_j \dots a_i \dots u \dots b_l \dots b_k \end{pmatrix},$$

для которых справедливы следующие соотношения:

$$e_p(\psi_u^t(\alpha_n)) = e_p(\psi_u^t(a_i)) = e_p(a_j) = a_i;$$

$$e_p(\psi_u^t(\alpha_m)) = e_p(\psi_u^t(b_k)) = e_p(a_i) = a_j;$$

$$e_p(\psi_u^t(\beta_s)) = e_p(\psi_u^t(a_j)) = e_p(b_l) = b_k;$$

$$e_p(\psi_u^t(\beta_r)) = e_p(\psi_u^t(b_l)) = e_p(b_k) = b_l;$$

т. е. для перестановки $\psi_u = \psi_u^t e_p$ имеет место

$$\psi_u(a_i) = a_i; \quad \psi_u(a_j) = b_k;$$

$$\psi_u(b_k) = a_j; \quad \psi_u(b_l) = b_l.$$

Отсюда в силу предположения относительно перестановки и свойств автоморфизма e_p имеем

$$[g_u = S_0^u[\psi_u(h)]] \& [g_i \simeq S_0^{a_i = \psi_u(a_j)}[\psi_u(h)]] \&$$

$$\& [g_{a_j} \simeq S_0^{a_j = \psi_u(b_k)}[\psi_u(h)]],$$

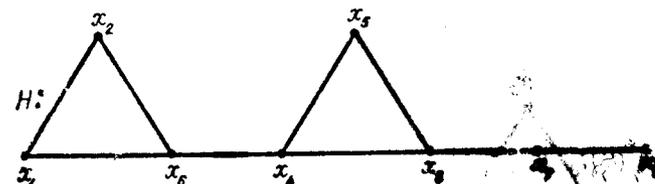
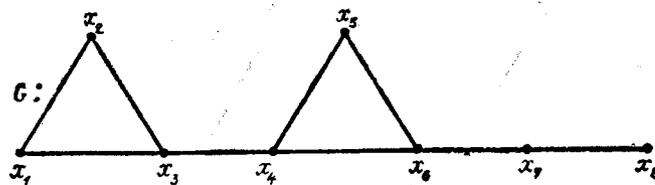


Рис. 5.7.3

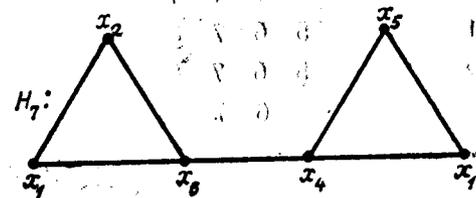
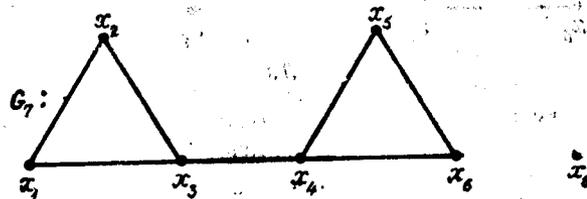


Рис. 5.7.4

$$(a_j \in P_{h_u}^{b_k}) \& (b_k \in P_{g_u}^{a_i}) \& (P_{h_u}^{b_k} \subseteq P_{g_u}^{a_i}),$$

что противоречит нашему предположению (5.7.11).

Поясним рассмотренный случай на следующем примере.

Пример В. Пусть g и h — квадратичные формы графов G и H , изображенных на рис. 5.7.3. Если положить $U = 7 \in M_g = M_h$, то квадратичные формы $g_7 = S_0^7[g]$ и $h_7 = S_0^7[h]$, соответствующие квадратичным формам $G_7 = G + x_7$ и $H_7 = H + x_7$, которые изображены на рис. 5.7.4, примут вид $g_7 = x_0x_1 + x_0x_2 + x_0x_3 + x_0x_4 + x_0x_5 + x_0x_6 + x_0x_8 + x_1x_2 + x_2x_3 + x_1x_3 + x_2x_4 + x_4x_5 + x_5x_6 + x_4x_6 + x_2x_8 + x_1x_8 + x_0x_3 + x_0x_4 + x_0x_5 + x_0x_6 + x_0x_8 + x_1x_2 + x_2x_6 + x_1x_6 + x_6x_4 + x_4x_5 + x_5x_3 + x_5x_4$.

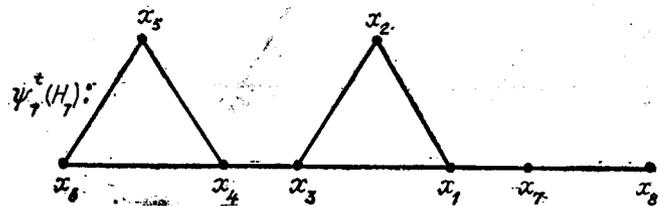
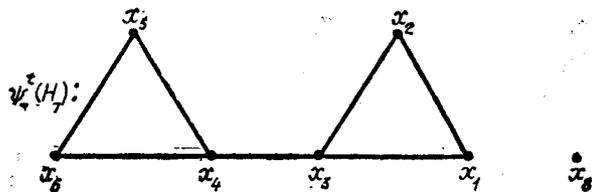


Рис. 5.7.5

Построим для данного примера множества E_{g_7} , Φ_7 и Ψ_7 . Элементы группы автоморфизма E_{g_7} имеют вид

$$\begin{aligned}
 e_0 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}; \\
 e_1 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 1 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} = (1, 2); \\
 e_2 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 6 & 5 & 7 & 8 \end{pmatrix} = (5, 6); \\
 e_4 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 1 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} = (1, 2)(5, 6); \\
 e_5 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 4 & 3 & 1 & 2 & 7 & 8 \end{pmatrix} = (1, 5)(2, 6)(3, 4); \\
 e_6 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 5 & 4 & 3 & 2 & 1 & 7 & 8 \end{pmatrix} = (1, 6)(2, 5)(3, 4); \\
 e_7 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 5 & 4 & 3 & 1 & 2 & 7 & 8 \end{pmatrix} = (1, 6, 2, 5)(3, 4); \\
 e_8 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 4 & 3 & 2 & 1 & 7 & 8 \end{pmatrix} = (1, 5, 2, 6)(3, 4).
 \end{aligned}$$

В состав множества Φ_7 , очевидно, кроме элементов $\psi_{ik}^k = e_k$ ($k = 0, 8$) множества E_{g_7} входят еще следующие перестанов-

$$\Phi_9 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 6 & 5 & 4 & 7 & 8 \end{pmatrix} = (4, 6);$$

$$\Phi_{10} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 1 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} = (1, 3);$$

$$\Phi_{11} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 5 & 3 & 4 & 2 & 6 & 7 & 8 \end{pmatrix} = (2, 5);$$

В качестве элемента множества Ψ_7 , рассмотрим перестановку

$$\Psi_7^t = \begin{pmatrix} a_i & b_k & b_l & a_j \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 5 & 1 & 3 & 2 & 4 & 7 & 8 \\ a_l & a_i & b_k & b_l \end{pmatrix} = (1, 6, 4, 3)(2, 5);$$

результат действия которой на квадратичные формы h_7 и h обозначим соответственно через $\psi_7^t(h_7)$ и $\psi_7^t(h)$. Графы, соответствующие этим квадратичным формам, примут вид, изображенный на рис. 5.7.5.

Очевидно, что $\psi_7^t(h_7) = S_0^7[\psi_7^t(h)] = g_7$, и для $a_i = 1$ справедливо отношение (5.7.11), т. е.

$$g_1 \neq S_0^1[\psi_7^t(h)].$$

По квадратичным формам g_7 , h_7 и $\psi_7^t(h_7)$ определяем классы

$$\begin{aligned}
 P_{g_7}^{a_i} &= \{1, 2, 5, 6\}; & P_{g_7}^{b_k} &= \{3, 4\}; \\
 P_{h_7}^{a_i} &= \{1, 2, 3, 4, 5\}; & P_{h_7}^{b_k} &= \{4, 6\};
 \end{aligned}$$

В силу тождества $g_7 = \psi_7^t(h_7)$ имеем

$$\begin{aligned}
 \psi_7^t(a_i) &= \psi_7^t(1) = 6 = a_j; & \psi_7^t(a_j) &= \psi_7^t(6) = (4) = b_l; \\
 \psi_7^t(b_l) &= \psi_7^t(4) = 3 = b_k; & \psi_7^t(b_k) &= \psi_7^t(3) = 1 = b_i; \\
 \psi_7^t(7) &= 7 \text{ и т. д.}
 \end{aligned}$$

Классы $\psi_7^t(P_{h_7})$ и $\psi_7^t(P_{g_7}^b)$ примут соответственно вид

$$\psi_7^t(P_{h_7}) = P_{g_7}^{a_i=1} = \{1, 2, 5, 6\}, \quad \psi_7^t(P_{g_7}^b) = P_{g_7}^{b_k=3} = \{3, 4\}.$$

Если искомому автоморфизму ε_p квадратичной формы $\psi(h_7) = g_7$ наложить ограничения

$$\varepsilon_p(6) = \varepsilon_p(\psi^t(1)) = 1; \quad \varepsilon_p(1) = \varepsilon_p(\psi^t(3)) = 6;$$

$$\varepsilon_p(4) = \varepsilon_p(\psi^t(6)) = 3; \quad \varepsilon_p(3) = \varepsilon_p(\psi^t(4)) = 4,$$

то, очевидно, что в качестве ε_p необходимо взять автоморфизм

$$\varepsilon_p \text{ вида } \varepsilon_p = \varepsilon_6 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 5 & 4 & 3 & 2 & 1 & 7 & 8 \end{pmatrix} = (1, 6)(2, 5)(3, 4),$$

который удовлетворяет соотношению $\psi^t \varepsilon_6 = \psi^t \in \Psi_7$.

Если вместо перестановки ψ^t взять перестановку ψ^z вида

$$\psi^z = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 6 & 4 & 5 & 3 & 7 & 8 \end{pmatrix} = (3, 6),$$

то становится очевидной для нее справедливость утверждения

$$[g_7 = S_0^7[\psi^z(h)]] \& \forall i \{g_i \simeq S_0^i[\psi^z(h)]\}.$$

Остается рассмотреть подслучай «в», когда $a_i \in Q_{\psi_u^t(h)=g_u}$.

Здесь также, как и в подслучае «а» очевидно, что

$$Q_{h_u}^{a_i} = Q_{\psi_u^t(h_u)=g_u}^{a_i}, \quad \psi_u^t(a_i) = a_i = a_i \in Q_{h_u}^{a_i}, \quad b_k = \psi_u^t(a_i) = a_i \in Q_{h_u}^{a_i},$$

ибо в противном случае (при $a_i \neq a_i$ и $\psi_u^t \neq \varepsilon_0$) из предположения $g_u = \psi_u^t(h_u)$ следовало бы существование некоторого автоморфизма ε_k квадратичной формы $\psi_u^t(h_u)$, для которого справедливо соотношение

$$[\psi_u^t(h_u) = \varepsilon(\psi_u^t(h_u))] \& [a_i = \varepsilon(a_i)],$$

что противоречит определению класса квазиподобия $Q_{h_u}^{a_i}$. Следовательно, приходим к выводу о ложности нашего исходного предположения (5.7.10). Доказательство для 1-го случая завершено.

Рассмотрим 2-й случай, когда $M = \emptyset, P = \emptyset$, а $Q \neq \emptyset$.

$$\text{Пусть } Q = \bigcup_{\xi=1}^M Q_\xi; \quad Q_\xi = \{u_{\xi,1}, u_{\xi,2}, \dots, u_{\xi,M_\xi}\},$$

$M \geq 2$, ибо при $M = 1$ утверждение леммы очевидно.

Из определения классов квазиподобия имеем

$$\forall i \forall j \{(i, j \in Q_\xi) \Rightarrow [g_i \simeq h_i \simeq h_j \simeq g_j]\},$$

в частности,

$$\forall \xi \forall i \forall j \{g_{u_{\xi,i}} \simeq S_0^{u_{\xi,j}}[h]\}.$$

Положим, что $\Phi = \{\varphi^t\}$ ($t = \overline{1, T}; T = \prod_{\xi=1}^M (M_\xi!)$) — множество перестановок p -й степени на множестве N_p таких, что

$$\forall \tau \forall i \{g_i \simeq S_0^i[\varphi^t(p)]\} \& \forall \tau \forall \xi \forall i \{(i \in Q_\xi) \Rightarrow [\varphi^t(i) \in Q_\xi]\}.$$

Покажем, что среди перестановок существует хотя бы одна перестановка φ^t ($1 \leq t \leq T$) такая, что для некоторого $u \in Q_\alpha$ ($1 \leq \alpha \leq M$) справедливо $g_{u_{\alpha,a}} = S_0^{u_{\alpha,a}}[\varphi^t(h)]$.

Действительно, из определения изоморфизма и класса квазиподобия имеем

$$\forall \xi \forall i \forall j \exists \psi \{[g_{u_{\xi,i}} = \varphi[S_0^{u_{\xi,j}}(h)] = S_0^{u_{\xi,j}}[\psi(h)] \& [\psi(u_{\xi,i}) = u_{\xi,i}]\}.$$

Для значения $\xi = \alpha$ обозначим через $\Psi_\alpha = \{\psi_{\alpha,t,i}\}$ ($1 \leq \alpha \leq M; 1 \leq i \leq T_\alpha; j = \overline{1, T_\alpha}$) — множество перестановок таких, что

$$\forall j \{[g_{u_{\alpha,i}} = \psi_{\alpha,t,i}[S_0^{u_{\alpha,t}}(h)] = S_0^{u_{\alpha,t}}[\psi_{\alpha,t,i}(h)] \& [u_{\alpha,i} = \psi_{\alpha,t,i}(u_{\alpha,i})]\}.$$

Откуда, положив для определенности $i = a, j = b$ ($1 \leq a, b \leq T_\alpha$), имеем

$$[g_{u_{\alpha,a}} = S_0^{u_{\alpha,a}}[\psi_{\alpha,a,b}(h)] \& [u_{\alpha,a} = \psi_{\alpha,a,b}(u_{\alpha,b})].$$

Допустим, что перестановка $\psi_{\alpha,a,b}$ такая, что она не удовлетворяет условию

$$\forall i \{g_i \simeq S_0^i[\psi_{\alpha,a,b}(h)]\}, \quad (5.7.14)$$

т. е. существует некоторая переменная $u_{\beta,o}$ такая, что

$$g_{u_{\beta,o}} \not\simeq S_0^{u_{\beta,o}}[\bar{h}],$$

где $\bar{h} = \psi_{\alpha,a,b}(h)$, иначе классы квазиподобия $Q_{\bar{h}}^{u_{\beta,o}}$ и $Q_h^{u_{\beta,o}}$ различны.

Займемся построением искомой перестановки $\psi_{\alpha,a,b}$ ($1 \leq \alpha \leq M; 1 \leq a, b \leq T_\alpha$), удовлетворяющей условиям (5.7.14). Предварительно покажем, что для каждой пары квадратичных форм $g_{u_{\alpha,a}}$ и $h_{u_{\alpha,b}}$ существует единственная (с точностью до автоморфизма

мов этих квадратичных форм) перестановка такая, что

$$[g_{u_{\alpha,a}} = \psi(h_{u_{\alpha,b}}) = S_0^{u_{\alpha,a}}[\psi(h)] \& [u_{\alpha,a} = (u_{\alpha,b})]. \quad (5.7.15)$$

Допустим, что это не так, т. е. существует по крайней мере две различные перестановки ψ_1 и ψ_2 , удовлетворяющие соотношениям (5.7.15). Здесь в силу возможных автоморфизмов у квадратичной формы $h_{u_{\alpha,b}}$ из-за возможного подобия некоторых переменных $u \in Q^u$ и $v \in Q^v$ полагаем для определенности, не умаляя общности $u, \psi_1(u), \psi_2(u) \in Q^u$ и $v, \psi_1(v), \psi_2(v) \in Q^v$.

Тогда для перестановок ψ_1 и ψ_2 справедливы равенства

$$g_{u_{\alpha,a}} = \psi_1(h_{u_{\alpha,b}}); \quad g_{u_{\alpha,a}} = \psi_2(h_{u_{\alpha,b}}),$$

т. е. имеем

$$g_{u_{\alpha,a}} = \psi_1(h_{u_{\alpha,b}}) = \psi_2(h_{u_{\alpha,b}}) \simeq h_{u_{\alpha,b}}.$$

Откуда следует, что

$$\begin{aligned} \psi_1^{-1}\psi_2(h_{u_{\alpha,b}}) &= \psi_1^{-1}\psi_2(h_{u_{\alpha,b}}) = h_{u_{\alpha,b}}; \\ \psi_2^{-1}\psi_1(h_{u_{\alpha,b}}) &= \psi_2^{-1}\psi_1(h_{u_{\alpha,b}}) = h_{u_{\alpha,b}}. \end{aligned}$$

а это возможно только при условии, что перестановка $\omega_1 = \psi_1^{-1}\psi_2$ и $\omega_2 = \psi_2^{-1}\psi_1$ является автоморфизмами квадратичной формы $h_{u_{\alpha,b}}$, что противоречит процессу построения ψ_1 и ψ_2 .

Следовательно, если положить

$$\psi_{\alpha,a,i}(h) = \sum_{k,l=0}^p \beta_{k,l}^i x_k x_l,$$

то очевидно, что $\forall j_1 \forall j_2 [\psi_{\alpha,a,i_1}(h) \neq \psi_{\alpha,a,i_2}(h)]$, т. е. $\forall k \forall i \forall j_1 \forall j_2 \exists k [\beta_{k,u_{\alpha,i_1}}^i \neq \beta_{k,u_{\alpha,i_2}}^i]$.

С другой стороны, если для некоторой квадратичной формы $h_{u_{\alpha,i}}$ построить множество перестановок $\bar{\Psi}_{\alpha,i} = \{\bar{\psi}_{\alpha,i,j}\}$ ($j = \bar{1}, \bar{T}_\alpha$) аналогично построению множества $\Psi_{\alpha,i}$, то будем иметь

$$\begin{aligned} h_{\alpha,i} &= S_0^{u_{\alpha,i}}[\bar{\psi}_{\alpha,i,1}(h)] = S_0^{u_{\alpha,i}}[\bar{\psi}_{\alpha,i,2}(h)] = \dots = \\ &= S_0^{u_{\alpha,i}}[\bar{\psi}_{\alpha,i,j}(h)] = \dots = S_0^{u_{\alpha,i}}[\bar{\psi}_{\alpha,i,T_\alpha}(h)], \end{aligned}$$

где

$$\bar{\psi}_{\alpha,i,j}(h) = \sum_{k,l=0}^p \bar{\beta}_{k,l}^j x_k x_l, \quad p = 3, \text{ например}$$

причем

$$\forall j_1 \forall j_2 \exists k [\bar{\beta}_{k,u_{\alpha,i_1}}^j \neq \bar{\beta}_{k,u_{\alpha,i_2}}^j]$$

а $\bar{\psi}_{\alpha,i,j}$ — единственная перестановка, удовлетворяющая соотношению (5.7.15)

Таким образом, становится очевидным, что каждая квази-подобная переменная $u_{\xi,i}$ квадратичных форм g и h уникальна в своем роде. Поэтому для каждого $\xi = \alpha$ и каждого $i = a$ будет существовать только единственное значение $j = b$, что

$$\forall k \{ \alpha_{k,u_{\alpha,a}} = \beta_{k,u_{\alpha,b}}^b \}, \quad (5.7.16)$$

иначе если совокупность перестановок $\{\psi_{\alpha,i,j}\}$ рассматривать как матрицу, то в этой матрице одна и та же перестановка будет встречаться только один раз в каждой строке и в каждом столбце, а именно перестановка $\psi_{\alpha,i,j}$, для которой справедливы равенства (5.7.3).

Вывод. Для построения искомой перестановки $\psi_{\alpha,i,l}$ достаточно определить такое взаимно однозначное соответствие между переменными с индексами $u_{\alpha,i}$ и их прообразами $\psi_{\alpha,i,l}^{-1}(u_{\alpha,i})$, определяющее эту пару некоторым уникальным только для этих переменных свойством, а именно соотношением (5.6.16) или, иначе, для построения искомой перестановки $\psi_{\alpha,i,l}$ достаточно использовать соотношение

$$g_{u_{\alpha,i}} + x_0 x_{p+1} + x_{p+1} x_{u_{\alpha,i}} \simeq h_{u_{\alpha,i}} + x_0 x_{p+1} + x_{p+1} x_{u_{\alpha,i}}, \quad (5.7.17)$$

где $u_{\alpha,i} = \psi_{\alpha,i,l}^{-1}(u_{\alpha,i})$, т. е. искомая перестановка однозначно определяется из тождества $g_{u_{\alpha,i}} = \psi_{\alpha,i,l}[h_{\alpha,i,l}]$.

Пример С. Пусть g и h — квадратичные формы соответственно графов G и H , изображены на рис. 5.7.6. Эти формы удовлетворяют условиям

$$\forall i \{ S_0^i[g] = g_i \}; \quad \forall i \{ S_0^i[h] = h_i \}; \quad \forall i \{ g_i \simeq h_i \},$$

причем $Q_\alpha = \{3, 12\}$; $Q_\beta = \{6, 9\}$, а g_3, g_{12}, h_3, h_{12} — квадратичные формы соответственно графов G_3, H_3 и H_{12} , изображенных на рис. 5.7.7.

Положив для определенности $u = 3 \in Q_\alpha$, имеем

$$\begin{aligned} \psi_{3,3} &= \\ &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 4 & 5 & 3 & 7 & 8 & 12 & 10 & 11 & 9 & 13 & 14 & 6 & 1 & 2 \end{pmatrix}; \end{aligned}$$

а квадратичной форме $\psi_{3,3}(h)$ будет соответствовать граф $\psi_{3,3}(H)$.

Очевидно, что в качестве $u_{\alpha,a}$ можно взять 6, т. е.

$$g_6 \neq S_0^6[\psi_{3,3}(h)], \quad \text{т. е. } Q_\alpha^6 \neq Q_{\psi_{3,3}}^6(h), \quad 6 \in Q_\beta,$$

$$\psi_{3,3}^{-1}(6) = 12 \in Q_\alpha,$$

иначе, индексы 6 и 12 принадлежат различным классам квази-подобия.

Из рис. 5.7.8 по квадратичной форме $g_{u_{\alpha,3}}$ определяем квадратичную форму $h_{\alpha,12}$, удовлетворяющую соотношению (5.7.17).

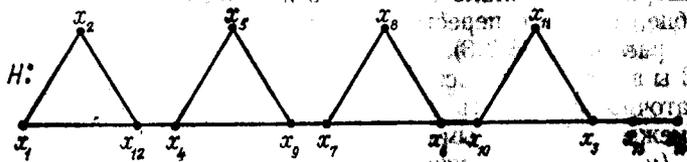
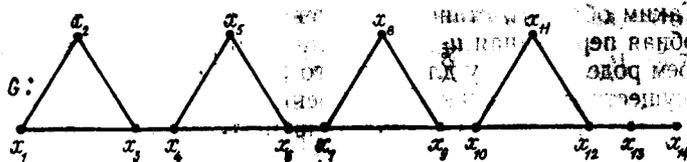


Рис. 5.7.6

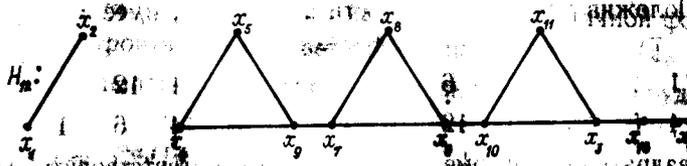
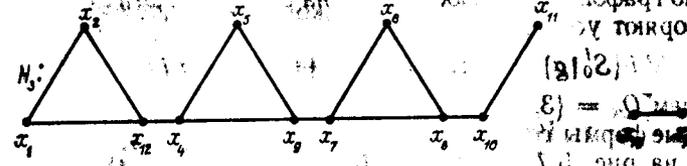
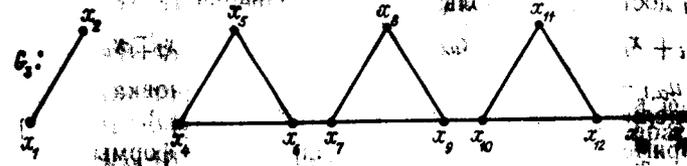


Рис. 5.7.7

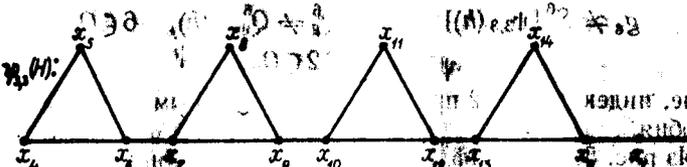


Рис. 5.7.8

Наконец, по тождеству $g_{u,\alpha,3} = \psi_{\alpha,3,12} [h_{u,\alpha,12}]$ определяем искомую перестановку

$$\psi_{\alpha,3,12} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 1 & 3 & 12 & 4 & 5 & 9 & 7 & 8 & 6 & 10 & 11 & 3 & 13 & 14 \end{pmatrix} = (3, 12)(6, 9),$$

которая удовлетворяет соотношениям

$$\forall i \{g_i \cong S_0^i[\psi_{\alpha,3,12}(h)]\} \& [g_3 = S_0^3[\psi_{\alpha,3,12}(h)]]$$

Рассмотрим 3-й случай, когда $M = \emptyset$, т. е. $(u \in P) \vee \vee (u \in Q)$. Здесь необходимо рассмотреть следующие три подслучая:

- а) $(u \in P_{g_u}^u) \& (u \notin Q_u)$; б) $(u \in P_{g_u}^u) \& (u \in Q_u^u)$; в) $(u \in Q) \& (u \notin P_g)$.

В подслучае «а» так же, как и в первом случае, в силу изоморфизмов (5.7.7) можно утверждать о существовании непустого множества перестановок $\Psi_u = \{\psi^u\}$, что

$$\forall \tau \{[\psi^\tau(u) = u] \& [\psi_u^\tau(h_u) = S_0^u[\psi_u^\tau(h)] = g_u]\}.$$

Покажем, что $\Psi_u \cap \Phi_u \neq \emptyset$, т. е. существует перестановка $\psi_u^i \in \Psi_u$, для которой справедливо соотношение

$$[\psi_u^i(u) = u] \& [\psi_u^i(h_u) = S_0^u[\psi_u^i(h)] = g_u] \& i \{g_i S_0^i[\psi_u^i(h)]\}.$$

Действительно. Предположим противное:

$$\neg [\exists \tau \forall i \{(\psi_u^i \in \Psi_u) \& [g_i \cong S_0^i[\psi_u^i(h)]]\}] \Leftrightarrow \Leftrightarrow \forall \tau \exists i \{[\psi_u^i \in \Psi_u] \& [g_i \not\cong S_0^i[\psi_u^i(h)]]\},$$

т. е. какая бы ни была перестановка $\psi_u^i \in \Psi_u$, для нее справедливо

$$\neg \exists v \exists \omega \{[g_u = S_0^u[\psi_u^i(h)]] \& [g_v \not\cong S_0^v[\psi_u^i(h)]] \& [v = \psi_u^i(\omega)]\}. \tag{5.7.18}$$

Здесь в свою очередь возможны варианты:

- $(\omega \in P_h^w = P_g^w) \& (v = \psi_u^i(\omega)); v \in P_{\psi_u^i(h)}^v \& (P_{\psi_u^i(h)}^v = P_h^w = P_g^w);$
- $(\omega \in P_h^w = P_g^w) \& (v = \psi_u^i(\omega), v \in P_{\psi_u^i(h)}^v) \& (P_{\psi_u^i(h)}^v \neq P_h^w = P_g^w).$

В первом варианте очевидна ложность предположения (5.7.18), ибо из определения классов подобия для классов P_g^w ,

$P_h^w \cap P_{\psi_u^t(h)}^v$ имеем $e_p \psi_u^t = e_p \psi_u^t$ удовлетворяют условиям (5.7.18)

$$(\omega \in P_h^w) \& (v \in P_{\psi_u^t(h)}^v) \& (P_{\psi_u^t(h)}^v \neq P_h^w = P_g^w) \Rightarrow$$

$$\Rightarrow [g_0 \simeq S_0^v = S_0^{\psi_u^t(h)} [\psi_u^t(h)]]$$

Проанализируем второй вариант. Пусть $\omega \in P_h^w = P_g^w$; $v = \psi_u^t(\omega)$; $v \in P_{\psi_u^t(h)}^v$, причем $P_{\psi_u^t(h)}^v = P_g^w = P_h^w$, т.е. перестановка ψ_u^t и классы $P_{\psi_u^t(h)}^v$, P_h^w имеют соответственно вид

$$P_{\psi_u^t(h)}^v = \{v_1, v_2, \dots, v_l = v = \psi_u^t(\omega), \dots, v_L\}$$

$$P_h^w = \{\omega_1, \omega_2, \dots, \omega_l = \omega, \dots, \omega_L\}$$

Рассуждения, аналогичные тем, которые проводились в предыдущем случае ($M \neq \emptyset$), позволяют по перестановке подобрать автоморфизм e_p квадратичной формы g_u вида

$$e_p = \begin{pmatrix} 1 & 2 & \dots & \psi_u^t(v) & \dots & u & \dots & p \\ e_p(1) & e_p(2) & \dots & v & \dots & u & \dots & e_p(p) \end{pmatrix}$$

Очевидно, что для перестановки $\psi_u = \psi_u^t e_p$ имеют место

$$\psi_u(v) = \psi_u^t e_p(v) = e_p(\psi_u^t(v)) = v,$$

откуда следует $[g_u = S_0^v [\psi_u(h)]] \& [g_0 \simeq S_0^v = S_0^{\psi_u^t(h)} [\psi_u^t(h)]]$, что не совместимо с нашим предположением (5.7.18).

Пример D. Пусть h и g — квадратичные формы соответственно графов H и G , изображенных на рис. 5.7.9 и удовлетворяющих условиям $\forall i [g_i \simeq S_0^i [h]]$.

Положив $u = 5$, имеем (рис. 5.7.10). Откуда следует, что если в качестве перестановки ψ_5^t взять перестановку вида

$$\psi_5^t = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 8 & 1 & 5 & 4 & 7 & 6 \end{pmatrix}$$

не трудно убедиться в справедливости соотношения $g_5 = \psi_5^t(h_5) = S_0^5 [\psi_5^t(h)]$,

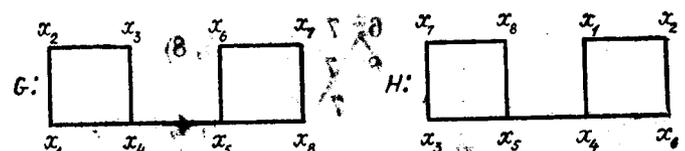


Рис. 5.7.9

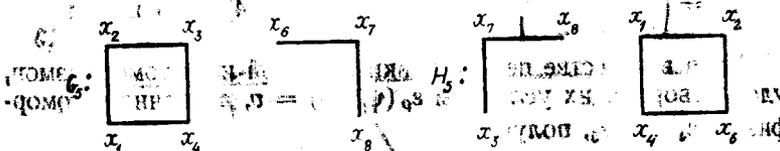


Рис. 5.7.10

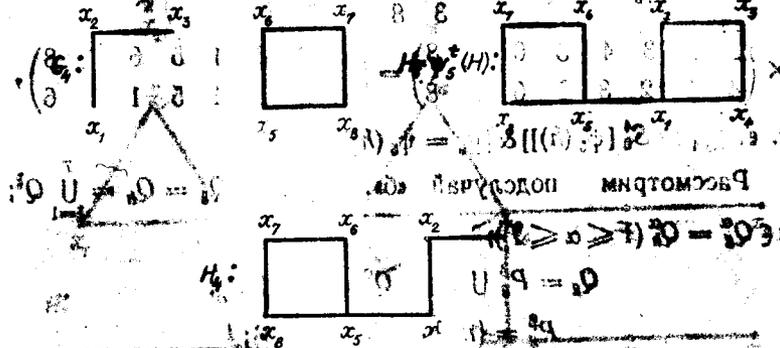


Рис. 5.7.11

причем $g_4 \neq S_0^4 [\psi_5^t(h)]$ (рис. 5.7.11), т.е. для рассматриваемого примера имеем

$$v = 4; \quad \omega = 6; \quad 4 = \psi_5^t(6); \quad \psi_5^t(4) = 1;$$

$$P_{h_5}^{v=4} = \{1, 2, v = 4, \omega = 6\};$$

$$P_{g_5}^{w=6} = P_{h_5}^{v=4} = \{\psi_5^t(4) = 1, 2, 3; v = 4\}$$

Группа автоморфизмов $\{e_k\}$ квадратичной формы g_5 имеет

$$e_0 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

$$e_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 1 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} = (1, 3);$$

$$e_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 4 & 3 & 2 & 5 & 6 & 7 & 8 \end{pmatrix} = (2, 4);$$

$$e_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 8 & 7 & 6 \end{pmatrix} = (6, 8);$$

$$e_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{pmatrix} = (1, 4, 3, 2);$$

$$e_5 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 & 5 & 6 & 7 & 8 \end{pmatrix} = (1, 4)(2, 3) \text{ и т. д.}$$

Взяв в качестве перестановки e_p любой из автоморфизмов, удовлетворяющих условиям $e_p(\psi_5^i(v)) = v$, а именно автоморфизмы e_4 или e_5 , получим

$$\psi_5 = \psi_5^i e_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 8 & 1 & 5 & 4 & 7 & 6 \end{pmatrix} \times$$

$$\times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 8 & 4 & 5 & 1 & 7 & 6 \end{pmatrix},$$

т. е. $[g_4 \simeq S_0^4[\psi_5(h)]] \& [g_5 = \psi_5(h_5)]$.

Рассмотрим подслучай «б». Пусть $Q_g = Q_h = \bigcup_{i=1}^T Q_i^{\alpha}$;

$$u \in Q_g^{\alpha} = Q_h^{\alpha} (1 \leq \alpha \leq T);$$

ит. $Q_g = P_g^{\beta} \cup P_g; \quad Q_h^{\alpha} = P_h^{\beta} \cup P_h^{\gamma};$

$$P_g^{\beta} = \{u_{\beta, i}\}; \quad P_g^{\gamma} = \{u_{\gamma, i}\};$$

$$P_h^{\beta} = \{u_{\beta, i}\}; \quad P_h^{\gamma} = \{u_{\gamma, i}\};$$

причём в силу того, что элемент α , принадлежащий некоторому классу квазиподобия Q_g^{α} , принадлежит также некоторому классу подобия $P_g^{\beta} \subset Q_g^{\alpha}$, то, очевидно, все элементы множества P_g^{β} будут обладать одним и тем же уникальным свойством « β », о котором говорилось в предыдущем случае.

В дальнейшем для краткости будем считать, что элементы множеств P_g^{β} и P_h^{β} обладают одним и тем же свойством « β ».

В зависимости от того, каким подмножествам $P_g^{\beta}, P_h^{\beta}, P_g^{\gamma}, P_h^{\gamma}$ множества Q^{α} принадлежит элемент u , возможны варианты

$$(u \in P_g^{\beta}) \& (u \in P_h^{\beta}); \quad (u \in P_g^{\beta}) \& (u \in P_h^{\gamma}).$$

В первом варианте существование такой перестановки ψ , что

$$P_g^{\beta} = \psi(P_h^{\beta}) = P_{\psi(h)}^{\beta},$$

т. е. множества P_g^{β} и $P_{\psi(h)}^{\beta}$ совпадают, а их элементы в силу (5.7.17) обладают одним и тем же свойством « β », следовательно-

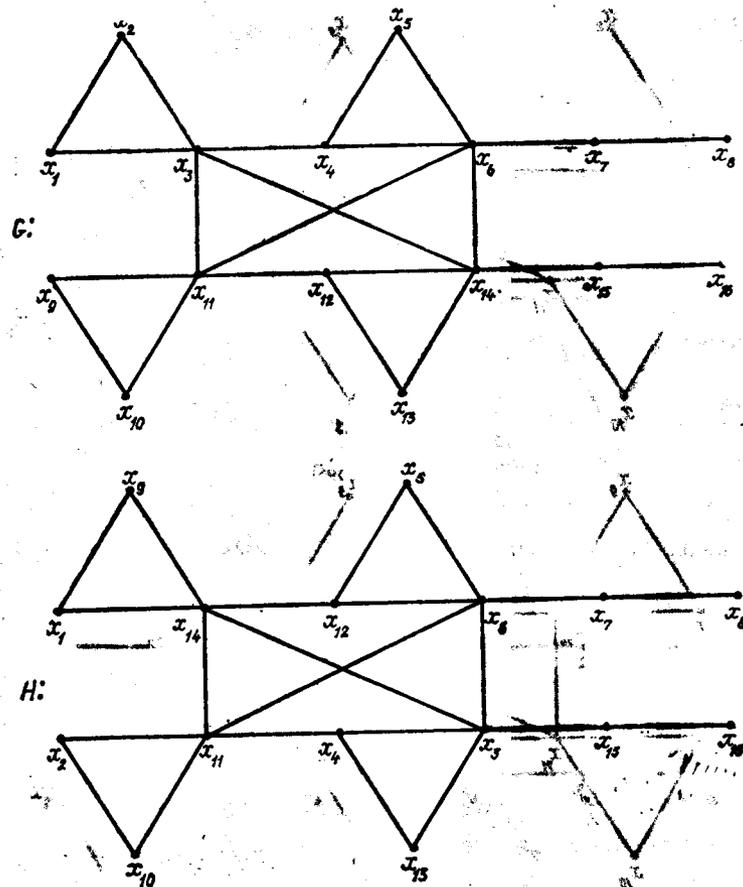


Рис. 5.7.12

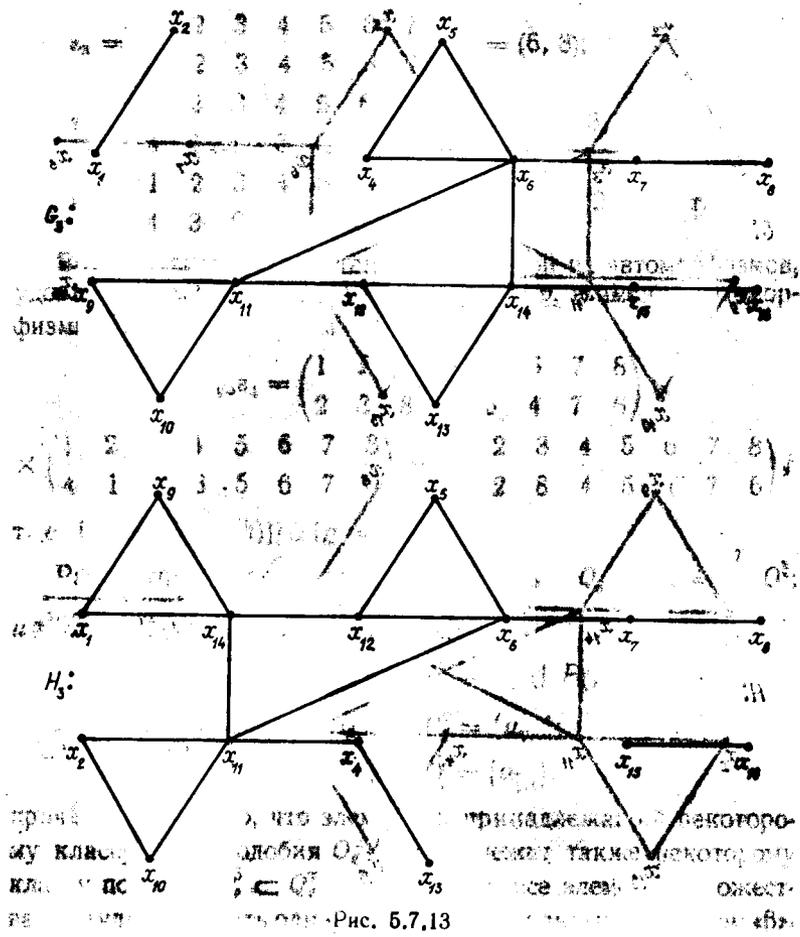
но, рассуждения наши почти идентичны рассуждениям в подслучае «а».

Во втором варианте очевиден аналог рассуждений рассуждениям, проведённым во 2-ом случае.

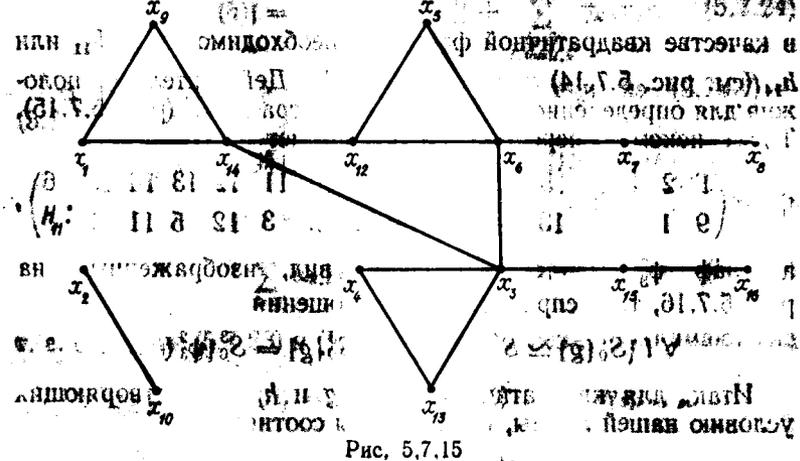
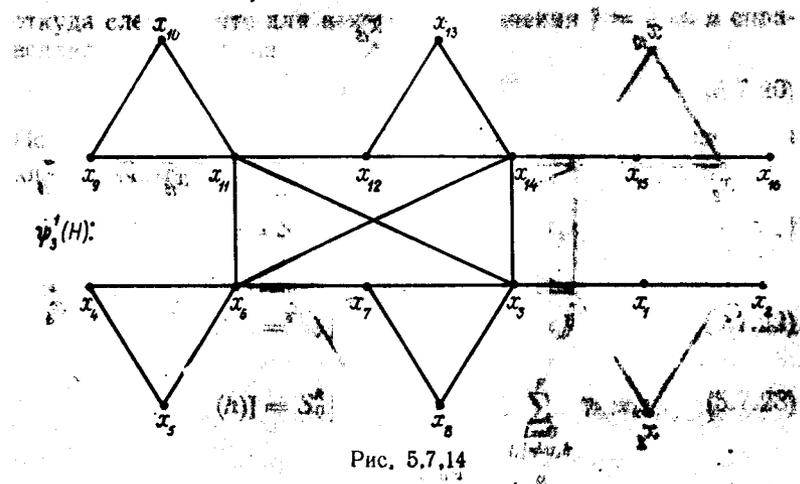
Осталось рассмотреть подслучай «в». И здесь явно видна идентичность рассуждений с рассуждениями, проведёнными во 2-ом случае.

Рассмотрим пример, относящийся ко второму варианту подслучая «б».

Пример Е. Пусть g и h — квадратичные формы соответственно графов G и H , изображенных на рис. 5.7.12, которые удовлетворяют условиям $\forall i \{S_0^i[g] \simeq S_0^i[h]\}$.



Положив для определенности $u = 3$, имеем (рис. 5.7.13) графы G_3 и H_3 .
 Для рассматриваемого примера имеем
 $Q_g = Q_h = \{3, 6, 11, 14\}$;
 $P_g^{\alpha=3} = \{3, 11\}$; $P_g^{\gamma=6} = \{6, 14\}$; $P_g^{\alpha=1} = P_h^{\alpha=1} = \{1, 2, 9, 10\}$;
 $P_h^{\beta=11} = \{11, 14\}$; $P_h^{\gamma=3} = \{3, 6\}$; $P_g^{\delta} = P_h^{\delta} = \{7, 15\}$;
 $Q_g = P_g^{\beta} \cup P_g^{\gamma}$; $Q_h = P_h^{\beta} \cup P_h^{\gamma}$.
 В силу условия $\forall i \{S_0^i[g] \simeq S_0^i[h]\}$
 $\exists \psi_3 \{S_0^3[g] = \psi_3[S_0^3[h]] = S_0^3[\psi_3(h)]\}$.



т. е. имеем перестановку ψ_3^1 вида

$$\psi_3^1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 9 & 4 & 3 & 7 & 13 & 14 & 15 & 16 & 10 & 5 & 6 & 12 & 8 & 11 & 1 & 2 \end{pmatrix}$$
 Тогда граф $\psi_3^1(H)$ примет вид (рис. 5.7.14).
 Очевидно, что $\neg (\forall i \{S_0^i[g] \simeq S_0^i[\psi_3^1(h)]\})$.
 И так как $P_{\psi_3^1(h)}^{\alpha=1} = \{1, 15\}$, а $P_g^{\delta=15} = \{7, 15\}$, то при $i = 1$ имеем $S_0^1[g] \not\simeq S_0^1[\psi_3^1(h)]$.
 Следовательно, для построения искомой перестановки необходимо воспользоваться соотношением (5.7.18), а именно $g_3 + x_0x_{17} + x_{17}x_7 \simeq h_1 + x_0x_{17} + x_{17}x_3$, откуда следует, что

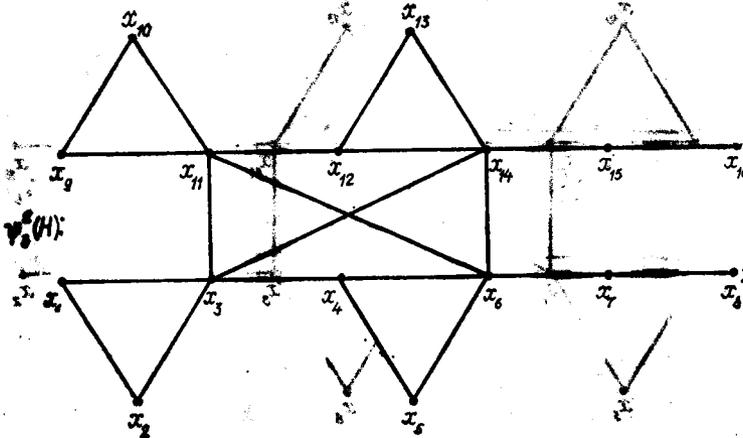


Рис. 5.7.16

в качестве квадратичной формы h_i необходимо взять h_{11} или h_{14} (см. рис. 5.7.14) ибо $11, 14 \in P_h^{p=11}$. Действительно, положив для определённости $j = 11$, имеем граф H_{11} (рис. 5.7.15). Тогда искомая перестановка ψ_3^2 примет вид

$$\psi_3^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 9 & 1 & 6 & 4 & 13 & 14 & 15 & 16 & 10 & 2 & 3 & 12 & 5 & 11 & 7 & 8 \end{pmatrix},$$

а граф $\psi_3^2(H)$ — соответственно вид, изображенный на рис. 5.7.16, т. е. справедливы соотношения

$$\forall i \{S_0^i[g] \simeq S_0^i[\psi_3^2(h)]\}; \quad S_0^3[g] = S_0^3[\psi_3^2(h)].$$

Итак, для квадратичных форм g и h , удовлетворяющих условию нашей леммы, справедливы соотношения

$$\forall i \exists \varphi_i \{g_i = S_0^i[\varphi_i(h)]\}; \quad \forall j \{g_j \simeq S_0^j[\varphi_j(h)]\}.$$

Положив для определённости $i = u$, имеем

$$g_u = S_0^u[\varphi_u(h)]; \quad \forall j \{g_j \simeq S_0^j[\varphi_u(h)]\}.$$

Покажем справедливость утверждения леммы, т. е.

$$\forall j \{g_j = S_0^j[\varphi_u(h)]\}.$$

Допустим обратное:

$$\exists j \{g_j \neq S_0^j[\varphi_u(h)]\},$$

т. е. $\exists j \{g_j \neq S_0^j[\varphi_u(h)]\}$.

откуда следует, что для некоторого значения $j = k \neq u$ справедливо соотношение

$$g_k \neq S_0^k[\varphi_u(h)]. \quad (5.7.20)$$

Покажем, что предположение (5.7.20) неверно. Представим квадратичные формы $g_u, g_k, S_0^u[\varphi_u(h)]$ и $S_0^k[\varphi_u(h)]$ в виде

$$g_u = S_0^k[S_0^u[g]] + \sum_{\substack{i=0 \\ i \neq u, k}}^p \alpha_{k,i} x_k x_i; \quad (5.7.21)$$

$$g_k = S_0^u[S_0^k[g]] + \sum_{i=0}^p \alpha_{u,i} x_u x_i; \quad (5.7.22)$$

$$S_0^u[\varphi_u(h)] = S_0^k[S_0^u[\varphi_u(h)]] + \sum_{i,j \neq u, k}^p \gamma_{k,i} x_k x_i; \quad (5.7.23)$$

$$S_0^k[\varphi_u(h)] = S_0^u[S_0^k[\varphi_u(h)]] + \sum_{i,j \neq u, k}^p \gamma_{u,i} x_u x_i. \quad (5.7.24)$$

В силу свойств оператора подстановки из (5.7.21) — (5.7.24) имеем

$$S_0^k[g_u] = S_0^k[S_0^k[S_0^u[g]]] = S_0^u[S_0^k[g]] = S_0^k[S_0^k[S_0^u[\varphi_u(h)]]] = S_0^u[S_0^k[S_0^u[\varphi_u(h)]]];$$

$$\sum_{i=0}^p \alpha_{k,i} x_k x_i = \sum_{i=0}^p \alpha_{k,i} x_k x_i; \quad (5.7.25)$$

т. е. формулы (5.7.23) и (5.7.24) соответственно принимают вид

$$g_u = S_0^u[\varphi_u(h)] = S_0^k[S_0^u[\varphi_u(h)]] + \sum_{i=0}^p \gamma_{k,i} x_k x_i;$$

$$S_0^k[\varphi_u(h)] = S_0^u[S_0^k[\varphi_u(h)]] + \sum_{i=0}^p \gamma_{u,i} x_u x_i. \quad (5.7.26)$$

Непосредственно из (5.7.22) и (5.7.26) следует, что предположение (5.7.20) выполнено при условии

$$\sum_{i=0}^p \alpha_{u,i} x_u x_i = \sum_{i=0}^p \gamma_{u,i} x_u x_i,$$

или $\neg (\forall i \{ \alpha_{u,i} = \gamma_{u,i} \})$, откуда следует

$$\exists n \exists \bar{n} \{ [\alpha_{u,n} \neq \gamma_{u,n}] \& [\alpha_{u,\bar{n}} \neq \gamma_{u,\bar{n}}] \}. \quad (5.7.27)$$

Из (5.7.23) и (5.7.25) следует, что

$$\sum_{i=0}^p \alpha_{u,i} x_u x_i \cong \sum_{i=0}^p \gamma_{u,i} x_u x_i, \quad (5.7.28)$$

$$\forall i \{ [\alpha_{n,i} = \gamma_{n,i}] \& [\alpha_{\bar{n},i} = \gamma_{\bar{n},i}] \}, \quad (5.7.29)$$

из (5.7.1) следует, что $\forall i [\tau_g(i) = \tau_h(i)]$, где $\tau_g(i)$ ($\tau_h(i)$) — число вхождений переменной x_i в квадратичную форму g (h). Следовательно,

$$[\tau_g(n) = \tau_h(n)] \& [\tau_g(\bar{n}) = \tau_h(\bar{n})],$$

что вместе с соотношениями (5.7.28) и (5.7.29) противоречит соотношениям (5.7.27), т. е. предположение (5.7.20) противоречиво.

Лемма 5.7.5. Если квадратичные формы

$$g = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j; \quad h = \sum_{i,j=0}^p \beta_{i,j} x_i x_j$$

удовлетворяют соотношению $\forall i \{ S_0^i[g] = S_0^i[h] \}$, то они равны $\forall i \forall j \{ \alpha_{i,j} = \beta_{i,j} \}$.

Доказательство. Допустим противное: $\exists m \exists n \times \times \{ \alpha_{m,n} \neq \beta_{m,n} \}$. Из условия леммы и определения тождества квадратичных форм следует, что для трех попарно различных значений $i = k, m, n$ ($1 \leq k, m, n \leq p$) справедливы равенства

$$\alpha_{i,j} = \beta_{i,j} \quad (i, j = \overline{1, p}); \quad (i \neq k; j \neq k);$$

$$\alpha_{i,j} = \beta_{i,j} \quad (i, j = \overline{1, p}); \quad (i \neq m; j \neq m);$$

$$\alpha_{i,j} = \beta_{i,j} \quad (i, j = \overline{1, p}); \quad (i \neq n; j \neq n),$$

откуда очевидна справедливость равенств $\forall i \forall j \{ \alpha_{i,j} = \beta_{i,j} \}$, в том числе и равенства $\alpha_{m,n} = \beta_{m,n}$, что явно противоречит предположению.

Доказательство теоремы 5.7.1 завершено.

Следствия из теоремы 5.7.1.

1. Если вершинно-допустимый набор $(p-1)$ -го порядка $\Gamma_G^1 = \{G_i^1\}$, $i = \overline{1, r}$, для некоторого p -вершинного графа не полон, т. е. $1 \leq r \leq p$, то существует эффективная процедура, позволяющая за конечное число шагов восстановить хотя бы один искомый граф G , такой, что

$$\forall i \forall j \{ S_0^i[G] = (G_i \div x_j) \} \quad (i = \overline{1, r}; 1 \leq j \leq p).$$

2. Любой полный вершинно-допустимый набор $(p-s)$ -го порядка $\Gamma_G^s = \{G_i^{p-s}\}$ ($i = \overline{1, p-s}$) для произвольного p -вершинного графа G при $s < p-1$ позволяет однозначно (с точностью до изоморфизма) восстановить искомый граф.

5.8. МИНИМАЛЬНЫЕ НАБОРЫ ПОДГРАФОВ

В данном параграфе дается подход к решению еще одной задачи из [4], которая на языке теории графов может быть сформулирована следующим образом.

Каково минимальное число из $m(p, k)$, обладающее тем свойством, что набор из $m(p, k)$ k -вершинных подграфов $\{G_n^k\}$ ($n = \overline{1, m(p, k)}$; $2 \leq k < p$) некоторого (p, q) -графа G несет полную информацию о всем графе G , т. е. граф G восстанавливается однозначно с точностью до изоморфизма?

Определение. Назовем минимальным вершинно-допустимым набором подграфов k -го порядка для p -вершинного G графа подмножество из $m(p, k)$ k -вершинных подграфов $\Gamma_{\min} = \{G_n^k\}$ ($n = \overline{1, m(p, k)}$; $p \geq 3$; $2 \leq k < p$) множества полного вершинно-допустимого набора k -вершинных подграфов $\Gamma = \{G_i^k\}$ ($n = \overline{1, c_k^p}$) графа G , если удаление хотя бы одного подграфа из Γ_{\min} приводит к тому, что найдется хотя бы одна пара вершин x_i и x_j графа G такая, что ни один из оставшихся в Γ_{\min} подграфов не несет информацию о смежности вершин x_i и x_j .

Допустим, что каждый из подграфов $\{G_n^k\}$ ($n = \overline{1, c_k^p}$) некоторого (p, q) -графа G задан своей квадратичной формой

$$g_n = \sum_{i,j \in I_n} \beta_{i,j}^{(n)} x_i x_j,$$

где $I_n = \{i_1^{(n)}, i_2^{(n)}, \dots, i_l^{(n)}, \dots, i_k^{(n)}\}$, $i_l \notin N_p = \{1, 2, \dots, p\}$; $1 \leq n \leq c_k^p$.

Тогда, по определению, коэффициенты $\alpha_{i,j}$ набора квадратичных форм, соответствующего минимальному вершинно-допустимому набору подграфов $\Gamma_{\min} = \{G_n^k\}$ ($n = \overline{1, m(p, k)}$), в совокупности должны «накрывать» все коэффициенты квадратичной формы $g = \sum_{i,j} \alpha_{i,j} x_i x_j$ графа G , т. е. каждый из коэффициентов $\alpha_{i,j}$ квадратичной формы g однозначно определяется соответствующими коэффициентами $\beta_{i,j}^{(n)}$ хотя бы одной из квадратичных форм g_n .

Ясно, что функция $m(p, k)$ при фиксированном значении аргумента p является убывающей функцией по аргументу k .

Следовательно, наша задача сводится к определению оценок нижней и верхней границ функций $m(p, k)$ при следующих ограничениях $p \geq 3$; $2 \leq k \leq p-1$.

Откуда очевидно, что функция $m(p, k)$ достигает наименьшего значения при $k = p-1$, а наибольшего — при $k = 2$.

Поэтому рассмотрим в отдельности следующие три случая: $k = p - 1$; $k = 2$; $2 < k < p - 1$.

Рассмотрим $k = p - 1$. Покажем, что для данного случая справедливо $\forall p \{m(p, p - 1) = 3\}$. Действительно. Пусть для попарно различных a, b и c имеем

$$g_a = \sum_{\substack{i,j=0 \\ i,j \neq a,b,c}}^p \beta_{i,j}^{(a)} x_i x_j + \sum_{\substack{i=0 \\ i \neq a,b,c}}^p \beta_{a,i}^{(a)} x_b x_i + \sum_{\substack{i=0 \\ i \neq a,b,c}}^p \beta_{c,i}^{(a)} x_c x_i,$$

$$g_b = \sum_{\substack{i,j=0 \\ i,j \neq a,b,c}}^p \beta_{i,j}^{(b)} x_i x_j + \sum_{\substack{i=0 \\ i \neq a,b,c}}^p \beta_{a,i}^{(b)} x_a x_i + \sum_{\substack{i=0 \\ i \neq a,b,c}}^p \beta_{c,i}^{(b)} x_c x_i,$$

$$g_c = \sum_{\substack{i,j=0 \\ i,j \neq a,b,c}}^p \beta_{i,j}^{(c)} x_i x_j + \sum_{\substack{i=0 \\ i \neq a,b,c}}^p \beta_{a,i}^{(c)} x_a x_i + \sum_{\substack{i=0 \\ i \neq a,b,c}}^p \beta_{b,i}^{(c)} x_b x_i.$$

В силу соотношений $S_{0,0}^{a,b}[g] \simeq S_0^b[g_a] \simeq S_0^a[g_b]$ очевидно существование перестановки p -го порядка $\varphi_{b,a}$ такой, что

$$\varphi_{b,a}(b) = b;$$

$$S_{0,0}^{a,b}[g] = S_0^b[g_a] = S_0^a[\varphi_{b,a}(g_b)];$$

$$\alpha_{i,j} = \beta_{i,j}^{(a)} = \bar{\beta}_{i,j}^{(b)} \quad (i, j \neq a, b),$$

где $\bar{\beta}_{i,j}^{(b)} = \varphi_{b,a}(\beta_{i,j}^{(b)})$.

Аналогично, положив $S_{0,0}^{a,c}[g] = S_0^c[\varphi_{c,a}(g_c)] = S_0^a[g_a]$, имеем $\alpha_{i,j} = \beta_{i,j}^{(a)} = \bar{\beta}_{i,j}^{(c)}$ ($i, j \neq a, c$), где $\varphi_{c,a}$ — перестановка p -го порядка такая, что $\varphi_{c,a}(c) = c$; $\bar{\beta}_{i,j}^{(c)} = \varphi_{c,a}(\beta_{i,j}^{(c)})$.

Для квадратичных форм g_b и g_c имеем

$$\alpha_{i,j} = \bar{\beta}_{i,j}^{(b)} = \bar{\beta}_{i,j}^{(c)} \quad (i, j \neq b, c).$$

Следовательно, из трех последних тождеств можем однозначно определить все коэффициенты $\alpha_{i,j}$ квадратичной формы g . Следует заметить: если у p -вершинного графа G имеются только два неизоморфных подграфа, то для его восстановления недостаточно знать только неизоморфные подграфы. Поэтому гипотеза Ф. Харари о том, что при $p \geq 4$ для восстановления p -вершинного графа G достаточно знать не все подграфы $G \pm x_i$, а только неизоморфные, справедлива, если у графа G не менее трех неизоморфных $(p - 1)$ -вершинных подграфов.

Если набор подграфов $\{G \pm x_i\}$ включает только два неизоморфных графа G_i и G_j , т. е. множество $\{G \pm x_i\}$ можно разбить на два таких класса A и B , что элементы одного изоморфны только G_i , а элементы другого изоморфны только G_j , то, очевидно, для однозначного восстановления исходного графа G необходимо знать $\|A\| \vee \|B\|$.

В случае $k = 2$, очевидно, что $m(p, k) = m(p, 2) = c_2^p$, так как каждая квадратичная форма g_n несет информацию о смежности только одной пары вершин графа G из общего числа c_2^p всевозможных пар вершин.

Рассмотрим случай $2 < k < p - 1$.

Для наглядности рассмотрим симметричную матрицу p -го порядка $M = \{\alpha_{i,j}\}$ ($i, j = \overline{1, p}$; $\alpha_{i,i} = 0$), которая соответствует квадратичной форме $g = \sum_{i,j=0}^p \alpha_{i,j} x_i x_j$. Так как эта матрица симметричная, будем оперировать верхней ее «косынкой». Очевидно, любая квадратичная форма

$$g_n^{(k)} = \sum_{\substack{i,j=0 \\ i,j \in I_n}}^p \beta_{i,j}^{(n)} x_i x_j, \quad 1 \leq n \leq \binom{p}{k},$$

несет информацию, которая может полностью «накрыть» $k(k - 1)$ элементов этой матрицы. Если взять для определенности квадратичные формы

$$g_1 = \sum_{\substack{i,j=0 \\ i,j \in I_1}}^p \beta_{i,j}^{(1)} x_i x_j, \quad I_1 = \{1, 2, \dots, k\};$$

$$g_2 = \sum_{\substack{i,j=0 \\ i,j \in I_2}}^p \beta_{i,j}^{(2)} x_i x_j, \quad I_2 = \{p, p - 1, \dots, p - (k - 1)\},$$

то коэффициенты этих двух квадратичных форм накроют $2k(k - 1) - l(l - 1)$, ($l = k - (p - k) = 2k - p$) элементов верхней «косынки» матрицы M . Остается покрыть оптимально оставшуюся часть нашей «косынки» коэффициентами других квадратичных форм g_n ($I_n \neq I_1, I_2$), а именно $(p - k)^2$ элементов верхней «косынки» матрицы M . Для этого необходимо последующие квадратичные формы g_n подбирать таким образом, чтобы

$$\max \|I_1 \cap I_n\| = \left\lfloor \frac{k}{2} \right\rfloor, \quad \text{а} \quad \max \|I_2 \cap I_n\| = k - \left\lfloor \frac{k}{2} \right\rfloor$$

или

$$\max \|I_1 \cap I_n\| = k - \left\lfloor \frac{k}{2} \right\rfloor, \quad \text{а} \quad \max \|I_2 \cap I_n\| = \left\lfloor \frac{k}{2} \right\rfloor,$$

т. е. задача сводится к вычислению отношения

$$\bar{m} = \frac{(p - k)^2}{\left\lfloor \frac{k}{2} \right\rfloor \left(k - \left\lfloor \frac{k}{2} \right\rfloor \right)}. \quad (5.8.1)$$

В частности, при $p - k \leq [k/2]$, т. е. при $k \geq \frac{3}{2}p$ соотношение (5.8.1) примет вид

$$\bar{m} = \frac{(p-k)^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)} \leq \frac{\left[\frac{k}{2}\right]^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)} = \frac{\left[\frac{k}{2}\right]}{k - \left[\frac{k}{2}\right]} \leq 1,$$

откуда следует, что $m(p, k) = m(p, \frac{3}{2}p) = 2 + \bar{m} = 2 + 1 = 3$.

Если положить $k \geq p/2$, то

$$\bar{m} = \frac{(p-k)^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)} \leq \frac{(2k-k)^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)}$$

$$= \frac{k^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)} = \frac{\left(\left[\frac{k}{2}\right] + k - \left[\frac{k}{2}\right]\right)^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)}$$

$$= \frac{\left[\frac{k}{2}\right]^2 + 2\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right) + \left(k - \left[\frac{k}{2}\right]\right)^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)}$$

$$= 2 + \frac{\left[\frac{k}{2}\right]}{k - \left[\frac{k}{2}\right]} + \frac{k - \left[\frac{k}{2}\right]}{\left[\frac{k}{2}\right]} \leq 5,$$

т. е. $m(p, k) = m\left(p, \left[\frac{p}{2}\right]\right) \leq 2 + \bar{m} \leq 7$.

Откуда в качестве оценок для минимального $m(p, k)$ имеем

$$2 + \frac{(p-k)^2}{\left[\frac{k}{2}\right]\left(k - \left[\frac{k}{2}\right]\right)} \leq m(p, k) \leq c_k^p = \frac{p(p-1)}{k(k-1)}.$$

В заключение дадим (без вывода) некоторую рекуррентную формулу для вычисления уточненной верхней границы функции $m(p, k)$ при $k < p/2$.

Введем следующие обозначения:

$$a = \left[\frac{k}{2}\right]; \quad b = k - a; \quad m_i = \left[\frac{p - ik + b}{b}\right];$$

$$n_i = \left[\frac{p - ik}{a}\right];$$

$$\alpha_i = p - m_i b; \quad \beta_i = p - n_i a, \quad \text{т. е.}$$

$$\alpha_i \in \{0, 1, \dots, \delta, \dots, b-1\}; \quad \beta_i \in \{0, 1, \dots, \delta, \dots, a-1\}.$$

Обозначим символом $\gamma_\delta \sum \alpha_i + \beta_i$, где α_i, β_i те α_i и β_i , для которых справедливо $\alpha_i = \beta_i = \delta$.

Тогда для вычисления верхней границы функции $m(p, k)$ при $k < p/2$ предлагается формула

$$\bar{m}(p, k) = \sum_{i=1}^{\left[\frac{p}{k}\right]-1} (m_i + n_i) + \sum_{i=0}^{a-1} \frac{\gamma_{a-i} - \rho_{a-i+1} + \rho_{a-i}}{b+i},$$

$$\text{где } \rho_a = \begin{cases} 0 & \text{при } \left\{\frac{\gamma_a}{b}\right\} = 0; \\ b - \left\{\frac{\gamma_a}{b}\right\}b & \text{при } \left\{\frac{\gamma_a}{b}\right\} \neq 0; \end{cases}$$

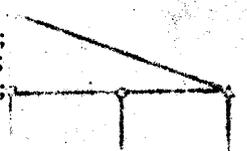
$$\rho_{a-i} = \begin{cases} 0 & \text{при } \left\{\frac{\gamma_{a-i} - \rho_a}{b+i}\right\} = 0; \\ b+i - \left\{\frac{\gamma_{a-i} - \rho_a}{b+i}\right\}(b+i) & \text{при } \left\{\frac{\gamma_{a-i} - \rho_a}{b+i}\right\} \neq 0. \end{cases}$$

Таким образом, в качестве оценок функции $m(p, k)$ имеем

$$3 \leq \frac{(p-k)^2}{ab} \leq m(p, k) \leq \bar{m}(p, k) \leq c_k^p.$$

Упражнения.

- Проверить на изоморфизм графы:
 - G_4 и G_5 , изображенные на рис. 5.1.1;
 - G и H , изображенные на рис. 5.3.3;
 - \bar{G}_1 и \bar{G}_3 , изображенные на рис. 5.1.1;
 - $G[H]$ и $\bar{G}[\bar{H}]$;
 - $G \& H$ и $\bar{G} \& \bar{H}$;
 - $G \times H$ и $\bar{G} \times \bar{H}$.
- Найти $\varphi(Q)$, $\epsilon(Q)$, $\gamma(Q)$, $\eta(Q)$, $F(Q)$, $E(Q)$, $\Gamma(Q)$, $H(Q)$ для графа Q , изображенного на рис. 1.5.2.
- Для графов G и H , изображенных на рис. 5.2.3, вычислить матрицы смежностей, спектры и матрицы минимальных расстояний.
- Для графов G, G', H и H' , изображенных на рис. 5.8.1, найти матрицы смежностей и минимальных расстояний.
- Вычислить мини-коды μ и макси-коды $\bar{\mu}$ для графов G_4 и G_5 , изображенных на рис. 5.1.1.
- Начертить граф G без изолированных вершин по его мини-коду $\mu(G) = 236$.
- По графам G и H , изображенным на рис. 5.8.2, определить такие нумерации их вершин, которые приводят к мини-кодам, найти эти коды. Решить аналогичные задачи для макси-кода.
- Начертить графы $G = P_2 \diamond C_3$ и $H = C_3 \diamond P_2$ и проверить их на изоморфизм.



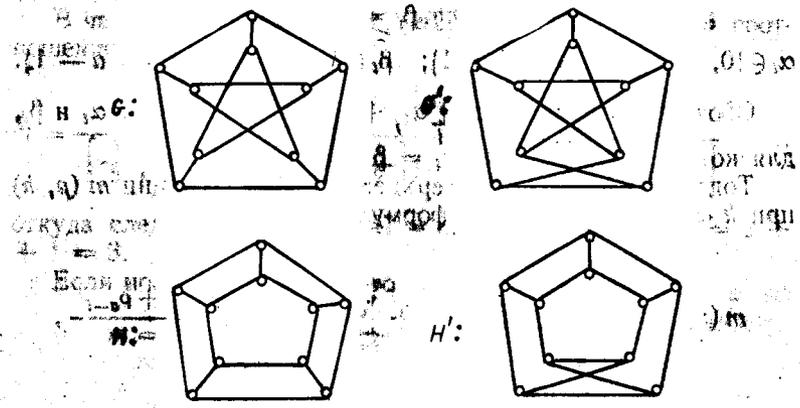


Рис. 5.8.1

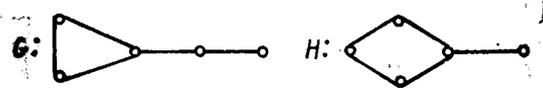


Рис. 5.8.2

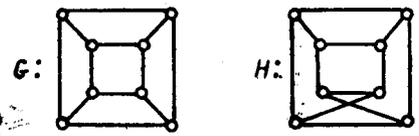


Рис. 5.8.3

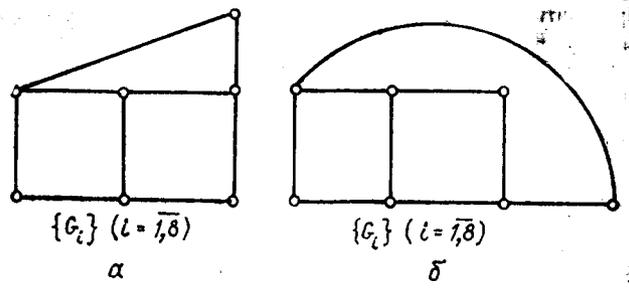


Рис. 5.8.4

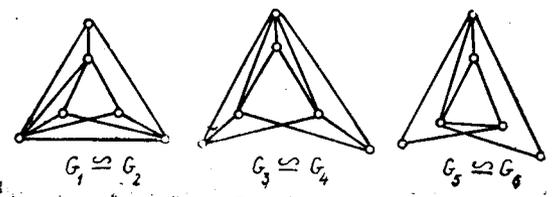


Рис. 5.8.5

9. Для 8-вершинных графов G и H , изображенных на рис. 5.8.3, определить вершинно-полные допустимые наборы 7-го порядка $\{G_i^7\}$ и $\{H_i^7\}$ ($i = \overline{1, 8}$). Проверить на изоморфизм $\{G_i^7\}$ и $\{H_i^7\}$.

10. Для графов $H(X, V)$, изображенных на рис. 5.8.3, начертить полный набор всех 8-вершинных надграфов $\{H_i(X_i, V_i)\}$, таких, что $\forall j (x_j, x_j) \in V_i$.

11. Восстановить 8-вершинный граф G по набору его подграфов $\{G_i\}$ ($i = \overline{1, 8}$), заданному на рис. 5.8.4.

12. Восстановить 6-вершинный граф G по набору его надграфов $\{G_i\}$ ($i = \overline{1, 6}$), заданному на рис. 5.8.5.

ЗАКЛЮЧЕНИЕ

Из всех задач представления и восстановления графов, несомненно, гипотеза Улама является наиболее представительной. Пользуясь выводами гл. 2 о классах P и NP , можем провести анализ этой задачи. Напомним, что класс NP в качестве подклассов содержит класс P и класс NP -полных задач. В предположении $P = NP$ эти два класса не пересекаются. Это означает, что существует класс задач, имеющий промежуточную сложность, т. е. находится между P и NP . То, что такой класс не пуст, есть следствие более общего результата, доказанного Ладнером.

Существует несколько естественных задач, которые можно отнести к этому классу. Наиболее известны из них следующие две.

ИЗОМОРФИЗМ ГРАФОВ.

УСЛОВИЕ. Заданы графы $G = (X, U)$ и $G' = (X, U')$.

ВОПРОС. Верно ли, что графы G и G' изоморфны, т. е. существует ли взаимно однозначная функция $f: X \leftrightarrow X$ такая, что $\{x, y\} \in U$ тогда и только тогда, когда $\{f(x), f(y)\} \in U'$?

СОСТАВНЫЕ ЧИСЛА.

УСЛОВИЕ. Задано положительное целое число K .

ВОПРОС. Существуют ли целые числа m и n ; $m, n > 1$ такие, что $K = mn$?

Исследователи, предпринимавшие попытки доказать, что задача ИЗОМОРФИЗМ ГРАФОВ является NP -полной, отмечают, что ее структура гораздо более ограничительна, чем структура типичной NP -полной задачи, какой является задача ИЗОМОРФИЗМ ПОДГРАФУ.

При построении алгоритма восстановления графа по его полному набору подграфов их изоморфизм задается. Поэтому общая проблема переносится и на построение номинального алгоритма восстановления графа. Это означает, что попытки как доказательства того, что эта задача лежит в промежуточном классе, так и поиски полиномиального алгоритма восстановления графа по его полному набору должны быть продолжены.

СПИСОК ЛИТЕРАТУРЫ

1. Берг К. Теория графов и ее применение. — М.: Изд-во иностр. лит., 1962. — 319 с.

2. Berge C. Graphes et hypergraphes. — Paris: Dunod, 1970. — 312 с.

3. Ore O. Теория графов. — М.: Наука, 1986. — 352 с.

4. Зыков А. А. Основы теории графов. — М.: Наука, 1987. — 384 с.

5. Григорян Ю. Г., Маноян Г. К. Некоторые вопросы арифметической интерпретации графов // Кибернетика. — 1977. — № 3. — С. 129—131.

6. Донец Г. А. О графах, задаваемых аналитическим способом // Теория оптимальных решений. — Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1987. — С. 20—27.

К главе 1

1. Берг К. Теория графов и ее применение. — М.: Изд-во иностр. лит., 1962. — 319 с.
2. Berge C. Graphes et hypergraphes. — Paris: Dunod, 1970. — 312 с.
3. Ore O. Теория графов. — М.: Наука, 1986. — 352 с.
4. Зыков А. А. Основы теории графов. — М.: Наука, 1987. — 384 с.
5. Григорян Ю. Г., Маноян Г. К. Некоторые вопросы арифметической интерпретации графов // Кибернетика. — 1977. — № 3. — С. 129—131.
6. Донец Г. А. О графах, задаваемых аналитическим способом // Теория оптимальных решений. — Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1987. — С. 20—27.

К главе 2

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979. — 536 с.
2. Кнут Д. Искусство программирования для ЭВМ: В 3-х т. — М.: Мир, 1976. — Т. 1.
3. Григорян Ю. Г., Маноян Г. К. Некоторые вопросы арифметической интерпретации графов // Кибернетика. — 1977. — № 3. — С. 129—131.
4. Донец Г. А. О графах, задаваемых аналитическим способом // Теория оптимальных решений. — Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1987. — С. 20—27.
5. Асельдерова И. М., Донец Г. А. Оптимальное кодирование циклов и однородных деревьев // Теория и практика разработки и внедрения интегральных АСУ. — Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1987. — С. 87—94.

К главе 3

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979. — 536 с.
2. Кук С. А. Сложность процедур вывода теорем // Кибернет. сб. — М.: Мир, 1975. — Вып. 2. — С. 5—15.
3. Карп Р. М. Сводимость комбинаторных задач // Там же. — С. 16—38.
4. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. — М.: Мир, 1972. — 624 с.
5. Трахтенброт Б. А., Бардзынь Я. М. Конечные автоматы: поведение и синтез. — М.: Наука, 1970. — 400 с.

К главе 4

1. Зыков А. А. Основы теории графов.— М.: Наука, 1987.— 384 с.
2. Харари Ф. Теория графов.— М.: Мир, 1973.— 300 с.
3. Асельдеров З. М. Представление графов и операции над ними.— Киев, 1987.— 18 с. (Препр./ АН УССР. Ин-т кибернетики им. В. М. Глушкова; 87—49).

К главе 5

1. Зыков А. А. Основы теории графов.— М.: Наука, 1987.— 384 с.
2. Цветкович Д., Дуб М., Закс Х. Спектры графов. Теория и применение.— Киев: Наук. думка, 1984.—
3. Визинг В. Г. Сведение проблем изоморфизма и изоморфного вложения к задаче неплотности графа // Тез. докл. III Всесоюз. конф. «Проблемы теоретической кибернетики», 1974 г.— Новосибирск, 1974.— С. 124—125.
4. Улам С. М. Нерешенные математические задачи.— М.: Наука, 1964.— 168 с.
5. Muller V. Probabilistic reconstruction from subgraphs // Comment. math. Univ. carol.— 1976.— 17, N 4.— P. 709—719.
6. Fisher J. A countereample to the countable version of a conjecture of Ulam // Journal of Combinatorial Theory.— 1969.— 7, N 4.— P. 364—365.
7. Clarke F. H. A graph polynomial and its applications // Discrete Mathematics.— 1972.— 3, N 4.— P. 305—313.
8. Pouzet M. Note sur le probleme de Ulam // Journal of Combinatorial Theory.— 1979.— 27, N 3.— P. 231—236.
9. Черняк Ж. А. Несколько дополнений к одной статье Маквела // Вестн АН БССР.— 1982.— N 6.— С. 44—49.
10. Harary F. A survey of the reconstruction conjecture // Lect. Notes Math.— 1974.— 406.— P. 18—28.
11. Булитко В. К. О графах с заданным окружением вершин // Тр. Ин-та мат. АН СССР.— 1973.— 133.— С. 78—94.
12. Калужский Л. А., Суцанский В. И. Преобразования и перестановки.— М.: Наука, 1986.— 160 с.
13. Гросман И., Магнус В. Группы и графы.— М.: Мир, 1971.— 189 с.
14. Харари Ф. Теория графов.— М.: Мир, 1973.— 300 с.
15. Асельдеров З. М. Представление графов и операции над ними.— Киев, 1987.— 18 с.— (Препр./ АН УССР. Ин-т кибернетики им. В. М. Глушкова; 87—49).

ОГЛАВЛЕНИЕ

401	ОГЛАВЛЕНИЕ	
501		
601		
311		
841		
771		
181		
381		
Предисловие		3
Список условных обозначений		4
ГЛАВА 1. Способы представления графов		5
1.1. Общее представление произвольных графов		5
1.2. Задание графов с помощью матриц		8
1.3. Двоичное представление графов		11
1.4. Бинарные отношения для графов		14
1.5. Задание графа в виде формальной квадратичной формы		18
1.6. Аналитическое представление графов		20
ГЛАВА 2. Проблемы оптимального представления графов		28
2.1. Представление графов с помощью структур данных		28
2.2. Представление деревьев		32
2.3. Оценка числа операций алгоритмов		35
2.4. Об оптимальной кодировке арифметических графов		37
ГЛАВА 3. Элементы теории сложности алгоритмов для задач на графах		55
3.1. Основные понятия		55
3.2. Классы P и NP		61
3.3. Полиномиальная сводимость и NP-полные задачи		67
3.4. Доказательство результатов об NP-полноте		71
3.5. Применение теории NP-полноты для анализа задач		77
ГЛАВА 4. Операция над обыкновенными графами		85
4.1. Операции над вершинами и ребрами		85
ГЛАВА 5. Восстановление графов		95
5.1. Изоморфизм		95
5.2. Инварианты		97

5.1.1.	106
5.1.2.	107
5.6. Траекторное управление	109
5.6. Алгоритмический подход к траекторному управлению	112
5.7. Теорема о существовании единственного решения	114
5.8. Матричные методы в теории управления	117
Заключение	124
Список литературы	125

ВНИМАНИЮ СПЕЦИАЛИСТОВ В ОБЛАСТИ МАТЕМАТИЧЕСКОЙ ТЕОРИИ УПРАВЛЕНИЯ

Лебедев Д. В., Ткаченко А. И. Системы инерциального управления. Алгоритмические аспекты. 12 л.— Ил.— 2 р. 70 к.

ИБ № 11516

Сдано в набор 02.11.90 Подп. в печ. 23.04.91.
 Формат 84×108^{1/2}. Бум. тип. № 1. Лит.
 гарн. Выс. печ. Усл. печ. л. 10,08. Усл.кр.-отт.
 10,4. Уч.-изд. л. 10,31. Тираж 810 экз.
 Заказ 0—3563. Цена 2 р. 70 к.

Издательство «Наукова думка» 252601 Киев 4, ул.
 Репина, 3.

Отпечатано с матриц Головного предприятия
 республиканского производственного объеди-
 нения «Полиграфкинга», 252057, Киев, ул.
 Довженко, 3, в-Нестеровской городской типо-
 графии 292310, Нестеров, Львовской обл., ул.

Горького, 8. Зак. 1433.

ВНИМАНИЮ СПЕЦИАЛИСТОВ В ОБЛАСТИ МАТЕМАТИЧЕСКОЙ ТЕОРИИ УПРАВЛЕНИЯ

В 1991 г. издательство «Наукова думка» выпускает в свет книги

Лебедев Д. В., Ткаченко А. И. Системы инерциального управления. Алгоритмические аспекты.
 12 л.— Ил.— 2 р. 70 к.

Объявлено в темплане 1991 г., поз. 164.

В монографии систематически изложены алгоритмические аспекты синтеза систем инерциального управления с информационной частью в виде бесплатформенной инерциальной навигационной системы (БИНС). Рассмотрены новые подходы к исследованию возникающего при этом комплекса задач. Представлен метод синтеза алгоритмов управления при наличии ограничений на управляющие параметры, когда их число превосходит число степеней свободы объекта как твердого тела. Изложена методика построения алгоритмов численного интегрирования кинематических и навигационных уравнений БИНС. Рассмотрены возможности использования метода наименьших квадратов в задачах коррекции БИНС. Исследованы вопросы формирования требований к техническим средствам системы инерциального управления.