Л. А. ГЛАДКОВ
В. В. КУРЕЙЧИК
В. М. КУРЕЙЧИК

ДИСКРЕТНАЯ МАТЕМАТИКА

УЧЕБНИК

ПОД РЕДАКЦИЕЙ
В. М. КУРЕЙЧИКА

Допущено Учебно-методическим объединением вузов
по университетскому политехническому образованию в качестве
учебника для студентов высших учебных заведений,
обучающихся по направлениям 230100 «Информатика
и вычислительная техника» и 230104 «Информационные системы»

МОСКВА
ФИЗМАТЛИТ®
2014

УДК 621.3+681.3
ББК 22.176
Г 52

Гладков Л.А., Курейчик В.В., Курейчик В.М. **Дискретная математика** / Под ред. В.М. Курейчика. — М.: ФИЗМАТЛИТ, 2014. — 496 с. — ISBN 978-5-9221-1575-9.

В книге представлены основные разделы дискретной математики: теория множеств, алгоритмов, графов, алгебра логики. Для лучшего усвоения материала использована современная методика обучения на основе решебников. Авторы рассмотрели вопросы исчисления множеств, задания отношений и соответствий, описания упорядоченных бесконечных множеств, мультимножеств и нечетких множеств, основные алгоритмические модели, логические функции и законы алгебры логики, виды и способы задания графов, алгоритмы решения задач на ориентированных и неориентированных графах, а также основные определения из теории гиперграфов и нечетких графов. Даются контрольные задачи, упражнения и глоссарий с пояснением терминов.

Учебник предназначен студентам вузов, обучающимся по направлениям «Информатика и вычислительная техника» и «Информационные системы», может быть полезен также специалистам, занятым разработкой интеллектуальных САПР, систем поддержки и принятия решений, новых информационных технологий в науке, технике, образовании, бизнесе и экономике.

Рецензенты:

кафедра прикладной математики МЭИ (зав. кафедрой д.т.н., профессор, лауреат премии президента РФ в области образования А.П. Еремеев);

Ю.О. Чернышев, заслуженный деятель науки РФ, д.т.н., профессор, Донской государственный технический университет.

ISBN 978-5-9221-1575-9

ОГЛАВЛЕНИЕ

Введение

Хорошее начало — половина всего.

Платон

В одном мгновенье — видеть Вечность,
Огромный мир — в зерне песка,
В единой горсти — бесконечность
И небо — в чашечке цветка.

У. Блейк

Данный учебник является частью курса «Дискретная математика», который читался в Таганрогском государственном радиотехническом университете в течение 20 лет (1986–2006). В настоящее время данный курс ставится и читается на основе инновационных, информационных и интеллектуальных технологий обучения в Южном федеральном университете (г. Ростов-на-Дону, г. Таганрог). В учебнике рассматриваются основные положения теории множеств, теории алгоритмов и алгебры логики, теории графов, а также их применение для решения практических задач науки и техники. Для лучшего усвоения материала была использована инновационная методика обучения на основе «решебников». В начале каждого модуля дается краткое изложение теории, затем подробно рассматриваются примеры и задачи с решениями. Также приводятся контрольные задачи, упражнения и глоссарий с пояснением основных терминов. Задачи и упражнения составлены авторами на основе фундаментальных научных исследований в этой области, а также современной учебной литературы. Опыт преподавания в вузах России, США, Германии, Франции и Японии показал эффективность такого метода представления материала.

Теория множеств и теория графов служат базой, на основе которой строится вся современная дискретная математика. Основное влияние дискретная математика оказала на развитие вычислительной техники, микроэлектроники, нанотехнологии, биологии, генетики, информатики и других наук. В настоящее время применение теории множеств и графов происходит повсеместно во всех областях науки и техники, поэтому дискретная математика не только фундамент современной математики, но и основное звено подготовки специалистов XXI века.

Учебник состоит из четырех модулей. В первом модуле авторы рассмотрели вопросы исчисления множеств, выполнения основных операций над множествами, представления упорядоченных множеств, задания отношений и соответствий, описания упорядоченных бесконечных множеств, мультимножеств и нечетких множеств. Теоретический материал модуля 1 состоит из большого числа моделей, аксиом, теорем и формулировок, понятий, используемых в пособии. Каждое новое понятие и положение выделяется курсивом и включается в глоссарий, где дается его пояснение. Авторы стремились показать широкие возможности применения теории множеств, ее универсальность и специализированность для решения задач информатики и вычислительной техники. Методы теории множеств применяются в таких разделах дискретной математики, как комбинаторика, алгебра логики, теория алгоритмов, теория графов, теория автоматов, математическое программирование. Положения теории множеств — это основа курсов «Математическая логика и теория алгоритмов», «Базы данных и СУБД», «Исследование операций», «Методы оптимизации» и др. Студент, обладающий знаниями в области теории множеств, будет подготовлен к эффективной деятельности в науке, бизнесе и на производстве.

Во втором и третьем модулях учебника рассматриваются основные положения теории алгоритмов и алгебры логики, а также их применение для решения практических задач. Авторы разбирают следующие вопросы: основные алгоритмические модели, свойства и классификация алгоритмов, виды универсальных алгоритмов, проблемы временно́й сложности алгоритмов, классы алгоритмов в зависимости от временно́й сложности, основные логические функции и законы алгебры логики, нормальные и совершенно нормальные формы представления булевых функций, проблемы функциональной полноты, проблемы минимизации булевых функций, логические схемы, проблемы представления булевых функций в виде логических элементов.

Основы теории алгоритмов и алгебры логики используются в таких разделах дискретной математики, как комбинаторика, теория графов, теория автоматов, математическое программирование и др. Положения теории алгоритмов являются основой курсов «Исследование операций», «Методы оптимизации», «Теория принятия решений», «Теория игр и комбинаторика» и др.

В четвертом модуле учебника рассматриваются основные положения теории графов, способы задания графов, основные числа графов, эйлеровы и гамильтоновы графы, алгоритмы определения кратчайших путей в графе, а также их применение для решения практических задач информатики и вычислительной техники. Авторы стремились показать широкие возможности применения теории алгоритмов и алгебры логики, их универсальность и специализированность для решения задач науки и техники.

Вместе с теорией множеств математическая логика и теория алгоритмов образуют теоретический фундамент современных вычислительных наук. При этом математическая логика трактуется в широком смысле, включающем в себя и собственно математическую логику, понимаемую как теория формализованных языков, и теорию алгоритмов.

Основная задача учебника — обучение студентов построению моделей множеств, методам доказательств различных тождеств с множествами и, самое главное, методам абстрактного мышления. Студенты должны владеть методами минимизации булевых функций и уметь строить основные схемы алгоритмов решения различных задач науки и техники. Студенты также должны знать способы задания графов, построения графовых моделей, выполнения операций на графах, методы определения кратчайших путей, цепей и циклов в графах, определения эйлеровых и гамильтоновых циклов на графах, определения таких инвариантных чисел на графах, как цикломатическое и хроматическое число, число внутренней и внешней устойчивости, построения деревьев и основные алгоритмы на графах.

Для успешного изучения курса «Дискретная математика» студентам необходимо знание информатики, высшей математики и основ программирования.

Авторы благодарны рецензентам — коллективу кафедры прикладной математики Московского энергетического института (зав. кафедрой — д. т. н., профессор, лауреат премии президента РФ в области образования А.П. Еремеев) и заслуженному деятелю науки РФ, д. т. н., профессору Донского государственного технического университета Ю.О. Чернышеву. Авторы благодарны д. т. н., профессору А.Б. Петровскому, к. т. н., доценту В.И. Полякову за ценные замечания по 1, 3 и 4 модулям данного учебника. Особая благодарность сотрудникам, аспирантам и студентам кафедры САПР за помощь в апробировании материала.

Любые замечания и предложения будут приняты с благодарностью.

Авторы

Цели и задачи преподавания дисциплины «Дискретная математика»

Цель преподавания дисциплины. Предмет курса — современные методы дискретной математики: теория множеств, теория графов и алгебра логики.

Цель дисциплины «Дискретная математика» — изучение теоретических и практических методов дискретной математики, освоение основных понятий и методов теории множеств, способов моделирования и решения основных алгоритмов, а также методов задания и преобразования переключательных функций средствами булевой алгебры.

Содержание курса включает основные сведения и определения о законах и методах теории множеств, а также методах и способах алгебры логики. Программа курса «Дискретная математика» предполагает изучение теоретических основ современной дискретной техники и включает в себя наиболее актуальные и популярные аспекты таких фундаментальных теоретических наук, как теория множеств (конечных, бесконечных и нечетких), алгебра логики и теория алгоритмов. Особое внимание в программе курса уделено рассмотрению вопросов практического применения указанных теоретических положений, что достигается за счет использования для обучения разработанных «решебников», соответствующих стандартам ведущих высших учебных заведений, а также наличием в указанном курсе большого количества примеров и ссылок на реальные примеры применения рассматриваемых теоретических положений.

Задачи изучения дисциплины. В результате изучения дисциплины «Дискретная математика» студенты должны:

- иметь основные представления о задачах теории множеств, а также алгебры логики;
- знать основные законы, тождества и операции над множествами;
- получить навыки практической работы по решению комбинаторно-логических задач;
- уметь строить и преобразовывать переключательные функции алгебры логики;
- использовать методы и алгоритмы теории графов для решения практических задач комбинаторной оптимизации.

В результате изучения данного курса студенты должны знать наиболее популярные и в то же время актуальные задачи оптимизации и принятия решений, а также основные типы алгоритмов, позволяющих решать такие задачи, уметь оценивать их сложность и применимость для решения каждой конкретной проблемы. Студенты также должны представлять достоинства и недостатки рассматриваемых методов, а также уметь самостоятельно разрабатывать новые алгоритмы решения актуальных инженерных проблем, связанных с оптимизацией, прогнозированием, мониторингом и принятием решений. Учащиеся должны в процессе изучения дисциплины научиться применять полученные знания об основах построения дискретной техники для решения актуальных практических задач проектирования, оптимизации и принятия решений, ознакомиться с существующими программными средствами построения математических моделей реальных электрических и коммутационных схем и реализации построенных теоретических моделей.

Курс является базовым для всей специальности САПР. Основные задачи пользователя и разработчика САПР — уметь составлять и решать алгоритмы любых практических оптимизационных задач. При изучении дисциплины существенную помощь окажет освоение автоматизированных обучающих модулей, а также выполнение индивидуальных творческих заданий по разработке алгоритмов и их программной реализации на ЭВМ.

Для проведения практических занятий по данной учебной дисциплине подготовлен и издан ряд учебных пособий («решебников»), включающих в себя необходимый для понимания теоретический материал, большое количество примеров решения практических задач и большой банк контрольных заданий и вопросов для проведения текущего и итогового контроля знаний студентов. Кроме того, для повышения качества преподавания и усвоения учебного материала и эффективности контроля полученных знаний, а также с целью использования в процессе дистанционного и заочного обучения по указанной учебной дисциплине был разработан и успешно опробован на заочном отделении специальности 230104 новый вид контроля знаний на основе тестовых заданий.

Структура учебной программы соответствует аналогичным курсам, входящим в учебные планы ведущих государственных высших учебных заведений США, например, программам курсов «Discrete mathematics. P. 1, 2», «Discrete structures in Computer Science» Мичиганского государственного университета и др.

В ходе изучения дисциплины предусмотрено выполнение индивидуального творческого задания, что позволяет студентам глубже ознакомиться с возможностями современного математического аппарата дискретной техники для решения различных оптимизационных задач, а также изучить различные алгоритмы решения классических NP-полных задач оптимизации. Важный аспект достижения целей настоящей учебной дисциплины — то, что в ходе подготовки и выполнения индивидуальных творческих заданий студенты получают возможность углубить свои знания и практические навыки в области программирования, а также применить полученные знания и навыки на практике в процессе разработки и реализации алгоритмов и программ решения конкретных оптимизационных задач.

МОДУЛЬ 1.
ОСНОВЫ ТЕОРИИ МНОЖЕСТВ (2 КРЕДИТА)

> В начале любая оригинальная теория признается абсурдной, потом — верной, потом — самоочевидной и незначительной и, наконец, столь важной и самобытной, что бывшие критики присваивают ее себе.
>
> *У. Джеймс*

Великий ученый Г. Кантор — основатель теории множеств. Теория множеств в настоящее время стала краеугольным камнем современной дискретной математики, тем базисом, на котором основываются все дальнейшие дисциплины информатики и разрабатываются новые информационные технологии. Теория множеств дала единые методы для изучения конечных и бесконечных систем предметов. Причем множество относится к числу простейших. Определение множества может быть только разъяснено, но не определено. Понятие множества связано с абстракцией. Объединяя предметы в множество и создавая новый предмет, мы игнорируем все свойства множества, зависящие от свойств входящих в него предметов, кроме свойств отличаться от всех других множеств. Это позволяет легко формализовать объект и упростить процесс подготовки реализации задачи на ЭВМ.

Комплексная цель и задачи изучения модуля. Цель модуля 1 — дать представление о фундаментальных понятиях, базовых принципах и законах основного раздела дискретной математики — теории множеств, рассмотреть постановку основных задач и проблем, изучить вопросы методологии решаемой проблемы.

В результате освоения модуля 1 студент должен быть готов продемонстрировать следующие *компетенции* и *уровень подготовки*:

1) знание основных понятий теории множеств;

2) умение применять на практике законы и правила теории множеств, доказывать справедливость тождеств теории множеств;

3) навыки решения практических задач, умение представить решаемую конкретную практическую задачу в виде абстрактной математической модели, пригодной для дальнейшего фор-

мального решения с помощью математического аппарата теории множеств.

Самостоятельная работа предусматривает проработку лекций (1, 2 часа в неделю), тестирование, а также изучение литературы, формулировку цели работы, объекта и задач исследования, методов, источников и средств библиографического поиска, использованных для достижения поставленной цели.

Модуль включает формулирование цели, проблемное изложение программного лекционного материала, тестовые вопросы для самоконтроля и список литературы. В процессе самостоятельного изучения представленных методических материалов происходит формирование указанных компетенций.

Глава 1
ИСЧИСЛЕНИЕ МНОЖЕСТВ

Само собой понятное и очевидное
Не следует определять:
Определение лишь затемнит его.

Б. Паскаль

Множество, определение и способы задания множеств, высказывания и основные операции над высказываниями, элемент множества, включение множеств, кванторы существования и общности

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать способы задания множеств;
- уметь решать задачи на включение и невключение множеств;
- знать понятие кванторов существования и общности;
- уметь строить таблицы истинности высказываний;
- знать основные свойства множеств.

1.1. Понятие множества

Более общего понятия, чем множество, в математике нет. Оно является исходным, первоначальным и, к сожалению, неопределенным понятием.

Основатель теории множеств — немецкий математик Георг Кантор (1845–1918). По его определению, *множество* — это любое объединение в одно целое M определенных, вполне различимых объектов m из нашего восприятия или мысли, которые можно считать элементами из M. Неформально можно сказать, что *множество* — это многое, мыслимое как единое.

Понятию множества в математике соответствуют семейство, совокупность, система, класс, область и т. д. Объекты, составляющие множества, могут иметь любую природу. Например, мы можем говорить о множестве живущих на Земле людей, множестве планет Солнечной системы, множестве букв русского алфавита, множестве целых чисел. Часто объекты объединяются в множество по какому-либо общему признаку. Обозначаются множества прописными латинскими буквами A, B, C и т. д.

Предметы, составляющие исследуемое множество, называются его *элементами*. Например, запись $A = \{x, a, b, c, d\}$ означает, что множество A состоит из элементов x, a, b, c, d. Элементами множества могут быть буквы, цифры, слова, тексты, целые алфавиты, другие множества и т. д. Обычно элементы множества обозначаются строчными латинскими буквами — a, b, c и т. д. Для обозначения принадлежности или непринадлежности элемента множеству используются символы «$\in$» и «$\notin$» соответственно. Запись $x \in A$ означает, что элемент *x принадлежит* множеству A, а $y \notin B$ — элемент *y не принадлежит* множеству B.

По определению, множество состоит из различных элементов. Моделью множества можно считать коробку с расположенными в ней произвольным образом пронумерованными различными цифрами кубиками. Например, множества $A = \{1, 2, 3\}$ и $A = \{2, 3, 1\}$ описывают одно и то же множество. Следовательно, множество характеризуется тем, что все элементы в нем различны, а их местоположение не имеет значения. Фигурные скобки в записи множества обозначают, что элементы объединены в одно целое, например, множество A.

Множества, не имеющие ни одного элемента, называются пустыми. Пустое множество обозначается знаком $\varnothing$. Например, множество людей, живущих на Луне, является пустым, множество целых чисел, для которых выполняется условие «больше 5 и одновременно меньше 3», также является пустым.

Множество, содержащее один элемент, называется одноэлементным.

В данном пособии будем использовать следующие обозначения часто используемых в математике множеств:

$\mathbb{N} = \{1, 2, 3, \ldots\}$ — множество натуральных чисел;

$\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \ldots\}$ — множество целых чисел;

$\mathbb{Q} = \{x/y \mid x, y \in Z, y \neq 0\}$ — множество рациональных чисел;

$\mathbb{R} = \{$все десятичные дроби$\}$ — множество вещественных (действительных) чисел.

С понятием множества тесно связано понятие «высказывание». Определим термин «высказывание». *Высказывание* — это предположение (предложение), которое считается истинным или ложным. Будем обозначать Истина — 1(И), Ложь — 0(Л).

Например, высказывание «два больше одного» — истинно, а высказывание «пять принадлежит множеству отрицательных чисел» — ложно.

Предикатом называется высказывание, содержащее переменные, принимающее значения 1 или 0 в зависимости от значения переменных. Например, высказывание $x^2 = 4$ является предикатом, так как оно истинно при $x = 2$ и ложно во всех остальных случаях.

К основным операциям над высказываниями относятся отрицание, дизъюнкция, конъюнкция, эквивалентность, импликация.

1. Операция отрицания (инверсия). Обозначается следующим образом: $\overline{A}$ — отрицание «A». Читается как «не A».

2. Операция дизъюнкции (логическое сложение, ИЛИ). Обозначается следующим образом: $A \vee B$. Читается как «A или B».

3. Операция конъюнкции (логическое умножение, И). Обозначается следующим образом: $A \,\&\, B$, либо $A \wedge B$. Читается как «A и B».

4. Операция импликации (логическое следствие). Обозначается следующим образом: $A \to B$. Читается как «A влечет B» или «если A, то B».

5. Операция эквивалентности (логическое равенство). Обозначается следующим образом: $A \leftrightarrow B$. Читается как «A равносильно B» или «A эквивалентно B».

Результат выполнения вышеуказанных операций определяется по таблицам истинности. Таблицы истинности этих высказываний приведены в модуле 3 настоящего учебного пособия.

Введем понятие кванторов существования и общности.

Квантор общности читается «для любого» и записывается так: $\forall$. Запись вида $(\forall x \in X)(B(x))$ означает, что для любого элемента x из множества X истинно высказывание $B(x)$ об этом элементе.

Квантор существования читается «существует» и записывается так: $\exists$. Запись вида $(\exists x \in X)(B(x))$ означает, что существует хотя бы один элемент x из множества X, для которого истинно высказывание $B(x)$ об этом элементе.

Множества бывают конечные и бесконечные.

Конечное множество — это такое множество, число элементов которого можно представить натуральным числом. Множество называется *бесконечным*, если оно не является конечным. Например, множество планет Солнечной системы конечно, так как количество планет в этой системе равно 8, а множество целых чисел является бесконечным.

1.2. Способы задания множеств

Существуют два основных способа задания множеств:

1) перечисление (перечислительный способ);

2) описание характеристического свойства (высказывательный способ).

Перечислительный способ состоит в составлении полного списка элементов множества, заключенного в фигурные скобки и применяется только для конечных множеств с небольшим числом элементов.

Например: $A = \{$Меркурий, Венера, Земля, Марс, Юпитер, Сатурн, Уран, Нептун$\}$ — это множество планет Солнечной системы, $B = \{$ФАВТ, РТФ, ФЭМП, ФЭП, ФИБ, ЕГФ$\}$ — это множество факультетов университета. В общем случае этим способом множество A задается следующим образом: $A = \{a_1, a_2, a_3, a_4, \ldots, a_n\}$ или $A = \{a_i\}$, где $i \in I = \{1, 2, 3, 4, \ldots, n\}$.

Высказывательный способ состоит в задании такого свойства, наличие которого у элементов определенного множества является истиной. Запись $A = \{x \in M \mid P(x)\}$ означает, что множество A состоит из элементов некоторого множества M, обладающих свойством P.

Например: запись $B = \{x \in \mathbb{R} \mid (x^4 - 2x^2 - 3 = 0)\}$ означает, что множество B состоит из действительных корней уравнения $x^4 - 2x^2 - 3 = 0$. Это же множество можно также задать перечислительным способом: $B = \{\sqrt{3}, -\sqrt{3}\}$.

Если из контекста понятно, какое множество M рассматривается, то высказывательную форму задания множества можно упростить в виде $A = \{x \mid P(x)\}$. Последняя запись читается так: «Множество A таких элементов x, для которых высказывание $P(x)$ истинно».

Конечное множество может быть задано обоими способами. Выбор способа задания зависит от дальнейшей работы с этим множеством. Бесконечные множества можно задать лишь высказывательным способом. Пустые множества относятся к конечным.

Если A — конечное множество, то через $|A|$ обозначается количество его элементов и эту величину называют *мощностью множества*.

Мощность пустого множества равна нулю. Например, если множество $A = \varnothing$, то его мощность равна нулю ($|\varnothing| = 0$). Мощность одноэлементного множества равна единице. Например, для множества $A = \{9\}$ имеем $|A| = 1$.

Если множества A и B содержат одни и те же элементы и мощности их равны, то эти множества считаются *равными*. Это обозначается так: $A = B$. *Неравные множества* состоят из различных элементов. Если в множестве A есть элементы, не принадлежащие множеству B, либо в множестве B имеются элементы, не принадлежащие множеству A, то множество A не равно множеству B. Неравенство множеств A и B обозначается $A \neq B$.

Из определения множеств следует, что порядок элементов в множестве несуществен. Следовательно, если $A = \{1, 2, 3, 4\}$, а $B = \{4, 3, 2, 1\}$, то $A = B$, т. е. A и B — одинаковые множества или одно и то же множество.

Из определения равенства множеств вытекает, что все пустые множества равны, т. е. существует только одно пустое множество.

Равенство множеств обладает следующими свойствами:
- $A = A$ — рефлексивность;
- если $A = B$, то $B = A$ — симметричность;
- если $A = B$ и $B = C$, то $A = C$ — транзитивность.

Запись $A = \{1, 2, 3, 2\}$ не является множеством, так как содержит повторяющиеся элементы.

Отметим, что множества, которые содержат себя в качестве одного из своих элементов, называются *экстраординарными*. Остальные множества, не относящиеся к ним, называются *ординарными*.

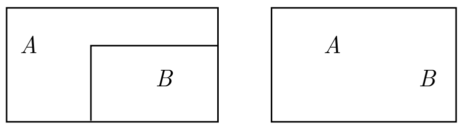
Примером экстраординарного множества может служить множество абстрактных понятий, так как оно (множество) само является абстрактным понятием.

Например, множество планет Солнечной системы, множество букв русского алфавита, множество денежных единиц являются ординарными множествами, так как первое множество не является планетой, второе — буквой, а третье — денежной единицей. Большинство рассматриваемых нами множеств являются ординарными.

Существуют некоторые множества, относительно которых неизвестно, пусты они или нет. Например, неизвестно, имеет ли положительное целочисленное решение уравнение $x^n + y^n = z^n$ при $n > 2$.

1.3. Подмножество

Различают два вида включений — строгое и нестрогое. Множество B является *подмножеством* множества A, если любой элемент множества B принадлежит множеству A. Говорят, что множество B *нестрого включается* в множество A, если выполняется одно из двух условий $(B \subseteq A \vee A = B)$ ($\subseteq$ — знак нестрогого включения). В этом случае также говорят, что множество A включает множество B или множество A является надмножеством множества B. На рис. 1.1 показан пример нестрогого включения. Здесь элементы множеств A, B представляются точками плоскости, расположенными внутри соответствующих прямоугольников.



$B \subseteq A$ $\qquad$ $A = B$

Рис. 1.1. Два случая нестрогого включения множеств

Множество B *строго включается* в множество A, если одновременно выполняются два условия $(B \subseteq A \,\&\, A \neq B)$. В этом случае также говорят, что множество B является истинным подмножеством множества A, и обозначают $A \subset B$, где «$\subset$» — символ строгого включения. Этим подчеркивают, что множество A содержит не только элементы множества B. Если множество B не включается в множество A, пишут $B \not\subset A$.

Теперь равенство множеств можно записать в теоретико-множественном виде:

$$A = B \to A \subseteq B \,\&\, B \subseteq A.$$

Очевидно, что для пустого множества справедливо $\varnothing \subseteq A$, $\varnothing \subset A$.

Приведем основные свойства включения множеств:
- $A \subseteq A$ — рефлексивность;
- $A \subseteq B \,\&\, B \subseteq C \to A \subseteq C$ — транзитивность;
- $A = B \to A \subseteq B \,\&\, B \subseteq A$ — равенство.

Из последнего свойства следует, что для того, чтобы доказать равенство двух множеств, достаточно доказать, что $A \subseteq B$, а затем — что $B \subseteq A$.

Определение истинного подмножества в высказывательной форме запишется следующим образом:

$$A \subset B \to ((\forall x \in A)(x \in B) \,\&\, A \neq B)).$$

Приведем основные свойства строгого включения:
- $A \not\subset A$ — антирефлексивность;

- $\left.\begin{array}{l}(A \subset B \,\&\, B \subseteq C) \to A \subset C \\ (A \subseteq B \,\&\, B \subset C) \to A \subset C\end{array}\right\}$ — транзитивность;
- $A \subset B \to A \neq B$ — неравенство.

Любое непустое множество M включает, по крайней мере, два подмножества — пустое и само себя, т. е. $M \subseteq M$ и $\varnothing \subseteq \subseteq M$. Доказательство этого утверждения следует из определения подмножества. Множества M и $\varnothing$ называют *несобственными подмножествами* множества M, все остальные подмножества множества M называются собственными подмножествами.

Обычно все множества, с которыми имеют дело при рассуждениях, являются подмножествами некоторого универсального множества U.

Множество всех подмножеств множества M называется *семейством подмножеств* множества M и обозначается $\Re(M)$ (читается «пэ от эм»). Из определения семейства следует, что $\varnothing \in \Re(M)$, $M \in \Re(M)$ и если $B \subseteq M$, то $\to B \in \Re(M)$.

Если множество M содержит m элементов, то мощность семейства подмножеств множества M равна 2^m.

Например, подсчитаем, сколько подмножеств имеет конечное множество $A = \{a, b\}$. Так как $m = 2$, то семейство подмножеств содержит $2^m = 4$ подмножества:

$$\{a, b\};\ \{a\},\ \{b\},\ \{\varnothing\}.$$

Для $B = \{a, b, c\}$ семейство подмножеств равно $2^m = 8$.

Если множество содержит n элементов, а его подмножество состоит из k элементов, то число k-элементных подмножеств, называемых сочетанием из n по k, равно

$$C_n^k = \frac{n!}{k!(n-k)!},$$

где $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (n-1) \cdot n$.

Примечания

1. *Пустое множество является единственным подмножеством самого себя*: $\varnothing \subseteq \varnothing$.
2. *Пустое множество не имеет собственных и истинных подмножеств.*
3. *Одноэлементное множество имеет два подмножества — само себя и* $\varnothing$, *т. е. не имеет собственных подмножеств.*

Примеры решения задач

Отметим, что часть примеров, приведенных ниже, разработаны авторами, а часть взята из литературы, приведенной в списке.

Пример 1.1. Задано множество $A = \{a, b, c, d\}$. Какие из следующих высказываний истинны?

а) $a \in A$; в) $d \in A$;

б) $\{a, b\} \in A$; г) $\{a, b, c, d\} \in A$.

Ответ. Высказывания $a \in A$ и $d \in A$ истинны, так как среди элементов множества A есть элементы a и d, высказывания $\{a, b\} \in A$ и $\{a, b, c, d\} \in A$ ложны, так как такие элементы не входят в множество A.

Пример 1.2. Перечислите элементы множеств:

а) $A = \{x \,|\, (\exists y)(y \in \{0, 1, 2\} \,\&\, x = y^3)\}$;

б) $B = \{x \,|\, x \in \mathbb{N} \,\&\, (\exists y)(y \in \mathbb{N} \,\&\, x < y)\}$;

в) $C = \{x \,|\, x \in \mathbb{N} \,\&\, (\forall y)(y \in \mathbb{N} \to x \leqslant y)\}$.

Ответ. а) Элементы множества A могут быть найдены вычислением третьей степени всех возможных значений y. Таким образом, $A = \{0, 1, 8\}$;

б) x и y — элементы множества натуральных чисел $\mathbb{N}$ и, каково бы ни было значение x, всегда можно в множестве $\mathbb{N}$ найти значение y, при котором будет выполняться условие $x < y$;

в) если x и y — элементы множества натуральных чисел $\mathbb{N}$, то условие $x \leqslant$ любого элемента множества $\mathbb{N}$ будет выполняться только для одного значения x, равного 0. Следовательно, $C = \{0\}$.

Пример 1.3. Задать множества высказывательным способом:

а) $A = \{3\}$;

б) $P = \{6, 7, 8, 9, 10\}$;

в) B — множество четных целых чисел.

Ответ. а) Один из возможных вариантов ответа $A = \{x \in \mathbb{N} \,|\, (x + 2 = 5)\}$;

б) один из возможных вариантов ответа $P = \{x \,|\, x \in \mathbb{N} \,\&\, (x > 5 \,\&\, x < 11)\}$;

в) $B = \{x \,|\, x \in \mathbb{Z} \,\&\, (\forall y)(y \in \mathbb{Z} \to x = 2y)\}$.

Пример 1.4. Определите, равны ли множества A и B, C и D, если $A = \{1, 2, 3, 4\}$; $B = \{4, 3, 2, 1\}$; $C = \{2, 3, 4, 5\}$; $D = \{3, 4, 5\}$.

Ответ. $A = B$, так как они содержат одни и те же элементы; $C \neq D$, так как множество C содержит элемент 2, не принадлежащий множеству D.

Пример 1.5. Пусть: $A = \{1, 7, 9, 15\}$; $B = \{7, 9\}$; $C = \{7, 9, 15, 20\}$.

Все ли следующие утверждения истинны?

а) $B \subseteq C$, д) $15 \in C$,

б) $B \subseteq A$, е) $\{7, 9\} \in B$,

в) $B \subset A$, ж) $\{7\} \subset A$,

г) $A \not\subset C$, з) $\varnothing \subseteq C$.

Ответ. Все утверждения, кроме е), истинны.

Пример 1.6. Доказать, что $A \subseteq B$, если $A = \{x \in \mathbb{N} \,|\, x \text{ кратно } 8\}$; $B = \{x \in \mathbb{N} \,|\, x \text{ кратно } 4\}$;

Доказательство. Пусть x — произвольный элемент множества A. Мы должны показать, что x также является элементом множества B, т. е. обладает свойством, характерным множеству B. Это означает, что x должно быть кратным 4. Поскольку x является элементом множества A, он обладает характерным свойством этого множества, следовательно, x кратно 8 и можно записать $x = 8 \cdot m$ для любого целого m. Это же выражение может быть записано в виде $x = 2 \cdot 4 \cdot m$ или $x = 4 \cdot k$, где k — четное целое число. Следовательно, x всегда кратен 4 и поэтому $x \in B$, что и требовалось доказать.

Пример 1.7. Доказать, что $A = B$, если $A = \{x \in \mathbb{N} \,|\, x^2 < 15\}$; $B = \{x \in \mathbb{N} \,|\, 2x < 7\}$;

Доказательство. Чтобы доказать, что $A = B$, необходимо доказать, что $A \subseteq B$, а затем что $B \subseteq A$. Пусть $x \in A$, следовательно, x должно обладать характерным свойством множества A, т. е. должны выполняться условия: $x^2 < 15$ и $x \in \mathbb{N}$. Этим двум условиям удовлетворяют следующие четыре значения x: 0, 1, 2, 3. Таким образом, только эти числа являются элементами множества A. Причем удвоенные значения любого элемента множества A меньше 7, т. е. каждый из них обладает характерным свойством множества B. Следовательно, $A \subseteq B$. Теперь докажем, что $B \subseteq A$. Любой элемент множества B должен быть положительным, а его удвоенное значение меньше 7. Множество B составляют следующие элементы: 0, 1, 2, 3. Квадрат любого из них меньше 15, таким образом, $B \subseteq A$. Итак, доказано, что $A = B$.

Контрольные вопросы

1. Что такое «множество», «элемент множества»?

2. Какие существуют способы задания множеств? Привести примеры.

3. Приведите определение понятий простого и составного логического высказывания.

4. Перечислите основные операции над логическими высказываниями. Приведите примеры.

5. Что представляет собой квантор существования и квантор общности?

6. Какие множества называются равными?

7. Какие подмножества называются несобственными?

8. В чем отличие между «принадлежностью к множеству» и «включением в множество»?

9. Приведите способы задания множеств.

10. Отличаются ли множества $A = \varnothing$ и $A = \{\varnothing\}$ и если да, то в чем состоит отличие.

11. Что такое строгое и нестрогое включение?

12. Как построить семейство подмножеств?

13. Сколько элементов содержит семейство подмножеств?

14. Приведите формулу подсчета числа подмножеств множества.

Задания для самостоятельной работы

1. Задано множество $S = \{0, 4, 12, 31\}$. Какие из следующих высказываний являются истинными?

а) $12 \in S$; б) $(0 + 31) \in S$; в) $(4 + 12) \in S$;

г) $\varnothing \in S$; д) $\{4\} \in S$; е) $\{0, 4, 12, 31\} \in S$.

2. Записать произвольное одноэлементное множество, элементом которого является множество.

3. Привести примеры таких множеств A, B, C и D, что $A \in B$, $B \notin D$, $D \in C$, $A \notin D$, $B \notin C$.

4. Задать каждое из множеств с помощью перечисления:

а) $A = \{x \,|\, x \in \mathbb{R} \,\&\, x^2 < 0\}$;

б) $B = \{x \in \mathbb{N} \,|\, 3 < x \leqslant 7\}$;

в) $D = \{x \,|\, x$ — месяц года, содержащий 30 дней$\}$;

г) $C = \{y \,|\, y$ — факультеты ТРТУ$\}$;

д) $E = \{x \in \mathbb{N} \,|\, \forall y)(y \in \{2, 3, 4, 5\} \to x > y)\}$;

е) $F = \{x \,|\exists y)(\exists z)(y \in \{1, 2\} \,\&\, z \in \{2, 3\} \,\&\, x = y + z)\}$.

5. Задать множества высказывательным способом:

а) $A = \{1, 4, 9, 16\}$;

б) $B = \{2, 3, 5, 7, 11, 17\}$;

в) $C = \{3, 5, 7, 9, 11, 13, 15\}$;

г) $D = \{$февраль$\}$;

д) $F = \{$понедельник, вторник, среда, четверг, пятница$\}$.

6. Запишите таблицы истинности основных операций над высказываниями и поясните их смысл.

7. Определить, равны ли множества A и B:

а) $A = \{a, b, c\}$; $B = \{b, c, a\}$;

б) $A = \{a, b, \{c\}\}$; $B = \{b, c, a\}$;

в) $A = \{\{a, b, c\}\}$; $B = \{a, b, c\}$;

г) $A = \{\{a, b, c\}\}$; $B = \{\{a\}, \{b\}, \{c\}\}$;

д) $A = \{a, b, c\}$; $B = \{\{a\}, \{b\}, \{c\}\}$.

8. Пусть $A = \{x \in \mathbb{N} \,|\, x > 5\}$; $B = \{10, 12, 16, 20\}$; $C = \{x \,|\, (\exists y)(y \in \mathbb{N} \,\&\, x = 2y)\}$. Какое из следующих высказываний истинно?

а) $B \subseteq C$,

б) $A \subseteq C$,

в) $\{11, 12, 13\} \subseteq A$,

г) $\{12\} \in B$,

д) $\{x \in \mathbb{N} \,|\, x < 20\} \not\subset B$,

е) $\{\varnothing\} \subseteq B$,

ж) $B \subset A$,

з) $26 \in C$,

и) $\{11, 12, 13\} \subset C$,

к) $\{12\} \subseteq B$,

л) $5 \subseteq A$,

м) $\varnothing \notin A$.

9. Пусть $A = \{x \in \mathbb{R} \,|\, x^2 - 4x + 3 = 0\}$; $B = \{x \in \mathbb{N} \,|\, 1 \leqslant x \leqslant 4\}$. Доказать, что $A \subset B$.

10. Доказать или опровергнуть, что $A = B$, если $A = \{x \in \mathbb{R} \,|\, 2x = 5\}$; $B = \{x \in \mathbb{R} \,|\, x > 3\}$.

11. Привести пример множества, имеющего только несобственные подмножества.

12. Выбрать из элементов множества $A = \{a, x, \{b, c\}, \varnothing, 5\}$ несобственные элементы.

13. Записать семейство подмножеств пустого множества.

14. Записать семейство всех подмножеств множества $A = \{x, y, z, k\}$ и $B = \{\varnothing, \{\varnothing\}\}$.

15. Задать множества A, B, C, удовлетворяющие следующим условиям:

а) $A \in B,\ \ A \subseteq B$; б) $A \in B,\ \ A \subseteq C$;

в) $A \in B,\ \ A \subseteq B,\ \ A \subseteq C$; г) $A \in B, B \subseteq C$;

д) $A \in B,\ \ A \subset B$; е) $A \subseteq B,\ \ B \subseteq A$.

Множество — это многое, мыслимое как единое.

Глава 2
ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

Действовать без правил — самое трудное и самое утомительное занятие на этом свете.

А. Мандзони

Объединение, пересечение и разность множеств, симметрическая разность множеств, дополнение множества, законы и тождества алгебры множеств, доказательство тождеств с множествами, метод двух включений, метод доказательства от противного, диаграммы Эйлера–Венна

ЦЕЛИ

Освоив эту главу, студенты должны:
- знать основные свойства операций над множествами;
- уметь выполнять операции над множествами;
- уметь доказывать тождества с множествами.

2.1. Объединение множеств

Объединением множеств A и B называют множество C, которое состоит из тех элементов, которые принадлежат или множеству A, или множеству B, или обоим множествам одновременно:

$$C = A \cup B,$$

где $\cup$ — знак объединения.

Теоретико-множественная запись операции объединения имеет следующий вид:

$$x \in C = A \cup B \to x \in A \vee x \in B,$$

что означает: элемент x принадлежит множеству C, если x принадлежит множеству A или множеству B. Если изобразить элементы множеств A и B точками плоскости, расположенными внутри прямоугольника, то $A \cup B$ представится множеством точек плоскости, расположенных внутри заштрихованной фигуры (рис. 2.1). Такое геометрическое представление множеств называется *диаграммой Эйлера–Венна*.

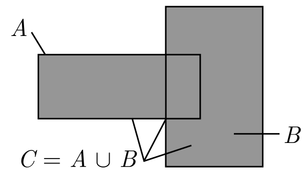


Рис. 2.1. Пример операции объединения множеств A и B

Также можно записать: элемент x не принадлежит множеству C, если он не принадлежит множеству A и не принадлежит множеству B:

$$x \notin C = A \cup B \to x \notin A \,\&\, x \notin B.$$

Например, найдем объединение множеств $A = \{$ЭВМ, студент$\}$ и $B = \{1, 2,$ стол, ЭВМ, стул$\}$. Результатом будет множество $A = A \cup B = \{1, 2,$ стол, ЭВМ, стул, студент$\}$.

Приведем основные свойства операции объединения:

- $A \cup A = A$ — идемпотентность;
- $A \cup B = B \cup A$ — коммутативность;
- $(A \cup B) \cup C = A \cup (B \cup C)$ — ассоциативность;
- $A \cup \varnothing = A$;
- $(A \subseteq A \cup B) \,\&\, (B \subseteq A \cup B)$.

Объединение в множествах является синонимом сложения в арифметике.

Заметим, что можно объединить не только два, но и любое количество множеств.

Объединением n множеств $A_1, A_2, \ldots, A_n$ называется множество, обозначаемое $\bigcup\limits_{i=1}^{n} A_i$, состоящее из всех тех и только тех элементов, которые принадлежат хотя бы одному из множеств A_i:

$$A_1 \cup A_2 \cup A_3 \cup \ldots A_n = \bigcup_{i=1}^{n} A_i.$$

Мощность объединения множеств равна числу содержащихся в нем неповторяющихся элементов. Например, $A = \{1, 2, 3\}$, $B = \{1, 2, 9, 10\}$, $C = A \cup B = \{1, 2, 3, 9, 10\}$ и мощность $|C| = 5$.

2.2. Пересечение множеств

Множество C называется *пересечением множеств* A и B, если оно состоит из тех элементов, которые принадлежат одновременно множеству A и множеству B:

$$C = A \cap B,$$

где $\cap$ — знак пересечения.

На рис. 2.2. приведена диаграмма Эйлера–Венна, иллюстрирующая пересечение множеств A и B. Множество $A \cap B$ изображено заштрихованной фигурой.

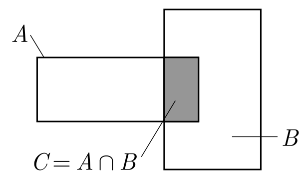


Рис. 2.2. Пример операции пересечения множеств

Теоретико-множественная запись операции пересечения имеет следующий вид:

$$x \in C = A \cap B \to x \in A \,\&\, x \in B,$$

что означает: элемент x принадлежит множеству C, если x одновременно принадлежит множеству A и множеству B.

Соответственно элемент x не принадлежит множеству C, если он не принадлежит множеству A или не принадлежит множеству B:

$$x \notin C = A \cap B \to x \notin A \vee x \notin B.$$

Приведем пример пересечения множеств $A = \{1, 2, 3\}$ и $B = \{3, 4, 5, 6\}$. Результатом выполнения данной операции будет множество $C = A \cap B = \{3\}$.

Свойства операции пересечения:

- $A \cap A = A$ — идемпотентность;
- $A \cap B = B \cap A$ — коммутативность;
- $(A \cap B) \cap C = A \cap (B \cap C)$ — ассоциативность;
- $A \cap \varnothing = \varnothing$;
- $A \cap B = A \leftrightarrow A \subseteq B$;
- $A \cap B \subseteq A \,\&\, A \cap B \subseteq B$.

Пример, когда пересечение двух множеств равняется пустому множеству, показан с помощью диаграммы Эйлера–Венна на рис. 2.3.

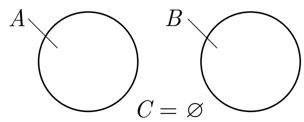


Рис. 2.3. Пример операции пересечения

Отметим, что операция пересечения может выполняться над любым количеством множеств. Пересечением n множеств $A_1, A_2, \ldots, A_n$ называется множество, обозначаемое через $\bigcap\limits_{i=1}^{n} A_i$, состоящее из тех и только тех элементов, которые принадлежат каждому из множеств:

$$\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap \ldots \cap A_n.$$

Примеры пересечения трех и четырех множеств на основе диаграмм Эйлера–Венна показаны на рис. 2.4, 2.5.

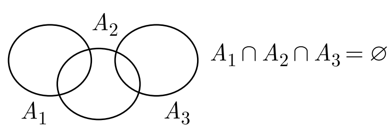


Рис. 2.4. Пример пересечения трех множеств

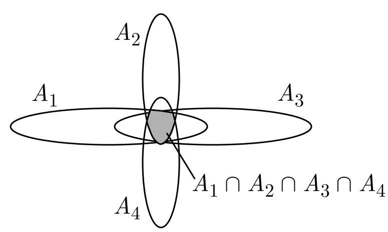


Рис. 2.5. Пример пересечения четырех множеств

Например, пересечением множеств $A = \{1, 2, 3\}$, $B = \{1, 2, 9, 10\}$ является множество $C = \{1, 2\}$ и его мощность $|C| = 2$.

2.3. Разность множеств

В отличие от первых двух операций операция разности применима только к двум множествам.

Множество C называется *разностью множеств* A и B, если C состоит из тех элементов, которые одновременно принадлежат множеству A и не принадлежат множеству B:

$$C = A \backslash B,$$

где $\backslash$ — знак разности.

Теоретико-множественная запись операции разности имеет следующий вид:

$$x \in C = A \backslash B \to x \in A \,\&\, x \notin B,$$

что означает: элемент x принадлежит множеству C, если x принадлежит A и одновременно x не принадлежит множеству B.

Соответственно элемент x не принадлежит множеству C, если он не принадлежит множеству A или принадлежит множеству B:

$$x \notin C = A \backslash B \to x \notin A \vee x \in B.$$

Например, заданы множества $A = \{1, 2\}$ и $B = \{3, 8, 9\}$. Тогда разность между множествами A и B равна $A \backslash B = \{1, 2\}$, а разность между множествами B и A равна $B \backslash A = \{3, 8, 9\}$.

На рис. 2.6 приведены диаграммы Эйлера–Венна, иллюстрирующие операцию разности множеств $A \backslash B$, а также разности множеств $B \backslash A$.

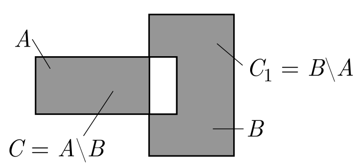


Рис. 2.6. Пример операции разности множеств

Очевидно, что:

$$A \backslash A = \varnothing; \quad A \backslash U = \varnothing; \quad B \backslash \varnothing = B; \quad \varnothing \backslash A = \varnothing.$$

Основные свойства операции разности множеств:
- $(A \backslash B) \backslash C = A \backslash (B \backslash C) = (A \backslash C) \backslash B$ — ассоциативность;
- $A \backslash B \subseteq A$, $B \backslash A \subseteq B$.

Множество C называется *симметрической разностью множеств* A и B, если C состоит из тех элементов и только тех элементов универсального множества U, которые принадлежат множеству A и не принадлежат множеству B или принадлежат множеству B и не принадлежат множеству A:

$$C = A \oplus B,$$

где $\oplus$ — знак симметрической разности.

Теоретико-множественная запись операции симметрической разности имеет следующий вид:

$$x \in C = A \oplus B \to (x \in A \,\&\, x \notin B) \vee (x \in B \,\&\, x \notin A).$$

Например, заданы множества $A = \{1, 2, 3, 8\}$ и $B = \{3, 4, 8, 9\}$. Тогда симметрическая разность A и B равна $A \oplus B = \{1, 2, 4, 9\}$.

2.4. Дополнение множества

Множество $\overline{A}$ называется *дополнением* множества A до некоторого универсального множества U, если оно состоит из элементов, принадлежащих множеству U и не принадлежащих множеству A. Теоретико-множественная запись операции дополнения имеет вид

$$x \in \overline{A} = U \backslash A \to x \in U \,\&\, x \notin A,$$
$$x \notin \overline{A} = U \backslash A \to x \notin U \vee x \in A.$$

На рис. 2.7 приведена диаграмма Эйлера–Венна, иллюстрирующая операцию дополнения множества A. Множество точек, расположенных в заштрихованной фигуре, образует множество $\overline{A}$.

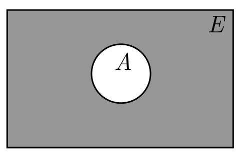


Рис. 2.7. Пример дополнения множества A до универсального множества U

Основные свойства операции дополнения множества:
- $\overline{A} = A$,
- $A \cup \overline{A} = U$,
- $A \cap \overline{A} = \varnothing$,
- $U = \varnothing$,
- $\overline{\varnothing} = U$.

2.5. Тождества алгебры множеств

Применяя операции объединения и пересечения, разности и дополнения к множествам, можно составить из множеств алгебраические выражения.

Если два или несколько алгебраических выражений над одними и теми же множествами представляют одно и то же множество, то эти выражения можно приравнять друг к другу, получив алгебраическое тождество.

Основные тождества алгебры множеств:

Пусть A, B и C — произвольные подмножества семейства $\Re(U)$.

- $\left.\begin{array}{l} A\cap(B\cap C)=(A\cap B)\cap C=A\cap B\cap C \\ A\cup(B\cup C)=(A\cup B)\cup C=A\cup B\cup C \end{array}\right\}$ — ассоциативность;
- $\left.\begin{array}{l} A\cap B=B\cap A \\ A\cup B=B\cup A \end{array}\right\}$ — коммутативность;
- $\left.\begin{array}{l} A\cap(B\cup C)=(A\cap B)\cup(A\cap C) \\ A\cup(B\cap C)=(A\cup B)\cap(A\cup C) \end{array}\right\}$ — дистрибутивность;
- $\left.\begin{array}{l} \overline{A\cup B}=\overline{A}\cap\overline{B} \\ \overline{A\cap B}=\overline{A}\cup\overline{B} \end{array}\right\}$ — закон де Моргана;
- $\left.\begin{array}{l} A\cap(A\cup B)=A \\ A\cup(A\cap B)=A \end{array}\right\}$ — закон поглощения;
- $\left.\begin{array}{l} A\cap A=A \\ A\cup A=A \end{array}\right\}$ — закон тавтологии;
- $\left.\begin{array}{l} A\cap\varnothing=\varnothing \\ A\cup\varnothing=A \end{array}\right\}$;
- $A\backslash B=A\backslash(A\cap B)$;
- $A\backslash B=A\cap\overline{B}$;
- $A\backslash(A\backslash B)=A\cap B$;
- $A\cap(B\backslash C)=(A\cap B)\backslash(A\cap C)$.

Приведем формулу включений–исключений, которая позволяет вычислять мощности объединения двух множеств:

$$|A\cup B|=|A|+|B|-|A\cap B|.$$

Например, заданы множества $A=\{1,2,3,8\}$ и $B=\{3,4,5,8,9\}$. Тогда $|A|=4$, $|B|=5$, $|A\cap B|=2$, $|A\cup B|=4+5-2=7$.

2.6. Доказательства тождеств с множествами

Введем понятие *записи*. Например, выражение $A\cap(B\backslash C)$ является записью. Число букв и знаков операций является *длиной* записи. В приведенном примере длина записи равна 5. Отметим, что различные по длине записи могут соответствовать одним и тем же множествам. Например, запись $A\cap(B\backslash C)$ и запись $(A\cap B)\backslash(A\cap C)$ равны, хотя имеют разные длины (5 и 7).

При построении логических устройств необходимо применять записи минимальной длины. Для их нахождения используются доказательства тождеств с множествами.

Рассмотрим способы доказательства равенств с множествами. Равенство с множествами имеет две формы:

1) $E(A, B, C \ldots) = F(A, B, C \ldots)$. E и F — высказывания над множествами A, B, C и т. д. Необходимо доказать или опровергнуть, что они равны.

2) $E(A, B, C \ldots) = \varnothing$. Доказать или опровергнуть, что высказывание равно $\varnothing$.

Рассмотрим методы доказательства тождеств с множествами. Здесь используется метод двух включений (взаимного включения), который основан на свойстве включения множеств:

$$E = F \to E \subseteq F \,\&\, F \subseteq E.$$

Причем доказательство включения $E \subseteq F$ называется необходимостью, а доказательство включения $F \subseteq E$ — достаточностью.

Например, необходимо доказать или опровергнуть справедливость тождества $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$. Левую часть тождества обозначим E, а правую F. Тогда нам необходимо доказать или опровергнуть, что

$$E \subseteq F \,\&\, F \subseteq E.$$

1. Докажем необходимость: $E \subseteq F$.

Для доказательства этой части предполагается, что существует какой-то элемент, принадлежащий множеству E, и далее путем различных преобразований делается попытка доказать, что этот элемент принадлежит F:

$$\begin{aligned} &a \in E \to a \in [(A \cup B) \cap C] \to a \in (A \cup B) \,\&\, a \in C \to \\ &\quad \to (a \in A \vee a \in B) \,\&\, a \in C \to a \in (A \cap C) \,\&\, a \in (B \cap C) \to \\ &\qquad \to a \in [(A \cap C) \cup (B \cap C)] \to a \in F. \end{aligned}$$

Следовательно, мы доказали, что E включается в F.

2. Докажем теперь достаточность, т. е. докажем, что если $a \in F$, то $a \in E$:

$$\begin{aligned} &a \in F \to a \in [(A \cap C) \cup (B \cap C)] \to \\ &\quad \to a \in (A \cap C) \vee a \in (B \cap C) \to a \in A \,\&\, a \in C \vee a \in B \,\&\, a \in C \to \\ &\qquad \to a \in (A \cup B) \,\&\, a \in C \to a \in [(A \cup B) \cap C] \to a \in E. \end{aligned}$$

Следовательно, мы доказали, что F включается в E.

Значит, $E = F$ и исходное тождество справедливо.

Для доказательства тождеств с множествами, записанными во второй форме, используется метод от противного. Например, докажем справедливость тождества

$$A\backslash[(A\cap B)\cup(A\backslash B)]=\varnothing.$$

Метод от противного предполагает, что это выражение не равно $\varnothing$, т. е. существует какой-то элемент, принадлежащий этому выражению. Здесь пытаются найти противоречие.

Обозначим левую часть тождества через E и будем считать, что она не равна пустому множеству. Тогда существует хотя бы один элемент, принадлежащий $\mathbb{E}$:

$$\begin{aligned}\mathbb{E}\neq\varnothing\to\exists a\in E\to a\in A\,\&\,a\notin[(A\cap B)\cup(A\backslash B)]\to\\ \to a\in A\,\&\,(a\notin(A\cap B)\,\&\,a\notin(A\backslash B))\to\\ \to\underline{a\in A}\,\&\,(\underline{a\notin A}\,\&\,a\notin B)\,\&\,(a\notin A\vee a\in B).\end{aligned}$$

Таким образом, мы получили противоречие, когда элемент a одновременно принадлежит и не принадлежит множеству A. Значит, наше первоначальное предположение неверно и исходное тождество справедливо, т. е. равно $\varnothing$.

Для множеств небольшого размера используют геометрический метод доказательства тождеств, основанный на применении диаграмм Эйлера–Венна. Для доказательства тождества геометрическим методом необходимо построить диаграммы Эйлера–Венна для анализируемых записей. На одной из них — выделить точки плоскости, представляющие множество левой части тождества, а на другой — точки плоскости, представляющие множество правой части тождества. Если фигуры на обеих диаграммах одинаковы, то тождество справедливо.

Примеры решения задач

Пример 2.1. Пусть $A=\{1,3,5,7,9\}$ и $B=\{3,5,6,10,11\}$. Множества A и B являются элементами семейства $\Re(\mathbb{N})$, где $\mathbb{N}$ — множество натуральных чисел. Найти $A\cup B$, $A\cap B$, $A\backslash B$, $\overline{A}$.

Ответ. $A\cup B=\{1,3,5,7,9,10,11,6\}$; $A\cap B=\{3,5\}$; $A\backslash B=\{1,7,9\}$; $\overline{A}=\{2,4,6,8,10,11,12,\ldots\}$. (Множество $\mathbb{N}$ в данной задаче является универсумом.)

Пример 2.2. Пусть $A=\{x\in\mathbb{N}\,|\,x$ — четное число$\}$, $B=\{x\,|\,(\exists y)(y\in\mathbb{N}\,\&\,x=2y+1)\}$, $C=\{x\,|\,(\exists y)(y\in\mathbb{N}\,\&\,x=4y\}$.

Найти $A\cup B$, $A\cap B$, $A\backslash B$, $A\cup C$, $A\backslash C$.

Решение. Множества A, B, C являются подмножествами множества $\mathbb{N}$. Множество B состоит из нечетных натуральных чисел, а множество A — из четных натуральных чисел, следовательно, $A \cap B = \varnothing$, $A \cup B = \mathbb{N}$, $A \backslash B = A$. Элементы множества C — натуральные числа, кратные четырем; следовательно, множество C является собственным подмножеством множества A, а $A \cup C = A$, $A \backslash C = \{x \in \mathbb{N} \,|\, (\exists y)(y \in \mathbb{N} \,\&\, x = 4y + 2)\}$.

Ответ. $A \cup B = \mathbb{N}$, $A \cap B = \varnothing$, $A \backslash B = A$, $A \cup C = A$, $A \backslash C = \{x \in \mathbb{N} \,|\, (\exists y)(y \in \mathbb{N} \,\&\, x = 4y + 2)\}$.

Пример 2.3. Докажите, что для записи последовательности операций объединения и пересечения необходимы круглые скобки, если количество операций больше одной.

Решение. Пусть имеется последовательность операций над произвольными множествами A, B, C — $A \cap B \cup C$, которую можно рассматривать как $A \cap (B \cup C)$ либо как $(A \cap B) \cup C$. Покажем, что последние два множества не равны.

Предположим, что $A = \varnothing$, а $C \neq \varnothing$.

Тогда $(A \cap B) \cup C = (\varnothing \cap B) \cup C = \varnothing \cup C = C$, в то время как $A \cap (B \cup C) = \varnothing \cap (B \cup C) = \varnothing$.

Ответ. Таким образом, доказано, что $(A \cap B) \cup C \neq A \cap (B \cup C)$.

Пример 2.4. Доказать, что $A \subseteq B \to \overline{B} \subseteq \overline{A}$.

Решение. Пусть A и B — подмножества некоторого универсума E и $A \subseteq B$, тогда справедлива следующая последовательность утверждений: $(\forall x \in E)(x \in A \to x \in B)$; $(\forall x \in E)(x \notin B \to x \notin A)$; $(\forall x \in E)(x \in \overline{B} \to x \in \overline{A})$, значит $\overline{B} \subseteq \overline{A}$.

Ответ. Мы доказали, что $A \subseteq B$ влечет $\overline{B} \subseteq \overline{A}$.

Пример 2.5. Доказать, используя метод взаимного включения, закон дистрибутивности пересечения относительно объединения:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

Для доказательства этого тождества необходимо доказать, что $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$, а затем что $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. Докажем первое включение, показывая истинность последовательности импликаций.

Необходимость.

$(\forall x \in E)(x \in (A \cap (B \cup C)) \to x \in A \,\&\, (x \in B \vee x \in C) \to$

(импликация истинна на основе определения операций объединения и пересечения);

$x \in A \,\&\, x \in B \vee x \in A \,\&\, x \in C \to x \in (A \cap B) \vee x \in (A \cap C) \to$

(использован дистрибутивный закон);

$$x \in ((A \cap B) \cup (A \cap C)) \to A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$$

(использовано определение включения в высказывательной форме).

Таким образом, доказано прямое включение множеств (необходимость).

Достаточность. Теперь докажем обратное включение:

$$(\forall x \in E)\big(x \in ((A \cap B) \cup (A \cap C))\big) \to x \in (A \cap B) \vee x \in (A \cap C) \to$$
$$x \in A \,\&\, x \in B \vee x \in A \,\&\, x \in C \to$$
$$x \in A \,\&\, (x \in B \vee x \in C) \to$$
$$x \in A \,\&\, x \in B \cup C \to x \in (A \cap (B \cup C)) \to$$
$$(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C).$$

Обратное включение также доказано.

Ответ. Таким образом, дистрибутивность пересечения относительно объединения доказана.

Пример 2.6. Доказать, используя геометрический метод, закон дистрибутивности объединения относительно пересечения:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$$

Для доказательства заданного тождества отметим на диаграмме рис. 2.8, *а* множество точек, соответствующих множеству $A \cup (B \cap C)$, а на диаграмме рис. 2.8, *б* — множество точек, соответствующих множеству $(A \cup B) \cap (A \cup C)$.

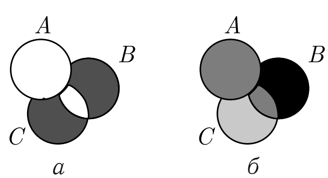


Рис. 2.8. Пример диаграммы Эйлера–Венна

На рис. 2.8, *а* светлой штриховкой отмечено множество $B \cap C$, а серой штриховкой — множество A, множество $A \cup (B \cap C)$ на диаграмме представлено фигурой, представляющей собой объединение фигур, заштрихованных вышеуказанным образом.

На рис. 2.8, *б* множество A заштриховано серым, множество B — черным, а множество C — белым цветом, тогда объединение множеств $A \cup B$ представляет собой фигуру, объединяющую серый и черный круги, объединение множеств $A \cup C$ — фигуру, объединяющую серый и белый круги, а пересечение множеств $A \cup B$ и $A \cup C$ представлено множеством A и множеством точек внутри фигуры, имеющей одинаковую с множеством A штриховку.

Ответ. Сравнивая эти два рисунка, можно сделать вывод, что эти множества равны, следовательно, тождество доказано.

Пример 2.7. Доказать, используя метод от противного, истинность тождества

$$A \cap (B \backslash A) = \varnothing.$$

Решение. Предположим, что $A \cap (B \backslash A) \neq \varnothing$, т. е. существует элемент x, принадлежащий множеству в левой части тождества. Покажем с помощью последовательных импликаций, что наше предположение ложно:

$$(\exists x)(x \in (A \cap (B \backslash A)) \to x \in A \,\&\, x \in (B \backslash A) \to$$

$$x \in A \,\&\, (x \in B \,\&\, x \notin A) \to \underline{x \in A} \,\&\, x \in B \,\&\, \underline{x \notin A} \to \varnothing \,\&\, x \in B \to \varnothing$$

(использовано свойство коммутативности)

Ответ. Таким образом, исходное тождество доказано.

Контрольные вопросы

1. Что такое взаимное включение множеств и в каком случае оно существует?

2. Что называется объединением, пересечением, разностью и дополнением множеств?

3. В каком случае объединение, пересечение и разность двух множеств равны пустому множеству?

4. Как определяется симметрическая разность множеств?

5. Приведите примеры множеств:

а) объединение которых равно их пересечению;

б) пересечение которых равно $\varnothing$, а их разность не является пустым множеством.

6. Приведите основные тождества алгебры множеств.

7. Поясните операцию дополнения множеств.

8. Какие методы доказательства тождеств с множествами вам известны?

9. Что представляет из себя метод доказательства тождеств с множествами от противного?

10. На чем основан метод взаимного включения?

11. На чем основан геометрический метод доказательства тождеств с множествами?

Задания для самостоятельной работы

1. Пусть множества A, B и C являются подмножествами множества $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $A = \{1, 2, 3, 4, 5, 10\}$, $B = \{2, 4, 7, 8, 9\}$, $C = \{5, 8, 10\}$.

Найти: $A \cup B$, $A \backslash C$, $\overline{B} \cap (A \cup C)$.

2. Пусть $A = \{x \,|\, x$ — женское имя$\}$; $B = \{$Мария, Иван, Петр, Иванов$\}$; $C = \{x \,|\, x$ — фамилия$\}$.

Найти: $A \cap B$, $A \cap C$, $B \cap C$.

3. Показать, что для записи последовательности операций объединения и разности необходимы круглые скобки.

4. Показать на диаграммах Эйлера–Венна справедливость утверждения задания 3.

5. Доказать, что $B \subseteq A \,\&\, C \subseteq A \Rightarrow B \cup C \subseteq A$.

6. Выполнить разбиения множества B на 5 классов: $B = \{a, b, c, 1, 2, 3\}$.

7. Для произвольных множеств $A, B, C \in \Re(E)$ доказать или опровергнуть следующие тождества методом включения множеств:

а) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$;
б) $\overline{A \cup B} = \overline{A} \cap \overline{B}$;
в) $\overline{A \cap B} = \overline{A} \cup \overline{B}$;
г) $A \backslash B = A \backslash (A \cap B)$;
д) $A \backslash B = A \cap \overline{B}$;
е) $A \backslash (A \backslash B) = A \cap B$;
ж) $A \cap (B \backslash C) = (A \cap B) \backslash (A \cap C)$;
з) $(A \cup B) \cap (B \cup C) \cap (A \cup C) = (A \cap B) \cup (B \cap C) \cup (A \cap C)$;
и) $A \backslash (B \cup C) = (A \backslash B) \cap (A \backslash C)$;
к) $A \backslash (B \cap C) = (A \backslash B) \cup (A \backslash C)$;
л) $A \backslash (B \backslash C) = (A \backslash B) \cup (A \cap B \cap C)$;
м) $A \cap (B \backslash C) = (A \cap B) \backslash (A \cap C)$;
н) $A \cup (B \backslash C) = (A \cup B) \backslash (A \cup C)$.

8. Для произвольных $A, B, C \in \Re(E)$ доказать или опровергнуть следующие тождества методом от противного:

а) $(A \backslash (A \backslash B)) \backslash (A \cap B) = \varnothing$;
б) $A \cap (B \backslash A) = \varnothing$;
в) $(A \cap C) \backslash (C \backslash (C \backslash A) = \varnothing$;
г) $(\overline{A \backslash C}) \backslash (A \cap \overline{C}) = \varnothing$;
д) $(\overline{A \cup B}) \backslash (\overline{A} \cap \overline{B}) = \varnothing$;
е) $(\overline{A \backslash B}) \backslash (\overline{A} \cup (A \cap B)) = \varnothing$;
ж) $((A \cap B) \cup (A \cap \overline{B})) \backslash A = \varnothing$.

9. Докажите, что $B \cup C = \varnothing$, если $B = \varnothing$ и $C = \varnothing$.

10. Покажите на примере, что выражение $A \cup B \cap C$ требует использования круглых скобок.

11. Докажите, что если $B \subseteq A \,\&\, C \subseteq A$, то $B \cup C \subseteq A$, и если $A \subseteq B \,\&\, A \subseteq C$, то $A \subseteq B \cap C$.

12. Докажите, что если $C \subseteq A$, то $B \cap C \subseteq A$ и $C \subseteq A \cup B$.

13. Докажите, $\Re(A) \cap \Re(B) = \Re(A \cap B)$.

14. Справедливо ли тождество $\Re(A) \cup \Re(B) = \Re(A \cup B)$? Если нет, то приведите пример.

15. Справедливо ли тождество $\{x \mid x \in \mathbb{Z} \,\&\, x = 2m\} \cap \{x \mid x \in \in \mathbb{Z} \,\&\, x = 3m\} = \{x \mid x \in \mathbb{Z} \,\&\, x = 6m\}$, где $m \in \mathbb{N}$?

16. Опровергните следующее утверждение: если $A \cap B = A \cap \cap C$, то $B = C$.

17. Пусть:

$\mathbb{I}$ — множество целых чисел, $\mathbb{I} = \{\ldots -1, 0, 1, \ldots\}$;

$\mathbb{N}$ — множество положительных целых чисел, $\mathbb{N} = = \{0, 1, 2, \ldots\}$;

$\mathbb{N}_p$ — множество отрицательных целых чисел, $\mathbb{N}_p = = \{\ldots, -2, -1, 0\}$;

$\mathbb{E}$ — множество четных чисел;

$\mathbb{P}$ — множество простых чисел.

Найдите: $\mathbb{N} \cap \mathbb{N}_p$, $\mathbb{I} \backslash \mathbb{N}$, $\mathbb{I} \backslash \mathbb{N}_p$, $\mathbb{N} \cup \mathbb{N}_p$, $\mathbb{I} \backslash \mathbb{E}$, $\mathbb{E} \cap \mathbb{P}$.

18. Доказать, что $A = B$, следуя каждому из следующих условий:

а) $A \backslash B = B \backslash A$;

б) $A \cup B = B \cap A$;

в) $A \cup C = B \cup C \,\&\, A \cap C = B \cap C$.

19. Доказать, что $A \subseteq \overline{A}$, если только $A = \varnothing$ и $\overline{A} \subseteq A$.

20. Используя диаграммы Венна, рассмотрите совместимость следующих утверждений:

а) $(A \cap B) \cup C = A \backslash B$ и $C \cap A = B \cap A$;

б) $(A \backslash (B \backslash C)) \subseteq C \cup B$ и $A \cap B \cap C = \varnothing$ и $C \backslash B \subseteq A$;

в) $(B \backslash A) \cap C \neq \varnothing$ и $C \backslash A \subseteq C \backslash B$.

21. Докажите следующие тождества:

а) $(A \cup B) \cap (C \cup D) = (A \cap C) \cup (A \cap D) \cup (B \cap C) \cup (B \cap D)$;

б) $A \backslash B = \overline{A} \cup (A \cap B)$;

в) $A \backslash (A \backslash (A \backslash B)) = A \backslash B$;

г) $A \backslash (A \backslash (A \backslash (A \backslash B))) = A \cap B$;

д) $A_1 \cup A_2 \cup \ldots \cup A_n = (A_1 \backslash A_2) \cup (A_2 \backslash A_3) \cup \ldots \cup (A_{n-1} \backslash A_n) \cup \cup (A_n \backslash A_1) \cup (A_1 \cap A_2 \cap \ldots \cap A_n) = A_1 \cup (A_2 \backslash A_1) \cup (A_3 \backslash (A_1 \cup \cup A_2)) \cup (A_4 \backslash (A_1 \cup A_2 \cup A_3)) \cup \ldots \cup (A_n \backslash (A_1 \cup A_2 \cup \ldots \cup A_{n-1}))$;

е) $A_1 \cap A_2 \cap \ldots \cap A_n = A_1 \backslash [(A_1 \backslash A_2) \cup (A_1 \backslash A_3) \cup \ldots \cup$
$\cup\, (A_1 \backslash A_n)]$.

22. Упростите следующие выражения:

а) $\overline{\overline{(A \cap B) \cup C}} \cap \overline{\overline{B}}$;

б) $((A \cup B) \cap \overline{A}) \cup \overline{(A \cap B)}$.

На основе операций над множествами можно составлять алгебраические выражения.

Доказательство тождеств с множествами позволяет получать записи минимальной длины для построения эффективных логических схем.

Глава 3

УПОРЯДОЧЕННЫЕ МНОЖЕСТВА

Условность нужно соблюдать
в частном, а порядок в общем.

Ж. Бернарден

Упорядоченное множество (*кортеж*), *равенство кортежей*, *декартово произведение множеств*, *степень множества*, *операции над упорядоченными множествами*, *график*, *диагональ*, *инвертирование*, *проектирование и композиция упорядоченных множеств и графиков*, *функциональные и инъективные графики*

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать определение упорядоченного множества (кортежа);
- уметь выполнять операции над кортежами;
- знать правила выполнения совместных операций над кортежами и множествами;
- уметь выполнять операции инверсии и композиции;
- уметь доказывать тождества с кортежами;
- иметь понятие о графиках и их проекциях на заданные оси;
- знать основные свойства графиков.

3.1. Кортеж (упорядоченное множество)

Кортеж, как и множество, является исходным неопределенным понятием. Синонимами термину «кортеж» являются «вектор» и «набор». Кортеж обозначается строчными греческими буквами. Компоненты кортежа — строчными латинскими буквами.

$$\alpha = \langle 2, 2, 3, 4, 4\rangle \ \text{ — пример кортежа.}$$

В этом примере кортеж α состоит из пяти компонент: «2», расположенной на первом месте; «2», расположенной на втором месте; «3», расположенной на третьем месте; «4», расположенной на четвертом месте; «4», расположенной на пятом месте. Т. е., в отличие от множества, кортеж может иметь повторяющиеся элементы, но все эти элементы различны. Число компонент кортежа называется его *длиной*. Длиной кортежа может быть любое целое неотрицательное число. *Компонента кортежа* эквивалентна элементу множества. Кортеж α длины S записывается $\alpha = \langle a_1, a_2, a_3, \ldots, a_S \rangle$, т. е. $|\alpha| = S$.

Компоненты кортежа могут обозначать любые понятия, объекты, в том числе элементы множества или кортежа. Например, $\alpha = \langle 3, \varnothing, 2x-1, \{2,1\}, \langle 3,7 \rangle \rangle$ — кортеж длины 5. Первая компонента — число 3, вторая — пустое множество, третья — многочлен, четвертая — множество и пятая — кортеж длины 2.

В отличие от элементов множества, компоненты кортежа могут частично или полностью совпадать. Например, $\beta = \langle 4,4,4,4,4 \rangle$ — кортеж длины 5. Кортеж длины 2 называется двойка или пара, длины 3 — тройка и так далее. Наряду с кортежами длины $2, 3, 4, \ldots$ существуют кортежи длины 1. Например, $\gamma = \langle 6 \rangle$.

Кортеж, который не содержит компонентов в своем составе, называется пустым кортежем и обозначается $\alpha = \langle\ \rangle$. Длина этого кортежа равна нулю.

Говорят, что два кортежа равны ($\alpha = \beta$), если α и β имеют одинаковую длину и каждая компонента кортежа α совпадает с ссответствующей компонентой кортежа β.

Кортежи α и β неравны ($\alpha \neq \beta$), когда они имеют разную длину или отдельные компоненты кортежа α не совпадают с соответствующими компонентами кортежа β.

Для любых кортежей α, β, γ справедливы утверждения:

- если $\alpha = \beta$, то $\beta = \alpha$,
- если $\alpha = \beta$ и $\beta = \gamma$, то $\alpha = \gamma$.

Например, два кортежа $\alpha = \langle 3,5 \rangle$ и $\beta = \langle 5,3 \rangle$ не равны между собой, а кортежи $\alpha = \langle 9,10 \rangle$ и $\beta = \langle 9,10 \rangle$ — равны.

В отличие от множеств, порядок следования компонент кортежа существенен. Отметим, что иногда кортеж называют упорядоченным множеством. Теоретико-множественная запись равенства кортежей имеет следующий вид:

для кортежей длины 2: $\alpha = \langle a,b \rangle$, $\beta = \langle c,d \rangle$, $\alpha = \beta \to a = c \,\&\, b = d$;

для кортежей длины 3: $\alpha = \langle a_1,a_2,a_3 \rangle$, $\beta = \langle b_1,b_2,b_3 \rangle$, $\alpha = \beta \to a_1 = b_1 \,\&\, a_2 = b_2 \,\&\, a_3 = b_3$.

Кортеж α называют кортежем над множеством M, если каждая компонента кортежа α принадлежит M. Согласно определению пустого кортежа, данный кортеж является кортежем над любым множеством, в том числе и над пустым множеством.

3.2. Декартово произведение

Декартовым произведением множеств A и B называется множество, состоящее из всех тех и только тех пар, т. е. кортежей длины 2, первая компонента которых принадлежит множеству A, а вторая — множеству B.

$A \times B$ — декартово произведение множеств A и B.

Из определения этого произведения следует, что для произвольного кортежа длины 2, например, $\langle x_1, x_2\rangle$, принадлежащего декартову произведению множеств, истинно высказывание

$$\langle x_1, x_2\rangle \in A \times B \to x_1 \in A \,\&\, x_2 \in B.$$

В том случае, если кортеж $\langle y_1, y_2\rangle$ не принадлежит декартову произведению, истинно высказывание

$$\langle y_1, y_2\rangle \notin A \times B \to y_1 \notin A \vee y_2 \notin B.$$

Например, заданы множества $A = \{a, b\}$, $B = \{c, d, e\}$. Тогда декартово произведение этих множеств равно $A \times B = \{\langle a, c\rangle, \langle a, d\rangle, \langle a, e\rangle, \langle b, c\rangle, \langle b, d\rangle, \langle b, e\rangle\}$.

Кортеж длины 2 $\langle a, b\rangle$ можно изобразить на координатной плоскости точкой, абсциссой которой является 1-й элемент кортежа, а ординатой — 2-й элемент кортежа.

Например, заданы множества $A = \{2, 3\}$, $B = \{1, 4\}$. Тогда $A \times B = \{\langle 2, 1\rangle, \langle 2, 4\rangle, \langle 3, 1\rangle, \langle 3, 4\rangle\}$, $B \times A = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 4, 2\rangle, \langle 4, 3\rangle\}$ (рис. 3.1).

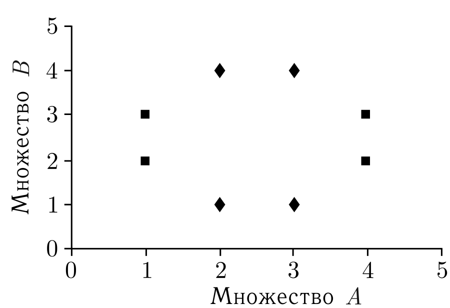


Рис. 3.1. Пример декартова произведения двух множеств ($B \times A$ — точки ■; $A \times B$ — точки ♦)

Аналогичным образом определяется декартово произведение трех, четырех и более множеств.

Декартовым произведением трех множеств A, B, C называется множество, состоящее из всех тех кортежей длины 3, 1-й элемент которых принадлежит множеству A, 2-й — множеству B, 3-й — множеству C. Например, если $A = \{2, 3\}$, $B = \{a, b\}$, $C = \{x, y\}$, то $A \times B \times C = \{\langle 2, a, x\rangle, \langle 2, a, y\rangle, \langle 2, b, x\rangle, \langle 2, b, y\rangle, \langle 3, a, x\rangle, \langle 3, a, y\rangle, \langle 3, b, x\rangle, \langle 3, b, y\rangle\}$.

Из определения декартова произведения следует, что $A \times B$ равно $\varnothing$, если $A = \varnothing$ или $B = \varnothing$:

$$A \times B = \varnothing \to A = \varnothing \vee B = \varnothing.$$

По аналогии можно утверждать, что произведение нескольких множеств равно пустому множеству тогда и только тогда, когда хотя бы одно из этих множеств пусто:

$$A_1 \times A_2 \times A_3 \times \ldots \times A_n = \\ = \varnothing \to A_1 = \varnothing \vee A_2 = \varnothing \vee A_3 = \varnothing \vee \ldots \vee A_n = \varnothing.$$

Исходя из свойств кортежа справедливо следующее: $A \times B \neq B \times A$.

Операцию декартова произведения используют для определения степеней множества.

Пусть M — произвольное множество. Назовем S-й степенью множества M и обозначим M^S прямое декартово произведение S-одинаковых множеств, равных M. Для $S = 2, 3, 4 \ldots$

$$M^S = \underbrace{M \times M \times M \times \ldots \times M}_{S \text{ раз}}.$$

По определению считают, что $M^1 = M$, $M^0 = \langle\ \rangle$. При $S \geqslant 2$ множество M^S является множеством всех кортежей длины S над множеством M.

Если M содержит n элементов и $S \geqslant 2$, то число элементов множества M^S равно n^S, где n — число элементов множества M. Например, $M=\{a, b\}$, $M^0=\langle\ \rangle$, $M^1 = M$, $M^2 = \{\langle a, a\rangle, \langle a, b\rangle, \langle b, a\rangle, \langle b, b\rangle\}$, $M^3=\{\langle a, a, a\rangle, \langle a, a, b\rangle, \langle a, b, a\rangle, \langle a, b, b\rangle, \langle b, a, a\rangle, \langle b, a, b\rangle, \langle b, b, a\rangle, \langle b, b, b\rangle\}$.

Декартово произведение двух множеств обладает следующими свойствами:

- $X \times Y \neq Y \times X$ — некоммутативность;
- $X \times (Y \times Z) = (X \times Y) \times Z = X \times Y \times Z$ — ассоциативность;

- $\left.\begin{array}{l} X\times(Y\cup Z)=(X\times Y)\cup(X\times Z)\\ (Y\cup Z)\times X=(Y\times X)\cup(Z\times X)\end{array}\right\}$ — дистрибутивность по объединению;
- $\left.\begin{array}{l} X\times(Y\cap Z)=(X\times Y)\cap(X\times Z)\\ (Y\cap Z)\times X=(Y\times X)\cap(Z\times X)\end{array}\right\}$ — дистрибутивность по пересечению;
- $\left.\begin{array}{l} X\times(Y\backslash Z)=(X\times Y)\backslash(X\times Z)\\ (Y\backslash Z)\times X=(Y\times X)\backslash(Z\times X)\end{array}\right\}$ — дистрибутивность по разности;
- $(X\times Y)\cap(W\times Z)=(X\cap W)\times(Y\cap Z)$.

Некоторые из перечисленных свойств следуют из определения декартова произведения. Для доказательства других свойств необходимо использовать методы доказательств тождеств с множествами.

3.3. Операция проектирования множеств

Введем еще одну операцию — *операцию проектирования*.

Операция проектирования унарна. Она применима не к двум множествам, а к одному множеству. Кроме того, операция проектирования применима только к множеству кортежей одинаковой длины. Проекция множества определяется через проекцию кортежей.

Определим понятие *проекции кортежей*.

Пусть задан кортеж $\alpha=\langle a_1,a_2,\ldots,a_s\rangle$ длины s, $s>0$.

1. Пусть $1\leqslant i\leqslant s$. Тогда проекцией кортежа α на i-ю ось называется i-я компонента кортежа α.

2. Пусть задано произвольное число q такое, что $2\leqslant q\leqslant s$. И пусть задано число осей $1\leqslant i_1\leqslant i_2\leqslant\ldots\leqslant i_q\leqslant s$. Тогда проекцией кортежа α на оси с номерами $i_1,i_2,\ldots,i_q$ называется кортеж $\langle a_{i1},a_{i2}\ldots,a_{iq}\rangle$, который обозначается следующим образом: $\text{пр}_{i1,i2,\ldots,iq}\,\alpha=\langle a_{i1},a_{i2},\ldots,a_{iq}\rangle$.

3. Проекцией кортежа α на пустое множество осей считается пустой кортеж. Аналогично проекцией пустого кортежа на пустое множество осей называется пустой кортеж.

Например, задан кортеж $\alpha=\langle 12,15,6,7,8\rangle$, $\text{пр}_{i1}\,\alpha=\langle 12\rangle$, $\text{пр}_{i2}\,\alpha=\langle 15\rangle$, $\text{пр}_{i3}\,\alpha=\langle 6\rangle$, $\text{пр}_{i4}\,\alpha=\langle 7\rangle$, $\text{пр}_{i5}\,\alpha=\langle 8\rangle$, $\text{пр}_{i1,i2}\,\alpha=\langle 12,15\rangle$, $\text{пр}_{i1,i5}\,\alpha=\langle 12,8\rangle$, $\text{пр}_{i6,i8}\,\alpha=\langle\ \rangle$.

Определим понятие проекции множества. Как отмечено, это понятие будет определено только для случая, когда проектируемое множество состоит из кортежей, причем все кортежи имеют одинаковую длину.

Проекция множества M — это множество проекций кортежей из M.

Пусть задано множество кортежей M длины s, $s > 0$.

1. Пусть $1 \leqslant i \leqslant s$, тогда проекцией множества M на i-ю ось называется множество проекций кортежей из M на i-ю ось и обозначается: $\text{пр}_i M$.

2. Пусть задано произвольное число q, такое, что $2 \leqslant q \leqslant s$, и задано число осей $1 \leqslant i_1 \leqslant i_2 \leqslant \ldots \leqslant i_q \leqslant s$. Тогда проекцией множества M на оси с номерами $i_1, i_2, \ldots, i_q$ называется множество проекций кортежей из M на оси с номерами $i_1, i_2, \ldots, i_q$.

3. Проекцией множества M на пустое множество осей называется множество проекций кортежей из M на пустое множество: $\text{пр}_{\varnothing} M$.

Рассмотрим пример. Пусть $M = \{\langle 1,2,3,4,5\rangle, \langle 2,1,3,5,5\rangle, \langle 3,3,3,3,3\rangle, \langle 3,2,3,4,3\rangle, \langle a,b,a,1,a\rangle\}$. Тогда $\text{пр}_2 M = \langle 2,1,3,2,b\rangle$, $\text{пр}_{2,4} M = \langle\langle 2,4\rangle, \langle 1,5\rangle, \langle 3,3\rangle, \langle 2,4\rangle, \langle b,1\rangle\rangle$, $\text{пр}_{6,7} M = \varnothing$.

Пусть M — произвольное множество, длина которого s, $s \geqslant 2$. Тогда множество M^s состоит из кортежей длины s и значит, его можно проектировать. Операция проектирования множества основана на описанных правилах построения проекций кортежей и множеств. Для любого натурального числа i, $1 \leqslant i \leqslant s$, проекция $\text{пр}_i M^s = M$.

Согласно определению операции проектирования, можно сказать, что для произвольного кортежа $\langle x, y\rangle$ истинны следующие высказывания:

$$\langle x, y\rangle \in A \to x \in \text{пр}_1 A \,\&\, y \in \text{пр}_2 A,$$

$$x \notin \text{пр}_1 A \vee y \notin \text{пр}_2 A \to \langle x, y\rangle \notin A.$$

Приведем основные свойства операции проектирования.

Пусть $A \subseteq X \times Y$, $B \subseteq X \times Y$, тогда для любых $x \in X$ и $y \in \in Y (\forall x \in X \,\&\, y \in Y)$ истинно:

- $\left.\begin{array}{l}\text{пр}_1(X \times Y) = X\\ \text{пр}_2(X \times Y) = Y\end{array}\right\}$,
- $\text{пр}_1 X^2 = \text{пр}_2 X^2 = X$,
- $\left.\begin{array}{l}\text{пр}_1(A \cup B) = \text{пр}_1 A \cup \text{пр}_1 B\\ \text{пр}_2(A \cup B) = \text{пр}_2 A \cup \text{пр}_2 B\end{array}\right\}$,
- $\left.\begin{array}{l}\text{пр}_1(A \cap B) \subseteq \text{пр}_1 A \cap \text{пр}_1 B\\ \text{пр}_2(A \cap B) \subseteq \text{пр}_2 A \cap \text{пр}_2 B\end{array}\right\}$,
- $\left.\begin{array}{l}\text{пр}_1 A = \text{пр}_2 A^{-1}\\ \text{пр}_2 A = \text{пр}_1 A^{-1}\end{array}\right\}$,

- $\left.\begin{array}{l} x \in \text{пр}_1 A \to (\exists y \in Y)(\langle x, y\rangle \in A) \\ y \in \text{пр}_2 A \to (\exists x \in X)(\langle x, y\rangle \in A) \end{array}\right\}$,
- $x \in \text{пр}_1 A \,\&\, y \in \text{пр}_2 A \to (\exists w \in Y)(\exists z \in X)(\langle x, w\rangle \in A \,\&\, \langle z, y\rangle \in A)$.

В то же время в общем случае ложными являются следующие высказывания:

- $\left.\begin{array}{l} x \in \text{пр}_1 A \to \langle x, y\rangle \in A \\ y \in \text{пр}_2 A \to \langle x, y\rangle \in A \end{array}\right\}$,
- $x \in \text{пр}_1 A \,\&\, y \in \text{пр}_2 A \to \langle x, y\rangle \in A$,
- $\left.\begin{array}{l} (\text{пр}_1 A = \text{пр}_1 B) \to A = B \\ (\text{пр}_2 A = \text{пр}_2 B) \to A = B \end{array}\right\}$,
- $(\text{пр}_1 A = \text{пр}_1 B) \,\&\, (\text{пр}_2 A = \text{пр}_2 B) \to A = B$.

Некоторые из перечисленных высказываний следуют из определения прямого произведения. Для доказательства других свойств необходимо использовать методы доказательств тождеств с множествами.

Рассмотрим операции над кортежами. Известны две основные операции: инверсия кортежа и композиция кортежа.

Инверсия.

Инверсия кортежа определяется следующим образом. Пара $\langle c, d\rangle$ называется *инверсией* пары $\langle a, b\rangle$, если $c = b \,\&\, d = a$. Инверсия пары α обозначается α^{-1}.

Например, $\alpha = \langle a, b\rangle$, тогда $\alpha^{-1} = \langle b, a\rangle$, $(\alpha^{-1})^{-1} = \alpha$, $((\alpha^{-1})^{-1})^{-1} = \alpha^{-1}$. Тогда $\alpha^{-n} = \alpha$ и $\alpha^{-(n-1)} = \alpha^{-1}$, при n четном.

Композиция.

Кортеж $\alpha = \langle x, y\rangle$ называется *композицией* двух кортежей $\beta = \langle x, z\rangle$ и $\gamma = \langle z, y\rangle$ и записывается $\alpha = \beta \cdot \gamma$. Операция композиции справедлива, когда вторая компонента кортежа β совпадает с первой компонентой кортежа γ. Здесь как бы происходит «склеивание» двух кортежей по компоненте z.

3.4. График

График — это множество, каждый элемент которого является парой или кортежем длины 2. Множество $\mathbb{P}$ называется графиком, если каждый элемент его пара.

Например, множество $\mathbb{P} = \{\langle a, b\rangle, \langle a, 1\rangle, \langle c, d\rangle\}$ является графиком.

Если M — произвольное множество, то M^2, а также любое множество $C \subseteq M^2$, — график. В частности, графиком является множество $\mathbb{D}^2$ действительных чисел. Пусть заданы множества A и B, тогда $A \times B$, $C \subseteq A \times B$ представляют графики.

Понятие графика является обобщенным. В принципе оно происходит от понятия графика функции.

Областью определения графика $\mathbb{P}$ называется множество $\text{пр}_1\mathbb{P}$ — проекция на первую ось (ось абсцисс) данного графика.

Областью значения графика называется множество проекций на вторую ось (ось ординат) $\text{пр}_2\mathbb{P}$.

Легко видеть, что если $\mathbb{P}$ — график, тогда если $\mathbb{P} = \varnothing$, то $\text{пр}_1\mathbb{P} = \varnothing \wedge \text{пр}_2\mathbb{P} = \varnothing$.

Рассмотрим операции над графиками. Известны две основные операции: инверсия графика и композиция графика.

Инверсия.

Инверсия графика определяется через инверсию кортежа.

Инверсией графика $\mathbb{P}$ называют множество инверсий пар из $\mathbb{P}$.

Например, $\mathbb{P} = \{\langle c, d\rangle, \langle a, b\rangle\}$, $\mathbb{P}^{-1} = \{\langle d, c\rangle, \langle b, a\rangle\}$.

График $\mathbb{Q}$ называется инверсией графика $\mathbb{P}$, если $\exists\alpha \in \mathbb{Q}$ тогда и только тогда, когда $\alpha^{-1} \in \mathbb{P}$, где α — произвольный кортеж.

В теоретико-множественном виде запишем:

$$\alpha^{-1} \in \mathbb{P} \to \alpha \in \mathbb{P}^{-1},$$
$$\alpha \in \mathbb{P} \to \alpha^{-1} \in \mathbb{P}^{-1}.$$

График $\mathbb{P}$ называется *симметричным*, если он наряду с любой своей парой содержит ее инверсию.

Например, график $\mathbb{P} = \{\langle a, b\rangle, \langle b, a\rangle\}$ является симметричным.

Пусть M — произвольное множество. Тогда считают ΔM — множество всех пар вида $\langle x, x\rangle$, где x присутствует во всем множестве M.

Таким образом, если $M = \{a, b\}$, то $\Delta M = \{\langle a, a\rangle, \langle b, b\rangle\}$ — симметричный график и называется *диагональю*.

Композиция.

График $\mathbb{R}$ является *композицией* двух графиков $\mathbb{P}$ и $\mathbb{Q}$, а также $\langle x, y\rangle \in \mathbb{R}$ тогда и только тогда, когда $\exists z$ такое, что $\langle x, z\rangle \in \mathbb{P} \,\&\, \langle z, y\rangle \in \mathbb{Q}$.

Переход от графиков $\mathbb{P}$ и $\mathbb{Q}$ к их композиции $(\mathbb{P} \cdot \mathbb{Q})$ называется *операцией композиции* (или просто композицией) графиков $\mathbb{P}$ и $\mathbb{Q}$.

Например, пусть $\mathbb{P} = \{\langle \underline{a, a}\rangle, \langle a, c\rangle\}$, а $\mathbb{Q} = \{\langle \underline{a, b}\rangle, \langle b, c\rangle\}$, тогда $\mathbb{P} \cdot \mathbb{Q} = \{\langle a, b\rangle\}$.

Композиция графика $\mathbb{P}$ и $\varnothing$ равна $\varnothing$, т. е. $\mathbb{P}\varnothing = \varnothing\mathbb{P} = \varnothing$.

Если M — произвольное множество и $\mathbb{P} \subseteq M^2$, то

$$\mathbb{P} \cdot \Delta M = \Delta M \cdot \mathbb{P} = \mathbb{P}.$$

Если операцию композиции графиков сопоставить с умножением чисел, то роль нуля будет играть пустое множество, а роль единицы — диагональ (Δ).

Пусть $\langle x, z\rangle$ — произвольная пара из $A \cdot B$, причем $A = X \times Y, B = W \times Z$. Тогда для нее справедливо высказывание

$$\langle x, z\rangle \in A \cdot B \to (\exists y \in (Y \cap W))(\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in B).$$

Если некоторая пара $\langle x, z\rangle$ не принадлежит $A \cdot B$, то истинно высказывание:

$$\langle x, z\rangle \notin A \cdot B \to (\forall y \in (Y \cap W))(\langle x, y\rangle \notin A \vee \langle y, z\rangle \notin B).$$

В операции композиции элемент y называется компонирующим элементом для пар $\langle x, y\rangle \in A$ и $\langle y, z\rangle \in B$. Если множество компонирующих элементов пусто, то и результат композиции является пустым множеством:

$$A \cdot B = \varnothing \to \text{пр}_2 A \cap \text{пр}_1 B = \varnothing \to A \cdot \varnothing = \varnothing \cdot A = \varnothing.$$

Приведем свойства операции композиции:

- $A \cdot B \neq B \cdot A$ — некоммутативность;
- $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ — ассоциативность;
- $\left.\begin{array}{l} A \cdot (B \cup C) = (A \cdot B) \cup (A \cdot C) \\ (B \cup C) \cdot A = (B \cdot A) \cup (C \cdot A) \end{array}\right\}$ — дистрибутивность по объединению;
- $\left.\begin{array}{l} A \cdot (B \cap C) \subseteq (A \cdot B) \cap (A \cdot C) \\ (B \cap C) \cdot A \subseteq (B \cdot A) \cap (C \cdot A) \end{array}\right\}$ — дистрибутивность по пересечению;
- $(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$.

Некоторые тождества следуют из определения операции композиции, остальные тождества доказываются уже известными методами.

Используя методы доказательства тождеств с множествами, можно доказать, что для любых трех графиков $\mathbb{P}$, $\mathbb{Q}$, $\mathbb{R}$ справедливо

$$(\mathbb{P} \cdot \mathbb{Q}) \cdot \mathbb{R} = \mathbb{P} \cdot (\mathbb{Q} \cdot \mathbb{R}).$$

Докажем это тождество.

Необходимость. Предположим, что существует произвольный кортеж $\langle a, b\rangle$, принадлежащий левой части тождества. Попытаемся доказать, что он принадлежит правой части тождества:

$$\begin{aligned}\langle a, b\rangle \in (\mathbb{P}\cdot\mathbb{Q})\cdot\mathbb{R} \to \langle a, z\rangle \in (\mathbb{P}\cdot\mathbb{Q})\,\&\,\langle z, b\rangle \in \mathbb{R} \to \\ \to \langle a, x\rangle \in \mathbb{P}\,\&\,\langle x, z\rangle \in \mathbb{Q}\,\&\,\langle z, b\rangle \in \mathbb{R} \to \\ \to \langle a, x\rangle \in \mathbb{P}\,\&\,\langle x, b\rangle \in (\mathbb{Q}\cdot\mathbb{R}) \to \langle a, b\rangle \in (\mathbb{P}\cdot(\mathbb{Q}\cdot\mathbb{R})).\end{aligned}$$

Следовательно, первая часть доказана.

Достаточность. Предположим, что существует произвольный кортеж $\langle a, b\rangle$, принадлежащий правой части тождества. Попытаемся доказать, что он принадлежит левой части тождества:

$$\begin{aligned}\langle a, b\rangle \in (\mathbb{P}\cdot(\mathbb{Q}\cdot\mathbb{R})) \to \langle a, x\rangle \in \mathbb{P}\,\&\,\langle x, b\rangle \in (\mathbb{Q}\cdot\mathbb{R}) \to \\ \to \langle a, x\rangle \in \mathbb{P}\,\&\,(\langle x, d\rangle \in \mathbb{Q}\,\&\,\langle d, b\rangle \in \mathbb{R}) \to \\ \to \langle a, x\rangle \in \mathbb{P}\,\&\,\langle x, d\rangle \in \mathbb{Q}\,\&\,\langle d, b\rangle \in \mathbb{R} \to \\ \to \langle a, d\rangle \in (\mathbb{P}\cdot\mathbb{Q})\,\&\,\langle d, b\rangle \in \mathbb{R} \to \langle a, b\rangle \in ((\mathbb{P}\cdot\mathbb{Q})\cdot\mathbb{R}).\end{aligned}$$

Следовательно, вторая часть доказана. Тогда делаем вывод, что исходное тождество справедливо.

Доказанное тождество позволяет определить композицию трех графиков $\mathbb{P}$, $\mathbb{Q}$, $\mathbb{R}$, понимая под этим результат любой перестановки скобок, приводящей к последовательному попарному компонированию графиков. При компонировании трех графиков возможны только два способа расстановки скобок. Это позволяет определить композицию произвольного попарного компонирования графиков.

Операции композиции и инверсии связаны следующим равенством:

$$(\mathbb{P}\cdot\mathbb{Q})^{-1} = \mathbb{Q}^{-1}\cdot\mathbb{P}^{-1}.$$

Докажем справедливость тождества $(\mathbb{P}\cdot\mathbb{Q})^{-1} \equiv \mathbb{P}^{-1}\cdot\mathbb{Q}^{-1}$.

Необходимость. Пусть $\langle a, b\rangle \in (\mathbb{P}\cdot\mathbb{Q})^{-1} \to \langle b, a\rangle \in (\mathbb{P}\cdot\mathbb{Q}) \to \to \langle b, x\rangle \in \mathbb{P}\,\&\,\langle x, a\rangle \in \mathbb{Q} \to \langle x, b\rangle \in \mathbb{P}^{-1}\,\&\,\langle a, x\rangle \in \mathbb{Q}^{-1}$. Получить теперь $\mathbb{P}^{-1}\cdot\mathbb{Q}^{-1}$ в общем виде невозможно, следовательно, исходное тождество неверно.

Докажем теперь справедливость другого тождества $(\mathbb{P}\times\times\mathbb{Q})^{-1} \equiv \mathbb{Q}^{-1}\cdot\mathbb{P}^{-1}$.

Необходимость. Пусть $\langle a, b\rangle \in (\mathbb{P}\cdot\mathbb{Q})^{-1} \to \langle b, a\rangle \in (\mathbb{P}\cdot\mathbb{Q}) \to \to \langle b, x\rangle \in \mathbb{P}\,\&\,\langle x, a\rangle \in \mathbb{Q} \to \langle x, b\rangle \in \mathbb{P}^{-1}\,\&\,\langle a, x\rangle \in \mathbb{Q}^{-1} \to \langle a, x\rangle \in \in \mathbb{Q}^{-1}\,\&\,\langle x, b\rangle \in \mathbb{P}^{-1} \to \langle a, b\rangle \in (\mathbb{Q}^{-1}\cdot\mathbb{P}^{-1})$.

Достаточность. Пусть $\langle a,b\rangle \in (\mathbb{Q}^{-1}\cdot\mathbb{P}^{-1}) \to \langle a,x\rangle \in \mathbb{Q}^{-1} \wedge \langle x,b\rangle \in \mathbb{P}^{-1} \to \langle x,a\rangle \in \mathbb{Q} \wedge \langle b,x\rangle \in \mathbb{P} \to \langle b,x\rangle \in \mathbb{P} \wedge \langle x,a\rangle \in \mathbb{Q} \to \langle b,a\rangle \in (\mathbb{P}\cdot\mathbb{Q}) \to \langle a,b\rangle \in (\mathbb{P}\cdot\mathbb{Q})^{-1}$.

Следовательно, это тождество справедливо.

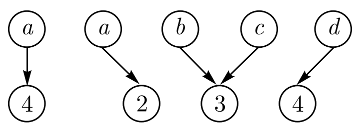
Рассмотрим два основных свойства графиков.

График $\mathbb{P}$ называется *функциональным*, если в нем нет пар с одинаковыми первыми и разными вторыми компонентами.

Например, график $\mathbb{P} = \{\langle b,a\rangle, \langle c,a\rangle, \langle d,a\rangle\}$ является функциональным графиком.

На рис. 3.2, *а*, *б* приведены примеры функциональных графиков.

$$\mathbb{P}_1 = \{\langle a,4\rangle\}, \quad \mathbb{P}_2 = \{\langle a,2\rangle, \langle b,3\rangle, \langle c,3\rangle, \langle d,4\rangle\}.$$



а *б*

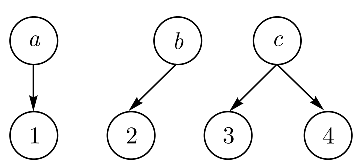
Рис. 3.2. Примеры функциональных графиков

График $\mathbb{P}$ называется *инъективным*, если в нем нет пар с различными первыми и одинаковыми вторыми компонентами.

Например, $\mathbb{P} = \{\langle a,b\rangle, \langle a,c\rangle, \langle a,d\rangle\}$ является инъективным графиком.

На рис. 3.3, *а*, *б* приведены примеры инъективных графиков.

$$\mathbb{P}_3 = \{\langle a,1\rangle\}, \quad \mathbb{P}_4 = \{\langle b,2\rangle, \langle c,3\rangle, \langle c,4\rangle\}.$$



а *б*

Рис. 3.3. Примеры инъективных графиков

Композиция функциональных графиков есть функциональный график, т. е. композиция сохраняет функциональность.

Композиция инъективных графиков инъективна. Доказательство данного утверждения можно найти в литературе, приведенной в конце книги.

Итак, говорят, что график $\mathbb{P}$ функционален тогда и только тогда, когда $\mathbb{P}^{-1}$ инъективен. График $\mathbb{P}$ инъективен тогда и только тогда, когда $\mathbb{P}^{-1}$ функционален.

Примеры решения задач

Пример 3.1. Пусть заданы множества $A = \{a, b\}$; $B = \{3, 1, 2\}$; $C = \{\alpha, \beta\}$.

Ответ. $A \times B \times C = \{\langle a, 1, \alpha\rangle, \langle a, 1, \beta\rangle, \langle a, 2, \alpha\rangle, \langle a, 2, \beta\rangle, \langle a, 3, \alpha\rangle, \langle a, 3, \beta\rangle, \langle b, 1, \alpha\rangle, \langle b, 1, \beta\rangle, \langle b, 2, \alpha\rangle, \langle b, 2, \beta\rangle, \langle b, 3, \alpha\rangle, \langle b, 3, \beta\rangle\}$.

Пример 3.2. Пусть задано произвольное множество $A = \{a, b\}$. Найти третью степень заданного множества.

Ответ. $A^3 = A \times A \times A = \{a, b\} \times \{a, b\} \times \{a, b\} = \{\langle a,a,a\rangle; \langle a, a, b\rangle; \langle a, b, a\rangle; \langle a, b, b\rangle; \langle b, a, a\rangle; \langle b, a, b\rangle; \langle b, b, a\rangle; \langle b,b,b\rangle\}$.

Пример 3.3. Доказать истинность тождества методом взаимного включения:

$$X \times (Y \cup Z) = (X \times Y) \cup (X \times Z).$$

Доказательство. *Необходимость.* Пусть для произвольного кортежа $\langle x, y\rangle$ истинно высказывание:

$$\langle x, y\rangle \in X \times (Y \cup Z) \to x \in X \,\&\, y \in (Y \cup Z) \to$$

(согласно свойству дистрибутивности)

$$x \in X \,\&\, (y \in Y \vee y \in Z) \to$$

(по определению операции объединения множеств)

$$x \in X \,\&\, y \in Y \vee x \in X \,\&\, y \in Z \to$$

(согласно дистрибутивному закону)

$$\langle x, y\rangle \in (X \times Y) \vee \langle x, y\rangle \in (X \times Z) \to \langle x, y\rangle \in ((X \times Y) \cup (X \times Z)).$$

(исходя из свойства дистрибутивности).

Таким образом, прямое включение доказано.

Достаточность. Аналогично доказывается обратное включение:

$$\begin{aligned} &\langle x, y\rangle \in ((X \times Y) \cup (X \times Z)) \to \\ &\to \langle x, y\rangle \in (X \times Y) \vee \langle x, y\rangle \in (X \times Z) \to \\ &\to (x \in X \,\&\, y \in Y) \vee (x \in X \,\&\, y \in Z) \to \\ &\to x \in X \,\&\, (y \in Y \vee y \in Z) \to \\ &\to x \in X \,\&\, y \in (Y \cup Z) \to \langle x, y\rangle \in (X \times (Y \cup Z)). \end{aligned}$$

Обратное включение доказано. Таким образом, поскольку оба включения истинны, истинно и исходное тождество.

Пример 3.4. Доказать справедливость тождества методом от противного:

$$((X \times Y) \cap (W \times Z)) \backslash ((X \cap W) \times (Y \cap Z)) = \varnothing.$$

Доказательство. Предположим, что данное выражение неверно и, следовательно, данное множество не равно пустому. Тогда должен существовать хотя бы один кортеж $\langle x, y \rangle$, принадлежащий исходному множеству:

$$\langle x, y \rangle \in [(X \times Y) \cap (W \times Z)] \backslash [(X \cap W) \times (Y \cap Z)] \to$$

$$\langle x, y \rangle \in ((X \times Y) \cap (W \times Z)) \,\&\, \langle x, y \rangle \notin ((X \cap W) \times (Y \cap Z)) \to$$

(согласно определению операции разности множеств)

$$[\langle x, y \rangle \in (X \times Y) \,\&\, \langle x, y \rangle \in (W \times Z)] \,\&\, [x \notin (X \cap W) \vee y \notin (Y \cap Z)] \to$$

(согласно определению операции пересечения множеств)

$$(x \in X \,\&\, y \in Y \,\&\, x \in W \,\&\, y \in Z) \,\&\, (x \notin X \vee x \notin W \vee y \notin Y \vee y \notin Z) \to$$

(согласно дистрибутивному закону алгебры логики)

$(x \in X \,\&\, y \in Y \,\&\, x \in W \,\&\, y \in Z \,\&\, x \notin X) \vee (x \in X \,\&\, y \in Y \,\&\, x \in W \,\&\, y \in Z \,\&\, x \notin W) \vee (x \in X \,\&\, y \in Y \,\&\, x \in W \,\&\, x \in Z \,\&\, y \notin Y) \vee (x \in X \,\&\, y \in Y \,\&\, x \in W \,\&\, y \in Z \,\&\, y \notin Z)$.

В полученном выражении в каждой из составляющих его конъюнкций мы обнаружили противоречие. Например, $x \in X \,\&\, x \notin X$. То есть очевидно, что наше предположение ложно, а значит исходное тождество истинно, что и требовалось доказать.

Пример 3.5. Пусть задан график $A = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 2, 5 \rangle, \langle 3, 4 \rangle, \langle 3, 6 \rangle\}$. Найти его инверсию.

Ответ. $A^{-1} = \{\langle 2, 1 \rangle, \langle 3, 2 \rangle, \langle 5, 2 \rangle, \langle 4, 3 \rangle, \langle 6, 3 \rangle\}$.

Пример 3.6. Пусть задан график $A = \{\langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 3 \rangle, \langle 3, 3 \rangle, \langle 3, 4 \rangle\}$. Найти первую и вторую проекции заданного графика.

Ответ. $\text{пр}_1 A = \{1, 2, 3\}$, $\text{пр}_2 A = \{2, 3, 4\}$.

Пример 3.7. Доказать справедливость тождества методом взаимного включения.

Ответ. $\text{пр}_1 (A \cap B) \subseteq \text{пр}_1 A \cap \text{пр}_1 B$.

Доказательство. *Необходимость.* Предположим, что исходное высказывание справедливо, тогда для любого элемента x, принадлежащего данной проекции, справедливо следующее высказывание:

$$x \in \text{пр}_1(A \cap B) \to (\exists y \in Y)(\langle x, y\rangle \in (A \cap B)) \to \\ \to \langle x, y\rangle \in A \,\&\, \langle x, y\rangle \in B \to x \in \text{пр}_1 A \,\&\, x \in \text{пр}_1 B \to \\ \to x \in (\text{пр}_1 A \cap \text{пр}_1 B)$$

(согласно определению операции проектирования).

Таким образом, прямое включение доказано.

Достаточность. Теперь докажем, что в общем случае обратное включение невозможно. Пусть для любого x истинно:

$$x \in (\text{пр}_1 A \cap \text{пр}_1 B) \to x \in \text{пр}_1 A \,\&\, x \in \text{пр}_1 B \to \\ \to (\exists y \in Y)(\langle x, y\rangle \in A) \,\&\, (\exists z \in Y)(\langle x, z\rangle \in B).$$

Поскольку в общем случае неизвестно, совпадают ли y и z, обратное включение в общем случае доказать невозможно.

Пример 3.8. Привести пример, демонстрирующий справедливость включения $\text{пр}_1(A \cap B) \subseteq \text{пр}_1 A \cap \text{пр}_1 B$.

Решение. Пусть заданы множества $A = \{\langle 1, 2\rangle\}$; $B = \{\langle 1, 3\rangle\}$. Очевидно, что $\text{пр}_1(A \cap B) = \text{пр}\varnothing = \varnothing$, в то время как $\text{пр}_1 A \cap \cap\, \text{пр}_1 B = \{1\} \cap \{1\} = \{1\}$. $\varnothing \subseteq \{1\}$. Таким образом, $\text{пр}_1(A \cap B) \neq \neq \text{пр}_1 A \cap \text{пр}_1 B$. В случае, если $A \cap B \neq \varnothing$, исходное высказывание преобразуется в равенство.

Пример 3.9. Привести пример того, что высказывание $\text{пр}_1 A = \text{пр}_1 B \to A = B$ ложно.

Решение. Пусть заданы множества: $A = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 1, 4\rangle, \langle 5, 6\rangle, \langle 5, 7\rangle\}$; $B = \{\langle 1, 6\rangle, \langle 1, 7\rangle, \langle 5, 2\rangle, \langle 5, 3\rangle, \langle 5, 4\rangle\}$.

Для доказательства построим проекции заданных множеств. Очевидно, что проекции множеств равны между собой: $\text{пр}_1 A = = \text{пр}_1 B = \{1, 5\}$; $\text{пр}_2 A = \text{пр}_2 B = \{2, 3, 4, 6, 7\}$, в то время как множества A и B не равны. Следовательно, задание выполнено.

Пример 3.10. Заданы произвольные графики: $A = = \{\langle a, a\rangle, \langle b, e\rangle, \langle c, k\rangle, \langle f, g\rangle\}$; $B = \{\langle a, d\rangle, \langle c, b\rangle, \langle g, f\rangle, \langle e, c\rangle, \langle b, b\rangle\}$. Найти композиции графиков $A \cdot B$ и $B \cdot A$.

Решение. Для того чтобы найти композицию графиков A и B, выбираем поочередно из кортежей множества вторые компоненты A, сравниваем со всеми первыми компонентами кортежей множества B. В тех случаях, когда обнаруживается совпадение, выполняем композицию соответствующих кортежей. Например, берем вторую компоненту первого кортежа из множества A. Это компонента a. Сравниваем ее поочередно со всеми первыми компонентами кортежей множества B. Находим совпадение с первой компонентой первого кортежа множества B. В результате получаем композицию этих двух кортежей:

$$\langle a, a\rangle \cdot \langle a, d\rangle = \langle a, d\rangle.$$

Затем выбираем вторую компоненту следующего кортежа множества A и сравниваем ее со всеми первыми компонентами кортежей множества B:

$$\langle b, e\rangle \cdot \langle e, c\rangle = \langle b, c\rangle.$$

Процесс продолжается аналогично до тех пор, пока не будут выявлены все совпадения и построены все композиции кортежей. В результате мы получим композицию множеств A и B. Для построения композиции множеств B и A необходимо аналогичным образом сравнить вторые компоненты кортежей множества B с первыми компонентами кортежей множества A.

Ответ. $A \cdot B = \{\langle a, d\rangle, \langle b, c\rangle, \langle f, f\rangle\}$, $B \cdot A = \{\langle c, e\rangle, \langle g, g\rangle, \langle e, k\rangle, \langle b, e\rangle\}$.

Пример 3.11. Доказать справедливость высказывания методом взаимного включения:

$$A \cdot (B \cup C) = (A \cdot B) \cup (A \cdot C).$$

Доказательство. *Необходимость.* Предположим, что данное тождество справедливо. Тогда существует произвольная пара:

$$\langle x, z\rangle \in (A \cdot (B \cup C)) \to (\exists y)(\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in (B \cup C)) \to$$

(согласно определению операции композиции)

$$\langle x, y\rangle \in A \,\&\, (\langle y, z\rangle \in B \vee \langle y, z\rangle \in C) \to$$

(по определению операции объединения множеств)

$$(\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in B) \vee (\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in C) \to$$

(согласно дистрибутивному закону)

$$(\langle x, z\rangle \in A \cdot B) \vee (\langle x, z\rangle \in A \cdot C) \to \langle x, z\rangle \in ((A \cdot B) \cup (A \cdot C)).$$

То есть прямое включение доказано.

Достаточность. Теперь докажем обратное включение. Предположим, что существует пара:

$$\langle x, z\rangle \in ((A \cdot B) \vee (A \cdot C)) \to \langle x, z\rangle \in (A \cdot B) \vee \langle x, z\rangle \in (A \cdot C) \to$$
$$(\exists y)(\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in B) \vee (\exists w)(\langle x, w\rangle \in A \,\&\, \langle w, z\rangle \in C) \to$$

$$(\text{так как в общем случае компонирующие элементы не обязательно совпадают})$$

$$(\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in (B \cup C)) \vee (\langle x, w\rangle \in A \,\&\, \langle w, z\rangle \in (B \cup C)) \to$$
$$\langle x, z\rangle \in (A \cdot (B\cup C)) \vee \langle x, z\rangle \in (A \cdot (B\cup C)) \to \langle x, z\rangle \in (A \cdot (B\cup C)).$$

Обратное включение также доказано, следовательно, исходное тождество верно.

Пример 3.12. Доказать справедливость высказывания методом взаимного включения:

$$A \cdot (B \cap C) \subseteq (A \cdot B) \cap (A \cdot C).$$

Доказательство прямого включения проводится аналогично доказательству, рассмотренному выше, и интереса не представляет. Рассмотрим обратное включение и докажем, что оно в общем случае не имеет места.

Достаточность. Предположим, что справедливо тождество:

$$(A \cdot B) \cap (A \cdot C) \subseteq A \cdot (B \cap C).$$

Тогда существует произвольная пара:

$$\langle x, z\rangle \in ((A \cdot B) \cap (A \cdot C)) \to \langle x, z\rangle \in (A \cdot B) \,\&\, \langle x, z\rangle \in (A \cdot C) \to$$
$$(\exists y)(\langle x, y\rangle \in A \,\&\, \langle y, z\rangle \in B) \,\&\, (\exists w)(\langle x, w\rangle \in A \,\&\, \langle w, z\rangle \in C)$$

(в общем случае $y \neq w$, значит надо продолжить цепочку преобразований, чтобы доказать, что обратное включение невозможно).

Таким образом, обратное включение в общем случае не доказывается.

Следовательно, высказывание $A \cdot (B \cap C) \subseteq (A \cdot B) \cap (A \cdot C)$ истинно.

Контрольные вопросы

1. Приведите примеры кортежей.
2. Как образуется прямое произведение множеств?
3. В каком случае число элементов прямого произведения множеств равняется нулю?
4. Что представляет собой прямое произведение двух множеств с точки зрения теории множеств?
5. Какими свойствами обладает декартово произведение множеств?

6. В чем заключается операция проектирования множеств?

7. В каком случае упорядоченная пара не принадлежит прямому произведению двух множеств?

8. Для каких множеств A и B справедливо $X \times Y = Y \times X$?

9. Равны ли множества $\text{пр}_1 A \cup \text{пр}_2 A$ и A, если $A \subseteq X \times Y$?

10. Что такое инверсия упорядоченного множества A?

11. Для какого множества $A \subseteq X \times Y$ справедливо $A = A^{-1}$?

12. В каком случае существует композиция двух произвольных упорядоченных множеств A и B?

13. В каком случае справедливо тождество $A \cdot B = B \cdot A$?

14. В каких случаях справедливо тождество $A \cdot A = A$?

15. Что такое график?

16. Приведите основные операции над графиками.

Задания для самостоятельной работы

1. Найти прямое произведение множеств X и Y, если:

а) $X = \{\{a, b\}, c, \{d, e, f\}\}$; $Y = \{g, h\}$;

б) $X = \{a, b, c\}$; $Y = \varnothing$;

в) $X = \{2, 4, \sqrt{3}\}$; $Y = \{\{\varnothing\}, a, b\}$.

2. Найти n-ю степень множества X, если:

а) $X = \{x\}, n = 5$;   в) $X = \{\{\varnothing\}, y\}$, $n = 2$;

б) $X = \{a, b\}, n = 3$;   г) $X = 0$, $n = 3$.

3. Доказать, что для произвольных множеств X, Y, W, Z справедливы следующие высказывания:

а) $(Y \cup Z) \times X = (Y \times X) \cup (Z \times X)$;

б) $X \times (Y \cap Z) = (X \times Y) \cap (X \times Z)$;

в) $X \times (Y \backslash Z) = (X \times Y) \backslash (X \times Z)$;

г) $(X \times Y) \cup (W \times Z) \subseteq (X \cup W) \times (Y \cup Z)$;

д) $(X \cup Y) \times (W \cup Z) = (X \times W) \cup (Y \times W) \cup (X \times Z) \cup (Y \times Z)$.

4. Равны ли множества $\text{пр}_1 A \cup \text{пр}_2 A = A$, если $A \subseteq X \times Y$?

5. Для каких множеств A и B справедливо $A \times B = B \times A$?

6. Для какого множества справедливо $A = A^{-1}$, если $A \subseteq \subseteq X \times Y$?

7. Доказать или опровергнуть, что для множеств A и B, где $A \subseteq X \times Y$, $B \subseteq X \times Y$, справедливы следующие высказывания:

а) $\text{пр}_1(A\backslash B) = \text{пр}_1 A\backslash \text{пр}_1 B$;

б) $\text{пр}_1(A \cup B)^{-1} = \text{пр}_2 A \cup \text{пр}_2 B$;

в) $\text{пр}_1(A \cup B) = \text{пр}_2 A^{-1} \cup \text{пр}_2 B^{-1}$;

г) $(A\backslash B)^{-1} = A^{-1}\backslash B^{-1}$;

д) $\text{пр}_1(A\backslash B)^{-1} = \text{пр}_2 A\backslash \text{пр}_2 B$;

е) $(A \cup B)^{-1} = A^{-1} \cup B^{-1}$;

ж) $(A \cap B)^{-1} = A^{-1} \cap B^{-1}$.

8. Доказать или опровергнуть, что для множеств A, B и C, где $A \subseteq X \times Y$, $B \subseteq X \times Y$, $C \subseteq X \times Y$, справедливы следующие тождества:

а) $(B \cup C) \cdot A = (B \cdot A) \cup (C \cdot A)$;

б) $A \cdot (B\backslash C) = (A \cdot B)\backslash(A \cdot C)$.

Декартово произведение множеств позволяет перейти к графическому представлению упорядоченных множеств.

Глава 4
ОТНОШЕНИЯ

Измени отношения к вещам, которые тебя беспокоят, и ты будешь от них в безопасности.

Марк Аврелий

Отношения, определение и способы задания отношений, основные операции над отношениями, свойства специальных отношений, разбиение множеств, отношение эквивалентности, отношение порядка, отношение квазипорядка, частичного порядка и доминирования

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать способы задания отношений;
- знать основные свойства отношений;
- уметь решать задачи на доказательство тождеств с отношениями;
- знать понятия отношений рефлексивности, симметричности, транзитивности и эквивалентности;
- уметь выполнять разбиение множеств;
- иметь понятие об отношении порядка.

4.1. Основные понятия отношений

Отношение определяет, в какой связи любые объекты в природе находятся между собой. На формальном языке *отношение* — это пара множеств, причем упорядоченная, первая компонента которой является подмножеством квадрата второй компоненты.

В отличие от понятия множества, понятие отношения — определенное понятие. Запись $\varphi = \langle \Phi, M \rangle$ называется отношением, если $\Phi \subseteq M^2$, т. е. $\Phi \subseteq M \times M$, где φ — отношение; Φ — график отношения; M — область задания отношения, представляющая из себя неупорядоченное множество. Отношение такого типа называется *бинарным*. Если $\Phi \subseteq M^k$, то отношение называется *k-арным*.

Отношение φ с областью задания M называется отношением на множестве M.

Пусть имеем элемент графика, представляющий собой пару $\langle a, b\rangle \in \Phi$. Тогда говорят, что если пара $\langle a, b\rangle \in \Phi$, то существует отношение $a\,\varphi\, b$, в котором элементы a и b находятся в некоторой связи. Другими словами, элемент a находится в отношении φ к элементу b. Если a и b — элементы множества M ($a, b \in M$), то $a\,\varphi\, b$ будет истинным или ложным высказыванием.

Рассмотрим некоторые примеры отношений.

1. Пусть дана область задания отношения $M = \{1, 2, 3, 4\}$ и график отношения $\Phi = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 2\rangle, \langle 3, 2\rangle\}$. Тогда можно получить $M^2 = M \times M = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle, \langle 1, 4\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \ldots, \langle 4, 4\rangle\}$. На рис. 4.1 точками изображено отношение $\varphi = \langle \Phi, M\rangle$.

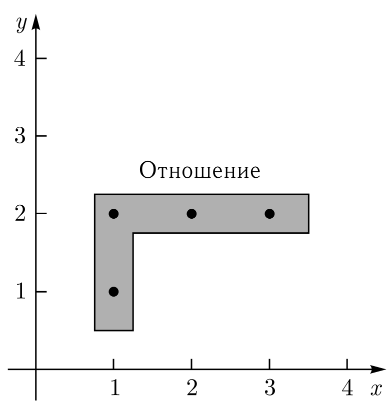


Рис. 4.1. Пример задания отношения на множестве M

Отметим, что отношение часто задают рисунками, изображая на координатной плоскости график определяемого отношения. Причем в отношение входят соответствующие точки графика.

Отношение также может быть задано в виде графа (рис. 4.2).

Существует еще один способ задания отношений через высказывательные формы.

Например, запись $x\,\varphi\, y \to x < y$ означает, что φ есть «отношение меньше» между элементами x и y.

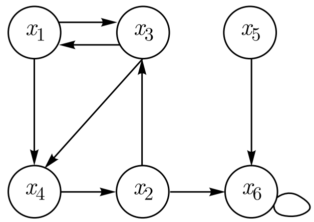


Рис. 4.2. Пример задания отношения в виде графа

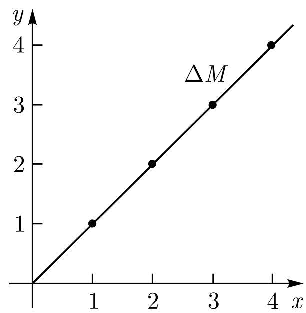


Рис. 4.3. Пример отношения

Тогда высказывание $5 \varphi 3$ является ложным, а высказывание $3 \varphi 4$ — истинным отношением.

Важным в отношениях выступает понятие диагонали $\varphi = \langle \Delta M, M \rangle$ на множестве M. Это такое отношение, когда все точки графика находятся на диагонали. Например, пусть в отношении $\varphi = \langle \Delta M, M \rangle$, $M = \{1, 2, 3, 4\}$ и график отношения $\Delta M = \{\langle 1, 1\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle, \langle 4, 4\rangle\}$. На рис. 4.3 показано графическое представление такого отношения.

4.2. Основные свойства отношений

Отношение $\varphi = \langle \Phi, M \rangle$ называется *полным*, если $\Phi = M^2$, т. е. $(\forall x, y \in M)(x \varphi y)$. Другими словами, для любых элементов x, y, принадлежащих множеству M, истинно высказывание, что элементы x, y находятся в отношении φ.

Отношение $\varphi = \langle \Phi, M \rangle$ называется *пустым*, если график Φ является пустым множеством, т. е. $\varphi = \langle \varnothing, M \rangle$. Другими словами, имеется область задания отношения, на которой не задан график отношения.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *равенства*, если $\Phi = \Delta M$. В теоретико-множественном плане можно записать $(\forall x, y \in M)(x \,\varphi\, y \to x = y)$. Например, задано $\varphi = \langle \Phi, M \rangle$, $M = \{1, 2, 3\}$, $\Phi = \{\langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle\}$. Данное отношение является отношением равенства.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *неравенства*, если $\Phi = M^2 \backslash \Delta M$, т.е. $(\forall x, y \in M)(x \,\varphi\, y \to x \neq y)$. Например, задано $\varphi = \langle \Phi, M \rangle$, $M = \{1, 2, 3\}$, $\Phi = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 1 \rangle\}$. Данное отношение является отношением неравенства. Отношения «$5 > 3$» и «$3 < 10$» также являются примерами отношения неравенства.

4.3. Операции над отношениями

На отношения переносятся основные операции над множествами, но они могут выполняться только на одной и той же области задания.

Объединением отношений φ_1 и φ_2 на множестве M называется отношение φ_3:

$$\varphi_1 \cup \varphi_2 = \varphi_3, \varphi_1 = \langle \Phi_1, M \rangle, \quad \varphi_2 = \langle \Phi_2, M \rangle,$$
$$\varphi_3 = \langle \Phi_1 \cup \Phi_2, M \rangle,$$
$$\langle a, b \rangle \in \Phi_1 \cup \Phi_2 \to \langle a, b \rangle \in \Phi_1 \vee \langle a, b \rangle \in \Phi_2 \,\&\, a, b \in M.$$

Например, пусть имеем два отношения: $\varphi_1 = \langle \Phi_1, M \rangle$, $\varphi_2 = \langle \Phi_2, M \rangle$,

$$M = \{2, 3, 4\}, \quad \Phi_1 = \{\langle 2, 1 \rangle, \langle 2, 2 \rangle, \langle 2, 4 \rangle\},$$
$$\Phi_2 = \{\langle 2, 1 \rangle, \langle 2, 3 \rangle, \langle 4, 4 \rangle\}.$$

Тогда объединение этих отношений $\varphi_3 = \langle \Phi_3, M \rangle$, $\Phi_3 = \Phi_1 \cup \Phi_2 = \{\langle 2, 1 \rangle, \langle 2, 2 \rangle, \langle 2, 4 \rangle, \langle 2, 3 \rangle, \langle 4, 4 \rangle\}$.

Отметим, что для операции объединения над отношениями справедлива следующая запись:

$$x(\varphi_1 \cup \varphi_2)y \to x \,\varphi_1\, y \vee x \,\varphi_2\, y.$$

Пересечением отношений φ_1 и φ_2 на множестве M называется отношение φ_3, для которого:

$$\varphi_1 \cap \varphi_2 = \varphi_3, \varphi_1 = \langle \Phi_1, M \rangle, \quad \varphi_2 = \langle \Phi_2, M \rangle,$$
$$\varphi_3 = \langle \Phi_1 \cap \Phi_2, M \rangle,$$
$$\langle a, b \rangle \in \Phi_1 \cap \Phi_2 \to \langle a, b \rangle \in \Phi_1 \,\&\, \langle a, b \rangle \in \Phi_2 \,\&\, a, b \in M.$$

Например, пусть имеем два отношения: $\varphi_1 = \langle \Phi_1, M \rangle$, $\varphi_2 = \langle \Phi_2, M \rangle$, $M = \{1, 2\}$, $\varphi_1 = \langle \{\langle 1, 1 \rangle, \langle 1, 2 \rangle\}, \{1, 2\} \rangle$, $\varphi_2 = \langle \{\langle 1, 2 \rangle, \langle 2, 2 \rangle\}, \{1, 2\} \rangle$.

Тогда пересечение этих отношений $\varphi_3 = \langle \Phi_3, M\rangle = \langle \{\langle 1,2\rangle\}, \{1,2\}\rangle$.

Отметим, что для операции пересечения над отношениями справедлива следующая запись:

$$x(\varphi_1 \cap \varphi_2)y \to x\,\varphi_1\,y\,\&\,x\,\varphi_2\,y.$$

Операции объединения и пересечения так же, как и для множеств, применимы для любого числа отношений.

Отношение φ_3 называется *разностью* отношений φ_1 и φ_2, если

$$\varphi_1 = \langle \Phi_1, M\rangle, \quad \varphi_2 = \langle \Phi_2, M\rangle, \quad \varphi_3 = \varphi_1 \backslash \varphi_2 = \langle \Phi_1 \backslash \Phi_2, M\rangle,$$
$$\langle a,b\rangle \in \Phi_1 \backslash \Phi_2 \to \langle a,b\rangle \in \Phi_1 \,\&\, \langle a,b\rangle \notin \Phi_2 \,\&\, a,b \in M.$$

Например, пусть имеем два отношения: $\varphi_1 = \langle \Phi_1, M\rangle$, $\varphi_2 = \langle \Phi_2, M\rangle$, $M = \{1,2,3\}$, $\Phi_1 = \{\langle 1,2\rangle, \langle 2,2\rangle, \langle 3,3\rangle\}$, $\Phi_2 = \{\langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle\}$. Тогда $\Phi_3 = \Phi_1 \backslash \Phi_2 = \{\langle 1,2\rangle\}$. Разность этих отношений $\varphi_3 = \langle \Phi_3, M\rangle = \langle \{\langle 1,2\rangle\}, \{1,2,3\}\rangle$.

Отметим, что для операции разности над отношениями справедлива следующая запись:

$$x(\varphi_1 \backslash \varphi_2)y \to x\,\varphi_1\,y\,\&\,x\,\overline{\varphi}_2\,y.$$

Над отношениями выполняются также операции инверсии и композиции.

Если $\varphi = \langle \Phi, M\rangle$, то инверсия $\varphi^{-1} = \langle \Phi^{-1}, M\rangle$.

Для того чтобы найти *инверсию* отношения, необходимо проинвертировать элементы его графика на множестве M. Отметим, что для операции инверсии над отношениями справедлива следующая запись:

$$x\,\varphi^{-1}\,y \to y\,\varphi\,x.$$

Например, для отношения $\varphi = \langle \Phi, M\rangle$, $M = \{1,2,3\}$, $\Phi = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle\}$, инверсия $\varphi^{-1} = \langle \Phi^{-1}, M\rangle$ и $\Phi^{-1} = \{\langle 1,1\rangle, \langle 2,1\rangle, \langle 3,1\rangle\}$.

Композицией двух отношений является новое отношение, у которого компонируют графики отношений:

$$\varphi_1 = \langle \Phi_1, M\rangle, \quad \varphi_2 = \langle \Phi_2, M\rangle.$$
$$\varphi_1 \cdot \varphi_2 = \langle \Phi_1 \cdot \Phi_2, M\rangle.$$

Например, $\varphi_1 = \langle \Phi_1, M\rangle$, $\varphi_2 = \langle \Phi_2, M\rangle$, $M = \{1,2,3\}$, $\Phi_1 = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 3,3\rangle\}$, $\Phi_2 = \{\langle 1,1\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle\}$. Тогда композиция графиков этих отношений равна $\Phi_1 \cdot \Phi_2 = \{\langle 1,1\rangle, \langle 1,3\rangle, \langle 1,2\rangle, \langle 3,2\rangle, \langle 3,1\rangle\}$.

Отметим, что все операции над отношениями могут выполняться только на одной и той же области задания, и в результате выполнения операций снова получается отношение с той же самой областью задания.

Введем операцию, меняющую область задания отношений.

Пусть $\varphi = \langle \Phi, M \rangle$ и $\exists A \subseteq M$, тогда *сужением отношения* φ на множестве A называется новое отношение

$$\varphi_1 = \langle \Phi \cap A^2, A \rangle.$$

Например, $\varphi = \langle \Phi, M \rangle$, $M = \{1, 2, 3\}$, $\Phi = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle\}$, $A = \{1, 2\}$. Тогда $\varphi_1 = \langle \{\langle 1, 1\rangle, \langle 1, 2\rangle\}, \{1, 2\}\rangle$.

4.4. Основные свойства специальных отношений

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *рефлексивности*, если

$$(\forall x \in M)(x \,\varphi\, x).$$

Другими словами, если для любого элемента $x \in M$ истинно высказывание $x \,\varphi\, x$, то φ — отношение рефлексивности. Это отношение является унарной операцией, т. е. операцией над одним элементом. Пример отношения рефлексивности — отношение равенства и отношение параллельности ($x = x$, $x \parallel x$). Очевидно, что отношение перпендикулярности между прямыми x и y не является отношением рефлексивности. Отношение φ рефлексивно тогда и только тогда, когда $\Delta M \subseteq \Phi$, т. е. когда все точки диагонали принадлежат графику отношений.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *антирефлексивности*, если

$$(\forall x \in M) \neg (x \,\varphi\, x),$$

т. е. $\Delta M \cap \Phi = \varnothing$.

Например, выражения $x > x$, $x \neq x$, $x \perp x$ — отношения *антирефлексивности*.

Отношение $\varphi = \langle \Phi, M \rangle$ называют отношением *симметричности*, если

$$(\forall x, y \in M)(x \,\varphi\, y \to y \,\varphi\, x).$$

Отношение симметричности — бинарная операция. Например, выражения $x = y$, $x \parallel y$ являются отношением симметричности.

Для графика отношения симметричности справедливо $\Phi = \Phi^{-1}$. Например, $a = b \to b = a$, $a \parallel b \to b \parallel a$.

Отношение $\varphi = \langle \Phi, M \rangle$, $\Phi \subseteq M \times M$ называется отношением *антисимметричности*, если

$$x \, \varphi \, y \wedge x \neq y \to \overline{(y \, \varphi \, x)}.$$

Очевидно, что отношение $\varphi = \langle \Phi, M \rangle$ антисимметрично тогда и только тогда, когда

$$\Phi \cap \Phi^{-1} \subseteq \Delta M.$$

Например, высказывания «больше», «меньше», «не равно» являются примерами отношений антисимметричности ($x > y$, $x \neq y$).

Отношение $\varphi = \langle \Phi, M \rangle$, $\Phi \subseteq M \times M$ называется *антисимметричным*, если оно одновременно антирефлексивно и антисимметрично.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *транзитивности*, если

$$(\forall x, y, z \in M)(x \, \varphi \, y \,\&\, y \, \varphi \, z \to x \, \varphi \, z).$$

Например, если $x = y \,\&\, y = z \to x = z$, то φ — отношение транзитивности. Для отношения параллельности справедливо: $x \parallel y \,\&\, y \parallel z \to x \parallel z$, следовательно, оно является отношением транзитивности.

Для отношения транзитивности (иногда говорят транзитивное отношение) справедливо:

$$\Phi \cdot \Phi \subseteq \Phi.$$

Примеры

Пример 4.1. Полное отношение $\varphi = \langle M^2, M \rangle$ является транзитивным. Пусть $M = \langle 1, 2 \rangle$, тогда $M^2 = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 1,1\rangle, \langle 2,2\rangle\}$ и $M^2 \cdot M^2 = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 2,2\rangle, \langle 2,1\rangle\}$, $M^2 \cdot M^2 \subseteq M^2$.

Пример 4.2. Пустое отношение является носителем всех свойств.

Интерес в задачах построения информационно-коммуникационных технологий представляют отношения, которые обладают комбинациями свойств.

Отношение φ называется отношением *эквивалентности*, если оно рефлексивно ($\Delta M \subseteq \Phi$), симметрично ($\Phi = \Phi^{-1}$) и транзитивно ($\Phi \cdot \Phi = \Phi$). Иногда, имея в виду отношения рефлексивности, симметричности, транзитивности и т. п., говорят, что эти отношения соответственно рефлексивности, симметричности, транзитивности и т. п.

$\varphi \sim \varphi$; «$\sim$» — знак эквивалентности.

Отношение равенства является отношением эквивалентности, так как оно рефлексивно ($a = a$), симметрично ($a = b \to b = a$) и транзитивно ($a = b \,\&\, b = c \to a = c$) на множестве $M = \{a, b, c\}$. Отношение параллельности является отношением эквивалентности, так как оно рефлексивно ($a \parallel a$), симметрично ($a \parallel b \to b \parallel a$) и транзитивно ($a \parallel b \,\&\, b \parallel a \to a \parallel c$) на множестве $M = \{a, b, c\}$. Отношения «больше», «меньше», «не равно», перпендикулярности и т. п. не являются отношениями эквивалентности, так как не обладают в совокупности свойствами рефлексивности, симметричности и транзитивности.

Отношение эквивалентности связано с понятием разбиения множеств.

4.5. Разбиение множеств

Понятие разбиения ассоциируется с классификацией живых и неживых организмов, а также широко используется при решении задач искусственного интеллекта и проектировании информационных и телекоммуникационных систем.

Приведем формальное определение разбиения.

Система W множеств называется *разбиением* одного множества M, если она удовлетворяет четырем условиям:

1) $(\forall X \in W)(X \subseteq M)$.
2) $(\forall X \in W)(X \neq \varnothing)$.
3) $(\forall X, Y \in W)(X \neq Y \to X \cap Y = \varnothing)$.
4) $\bigcup\limits_{i=1}^{n} X_i = M$.

Первое условие говорит о том, что подмножество обязательно должно включаться и в систему, и в отдельное множество. Второе условие говорит, что подмножества разбиений не должны быть пустыми. Третье показывает, что подмножества должны быть разными и обязательно находиться в различных разбиениях. И, наконец, четвертое условие говорит о том, что все подмножества разбиения в совокупности должны представлять множество M.

Подмножества разбиения называются блоками, причем блоки разбиения не могут иметь общих элементов.

Разбиение системы W множества M называется *поэлементным*, если каждый блок разбиения является одноэлементным множеством.

Например, множество $M = \{1, 2, 3, 4\}$ может быть разбито на четыре одноэлементных подмножества $W = \{M_1, M_2, M_3, M_4\}$, $M_1 = \{1\}$, $M_2 = \{2\}$, $M_3 = \{3\}$, $M_4 = \{4\}$.

Например, множество $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ можно разбить на 4 блока: $M = \{X_1, X_2, X_3, X_4\}$, где $X_1 = \{1, 3, 5, 7, 9\}$, $X_2 = \{2\}$, $X_3 = \{4\}$, $X_4 = \{6, 8, 10\}$.

Разбиение системы называется *целым*, если $W = \{M\}$. Для предыдущего примера получим $W = \{1, 2, 3, 4\}$.

Поэлементное и целое разбиения называются тривиальными. Остальные разбиения, если они существуют, называются нетривиальными.

Примеры

Пример 4.3. Пустое множество имеет единственное разбиение, т. е. пустую систему множеств ($W = \varnothing$).

Пример 4.4. Одноэлементное множество $M = \{a\}$ имеет единственное разбиение $W = M = \{a\}$, которое одновременно является и целым, и поэлементным.

Пример 4.5. Множество $M = \{a, b\}$ имеет два разбиения: $W_1 = \{M_1(a), M_2(b)\}$, $W_2 = \{M\}$.

Пример 4.6. Множество $M = \{a, b, c\}$ имеет пять разбиений следующего вида: $W_1 = \{a, b, c\}$, $W_2 = \{\{a\}, \{b\}, \{c\}\}$, $W_3 = \{\{a\}, \{b, c\}\}$, $W_4 = \{\{b\}, \{a, c\}\}$, $W_5 = \{\{c\}, \{a, b\}\}$. Очевидно, что из пяти разбиений два являются тривиальными (W_1, W_2), а три — нетривиальными (W_3–W_5).

Пример 4.7. Система курсов факультета является разбиением множества студентов, если ни один студент не учится одновременно на двух курсах.

Диаграмма Эйлера–Венна, иллюстрирующая возможное разбиение множества $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ на четыре блока, приведена на рис. 4.4. Если принять, что мощность каждого из подмножеств разбиения пропорциональна величине занимаемого сектора окружности, то можно записать следующее разбиение: $X_1 = \{1, 2, 3, 4, 5\}$; $X_2 = \{6\}$; $X_3 = \{7\}$; $X_4 = \{8, 9, 10\}$.

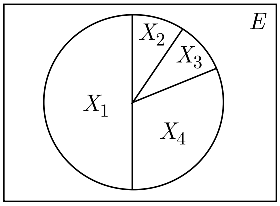


Рис. 4.4. Пример разбиения множества A на четыре блока

Известно, что отношение φ на множестве M и разбиение этого множества *сопряжены*, если высказывание ($\forall x, y \in M\; x \varphi y$)

истинно тогда, когда x и y принадлежат одному и тому же разбиению (к одному и тому же блоку разбиения).

Приведем ряд теорем, показывающих взаимосвязь между отношением эквивалентности и разбиением.

Теорема 4.1 (о единственности разбиения, сопряженного с данным отношением).

Если два разбиения множества M сопряжены с одним и тем же отношением на этом же множестве, то они совпадают.

Другими словами, разбиение, сопряженное с данным отношением, единственно.

Теорема 4.2 (об отношении, сопряженном с некоторым разбиением).

Если отношение φ на множестве M сопряжено с каким-нибудь разбиением этого множества, то φ — отношение эквивалентности.

Теорема 4.3 (о существовании разбиения, сопряженного с данным отношением эквивалентности).

Для любого отношения эквивалентности φ на множестве M существует сопряженное с ним разбиение множества M.

Итак, разбиение, сопряженное с данным отношением φ, существует тогда (теорема 3) и только тогда (теорема 2), когда φ — отношение эквивалентности, причем в этом случае сопряженное разбиение единственно (теорема 1).

Теорема 4.4. *Пусть φ — отношение эквивалентности на непустом множестве M, тогда различные классы эквивалентности определяют разбиение множества M.*

4.6. Отношение порядка

Отношение порядка связано с отношениями типа $\leqslant$, $<$, $>$, $\geqslant$.

Отношение φ называется отношением *нестрогого порядка*, если оно транзитивно ($\Phi \cdot \Phi = \Phi$), рефлексивно ($\Delta M \subseteq \Phi$) и антисимметрично ($\Phi \cap \Phi^{-1} \subseteq \Delta M$).

Отношение φ называется отношением *строгого порядка*, если оно транзитивно ($\Phi \cdot \Phi = \Phi$) и антирефлексивно ($\Delta M \cap \Phi = \varnothing$).

Введем понятие отношения *связности*. Отношение $\varphi = \langle \Phi, M \rangle$ называется *связным*, если выполняется условие $M^2 \backslash \Delta M \subseteq \Phi \cup \Phi^{-1}$.

Отношение φ называется отношением *совершенно нестрогого порядка*, если оно является отношением связности ($M^2 \backslash \Delta M \subseteq \Phi \cup \Phi^{-1}$) и отношением нестрогого порядка.

Отношение φ называется отношением *совершенно строгого порядка*, если оно является отношением связности ($M^2 \backslash \Delta M \subseteq \Phi \cup \Phi^{-1}$) и отношением строгого порядка.

Приведем таблицу Шихановича, показывающую связь между отношением порядка и специальными отношениями.

Порядок	Транзитивность $\Phi \cdot \Phi \subseteq \Phi$	Рефлексивность $\Delta M \subseteq \Phi$	Антирефлексивность $\Delta M \cap \Phi = \varnothing$	Антисимметричность $\Phi \cap \Phi^{-1} \subseteq \Delta M$	Связность $M^2 \backslash \Delta M \subseteq \Phi \cap \Phi^{-1}$
нестрогий	+	+		+	
совершенно нестрогий	+	+		+	+
строгий	+		+		
совершенно строгий	+		+		+

Примеры

Пример 4.8. $x \varphi y \to x > y$. Данное высказывание говорит о том, что отношение φ является отношением совершенно строгого порядка.

Пример 4.9. $x \varphi y \to x \geqslant y$. Данное высказывание говорит о том, что отношение φ является отношением совершенно нестрогого порядка.

Пример 4.10. $\forall x, y\ X \varphi Y \to X \subseteq Y$. В данном случае имеет место отношение нестрогого порядка.

Пример 4.11. $\forall x, y\ X \varphi Y \to X \subset Y$. В данном случае имеет место отношение строгого порядка.

Пример 4.12. Отношение равенства на множестве M является нестрогим порядком, причем это самый маленький нестрогий порядок.

Пример 4.13. Отношение строгого порядка можно задать числовой осью $x \varphi y$. Инверсию строгого порядка можно изобразить перевернутой числовой осью $x \varphi^{-1} y$.

Как видно из приведенных примеров, первое и второе отношения справедливы для высказываний, а третье и четвертое — для множеств.

Совершенно нестрогий порядок есть упорядоченная пара $\langle X, \leqslant \rangle$, где отношение $\leqslant$ — частично упорядоченное множество X.

Например, если F есть некоторая система множеств, то $\langle F, \leqslant\rangle$ — частично упорядоченное множество. Для любого предложения об упорядоченных множествах можно указать эквивалентное ему предложение об отношениях порядка и обратно.

В тех случаях, когда множество X означает множество объектов или группы объектов, говорят об отношении *доминирования*.

Будем говорить, что x доминирует y и писать $x \gg y$, если x в чем-то превосходит y. Так, x может быть спортсменом или командой, победившей спортсмена или команду y, или лицом, пользующимся авторитетом у лица y, или свойством, которое мы предпочитаем свойству y. Будем говорить, что между элементами множества X имеет место отношение доминирования, если эти элементы обладают следующими двумя свойствами.

1) Никакой элемент не может доминировать сам себя, т. е. $x \gg x$ — ложно, имеет место отношение антирефлексивности $(\Delta M \cap \Phi = \varnothing)$.

2) В каждой паре элементов в точности один элемент доминирует второго, т. е. $x \gg y$ и $y \gg x$ — взаимоисключается, значит справедливо отношение антисимметричности $(\Phi \cap \Phi^{-1} \subseteq \Delta M)$.

В отношении доминирования свойство транзитивности не имеет места. Например, если в соревнованиях команда x победила команду y, а команда y победила команду z, то отсюда еще не следует, что команда x обязательно победит команду z. Для кортежей пример отношения доминирования имеет вид $\alpha = \langle 3,4\rangle \gg \beta = \langle 1,2\rangle$, т. е. для первых элементов кортежей α и β справедливо $3 \gg 1$, а для вторых элементов кортежей α и β справедливо $4 \gg 2$.

Существует еще одно отношение порядка, которое называется квазипорядком (от «квази» — почти).

$\varphi = \langle\Phi, M\rangle$ называется *отношение квазипорядка*, если оно транзитивно и рефлексивно.

Для отношения квазипорядка справедливо

$$\Phi \cdot \Phi \subseteq \Phi \wedge \Delta M \subseteq \Phi.$$

Отношение $\varphi = \langle\Phi, M\rangle$ называется *отношением толерантности*, если оно рефлексивно $(\Delta M \subseteq \Phi)$ и симметрично $(\Phi = \Phi^{-1})$. Следовательно, для отношения толерантности справедливо

$$\Delta M \subseteq \Phi \wedge \Phi = \Phi^{-1}.$$

Другими словами, для отношения толерантности

$$(\forall x \in M)(x \,\varphi\, x),$$
$$(\forall x, y \in M)(x \,\varphi\, y \to y \,\varphi\, x).$$

Примеры

Пример 4.14. Отношение эквивалентности всегда является отношением квазипорядка, так как оно транзитивно, рефлексивно и симметрично.

Пример 4.15. Отношение $A \,\varphi\, B$, где φ определяет, что A не старше курсом, чем B, является примером отношения квазипорядка, так как оно транзитивно и рефлексивно.

Пример 4.16. Отношение параллельности является отношением толерантности, так как оно рефлексивно $(a \parallel a)$ и симметрично $(a \parallel b \to b \parallel a)$ на множестве $M = \{a, b\}$.

Контрольные вопросы

1. Дайте определение отношения.
2. Как строятся матрица и график отношения $\varphi = \langle \Phi, X \rangle$?
3. Перечислите основные операции над отношениями.
4. Что называется инверсией и композицией отношений?
5. Как определить понятие образа и прообраза множества при композиции двух отношений?
6. В каком случае композиция двух отношений φ и ψ будет являться:

а) функциональным отношением;

б) инъективным отношением;

в) биективным отношением?

7. Дайте определение и приведите пример рефлексивного отношения.
8. Дайте определение и приведите пример симметричного отношения.
9. Дайте определение и приведите пример транзитивного отношения.
10. Дайте определение и приведите пример связного отношения.
11. Может ли антисимметричное отношение быть также рефлексивным?
12. Может ли асимметричное отношение быть также рефлексивным?
13. Может ли рефлексивное отношение быть несвязным?

14. Приведите определение и примеры отношения толерантности.

15. Покажите, каким образом отношение толерантности связано с отношениями эквивалентности, доминирования и порядка.

Задания для самостоятельной работы

1. Пусть задано произвольное множество $X = \{x, y, z, t\}$. Задайте произвольное отношение на этом множестве теоретическим, матричным и графическим способами.

2. Пусть заданы произвольные отношения $\varphi = \langle F, X\rangle$ и $\psi = \langle P, X\rangle$, где $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$; $F = \{\langle x_1, x_2\rangle; \langle x_1, x_3\rangle; \langle x_1, x_5\rangle; \langle x_2, x_3\rangle; \langle x_2, x_4\rangle; \langle x_2, x_6\rangle; \langle x_3, x_2\rangle; \langle x_3, x_4\rangle; \langle x_4, x_1\rangle; \langle x_4, x_4\rangle; \langle x_5, x_4\rangle; \langle x_5, x_6\rangle; \langle x_6, x_1\rangle; \langle x_6, x_2\rangle\}$; $P = \{\langle x_1, x_3\rangle; \langle x_1, x_5\rangle; \langle x_3, x_6\rangle; \langle x_3, x_7\rangle; \langle x_5, x_1\rangle; \langle x_5, x_6\rangle; \langle x_6, x_3\rangle; \langle x_7, x_1\rangle; \langle x_7, x_7\rangle\}$.

Найти и записать теоретическим, матричным и графическим способами:

а) $\varphi \cup \psi$;
б) $\varphi \cap \psi$;
в) $\varphi \backslash \psi$;
г) $\varphi \cdot \psi$;
д) дополнение $\overline{\underline{\varphi}}$ отношения φ до отношения ψ;
е) дополнение $\overline{\underline{\psi}}$ отношения ψ до отношения φ.

3. Пусть заданы отношения φ, ψ, σ на множестве X. Доказать или опровергнуть истинность следующих тождеств:

а) $\varphi \cdot (\psi \cup \sigma) = (\varphi \cdot \psi) \cup (\varphi \cdot \sigma)$;
б) $\varphi \cup (\psi \cap \sigma) = (\varphi \cup \psi) \cap (\varphi \cup \sigma)$;
в) $\varphi \cdot (\psi \cap \sigma) \subseteq (\varphi \cdot \psi) \cap (\varphi \cdot \sigma)$;
г) $\varphi \cap (\psi \cup \sigma) = (\varphi \cap \psi) \cup (\varphi \cap \sigma)$.

4. Построить график отношения, являющегося:

а) рефлексивным и симметричным;
б) нерефлексивным и связным;
в) асимметричным и транзитивным;
г) рефлексивным, симметричным и транзитивным;
д) антирефлексивным, несимметричным и связным;
е) рефлексивным, антисимметричным и транзитивным;
ж) рефлексивным, асимметричным, транзитивным и связным.

Использование отношений позволяет строить модели взаимосвязей между любыми объектами в природе.

Глава 5
СООТВЕТСТВИЯ

Почти каждому мудрому изречению соответствует противоположное по смыслу, при этом не менее мудрое.

Д. Сантаяна

Соответствия, *определение и способы задания соответствий*, *основные операции над соответствиями*, *инверсия и композиция соответствий*, *образ и прообраз соответствия*, *тождества с соответствиями*, *свойства соответствий*, *функция и ее свойства*, *принцип Дирихле*, *взаимно-однозначное соответствие*, *отображение*

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать определение и способы задания соответствий;
- знать основные свойства соответствий;
- уметь выполнять операции над соответствиями;
- уметь решать задачи на доказательство тождеств с соответствиями;
- знать понятия функциональности, инъективности, сюръективности, всюду определенности;
- уметь определять и строить взаимно-однозначное соответствие;
- знать определение, способы задания и свойства функции.

5.1. Определение соответствия

Говорят, что между множествами X, Y установлено *соответствие*, если указано произвольное подмножество $G \subseteq X \times Y$, которое обладает некоторыми свойствами. Соответствия обычно обозначают прописными буквами греческого алфавита, например, Γ, Δ и т. д. Тогда *соответствие* (Γ) — это тройка множеств

$\Gamma = \langle G, X, Y \rangle$, первая компонента которой является графиком G, вторая — множеством X и третья — множеством Y:

$$\Gamma = \langle G, X, Y \rangle, \quad G \subseteq X \times Y - \textit{определение соответствия.}$$

Множество X задает область отправления соответствия, а множество Y — область прибытия соответствия.

Существует три способа задания соответствий: *теоретический*, *матричный*, *графический*. Теоретический способ заключается в задании графика соответствия и множеств X и Y.

Для графика соответствия справедливо

$$G \subseteq X \times Y \to G = X \times Y \vee G \subset X \times Y.$$

Часто область отправления называют областью определения соответствия (это проекция G на первую ось, $\text{пр}_1 G$), а область прибытия — областью значений соответствия (проекция на вторую ось, $\text{пр}_2 G$).

При задании матричным способом соответствие представляется в виде матрицы R_Γ размером $n \times m$, где строки представляют элементы множества X, столбцы — элементы множества Y, а элемент матрицы $r_{i,j}$ принимает значения:

$r_{i,j} = 1$, если существует кортеж $\langle x_i, y_j \rangle \in F$;

$r_{i,j} = 0$ в противном случае.

Таким образом, соответствие можно представить, например, в виде следующей матрицы:

$$\mathrm{R}_\Gamma = \begin{array}{c|cccc|} & a & b & c & d \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{array}.$$

Соответствие, заданное в графическом виде, показано на рис. 5.1. В этом случае соответствие представляет собой график, вершинами которого являются элементы, принадлежащие множествам X и Y соответствия $\Gamma = \langle X, Y, F \rangle$, а кортежи вида

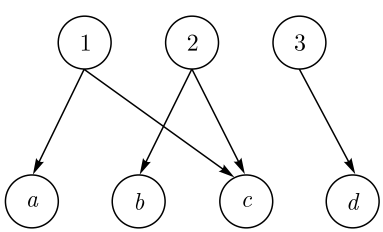


Рис. 5.1. Пример задания соответствия в графическом виде

$\langle x_i, y_j \rangle$, принадлежащие множеству F, присутствуют на графике соответствия в виде стрелок, направленных от x_i к y_j.

Соответствие, график которого $F = X \times Y$, называется соответствием *с полным графиком* и обозначается Γ_Π. Соответствие, график которого $F = \varnothing$, называется соответствием *с пустым графиком* и обозначается $\Gamma_\varnothing$.

Пусть задано соответствие $\Gamma = \langle G, X, Y \rangle$ и $G = \{\langle a, b \rangle\}$, $X = \{a\}$, $Y = \{b\}$. Тогда говорят, что элемент (b) соответствует элементу (a). При этом элемент $a \in \text{пр}_1 G$, и соответствие Γ определено на этом элементе a. Элемент $b \in \text{пр}_2 G$ и является значением соответствия Γ.

Если $X = Y$, то соответствие Γ превращается в отношение. Можно сказать, что отношение — частный случай соответствия. Все свойства и операции над отношениями можно переносить на соответствия.

Примеры

Пример 5.1. $\Gamma = \langle \{\langle a, \alpha \rangle, \langle b, \beta \rangle, \langle c, \alpha \rangle\}, X = \{a, b, c\}, Y = \{\alpha, \beta\} \rangle$. На рис. 5.2 представлено графическое задание этого соответствия.

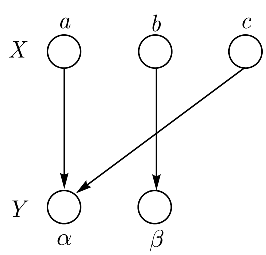


Рис. 5.2. Пример графического задания соответствия Γ

Пример 5.2. Пусть $\Gamma = \langle G, X, Y \rangle$, $X = \{1, 2, 3\}$, $Y = \{4, 5\}$, $G = \{\langle 1, 4 \rangle; \langle 1, 5 \rangle; \langle 2, 4 \rangle; \langle 2, 5 \rangle; \langle 3, 4 \rangle; \langle 3, 5 \rangle\}$. Тогда $\text{пр}_1 G = \{1, 2, 3\}$, $\text{пр}_2 G = \{4, 5\}$. Как видно, это соответствие является полным на множестве $X \times Y$, т. е. $G = X \times Y$.

Пример 5.3. На предприятии имеются три машины A, B, C. Машина C находится в ремонте. В штате три шофера: X, Y, Z, из которых шофер X находится в отпуске. Распределение шоферов по машинам является примером соответствия. Пусть M — область отправления, а N — область прибытия соответствия $\Gamma = \langle G, M, N \rangle$.

Тогда $M = \{X, Y, Z\}$, $N = \{A, B, C\}$, а один из возможных способов распределения шоферов по автомашинам задает график $G = \{\langle Y, B\rangle, \langle Z, A\rangle\}$. В нашем случае $G \subseteq M \times N$. На рис. 5.3 показано графическое представление этого соответствия.

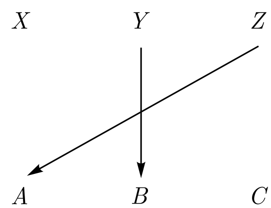


Рис. 5.3. Графическое представление соответствия $\Gamma = \langle G, M, N\rangle$

Пример 5.4. Выражение $\langle\varnothing, \varnothing, \varnothing\rangle$ является соответствием.

Отметим следующее: если $G \subseteq X \times Y$ и $G^{-1} \subseteq Y \times X$, то в случае, когда $\langle G, X, Y\rangle$ — соответствие, $\langle G^{-1}, Y, X\rangle$ — инверсия этого соответствия.

5.2. Операции над соответствиями

Для соответствий действительны все операции, определенные для множеств и отношений, т. е. объединение, пересечение, разность, инверсия, композиция.

Объединением соответствий $\Gamma = \langle X, Y, F\rangle$ и $\Delta = \langle W, Z, P\rangle$ является соответствие $\Gamma = \langle X \cup W, Y \cup Z, F \cup P\rangle$.

Пересечением соответствий Γ и Δ является соответствие $\Gamma = \langle X \cap W, Y \cap Z, F \cap P\rangle$.

Разностью соответствий Γ и Δ является соответствие $\Gamma = \langle X \backslash W, Y \backslash Z, F \backslash P\rangle$.

Инверсией соответствия Γ является соответствие Γ^{-1} такое, что множество Y — область отправления соответствия Γ^{-1}; множество X — область прибытия соответствия Γ^{-1}, а график соответствия F^{-1} — инверсия графика F соответствия Γ.

Композицией соответствий $\Gamma_1 = \langle X, Y, F\rangle$ и $\Gamma_2 = \langle W, Z, P\rangle$ является соответствие такое, что его областью отправления является область отправления соответствия Γ_1, областью прибытия — область прибытия соответствия Γ_2, а графиком — композиция графиков F и P: $\Gamma_1 \cdot \Gamma_2 = \langle X, Z, F \cdot P\rangle$.

В случае, если $Y \cap W = \varnothing$, то результатом композиции соответствий будет соответствие *с пустым графиком*.

Соответствие Ω называется *инверсией соответствия* Γ, если область отправления Γ равна области прибытия Ω и график Γ является инверсией графика Ω.

Четная инверсия оставляет соответствие самим собой, а нечетная — инвертирует, т. е. $(\Gamma^{-1})^{-1} = \Gamma$, а $((\Gamma^{-1})^{-1})^{-1} = \Gamma^{-1}$. Соответствие $\Gamma^{-1} = \Gamma$ тогда и только тогда, когда график соответствия симметричен: $G = G^{-1}$, а область отправления соответствия совпадает с областью прибытия. Например, $\Gamma = \langle G, X, X\rangle$, $X = \{1, 2, 3\}$, $G = \{\langle 1, 1\rangle, \langle 2, 2\rangle\}$. На рис. 5.4 приведено графическое представление этого соответствия.

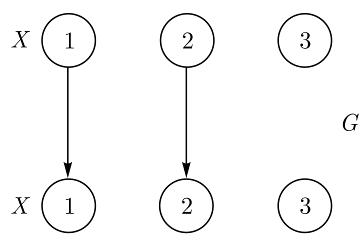


Рис. 5.4. Графическое представление соответствия

Для соответствия так же, как для отношений и множеств, справедлива операция композиции. *Композиция соответствий* определяется через композицию их графиков. Композиция соответствий не пуста, если существует хотя бы один элемент $y \in Y \,\&\, y \in Z$. Пусть заданы соответствия $\Gamma_1 = \langle G, X, Y\rangle$, $\Gamma_2 = \langle H, Z, U\rangle$. Тогда $\Gamma_1 \cdot \Gamma_2 = \langle G \cdot H, X, U\rangle$ определяет композицию двух соответствий.

Например, пусть заданы множества $X = \{a, b\}$, $Y = \{c, d\}$, $Z = \{d, e\}$, $U = \langle k, m\rangle$. Для получения непустого результата композиции соответствий множество Z должно частично или полностью совпадать с множеством Y.

Зададим произвольные графики $G = \{\langle a,c\rangle,\langle b,d\rangle\}$ и $H = \{\langle d,k\rangle, \langle d, m\rangle, \langle e, k\rangle\}$. Тогда композиция графиков $G \cdot H = \{\langle b, k\rangle, \langle b, m\rangle\}$ и композиция соответствий запишется как $\Gamma_1 \cdot \Gamma_2 = \langle\{\langle b, k\rangle, \langle b, m\rangle\}, X, U\rangle$. Графическое представление композиции соответствий Γ_1 и Γ_2 показано на рис. 5.5.

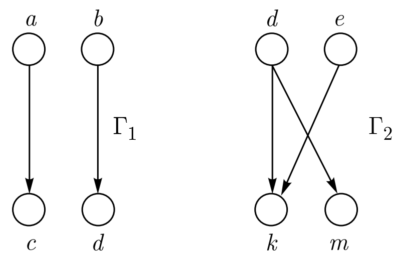


Рис. 5.5. Пример графического представления композиции соответствий

Для любых трех соответствий существует следующее правило композиции:

$$(\Gamma_1 \cdot \Gamma_2) \cdot \Gamma_3 = \Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_3).$$

Докажем это тождество.

1. Докажем *необходимость*. Предположим, что $\langle a, b\rangle \in (\Gamma_1 \cdot \Gamma_2) \cdot \Gamma_3 \to \langle a, x\rangle \in \Gamma_1 \cdot \Gamma_2 \wedge \langle x, b\rangle \in \Gamma_3 \to \langle a, x_1\rangle \in \Gamma_1 \wedge \langle x_1, x\rangle \in \Gamma_2 \wedge \langle x, b\rangle \in \Gamma_3 \to \langle a, x_1\rangle \in \Gamma_1 \wedge \langle x_1, b\rangle \in \Gamma_2 \cdot \Gamma_3 \to \langle a, b\rangle \in \Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_3)$.

Первая часть доказана.

2. Теперь докажем вторую часть, т. е. *достаточность*.

Пусть $\langle a, b\rangle \in \Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_3) \to \langle a, x\rangle \in \Gamma_1 \wedge \langle x, b\rangle \in \Gamma_2 \cdot \Gamma_3 \to \langle a, x\rangle \in \Gamma_1 \wedge \langle x, z\rangle \in \Gamma_2 \wedge \langle z, b\rangle \in \Gamma_3 \to \langle a, x\rangle \in \Gamma_1 \wedge \langle x, z\rangle \in \Gamma_2 \wedge \langle z, b\rangle \in \Gamma_3 \to \langle a, z\rangle \in \Gamma_1 \cdot \Gamma_2 \wedge \langle z, b\rangle \in \Gamma_3 \to \langle a, b\rangle \in (\Gamma_1 \cdot \Gamma_2) \cdot \Gamma_3$, что и требовалось доказать. Следовательно, тождество справедливо.

Примеры

Пример 5.5. Заданы соответствия $\Gamma_1 = \langle F, X, Y\rangle$; $X = \{1, 2, 3\}$; $Y = \{a, b, c, d\}$; $F = \{\langle 1, a\rangle; \langle 1, c\rangle; \langle 2, b\rangle; \langle 2, c\rangle; \langle 3, d\rangle\}$; $\Gamma_2 = \langle P, W, Z\rangle$; $W = \{2, 4, 5, 6\}$; $Z = \{a, c, d, e\}$; $P = \{\langle 2, a\rangle; \langle 2, e\rangle; \langle 4, a\rangle; \langle 4, d\rangle; \langle 5, c\rangle; \langle 5, e\rangle; \langle 6, c\rangle; \langle 6, d\rangle\}$. Найдем их пересечение, объединение и разность.

Ответ. Объединение, пересечение и разность соответствий Γ и Δ изображены на рис. 5.6, *а–в*.

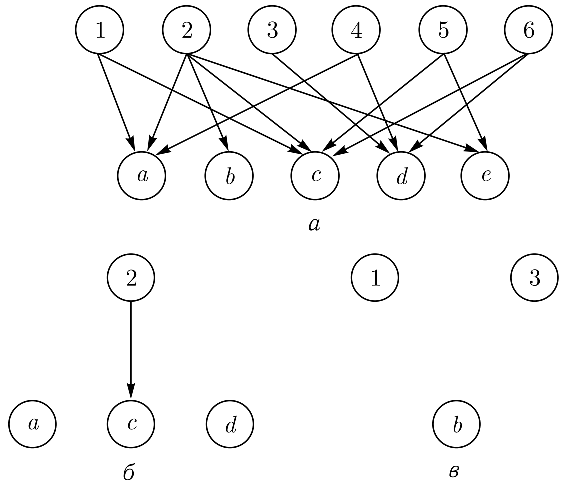


Рис. 5.6. Результаты операций объединения, пересечения и разности соответствий Γ_1 и Γ_2

Пример 5.6. Для соответствий Γ_1 и Γ_2, заданных в предыдущем примере, найти инверсию соответствия Γ_2 и композицию соответствий $\Gamma_1 \cdot \Gamma_2^{-1}$.

Ответ. Инверсия соответствия Γ_2 изображена на рис. 5.7, а композиция соответствий Γ_1 и Γ_2^{-1} — на рис. 5.8.

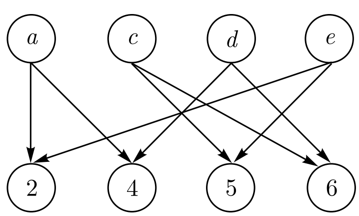


Рис. 5.7. Результат операции инверсии соответствия Γ_2

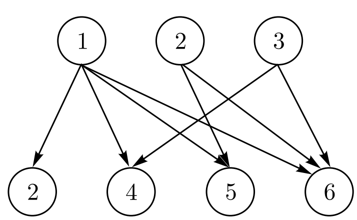


Рис. 5.8. Результат операции композиции соответствий Γ_1 и Γ_2^{-1}

5.3. Понятия образа и прообраза при соответствии

Введем два новых понятия.

Пусть $\Gamma = \langle G, X, Y \rangle$ и задано множество $A \subseteq X$. Тогда *образом* множества A при соответствии Γ называется подмножество тех элементов Y, которые соответствуют элементам из A. Запись $\Gamma(A) = \{y \in Y / \exists x \in A, \langle x, y \rangle \in G\}$ является определением образа множества A при соответствии Γ.

Рассмотрим пример. Задано соответствие $\Gamma = \langle G, X, Y \rangle$, $X = \{1, 2, 3\}$, $Y = \{a, b, c\}$, $G = \{\langle 1, a \rangle, \langle 1, b \rangle, \langle 2, c \rangle, \langle 3, b \rangle, \langle 3, a \rangle\}$ и задано произвольное множество $A \subseteq X$, $A = \{1, 3\}$. Тогда образом данного множества A является $\Gamma(A) = \{a, b\}$ (рис. 5.9).

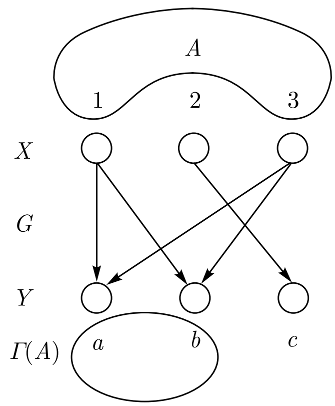


Рис. 5.9. Изображение образа множества A при соответствии Γ

На графическом языке $\Gamma(A)$ — это множество концов стрелок, выходящих из элементов множества A. Образ множества A можно также определить следующим образом:

$$\Gamma(A) = \text{пр}_2 \, [(A \times Y) \cap G].$$

Если Γ_1 и Γ_2 — произвольные соответствия, то справедливо выражение

$$(\Gamma_1 \cdot \Gamma_2)(A) = \Gamma_2(\Gamma_1(A)).$$

Прообразом множества B при соответствии $\Gamma = \langle G, X, Y \rangle$ называется множество тех элементов области отправления, каждому из которых соответствует какой-нибудь элемент множества B. $\Gamma^{-1}(B)$ — обозначение прообраза. Запись $\Gamma^{-1}(B) = \{x \in X / \exists y \in B, \langle x, y \rangle \in G\}$ является определением прообраза множества B при соответствии Γ.

Прообраз множества B можно также определить следующим образом:

$$\Gamma^{-1}(B) = \text{пр}_1\,[G \cap (X \times B)].$$

Следовательно, прообраз на графическом языке представляет собой множество начал стрелок графика G.

Пример 5.7. Пусть задано соответствие $\Gamma = \langle G, X, Y\rangle = \langle\{\langle 1,a\rangle, \langle 1,b\rangle, \langle 1,c\rangle, \langle 2,a\rangle\}, \{1,2\}, \{a,b,c\}\rangle$, $B \subseteq Y$, $B = \{a,c\}$. В этом случае прообраз множества B — $\Gamma^{-1}(B) = \{1,2\}$ (рис. 5.10).

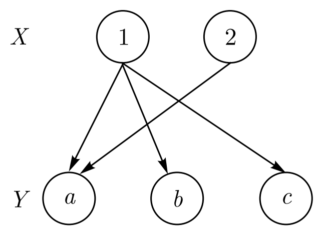


Рис. 5.10. Графическое представление прообраза множества B

Введем понятие сужения и продолжения соответствия. Пусть B — произвольное множество, $B \subseteq X$. Тогда *сужением соответствия* Γ на множество B называется соответствие

$$\Gamma_B = \langle G \cap (B \times Y), X, Y\rangle.$$

Приведем пример сужения соответствия. Дано: $\Gamma = \langle G, X, Y\rangle$, $X = \{1,2\}$, $Y = \{a,b,c\}$, $G = \{\langle 1,a\rangle, \langle 1,b\rangle, \langle 1,c\rangle, \langle 2,b\rangle, \langle 2,c\rangle\}$, $B = \{1\}$. Тогда $B \times Y = \{\langle 1,a\rangle, \langle 1,b\rangle, \langle 1,c\rangle\}$ и сужение соответствия на множество B — $\Gamma_B = \langle G \cap (B \times Y), X, Y\rangle = \langle\{\langle 1,a\rangle, \langle 1,b\rangle, \langle 1,c\rangle\}, \{1,2\}, \{a,b,c\}\rangle$. Построим это сужение (рис. 5.11).

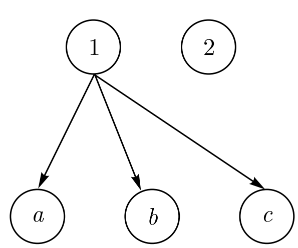


Рис. 5.11. Пример сужения соответствия

Двойственным к понятию сужения является *продолжение соответствия*. Пусть заданы соответствия $\Gamma = \langle G, X, Y \rangle$ и $\Omega = \langle H, Z, U \rangle$, причем $G \subseteq H$, $Z = X$, $U = Y$. Тогда соответствие Ω является продолжением соответствия Γ. Из этого следует, что произвольный элемент b, соответствующий произвольному элементу a в Γ, обязательно соответствует тому же самому элементу в Ω, но не наоборот:

$$b \to a.$$

Пример 5.8. Пусть заданы соответствия $\Gamma = \langle G, X, Y \rangle$, $X = \{1, 2, 3\}$, $Y = \{a, b, c, d\}$, $G = \{\langle 1, a\rangle, \langle 1, b\rangle, \langle 2, c\rangle, \langle 3, d\rangle\}$ и $\Omega = \langle H, Z, U \rangle$, $X = \{1, 2, 3\}$, $Y = \{a, b, c, d\}$, $H = \{\langle 1, a\rangle, \langle 1, b\rangle, \langle 2, b\rangle, \langle 2, c\rangle, \langle 3, a\rangle, \langle 3, d\rangle\}$. В этом случае соответствие Ω является продолжением соответствия Γ. На рис. 5.12 показано соответствие Γ, а на рис. 5.13 — его продолжение — соответствие Ω.

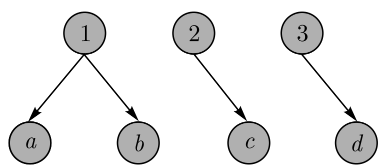


Рис. 5.12. Соответствие Γ

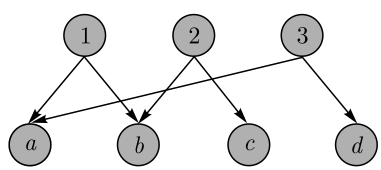


Рис. 5.13. Соответствие Ω (продолжение соответствия Γ)

Пример 5.9. Пусть задано соответствие $\Gamma = \langle F, X, Y \rangle$; $X = \{1, 2, 3\}$; $Y = \{a, b, c, d\}$; $F = \{\langle 1, a\rangle; \langle 1, c\rangle; \langle 2, b\rangle; \langle 2, c\rangle; \langle 3, d\rangle\}$. Найти образ элемента $1 \in X$ и множества $A = \{1, 2\}$ на заданном соответствии.

Ответ. Образом элемента 1 будет множество $\Gamma(1) = \{a, c\}$. Образ же множества A — следующее множество элементов: $\Gamma(A) = \{a, b, c\}$.

Пример 5.10. Для соответствия из предыдущего примера найти прообраз элемента $a \in Y$ и множества $B = \{a, b, c\}$.

Ответ. Прообразом элемента a будет множество $\Gamma^{-1}(a) = \{1\}$. Прообразом же множества $B = \{a, b, c\}$ — множество $\Gamma^{-1}(B) = \{1, 2\}$.

5.4. Доказательства тождеств с соответствиями

Исходя из свойств образа и прообраза произвольного множества при соответствии $\Gamma = \langle F, X, Y\rangle$ можно сказать, что будут истинны следующие тождества:

- $\Gamma(A \cup B) = \Gamma(A) \cup \Gamma(B)$;
- $\Gamma(A \cap B) \subseteq \Gamma(A) \cap \Gamma(B)$;
- $\Gamma(A \backslash B) \subseteq \Gamma(A) \backslash \Gamma(B)$;
- $\Gamma^{-1}(A \cup B) = \Gamma^{-1}(A) \cup \Gamma^{-1}(B)$;
- $\Gamma^{-1}(A \cap B) \subseteq \Gamma^{-1}(A) \cap \Gamma^{-1}(B)$;
- $\Gamma^{-1}(A \backslash B) \subseteq \Gamma^{-1}(A) \backslash \Gamma^{-1}(B)$.

Для доказательства тождеств с соответствиями, а также тождеств с образом и прообразом множества на заданном соответствии используются методы, уже рассмотренные нами при доказательстве тождеств с множествами.

Например, задано соответствие $\Gamma = \langle F, X, Y\rangle$ и произвольные множества A и B, для которых справедливо условие $A \subseteq X$, $B \subseteq Y$. Доказать справедливость тождества

$$\Gamma(A \cup B) = \Gamma(A) \cup \Gamma(B).$$

Доказательство. Воспользуемся методом взаимного включения. Докажем сначала, что $\Gamma(A \cup B) \subseteq \Gamma(A) \cup \Gamma(B)$. Предположим, что исходное тождество истинно. Тогда существует элемент y, для которого справедливо следующее высказывание:

$$y \in \Gamma(A \cup B) \to y \in \text{пр}_2\,(((A \cup B) \times Y) \cap F) \to$$

(согласно определению образа множества)

$$(\exists x \in X)(\langle x, y\rangle \in (((A \cup B) \times Y) \cap F)) \to$$

(согласно свойству операции проектирования)

$$\langle x, y\rangle \in ((A \cup B) \times Y)\,\&\,\langle x, y\rangle \in F \to$$

(исходя из свойств операции пересечения)

$$x \in (A \cup B)\,\&\,y \in Y\,\&\,\langle x, y\rangle \in F \to$$

(исходя из свойств прямого произведения множеств)

$$(x \in A \vee x \in B)\,\&\,y \in Y\,\&\,\langle x, y\rangle \in F \to$$

(исходя из свойств операции объединения)

$$x \in A \,\&\, y \in Y \,\&\, \langle x, y\rangle \in F \vee x \in B \,\&\, y \in Y \,\&\, \langle x, y\rangle \in F \to$$

(согласно закону дистрибутивности)

$$\langle x, y\rangle \in (A \times Y) \,\&\, \langle x, y\rangle \in F \vee \langle x, y\rangle \in (B \times Y) \,\&\, \langle x, y\rangle \in F \to$$
$$\langle x, y\rangle \in ((A \times Y) \cap F) \vee \langle x, y\rangle \in ((B \times Y) \cap F) \to$$
$$\langle x, y\rangle \in [((A \times Y) \cap F) \cup ((B \times Y) \cap F)] \to$$
$$y \in \text{пр}_2\,[((A \times Y) \cap F) \cup ((B \times Y) \cap F)] \to y \in \Gamma(A) \cup \Gamma(B).$$

Таким образом, мы доказали справедливость прямого включения.

Теперь докажем справедливость обратного включения:

$$\Gamma(A) \cup \Gamma(B) \subseteq \Gamma(A \cup B).$$

Допустим, что истинно выражение

$$y \in \Gamma(A) \cup \Gamma(B) \to y \in \Gamma(A) \vee y \in \Gamma(B) \to$$
$$y \in \text{пр}_2((A \times Y) \cap F) \vee y \in \text{пр}_2((B \times Y) \cap F) \to$$
$$(\exists x_i \in X)(\langle x_i, y\rangle \in ((A \times Y) \cap F) \vee (\exists x_j \in X)(\langle x_j, y\rangle \in ((B \times Y) \cap F) \to$$

(поскольку в общем случае x_i и x_j не обязательно совпадают)

$$(\langle x_i, y\rangle \in (A \times Y) \,\&\, \langle x_i, y\rangle \in F) \vee (\langle x_j, y\rangle \in (B \times Y) \,\&\, \langle x_j, y\rangle \in F) \to$$
$$(x_i \in A \,\&\, y \in Y \,\&\, \langle x_i, y\rangle \in F) \vee (x_j \in B \,\&\, y \in Y \,\&\, \langle x_j, y\rangle \in F) \to$$

$$((x_i \in A \vee x_j \in B) \,\&\, y \in Y \,\&\, \langle x_i, y\rangle \in F) \vee$$
$$\vee\, ((x_i \in A \vee x_j \in B) \,\&\, y \in Y \,\&\, \langle x_j, y\rangle \in F) \to$$

(согласно закону дистрибутивности)

$$((x \in (A \cup B) \,\&\, y \in Y \,\&\, \langle x, y\rangle \in F) \to$$

(согласно свойствам операции дизъюнкции для истинности выражения достаточно истинности хотя бы одного из его членов)

$$\langle x, y\rangle \in ((A \cup B) \times Y) \,\&\, \langle x, y\rangle \in F \to \langle x, y\rangle \in (((A \cup B) \times Y) \cap F) \to$$
$$y \in \text{пр}_2(((A \cup B) \times Y) \cap F) \to y \in \Gamma(A \cup B).$$

Таким образом, мы доказали справедливость обратного включения.

Следовательно, исходное тождество справедливо.

5.5. Основные свойства соответствий

Соответствие Γ называется *функциональным*, если его график G функционален. График G называется *функциональным*, если в нем нет пар с одинаковыми первыми и разными вторыми элементами. Другими словами, из элементов области отправления может выходить не более одной стрелки. На рис. 5.14–5.16 приведены примеры функциональных соответствий и на рис. 5.17 — нефункционального соответствия.

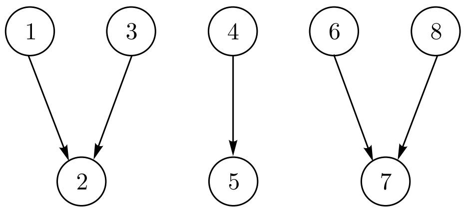


Рис. 5.14. Пример функционального соответствия

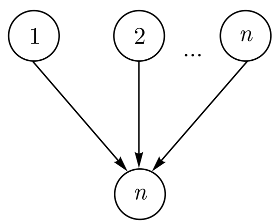


Рис. 5.15. Пример функционального соответствия

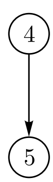


Рис. 5.16. Пример функционального соответствия

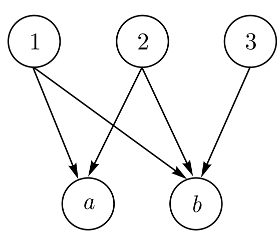


Рис. 5.17. Пример нефункционального соответствия

Следовательно, соответствие $\Gamma = \langle G, X, Y \rangle$ функционально тогда, когда истинно

$$(\forall x \in X, \forall y_1, y_2 \in Y)[\langle x, y_1 \rangle \in G \wedge \langle x, y_2 \rangle \in G \rightarrow y_1 = y_2].$$

Соответствие Γ называется *инъективным*, если его график инъективен. График G называется *инъективным*, если в нем нет пар с разными первыми и одинаковыми вторыми элементами. На рис. 5.18, 5.19 приведены примеры инъективных соответствий. На рис. 5.20 приведен пример неинъективного соответствия.

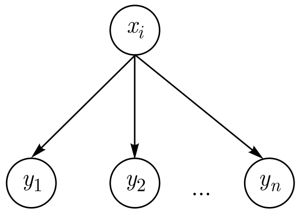


Рис. 5.18. Пример инъективного соответствия

Рис. 5.19. Пример инъективного соответствия

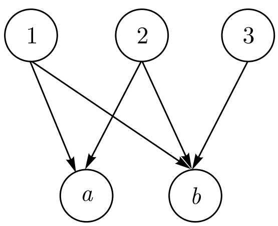


Рис. 5.20. Пример неинъективного соответствия

Отметим, что в частном случае инъективные и функциональные графики могут совпадать, например, графики соответствий на рис. 5.16 и 5.19.

Соответствие инъективно, когда справедливо

$$(\forall x_1, x_2 \in X, \forall y \in Y)[\langle x_1, y \rangle \in G \wedge \langle x_2, y \rangle \in G \rightarrow x_1 = x_2].$$

Соответствие $\Gamma = \langle G, X, Y \rangle$ называется *всюду определенным*, если его область определения совпадает с его областью отправления. Например, $\Gamma = \langle G, X, Y \rangle = \langle \{\langle 1, 2\rangle, \langle 3, 2\rangle, \langle 4, 5\rangle\}; \{1, 3, 4\}; \{2, 5\}\rangle$. Здесь область отправления соответствия $X = \{1, 3, 4\}$ совпадает с областью определения. На рис. 5.21 приведен пример всюду определенного соответствия.

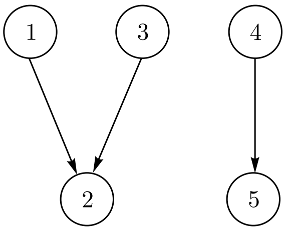


Рис. 5.21. Пример всюду определенного соответствия

Для всюду определенного соответствия справедливо выражение

$$\text{пр}_1\, G = X.$$

Аналогично можно записать

$$(\forall x)(\exists y)[\langle x, y\rangle \in G].$$

Соответствие $\Gamma = \langle G, X, Y \rangle$ называется *сюръективным*, если его область значений совпадает с его областью прибытия. Например, $\Gamma = \langle G, X, Y \rangle = \langle \{\langle 1, b\rangle, \langle 2, a\rangle\}; \{1, 2, 3\}; \{a, b\}\rangle$. Здесь область прибытия соответствия $X = \{a, b\}$ совпадает с областью значений. На рис. 5.22 приведен пример сюръективного соответствия.

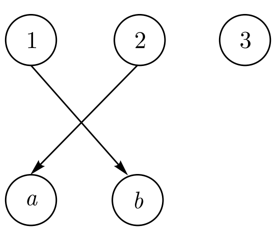


Рис. 5.22. Пример сюръективного соответствия

Для сюръективного соответствия справедливо выражение

$$\text{пр}_2\, G = Y.$$

Аналогично можно записать

$$(\forall y)(\exists x)[\langle x, y\rangle \in G].$$

Соответствие $\Gamma = \langle G, X, Y \rangle$ называется *биективным* соответствием или *биекцией*, или *взаимно-однозначным соответствием*, если оно функционально, инъективно, всюду определено и сюръективно. На рис. 5.23–5.25 показаны примеры взаимно-однозначных (биективных) соответствий, а на рис. 5.26 — соответствия, не являющиеся биективными.

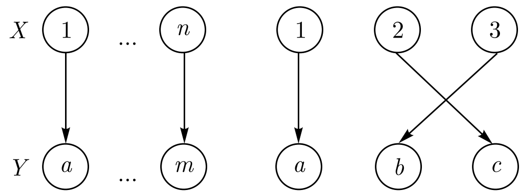


Рис. 5.23. Пример взаимно-однозначного соответствия

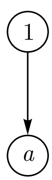


Рис. 5.24. Пример взаимно-однозначного соответствия

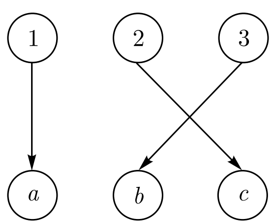


Рис. 5.25. Пример взаимно-однозначного соответствия

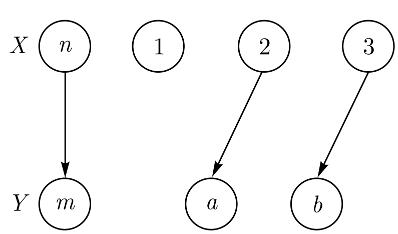


Рис. 5.26. Пример небиективного соответствия

Частным случаем соответствия является понятие *отображения*. Всюду определенное соответствие называется *отображением* X в Y и записывается

$$G : X \to Y.$$

Если в отображении G множества X и Y совпадают, получаем отображение множества X самого в себя:

$$G : X \to X.$$

Приведем пример. Пусть $X = \{a_1, a_2, b_1, b_2\}$ — множество, состоящее из двух пар родителей. Обозначим $X_1 = \{a_{1-2,1}, a_{1-2,2}, b_{1-2,1}, b_{1-2,2}, b_{1-2,3}\}$ — множество детей этих пар родителей, в которое входят как дети пары a, так и дети пары b. $X_2 = \{ab_{2-1,1}, ab_{2-1,2}\}$ — множество внуков этих родителей. Построив отображения, получим пример генеалогического дерева (рис. 5.27).

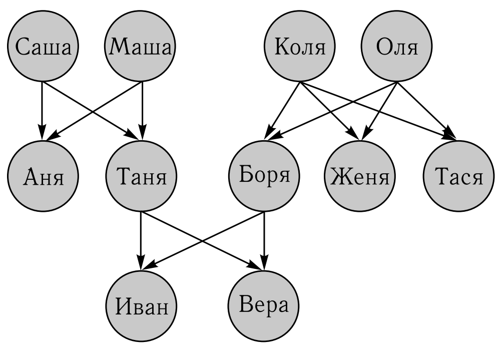


Рис. 5.27. Пример генеалогического дерева для двух пар родителей

Запишем упорядоченные пары, находящиеся в отношениях на множестве членов этой семьи.

Пусть $T = \{(x, y)\colon x$ — дедушка $y\}$. Тогда T содержит следующие кортежи: $\langle$Саша, Иван$\rangle$; $\langle$Саша, Вера$\rangle$; $\langle$Коля, Иван$\rangle$; $\langle$Коля, Вера$\rangle$.

Пусть $Q = \{(x, y)\colon x$ — брат $y\}$. Тогда Q содержит следующие кортежи: $\langle$Боря, Женя$\rangle$.

Для того чтобы доказать, что между заданными множествами X и Y можно построить взаимно-однозначное соответствие, нужно доказать, что существует такой график $G \subseteq X \times Y$, чтобы тройка $\langle G, X, Y\rangle$ была биекцией.

Пример 5.11. Привести примеры функционального, нефункционального и антифункционального соответствий.

Решение. Соответствие $\Gamma = \langle F, X, Y\rangle$ называется *функциональным*, если образ любого элемента $x \in X$ — $\Gamma(x)$ содержит не более одного элемента, т. е. график такого соответствия не содержит пар с одинаковыми первыми и разными вторыми элементами. В противном случае соответствие является *нефункциональным*. В случае, если каждому элементу $x \in X$ соответствует более одного элемента из Y, соответствие называется *антифункциональным*.

На рис. 5.28, *а–в* даны примеры функционального, нефункционального и антифункционального соответствий.

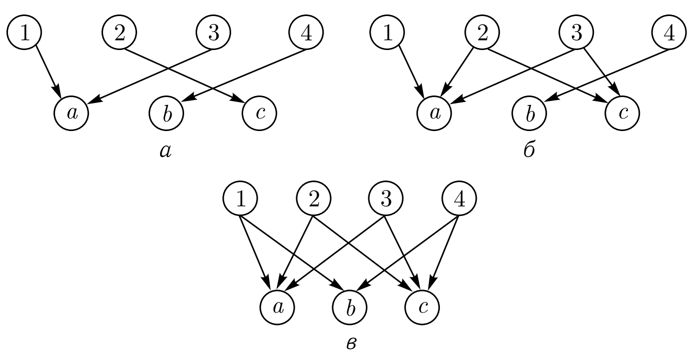


Рис. 5.28. Примеры функционального, нефункционального и антифункционального соответствий

Пример 5.12. Привести примеры инъективного, неинъективного и антиинъективного соответствий.

Решение. Соответствие $\Gamma = \langle F, X, Y\rangle$ называется *инъективным*, если прообраз $\Gamma^{-1}(y)$ любого элемента $y \in Y$ содержит не более одного элемента из X, т. е. график такого соответствия не содержит пар с одинаковыми вторыми и разными первыми элементами.

По аналогии с нефункциональными и антифункциональными соответствиями можем определить понятие *неинъективного* и *антиинъективного* соответствий.

На рис. 5.29, *а–в* показаны примеры инъективного, неинъективного и антиинъективного соответствий.

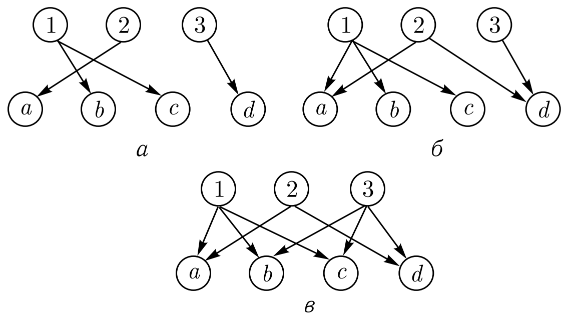


Рис. 5.29. Примеры инъективного, неинъективного и антиинъективного соответствий

Пример 5.13. Привести примеры всюду определенного, не всюду определенного, сюръективного и несюръективного соответствий.

Решение. Соответствие $\Gamma = \langle F, X, Y\rangle$ называется *всюду определенным*, если для каждого $x \in X$ его образ $\Gamma(x)$ не равен пустому множеству, т. е. в графике такого соответствия из любой вершины $x \in X$ выходит по крайней мере одна стрелка. В противном случае соответствие является *не всюду определенным*. Соответствие $\Gamma = \langle F, X, Y\rangle$ называется *сюръективным*, если для любого $y \in Y$ его прообраз $\Gamma^{-1}(y)$ не равен пустому множеству, т. е. в графике такого соответствия в любую вершину $y \in Y$ входит хотя бы одна стрелка. В противном случае соответствие называется *несюръективным*.

Все соответствия, показанные на рис. 5.28, 5.29, являются всюду определенными и сюръективными. Примеры не всюду определенного и несюръективного соответствий приведены на рис. 5.30 и 5.31.

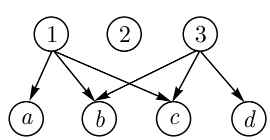


Рис. 5.30. Пример не всюду определенного соответствия

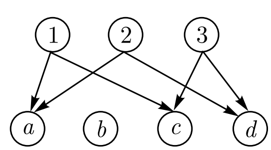


Рис. 5.31. Пример несюръективного соответствия

Пример 5.14. Привести пример взаимно-однозначного соответствия.

Решение. Соответствие называется *биективным* или *взаимно-однозначным*, если оно функционально, инъективно, всюду определено и сюръективно. Пример биективного соответствия показан на рис. 5.32.

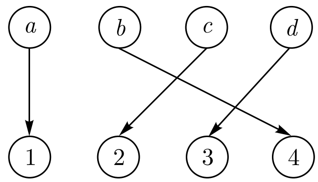


Рис. 5.32. Пример взаимно-однозначного соответствия

Существует много способов установления взаимно-однозначного соответствия. Например, часто употребляемым является лексикографический метод. Он устанавливает соответствие в порядке следования. Тогда для примера, показанного на рис. 5.32, получим следующее взаимно-однозначное соответствие:

$$a \to 1, \quad b \to 4, \quad c \to 2, \quad d \to 3.$$

Пример 5.15. Задано произвольное множество $X = \{x_1, x_2, x_3, x_4, x_5\}$. Привести пример произвольного взаимно-однозначного соответствия на множестве X. Взаимно-однозначное соответствие на одном множестве называется подстановкой.

Ответ. Можем задать на множестве X следующую произвольную подстановку t:

$$t = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ x_4 & x_3 & x_1 & x_5 & x_5 \end{pmatrix},$$

т. е. $x_1 \to x_4$, $x_2 \to x_3$, $x_3 \to x_1$, $x_4 \to x_5$, $x_5 \to x_5$.

5.6. Функция

Понятие функции ввели Ферма и Декарт. Полное понятие функции привел Леонард Эйлер в XIX веке.

Функция — это функциональное соответствие, т. е. соответствие с функциональным графиком. $F = \langle \Phi, X, Y \rangle$, Φ — функциональный график.

Функцией из множества X в множество Y называется соответствие, где каждый элемент множества X связан с единственным элементом множества Y. Другими словами, для каждого $x \in X$ в соответствии существует ровно одна пара вида $\langle x, y \rangle$. В графической интерпретации из каждого элемента X выходит ровно одна стрелка.

Говорят, что функция — это кортеж длины 3, в котором первая компонента является подмножеством прямого декартового произведения второй и третьей компонент и первая компонента — функциональный график.

На рис. 5.33 изображена функция из множества $X = \{a, b, c, d, e\}$ в множество $Y = \{1,2,3\}$, состоящая из пар $\{\langle a,1\rangle, \langle b, 1\rangle, \langle c, 2\rangle, \langle d, 3\rangle, \langle e, 3\rangle\}$.

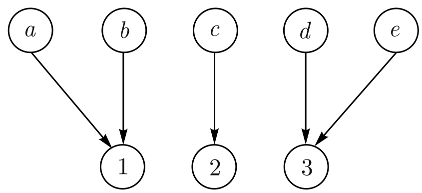


Рис. 5.33. Пример функции из множества X в множество Y

Выражение для определения функции можно записать в виде

$$F = \langle \Phi, X, Y\rangle,$$

где Φ — функциональный график, $\Phi \subseteq X \times Y$.

Выражение, обозначающее функцию, иногда записывают так:

$$X \xrightarrow{F} Y \quad \text{или} \quad F\colon X \to Y.$$

Другими словами, если имеется некоторый элемент $\langle a, b\rangle \in \in F$, то говорят, что функция отображает элемент a в элемент b. Или говорят, что значение функции $(F(a))$ на элементе a равно $F(a) = b$.

Функция F называется *тождественной*, если

$$\Phi = \Delta X \,\&\, X = Y,$$

где ΔX — диагональ на области $X \times X$.

Функция F называется *константной*, если существует такой элемент $b \in Y$, что график $\Phi = X \times \{b\}$.

Функция F тогда и только тогда *всюду определена*, когда

$$(\forall x \in X)[\exists F(x)].$$

То есть для любого элемента множества X всегда существует как минимум одно отображение.

Функция F тогда и только тогда *биективна*, когда она всюду определена, сюръективна и инъективна.

Отметим следующее: для функции справедливы операции инверсии и композиции. Композиция функции также будет являться функцией. Для инверсии же существует следующее правило.

Инверсия F^{-1} функции F, т. е. $F^{-1} = \langle \Phi^{-1}, Y, X \rangle$ необязательна должна быть функцией. Существует *критерий функциональности* инверсий.

Инверсия F^{-1} функции F тогда и только тогда является функцией, когда функция F инъективна.

Для любых значений функций F_1 и F_2 справедливо

$$(F_1 \cdot F_2)(x) = F_2(F_1(x)).$$

Рассмотрим понятие обратной функции. Обратной функцией является функция F^{-1}.

Отметим три основных свойства F^{-1}:

1. F^{-1} является инверсией.
2. $F \cdot F^{-1}$ тождественна на $\text{пр}_1 F$.
3. $F^{-1} \cdot F$ тождественна на $\text{пр}_2 F$.

Приведем *принцип Дирихле*. Пусть $f\colon X \to Y$ — функция, причем X и Y — конечные множества. Принцип Дирихле гласит, что если $|X| > |Y|$, то по крайней мере одно значение f встретится более одного раза. Неформально принцип Дирихле можно например записать следующим образом.

Если X — множество белок, а Y — множество клеток, и $|X| = 12$, а $|Y| = 11$, то 12 белок нельзя посадить в 11 клеток так, чтобы в каждой клетке находилась одна белка.

Контрольные вопросы

1. Приведите определение и способы задания соответствия.
2. Как называется и обозначается соответствие $\Gamma = (F, \varnothing, X)$?
3. Из каких элементов образуются матрицы R_Γ соответствий $\Gamma_\varnothing$ и $\Gamma_\text{п}$?
4. Покажите, как выполняются операции над соответствиями.
5. Что такое инверсия соответствия и композиция соответствий?
6. В каких случаях композиция соответствий приводит к соответствию с пустым графиком?
7. В каком случае образ множества при данном соответствии является пустым множеством?
8. Как найти образ и прообраз множества при данном соответствии?
9. Определите сужение и продолжение соответствий.

10. Приведите примеры доказательства тождеств с соответствиями.

11. Определите понятие отображения.

12. Какое соответствие называется:

а) функциональным;

б) инъективным;

в) всюду определенным;

г) сюръективным?

13. Возможно ли нефункциональное, неинъективное, не всюду определенное соответствие? Если да, привести пример.

14. Определите понятие функция.

15. Запишите критерий функциональности инверсий.

16. Поясните принцип Дирихле.

Задания для самостоятельной работы

1. Задать произвольное соответствие $\Gamma = (F, X, Y)$, $|X| = 5$, $|Y| = 5$, $|F| = 9$ теоретическим, матричным и графическим способом.

2. Заданы соответствия $\Gamma = \langle F, X, Y \rangle$; $\Delta = \langle P, W, Z \rangle$, где $X = \{1, 2, 3, 4\}$; $Y = \{a, b, c, d\}$; $F = \{\langle 1, a\rangle; \langle 1, c\rangle; \langle 1, d\rangle; \langle 2, b\rangle; \langle 2, c\rangle; \langle 3, a\rangle; \langle 3, d\rangle; \langle 4, b\rangle; \langle 4, c\rangle\}$; $W = \{1, 3, 5, 6\}$; $Z = \{b, c, d, e\}$; $P = \{\langle 1, b\rangle; \langle 1, c\rangle; \langle 3, b\rangle; \langle 3, d\rangle; \langle 3, e\rangle; \langle 5, c\rangle; \langle 5, d\rangle; \langle 6, d\rangle\}$.

Найти:

а) $\Gamma \cup \Delta$;

б) $\Gamma \cup \Delta^{-1}$;

в) $\Gamma \cap \Delta$;

г) $\Gamma^{-1} \cap \Delta$;

д) $\Gamma \backslash \Delta$;

е) $\Gamma \backslash \Delta^{-1}$;

ж) $\Gamma^{-1} \backslash \Delta$;

з) $\Gamma^{-1} \backslash \Delta^{-1}$.

3. Заданы соответствия $\Gamma = \langle F, X, Y \rangle$; $\Delta = \langle P, W, Z \rangle$, где $X = \{1, 2, 3, 4\}$; $Y = \{a, b, c, d\}$; $F = \{\langle 1, a\rangle; \langle 1, c\rangle; \langle 1, d\rangle; \langle 2, b\rangle; \langle 2, c\rangle; \langle 3, a\rangle; \langle 3, d\rangle; \langle 4, b\rangle; \langle 4, c\rangle\}$; $W = \{a, c, d, e\}$; $Z = \{\mathrm{I}, \mathrm{II}, \mathrm{IV}, \mathrm{V}, \mathrm{VI}\}$; $P = \{\langle a, \mathrm{I}\rangle; \langle a, \mathrm{IV}\rangle; \langle a, \mathrm{V}\rangle; \langle c, \mathrm{II}\rangle; \langle c, \mathrm{IV}\rangle; \langle d, \mathrm{II}\rangle; \langle d, \mathrm{V}\rangle; \langle d, \mathrm{VI}\rangle; \langle e, \mathrm{I}\rangle\}$ и произвольные множества $A = \{1, 2, 4\}$ и $B = \{a, c, d\}$.

Найти:

а) $\Gamma \cdot \Delta$;

б) $\Delta^{-1} \cdot \Gamma^{-1}$;

в) $\Gamma(A)$;

г) $\Gamma^{-1}(B)$;

4. Пусть задано произвольное соответствие $\Gamma = (F, X, Y)$ и множества $A \subseteq X$, $B \subseteq X$, $C \subseteq Y$, $D \subseteq Y$.

Доказать справедливость следующих тождеств:
а) $\Gamma(A \cap B) \subseteq \Gamma(A) \cap \Gamma(B)$;
б) $\Gamma(A) \backslash \Gamma(B) \subseteq \Gamma(A \backslash B)$;
в) $\Gamma^{-1}(C \cap D) \subseteq \Gamma^{-1}(C) \cap \Gamma^{-1}(D)$;
г) $\Gamma^{-1}(C) \backslash \Gamma^{-1}(D) \subseteq \Gamma^{-1}(C \backslash D)$.

Соответствие определяет связь между любыми объектами в природе, и ее модели помогают решать оптимизационные задачи.

Если область прибытия соответствия совпадает с его областью отправления, то соответствие преобразуется в отношение.

Глава 6
УПОРЯДОЧЕННЫЕ БЕСКОНЕЧНЫЕ МНОЖЕСТВА

> Совершенство только тогда совершенство, когда оно представляется достижимым только в бесконечности и когда поэтому возможность приближения к нему бесконечна.
>
> *Л. Толстой*

Понятие упорядоченного бесконечного множества, *определение и способы их задания*, *интервал*, *сегмент*, *соответствие подобия*, *сечение*, *скачок*, *щель*, *дедекиндово сечение*, *принцип трансфинитной индукции*, *проблема континуума*, *счетные и несчетные множества*, *мощность континуума*, *алгебраические и трансцендентные числа*

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать основные сведения об упорядоченных бесконечных множествах;
- иметь понятия об интервале и сегменте;
- уметь определять 4 типа сечений;
- уметь строить счетные и несчетные множества;
- знать, что представляет собой проблема континуума;
- иметь понятие о парадоксах в теории множеств;
- иметь понятие о принципе трансфинитной индукции и уметь применять его на практике.

6.1. Основные сведения об упорядоченных бесконечных множествах

Простейшим примером *упорядоченных бесконечных множеств* (УБМ) является множество всех целых, натуральных, рациональных, неотрицательных и так далее чисел. Одно и то же бесконечное множество можно упорядочить различными способами.

Пусть множество X является УБМ, причем элементы $a, b, x \in \in X$. Если справедлива запись $a \to x$ и $x \to b$ $(a \to x \to b)$, то говорят, что элемент x лежит между элементами a и b. Множество всех элементов, лежащих между a и b, называется *интервалом* (a, b) упорядоченного бесконечного множества X.

Если к интервалу (a, b) добавить два его конца, т. е. элементы a и b, то получим *сегмент* $[a, b]$. Два элемента a и b множества X называются *соседними*, если интервал (a, b) пустой. Если элемент из упорядоченного бесконечного множества X таков, что $(\forall x \in X)(x \neq a \wedge a \Rightarrow x)$, то a является первым или *наименьшим элементом* X.

В противоположном случае, если $(\forall x \in X)(x \neq a \wedge x \Rightarrow a)$, то a является *наибольшим элементом*.

Очевидно, что в каждом УБМ может быть не больше одного наименьшего и одного наибольшего элемента. Например, в множестве положительных чисел наименьшим элементом будет ноль. Соответственно в множестве неположительных чисел ноль — наибольший элемент.

Для отношения порядка в УБМ справедливо следующее выражение:

$$x \Rightarrow y \to f(x) \Rightarrow f(y).$$

Другими словами, взаимно-однозначное отображение УБМ X на УБМ Y сохраняет порядок, т. е. $x \to y$ влечет, что $f(x) \to f(y)$, где $x \in X$, $y \in Y$. Такое отображение часто называют *соответствием подобия*. Очевидно, что два подобных упорядоченных бесконечных множества эквивалентны между собой.

Введем понятие сечения. *Сечением* упорядоченного бесконечного множества X называется разбиение этого множества на два непустых подмножества A и B таких, что каждый элемент одного множества низшего класса всегда предшествует каждому элементу второго множества верхнего класса.

Существует 4 типа сечений.

1. В нижнем классе A есть наибольший элемент (a), а в верхнем классе B есть наименьший элемент (b). Такое сечение называется *скачком*. В этом случае интервал (a, b) является пустым.

2. В нижнем классе A есть a, а в верхнем классе B нет b.

3. В нижнем классе A нет a, а в верхнем классе B есть b. Второй и третий классы называются *дедекиндовыми* сечениями.

4. В нижнем классе A нет a, а в верхнем классе B нет b. Такое сечение называется *щелью*.

Рассмотрим принцип *трансфинитной индукции*.

Пусть дано упорядоченное бесконечное множество X и имеется высказывание $P = P(x)$, которое зависит от этого множества. Если высказывание P верно для первого элемента $x_0 \in X$ и если из того, что оно верно для всех элементов $x \in X$, предшествующих данному, следует, что предположение P верно и для элемента x', то высказывание P верно для любого элемента, принадлежащего множеству X.

Когда множество X является множеством натуральных чисел, принцип трансфинитной индукции превращается в обычный принцип полной индукции.

Приведем пример. Докажем по индукции, что равенство

$$\frac{1+2+\ldots+n}{2} = \frac{n(n+1)}{4}$$

выполняется при всех натуральных n.

Пусть $P(n)$ — предикат $\dfrac{1+2+\ldots+n}{2} = \dfrac{n(n+1)}{4}$. При $n = 1$ левая часть равна $1/2$, а правая часть $(1(1+1))/4$ также равна $1/2$. Следовательно, выражение при $P(1)$ истинно.

Предположим, что равенство $\dfrac{1+2+\ldots+n}{2} = \dfrac{n(n+1)}{4}$ справедливо для произвольного числа m. Тогда

$$\frac{1+2+\ldots+m}{2} + \frac{m+1}{2} = \frac{(m+1)(m+2)}{4}.$$

Тогда при любом натуральном m импликация $P(m) \to P(m+1)$ истинна. Следовательно, на основании принципа математической индукции мы показали, что предикат $P(n)$ — истина для любых натуральных n.

Приведем ряд известных положений об УБМ.

Каждое множество, содержащее бесконечное подмножество, бесконечно. Тогда очевидно, что множество всех подмножеств бесконечного множества также бесконечно.

Каждое УБМ имеет собственное подмножество, которому оно равномощно, и наоборот, если множество равномощно своему подмножеству, то оно бесконечно.

Никакое множество не является равномощным множеству всех своих подмножеств.

Произвольное множество называется не более чем счетным, если оно конечно или равномощно множеству натуральных чисел.

Георг Кантор предположил способ сравнения двух упорядоченных бесконечных множеств, основанный на установлении взаимно-однозначного соответствия между элементами этих множеств. Это выполняется объединением элементов одного множества в пары с элементами другого множества так, что каждый элемент входит в одну-единственную пару.

Если для двух УБМ установлено взаимно-однозначное соответствие, то эти множества равнозначны, т. е. имеют одинаковое количество элементов, или $|A| = |B|$.

В УБМ встречаются парадоксы.

Например, некоторые множества могут быть равномощными своей части. Например, можно установить взаимно-однозначное соответствие между множеством натуральных чисел $\mathbb{N}$ и множеством их квадратов:

$$\begin{array}{ll} \mathbb{N} & 0, 1, 2, 3, 4, 5, \ldots \\ \mathbb{N}^2 & 0, 1, 4, 9, 16, 25, \ldots \end{array}$$

В теории множеств натуральный ряд чисел принят за эталон класса бесконечных множеств.

6.2. Проблема континуума

Любое множество, равномощное натуральному ряду, называется *счетным*. Считают, что множество действительных чисел является *несчетным* множеством.

Множество, равномощное множеству действительных положительных чисел, не превосходящих единицу, называется множеством *мощности континуума*, или *континуальным*.

Пусть имеем множество натуральных чисел. Тогда обозначим через $\mathbb{N}$ его мощность. Обозначим через $\mathbb{K}$ мощность множества действительных положительных чисел, не превосходящих единицу.

Пусть имеется произвольное бесконечное множество $\mathbb{M}$. Тогда мощность $|\mathbb{M}|$ равна произвольному числу m. В этом случае число всех подмножеств множества $\mathbb{M}$ будет равно 2^m.

Покажем теперь, что несчетные множества существуют и причем с различными мощностями. Эталоном несчетного множества является множество точек на прямой линии. Покажем, что множество всех точек на прямой линии несчетно.

Рассмотрим множество всех действительных чисел, так как каждой точке прямой можно поставить в соответствие одно действительное число и наоборот. Каждое действительное число запишем в виде бесконечной дроби вида $0, \alpha_1\, \alpha_2\, \alpha_3 \ldots$ Предположим, что удалось пронумеровать все действительные числа. Для опровержения этого предположения надо найти хотя бы одно не пронумерованное действительное число.

Согласно известной эвристике способ построения заключается в использовании двух правил:

1. Если $\alpha = 1$, то ставят 2.

2. Если $\alpha \neq 1$, то ставят 1.

Возьмем ноль и поставим после него запятую. Рассмотрим первую цифру, если эта цифра отлична от 1, то в соответствующий ей разряд ставим 1, иначе 2. И далее процесс продолжается аналогично до просмотра всего действительного числа. Очевидно, что построенное число не получило никакого номера, так как в первом десятичном знаке оно отличается от числа с номером 1, во втором — от числа с номером 2, в n-знаке — от числа с номером n и т. д.

Например, имеем несколько действительных чисел:

1) $0,\ 2\ 7\ 3\ 6\ 1 \ldots$

2) $0,\ 4\ 1\ 1\ 2\ 2 \ldots$

3) $0,\ 6\ 7\ 3\ 1\ 1 \ldots$

4) $0,\ 7\ 1\ 1\ 4\ 4 \ldots$

5) $0,\ 9\ 1\ 8\ 1\ 9 \ldots$

Тогда число, не получившее никакого номера, может иметь такой вид 0,12111...

В упорядоченных бесконечных множествах выделяют понятия алгебраического и трансцендентного числа. *Алгебраическими* называются числа, которые являются корнями уравнения вида $a_0x^n + a_1x^{n-1} + \ldots + a_kx + a_{k+1} = 0$, т. е. корнями уравнений с целыми коэффициентами. Числа, которые не являются корнями таких уравнений, называются *трансцендентными числами*. Известным трансцендентным числом является число $\pi \approx 3,14159\ldots$

Оказалось, что алгебраических чисел относительно мало (так как это счетные множества), а трансцендентных относительно много (это несчетные множества). Луи Виль привел алгоритм построения трансцендентных чисел. Он заключается в том, что в заданном действительном числе после первой единицы ставят ноль, после второй — два нуля, после третьей шесть, после n-й — $n!$ нулей. Пример трансцендентного числа — число 0,101000100000001...

Существует утверждение, что на коротком и длинном отрезке точек поровну. Это следует из того, что всегда можно установить взаимно-однозначное соответствие между точками любых отрезков. Следовательно, любая геометрическая фигура, содержащая хоть одну линию, имеет столько же точек, сколько и отрезок!

Множества точек, расположенных на отрезках, получили название *множество мощности континуума* (континуум — от лат. непрерывный). Множество мощности континуума самое большое несчетное множество.

Проблема континуума заключается в попытке ответить на вопрос: существует ли промежуточное множество, у которого больше элементов, чем множество натуральных чисел, и меньше элементов, чем множество точек на прямой.

Рассмотрим некоторые операции над мощностями конечных и бесконечных множеств. Часто мощность множества $\mathbb{M} = m$ называют *кардинальным числом* и обозначают $\operatorname{card} \mathbb{M} = m$.

Пусть заданы конечные множества A и B, $|A| = m$, $|B| = n$, $\mathbb{N}$ — мощность множества натуральных чисел (счетного множества), $\mathbb{K}$ — мощность несчетного множества (континуума).

Отметим, что мощности можно складывать так же, как складывают натуральные числа.

Тогда имеем

1. $m + n = n + m$,
2. $m + \mathbb{N} = \mathbb{N}$,
3. $\mathbb{N} + \mathbb{N} = \mathbb{N}$,
4. $\mathbb{N} + \mathbb{K} = \mathbb{K}$,
5. $\mathbb{K} + \mathbb{K} = \mathbb{K}$.

Второе выражение означает, что сумма конечного и счетного множеств является счетным множеством. Третье выражение означает, что сумма мощностей счетных множеств равна мощности счетного множества. Четвертое выражение означает, что сумма мощности счетного множества и множества мощности континуума дает множество мощности континуума. Пятое выражение означает, что сумма множеств мощности континуума дает множество мощности континуума.

Аналогичные выражения существуют при декартовом произведении мощностей:

6. $m \times n = n \times m$,
7. $\mathbb{N} \times \mathbb{N} = \mathbb{N}$,
8. $\mathbb{N} \times \mathbb{K} = \mathbb{K}$,
9. $\mathbb{K} \times \mathbb{K} = \mathbb{K}$.

Седьмое выражение означает, что если множества счетные, то и множества всех пар из этих множеств счетно. Это другая формулировка утверждения, что сумма счетных множеств является счетным множеством. Равенство (9) означает, что число точек на отрезке и в квадрате одно и то же (например, пусть k — это число точек на отрезке, тогда $k \times k$ — число точек в декартовом произведении отрезка на себя, т. е. число точек в квадрате).

Очевидно, что все натуральные числа содержатся среди кардинальных чисел, но существуют кардинальные числа, не являющиеся натуральными. Приведем ряд известных теорем Кантора, доказательство которых приведено в специальной литературе, указанной в конце данной части.

Теорема 1. *Для любого кардинального числа m справедливо неравенство $m < 2^m$.*

Теорема 2. *Мощность m' подмножества множества, имеющего мощность m, удовлетворяет неравенству $m' \leqslant m$.*

В теории множеств обнаружены противоречия (парадоксы), называемые *антиномией*. Приведем наиболее популярные из них.

Парадокс Кантора. Пусть $\mathbb{M}$ — множество всех множеств, $\mathrm{card}\,\mathbb{M} = m$. По теореме 1 кардинальное число множества его подмножеств 2^m удовлетворяет условию $2^m > m$. С другой стороны, подмножества $\mathbb{M}$ являются множествами и, следовательно, элементами $\mathbb{M}$, а их множество тогда является подмножеством множества $\mathbb{M}$. Тогда по теореме 2 должно выполняться условие $2^m \leqslant m$, что приводит к противоречию.

Парадокс Рассела. Пусть имеется множество мужчин $A = \{a_1, a_2, \ldots, a(p), a(k), a(p+1), a(p+2), \ldots, a(t)\}$, здесь $a(k)$ — парикмахер. И задано два подмножества: $A_1 = \{a_1, a_2, \ldots, a(p)\}$ — подмножество мужчин, которые бреются сами. $A_2 = \{a(p+1), a(p+2), \ldots, a(t)\}$ — подмножество мужчин, которые не бреются сами. Парикмахеру приказано брить тех и только тех мужчин, которые не бреются сами.

Тогда возникает парадокс: ни в множество A_1, ни в множество A_2 нельзя внести парикмахера (элемент $a(k)$). Так как приказано брить только тех, кто себя не бреет, то побрить себя нельзя. Не брить себя тоже нельзя, так как приказано брить всех, кто себя не бреет.

Эти и другие парадоксы возникают при определенных способах приписывания множествам числовых характеристик или при других условиях.

Приведем ряд основных аксиом о бесконечных множествах.

А1. *Упорядоченное бесконечное множество, не являющееся счетным, считается несчетным.*

А2. *Каждое несчетное множество бесконечно.*

А3. *Каждое бесконечное подмножество счетного множества счетно.*

А4. *Бесконечное подмножество счетного множества счетно.*

А5. *Объединение и непустое пересечение счетной совокупности счетных множеств являются счетными множествами.*

А6. *Множества целых* ($\mathbb{Z}$) *и рациональных чисел* ($\mathbb{Q}$) *являются счетными множествами.*

А7. *Множество действительных чисел* ($\mathbb{R}$) *и множество комплексных чисел* ($\mathbb{C}$) *являются несчетными множествами.*

Существует стандартная система аксиом теории множеств, в рамках которой излагают все общепринятые в математике способы рассуждений. Эта система позволяет строить все математические объекты на основе пустого множества. Такая система носит название системы *аксиом Цермело–Френкеля* (ZF). Аксиоматика ZF позволяет определять и реализовывать теоретико-множественные операции.

Примеры решения задач

Пример 6.1. Установим взаимно-однозначное соответствие между множеством $\mathbb{N}^2$ и множеством $\mathbb{A}$ отрицательных чисел.

Решение. Необходимо построить соответствие Г. Областью отправления будет $\mathbb{N}^2$, а областью прибытия — $\mathbb{A}$. Теперь необходимо построить график Ф. При создании пар $\langle n^2, -n\rangle$ и включении их в Ф будет получено искомое соответствие $\Gamma = \langle \Phi, \mathbb{N}^2, \mathbb{A}\rangle$, $n = \{1, 2, 3, \ldots\}$.

Пример 6.2. Установить взаимно-однозначное соответствие между сегментами $[a, b]$ и $[x, y]$.

Решение. Выбирается произвольная точка плоскости C и через нее проводятся линии, проходящие через два сегмента (рис. 6.1). Это позволяет установить взаимно-однозначное соответствие между любыми произвольными интервалами, полусегментами вида $[a, b)$, $[x, y)$ или $(a, b]$, $(x, y]$.

Пример 6.3. Установить взаимно-однозначное соответствие между произвольными полусегментами $[a, b)$ и $[x, y)$.

Решение. Создадим произвольное взаимно-однозначное соответствие $\Gamma = \langle \Phi, (a, b), (x, y)\rangle$ между интервалами (a, b), (x, y)

и добавим к графику Φ кортеж $\langle a, x \rangle$. Тогда соответствие $\Gamma_1 = \langle \Phi \cup \{\langle a, y \rangle\}, [a, b), [x, y) \rangle$ будет искомым.

Пример 6.4. Имеем множество A, состоящее из всех точек прямой линии, и множество B, состоящее из всех точек полуокружности $x^2 + (y-1)^2 = 1$, $y < 1$ с центром в точке $(0, 1)$ (рис. 6.2). В связи с тем, что $y < 1$, концы полуокружности (точки $(1, 1)$ и $(-1, 1)$) не принадлежат к ней.

Решение. Устанавливаем взаимно-однозначное соответствие между множествами A и B следующим образом. Каждой точке a_i прямой ставим в соответствие ту точку b_i полуокружности, в которой эту окружность пересекает луч, соединяющий центр круга с точкой a_i.

Пример 6.5. Показать, что множество $\mathbb{Q}$ всех пар натуральных чисел счетно.

Примечание. Высотой кортежа $\langle a, b \rangle$ называется натуральное число $a + b$.

Решение. Известно, что имеется $n - 1$ кортежей данной высоты n $(n > 1)$. Перечислим их:

$$\langle 1, n-1 \rangle,\ \langle 2, n-2 \rangle, \ldots, \langle n-1, 1 \rangle.$$

Обозначим через $\mathbb{Q}_n$ множество всех кортежей высоты n. Тогда видно, что $\mathbb{Q}$ есть сумма счетного множества конечных множеств $\mathbb{Q}_n$, т. е. счетное множество.

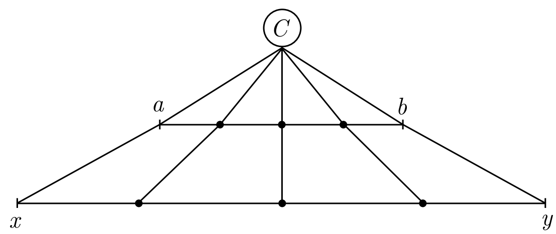


Рис. 6.1. Пример установления взаимно-однозначного соответствия

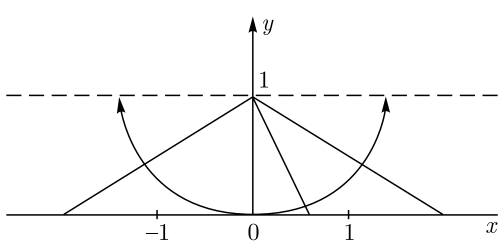


Рис. 6.2. Пример установления взаимно-однозначного соответствия

Контрольные вопросы

1. Что такое упорядоченное бесконечное множество?
2. Приведите способы упорядочивания бесконечных множеств.
3. Поясните понятия «интервал» и «сегмент».
4. Что такое соответствие подобия?
5. Поясните понятие сечения. Приведите примеры.
6. Поясните работу принципа трансфинитной индукции.
7. Каким образом можно сравнить два упорядоченных бесконечных множества?
8. Приведите примеры счетных и несчетных множеств.
9. В чем заключается проблема континуума?
10. Приведите эвристику, показывающую, что множество всех точек прямой линии несчетно.
11. Приведите примеры алгебраических и трансцендентных чисел.
12. Приведите операции над мощностями конечных и бесконечных множеств.
13. В чем заключается теорема Кантора о кардинальных числах?
14. Приведите примеры антиномий.

Задания для самостоятельной работы

1. Постройте примеры упорядоченных бесконечных множеств.
2. Постройте взаимно-однозначное соответствие между множествами натуральных чисел и их кубами.
3. Приведите операции над мощностями бесконечных множеств. Докажите их справедливость.
4. Постройте примеры подобных упорядоченных бесконечных множеств.
5. Докажите счетность или несчетность натуральных чисел.
6. Докажите несчетность множества точек прямой.
7. Постройте основные виды сечений упорядоченных бесконечных множеств.
8. Определите мощность множества точек квадрата.
9. Определите мощность множества точек куба.
10. Докажите существование несчетных множеств с различными мощностями.

11. Доказать на основе индукции истинность предиката $P(n)$:

$$\sum_{k=1}^{n} k = \frac{1}{2}n(n+1) \quad \text{для всех} \quad n \in \mathbb{N}.$$

12. Определить пересечение всех множеств $\mathbb{A}_n$ рациональных чисел, абсолютная величина которых меньше $1/n$, где n — натуральное число.

13. Показать, что сумма конечного или счетного числа конечных или счетных множеств есть конечное или счетное множество.

14. Показать, что всякое бесконечное множество $\mathbb{M}$ содержит счетное подмножество.

15. Показать, что множество всех пар натуральных чисел счетно.

16. Доказать или опровергнуть, что множество всех рациональных, т. е. целых и дробных чисел, счетно.

17. Показать, что среди действительных чисел нет наибольшего и наименьшего числа.

18. Определить, чему равна сумма счетного числа бесконечных множеств.

19. Установите взаимно-однозначное соответствие между множествами $\mathbb{N}$ и $\mathbb{R}$.

20. Установите взаимно-однозначное соответствие между двумя произвольными объектами вида:

$$[x, +\infty),\ [y, +\infty);$$

$$(x, +\infty),\ (y, +\infty);$$

$$(-\infty, x),\ (-\infty, y).$$

21. Установите взаимно-однозначное соответствие между интервалом $(-\pi, \pi)$ и $\mathbb{R}$.

22. Покажите, что между множествами $\mathbb{N}$ и $[0, 1]$ нельзя установить взаимно-однозначное соответствие.

23. Приведите примеры трансцендентных чисел.

24. Приведите примеры алгебраических чисел.

25. Докажите, что если $\mathbb{A}$ — счетное множество и $S \in \mathbb{N}^2$, то $\mathbb{M}^S$ является счетным множеством.

Между множествами натуральных чисел и точек сегмента [0, 1] нельзя установить взаимно-однозначное соответствие.

Любая геометрическая фигура, содержащая хоть одну линию, имеет столько же точек, сколько и отрезок.

Проблема континуума: существует ли промежуточное множество, содержащее больше элементов, чем множество натуральных чисел, и меньше, чем множество точек на прямой.

Глава 7
ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МУЛЬТИМНОЖЕСТВ

> Направляя свое внимание исключительно в одну точку, мы можем легко ограничить нашу способность схватывать вещи и никогда не достигнем той дальнозоркости, которую дает нам широкий, хотя, быть может, и менее точный взгляд.
>
> *Г. Бокль*

Понятие мультимножества, компоненты мультимножеств, определение и способы задания мультимножеств, основные характеристики мультимножеств, операции над мультимножествами

ЦЕЛИ

Освоив эту главу, студенты должны:
- иметь понятия о мультимножествах;
- знать основные характеристики мультимножеств;
- знать способы задания мультимножеств;
- уметь выполнять операции над мультимножествами.

7.1. Понятие мультимножества

Мы рассмотрели конечные множества, в которых отсутствуют повторяющиеся элементы. В кортежах возможны повторяющиеся элементы, но при этом значение каждого элемента определяется его местоположением. В задачах искусственного интеллекта начинают использоваться объекты с повторяющимися элементами. Материалы данной главы основаны на монографии А. Б. Петровского, приведенной в списке литературы.

Мультимножество — это множество с повторяющимися элементами, где один и тот же элемент может присутствовать многократно. Особенностью мультимножества является понятие кратности вхождения элемента. Элементы мультимножеств будем обозначать строчными буквами с подстрочным индексом M: $a_M, b_M, \ldots$, мультимножества — прописными буквами с подстрочным индексом M — $A_M, B_M, \ldots$

Примером мультимножеств могут служить, например, следующие совокупности элементов a, b, c, d, e, f, g, h:

$$A_M = \{a, b, a, d, e, c, a, b, h, h\},$$

$$B_M = \{d, d, e, b, b, d, e, e, h\},$$

$$C_M = \{a, a, d, a, c, a, a, e, c, c, g, g, g\}.$$

Порядок следования элементов в мультимножестве считается несущественным. Тогда приведенные мультимножества A, B, C можно переписать следующим образом:

$$A_M = \{3a, 2b, c, d, e, 2h\},$$

$$B_M = \{2b, 3d, 3e, h\},$$

$$C_M = \{5a, 3c, d, e, 3g\}.$$

Отметим, что отсутствующие элементы не указываются в записи мультимножества.

Приведем формальное определение мультимножества, данное А. Б. Петровским. *Мультимножеством* A_M, определенным на множестве $A = \{x_1, x_2, \ldots\}$, все элементы x_i которого различны, называется совокупность групп одинаковых элементов

$$A_M = \{k_1 x_1, k_2 x_2, \ldots\}, \quad x_i \in A.$$

Группу одинаковых элементов $k_i x_i$ называют *компонентой мультимножества*, элементы x_i, входящие в компоненту $k_i x_i$, — *экземплярами элементов мультимножества*. Функция k_i, принимающая числовые значения, определяет число вхождений элемента $x_i \in A$ в мультимножество A_M или «вес» элемента x_i в мультимножестве A_M. Ее также называют *функцией кратности* или *функцией числа экземпляров мультимножества* A_M. Другими словами, мультимножество — это множество, состоящее из различных групп одинаковых экземпляров элементов.

Говорят, что *элемент* x_i *принадлежит мультимножеству* A_M (обозначается $x_i \in A_M$) и *в мультимножестве* A_M *имеется ровно* k *экземпляров элемента* x_i тогда и только тогда, когда кратность элемента x_i равна $k_i x_i > 0$. Когда кратность элемента x_i равна нулю $k_i x_i = 0$, говорят, что *элемент* x_i *не содержится в мультимножестве* A_M (обозначается $x_i \notin A_M$).

Тем самым *принадлежность элемента* x_i мультимножеству A_M определяется значением функции кратности.

Если все мультимножества семейства $\Theta(A_M)=\{A1_M, A2_M, \ldots\}$ образуются из элементов одного и того же множества $G = \{x_1, x_2, \ldots\}$, то множество G называется *порождающим множеством* или *доменом* для семейства $\Theta(A_M)$. В качестве порождающего множества G может выступать любое непустое (конечное или бесконечное) множество.

Основными характеристиками мультимножества являются мощность и размерность. *Мощность мультимножества* A_M определяется как общее число экземпляров всех его элементов

$$|A_M| = \mathrm{card}\, A_M,$$

а *размерность мультимножества* A — как общее число различных элементов

$$/A_M/ = \dim A_M.$$

Размерность мультимножества не превосходит его мощности и мощности домена $/A_M/ \leqslant |A_M|$, $/A_M/ \leqslant |G|$. Мощность мультимножества $|A_M|$ в общем случае не связана с мощностью домена $|G|$. Конечные мультимножества, имеющие мощность m и состоящие из m элементов (считая повторения), называют *m-кардинальными мультимножествами*, или *m-мультимножествами*, а имеющие размерность n и состоящие из n компонент — *n-мерными мультимножествами*.

Высотой или *пиковым значением мультимножества* A_M называется максимальное значение его функции кратности k_i, а элемент x_A, для которого функция кратности k_A максимальна, — *пиком* или *пиковым элементом* мультимножества A_M.

Мультимножество удобно изображать графически в виде ступенчатой гистограммы, по оси абсцисс которой расположены элементы основного множества A или домена G, а по оси ординат отложены значения $k_i(x_i)$ функции кратности, показывающие количество экземпляров элемента x_i в мультимножестве A_M. Таким образом, каждый столбец гистограммы соответствует определенной компоненте мультимножества A_M. Ширина гистограммы равна размерности $/A_M/$ мультимножества, а высота гистограммы есть высота мультимножества A_M. Мощность мультимножества $|A_M|$ будет численно равна площади фигуры, ограниченной гистограммой.

Для мультимножеств справедливы теоретико-множественные понятия, введенные для множеств.

Рассмотрим возможные способы сопоставления мультимножеств, обусловленные особенностями их различных характеристик. Мультимножества A_M и B_M называются *равными* ($A_M = B_M$), если $k_i(x_i) = k_j(x_j)$ для всех элементов x_i, $x_j \in G$, $k_i(x_i) \in A_M$, $k_j(x_j) \in B_M$. В противном случае эти мультимножества неравны. Для равных мультимножеств имеем $|A| = |B|$, $/A/ = /B/$.

Мультимножества A и B называют *равномощными*, если $|A| = |B|$. Например, мультимножества $A_M = \{3a, 2b\}$, $B_M = \{2a, 3b\}$ являются равномощными, так как мощность мультимножества $A_M = 5$ и мощность мультимножества $B_M = 5$, а мультимножества $C_M = \{3d, e\}$, $D_M = \{e, 2d\}$ — неравными, так как $4 \neq 3$.

Мультимножества A и B называют *равноразмерными*, если $/A/ = /B/$. Например, мультимножества $A_M = \{3a, 2b, c\}$, $B_M = \{2a, 2b, 2c\}$ являются равноразмерными мощными, так как число различных экземпляров мультимножества $/A_M/ = /B_M/ = 3$, а мультимножества $C_M = \{3d, e\}$, $D_M = \{e, 2d\}$ — неравными, так как $4 \neq 3$.

Мультимножества A_M и B_M называют равными, если они равномощны и равноразмерны.

Говорят, что мультимножество B_M *содержится* или *включено* в мультимножество A_M ($B_M \subseteq A_M$), если $k_j x_j \leqslant k_i x_i$, для каждого элемента x_i, $_j \in G$, $k_i x_i \in A_M$, $k_j x_j \in B_M$. Мультимножество B_M называется тогда *подмультимножеством* мультимножества A_M, а мультимножество A_M — *надмультимножеством* мультимножества B_M. В этом случае $|B| \leqslant |A|$, $/B/ \leqslant /A/$. Как и в случае обычных множеств, одновременное выполнение условий $B_M \subseteq A_M$ и $A_M \subseteq B_M$ влечет равенство мультимножеств $A_M = B_M$.

Замечание. Включение мультимножества обладает свойствами рефлексивности ($A_M \subseteq A_M$) и транзитивности ($A_M \subseteq B_M, B_M \subseteq C_M \Rightarrow A_M \subseteq C_M$), а значит, является отношением предпорядка.

7.2. Операции над мультимножествами

Над мультимножествами определены следующие основные операции: объединение, пересечение, арифметическое сложение, арифметическое вычитание, дополнение, симметрическая разность, умножение на число, арифметическое умножение и возведение в арифметическую степень, прямое произведение и возведение в прямую степень.

Объединением мультимножеств A_M и B_M называется мультимножество, состоящее из всех элементов, которые присутствуют хотя бы в одном из мультимножеств, и кратность каждого элемента равна максимальной кратности соответствующих элементов в объединяемых мультимножествах:

$$C_M = A_M \cup B_M = \{\max(k_i x_i, k_j x_j)\}, \quad k_i x_i \in A_M, \quad k_j x_j \in B_M.$$

Другими словами, производится попарное сравнивание каждого экземпляра мультимножеств и в каждой паре выбирается экземпляр с наибольшим значением функции кратности.

Например, заданы мультимножества $A_M = \{3a, 2c, d\}$, $B_M = \{3b, 2d, 2e\}$. Тогда их объединение $C_M = \{3a, 3b, 2c, 2d, 2e\}$.

Отметим, что можно выполнять операцию объединения произвольного числа мультимножеств.

Объединением произвольного числа мультимножеств Ai_M называется мультимножество

$$A1_M \cup A2_M \cup \ldots \cup An_M = \{\max(k_i x_i, k_j x_j, \ldots, k_n x_n)\},$$
$$k_i x_i \in A1_M, \quad k_j x_j \in A2_M, \ldots, k_n x_n \in An_M.$$

Например, заданы мультимножества $A1_M = \{2a, 3c, d\}$, $A2_M = \{2b, 2d, e\}$, $A3_M = \{a, 3b, 2c, 3d, 4f\}$. Тогда их объединение $C_M = \{2a, 3b, 3c, 3d, e, 4f\}$.

Пересечением мультимножеств A_M и B_M называется мультимножество, состоящее из всех элементов, которые одновременно присутствуют в каждом из мультимножеств, и кратность каждого элемента равна минимальной кратности соответствующих элементов в пересекаемых мультимножествах:

$$C_M = A_M \cap B_M = \{\min(k_i x_i, k_j x_j)\}, \quad k_i x_i \in A_M, \quad k_j x_j \in B_M.$$

То есть производится попарное сравнивание каждого экземпляра мультимножеств и в каждой паре выбирается экземпляр с наименьшим значением функции кратности.

Например, заданы мультимножества $A_M = \{3a, 2b, 4c\}$, $B_M = \{3b, 2c, 2d\}$. Тогда их пересечение $C_M = \{2b, 2c\}$.

Пересечением произвольного числа мультимножеств Ai_M называется мультимножество

$$A1_M \cap A2_M \cap \ldots \cap An_M = \{\min(k_i x_i, k_j x_j, \ldots, k_n x_n)\},$$
$$k_i x_i \in A1_M, \quad k_j x_j \in A2_M, \ldots, k_n x_n \in An_M.$$

Например, заданы мультимножества $A1_M = \{2a, 3c, d\}$, $A2_M = \{3a, 2b, 2d, e\}$, $A3_M = \{4a, 3b, 2c, 3d, f\}$. Тогда их пересечение $C_M = \{2a, d\}$.

Арифметической суммой мультимножеств A_M и B_M называется мультимножество, состоящее из всех элементов, которые присутствуют хотя бы в одном из мультимножеств, и кратность каждого элемента равна сумме кратностей соответствующих элементов в складываемых мультимножествах:

$$C_M = A_M + B_M = \{kx \,|\, kx = k_i x_i + k_j x_j\},$$
$$k_i x_i \in A_M, \quad k_j x_j \in B_M.$$

Например, заданы мультимножества $A_M = \{3a, 2b, 4c\}$, $B_M = \{3b, 2c, 2d\}$. Тогда их арифметическая сумма $C_M = \{3a, 5b, 6c, 2d\}$.

Суммой произвольного числа мультимножеств Ai_M называется мультимножество

$$A1_M + A2_M + \ldots + An_M = \{kx \,|\, kx = k_i x_i + k_j x_j + \ldots + k_n x_n\},$$
$$k_i x_i \in A1_M, \quad k_j x_j \in A2_M, \ldots, k_n x_n \in An_M.$$

Например, заданы мультимножества $A1_M = \{2a, c, d\}$, $A2_M = \{2b, 2d, e\}$, $A3_M = \{2a, 3b, 2c, 3d, 2f\}$. Тогда их сумма $C_M = \{4a, 5b, 3c, 6d, e, 2f\}$.

Арифметической разностью мультимножеств A_M и B_M называется мультимножество, состоящее из тех элементов мультимножества A_M, кратность которых превышает кратность соответствующих элементов в мультимножестве B_M. Кратность каждого элемента результирующего множества равна разности кратностей соответствующих элементов в вычитаемых мультимножествах:

$$C_M = A_M - B_M = \left\{ kx \,\middle|\, \begin{array}{l} kx = k_i x_i - k_j x_j, \text{ если } k_i > k_j \\ 0, \text{ в противном случае} \end{array} \right\},$$
$$k_i x_i \in A_M, \quad k_j x_j \in B_M.$$

Например, заданы мультимножества $A_M = \{3a, 2b, c, 4d\}$, $B_M = \{2a, 3b, 2c, 2d\}$. Тогда их арифметическая разность $C_M = \{a, 2d\}$.

Операция разности мультимножеств определена только для двух мультимножеств. При этом если $A_M \subseteq B_M$, то $A_M - B_M = \varnothing$. Справедливо и обратное утверждение: если $A_M - B_M = \varnothing$, то $A_M \subseteq B_M$.

Симметрической разностью мультимножеств A_M и B_M называется мультимножество, состоящее из тех элементов мультимножеств A_M и B_M, кратности которых различны. Кратность каждого элемента результирующего мультимножества равна модулю разности кратностей соответствующих элементов в вычитаемых мультимножествах:

$$C_M = A_M \Delta B_M = \{kx \mid kx = |k_i x_i - k_j x_j|\},$$
$$k_i x_i \in A_M, \quad k_j x_j \in B_M.$$

Например, заданы мультимножества $A_M = \{a, 2b, 3c, 4d\}$, $B_M = \{3a, 2b, 3c, 2d\}$. Тогда их симметрическая разность $C_M = \{2a, 2d\}$.

Симметрическая разность мультимножеств применима только к двум мультимножествам.

Дополнением мультимножества A_M до универсума Z называется мультимножество, состоящее из элементов, кратность которых равна разности кратностей соответствующих элементов в универсуме Z и дополняемом мультимножестве A_M. Под *универсумом* в данном случае понимается некоторое мультимножество Z, такое, что все остальные мультимножества являются подмультимножествами данного множества Z.

$$\overline{A}_M = Z - A_M = \{k_{\overline{A}} x | k_{\overline{A}} x = k_z x - k_A x, \ \forall x \in Z\}.$$

Из определений пустого мультимножества и дополнения мультимножества следует, что пустое мультимножество $\varnothing$ и универсум Z взаимно дополняют друг друга: $\overline{\varnothing} = Z$, $\overline{Z} = \varnothing$.

Арифметическим произведением мультимножеств A_M и B_M называется мультимножество, состоящее из элементов, которые одновременно присутствуют в каждом из мультимножеств, и их кратность равна произведению кратностей соответствующих элементов в перемножаемых мультимножествах:

$$C_M = A_M \cdot B_M = \{kx \mid kx = k_i x_i \cdot k_j x_j\},$$
$$k_i x_i \in A_M, \quad k_j x_j \in B_M.$$

Например, заданы мультимножества $A_M = \{a, 2b, 3d, 4f\}$, $B_M = \{2b, 3c, d, 4e, 3f\}$. Тогда их арифметическое произведение $C_M = \{4b, 3d, 12f\}$.

Арифметическим произведением произвольного числа мультимножеств Ai_M называется мультимножество

$$A1_M \cdot A2_M \cdot \ldots \cdot An_M = \{kx \mid kx = k_i x_i \cdot k_j x_j \cdot \ldots \cdot k_n x_n\},$$
$$k_i x_i \in A1_M, \quad k_j x_j \in A2_M, \ldots, k_n x_n \in An_M.$$

Например, заданы мультимножества $A1_M = \{3a, c, 4d\}$, $A2_M = \{a, 2b, d, e\}$, $A3_M = \{2a, 3b, 2c, 3d, 2f\}$. Тогда их арифметическое произведение $C_M = \{6a, 12d\}$.

Арифметической n-й степенью мультимножества A_M называется арифметическое произведение n одинаковых мультимножеств A_M:

$$A_M^n = \{k_{A_M^n} x \mid k_{A_M^n} x = (k_{A_M} x)^n\}.$$

Произведением мультимножества A_M *на целое число* h или *репродукцией* мультимножества A_M называется мультимножество, состоящее из элементов мультимножества A_M, причем кратность каждого элемента увеличена в h раз:

$$h \cdot A_M = \{h \cdot kx\}.$$

Репродукцию мультимножества можно также представить как сумму h одинаковых мультимножеств A_M: $h \cdot A_M = A_M + \ldots + A_M$ (h раз).

Прямым произведением мультимножеств A_M и B_M называется мультимножество, состоящее из всех упорядоченных пар элементов $\langle x_i, x_j \rangle$ таких, что первый элемент каждой пары является элементом первого сомножителя $x_i \in A_M$, второй элемент пары — элементом второго сомножителя $x_j \in B_M$ и кратность каждой пары $\langle x_i, x_j \rangle$ равна произведению кратностей элементов x_i и x_j в перемножаемых мультимножествах:

$$C_M = A_M \times B_M = \\ = \{k_{A\times B}\langle x_i, x_j \rangle \mid k_{A\times B} = k_i x_i \cdot k_j x_j,\ x_i \in A_M,\ x_j \in B_M\}.$$

Например, заданы мультимножества $A_M = \{a, 2b, 3c\}$, $B_M = \{d, 4e\}$. Тогда их прямое произведение $C_M = \{\langle a, d\rangle; 4\langle a, e\rangle; 2\langle b, d\rangle, 8\langle b, e\rangle, 3\langle c, d\rangle, 12\langle c, e\rangle\}$.

Прямым произведением конечного числа мультимножеств Ai_M называется мультимножество, состоящее из всех n-элементных кортежей $\langle x_1, \ldots, x_n\rangle$ таких, что i-й элемент кортежа является элементом i-го сомножителя $x_i \in A_i$ и кратность каждого кортежа равна произведению кратностей элементов в перемножаемых мультимножествах:

$$A1_M \times A2_M \times \ldots \times An_M = \\ = \{k\langle x_1, \ldots, x_n\rangle \mid k\langle x_1, \ldots, x_n\rangle = k_i x_i \cdot k_j x_j \ldots k_n x_n\},$$

$$k_i x_i \in A1_M, \quad k_j x_j \in A2_M, \ldots, k_n x_n \in An_M.$$

Например, заданы мультимножества $A1_M = \{3a, c\}$, $A2_M = \{2b, d\}$, $A3_M = \{3e, 2f\}$. Тогда их прямое произведение $C_M = \{18\langle a, b, e\rangle; 12\langle a, b, f\rangle; 9\langle a, d, e\rangle, 6\langle a, d, f\rangle, 6\langle c, b, e\rangle, 4\langle c, b, f\rangle, 3\langle c, d, e\rangle, 2\langle c, d, f\rangle\}$.

Прямой n-й степенью мультимножества A_M называется прямое произведение n одинаковых мультимножеств A_M:

$$(\times A_M^n)=\left\{k_{(\times A_M)^n}\cdot\langle x_1,\ldots,x_n\rangle | k_{(\times A_M)^n}=\prod_{i=1}^{n} k_{A_M}(x_i), x_i\in A_M\right\}.$$

По аналогии с множествами для мультимножеств также можно сформулировать некоторые правила выполнения операций:

$$\overline{A_M\cup B_M}=\overline{A_M}\cap\overline{B_M};$$

$$\overline{A_M\cap B_M}=\overline{A_M}\cup\overline{B_M};$$

$$\overline{A_M+B_M}=\overline{A_M}-B_M=\overline{B_M}-A_M;$$

$$\overline{A_M-B_M}=\overline{A_M}+B_M;$$

$$\overline{A_M}-\overline{B_M}=B_M-A_M;$$

$$A_M+B_M=(A_M\cup B_M)+(A_M\cap B_M);$$

$$A_M\Delta B_M=(A_M\cup B_M)-(A_M\cap B_M)=\overline{A_M}\Delta\overline{B_M};$$

$$(A_M-B_M)\cap(B_M-A_M)=\varnothing;$$

$$\begin{aligned}A_M\cup B_M&=(A_M+B_M)-(A_M\cap B_M)=\\&=(A_M\cap B_M)+(A_M\Delta B_M)=\\&=A_M+(B_M-A_M)=B_M+(A_M-B_M);\end{aligned}$$

$$\begin{aligned}A_M\cap B_M&=(A_M+B_M)-(A_M\cup B_M)=\\&=(A_M\cup B_M)-(A_M\Delta B_M)=\\&=A_M-(A_M-B_M)=B_M-(B_M-A_M);\end{aligned}$$

$$\begin{aligned}A_M-B_M&=(A_M\cup B_M)-B_M=A_M-(A_M\cap B_M)=\\&=(A_M\cup B_M)\Delta B_M=A_M\Delta(A_M\cap B_M);\end{aligned}$$

$$\begin{aligned}A_M\Delta B_M&=(A_M\cup B_M)-(A_M\cap B_M)=\\&=(A_M-B_M)+(B_M-A_M)=\\&=(A_M-B_M)\cup(B_M-A_M)=(A_M+B_M-2(A_M\cap B_M);\end{aligned}$$

$$A_M+B_M=(A_M\cup B_M)+(A_M\cap B_M)=(A_M\Delta B_M)+2(A_M\cap B_M).$$

Примеры решения задач

Пример 7.1. Определить мощность и размерность мультимножества A_M:

$$A_M = \{3a, 7b, 9c, 2d, 4e, 5f\}.$$

Решение. Мощность мультимножества A — это общее число экземпляров всех его элементов. Тогда $|A| = 30$.

Размерность мультимножества — это общее число различных элементов. Тогда $/A/ = 6$.

Пример 7.2. Представьте мультимножество A_M из первого примера в виде гистограммы.

Решение. На рис. 7.1 представлено мультимножество A_M в виде гистограммы. По оси ординат отмечены элементы, а по оси абсцисс — кратность их вхождения в мультимножество.

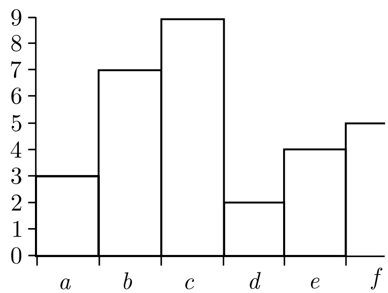


Рис. 7.1. Представление мультимножества A_M в виде гистограммы

Пример 7.3. Постройте примеры равных, неравных, равномощных и равноразмерных мультимножеств.

Решение. Согласно приведенным выше определениям мультимножества $A_M = \{3a, 3c, 2d\}$ и $B_M = \{3c, 2d, 3a\}$ являются равными.

Мультимножества $A_M = \{3a, 2b, c\}$ и $B_M = \{a, 3b, 2c\}$ являются неравными.

Мультимножества $A_M = \{3a, 3b, 2c\}$ и $B_M = \{4a, 3b, c\}$ являются равномощными, так как сумма экземпляров элементов этих мультимножеств совпадает и равна 8. Данные мультимножества также равноразмерны, так как у них одинаковое общее число различных элементов — 3.

Пример 7.4. Примените вышеописанные операции к мультимножествам $A_M = \{2a, 6b, 3c, d\}$ и $B_M = \{3b, 5c, 2d, 4e\}$.

Решение. Согласно приведенным выше определениям результат запишется следующим образом:

$$A_M \cup B_M = \{2a, 6b, 5c, 2d, 4e\};$$

$$A_M \cap B_M = \{3b, 3c, d\};$$

$$A_M + B_M = \{2a, 9b, 8c, 3d, 4e\};$$

$$A_M - B_M = \{2a, 3b\};$$

$$A_M \Delta B_M = \{2a, 3b, 2c, d, 4e\};$$

$$A_M \cdot B_M = \{18b, 15c, 2d\};$$

$$A_M \times B_M = \{2a, 6b, 5c, 2d, 4e\}.$$

Пример 7.5. Найдите прямое произведение мультимножеств $A_M = \{2a, d\}$ и $B_M = \{3b, 5c\}$.

Решение. Согласно приведенным выше определениям результат запишется следующим образом:

$$A_M \times B_M = \{6\langle a, b\rangle; 10\langle a, c\rangle; 3\langle d, b\rangle, 5\langle d, c\rangle\}.$$

Контрольные вопросы

1. Дайте понятие мультимножества.
2. Приведите примеры мультимножеств.
3. Дайте формальное определение мультимножества.
4. В чем сходство и различие множества и мультимножества?
5. Что такое мощность и размерность мультимножества? Приведите примеры.
6. Приведите способы сопоставления мультимножеств.
7. Какие мультимножества являются равными, неравными, равномощными, равноразмерными?
8. Опишите операцию объединения мультимножеств.
9. Приведите операцию пересечения мультимножеств.
10. Опишите операцию арифметической суммы и разности мультимножеств.
11. В чем заключается операция прямого произведения мультимножеств?
12. Приведите основные свойства операций над мультимножествами.

Задания для самостоятельной работы

1. Определите мощность и размерность следующих мультимножеств: $A_M = \{3a, 5b, c, 8d, 7e, 9f\}$, $B_M = \{a, 2b, 9c, 4d, f\}$, $C_M = \{2a, 4b, 6c, 3d, 5e, 7f, 9g\}$.

2. Представьте мультимножества из задания 1 в виде гистограммы.

3. Определите, являются ли следующие мультимножества равными или неравными:

$$A_M = \{3a, 5b, 9c\};$$
$$B_M = \{5a, 3b, 9c\}.$$

4. Определите, являются ли мультимножества из заданий 1, 3 равномощными или равноразмерными.

5. Выполните операцию объединения мультимножеств из заданий 1, 3.

6. Выполните операцию пересечения мультимножеств из заданий 1, 3.

7. Реализуйте операцию арифметической суммы и разности мультимножеств из задания 3.

8. Реализуйте операцию симметрической разности и дополнения следующих мультимножеств:

$$A_M = \{10a, 8b, 6c, 3d, 5e, 7f, 9g\},$$
$$B_M = \{2a, 4b, 6c, 9e, 7f, 5g\}.$$

9. Определите прямое произведение следующих мультимножеств:

$$A_M = \{3a, 2b\};$$
$$B_M = \{2c, 9d\}.$$

10. Покажите справедливость следующих выражений:

$$\overline{A_M \cup B_M} = \overline{A_M} \cap \overline{B_M};$$
$$\overline{A_M \cap B_M} = \overline{A_M} \cup \overline{B_M};$$
$$\overline{A_M + B_M} = \overline{A_M} - B_M = \overline{B_M} - A_M;$$
$$\overline{A_M - B_M} = \overline{A_M} + B_M;$$
$$\overline{A_M} - \overline{B_M} = B_M - A_M$$

для мультимножеств из задания 8.

11. Покажите справедливость следующих выражений:

$$(A_M - B_M) \cap (B_M - A_M) = \varnothing;$$
$$A_M = (A_M - B_M) + (A_M \cap B_M);$$
$$B_M = (B_M - A_M) + (A_M \cap B_M);$$
$$A_M \cup B_M = B_M + (A_M - B_M);$$
$$A_M \cap B_M = B_M - (B_M - A_M)$$

для мультимножеств из задания 8.

Возможность многократного вхождения элементов в мультимножество создает новое качество и позволяет расширить класс описываемых, анализируемых и синтезируемых математических объектов.

Глава 8
НЕЧЕТКИЕ МНОЖЕСТВА

Часто неясность происходит столько же от многословия, сколько и от излишней краткости.

Ж. Даламбер

Понятие нечеткого высказывания, степень истинности нечеткого высказывания, функция принадлежности, операции над нечеткими высказываниями, определение и способы задания нечетких множеств, операции над нечеткими множествами, нечеткие отношения и соответствия

ЦЕЛИ

Освоив эту главу, студенты должны:

- иметь понятия о нечетких множествах;
- уметь строить функцию принадлежности;
- определять степень истинности составных высказываний;
- уметь выполнять операции над нечеткими множествами и высказываниями;
- иметь понятие о нечетких отношениях и соответствиях и операциях над ними.

8.1. Нечеткие высказывания

В множествах можно однозначно указать, принадлежит или не принадлежит рассматриваемый элемент данному множеству. Следовательно, рассматриваемые выше множества имеют четкие границы, определяющие принадлежность или не принадлежность элемента множеству. В интеллектуальных системах часто нельзя провести четкую грань между принадлежностью и не принадлежностью элементов множеству. В этой связи возникло предположение описывать множества объектов с расплывчатыми границами принадлежности. Например, как определить понятие красоты, любви, удачи, счастья и других расплывчатых понятий. Л. Заде является основателем теории нечетких множеств. Он предложил ввести числовую меру описания расплывчатых понятий. Это дает возможность использовать существующий математический аппарат для описания, преобразования и исследования расплывчатых понятий.

Пусть задано произвольное непустое множество X:

$$X = \{x_1, x_2, \ldots, x_n\}.$$

Множество $\widetilde{A}$ называется *нечетким* или *расплывчатым множеством* в X и обозначается $\widetilde{A} = \{\langle \mu_{\widetilde{A}}, x \rangle\}$, $x \in X$.

В нечетком множестве каждый элемент есть кортеж длины два, первая компонента которого обозначается $\mu_{\widetilde{A}}$ и называется *функцией принадлежности*. Она задается на сегменте $[0, 1]$. Вторая компонента кортежа — элемент множества X.

При задании множества $\widetilde{A}$ каждому элементу x присваивается число $0 \leqslant \mu_{\widetilde{A}} \leqslant 1$. Это число и определяет степень принадлежности элемента x множеству $\widetilde{A}$.

Например, пусть задано некоторое множество $X = \{x_1, x_2, x_3, x_4\}$. Тогда одним из возможных нечетких множеств на множестве X будет множество $\widetilde{A} = \{\langle 0{,}1, x_1 \rangle, \langle 1, x_2 \rangle, \langle 0, x_3 \rangle, \langle 0{,}8, x_4 \rangle\}$.

Носителем нечеткого множества $\widetilde{A}$ называется подмножество $A \subseteq X$, содержащее те элементы из X, для которых $\mu_{\widetilde{A}} > 0$. Функция принадлежности для каждого нечеткого множества определяется субъективно. В настоящее время для нахождения функции принадлежности используют экспертные системы.

Рассмотрим нечеткие высказывания. *Нечетким высказыванием* называется предложение, относительно которого можно судить о степени его истинности или ложности.

Например, нечеткое высказывание: «2 — маленькое число». Данное высказывание является субъективным. Если четкое множество X, на котором выбрано число 2, является относительно большим, например, $|X| = 100$. Тогда нечеткое высказывание: «2 — маленькое число» является истинным. Если, например, $|X| = 3$, тогда рассматриваемое высказывание является ложным.

Нечеткое высказывание, равное 0,5, называется *индифферентным*, так как оно истинно в той же степени, что и ложно. Следовательно, истинность нечеткого высказывания является субъективной характеристикой. Нечеткие высказывания бывают простые и составные. Составные образуются из простых с помощью логических операций: отрицания, дизъюнкции, конъюнкции, импликации и эквивалентности.

Отрицанием высказывания $\widetilde{A}$ называется нечеткое высказывание не $\widetilde{A}$, степень истинности которого определяется по формуле
$$\rceil\widetilde{A} = 1 - \widetilde{A}. \tag{8.1}$$

Символ $\rceil$ в данной главе используется для обозначения инверсии.

Конъюнкцией высказываний $\widetilde{A}$, $\widetilde{B}$ называется нечеткое высказывание, степень истинности которого определяется по формуле
$$\widetilde{A} \wedge \widetilde{B} = \min(\widetilde{A} \wedge \widetilde{B}). \tag{8.2}$$

Дизъюнкцией высказываний называется нечеткое высказывание, степень которого определяется по формуле
$$\widetilde{A} \vee \widetilde{B} = \max(\widetilde{A} \vee \widetilde{B}). \tag{8.3}$$

Импликацией высказываний называется нечеткое высказывание:
$$\widetilde{A} \to \widetilde{B} = \max(1 - \widetilde{A}, \widetilde{B}). \tag{8.4}$$

Из последнего выражения вытекает, что степень истинности не меньше, чем степень неистинности ее посылки или степень истинности ее следствия.

Эквивалентностью высказываний называется новое составное нечеткое высказывание, обозначаемое
$$\widetilde{A} \leftrightarrow \widetilde{B} = \min\big(\max(1 - \widetilde{A}, \widetilde{B}), \max(1 - \widetilde{B}, \widetilde{A})\big). \tag{8.5}$$

Если степени истинности нечетких высказываний $\widetilde{A}$ и $\widetilde{B}$ одинаковы, то степень истинности эквивалентности лежит в сегменте $[0{,}5, 1]$.

Пример. Пусть $\widetilde{A} = 0{,}7$, $\widetilde{B} = 0{,}4$, $\widetilde{C} = 0{,}9$ и требуется найти степень истинности следующего составного нечеткого высказывания:
$$\widetilde{D} = (\widetilde{A} \wedge \rceil\widetilde{B} \vee \rceil\widetilde{A} \wedge \widetilde{B}) \to \rceil(\widetilde{A} \wedge \widetilde{C}).$$

$$\begin{aligned}
\widetilde{D} &= \max(1 - (\widetilde{A} \wedge \rceil\widetilde{B} \vee \rceil\widetilde{A} \wedge \widetilde{B}), \rceil(\widetilde{A} \wedge \widetilde{C})) = \\
&= \max((1 - \max(\widetilde{A} \wedge \rceil\widetilde{B}, \rceil\widetilde{A} \wedge \widetilde{B})), 1 - \widetilde{A} \wedge \widetilde{C}) = \\
&= \max((1 - \max(\min(\widetilde{A}, 1 - \widetilde{B}), \min(1 - \widetilde{A}, \widetilde{B}))), 1 - \min(\widetilde{A}, \widetilde{C})) = \\
&= \max((1 - \max(\min(0{,}7, 0{,}6), \min(0{,}3, 0{,}4))), 1 - \min(0{,}7, 0{,}9)) = \\
&= \max(1 - \max(0{,}6, 0{,}3), 0{,}3) = \max(0{,}4, 0{,}3) = 0{,}4.
\end{aligned}$$

При вычислении степени истинности данного составного высказывания использовались формулы (8.1)–(8.5).

Введем понятие *степени включения нечетких множеств*. Пусть задано множество X и на нем нечеткие множества $\widetilde{A}, \widetilde{B}$. Тогда степень включения нечеткого множества $\widetilde{A}$ в нечеткое множество $\widetilde{B}$ обозначается $\nu(\widetilde{A}, \widetilde{B})$ и определяется по формуле

$$\nu(\widetilde{A}, \widetilde{B}) = \wedge(\mu \sim A(x) \to \mu \sim B(x)), \quad x \in X.$$

Если $\nu(\widetilde{A}, \widetilde{B}) \geqslant 0{,}5$, то считается, что нечеткое множество $\widetilde{A}$ нестрого включается в $\widetilde{B}$ ($\widetilde{A} \subseteq \widetilde{B}$). Если $\nu(\widetilde{A}, \widetilde{B}) < 0{,}5$, то $\widetilde{A}$ не включается в $\widetilde{B}$.

Например. Пусть на множестве $X = \{x_1, x_2, x_3, x_4, x_5\}$ заданы два нечетких множества $\widetilde{A} = \{\langle 0{,}3, x_2\rangle, \langle 0{,}7, x_3\rangle, \langle 0{,}9, x_4\rangle\}$, $\widetilde{B} = \{\langle 0{,}1, x_1\rangle, \langle 0{,}8, x_2\rangle, \langle 0{,}3, x_3\rangle, \langle 0{,}9, x_5\rangle\}$. Тогда степень включения нечеткого множества $\widetilde{A}$ в $\widetilde{B}$ определяется следующим образом:

$$\nu(\widetilde{A}, \widetilde{B}) = (0 \to 0{,}1) \wedge (0{,}3 \to 0{,}8) \wedge (0{,}7 \to 0{,}3) \wedge \\ \wedge (0{,}9 \to 0) \wedge (0 \to 0{,}9) = 1 \wedge 0{,}7 \wedge 0{,}7 \wedge 1 \wedge 1 = 0{,}7.$$

Следовательно, $\widetilde{A} \subseteq \widetilde{B}$.

Отметим, что степень включения одного нечеткого множества в другое может быть определена для любых двух нечетких множеств, при этом она может принимать любое значение от 0 до 1.

8.2. Операции над нечеткими множествами

Рассмотрим основные операции *над нечеткими множествами*.

1. Дано множество X и на нем заданы нечеткие множества $\widetilde{A} = \{\langle \mu \sim A(x), x\rangle, x \in X\}$, $\widetilde{B} = \{\langle \mu \sim B(x), x\rangle, x \in X\}$.

Объединением нечетких множеств называется выражение вида

$$\widetilde{A} \cup \widetilde{B} = \{\langle \mu \sim A \cup \sim B(x), x\rangle, x \in X\}.$$

Значение $\mu \sim A \cup \sim B$ находится согласно следующей формуле:

$$\mu \sim A \cup \sim B = \mu \sim A \vee \mu \sim B.$$

А дизъюнкция находится по формуле (8.3).

Пример. Пусть задано множество $X = \{x_1, x_2, x_3, x_4\}$ и нечеткие множества $\widetilde{A} = \{\langle 0{,}1, x_1\rangle, \langle 1, x_3\rangle\}$, $\widetilde{B} = \{\langle 0{,}8, x_1\rangle, \langle 1, x_2\rangle,$

$\langle 0{,}3, x_3\rangle, \langle 0{,}7, x_4\rangle\}$. Тогда их объединение запишется $\widetilde{A} \cup \widetilde{B} = \{\langle 0{,}8, x_1\rangle, \langle 1, x_2\rangle, \langle 1, x_3\rangle, \langle 0{,}7, x_4\rangle\}$.

2. $\widetilde{A} \cap \widetilde{B}$ — *пересечение* нечетких множеств, если

$$\widetilde{A} \cap \widetilde{B} = \{\langle \mu \sim A \cap \sim B(x), x\rangle, x \in X\},$$

$$\mu \sim A \cap \sim B = \mu \sim A \wedge \mu \sim B.$$

Причем выражение для конъюнкции вычисляется по формуле (8.2).

Пример. Пусть задано множество $X = \{x_1, x_2, x_3, x_4\}$ и нечеткие множества $\widetilde{A} = \{\langle 0{,}1, x_1\rangle, \langle 1, x_3\rangle\}$, $\widetilde{B} = \{\langle 0{,}8, x_1\rangle, \langle 1, x_2\rangle, \langle 0{,}3, x_3\rangle, \langle 0{,}7, x_4\rangle\}$. Тогда их пересечение запишется $\widetilde{A} \cap \widetilde{B} = \{\langle 0{,}1, x_1\rangle, \langle 0, x_2\rangle, \langle 0{,}3, x_3\rangle, \langle 0, x_4\rangle\}$.

3. *Разность.* $\widetilde{A} \backslash \widetilde{B}$ называется разностью нечетких множеств, если

$$\widetilde{A} \backslash \widetilde{B} = \{\langle \mu \sim A \backslash \sim B(x), x\rangle, x \in X\},$$

$$\mu \sim A \backslash \sim B = \mu \sim A \wedge \mu \sim B.$$

Согласно формулам (8.1) и (8.2) определяется $\mu \sim A \backslash \sim B$.

Пример. Пусть задано множество $X = \{x_1, x_2, x_3, x_4\}$ и нечеткие множества $\widetilde{A} = \{\langle 0{,}1, x_1\rangle, \langle 1, x_3\rangle\}$, $\widetilde{B} = \{\langle 0{,}8, x_1\rangle, \langle 1, x_2\rangle, \langle 0{,}3, x_3\rangle, \langle 0{,}7, x_4\rangle\}$. Тогда их разность запишется $\widetilde{A} \backslash \widetilde{B} = \{\langle 0{,}1, x_1\rangle, \langle 0, x_2\rangle, \langle 0{,}7, x_3\rangle, \langle 0, x_4\rangle\}$.

Симметрической разностью нечетких множеств $\widetilde{A}$ и $\widetilde{B}$ называется выражение вида

$$\widetilde{A} \,\Theta\, \widetilde{B} = \{\langle \mu \sim A \Theta \sim B(x), x\rangle, x \in X\},$$

где Θ — знак симметрической разности. Выражение для степени принадлежности определяется по формуле

$$\mu \sim A \Theta \sim B = \mu \sim A \backslash \sim B \vee \rceil \mu \sim B \backslash \sim A.$$

Пример. Пусть задано множество $X = \{x_1, x_2, x_3, x_4\}$ и нечеткие множества $\widetilde{A}=\{\langle 0{,}1, x_1\rangle, \langle 0{,}7, x_2\rangle\}$, $\widetilde{B}=\{\langle 0{,}4, x_1\rangle, \langle 0{,}3, x_2\rangle\}$. Тогда степень принадлежности для описанных выше операций определится

а) $\mu \sim A \backslash \sim B(x) = \mu \sim A(x) \wedge \rceil \mu \sim B(x)$,

$\mu \sim A \backslash \sim B(x) = \{\langle 0{,}1, x_1\rangle, \langle 0{,}7, x_2\rangle\}$;

б) $\mu \sim B \backslash \sim A(x) = \mu \sim B(x) \wedge \rceil \mu \sim A(x)$,

$\rceil \mu \sim A(x) = \{\langle 0{,}9, x_1\rangle, \langle 0{,}3, x_2\rangle\}$,

$\mu \sim B \backslash \sim A(x) = \{\langle 0{,}4, x_1\rangle, \langle 0{,}3, x_2\rangle\}$;

в) $\mu \sim A \,\Theta\, \sim B(x) = \{\langle 0{,}4, x_1\rangle, \langle 0{,}7, x_2\rangle\}$.

Пример. Пусть $\widetilde{A}$ — нечеткое множество высоких людей, а $\widetilde{B}$ — нечеткое множество красивых людей. Тогда $\widetilde{A} \cup \widetilde{B}$ — это нечеткое множество или высоких, или красивых людей, или обоих вместе.

$\widetilde{A} \cap \widetilde{B}$ — это нечеткое множество одновременно высоких и красивых людей.

$\rceil \widetilde{A}$ — это нечеткое множество невысоких людей.

$\widetilde{A} \backslash \widetilde{B}$ — это нечеткое множество одновременно высоких и некрасивых людей.

$\widetilde{A} \Theta \widetilde{B}$ — это нечеткое множество одновременно высоких и некрасивых людей или одновременно красивых и невысоких.

8.3. Нечеткие отношения и соответствия

Рассмотрим понятия *нечеткого соответствия* и *отношения*.

Нечетким соответствием называют тройку множеств $\sim \Gamma = \langle \sim F, X, Y \rangle$, причем X и Y — это конечные четкие множества, а $\sim F$ — нечеткий график соответствия. *Нечетким графиком* называется множество, состоящее из нечетких упорядоченных пар (кортежей). Например, на множествах $X = \{1, 2\}$, $Y = \{a, b\}$ может быть построен нечеткий график вида $\sim F = \{\langle 0{,}2, \langle a, 1\rangle\rangle, \langle 0{,}9, \langle a, 2\rangle\rangle, \langle 0{,}7, \langle b, 2\rangle\rangle\}$.

Нечеткие соответствия можно задавать в теоретико-множественном, матричном или графическом виде. В первом случае необходимо перечислить элементы X и Y и задать разбиение множества $\sim F$ в $X \times Y$.

Представим вышеописанное соответствие в графическом виде (рис. 8.1).

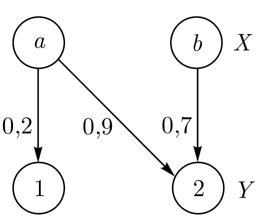


Рис. 8.1. Пример графического задания нечеткого соответствия

Зададим теперь это же соответствие в матричном виде. Строки — элементы X, а столбцы — элементы Y:

$X\backslash Y$	1	2
a	0,2	0,9
b	0	0,7

Все три способа равнозначны и задают одно и то же нечеткое соответствие.

В нечетких соответствиях справедливы операции инверсии и композиции.

Инверсией нечеткого соответствия $\sim \Gamma = \langle \sim F, X, Y\rangle$ называется соответствие $\sim \Gamma^{-1} = \langle \sim F^{-1}, Y, X\rangle$, но X в инвертируемом соответствии будет областью прибытия, а Y — областью отправления.

Композиция. Даны два нечетких соответствия $\sim \Gamma_1 = \langle \sim F_1, X, Y\rangle$, $\sim \Gamma_2 = \langle \sim F_2, Y, Z\rangle$. Тогда два соответствия компонируют:

$$\sim \Gamma_1 \cdot \sim \Gamma_2 = \langle \sim F_3, X, Z\rangle.$$

Например. Пусть $\sim \Gamma_1 = \langle\{x_1, x_2, x_3\}, \{y_1, y_2\}, \{\langle x_1, 0{,}5, y_1\rangle, \langle x_1, 0{,}1, y_2\rangle, \langle x_2, 0{,}9, y_2\rangle, \langle x_3, 0{,}4, y_2\rangle\}\rangle$. Графически это $\sim \Gamma_1$ представлено на рис. 8.2.

Пусть $\sim \Gamma_2 = \langle\{y_1, y_2, y_3\}, \{z_1, z_2, z_3\}, \{\langle y_1, 0{,}4, z_1\rangle, \langle y_2, 0{,}8, z_2\rangle, \langle y_3, 0{,}2, z_3\rangle\}\rangle$. Графически $\sim \Gamma_2$ представлено на рис. 8.3.

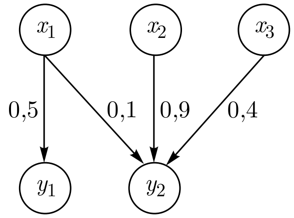


Рис. 8.2. Нечеткое соответствие $\sim \Gamma_1$

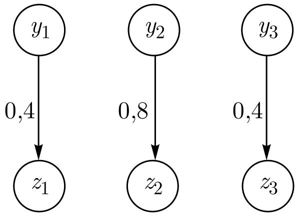


Рис. 8.3. Нечеткое соответствие $\sim \Gamma_2$

В процессе композиции получим рис. 8.4.

При склеивании элементов из множества Y остаются стрелки с меньшим весом. Тогда окончательный результат композиции рис. 8.5 будет $\sim \Gamma_3 = \sim \Gamma_1 \sim \Gamma_2 = \langle\{x_1, x_2, x_3\}, \{z_1, z_2, z_3\}, \{\langle x_1, 0{,}4, z_1\rangle, \langle x_1, 0{,}1, z_2\rangle, \langle x_2, 0{,}8, z_2\rangle, \langle x_3, 0{,}4, z_2\rangle\}\rangle$.

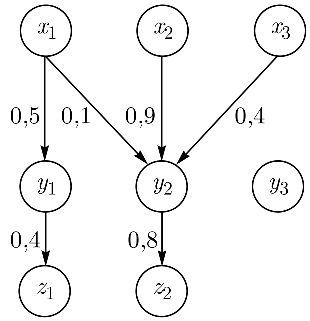


Рис. 8.4. Процесс композиции двух нечетких соответствий

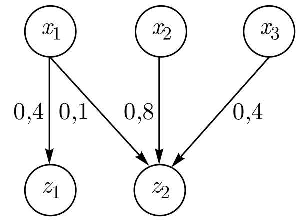


Рис. 8.5. Результат композиции двух нечетких соответствий

Нечетким отношением на произвольном множестве X называется выражение $\sim \varphi = \langle \sim \Phi, X \rangle$, причем $\sim \Phi \subseteq X \times X$, $\sim \Phi \subseteq X^2$.

X — область задания отношения;

$\sim \Phi$ — нечеткий график отношения.

Отметим, что нечеткое отношение является частным случаем нечеткого соответствия, у которого $X = Y$.

Нечеткие отношения также могут быть заданы в теоретико-множественном, матричном или графическом виде.

Например. $X = \{a, b, c\}$, $\sim\varphi = \langle\langle 0{,}2, \langle a, b\rangle\rangle, \langle 0{,}1, \langle b, c\rangle\rangle, \langle 0{,}9, \langle a, a\rangle\rangle, \langle 0{,}7, \langle c, c\rangle\rangle, \{a, b, c\}\rangle$.

В матричном виде:

R =

	a	b	c
a	0,9	0,2	0
b	0	0	0,1
c	0	0	0,7

В графическом виде (рис.8.6):

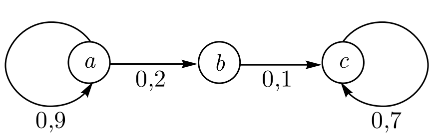


Рис. 8.6. Пример графического задания нечеткого отношения

Для расплывчатых отношений и соответствий справедливы все теоретико-множественные операции на основе определенных формул.

Примеры решения задач

Пример 8.1. Пусть заданы нечеткие высказывания $\widetilde{A} = 0{,}6$, $\widetilde{B} = 0{,}3$. Найти их импликацию и эквивалентность.

Решение. $\widetilde{A} \leftrightarrow \widetilde{B} = \min(\max(1-0{,}6; 0{,}3); \max(1-0{,}3; 0{,}6)) = \min(0{,}4; 0{,}7) = 0{,}4$.

Ответ. $\widetilde{A} \to \widetilde{B} = \max(1 - 0{,}6; 0{,}3) = 0{,}4$.

Пример 8.2. Пусть задано составное нечеткое высказывание:

$$\widetilde{D} = (\widetilde{A} \,\&\, \widetilde{B} \vee \widetilde{A} \,\&\, \rceil\widetilde{C} \vee \rceil\widetilde{B} \,\&\, \widetilde{C}) \to \rceil(\rceil\widetilde{A} \,\&\, \widetilde{B} \,\&\, \widetilde{C}),$$

при том, что $\widetilde{A} = 0{,}3$; $\widetilde{B} = 0{,}7$; $\widetilde{C} = 0{,}4$.

Найти степень истинности этого составного нечеткого высказывания.

Решение. $\widetilde{D} = \max[1 - (\widetilde{A} \,\&\, \widetilde{B} \vee \widetilde{A} \,\&\, \rceil\widetilde{C} \vee \rceil\widetilde{B} \,\&\, \widetilde{C}); 1 - (\rceil\widetilde{A} \,\&\, \widetilde{B} \,\&\, \widetilde{C})] = \max[1 - (\min(\widetilde{A}, \widetilde{B}) \vee \min(\widetilde{A}; 1 - \widetilde{C}) \vee \min(1 - \widetilde{B}; \widetilde{C})); 1 - \min(1 - \widetilde{A}; \widetilde{B}; \widetilde{C})] = \max[1 - \max(\min(\widetilde{A}; \widetilde{B}); \min(\widetilde{A}; 1 - \widetilde{C}); \min(1 - \widetilde{B}; \widetilde{C})); 1 - \min(1 - \widetilde{A}; \widetilde{B}; \widetilde{C})] = \max[1 - \max(0{,}3; 0{,}3; 0{,}3); 1 - 0{,}4] = \max[0{,}7; 0{,}6] = 0{,}7$.

Ответ. Степень истинности нечеткого высказывания $\widetilde{D}$ равна 0,7.

Пример 8.3. Заданы нечеткие множества $\widetilde{A} = \{\langle 0{,}6; x_1\rangle; \langle 0{,}3; x_2\rangle; \langle 0{,}8; x_4\rangle; \langle 0{,}4; x_5\rangle\}$ и $\widetilde{B} = \{\langle 0{,}2; x_2\rangle; \langle 0{,}5; x_3\rangle; \langle 0{,}7; x_4\rangle; \langle 0{,}9; x_5\rangle\}$ на множестве $X = \{x_1, x_2, x_3, x_4, x_5\}$.

Найти степень включения нечеткого множества $\widetilde{A}$ в $\widetilde{B}$.

Решение. $\nu(\widetilde{A}, \widetilde{B}) = (0{,}6 \to 0) \,\&\, (0{,}3 \to 0{,}2) \,\&\, (0 \to 0{,}5) \,\&\, (0{,}8 \to 0{,}7) \,\&\, (0{,}4 \to 0{,}9) = 0{,}4 \,\&\, 0{,}7 \,\&\, 1 \,\&\, 0{,}7 \,\&\, 0{,}9$.

Ответ. Степень включения нечеткого множества $\widetilde{A}$ в $\widetilde{B}$ равна $\nu(\widetilde{A}, \widetilde{B}) = 0{,}4$.

Пример 8.4. Заданы нечеткие множества $\widetilde{A} = \{\langle 0{,}4; x_1\rangle; \langle 0{,}7; x_2\rangle; \langle 0{,}2; x_4\rangle; \langle 0{,}9; x_5\rangle\}$ и $\widetilde{B} = \{\langle 0{,}6; x_3\rangle; \langle 0{,}3; x_4\rangle; \langle 0{,}8; x_5\rangle\}$ на множестве $X = \{x_1, x_2, x_3, x_4, x_5\}$.

Найти пересечение, объединение и разность этих множеств, а также дополнение множества $\widetilde{B}$.

Ответ.

$$\widetilde{A} \cap \widetilde{B} = \{\langle 0{,}2; x_4\rangle; \langle 0{,}8; x_5\rangle\};$$

$$\widetilde{A} \cup \widetilde{B} = \{\langle 0{,}4; x_1\rangle; \langle 0{,}7; x_2\rangle; \langle 0{,}6; x_3\rangle; \langle 0{,}3; x_4\rangle; \langle 0{,}9; x_5\rangle\};$$
$$\widetilde{A} \backslash \widetilde{B} = \{\langle 0{,}4; x_1\rangle; \langle 0{,}7; x_2\rangle; \langle 0{,}2; x_4\rangle; \langle 0{,}2; x_5\rangle\};$$
$$\widetilde{B} = \{\langle 1; x_1\rangle; \langle 1; x_2\rangle; \langle 0{,}4; x_3\rangle; \langle 0{,}7; x_4\rangle; \langle 0{,}2; x_5\rangle\}.$$

Пример 8.5. Задать различными способами произвольное соответствие $\widetilde{\Gamma} = (\widetilde{F}, X, Y)$.

Решение.

а) Пусть заданы конечные множества $X = \{x_1, x_2, x_3, x_4\}$; $Y=\{y_1, y_2, y_3, y_4\}$; $\widetilde{F}=\{\langle 0{,}3; \langle x_1; y_1\rangle\rangle; \langle 0{,}2; \langle x_1; y_3\rangle\rangle; \langle 0{,}4; \langle x_2; y_2\rangle\rangle;$ $\langle 0{,}6; \langle x_3; y_2\rangle\rangle; \langle 0{,}1; \langle x_4; y_1\rangle\rangle; \langle 0{,}8; \langle x_4; y_4\rangle\rangle\}$. Тогда можно сказать, что задано нечеткое соответствие $\widetilde{\Gamma} = (\widetilde{F}, X, Y)$ теоретическим способом.

б) То же соответствие, заданное матричным способом, будет иметь следующий вид:

$$\mathrm{R}_{\widetilde{\Gamma}} = \begin{vmatrix} 0{,}3 & 0 & 0{,}2 & 0 \\ 0 & 0{,}4 & 0 & 0 \\ 0 & 0{,}6 & 0 & 0 \\ 0{,}1 & 0 & 0 & 0{,}8 \end{vmatrix},$$

где строки матрицы соответствуют элементам множества X, а столбцы — элементам множества Y. Элементы матрицы принимают значения, равные степени принадлежности элементов множества $X \times Y$ множеству $\widetilde{F}$.

в) Графический способ (рис. 8.7). Графически нечеткое соответствие может быть задано в виде ориентированного двудольного взвешенного графа, где вес ребра совпадает со значением степени принадлежности элемента множеству $\widetilde{F}$.

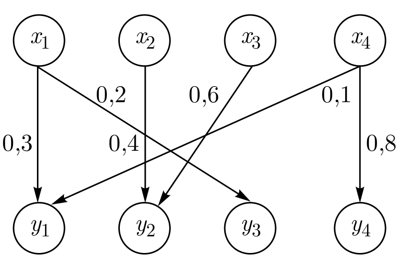


Рис. 8.7. График соответствия $\widetilde{\Gamma}$

Пример 8.6. Заданы нечеткие соответствия $\widetilde{\Gamma} = (\widetilde{F}, X, Y)$ и $\widetilde{\Delta} = (\widetilde{P}, Y, Z)$. График соответствия $\widetilde{\Gamma}$ изображен на рис. 8.7, а график соответствия $\widetilde{\Delta}$ — на рис. 8.8. Найти композицию этих соответствий.

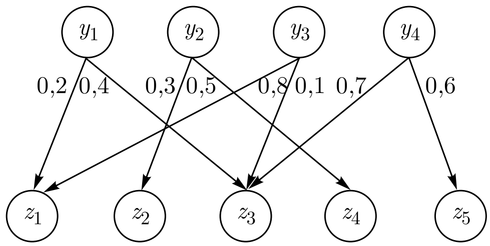


Рис. 8.8. График соответствия $\widetilde{\Delta}$

Ответ. Композиция соответствий $\widetilde{\Gamma}$ и $\widetilde{\Delta}$ показана на рис. 8.9.

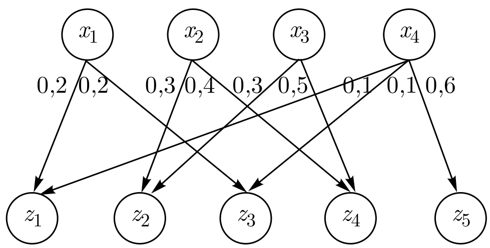


Рис. 8.9. Результат композиции соответствий $\widetilde{\Gamma}$ и $\widetilde{\Delta}$

Пример 8.7. Задать произвольное нечеткое отношение $\widetilde{\varphi}$.

Ответ.

а) Теоретический способ. $\widetilde{\varphi} = (\widetilde{F}, X)$; $X = \{x_1, x_2, x_3, x_4\}$; $\widetilde{F} = \{\langle 0{,}4; \langle x_1, x_2\rangle\rangle$; $\langle 0{,}7; \langle x_1, x_3\rangle\rangle$; $\langle 0{,}2; \langle x_2, x_1\rangle\rangle$; $\langle 0{,}5; \langle x_2, x_4\rangle\rangle$; $\langle 0{,}7; \langle x_3, x_2\rangle\rangle$; $\langle 0{,}3; \langle x_3, x_4\rangle\rangle$; $\langle 0{,}8; \langle x_4, x_2\rangle\rangle$; $\langle 0{,}1; \langle x_4, x_3\rangle\rangle\}$.

б) Матричный способ.

$$\mathrm{R}_{\widetilde{\varphi}} = \begin{vmatrix} 0 & 0{,}4 & 0{,}7 & 0 \\ 0{,}2 & 0 & 0 & 0{,}5 \\ 0 & 0{,}7 & 0 & 0{,}3 \\ 0 & 0{,}8 & 0{,}1 & 0 \end{vmatrix}.$$

Строки и столбцы матрицы соответствуют элементам множества X. Элементы матрицы принимают значение, равное степени принадлежности элементов множества X_2 множеству $\widetilde{F}$.

в) Графический способ. Графически нечеткое отношение может быть задано в виде ориентированного взвешенного графа (рис. 8.10), где вес ребра соответствует значению степени принадлежности элемента множеству $\widetilde{F}$.

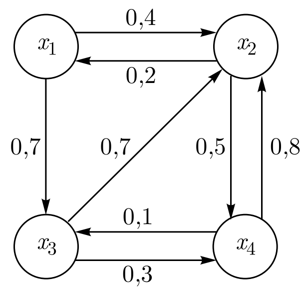


Рис. 8.10. График отношения $\widetilde{\varphi}$

Пример 8.8. Заданы нечеткие отношения $\widetilde{\varphi} = (\widetilde{F}, X)$ (рис. 8.10) и $\widetilde{\phi} = (\widetilde{P}, X)$ (рис. 8.11). Найти их объединение, пересечение и композицию.

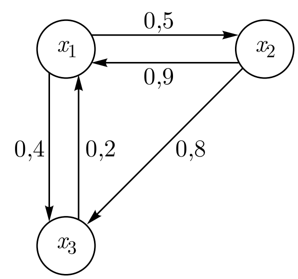


Рис. 8.11. График отношения $\widetilde{\phi}$

Ответ.

а) Объединение (рис. 8.12).

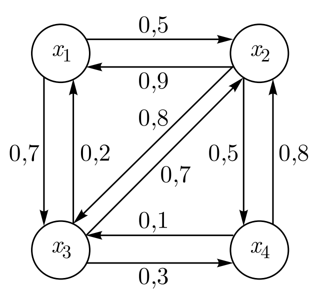


Рис. 8.12. Результат объединения отношений $\widetilde{\varphi}$ и $\widetilde{\phi}$

б) Пересечение (рис. 8.13).

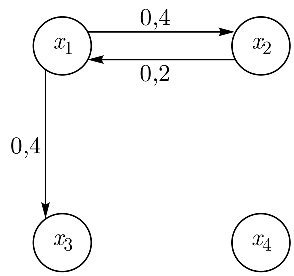


Рис. 8.13. Результат пересечения отношений $\widetilde{\varphi}$ и $\widetilde{\phi}$

в) Композиция (рис. 8.14).

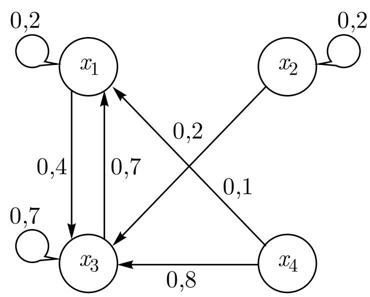


Рис. 8.14. Результат композиции отношений $\widetilde{\varphi}$ и $\widetilde{\phi}$

Для работы с нечеткостями используется также теория приближенных множеств, разработанная З. Павлаком. Основное преимущество этой теории — то, что при работе с ней не требуется предварительных сведений о степени принадлежности элемента множеству (как в нечетких множествах) и т. п. Использование этой теории помогает исследовать и анализировать неточные знания. Они определяются приближенно в рамках заданного обучающего множества на основе понятий нижнего и верхнего приближений. Причем нижние приближения включают те элементы исходного множества, которые наверняка принадлежат понятию. Верхние приближения включают все элементы исходного множества, которые возможно принадлежат понятию. Промежуточные элементы между этими приближениями образуют граничную область. Она содержит элементы исходного множества, которые не могут быть классифицированы точно на основе исходных данных.

Пусть на обучающем множестве X введено отношение эквивалентности $\varphi = \langle \Phi, X \rangle$, $\Phi \subseteq X \times X$. Здесь Φ — график отношения, X — область задания отношения. Упорядоченную пару $Q = \langle \varphi, X \rangle$ называют пространством приближений. Множество $A = \{Q_1, Q_2, \ldots, Q_l\}$ будем считать приближенным множеством.

Классы эквивалентности по отношению к φ называют элементарными множествами в кортеже Q. Составным множеством в Q считается конечное объединение элементарных множеств в Q. Следуя В. Н. Вагину и др., все множество классов эквивалентности обозначим $\varphi^*(D) = \{e_1, e_2, \ldots, e_n\}$, где D — множество атрибутов.

Пустое множество является также элементарным для любого пространства приближений. Любое конечное объединение элементарных множеств в Q называется составным множеством.

Под нижним приближением множества X в Q будем понимать наибольшее составное множество в Q, содержащееся в X, а под верхним приближением X в Q — наименьшее составное множество в Q, содержащее X. Обозначим нижнее и верхнее приближения множества X в пространстве Q через $\underline{Q}X$ и $\overline{Q}X$ соответственно. Множество $Bn_P(X) = \overline{Q}X - \underline{Q}X$ будем называть границей X в Q.

Рассмотрим пример. Пусть задана следующая табл. 8.1.

Таблица 8.1

Фамилия	Сфера деятельности	Уровень доходов
Иванов	Управление	Высокий
Петрова	Торговля	Средний +
Сидоров	Производство	Средний −
Николаев	Наука	Низкий
Попова	Управление	Высокий

Положим $D = \{$Сфера деятельности$\}$, тогда обучающая выборка разбивается на четыре класса эквивалентности:

$$R^*(D) = \{\{\text{Иванов, Попова}\}, \{\text{Петрова}\}, \{\text{Сидоров}\}, \{\text{Николаев}\}\}.$$

Здесь $e_1 = \{$Иванов, Попова$\}$, $e_2 = \{$Петрова$\}$, $e_3 = \{$Сидоров$\}$, $e_4 = \{$Николаев$\}$.

Любое объединение элементарных множеств называется составным множеством. Так, множество $\{e_2, e_3\} = \{$Петрова, Сидоров$\}$ — составное множество.

Пусть в нашем примере X — класс людей с высоким доходом. Тогда

$$\underline{Q}X = \{\text{Иванов, Попова}\},$$

$$\overline{Q}X = \{\text{Иванов, Попова, Петрова, Сидоров}\},$$

$$Bn_P(X) = \overline{Q}X - \underline{P}X = \{\text{Петрова, Сидоров}\}.$$

Приближенные множества в основном используются для классификации объектов.

Два произвольных множества Y и Z приближенно равны снизу в P, если равны их нижние приближения в пространстве P: $\underline{Q}Y = \underline{Q}Z$.

Множества Y и Z приближенно равны сверху, если совпадают их верхние приближения: $\overline{Q}Y = \overline{Q}Z$.

Если два множества приближенно равны сверху и снизу одновременно, они называются приближенно равными и обозначаются как $Y \approx Z$.

Для приближенных множеств справедливы следующие выражения:

1. $\underline{Q}Y \subset Y \subset \overline{Q}Y$,
2. $\underline{Q}1 = \overline{Q}l = 1$,
3. $\underline{Q}0 = \overline{Q}0 = 0$,
4. $\underline{\underline{Q}}(Y \cap Z) = \underline{\underline{Q}}Y \cap \underline{\underline{Q}}Z$,
5. $\overline{\overline{Q}}(Y \cup Z) = \overline{\overline{Q}}Y \cup \overline{\overline{Q}}Z$.

Здесь 0 обозначает пустое множество, 1 обозначает все множество X.

Контрольные вопросы

1. Что такое нечеткое множество?
2. Привести способы задания нечеткого множества.
3. Что называется нечетким высказыванием? Привести примеры нечетких высказываний.
4. Как определяется степень истинности сложного нечеткого высказывания?
5. Как определяется степень включения двух нечетких множеств?
6. Как определяется операция дополнения нечеткого множества?
7. Как определяются операции объединения, пересечения, разности двух или более нечетких множеств?
8. Что называется нечетким соответствием?

9. Как определяются операции инверсии и композиции для нечетких соответствий?

10. Что называется нечетким отношением?

11. Как определяются операции над нечеткими отношениями?

12. Что такое приближенное множество?

13. Привести способы задания приближенного множества.

14. Приведите примеры приближенных множеств.

Задания для самостоятельной работы

1. Пусть задано произвольное четкое множество $X = \{0, 1, 2, \ldots, 10\}$. Задать нечеткие множества, соответствующие, по вашему мнению, понятиям «не много», «не мало», «средне», «очень мало», «очень много», «вполне достаточно», «не много и не мало», «не очень много и не очень мало».

2. Пусть $\widetilde{A}$, $\widetilde{B}$, $\widetilde{C}$ — нечеткие высказывания, причем $\widetilde{A} = 0{,}3$, $\widetilde{B} = 0{,}8$, $\widetilde{C} = 0{,}1$.

Определить степень истинности составного высказывания $\widetilde{D}$, если:

а) $\widetilde{D} = (\widetilde{A} \to \widetilde{B}) \to \widetilde{C}$;

б) $\widetilde{D} = (\rceil \widetilde{A} \vee \rceil \widetilde{B}) \,\&\, \rceil \widetilde{C}$;

в) $\widetilde{D} = (\rceil \widetilde{A} \to \widetilde{C}) \vee \rceil \widetilde{B}$;

г) $\widetilde{D} = \rceil((\rceil \widetilde{A} \leftrightarrow \rceil \widetilde{C}) \,\&\, (\rceil \widetilde{C} \to \rceil \widetilde{B}))$;

д) $\widetilde{D} = \rceil((\widetilde{B} \vee \rceil \widetilde{C}) \to \widetilde{A}) \,\&\, \widetilde{C}$;

е) $\widetilde{D} = (\rceil(\rceil \widetilde{A} \vee \widetilde{A}) \,\&\, \rceil(\widetilde{B} \vee \rceil \widetilde{B}) \vee (\widetilde{C} \,\&\, \rceil \widetilde{C})) \to \widetilde{A} \,\&\, \widetilde{B} \,\&\, \widetilde{C}$.

3. Пусть заданы произвольные нечеткие множества:

$$\widetilde{A} = \{\langle 0{,}8; x_1 \rangle; \langle 0{,}4; x_2 \rangle; \langle 0{,}3; x_3 \rangle; \langle 0{,}6; x_4 \rangle\};$$
$$\widetilde{B} = \{\langle 0{,}7; x_2 \rangle; \langle 0{,}6; x_3 \rangle; \langle 0{,}2; x_4 \rangle; \langle 0{,}1; x_5 \rangle\}$$

на произвольном четком множестве $X = \{x_1, x_2, x_3, x_4, x_5\}$.

Найдите нечеткое множество $\widetilde{C}$, такое что:

а) $\widetilde{C} = \widetilde{A} \cap \widetilde{B}$;

б) $\widetilde{C} = \rceil \widetilde{A} \cap \widetilde{B}$;

в) $\widetilde{C} = \rceil(\rceil \widetilde{A} \cap \rceil \widetilde{B})$;

г) $\widetilde{C} = \widetilde{A} \cup \widetilde{B}$;

д) $\widetilde{C} = \widetilde{A} \cup \rceil \widetilde{B}$;

е) $\widetilde{C} = \rceil(\rceil \widetilde{A} \cup \widetilde{B})$;

ж) $\widetilde{C} = \widetilde{A} \backslash \widetilde{B}$;

з) $\widetilde{C} = \rceil \widetilde{A} \backslash \rceil \widetilde{B}$;

и) $\widetilde{C} = \rceil (\widetilde{A} \backslash \rceil \widetilde{B})$.

4. Пусть задано нечеткое соответствие $\widetilde{\Gamma} = (\widetilde{F}, X, Y)$, где $X = \{x_1, x_2, x_3, x_4\}$, $Y = \{y_1, y_2, y_3, y_4, y_5\}$, $\widetilde{F} = \{\langle 0{,}7, \langle x_1, y_2\rangle\rangle$, $\langle 0{,}9, \langle x_1, y_5\rangle\rangle, \langle 0{,}1, \langle x_2, y_1\rangle\rangle, \langle 0{,}3, \langle x_2, y_2\rangle\rangle, \langle 0{,}6, \langle x_2, y_3\rangle\rangle, \langle 1, \langle x_3, y_4\rangle\rangle$, $\langle 0{,}8, \langle x_4, y_2\rangle\rangle, \langle 0{,}4, \langle x_4, y_3\rangle\rangle\}$.

Запишите матрицу инциденций R_Γ, постройте граф соответствия $\widetilde{\Gamma}$.

5. Для нечетких соответствий $\widetilde{\Gamma} = (\widetilde{F}, X, Y)$ и $\widetilde{\Delta} = (\widetilde{P}, Y, Z)$, где $X = \{x_1, x_2, x_3, x_4\}$, $Y = \{y_1, y_2, y_3, y_4, y_5\}$, $\widetilde{F} = \{\langle 0{,}7, \langle x_1, y_2\rangle\rangle$; $\langle 0{,}9, \langle x_1, y_5\rangle\rangle; \langle 0{,}1, \langle x_5, y_1\rangle\rangle; \langle 0{,}3, \langle x_2, y_2\rangle\rangle; \langle 0{,}6, \langle x_2, y_3\rangle\rangle; \langle 1, \langle x_3, y_4\rangle\rangle$; $\langle 0{,}8, \langle x_4, y_2\rangle\rangle; \langle 0{,}4, \langle x_4, y_3\rangle\rangle\}$, $Y = \{y_1, y_2, y_3, y_4\}$, $Z = \{z_1, z_2, z_3\}$, $\widetilde{P} = \{\langle 0{,}6, \langle y_1, z_1\rangle\rangle; \langle 0{,}3, \langle y_1, z_2\rangle\rangle; \langle 1, \langle y_3, z_3\rangle\rangle; \langle 0{,}8, \langle y_4, z_1\rangle\rangle\}$. Найти:

а) $\widetilde{\Gamma}^{-1}$;

б) $\widetilde{\Delta}^{-1}$;

в) $\widetilde{\Gamma} \cdot \widetilde{\Delta}$;

г) $\widetilde{\Delta}^{-1} \cdot \widetilde{\Gamma}^{-1}$.

6. Заданы нечеткие отношения $\widetilde{\varphi} = (\widetilde{F}, X)$ и $\widetilde{\phi} = (\widetilde{P}, X)$, где $X = \{x_1, x_2, x_3, x_4, x_5\}$, $\widetilde{F} = \{\langle 0{,}4, \langle x_1, x_2\rangle\rangle; \langle 0{,}3, \langle x_1, x_3\rangle\rangle$; $\langle 0{,}5, \langle x_1, x_5\rangle\rangle; \langle 0{,}9, \langle x_2, x_1\rangle\rangle; \langle 0{,}8, \langle x_2, x_3\rangle\rangle; \langle 0{,}1, \langle x_3, x_2\rangle\rangle; \langle 0{,}6, \langle x_3, x_3\rangle\rangle$; $\langle 0{,}7, \langle x_5, x_1\rangle\rangle; \langle 0{,}2, \langle x_5, x_2\rangle\rangle\}$, $\widetilde{P} = \{\langle 0{,}3, \langle x_1, x_1\rangle\rangle; \langle 0{,}9, \langle x_1, x_4\rangle\rangle$; $\langle 0{,}1, \langle x_2, x_3\rangle\rangle; \langle 0{,}4, \langle x_3, x_1\rangle\rangle; \langle 0{,}5, \langle x_4, x_2\rangle\rangle\}$.

Найти их пересечение, объединение, инверсию и композицию.

7. На основе данных о прогнозе погоды на неделю (взять из Интернета) построить примеры приближенных множеств для следующих действий:

а) путешествия;

б) занятия спортом на открытом воздухе;

в) строительство;

г) остаться дома.

Тестовые задания к модулю 1

1. Объект, принадлежащий множеству, называется его ...

- элементом
- подмножеством
- мощностью
- семейством
- надмножеством

2. Запись $\rceil(\forall x \in X)(x \in Y)$ означает:

- любой элемент x, принадлежащий множеству X, принадлежит и множеству Y
- существует элемент x, принадлежащий множеству X, являющийся элементом множества Y
- ни один элемент множества X не принадлежит множеству Y
- существует элемент x, который принадлежит множеству X и не принадлежит множеству Y
- не каждый элемент из множества X принадлежит множеству Y

3. Количество элементов конечного множества называется ...

- элементом
- подмножеством
- мощностью
- семейством
- любой из перечисленных вариантов

4. Множество A, любой элемент которого принадлежит множеству B, называется ... множества B.

- элементом
- подмножеством
- мощностью
- семейством
- надмножеством

5. Запись $A \subset B$ означает:

- элемент A принадлежит B
- множество A строго включается в множество B
- множество A нестрого включается в множество B
- множество B является надмножеством множества A
- множество A эквивалентно множеству B

6. Множества A и B равны, если ...

- $A = \{a, b, c\}$; $B = \{b, c, a\}$

- $A = \{\{a, b, c\}\}$; $B = \{a, b, c\}$
- $A = \{a, b, \{c\}\}$; $B = \{b, c, a\}$
- $A = \{\{a, b, c\}\}$; $B = \{\{a\}, \{b\}, \{c\}\}$
- $A = \{a, b\}$; $B = \{b, c\}$

7. Утверждение $((\forall x \in A)(x \in B \,\&\, A \neq B))$ **означает, что множество** A **...**

- строго включается в множество B
- включается в множество B
- является истинным подмножеством множества B
- является подмножеством множества B
- является надмножеством множества B

8. Запись $A \subseteq B$ **означает:**

- элемент A принадлежит B
- множество A строго включается в множество B
- множество A нестрого включается в множество B
- множество B является надмножеством множества A
- множество A эквивалентно множеству B

9. Множество, состоящее из элементов, принадлежащих множеству A **или** B**, называется ...**

- объединением множеств A и B
- пересечением множеств A и B
- разностью множеств A и B
- дополнением множества A до B
- произведением множеств A и B

10. Множество, состоящее из элементов, принадлежащих множеству A **и** B**, называется ...**

- объединением множеств A и B
- пересечением множеств A и B
- разностью множеств A и B
- дополнением множества A до B
- произведением множеств A и B

11. Множество, состоящее из элементов, не принадлежащих множеству A **и принадлежащих множеству** B**, называется ...**

- объединением множеств A и B
- пересечением множеств A и B
- разностью множеств A и B
- дополнением множества A до B
- произведением множеств A и B

12. Не применяется для доказательства тождеств с множествами:

- метод взаимного включения
- диаграммы Эйлера–Венна
- метод от противного
- последовательный метод
- любой из перечисленных методов

13. Тривиальным разбиением множества A является:

- разбиение множества A на непустые подмножества, объединение которых равно множеству A
- разбиение множества A на попарно непересекающиеся подмножества
- разбиение множества A, содержащее один класс, совпадающий с множеством A
- поэлементное разбиение множества A на непустые подмножества, объединение которых равно множеству A
- разбиение множества A на непустые подмножества

14. Множество $A = \varnothing$ может иметь ... разбиений.

- 0
- 1
- 2
- n
- бесконечное число

15. Множество $|A| = 2$ имеет ... разбиений.

- 0
- 1
- 2
- не больше двух
- больше двух

16. Множество, элементами которого являются все возможные наборы вида $\langle x_1, x_2, \ldots, x_n \rangle$, причем $x_1 \in X_1$, $x_2 \in \in X_2, \ldots, x_n \in X_n$, называется ...

- декартовым произведением множеств $X_1, X_2, \ldots, X_n$
- кортежем длины n
- композицией множеств $X_1, X_2, \ldots, X_n$
- проекцией множеств $X_1, X_2, \ldots, X_n$
- биекцией множеств $X_1, X_2, \ldots, X_n$

17. Упорядоченный набор $\langle x_1, x_2, \ldots, x_n \rangle$ элементов множеств $X_1, X_2, \ldots, X_n$, причем таких, что $x_1 \in X_1$, $x_2 \in X_2, \ldots, x_n \in X_n$, называется ...

- декартовым произведением множеств $X_1, X_2, \ldots, X_n$
- кортежем длины n
- композицией множеств $X_1, X_2, \ldots, X_n$
- проекцией множеств $X_1, X_2, \ldots, X_n$
- биекцией множеств $X_1, X_2, \ldots, X_n$

18. Множество является ..., если каждый элемент его — кортеж длины два.

- степенью
- инверсией
- композицией
- проекцией
- графиком

19. Множество является графиком, если каждый элемент его — ... длины два.

- график
- кортеж
- множество
- число
- порядок

20. Инверсия графика определяется через ... кортежа.

- произведение
- инверсию
- проекцию
- композицию
- биекцию

21. Множество, являющееся проекцией на первую координату данного графика, называется его ...

- первой проекцией
- второй проекцией
- областью значения
- областью определения
- областью задания

22. Множество, являющееся проекцией на вторую координату данного графика, называется его ...

- первой проекцией
- второй проекцией
- областью значения

- областью определения
- областью задания

23. Среди предложенных вариантов истинны следующие:

- декартово произведение двух множеств является кортежем
- кортеж длины два, первый элемент которого принадлежит множеству A, а второй множеству B, является элементом декартова произведения множеств A и B
- декартово произведение двух множеств обладает свойством некоммутативности
- декартово произведение множеств не обладает свойством ассоциативности
- декартово произведение множеств обладает свойством идемпотентности

24. Множество называется графиком, если каждый его элемент ...

- кортеж длины два
- кортеж длины три
- принадлежит множеству
- не принадлежит множеству
- является точкой на координатной плоскости

25. Соответствие может быть задано ... способом.

- теоретическим
- графическим
- матричным
- высказывательным
- аналитическим

26. Соответствия $\Gamma = \langle F, X, Y\rangle$ и $\Delta = \langle P, W, Z\rangle$ называются равными, если ...

- $X = W$
- $X = W \,\&\, Y = Z$
- $F = P$
- $X = W \,\&\, Y = Z \,\&\, F = P$
- $Y = Z$

27. Для соответствий $\Gamma = \langle F, X, Y\rangle$; $\Delta = \langle P, W, Z\rangle$ справедливы выражения:

- $\Gamma \cdot \Delta = \langle X \cdot W; Y \cdot Z; F \cdot P\rangle$
- $\Gamma \cdot \Delta = \langle X; Z; F \cdot P\rangle$
- $\Gamma \cdot \Delta = \langle W; Y, F \cdot P\rangle$
- $\Gamma \cdot \Delta = \langle X; Z; F \times P\rangle$
- $\Gamma \cdot \Delta = \langle W; Y, F \times P\rangle$

28. Соответствие $\Gamma_B = \langle F \cap (B \times Y), X, Y \rangle$ называется ... соответствия Γ на множество B.

- образом
- прообразом
- сужением
- продолжением
- сечением

29. К соответствиям могут применяться операции:

- композиция
- инверсия
- объединение
- дополнение
- сумма

30. Если каждому элементу $x \in X$ соответствует строго один элемент из множества Y, то график соответствия $\Gamma = \langle F, X, Y \rangle$ называется ...

- функциональным
- антифункциональным
- инъективным
- антиинъективным
- нефункциональным

31. Если график F соответствия Γ не содержит пар с разными первыми и одинаковыми вторыми элементами, то он называется ...

- функциональным
- антифункциональным
- инъективным
- антиинъективным
- нефункциональным

32. Если соответствие является функциональным, всюду определенным, инъективным, сюръективным, то оно называется ...

- полным
- комплексным
- совершенным
- биективным
- оптимальным

33. Пара множеств, одно из которых является подмножеством квадрата другого, называется ...

- упорядоченным множеством

- кортежем
- отношением
- соответствием
- множеством

34. Пусть задано отношение $\varphi = (F, X)$. Тогда множество F называется ...

- областью задания отношения
- графиком отношения
- областью прибытия отношения
- областью отправления отношения
- множеством отношения

35. Пусть задано отношение $\varphi = (F, X)$. Тогда неупорядоченное множество X называется ...

- областью задания отношения
- графиком отношения
- областью прибытия отношения
- областью отправления отношения
- отображением отношения

36. Отношение $\varphi = (F, X)$ называется отношением равенства, если ...

- $X = \varnothing$
- $F = \Delta X$
- $F = X^2$
- $F \subseteq X^2$
- $X^2 \subseteq F$

37. Отношение $\varphi = (F, X)$ называется полным отношением, если ...

- $X = \varnothing$
- $F = \varnothing$
- $F = X^2$
- $F \subseteq X^2$
- $X^2 \subseteq F$

38. График отношения $= (F, X)$, где $X = \{x_1, x_2, x_3\}$; $F = \{\langle x_1, x_1\rangle; \langle x_1, x_2\rangle; \langle x_2, x_2\rangle; \langle x_2, x_3\rangle; \langle x_3, x_1\rangle; \langle x_3, x_3\rangle\}$ обладает свойствами:

- полнота
- транзитивность
- рефлексивность
- симметричность
- эквивалентность

39. График отношения $\varphi = (F, X)$**, где** $X = \{x_1, x_2, x_3\}$**;** $F = \{\langle x_1, x_2\rangle; \langle x_1, x_3\rangle; \langle x_2, x_1\rangle; \langle x_3, x_1\rangle; \langle x_3, x_3\rangle\}$ **обладает свойствами:**

- полнота
- транзитивность
- рефлексивность
- симметричность
- эквивалентность

40. График отношения $\varphi = (F, X)$**, где** $X = \{x_1, x_2, x_3\}$**;** $F = \{\langle x_1, x_2\rangle; \langle x_1, x_3\rangle; \langle x_2, x_3\rangle; \langle x_3, x_3\rangle\}$ **обладает свойствами:**

- полнота
- транзитивность
- рефлексивность
- симметричность
- эквивалентность

КРИТЕРИИ ОЦЕНКИ

Для оценки уровня полученных знаний при выполнении разработанных тестовых заданий по модулю 1 предлагается использовать следующую шкалу:

85–100 % правильных ответов — оценка «отлично»;

70–85 % правильных ответов — оценка «хорошо»;

55–70 % правильных ответов — оценка «удовлетворительно»;

менее 55 % правильных ответов — оценка «неудовлетворительно».

Нечеткие и приближенные высказывания, множества, соответствия и отношения позволяют формально задавать расплывчатую информацию в виде, удобном для обработки на ЭВМ.

Глоссарий к модулю 1

Словарь — это Вселенная
в алфавитном порядке.

М. Вольтер

Глава 1

Множество — это любое объединение в одно целое M определенных, вполне различимых объектов m из нашего восприятия или мысли, которые можно считать элементами из M. Существует неформальное определение: *множество* — это многое, мыслимое как единое.

Элементами множества являются предметы, его составляющие.

Принадлежность элемента x множеству A определяется записью $x \in A$. *Непринадлежность элемента* x множеству B определяется записью $y \notin B$.

Высказывание — это предположение (предложение), которое считается истинным или ложным.

Предикатом называется высказывание, содержащее переменные, принимающие значения 1 или 0 в зависимости от значения переменных.

Мощностью множества конечного множества называется количество его элементов.

Квантор общности $[(\forall x \in X)B(x)]$ означает, что для любого элемента x из множества X истинно высказывание $B(x)$ об этом элементе.

Квантор существования $[(\exists x \in X)B(x)]$ означает, что существует хотя бы один элемент x из множества X, для которого истинно высказывание $B(x)$ об этом элементе.

Конечное множество — это такое множество, число элементов которого представляет натуральное число. Множество называется *бесконечным*, если оно не является конечным.

Перечислительный способ задания множества состоит в составлении полного списка элементов множества, заключенного в фигурные скобки, и применяется только для конечных множеств.

Высказывательный способ задания множества состоит в задании такого свойства множества, наличие которого у элементов данного множества является истиной.

Равными множествами считаются множества, содержащие одни и те же элементы и имеющие равную мощность.

Неравные множества состоят из различных элементов.

Экстраординарными множествами называются множества, которые содержат себя в качестве одного из своих элементов.

Ординарными множествами называются множества, не относящиеся к экстраординарным.

Подмножеством множества A называется множество B, если любой элемент множества B принадлежит множеству A.

Множество B *строго включается* в множество A, если одновременно выполняются два условия ($B \subseteq A \,\&\, A \neq B$).

Множество B *нестрого включается* в множество A, если выполняется одно из двух условий ($B \subset A \vee A = B$).

Глава 2

Семейством подмножеств множества M называется множество всех подмножеств множества M.

Объединением множеств A и B называют множество C, которое состоит из тех элементов, которые принадлежат или множеству A, или множеству B, или обоим множествам одновременно.

Пересечением множеств A и B называют множество C, состоящее из тех элементов, которые принадлежат одновременно и множеству A, и множеству B.

Разностью множеств A и B называется множество C, состоящее из тех элементов, которые принадлежат множеству A и не принадлежат множеству B.

Дополнением множества A до некоторого универсального множества E называется множество, состоящее из элементов, принадлежащих множеству E и не принадлежащих множеству A.

Записью считается выражение, содержащее буквы и знаки операций над множествами.

Длиной записи является число букв и знаков операций, входящих в данную запись.

Метод двух включений (взаимного включения) доказательства равенства множеств основан на свойстве включения множеств

$$E = F \to E \overset{\substack{(\text{необходимость})\\ \downarrow}}{\subseteq} F \,\&\, F \underset{\substack{\uparrow \\ (\text{достаточность})}}{\subseteq} E.$$

Причем доказательство включения $E \subseteq F$ называется необходимостью, а доказательство включения $F \subseteq E$ — достаточностью.

Глава 3

Декартовым произведением двух множеств называется новое множество, состоящее из всех тех и только тех пар, т. е. кортежей длины 2, первая компонента которых принадлежит первому множеству, а вторая — второму.

Степенью S множества M называется декартово произведение S-одинаковых множеств, равных M.

Проекцией кортежа α на i-ю ось называется i-я компонента кортежа α.

Проекция множества M — это множество проекций кортежей из M.

График — это множество пар, т. е. множество, каждый элемент которого является парой или кортежем длины 2.

Инверсией кортежа $\alpha = \langle c, d\rangle$ называется кортеж $\beta = \langle d, c\rangle$.

Композицией двух кортежей $\beta = \langle x, z\rangle$ и $\gamma = \langle z, y\rangle$ называется кортеж $\alpha = \langle x, y\rangle$.

Инверсией графика называют множество инверсий его пар (элементов).

График называется *симметричным*, если он наряду с любой своей парой содержит ее инверсию.

Диагональю множества X ($|X| = n$) называется график, содержащий пары вида $\langle x_1, x_1\rangle$, $\langle x_2, x_2\rangle, \ldots, \langle x_n, x_n\rangle$.

Композицией двух графиков $\mathbb{P}$ и $\mathbb{Q}$ называется график $\mathbb{R}$ тогда и только тогда, когда $\exists z$ такое, что $\langle x, z\rangle \in \mathbb{P} \,\&\, \langle z, y\rangle \in \mathbb{Q}$.

График называется *функциональным*, если в нем нет пар с одинаковыми первыми и разными вторыми компонентами.

График называется *инъективным*, если в нем нет пар с различными первыми и одинаковыми вторыми компонентами.

Глава 4

Отношение — это связь между любыми объектами в природе. *Отношение* — это пара множеств, причем упорядоченная, первая компонента которой является подмножеством квадрата второй компоненты.

Запись $\varphi = \langle \Phi, M\rangle$ называется отношением, т. е. $\Phi \subseteq M \times M$, где φ — отношение; Φ — график отношения; M — область задания отношения, представляющая из себя неупорядоченное множество.

Отношение называется *бинарным*, если $\Phi \subseteq M^2$.

Отношение называется *k-арным,* если $\Phi \subseteq M^k$.

Отношение $\varphi = \langle \Phi, M \rangle$ называется *полным* отношением, если для любых элементов x, y, принадлежащих множеству M, истинно высказывание, что элементы x, y находятся в отношении φ.

Отношение $\varphi = \langle \Phi, M \rangle$ называется *пустым*, если график Φ является пустым множеством.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *равенства*, если $\Phi = \Delta M$.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *неравенства*, если $\Phi = M^2 \backslash \Delta M$.

Инверсия отношения определяется через инверсию элементов его графика на заданном множестве.

Отношение φ называется отношением *рефлексивности*, если для любого элемента $x \in M$ истинно высказывание $x \, \varphi \, x$.

Отношение $\varphi = \langle \Phi, M \rangle$ называется отношением *антирефлексивности*, если $(\forall x \in M) \neg (x \, \varphi \, x)$, т. е. $\Delta M \cap \Phi = \varnothing$.

Отношение $\varphi = \langle \Phi, M \rangle$ называют отношением *симметричности*, если $(\forall x, y \in M)(x \, \varphi \, y \to y \, \varphi \, x)$.

Отношение $\varphi = \langle \Phi, M \rangle$ называется *транзитивным*, если $(\forall x, y, z \in M)(x \, \varphi \, y \,\&\, y \, \varphi \, z \to x \, \varphi \, z)$.

Отношение φ называется отношением *эквивалентности*, если оно рефлексивно, симметрично и транзитивно.

Система W множеств называется *разбиением* одного множества M, если она удовлетворяет следующим условиям: подмножество обязательно должно включаться как в систему, так и в отдельное множество; подмножества разбиений не должны быть пустыми; подмножества должны быть разными и обязательно находиться в различных разбиениях; все подмножества разбиения в совокупности должны представлять множество M.

Отношение φ на множестве M и разбиение этого множества *сопряжены*, если $\forall x, y \in M \; x \, \varphi \, y$ истина тогда, когда x и y принадлежат одному и тому же блоку разбиения.

Отношение $\varphi = \langle \Phi, M \rangle$ называется *связным*, если выполняется условие $M^2 \backslash \Delta M \subseteq \Phi \cup \Phi^{-1}$.

Отношение φ называется отношением *нестрогого порядка*, если оно транзитивно, рефлексивно и антисимметрично.

Отношение φ называется отношением *совершенно нестрогого порядка*, если оно связно и является отношением нестрогого порядка.

Отношение φ называется отношением *совершенно строгого порядка*, если оно связно и является отношением строгого порядка.

Отношение φ называется отношением *доминирования* (x доминирует y), если $x \gg y$, т. е. x в чем-то превосходит y.

Отношение φ называется отношением *квазипорядка*, если оно транзитивно и рефлексивно.

Отношение φ называется *отношением толерантности*, если оно рефлексивно и симметрично.

Глава 5

Соответствие — это тройка множеств, первая компонента которой является графиком, вторая компонента — областью отправления графика и третья — областью прибытия графика.

Композиция соответствий определяется через композицию их графиков.

Образом множества A при соответствии Γ называется подмножество тех элементов области прибытия, которые соответствуют элементам множества A.

Прообразом множества B при соответствии Γ называется множество тех элементов области отправления, каждому из которых соответствует какой-нибудь элемент множества B.

Соответствие называется *функциональным*, если его график функционален.

Соответствие называется *инъективным*, если его график инъективен.

Соответствие называется *всюду определенным*, если его область определения совпадает с его областью отправления.

Соответствие называется *сюръективным*, если его область значений совпадает с его областью прибытия.

Соответствие называется *биективным* соответствием, или *биекцией*, или *взаимно-однозначным соответствием*, если оно функционально, инъективно, всюду определено и сюръективно.

Отображением множества X в множество Y называется всюду определенное соответствие.

Функцией из множества X в множество Y называется соответствие, при котором каждый элемент множества X связан с единственным элементом множества Y.

Функция F: $X \to X$ называется *тождественной*, когда множество X отображается само в себя.

Функция всюду определена, когда для любого элемента множества X всегда существует как минимум одно отображение.

Функция тогда и только тогда *биективна*, когда она всюду определена, сюръективна и инъективна.

Функция F называется *константной*, если существует такой элемент $b \in Y$, что график $\Phi = X \times \{b\}$.

Принцип Дирихле. Пусть $f\colon X \to Y$ — функция, причем X и Y — конечные множества. Если $|X| > |Y|$, то, по крайней мере, одно значение f встретится более одного раза.

Глава 6

Множество всех целых, натуральных, рациональных, неотрицательных и так далее чисел является примером *упорядоченных бесконечных множеств*.

Интервалом (a, b) *упорядоченного бесконечного множества* X называется множество всех элементов, лежащих между a и b.

Сегментом $[a, b]$ *упорядоченного бесконечного множества* называется интервал, включающий два его конца, т. е. элементы a и b.

Два элемента a и b множества X называются *соседними*, если интервал (a, b) пустой.

Соответствием подобия называют взаимно однозначное отображение одного множества на другое, которое сохраняет порядок.

Сечением упорядоченного бесконечного множества называется разбиение этого множества на два непустых подмножества таких, что каждый элемент одного множества низшего класса всегда предшествует каждому элементу второго множества верхнего класса.

Скачком называется сечение, в котором в нижнем классе есть наибольший элемент, а в верхнем классе есть наименьший элемент.

Дедекиндовыми сечениями называются сечения, в которых в нижнем классе нет наибольшего элемента, а в верхнем классе есть наименьший или наоборот.

Щелью называется сечение, при котором в нижнем классе нет наибольшего элемента, а в верхнем классе нет наименьшего элемента.

Счетным множеством называется любое множество, равномощное натуральному ряду чисел.

Несчетным множеством называется любое множество, сводимое к множеству действительных чисел.

Множество *мощности континуума* (*континуальное* множество) называется множество, равномощное множеству действительных положительных чисел, не превосходящих единицы. Множества точек, расположенных на отрезках, получили название *множества мощности континуума*.

Проблема континуума заключается в попытке ответить на вопрос: существует ли промежуточное множество, у которого больше элементов, чем множество натуральных чисел, и меньше элементов, чем множество точек на прямой.

Антиномия — противоречия (парадоксы), возникающие в теории множеств.

Глава 7

Мультимножество — это множество с повторяющимися элементами, где один и тот же элемент может присутствовать многократно.

Компонента мультимножества — это группа одинаковых элементов.

Экземпляры элементов мультимножества — элементы мультимножества, входящие в компоненту.

Функция кратности — это функция, принимающая числовые значения и определяющая число вхождений элемента в мультимножество.

Принадлежность элемента мультимножеству определяется значением функции кратности.

Порождающим множеством или *доменом* называется мультимножество из элементов которого образуются все мультимножества некоторого семейства.

Мощность мультимножества определяется как общее число экземпляров всех его элементов.

Размерность мультимножества — это общее число его различных элементов.

Высота (пиковое значение) мультимножества — максимальное значение его функции кратности.

Мультимножества A_M и B_M называются *равными* ($A_M = B_M$), если $k_i(x_i) = k_j(x_j)$ для всех элементов $x_i, x_j \in G$, $k_i(x_i) \in A_M$, $k_j(x_j) \in B_M$.

Мультимножества A и B называют *равномощными*, если $|A| = |B|$.

Мультимножества A и B называют *равноразмерными*, если $/A/ = /B/$.

Мультимножество B_M *включено* (содержится) в мультимножество A_M ($B_M \subseteq A_M$), если $k_j x_j \leqslant k_i x_i$, для каждого элемента $x_i, x_j \in G$, $k_i x_i \in A_M$, $k_j x_j \in B_M$.

Объединением мультимножеств называется мультимножество, состоящее из всех элементов, которые присутствуют хотя бы в одном из мультимножеств, и кратность каждого элемента равна максимальной кратности соответствующих элементов в объединяемых мультимножествах.

Пересечением мультимножеств называется мультимножество, состоящее из всех элементов, которые одновременно присутствуют в каждом из мультимножеств, и кратность каждого элемента равна минимальной кратности соответствующих элементов в пересекаемых мультимножествах.

Арифметической суммой мультимножеств называется мультимножество, состоящее из всех элементов, которые присутствуют хотя бы в одном из мультимножеств, и кратность каждого элемента равна сумме кратностей соответствующих элементов в складываемых мультимножествах.

Арифметической разностью мультимножеств A_M и B_M называется мультимножество, состоящее из тех элементов мультимножества A_M, кратность которых превышает кратность соответствующих элементов в мультимножестве B_M.

Симметрической разностью мультимножеств A_M и B_M называется мультимножество, состоящее из тех элементов мультимножеств A_M и B_M, кратности которых различны.

Универсум — некоторое мультимножество такое, что все остальные мультимножества являются подмультимножествами данного множества.

Дополнением мультимножества до универсума называется мультимножество, состоящее из элементов, кратность которых равна разности кратностей соответствующих элементов в универсуме и дополняемом мультимножестве.

Арифметическим произведением мультимножеств называется мультимножество, состоящее из элементов, которые одновременно присутствуют в каждом из мультимножеств, и их кратность равна произведению кратностей соответствующих элементов в перемножаемых мультимножествах.

Прямым произведением мультимножеств называется мультимножество, состоящее из всех упорядоченных пар элементов таких, что первый элемент каждой пары является элементом первого сомножителя, второй элемент пары — элементом второго сомножителя и кратность каждой пары равна произведению кратностей элементов и в перемножаемых мультимножествах.

Глава 8

Множество $\widetilde{A}$ называется *нечетким* или *расплывчатым множеством* в непустом множестве $X = \{x_1, x_2, \ldots, x_n\}$ и обозначается $\widetilde{A} = \{\langle \mu \sim A(x), x\rangle\}$, $x \in X$.

Функция принадлежности есть кортеж длины два, первая компонента которого обозначается $\mu \sim A(x)$ и задается на сегменте $[0, 1]$. Вторая компонента кортежа — элемент множества X.

Нечеткое высказывание — предложение, относительно которого можно судить о степени его истинности или ложности.

Индифферентное нечеткое высказывание равно 0,5, оно истинно в той же степени, что и ложно.

Отрицанием нечеткого высказывания $\widetilde{A}$ называется высказывание не $\widetilde{A}$, степень истинности которого определяется по формуле $\rceil\widetilde{A} = 1 - \widetilde{A}$.

Конъюнкцией нечетких высказываний $\widetilde{A}$, $\widetilde{B}$ называется высказывание, степень истинности которого определяется по формуле: $\widetilde{A} \wedge \widetilde{B} = \min(\widetilde{A} \wedge \widetilde{B})$.

Дизъюнкцией нечетких высказываний называется нечеткое высказывание, степень истинности которого определяется по формуле: $\widetilde{A} \vee \widetilde{B} = \max(\widetilde{A} \vee \widetilde{B})$.

Импликацией нечетких высказываний называется высказывание, степень истинности которого не меньше, чем степень неистинности ее посылки или степень истинности ее следствия: $\widetilde{A} \to \widetilde{B} = \max(1 - \widetilde{A}, \widetilde{B})$.

Эквивалентностью нечетких высказываний называется новое составное нечеткое высказывание, обозначаемое $\widetilde{A} \leftrightarrow \widetilde{B} = \min(\max(1 - \widetilde{A}, \widetilde{B}), \max(1 - \widetilde{B}, \widetilde{A}))$.

Степень включения нечеткого множества $\widetilde{A}$ в нечеткое множество $\widetilde{B}$ обозначается $\nu(\widetilde{A}, \widetilde{B})$ и определяется по формуле $\nu(\widetilde{A}, \widetilde{B}) = \wedge(\mu \sim A(x) \to \mu \sim B(x))$.

Объединением нечетких множеств называется выражение $\widetilde{A} \cup \widetilde{B} = \{\langle \mu \sim A \cup \sim B(x), x\rangle, x \in X\}$, $\mu \sim A \cup \sim B = \mu \sim A \vee \mu \sim B$.

Пересечение нечетких множеств называется выражение $\widetilde{A} \cap \widetilde{B} = \{\langle \mu \sim A \cap \sim B(x), x\rangle, x \in X\}$, $\mu \sim A \cap \sim B = \mu \sim A \wedge \mu \sim B$.

Разностью нечетких множеств называется выражение $\widetilde{A} \backslash \widetilde{B} = \{\langle \mu \sim A \setminus \sim B(x), x\rangle, x \in X\}$, $\mu \sim A \setminus \sim B = \mu \sim A \wedge \rceil\mu \sim B$.

Симметрической разностью нечетких множеств $\widetilde{A}$ и $\widetilde{B}$ называется выражение $\widetilde{A}\,\Theta\,\widetilde{B} = \{\langle\mu \sim A\,\Theta\, \sim B(x), x\rangle, x \in X\}$, $\mu \sim A\,\Theta\, \sim B = \mu \sim A \setminus \sim B \vee \rceil \mu \sim B \setminus \sim A$.

Нечетким графиком называется множество, состоящее из нечетких упорядоченных пар (кортежей).

Нечетким соответствием называют тройку множеств $\sim\Gamma = \langle\sim F, X, Y\rangle$, причем X и Y — это конечные четкие множества, а $\sim F$ — нечеткий график соответствия.

Инверсией нечеткого соответствия $\sim\Gamma = \langle\sim F, X, Y\rangle$ называется соответствие $\sim\Gamma^{-1} = \langle\sim F^{-1}, Y, X\rangle$, но X в инвертируемом соответствии будет областью прибытия, а Y — областью отправления.

Композицией двух нечетких соответствий $\sim\Gamma_1 = \langle F_1, X, Y\rangle$ и $\sim\Gamma_2 = \langle\sim F_2, Y, Z\rangle$ называется выражение $\sim\Gamma_1\cdot \sim\Gamma_2 = \langle\sim F_3, X, Z\rangle$.

Нечетким отношением на произвольном множестве X называется выражение $\sim\varphi = \langle\sim\Phi, X\rangle$, причем $\sim\Phi \subseteq x \times X$, $\sim\Phi \subseteq X^2$, где X — область задания отношения, а $\sim\Phi$ — нечеткий график отношения.

Приближенным множеством называется множество пар $\{Q_1, Q_2, \ldots, Q_l\}$. Упорядоченную пару $Q = \langle\varphi, X\rangle$ называют пространством приближений. При этом на обучающем множестве X введено отношение эквивалентности $\varphi = \langle\Phi, X\rangle$, $\Phi \subseteq X \times X$. Здесь Φ — график отношения, X — область задания отношения.

МОДУЛЬ 2.
ОСНОВЫ ТЕОРИИ АЛГОРИТМОВ (1,5 КРЕДИТА)

> Исследуя природу, естествоиспытатель должен следовать какой-то математической модели.
>
> *Галилей*

Комплексная цель и задачи изучения модуля

Цель модуля 2 — дать представление о фундаментальных понятиях, базовых принципах и законах таких важных разделов дискретной математики, как теория алгоритмов и алгебра логики, рассмотреть постановку основных задач и проблем, изучить вопросы методологии решаемой проблемы.

В результате освоения модуля 2 студент должен быть готов продемонстрировать следующие *компетенции* и *уровень подготовки*:

1) знание основных понятий теории алгоритмов;

2) умение оценивать и определять временную сложность алгоритмов;

3) знание основных алгоритмических моделей и умение их самостоятельно строить;

4) навыки решения практических задач оптимизации, умение представить решаемую конкретную практическую задачу в виде алгоритмической модели, пригодной для дальнейшего формального решения на основе существующих языков программирования средствами вычислительной техники.

Самостоятельная работа предусматривает проработку лекций (1,2 часа в неделю), тестирование, а также изучение литературы, формулировку цели работы, объекта и задач исследования, методов, источников и средств библиографического поиска, использованных для достижения поставленной цели.

Модуль включает в себя формулировку цели, проблемное изложение программного лекционного материала, тестовые вопросы для самоконтроля и список литературы. В процессе самостоятельного изучения представленных методических материалов происходит формирование указанных компетенций.

Глава 9
ВВЕДЕНИЕ В ТЕОРИЮ АЛГОРИТМОВ

Алгоритмы, определение и основные понятия алгоритмов, свойства алгоритмов, классификация алгоритмов

ЦЕЛИ

Освоив эту главу, учащиеся должны:
- знать определение и основные понятия алгоритмов;
- знать основные свойства алгоритмов;
- знать основные виды алгоритмов;
- уметь строить схемы алгоритмов.

9.1. Понятие алгоритма

Понятие «алгоритм» давно стало привычным для специалистов различных отраслей науки и техники. Оно является концептуальной основой разнообразных процессов обработки информации. Наличие соответствующих алгоритмов обеспечивает возможность автоматизации любой деятельности человека. Термин «алгоритм» обязан своим происхождением ученому средневекового Востока, который жил приблизительно с 783 по 850 г., чье имя — Мухаммад ибн Мусса аль-Хорезми, причем «аль-Хорезми», означает «происхождением из Хорезма». В латинских названиях, составленных в XVII веке, при изложении трудов аль-Хорезми его имя переводилось как алхоритм или алгоритм. Отсюда и пошло понятие «алгоритм», которое сначала использовалось для обозначения десятичной позиционной арифметики и алгоритмов цифровых вычислений (т. е. первых алгоритмических процедур, имеющих дело с символами), а затем для обозначения произвольных алгоритмов.

Дональд Эрвин Кнут в первой главе своей многотомной монографии «Искусство программирования» писал: «Понятие алгоритма является основным при составлении любого вида программ для ЭВМ... Современное значение слова “алгоритм” во многом аналогично таким понятиям, как рецепт, процесс, метод, способ, процедура, программа, но все-таки слово алгоритм имеет дополнительный смысловой оттенок. Алгоритм — это не просто набор конечного числа правил, задающих последовательность выполнения операций для решения задачи определенного типа. Помимо этого он имеет пять важных особенностей:

1) Конечность. Алгоритм всегда должен заканчиваться после выполнения конечного числа шагов.

2) Определенность. Каждый шаг алгоритма должен быть точно определен.

3) Ввод. Алгоритм имеет некоторое число входных данных, т. е. величин, которые задаются до начала его работы.

4) Вывод. У алгоритма есть одно или несколько выходных данных, т. е. величин, имеющих вполне определенную связь с входными данными.

5) Эффективность. Алгоритм считается эффективным, если все его операции можно точно выполнить в течение конечного промежутка времени».

Понятие алгоритма является не только центральным понятием теории алгоритмов, не только одним из главных понятий математики вообще, но и одним из главных понятий современной науки.

Однозначного и точного определения алгоритма не существует. В связи с этим приведем несколько наиболее распространенных определений понятия алгоритм.

Алгоритм (*algorithm*) — это формально описанная вычислительная процедура, получающая исходные данные, называемые также входом алгоритма или его аргументом, и выдающая результат вычисления на выход.

Алгоритм — это набор правил, указывающих определенные действия, в результате которых входные данные преобразуются в выходные. Последовательность действий в алгоритме называется *алгоритмическим процессом*, а каждое отдельное действие — *шагом алгоритма*.

Понятие *алгоритма* также можно интуитивно определить как конечную последовательность точно заданных правил решения произвольного класса задач.

Интуиция — это знание, полученное на основе опыта, но не подвергнутое научному анализу.

К алгоритмам не относятся запрещающие и разрешающие правила. Например: «Не курить! Не сорить!».

Особенность алгоритмов — это дискретный характер процесса, определяемого самим алгоритмом.

Алгоритмы строятся для решения тех или иных вычислительных задач. Формулировка задачи описывает, каким требованиям должно удовлетворять решение задачи, а алгоритм, решающий эту задачу, находит объект, удовлетворяющий этим требованиям. Алгоритм обычно имеет дело с тремя типами данных:

- входные (исходные) данные;
- промежуточные данные;
- выходные данные.

Алгоритмы, в которых решения поставленных задач сводятся к арифметическим действиям, называются *численными алгоритмами*.

В некоторых случаях алгоритмы представляют в виде виртуального устройства, называемого «черным ящиком» (рис. 9.1).

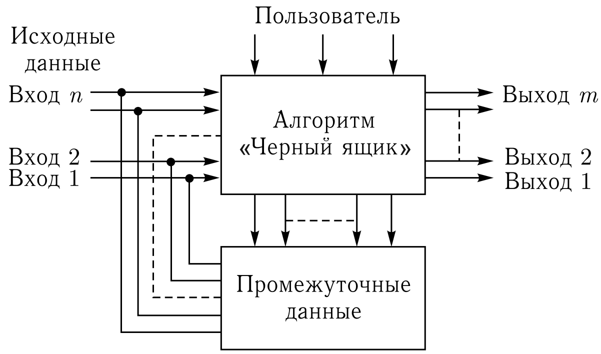


Рис. 9.1. Представление алгоритма

Особенность «черного ящика» в том, что он осуществляет переработку информации, но при этом неизвестно, каким способом. Другими словами, на одинаковый набор исходных данных «черный ящик» должен выдавать одинаковые выходные результаты.

Для размещения данных алгоритма требуется память. В теоретическом плане считают, что память состоит из одинаковых ячеек, причем каждая ячейка памяти может содержать один или несколько символов данных.

В существующих в настоящее время алгоритмах последовательность выполняемых элементарных шагов детерминирована, т. е. после каждого шага алгоритма указывается номер следующего шага. От каждого алгоритма требуется остановка после конечного числа шагов, дающих требуемый результат. Различают описание алгоритмов и процесс их реализации.

Алгоритм считают *правильным*, если на любом допустимом (для данной задачи) входе он заканчивает работу и выдает результат, удовлетворяющий требованиям задачи. В этом случае говорят, что алгоритм *решает* данную вычислительную задачу. *Неправильный алгоритм* может (для некоторого входа) не остановиться или дать неправильный результат.

Алгоритм может быть записан на любом естественном или формальном языке, а также в виде компьютерной программы или в машинных кодах. При этом важно, чтобы процедура вычислений была четко описана. Алгоритмы также могут записываться с помощью *псевдокода*, который представляет симбиоз из основных операторов различных языков программирования. Разница в том, что в псевдокоде могут использоваться не только эти операторы, но и словесное описание действий алгоритма, а также отсутствуют технологические подробности (например, процедура обработки ошибок), имеющиеся в реальной программе. Примером алгоритма может быть книга кулинарных рецептов, инструкция по эксплуатации прибора, стандарты проектирования космических кораблей и др.

9.2. Основные свойства алгоритмов

Основными свойствами алгоритмов являются массовость и результативность.

Массовость — это свойство алгоритма быть применимым для множества исходных данных. Термин «массовость» является расплывчатым, так как существуют алгоритмы, имеющие единственный вариант исходных данных.

Отметим, что для каждого алгоритма существует класс объектов, которые могут быть допустимы в качестве исходных данных. Тогда под массовостью понимают допустимость всех объектов этого класса, а не некоторого их числа.

Результативность — это свойство алгоритма, обеспечивающее получение результата через конечное число шагов. Считают, что алгоритм применим к допустимым исходным данным, если на его основе можно получить конечный результат. Если конечный результат получить нельзя, алгоритм к этим исходным данным неприменим. Кроме массовости и результативности, выделяют еще такие свойства алгоритма, как определенность, корректность и детерминированность.

Формулировка алгоритма должна быть точной и полностью определять все действия. Это свойство называется *определенностью* алгоритма.

Существуют два вида определений в алгоритмах. Первый вид определения называется прямым определением. В них новые понятия выражаются через одно или несколько известных. Второй вид называется «*порочный круг*». Это определения, в которых новое понятие выводится либо из самого себя, либо из другого понятия, которое было из него выведено.

Часто используют *рекурсивные* определения. Они состоят из двух частей: 1-я — циклическая, 2-я — прямая часть определения. Она является входом в циклическую.

Действие, выполняемое на каждом шаге алгоритма однозначно и независимо от исходных данных, называется *детерминированным*.

Если алгоритм создан для решения определенного класса задач, то необходима уверенность, что для всех допустимых исходных данных он даст правильные результаты. Это свойство называется *корректностью* алгоритма.

Для установления корректности алгоритмов используют три приема:

- конструирование новых алгоритмов путем комбинирования корректных алгоритмов;
- эквивалентные преобразования корректных алгоритмов;
- сужающие преобразования корректных алгоритмов.

Сформулируем определение алгоритма, характеризующее его с точки зрения вышеизложенного материала.

Алгоритм — это набор правил, сформулированный на некотором языке и определяющий процесс переработки допустимых исходных данных в конечные результаты, причем набор правил характеризуется корректностью, массовостью, определенностью и результативностью.

Каждый алгоритм обычно связан с двумя алфавитами (иногда их называют языками). На основании одного языка формулируется алгоритм, а предложения другого являются допустимыми для алгоритма вариантами исходных данных.

Первый язык называется *алгоритмическим*, а второй — языком *операндов*.

Операнды — это объекты, над которыми выполняются операции, предписываемые алгоритмами.

9.3. Классификация алгоритмов

Существует довольно большое число различных классификаций алгоритмов. Рассмотрим некоторые из них. В отдельные классы выделяют имитирующие и эмпирические алгоритмы.

Имитирующие алгоритмы создаются на основе описаний действий объектов, наблюдений, зависимости исходных данных от изменяющихся условий.

Эмпирические алгоритмы — это алгоритмы, основанные на опыте, изучении фактов и опирающиеся на непосредственное наблюдение и эксперимент.

Выделяют также *случайные* и *самоизменяющиеся* алгоритмы.

Алгоритм называется *случайным*, если в процессе его выполнения предусматривается возможность случайного выбора отдельных действий.

Алгоритм называется *самоизменяющимся*, если он не только перерабатывает входные слова, но и сам изменяется в процессе такой переработки.

Эвристические алгоритмы (или эвристики) — это алгоритмы теоретически не обоснованные, но позволяющие сократить количество переборов в пространстве поиска. Они характеризуются как алгоритмы:

- находящие обычно «хорошие», но не обязательно наилучшие (оптимальные) решения;
- имеющие более простую реализацию, чем любой известный алгоритм, гарантирующий оптимальное решение.

Понятие «хороший» меняется от задачи к задаче. Универсальной структуры описания эвристических алгоритмов не существует.

Например, во многих задачах информатики требуется выборка A предметов из множества B предметов. Один эвристический подход состоит в том, чтобы выбирать предметы с вероятностью A/B. Здесь каждый предмет принимается или отвергается в момент его просмотра и можно делать только один просмотр вдоль всего набора предметов. В общем виде эта процедура не гарантирует, что будет выбрано в точности A предметов. Другой эвристический подход при выборе A чисел из B состоит в том, чтобы многократно генерировать случайные целые числа до тех пор, пока не окажется A различных чисел.

Приведем другую классификацию. Согласно ей все алгоритмы можно разделить на три группы:

- линейные;
- разветвляющиеся;
- циклические.

Алгоритм называется *линейным*, если он предусматривает получение результата путем однократного выполнения одной и той же последовательности действий при каждом значении исходных данных.

Алгоритм называется *разветвляющимся*, если он предусматривает выбор одной из нескольких альтернатив последовательностей действий в зависимости от исходных и промежуточных данных. Каждая из этих последовательностей называется *ветвью* алгоритма.

Наиболее сложным является *циклический* алгоритм (по структуре), который выполняется путем многократного повторения некоторой последовательности действий. Циклические алгоритмы делятся на

- последовательные, с явно заданным числом повторений цикла;
- итерационные, которые имеют одинаковую структуру с последовательными, но выход из цикла осуществляется при выполнении общего условия, связанного с проверкой значения монотонности изменения в алгоритме исходной величины;
- поисковые (иерархические), когда исходная задача разбивается на некоторое число подзадач и делаются попытки решить каждую из них.

На рис. 9.2, *а–в* приведена условная структура последовательных, итерационных и поисковых алгоритмов.

Во многих случаях эффективное решение задачи получают при использовании *иерархического* алгоритма. При этом исходная задача Z_0 разбивается на некоторое число подзадач $Z_1, Z_2, \ldots, Z_k$ и делаются попытки решить каждую из них. Такое разбиение обычно описывается деревом поиска решения. Смысл разбиения на подзадачи состоит в том, что после разбиения за счет снижения объема конкретной подзадачи и возможного изменения ее структуры затраты на решение отдельной подзадачи значительно меньше, нежели на решение исходной задачи в целом. Важной проблемой при использовании иерархических алгоритмов является поиск оптимального решения, реализуемого на основе перебора путей на дереве решений. Выделяют следующие базовые виды поиска:

- *в глубину*;
- *в ширину*.

При *поиске в глубину* в каждой исследуемой вершине дерева решений выбирается один из возможных путей и исследуется до конца; при этом другие пути не рассматриваются, пока сохраняется возможность получить конкретный результат. Если такая возможность отсутствует, процесс поиска продолжается от бли-

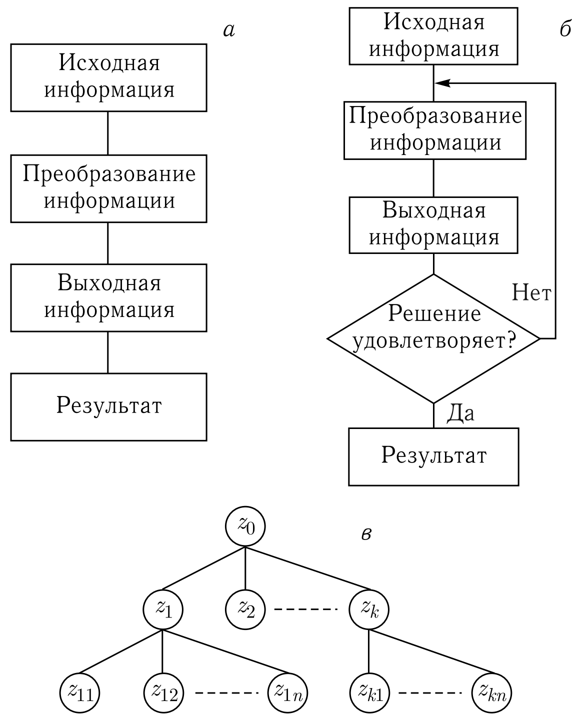


Рис. 9.2. Условная структура алгоритмов: *а*) последовательная; *б*) итерационная; *в*) иерархическая (поисковая)

жайшей вершины дерева решений, имеющей неисследованные пути.

Поиск в ширину предусматривает ветвление на дереве решений от уровня к уровню, причем все подзадачи уровня i исследуются раньше, чем любая подзадача уровня $(i-1)$.

Необходимо отметить, что, независимо от структуры поиска, многие задачи требуют значительного перебора вариантов, близкого к полному. Иногда удается заменить полный перебор перебором с отсечениями. Наиболее часто употребляется в этих целях метод ветвей и границ, когда полный перебор заменяется частичным за счет использования априорных оценок вариантов решения, позволяющих отбрасывать непродуктивные ветви дерева решений.

Выделяют следующие виды алгоритмов: логические; обработки данных; регулирования; эволюционные; оптимизации; решения задач вычислительной математики; проблемно-ориентированные; предметно-ориентированные; решения системных задач и др.

Примерами логических алгоритмов являются алгоритмы кодирования, декодирования, поиска элемента или пути и т. п. Эта разновидность алгоритмов представляет собой основу любого комплексного алгоритма.

Примерами алгоритмов обработки данных являются алгоритмы сортировки, упорядочения, перегруппировки и упаковки массивов и т. п. По своему характеру алгоритмы этой группы близки к логическим алгоритмам. Для обоих упомянутых типов алгоритмов характерны простота структуры и малое число операций.

Алгоритмы регулирования связаны с управлением технологическими процессами и применяются в задачах автоматического управления и регулирования технических систем.

Эволюционные алгоритмы моделируют процессы, происходящие в живой природе. Они основаны на сокращении перебора вариантов решения путем реализации генетических операторов. Эволюционные алгоритмы, как правило, не гарантируют нахождения оптимального варианта решения, но позволяют получать наборы решений, близкие к оптимальному и достаточные для решения практических задач.

Алгоритмы оптимизации связаны с решением оптимизационных проблем. При этом общая задача оптимизации может быть сформулирована как задача отыскания экстремума целевой функции $F(x)$ при заданных ограничениях $G(x)$ и граничных условиях $Q(x)$. Если $F(x)$, $G(x)$ и $Q(x)$ выражения произвольного вида, то универсальным алгоритмом решения оптимизационной задачи будет являться перебор всех вариантов решения и выбор наилучшего решения с учетом заданных ограничений.

Алгоритмы решения задач вычислительной математики представляют собой комплексные алгоритмы решения конечных уравнений вида $F(x) = 0$, однородных дифференциальных уравнений $\partial x/\partial t = f(x(t), t)$, линейных уравнений вида $Ax = B$ и т. д. Обычно эти алгоритмы оформляются в виде стандартных программ.

Проблемно-ориентированные алгоритмы связаны с решением задач общенаучной тематики (например, статистической обработки информации).

Предметно-ориентированные алгоритмы направлены на решение задач, относящихся к определенному виду предметной области (например, расчет математических моделей объектов проектирования).

Алгоритмы решения системных задач связаны с организацией работы операционной системы, диспетчеризации, управления программами и данными в ЭВМ.

Примеры решения задач

Для иллюстрации различных понятий и методов, связанных с понятием алгоритмов, рассмотрим задачу сортировки, которая имеет важное практическое значение. Имеется много разных алгоритмов сортировки. Она описывается так:

Вход: последовательность чисел $(a_1, a_2, a_3, \ldots, a_n)$.

Выход: перестановка $(a'_1, a'_2, a'_3, \ldots, a'_n)$ исходной последовательности, для которой $a'_1 \leqslant a'_2 \leqslant \ldots \leqslant a'_n$. Например, для входной последовательности (31, 59, 26, 58) алгоритм сортировки должен выдать (26, 31, 58, 59).

Алгоритмы сортировки часто используются в качестве промежуточного шага в других комплексных алгоритмах. В каждой конкретной ситуации выбор зависит от длины сортируемой последовательности, а также от типа имеющейся памяти (оперативная память, диски, флеш-память и т. п.).

Рассмотрим один из важных методов *сортировки вставками*, описанный Т. Корменом, Ч. Лейзерсоном, Р. Ривестом. Он эффективен при сортировке коротких числовых последовательностей. Метод основан на выборе очередного числа и вставке его в нужное место, сравнивая с имеющимися и передвигаясь справа налево. Исходными данными алгоритма является массив $A[0, \ldots, n]$ (последовательность элементов (чисел) длины n, подлежащая сортировке). Число элементов в массиве A обозначается через $\mathrm{length}\,[A]$. Последовательность сортируется без дополнительной памяти. Здесь используется входной массив и фиксированное число ячеек памяти.

На рис. 9.3 показана работа алгоритма сортировки Т. Кормена, Ч. Лейзерсона, Р. Ривеста на примере числовой последовательности $A = (4, 1, 3, 5, 0, 2)$.

Приведем описание алгоритма на верхнем уровне с пояснениями. Такое описание эффективно используется на практике при решении задач информатики и вычислительной техники и называется псевдокодом алгоритма.

Псевдокод алгоритма сортировки:

1. for $j \leftarrow 2$ to length $[A]$
2. do key $\leftarrow A[j]$
3. «добавить $A[j]$ к отсортированной части $A[0, \ldots, j-1]$»
4. $i \leftarrow j - 1$

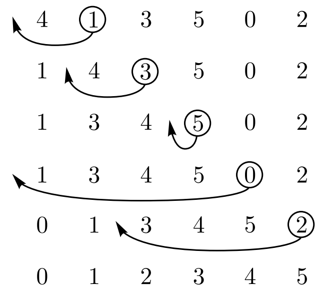


Рис. 9.3. Работа процедуры сортировки

5. while $i > 0$ and $A[i] >$ key
6. do $A[i+1] \leftarrow A[i]$
7. $i \leftarrow j - 1$
8. $A[i+1] \leftarrow$ key

Индекс j указывает выбранный очередной элемент. Часть массива $A[0, \ldots, j-1]$ составляют уже просмотренные элементы, а $-A[j+1, \ldots, n]$ — еще не просмотренные элементы. После выполнения процедуры сортировки массив A упорядочен по возрастанию.

В цикле for индекс j проходит массив слева направо. Просматривается элемент $A[j]$ (строка 2 алгоритма) и производится сдвиг стоящих перед ним больших по величине элементов (начиная с $(j-1)$-го) вправо, освобождая место для выбранного элемента (строки 4–7). В строке 8 элемент $A[j]$ помещается на свободное место.

Комментарий.

1. Отступ от левого поля указывает на уровень вложения. Например, тело цикла for (строка 1) состоит из строк 2–8, а тело цикла while (строка 5) содержит строки 6–7.

2. Циклы while, for, repeat и условные конструкции if–then–else имеют тот же смысл, что и в языке программирования «Паскаль».

3. Кавычки начинают и заканчивают комментарий (идущий до конца строки).

4. Одновременное присваивание $i \leftarrow j \leftarrow e$ (переменные i и j получают значение e) заменяет два присваивания $i \leftarrow e$ и $i \leftarrow j$ (в этом порядке).

5. Переменные (в данном случае i, j, key) локализуются внутри описанной процедуры (если не оговорено иное).

6. Индекс массива пишется в квадратных скобках, т. е. запись $A[v]$ соответствует v-му элементу в массиве A.

Контрольные вопросы

1. Откуда произошло понятие алгоритма?
2. Приведите пять основных особенностей алгоритма, данные Д. Кнутом.
3. Приведите основные определения алгоритма.
4. Что такое алгоритмический процесс, шаг алгоритма, интуиция?
5. Что является основной особенностью алгоритма?
6. Какие типы данных используются в алгоритме?
7. Приведите пример виртуального представления алгоритма в виде «черного ящика».
8. Что такое правильный и неправильный алгоритмы?
9. Перечислите способы записей алгоритмов.
10. Что такое массовость и результативность алгоритма?
11. Что такое определенность алгоритма?
12. Поясните понятие «порочный круг».
13. Что такое детерминированность и корректность алгоритма?
14. Что такое алгоритмический язык и язык операндов?
15. Что такое имитирующие и эмпирические алгоритмы? Приведите примеры.
16. Что такое случайные и самоизменяющиеся алгоритмы?
17. Дайте определение эвристических алгоритмов и приведите примеры.
18. Что такое линейные, разветвляющиеся и циклические алгоритмы?
19. Что такое поисковые алгоритмы? Приведите примеры алгоритмов поиска в глубину и в ширину.
20. Приведите примеры следующих видов алгоритмов: логических; обработки данных; регулирования; эволюционных; оптимизации; решения задач вычислительной математики; проблемно-ориентированных; предметно-ориентированных; решения системных задач ЭВМ.
21. В чем заключаются алгоритмы сортировки?
22. Приведите пример представления псевдокода алгоритма.

Задания для самостоятельной работы

1. Следуя образцу (рис. 9.3), покажите, как работает алгоритм сортировки вставками для входной числовой последовательности $A(101, 71, 49, 66, 81, 38)$.

2. Предложите другой алгоритм сортировки по убыванию (возрастанию).

3. Постройте алгоритм поиска в глубину для нахождения максимального числа в массиве.

4. Запишите псевдокод алгоритма поиска в ширину нахождения минимального числа в массиве $A = (2, 17, 33, 10, 22)$.

5. Приведите итерационный алгоритм нахождения максимального числа в массиве $B = (20, 10, 40, 60, 30)$.

6. Постройте псевдокод последовательного алгоритма сортировки одномерного массива $C = (55, 32, 20, 13, 48)$ по возрастанию.

7. Пусть задана матрица размера 3×4. Запишите псевдокод для нахождения суммы элементов главной диагонали.

8. Пусть задана матрица размера 3×4. Запишите псевдокод для нахождения суммы элементов первой строки и четвертого столбца.

Алгоритм — это набор правил, указывающих определенные действия, в результате которых входные данные преобразуются в выходные.

Все действия в живой и неживой природе можно описать с помощью алгоритмов.

Глава 10
УНИВЕРСАЛЬНЫЕ АЛГОРИТМИЧЕСКИЕ МОДЕЛИ

> Природа устроена рационально, а все явления протекают по точному и неукоснительному плану, который в конечном счете является математическим.
>
> *Аристотель*

Виды универсальных алгоритмов, абстрактный алфавит, числовая функция, машина Тьюринга, графическая схема алгоритма, структурная схема алгоритма, логическая схема алгоритма, алгоритмы Маркова, алгоритмы Ван-Хао, словесное описание алгоритма

ЦЕЛИ

Освоив эту главу, учащиеся должны:

- уметь строить алгоритмические модели;
- уметь составлять логическую схему алгоритма;
- уметь записывать графическую схему алгоритма;
- уметь составлять структурную схему алгоритма;
- уметь записывать словесную схему алгоритма;
- уметь выполнять преобразования на основе алгоритмов Маркова и Ван-Хао.

10.1. Преобразование слов в произвольных абстрактных алфавитах

Существуют три основных типа универсальных алгоритмических моделей:

- преобразование слов в произвольных абстрактных алфавитах;
- преобразование числовых функций;
- представление алгоритмов как некоторого детерминированного устройства, способного выполнять простейшие операции.

В первом типе универсальных алгоритмических моделей под алгоритмом понимают задаваемые соответствия между словами в *абстрактных алфавитах* (АА). Под абстрактным алфавитом понимается конечная совокупность объектов, называемых буквами или символами этого алфавита. *Символами* абстрактного алфавита считают буквы, цифры, любые знаки, слова, предложения, любые тексты. Следовательно, абстрактный алфавит — это конечное множество различных символов и, подобно множеству, его можно задать путем перечисления элементов или символов.

Например, A — абстрактный алфавит: $A = \{+, a, \text{студент}, b, c, \text{учебник}\}$.

Абстрактный алфавит многоступенчатый, т. е. каждый символ может сам являться алфавитом или его частью. Тогда в соответствии с определением первой алгоритмической модели *алгоритмами* называют преобразование слов в абстрактном алфавите. Соответственно *словом* в абстрактном алфавите называют любую конечную упорядоченную последовательность символов. В качестве примера приведем следующую запись:

Слово $A = \langle +, a\rangle$, Слово $B = \langle b, +, c\rangle$, Слово $C = \langle\ \ \rangle$, Слово $D = \langle +\rangle$. Здесь слово A состоит из двух символов, B — из трех символов, C не содержит символов и является пустым, D содержит один символ.

Число символов в слове абстрактного алфавита называется *длиной* этого слова.

Алфавитным оператором (АО) называют функцию, которая задает соответствие между словами входного и выходного АА.

Пусть задана система с входным абстрактным алфавитом X и выходным Y. Пусть $X = \{a, b, \text{ЭВМ}, \text{ПЭВМ}\}$; $Y = \{a + b, a \times b, \text{система}, \text{ВЦ}\}$. Построим АО, который перерабатывает входные слова алфавита X в выходные слова алфавита Y. На рис. 10.1 показан один из возможных вариантов алфавитного оператора.

Алфавитные операторы бывают *однозначные* и *многозначные*. В однозначном АО каждому входному слову ставится в соответствие одно выходное слово. В многозначных АО каждому входному слову может быть поставлено в соответствие некоторое множество выходных слов. АО, который не ставит в соответствие входному слову никакого выходного, называется неопределенным на этом слове. В примере на рис. 10.1 приведен один из возможных вариантов многозначного АО.

Множество слов, на котором АО определен, называется его *областью определения*.

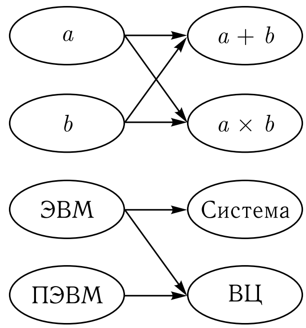


Рис. 10.1. Пример алфавитного оператора

Выделяют алфавитные операторы с конечной и бесконечной областями определения. Если область определения АО конечна, то его обычно задают таблицей соответствий. В левой части таблицы записывают слова, входящие в область определения, в правую часть таблицы — слова, получающиеся в результате применения АО к входным словам. Если область определения алфавитного оператора бесконечна, то его задают набором правил, которые за конечное число шагов позволяют найти выходное слово, соответствующее входному.

Алфавитный оператор, задаваемый с помощью конечной системы правил, называется *алгоритмом*.

Следует различать понятие алгоритма и алфавитного оператора. В АО основное — это соответствие между словами, а не способ его установления. В алгоритме же основное — способ установления соответствия между словами. Следовательно, *алгоритм* — это АО с правилами, определяющими его действие.

Два алгоритма называются *равными*, если равны их АО и совпадают системы правил, задающих действие этих алгоритмов на входные слова.

Алгоритмы считаются *эквивалентными*, если они имеют совпадающие АО, но не совпадающие способы их задания.

Например, пусть алгоритм А1 определяется следующим образом:

$X = \{a, b, c\}$ — входной абстрактный алфавит;

$Y = \{\alpha, \beta, \gamma\}$ — выходной абстрактный алфавит;

Алфавитный оператор (АО1): $a \to \alpha$, $b \to \beta$, $c \to \gamma$.

Система правил, задающая действие алфавитного оператора: A, B, C.

Пусть алгоритм А2 определяется следующим образом:

$X = \{a, b, c\}$ — входной абстрактный алфавит;

$Y = \{\alpha, \beta, \gamma\}$ — выходной абстрактный алфавит;

Алфавитный оператор (АО2): $a \to \alpha$, $b \to \beta$, $c \to \gamma$.

Система правил, задающая действие алфавитного оператора: A, B, C.

Тогда алгоритм А1 равен алгоритму А2. Если в алгоритме А2 система правил, задающая действие алфавитного оператора АО2, B, C, A, то алгоритм А1 эквивалентен алгоритму А2.

10.2. Числовые функции

Из определения алгоритма следует, что он ставит в соответствие исходным данным выходные. Поэтому каждому алгоритму можно поставить в соответствие функцию, которую он вычисляет. Рассмотрим теперь второй тип алгоритмических моделей. Их называют *числовые функции*.

Функцию называют вычислимой, если имеется способ получения ее значений. В алгоритмах часто используются рекурсивные функции, представляющие собой частичный класс вычислимых функций.

Процедуру, которая прямо или косвенно обращается к себе в процессе выполнения алгоритма, называют *рекурсивной*.

Рекурсия — это способ задания функций, когда ее значения для произвольных значений аргументов выражаются известным образом через значения определяемой функции для меньших значений аргументов.

Алгоритмы, определяющие способы задания рекурсивных функций, часто называют *алгоритмами, сопутствующими рекурсивным функциям*. Выделяют три способа построения рекурсивных функций: на основе оператора суперпозиции, оператора рекурсии и оператора построения по первому нулю.

Функции, полученные без использования оператора построения по первому нулю, называются *примитивно рекурсивными*.

Рекурсивные функции, определенные не для всех возможных значений аргумента, называются *частично рекурсивными*.

Рассмотрим рекурсивную функцию (РФ).

В ней различают:

1) имя, т. е. ее обозначение;

2) запись ($W = f(x_1, x_2, x_3)$ или $F = f(x_1, \ldots, x_n)$);

3) значение функции.

Значение функции отвечает конкретным значениям аргумента. Например, в функции $f(x_1, x_2, x_3)$ аргументы x_1, x_2, x_3 имеют следующие значения: $x_1 = 1$, $x_2 = 3$, $x_3 = 100$. Эту же функцию можно эквивалентно представить в следующем виде: $f(1, 3, 100)$.

Рекурсивные функции бывают простые и сложные. Простейшие РФ называют базовыми. Представляющие эти РФ алгоритмы называются одношаговыми. Из базовых РФ на основе операторов можно получать любые составные РФ. Выделяют три типа базовых рекурсивных функций.

1. Функции любого числа независимых переменных, тождественно равные нулю. Их обозначают f_0, φ_0.

Если функция записывается в виде φ_0 или f_0, то любой совокупности значений аргументов данной функции ставится в соответствие значение ноль. Например, $\varphi_0(0) = 0$, $\varphi_0(3,5,7) = 0$, $\varphi_0(4,17) = 0$.

2. Тождественные функции любого числа независимых переменных вида $\varphi_{n,i}$, причем $1 \leqslant i \leqslant n$.

Представляющий эту функцию алгоритм запишется: если функциональный знак имеет вид $\varphi_{n,i}$, то значением функции следует считать значение (слева направо) i-го независимого переменного. Пример, $W = \varphi_{3,2}(X,Y,Z) = Y$, $\varphi_{3,3}(3,4,8) = 8$. $\varphi_{3,4}(3,4,8) = 0$ — эта запись не имеет смысла, так как $n = 3$, $i = 4$ и не выполняется условие $1 \leqslant i \leqslant n$. Для тождественных функций $n, i \neq 0$.

3. Функции следования. Представляющий эти функции алгоритм запишется: если функциональный знак имеет вид λ, то значением функции следует считать число, непосредственно следующее за числом, являющимся значением алгоритма. Например, операцию получения значения функции следования обозначают «$'$» т. е. $0' = 1$, $1' = 2 \ldots \lambda'(5) = 6$.

Отметим, что базовые функции имеют значения, если заданы значения их аргументов.

Рассмотрим оператор суперпозиции или подстановки. Оператор по функции F от n аргументов и по функциям $f_1, f_2, \ldots, f_n$ строит новую функцию Φ так, что справедливо выражение:

$$\Phi \equiv F(f_1, f_2, \ldots, f_n).$$

Алгоритм, представляющий данный оператор, запишется: значения функций $f_1, f_2, \ldots, f_n$ принять за значения аргументов функции F и вычислить ее значение.

Пример. Если $f_1 = \lambda(y)$, а $F = \lambda(x)$ и $x = \lambda(y)$, то на основе оператора суперпозиции получится новая функция $r(y) = \lambda(\lambda(y)) = \lambda(y') = y''$. Тогда получили новую функцию $r(y) \equiv \Phi$.

Иногда в алгоритмах вводится знак «::=», что означает «по определению» или «присвоить». Применение этого знака реализует оператор подстановки, который обозначается S.

Например, построение функции Φ из функций F и f_i с помощью оператора суперпозиции можно записать в следующем виде:

$$\Phi ::= S[F; f_1, f_2, \ldots, f_n].$$

Для $r(y) ::= S[\lambda(x), \lambda(y)]$.

Рассмотрим оператор рекурсии. Он реализуется на основе двух функций, одна из которых имеет $(n-1)$ независимую переменную, а другая, кроме указанных, еще две, т. е. $(n+1)$.

При этом один из дополнительных аргументов войдет вместе с аргументами первой функции в число аргументов вновь получаемой, а другой, дополнительный, — при выполнении оператора рекурсии.

Оператор рекурсии обозначается R. Его применение записывают в виде следующей строки:

$f ::= R[f_1, f_2, x(y)]$, где f — получаемая функция;

f_1 — функция $(n-1)$ независимых переменных;

f_2 — функция $(n+1)$ независимых переменных;

x — главный из дополнительных аргументов;

y — вспомогательный аргумент.

Оператор рекурсии задает функцию с помощью двух условий, в которые входят функции f_1 и f_2.

Например, построение функции предшествования:

$$f(0) = f_1,\ f(i') = f_2(i, f(i)),$$

$$\text{пр}(0) = 0,\ \text{пр}(1) = 0,\ \text{пр}(2) = 1,\ \text{пр}(3) = 2\ldots$$

Здесь пр — это обозначение функции предшествования.

Рассмотрим оператор построения функции по первому нулю. Он по заданной функции, зависящей от $(n+1)$ аргументов, строит новую функцию от n аргументов. Здесь вспомогательный аргумент исчезает. Например, для функции вида $f ::= \mu[f_1(x)]$ x — исчезающий аргумент.

Тогда алгоритм запишется: придавать вспомогательному аргументу последовательные значения с нуля до тех пор, пока не окажется, что функция f_1 не стала равна нулю (в первый раз). Полученное значение вспомогательного аргумента принять за значение определяемой функции, соответствующее тем значениям основных аргументов, при которых осуществляется описанный процесс.

С помощью перечисленных операторов и функций строится любая сложная функция, которая может представить исследуемый алгоритм.

Сформулируем *тезис Чёрча*. Какова бы ни была вычислимая неотрицательная целочисленная функция от неотрицательных целочисленных аргументов, всегда существует тождественно равная ей рекурсивная функция.

Тогда *тезис Чёрча* можно интерпретировать для анализа алгоритмов следующим образом: каков бы ни был алгоритм, перерабатывающий наборы целых неотрицательных чисел, всегда существует алгоритм, представляющий РФ, который ему эквивалентен.

Тогда реализация алгоритма в определенном смысле эквивалентна вычислению значений РФ. При этом невозможность построения РФ будет означать невозможность построения алгоритма.

10.3. Построение алгоритмов по принципу «разделяй и властвуй»

Существует много стандартных приемов, используемых при построении алгоритмов. Например, сортировка вставками является одним из таких стандартных приемов. При этом алгоритм действует по шагам и происходит последовательное добавление элементов одного за другим к отсортированной части массива. При решении сложных задач большой размерности их часто разбивают на части, определяют решения, а затем из них получают решение всей задачи. Этот прием, если принять его рекурсивно, как правило, приводит к эффективному решению основной задачи, подзадачи которой составляют ее меньшие версии. Этот прием называется принцип «разделяй и властвуй». На основе принципа «разделяй и властвуй» можно построить более быстрый алгоритм сортировки.

Отметим, что многие алгоритмы по природе своей рекурсивны, т. е. решая некоторую задачу, они обращаются сами к себе для решения ее подзадач. Идея метода «*разделяй и властвуй*» состоит как раз в этом. Сначала решаемая задача разбивается на задачи меньшего размера. Затем эти задачи решаются (с помощью рекурсивного вызова или непосредственно, если размер задачи достаточно мал). Наконец, полученные решения объединяются и получается решение исходной задачи.

Для задачи сортировки процесс решения выглядит так. Сначала массив разбивается на две половины, каждая из которых сортируется отдельно. После чего два упорядоченных массива соединяют в один. Рекурсивное разбиение задачи происходит до тех пор, пока размер массива не станет равным единице (любой массив длины 1 можно считать упорядоченным).

Нетривиальным этапом является объединение двух упорядоченных массивов в один. Оно выполняется с помощью различных вспомогательных процедур.

10.4. Представление алгоритма в виде детерминированного устройства

Математик А. Тьюринг — один из основателей анализа сложности алгоритма. Он разработал идеи представления алгоритмов на основе детерминированных устройств. Такое устройство называется *машиной Тьюринга* (МТ). Модель машины Тьюринга является третьей алгоритмической моделью. Она состоит из трех частей — ленты, считывающего устройства (головки) и управляющего устройства (УУ) (рис. 10.2).

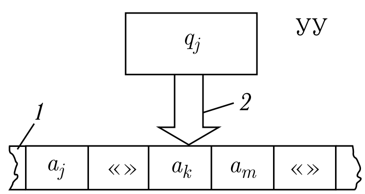


Рис. 10.2. Представление машины Тьюринга

На рис. 10.2 цифра *1* обозначает ленту МТ. Она представляет собой бесконечную память машины. В качестве ленты можно использовать любой носитель памяти (бумажную или магнитную ленту). Отметим, что лента считается бесконечной длины в обе стороны. Она разбивается на стандартные ячейки одинакового размера. В каждой ячейке может быть записан только один любой символ конечного алфавита. Число возможных символов конечно, и они образуют абстрактный алфавит машины $A = \{a_1, a_2, \ldots, a_m\}$. Отсутствие символа в ячейке обозначается специальным «пустым» символом «».

МТ имеет считывающую головку (цифра *2* на рис. 10.2), способную считывать содержимое ячеек. Головка всегда располагается над одной из ячеек ленты. Она может читать и писать символы, стирать их. Лента перемещается вдоль головки в обе стороны таким образом, чтобы в любой момент считывать содержимое одной ячейки ленты. На каждом шаге работы машины головка находится в одном из множества состояний

$Q = \{q_1, \dots, q_n\}$, среди которых выделяют начальное q_1 и конечное q_z состояния.

Элементарный шаг машины Тьюринга состоит из следующих действий:

• головка считывает символ, записанный в ячейке, над которой она находится;

• считанный символ a_k и текущее состояние головки q_j однозначно определяют новое состояние q_i, новый записываемый символ a_l и перемещение головки d_p (на одну ячейку влево, вправо или остаться на месте).

Кроме того, МТ имеет устройство управления (УУ), которое в каждый момент находится в одном из состояний $Q = \{q_{1,2}, \dots, q_n\}$. Состояние устройства управления называется внутренним состоянием МТ. Одно из состояний является заключительным состоянием МТ и управляет окончанием работы машины. Устройство управления хранит и выполняет команды машины, например, вида $q_j a_k \to q_i a_l d_p$.

Данные в машине Тьюринга — это слова в абстрактном алфавите ленты. На ленте МТ записывают исходные данные и конечные результаты.

Полное состояние машины Тьюринга называется конфигурацией и включает в себя состояние головки, состояние ленты (слово, записанное за ней) и положение головки на ленте. Конфигурация машины Тьюринга описывается тройкой $a_1\, q\, a_2$, где q — текущее состояние головки, a_1 — слово слева, а a_2 — слово справа от головки (включая символ, над которым находится головка). Конфигурация, изображенная на рис. 10.2, может быть записана так: a_j «» $q_j\, a_k\, a_m$. Конкретную машину Тьюринга (и алгоритм соответственно) можно задать, перечислив элементы алфавита A, множества состояний Q и команды машины.

Перед началом работы на пустую ленту записывается исходное слово a абстрактного алфавита. Головка устанавливается над первым его символом и, как следствие, начальной конфигурацией является запись вида $q_1\, a$.

Работа машины Тьюринга состоит в изменении конфигураций. *Конфигурацией* машины Тьюринга называется ее полное состояние, по которому можно однозначно определить дальнейшее поведение машины. Оно обозначается тройкой $\alpha_1\, q_i\, \alpha_2$, где α_1 — слово слева от головки, α_2 — слово, состоящее из символа, просматриваемого головкой и расположенного справа. Стандартной начальной конфигурацией называется конфигурация вида $q_1\, \alpha$, где просматривается крайний слева символ, записанный на ленте. Стандартной заключительной конфигурацией называется выражение вида $q_2\,\alpha$. Ко всякой произвольной конфигурации k в машине Тьюринга применима только одна команда типа $q_i\,a_j \to q_i'\,a_j'\,s_k$, которая переводит конфигурацию k в k'.

Совокупность всех команд, которые может выполнять машина Тьюринга, называется *программой*. При решении задач с входными данными сопоставляется начальная конфигурация k_0, а выходные данные определяются заключительной конфигурацией, в которую машина Тьюринга переводит k_0.

Например, рассмотрим машину Тьюринга, переводящую последовательность $a_1, a_2, \ldots, a_n$ в последовательность $b_1, b_2, \ldots, b_n$, работающую в соответствии с программой $q_i\,a_j \to q_i'\,a_j'\,s_k$. Считывающая головка, перемещая ленту направо в соответствии с программой, будет стирать символы $a_1, a_2, \ldots, a_n$ и вместо них соответственно записывать выходные символы $b_1, b_2, \ldots, b_n$. Если потребовать, чтобы при просмотре клетки ленты с пустым символом машина Тьюринга переходила в заключительное состояние, то после остановки машины на ленте будет выходная последовательность $b_1, b_2, \ldots, b_n$.

Известно, что на машине Тьюринга можно имитировать все алгоритмические процессы, которые можно представить в математическом виде.

А. Тьюринг показал, что если проблемы не могут быть решены на его машине, то они не могут быть решены ни на какой другой автоматической ЭВМ, т. е. это проблемы, для которых алгоритмы не могут быть составлены даже в принципе. Следовательно, невозможность построения машины Тьюринга означает невозможность построения алгоритма решения данной проблемы. Следует отметить, что речь идет об отсутствии алгоритма, решающего всю данную проблему, что не исключает возможности решения этой проблемы в частных случаях различными для каждого случая методами.

10.5. Универсальные схемы алгоритмов

Все операции, выполняемые в алгоритмах, разделяют на две группы: арифметические операции и логические операции. Арифметические операции выполняют непосредственное преобразование информации. Логические операции определяют последовательность выполнения арифметических операций.

Рассмотрим основные универсальные схемы алгоритмов. К ним относят:

- алгоритмы Ван-Хао;
- алгоритмы Маркова;

- алгоритмы Ляпунова;
- графические схемы алгоритмов (ГСА);
- структурные схемы алгоритмов (ССА);
- словесные схемы алгоритмов (СА).

Универсальные схемы алгоритмов Ван-Хао появились в 50-х годах прошлого столетия. *Операторный алгоритм Ван-Хао* задается последовательностью приказов специального вида. Каждый приказ имеет определенный номер и указание, какую операцию необходимо выполнить над заданным объектом и с каким номером следует далее выполнять действие над результатом этой операции. Например, некоторый приказ записывается следующим образом (рис. 10.3):

i:	ω	α	β

Рис. 10.3. Пример приказа

На рис. 10.3 i — номер приказа, ω — элементарная операция над заданным объектом, α и β — номера произвольных других приказов. Выполнить приказ i над объектом или числом x означает вычислить $\omega(x)$ и затем перейти к выполнению над $\omega(x)$ приказа с номером α, если $\omega(x)$ не определяется, то перейти к выполнению над числом x приказа с номером β. Заключительному состоянию в алгоритме Ван-Хао соответствует приказ, представленный на рис. 10.4:

i:	Стоп

Рис. 10.4. Приказ на окончание работы алгоритма Ван-Хао

Такая запись означает, что вычисления следует прекратить.

В настоящее время схемы алгоритмов Ван-Хао не имеют широкого практического применения, а используются для теоретического представления алгоритмов.

Интерес в практической деятельности представляют нормальные алгоритмы Маркова (НАМ). Они основаны на использовании абстрактных алфавитов и марковских подстановок. В качестве исходных данных и конечных результатов нормальные алгоритмы Маркова (НАМ) используют строки символов, т. е. слова абстрактных алфавитов.

Марковской подстановкой называют операцию над словами, задаваемую парой слов $\langle P, Q\rangle$, где P и Q — слова в абстрактном алфавите. Суть подстановки заключается в следующем. В исходном слове C определяется первое вхождение слова P. Если оно есть, то производится замена этого вхождения словом Q без изменения остальных частей слова C. При этом слова P и Q должны иметь одинаковую размерность.

Пример 10.1. Пусть задано слово $C = 832547$ и задана пара $\langle P, Q\rangle = \langle 32, 00\rangle$. Тогда результатом работы нормального алгоритма Маркова будет слово $C' = 800547$.

Пример 10.2. Пусть задано слово $C = 832547$ и задана пара $\langle P, Q\rangle = \langle 9, 4\rangle$. Отметим, что такая марковская подстановка к слову C неприменима, так как в нем отсутствует цифра 9.

Нормальные алгоритмы Маркова применяют для анализа и преобразования формульных записей. Тогда НАМ работает следующим образом. Двигаясь по столбцу формул, определяют первую формулу, левая часть которой входит в преобразуемое слово; если такой формулы нет, то процесс окончен. При ее наличии выполняется марковская подстановка, изменяющая преобразуемое слово. После этого проверяется, заключительная ли это подстановка или нет. Если да, то процесс окончен, если нет, то весь процесс повторяется с самого начала.

Например, пусть заданы следующие выражения:

$$\begin{gathered} 1234567; \\ 9713256; \\ 0001111; \\ 2224444 \end{gathered}$$

и задана пара $\langle P, Q\rangle = \langle 2, B\rangle$.

Тогда результат работы НАМ запишется таким образом:

$$\begin{gathered} 1B34567; \\ 9713B56; \\ BBB4444. \end{gathered}$$

Нормальные алгоритмы Маркова в основном применяются для преобразования, минимизации и доказательства алгебраических тождеств.

Рассмотрим схемы алгоритмов Ляпунова. Для описания алгоритмов Ляпунова используют *логические схемы алгоритмов* (ЛСА). Логическими схемами алгоритмов называют выражения, составленные на основе абстрактного алфавита и состоящие из последовательной цепочки операторов и логических условий. В ЛСА операторами являются действия алгоритма, в результате которых происходит обработка и преобразование информации. Операторы обозначают строчными буквами. Прописными буквами в ЛСА обозначают проверяемые логические условия. Последовательность выполнения нескольких операторов представляется в виде их произведения. Операторы в таком предложении выполняются слева направо. После каждого логического условия ставится стрелка вверх (назовем ее условно восходящей), которая обозначает собой переход по условию. Каждая стрелка имеет свой собственный номер. Каждой восходящей стрелке соответствует одна нисходящая (т. е. стрелка вниз) стрелка с тем же номером, что и у восходящей. Таким образом, если заданное логическое условие справедливо, то выполняется следующий оператор, расположенный справа от условного. Если условие ложно, то выполняется оператор, расположенный справа от нисходящей стрелки, имеющей тот же номер.

Приведем пример записи ЛСА:

$$A_0 \downarrow^1 A_1\, q \uparrow^1 \downarrow^2 A_2\, p \uparrow^2 A_k.$$

Согласно этому выражению работа алгоритма начинается с выполнения левого оператора. Обычно это оператор A_0, представляющий собой начало алгоритма. Затем выполняется следующий справа оператор A_1. Следующий за оператором A_1 оператор q является логическим условием. Поэтому в зависимости от результатов проверки истинности этого условия будут выполняться либо оператор A_2, если условие q истинно, либо оператор A_1 (по нисходящей стрелке). Аналогичные действия выполняются и для другого логического условия p. Последний в алгоритме — крайний правый оператор, обозначаемый обычно A_k. Он называется оператором окончания алгоритма.

В общем случае ЛСА Ляпунова должны удовлетворять следующим условиям:

1) содержать один начальный и один конечный оператор, соответственно A_0 и A_k;

2) каждому логическому условию должна соответствовать одна стрелка, направленная вверх;

3) каждому логическому условию всегда соответствует одна стрелка, направленная вниз.

4) каждой восходящей (нисходящей) стрелке обязательно соответствует одна нисходящая (восходящая) стрелка.

Пример 10.3. Пусть задана последовательность из n натуральных целых чисел. Составить логическую схему алгоритма нахождения среднего арифметического этих чисел. Тогда данный алгоритм можем записать в виде логической схемы следующего вида:

$$A_0 \downarrow^1 A_1\, q \uparrow^1 A_2\, A_3 \downarrow^2 A_4\, A_5\, p \uparrow^2 A_6\, A_7\, A_k,$$

A_0 – начало алгоритма;

A_1 — ввод исходной последовательности чисел;

q — проверка логического условия: все ли числа введены;

A_2 — задание начального значения для подсчета суммы чисел: $S = 0$;

A_3 — ввод начального значения переменной цикла: $i = 1$;

A_4 — суммирование значений чисел последовательности: $S = S + x_i$;

A_5 — увеличение переменной цикла: $i = i + 1$;

p — проверка логического условия выхода из цикла: $i > n$;

A_6 — нахождение среднего арифметического последовательности чисел: $\Delta = S/n$;

A_7 — вывод найденного значения;

A_k — конец работы алгоритма.

Рассмотрим *графические схемы алгоритмов* (ГСА). Они представляют собой графическую интерпретацию алгоритмов Ляпунова. Приведем основные правила представления графических схем алгоритмов:

- наличие конечного числа вершин;
- наличие одной начальной и одной конечной вершин;
- входы и выходы вершин соединяются между собой с помощью стрелок, направленных от входа к выходу;
- каждый выход соединяется только с одним входом, каждый вход соединяется, по крайней мере, с одним выходом;
- логическое условие имеет некоторое множество входов, но только два выхода («да», «нет»).

Следовательно, оператор начала A_0 может иметь только выходящие стрелки (рис. 10.5).

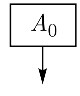


Рис. 10.5. Оператор начала ГСА

Оператор преобразования может иметь входящие и выходящие стрелки (рис. 10.6).

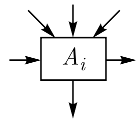


Рис. 10.6. Оператор преобразования ГСА

Логическое условие может иметь много входящих стрелок, выходящих стрелок может быть только две (рис. 10.7)

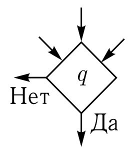


Рис. 10.7. Оператор логического условия

Оператор конца алгоритма имеет только входящие стрелки (рис. 10.8).

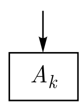


Рис. 10.8. Оператор окончания ГСА

Пример 10.4. Пусть задана логическая схема алгоритма следующего вида:

$$A_0 \downarrow^1 A_1\, p_1 \uparrow^1 A_2\, p_2 \uparrow^2 A_3 \downarrow^2 A_4\, A_5\, A_k.$$

На основе приведенных правил построим эту схему в виде ГСА (рис. 10.9).

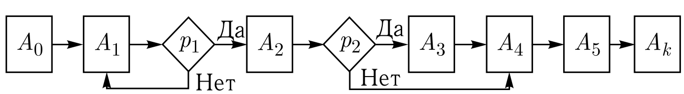


Рис. 10.9. Графическая схема алгоритма (ГСА)

Важной разновидностью графического представления алгоритмов являются *структурные схемы алгоритмов* (ССА). При такой записи алгоритмов происходит как бы процесс декомпозиции (т. е. расчленения алгоритма на отдельные этапы — блоки). Блоки изображаются графически в виде геометрических фигур. Каждому блоку присваивается собственный номер. Внутри блока в виде формул или словесно записывается информация, характеризующая выполняемые им функции. Блоки соединяются линиями, символизирующими межблочные связи. Блоки могут описывать элементарные шаги алгоритма или представлять собой части алгоритмов или отдельные алгоритмы.

Правила выполнения структурных схем алгоритмов определены ГОСТом 19.701-90 (ИСО 5807-85) «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». Фрагмент текста ГОСТа приведен в приложении. Руководствуясь требованиями данного ГОСТа, можно построить структурную схему алгоритма для решения конкретной задачи.

Пример 10.5. Пусть необходимо построить ССА нахождения минимального числа в заданном множестве чисел.

$$A = \{a_1, a_2, a_3, a_4\}; \quad |A| = 4 = n,$$
$$a_1 = 2, \quad a_2 = 1, \quad a_3 = 8, \quad a_4 = 5.$$

Перед разработкой ССА производится распределение памяти, причем это распределение условное.

Блок «Начало» соответствует началу работы алгоритма.

Блок 1 — ввод исходных данных.

Блок 2 задает начальное значение переменной i.

Блок 3 присваивает переменной x значение a_i, причем на первом шаге значение a_i равно a_1.

Блок 4 сравнивает значение переменной i с n. Это необходимо для определения момента окончания алгоритма.

Блок 5 увеличивает значение переменной i на единицу.

Блок 6 сравнивает предыдущее число a_i с последующим a_{i+1}.

Блок 7 печатает значение найденного минимального элемента $a_2 = 1$.

Блок «Конец» соответствует окончанию работы алгоритма.

На рис. 10.10 приведена структурная схема алгоритма.

Пример 10.6. Составить алгоритм определения среднего роста студента учебной группы.

Решение.

1. Область допустимых значений исходных данных определена в самом условии задачи и предполагает, что все обрабатываемые значения могут быть только положительными числами (целыми или дробными, в зависимости от принятой единицы измерения).

2. Математическая модель задачи может быть выражена соотношением

$$L = \left(\sum_{i=1}^{n} l_i\right)/n,$$

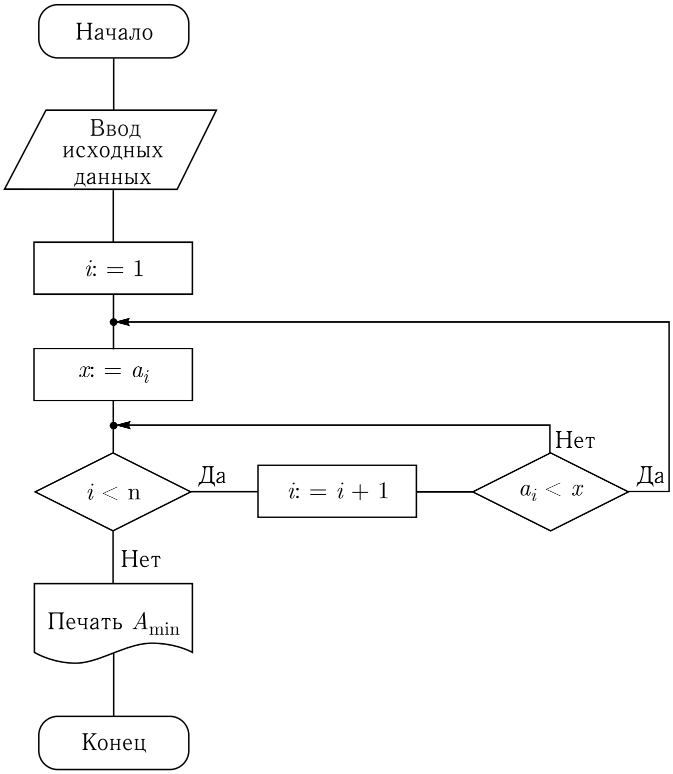


Рис. 10.10. Пример структурной схемы алгоритма

где l_i — рост i-го студента, n — число студентов в учебной группе.

3. Составляем структурную схему алгоритма. Вариант структурной схемы алгоритма для решения этой задачи показан на рис. 10.11.

Следует отметить, что основа структуры алгоритма представляет собой ограниченный набор стандартных способов соединения базовых блоков для выполнения типичных последовательностей действий. Следовательно, любой алгоритм может быть построен на использовании нескольких базовых блоков путем их комбинации друг с другом на основе связок. Такой подход к разработке алгоритмов называется *структурным*. К связкам относят:

• следование — последовательное размещение блоков или их групп;

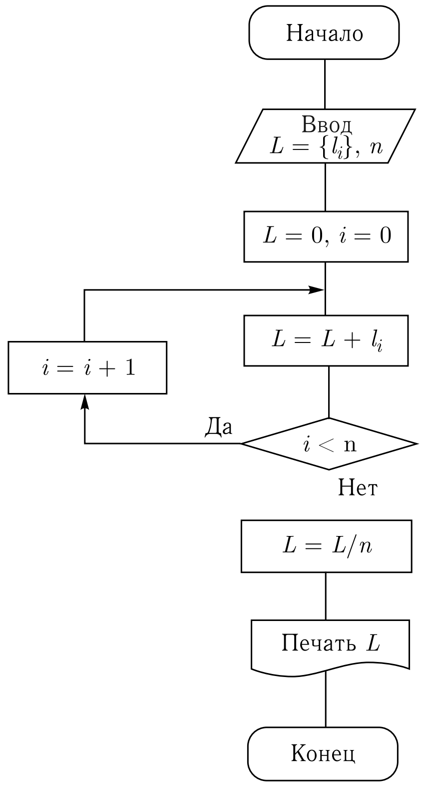


Рис. 10.11. Вариант структурной схемы алгоритма к примеру 10.6

• цикл — структура, определяющая выполнение одного и того же действия несколько раз с различными параметрами вычисляемой величины. Циклическая структура предполагает наличие следующих блоков:

а) *присваивание*, где происходит задание начальных значений параметров вычисляемой величины;

б) *тело цикла*, представляющего собой последовательность многократно выполняемых действий;

в) *условия выполнение цикла*, определяющего выполнение действий в теле цикла или прекращение их выполнения.

Различают два типа циклических структур — цикл «с предусловием» (цикл «до») и цикл «с постусловием» (цикл «пока»). Цикл «до» (рис. 10.12, *а*) позволяет выполнять заданные действия (тело цикла) несколько раз, до тех пор, пока выполняется некоторое условие. Особенность преобразования информации здесь состоит в том, что цикл «до» выполняется всегда (по крайней мере, один раз), так как проверка условия выхода из цикла осуществляется только после выполнения действий. Цикл «пока» (рис. 10.12, *б*) позволяет выполнить заданные действия только после проверки условия. В этом случае цикл «пока», в отличие от цикла «до», может быть не выполнен ни разу;

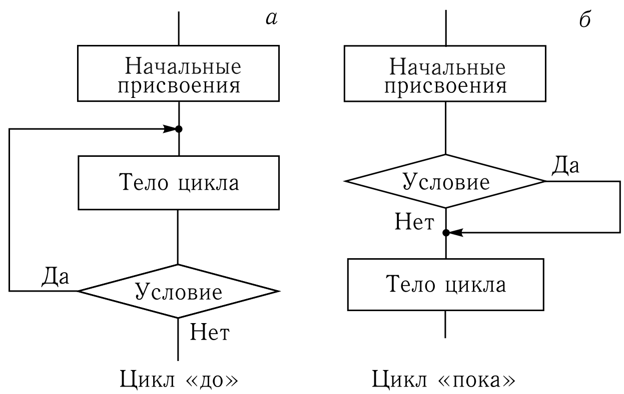


Рис. 10.12. Типы циклических структур

• разветвление (рис. 10.13) позволяет выполнять, в зависимости от условия, то или иное действие (или группу действий);

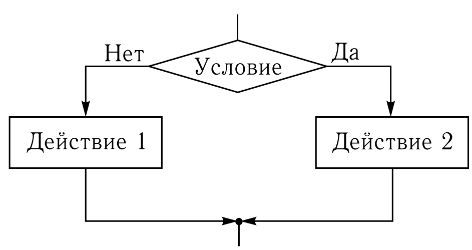


Рис. 10.13. Структура разветвления

• обход (рис. 10.14) представляет собой частный случай разветвления, когда при выполнении (или, наоборот, невыполнении) условия никаких действий не выполняется.

Особенность приведенных структурных связок состоит в том, что все они имеют единственные вход и выход. Это позволяет просто и наглядно формировать из них в любой последовательности структуру алгоритма. В этой связи, если принять рассмотренные структурные связки за элементарные алгоритмы,

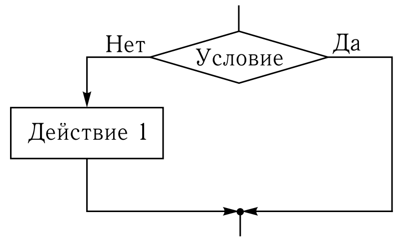


Рис. 10.14. Структура обхода

алгоритм любой задачи можно представить следующим образом:

$$\text{АЛГОРИТМ} = \{\langle\text{элементарные алгоритмы}\rangle\}.$$

Другими словами, алгоритм состоит из набора элементарных алгоритмов. Приведенное определение алгоритма позволяет при разработке алгоритмов сложных задач использовать *иерархический подход*. Суть данного подхода состоит в следующем: сначала анализируется и записывается структура алгоритма (при этом в полной мере возможно использование описанных выше структурных связок), а затем детализируются отдельные блоки предложенной структуры. Процесс детализации является итерационным, причем уровень детализации с каждой итерацией повышается. Очевидно, что общая структура алгоритма решения задачи может влиять и влияет на результат программной реализации.

Рассмотрим распространенный способ записи алгоритмов — *словесное описание*. В этом случае алгоритм задается через описание и перечисление операторов, аналогичных блокам, используемым структурной схемой алгоритма. Таким образом, словесная запись алгоритма представляет собой перечень операторов, пронумерованный в порядке их исполнения алгоритмом. При записи условных операторов, помимо указания условия, истинность которого необходимо проверить, указываются также номера операторов, которым будет передаваться управление в зависимости от результатов проверки.

Пример 10.7. Приведем словесное описание структурной схемы алгоритма из примера 10.5.

1. Присвоить переменной i значение, равное 1.

2. Присвоить x значение, равное a_i (x — произвольная ячейка памяти).

3. Проверяем, i меньше n или нет. Если меньше, то переходим к п. 4, иначе переходим к п. 6.

4. Переменной i присвоить значение, равное $i + 1$.

5. Проверка логического условия $a_i < x$. Если условие выполняется, то переходим к п. 2, если нет, то переходим к п. 3.
6. Печать минимального числа из множества A.
7. Конец работы алгоритма.

Примеры решения задач

Пример 10.1. Построить логическую схему алгоритма (ЛСА) подсчета арифметического выражения:

$$Y = \frac{1}{n}\sum_{i=1}^{n} a_i \times b_{n-i+1}.$$

Решение. Для записи логической схемы алгоритма подсчета данного выражения нами будут использованы следующие операторы:

A_0 — начало программы;

A_1 — ввод объема выборки n;

A_2 — задание начальных значений переменной цикла $i = 0$ и промежуточной переменной $C = 0$;

A_3 — оператор увеличения переменной цикла: $i = i + 1$;

A_4 — вычисление значения индекса переменной b: $j{=}n{-}i{+}1$;

A_5 — вычисление значения произведения $p = a_i \times b_j$;

A_6 — суммирование промежуточных значений p: $C = C + p$;

q — логический оператор проверки условия $i = n$;

A_7 — вычисление значения исходного выражения: $y = C/n$;

A_k — конец программы.

Ответ. Таким образом, логическая схема алгоритма подсчета данного выражения будет иметь следующий вид:

$$A_0\, A_1\, A_2 \downarrow^1 A_3\, A_4\, A_5\, A_6\, q \uparrow^1 A_7\, A_k.$$

Пример 10.2. Построить граф-схему алгоритма (ГСА) подсчета среднего арифметического значения ряда целых четных чисел длины n.

Решение. Графическая схема алгоритма (ГСА) показана на рис. 10.15.

Для построения граф-схемы данного алгоритма используются следующие операторы:

A_0 — начало программы;

A_1 — оператор ввода размера интервала n;

A_2 — оператор ввода начального значения шага: $i = 0$;

A_3 — наращивание значения шага: $i = i + 1$;

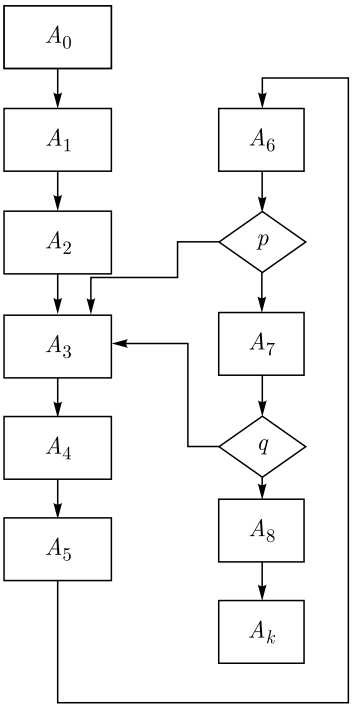


Рис. 10.15. Граф-схема алгоритма

A_4 — оператор выбора начального числа из заданного интервала a_i;

A_5 — вычисление значения $b = a_i/2$;

A_6 — вычисление значения $b = b \times 2$;

p — логический оператор проверки условия $a_i = b$;

A_7 — промежуточное суммирование: $S = S + a_i$;

q — логический оператор проверки условия $i = n$;

A_8 — подсчет среднего арифметического: $Y = S/2$;

A_k — конец программы.

Пример 10.3. Составить структурную схему алгоритма вычисления 50 значений функции:

$$Y_i = \sin(\alpha x_i) \times x_i,$$

где x_i — это одномерный массив натуральных чисел; i — индекс переменной в массиве: $i = 1, 2, \ldots, 50$; α — константа.

Решение. Для построения структурной схемы данного алгоритма используем следующие блоки (согласно ГОСТ 19.701-90).

Блок «НАЧАЛО» — «Терминатор» — неисполняемый блок, символизирующий начало работы алгоритма, всегда располагается перед остальными блоками структурной схемы.

Блок 1 — «Ввод исходных данных» — «Процесс» — в данном блоке осуществляется ввод массива значений переменной x_i и значение константы α.

Блок 2 — «$i = 1$» — «Процесс» — задает начальное значение индекса переменной.

Блок 3 — «$Y_i = \sin(\alpha x_i) \times x_i$» — «Процесс» — производится подсчет значения функции Y_i при данном значении переменной x_i.

Блок 4 — «Печать значений Y_i и x_i» — «Процесс» — данный блок задает процедуру вывода на печать значений переменных Y_i и x_i.

Блок 5 — «$i = i + 1$» — «Процесс» — увеличивает значение индекса i переменной x на единицу после каждого шага выполнения алгоритма.

Блок 6 — «$i > 50$» — «Решение» — блок, осуществляющий управление циклом. В случае, если индекс i переменной x меньше или равен 50, т. е. алгоритм подсчитал меньше 50 значений функции Y_i, блок 6 передаст управление блоку 3, таким образом цикл будет повторяться до тех пор, пока не будет произведено 50 вычислений. После этого блок 6 передаст управление следующему по порядку блоку.

Блок «КОНЕЦ» — «Терминатор» — неисполняемый блок, символизирующий окончание работы алгоритма, всегда располагается после остальных блоков структурной схемы.

Структурная схема алгоритма приведена на рис. 10.16.

Нумерация блоков ведется сверху вниз. В данном примере используется следующий порядок описания блоков: номер блока, затем его название в данной структурной схеме, затем название блока в соответствии с ГОСТом и, наконец, небольшой комментарий к содержанию блока. Стрелки на концах некоторых соединительных линий необходимы, так как в соответствии с ГОСТ 19.701-90 стандартным считается «направление потока слева направо и сверху вниз», а в рассматриваемой структурной схеме есть несколько потоков, имеющих нестандартную направленность (например, снизу вверх).

Пример 10.4. Записать словесный алгоритм вычисления выражения, заданного в примере 10.3.

Ответ. Словесная запись алгоритма вычисления исходного выражения будет выглядеть следующим образом:

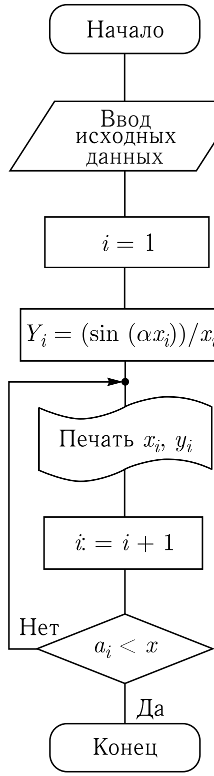


Рис. 10.16. Структурная схема алгоритма

1. Начало работы алгоритма.
2. Ввод исходных данных.
3. Присвоить i значение 1.
4. Вычислить значение переменной: $Y_i = \sin(\alpha x_i)/x_i$.
5. Вывести на печать значения переменных x_i, Y_i.
6. Присвоить i значение $i + 1$.
7. Проверка условия $i > 50$. Если условие истинно, то перейти к пункту 8 настоящего алгоритма, если условие ложно, то перейти к пункту 4.
8. Конец работы алгоритма.

Пример 10.5. Построить структурную схему алгоритма для подсчета среднего арифметического значения ряда целых четных чисел длины n, заданного в примере 10.2.

Решение. Подробное описание всех блоков структурной схемы алгоритма не приводится, поскольку блоки структурной

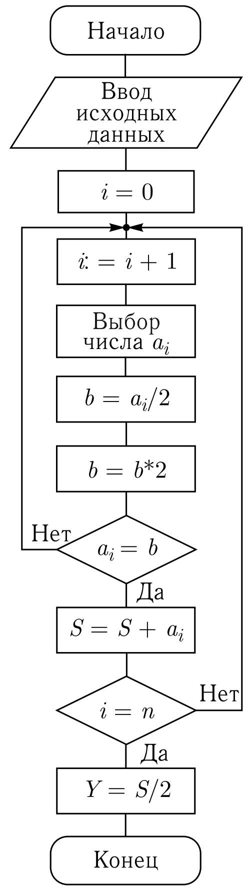


Рис. 10.17. Пример структурной схемы алгоритма

схемы будут аналогичны операторам граф-схемы алгоритма из примера 10.2. Приведем содержание основных 1-го и 3-го блоков.

Блок 1 — Ввод размера интервала n и начального значения шага $i = 0$;

Блок 3 — Выбор начального числа из интервала a.

Ответ. Структурная схема алгоритма показана на рис. 10.17.

10.6. «Жадные» алгоритмы

Для решения оптимизационных задач можно использовать класс алгоритмов, называемых «*жадными*». В таком алгоритме на каждом шаге делается локально оптимальный выбор, в предположении, что итоговое решение окажется оптимальным. В общем виде это не верно, но для многих оптимизационных задач такие алгоритмы позволяют получить локальный оптимум, а в частном случае — глобальный оптимум.

Рассмотрим стандартную задачу (Кормен, Лейзерсон, Ривест) о выборе заявок. Пусть даны n заявок на проведение занятий в одной и той же аудитории. Два разных занятия не могут проводиться в одно время. В каждой заявке указаны начало и конец занятия (s_i и f_i для i-й заявки). Разные заявки могут пересекаться, и тогда можно реализовать только одну из них. Каждая заявка отождествляется с промежутком $[s_i, f_i]$, так что конец одного занятия может совпадать с началом другого, и это не считается пересечением. Целевая функция задачи заключается в наборе максимального количества совместных друг с другом заявок при временных ограничениях и граничных условиях. Жадный алгоритм работает следующим образом. Предполагаем, что заявки упорядочены в порядке возрастания времени окончания, т. е. $f_1 \leqslant f_2 \leqslant \ldots \leqslant f_n$. Заявки с одинаковым временем конца располагаем в произвольном порядке. Тогда запишем псевдокод алгоритма, приведенный в учебнике (Кормен, Лейзерсон, Ривест):

ВЫБОР ЗАЯВОК (s, f — массивы)
1 $n \leftarrow$ length $[s]$
2 $A \leftarrow \{1\}$
3 $j \leftarrow 1$
4 for $i \leftarrow 2$ to n
5 do if $s_i \geqslant f_j$
6 then $A \leftarrow A \cup \{i\}$
7 $j \leftarrow i$
8 return A

Множество A состоит из номеров выбранных заявок, j — номер последней из них; при этом $f_j = \max\{f_k : k \in A\}$, поскольку заявки отсортированы по возрастанию времени окончания. Вначале A содержит заявку номер 1, и $j = 1$ (строки 2–3). Далее (цикл в строках 4–7) ищется заявка, начинающаяся не раньше окончания заявки номер j. Если таковая найдена, она включается в множество A и переменной j присваивается ее номер (строки 6–7). Описанный алгоритм выбора заявок требует всего лишь $O(n)$ шагов (не считая предварительной сортировки). Как и подобает жадному алгоритму, на каждом шаге он делает выбор так, чтобы остающееся свободным время было максимально.

Говорят, что к оптимизационной задаче применим *принцип жадного выбора*, если последовательность локально оптимальных (жадных) выборов дает глобально оптимальное решение.

Отметим, что задачи, решаемые с помощью жадных алгоритмов, должны обладать свойством *оптимальности для подзадач*. А именно: оптимальное решение всей задачи содержит в себе оптимальные решения подзадач.

10.7. Нечеткие (расплывчатые) алгоритмы

Нечеткий (*расплывчатый*) *алгоритм* — это упорядоченное множество расплывчатых инструкций, которые при их реализации позволяют получать приближенное решение задач.

Инструкции в расплывчатых алгоритмах делят на три группы.

1-я группа — назначающие инструкции. Например, $x =$ «большой» или $x =$ «вкусный».

2-я группа — расплывчатые высказывания. Например, если x — «очень дорогой», то перейти к y, или если x «не помещается в комнате», то «увеличить размер комнаты».

3-я группа — безусловные активные предложения. Например, «немного увеличить» x, или «сильно увеличить» x, или перейти к другому значению параметра x.

Отметим, что все инструкции могут быть нечеткими или четкими.

Нечеткие алгоритмы определения — это конечное множество инструкций, определяющих расплывчатое множество в терминах других расплывчатых множеств или дающих метод для определения степени принадлежности каждого элемента для определяемого множества. *Алгоритмы идентификации* устанавливают степень принадлежности элементов множеству. *Алгоритмы порождения* используются для порождения новых, а не для определения данных расплывчатых множеств. Примерами таких алгоритмов могут служить алгоритмы сочинения стихов, конструирования ЭВА, разработки технических заданий на изделие. Нечеткие алгоритмы, используемые для описания поведения произвольной системы, называются *бихевиористическими*.

Расплывчатой переменной считают кортеж $\Pi = \langle \alpha, X, \widetilde{A} \rangle$, где α — наименование расплывчатой переменной; $X = \{x_1, x_2, \ldots, x_n\}$ — область ее определения; $A = \{\mu_{\widetilde{A}}(x_i) \,|\, x_i\} x_i \in X$ — расплывчатое множество в X, определяющее ограничения на возможные значения П. *Лингви-*

стической переменной называют кортеж $Л = \langle \beta, T, X \rangle$, где β — наименование Л; T — множество ее значений, представляющих наименования П, областью определения которых является X.

Например, пусть оценивается качество технологического процесса выхода годных с помощью понятий «плохой», «средний», «хороший». Максимальный выход годных 80 %. Тогда $Л = \langle \text{качество}, T, [0; 80] \rangle$, где $T = \{$«плохой», «средний», «хороший»$\}$. Значения Л описываются расплывчатыми переменными. Например, значение «плохой» описывается расплывчатой переменной $\langle \text{плохой}, [0; 80], \widetilde{A} \rangle$, где $\widetilde{A}$ может иметь вид $\widetilde{A} = \{\langle 1/0 \rangle, \langle 0,8/5 \rangle, \langle 0,6/7 \rangle, \langle 0,2/20 \rangle\}$. Значение «хороший» можно, например, описать так: $\langle \text{хороший}, [0; 80], \widetilde{A} \rangle$, где $\widetilde{A} = \{\langle 1/80 \rangle, \langle 0,8/75 \rangle, \langle 0,6/70 \rangle, \langle 0,2/40 \rangle\}$.

Функцию принадлежности при наличии k экспертов упрощенно определяют так. Пусть k_1 экспертов на вопрос о принадлежности $x \in X$ к $\widetilde{A}$ отвечают положительно, а $k_2 = k - k_1$ экспертов на этот же вопрос отвечают отрицательно. Тогда устанавливают следующее значение функции принадлежности: $\mu_{\widetilde{A}}(x_i) = k_1/\{k_1 + k_2)$. Для точных оценок приводят парные сравнения функций принадлежности и др. Лингвистические переменные используют для построения расплывчатых моделей и алгоритмов.

Например, приведем нечеткий алгоритм. Пусть y — быстродействие системы, состоящей из двух ЭВМ, а x — быстродействие одной ЭВМ в системе.

Тогда запишем следующий нечеткий алгоритм построения системы с заданным быстродействием:

1°. Если x «мало» и x «незначительно увеличить», то y «незначительно увеличится».

2°. Если x «мало» и x «значительно увеличить», то y «увеличится значительно».

3°. Если x «велико» и x «увеличить значительно», то y «увеличится очень значительно».

Заметим, что значения расплывчатых условных предложений в этом алгоритме можно определить, если даны значения первичных терминов «большой», «маленький» и расплывчатых терминов «незначительно», «значительно», «очень значительно». Значения указанных терминов субъективны и зависят от носителя нечеткого множества. Например, если четкое множество $X = \{1, 2, 3, \ldots, 10\}$, то при $x = 9$ мы имеем, скорее всего, термин «большой». Если четкое множество $X = \{1, 2, 3, \ldots, 1000\}$, то при $x = 9$ мы имеем, скорее всего, термин «маленький».

Нечеткий алгоритм, который служит для получения приближенного описания стратегии и решающего правила, называется нечетким *алгоритмом принятия решений*. К такому классу можно отнести алгоритмы принятия решений при пересечении перекрестка, компоновке кристаллов интегральной микросхемы, определении жизненного цикла изделий и др. На рис. 10.18 приведена структурная схема нечеткого алгоритма трассировки (определения путей между частями интегральной схемы).

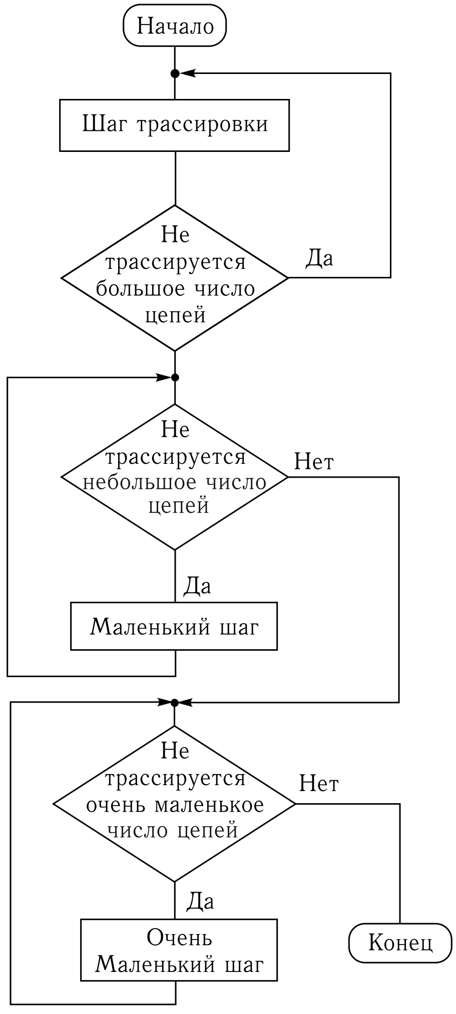


Рис. 10.18. Структурная схема нечеткого алгоритма «Трассировка»

Эту структурную схему (рис. 10.18) можно представить в виде следующих нечетких инструкций:

1°. Сделать шаг трассировки. Если не трассируется «большое» число связей, то перейти к 1°.

2°. Если не трассируется «небольшое» число связей, то сделать «малый» шаг и перейти к 2°.

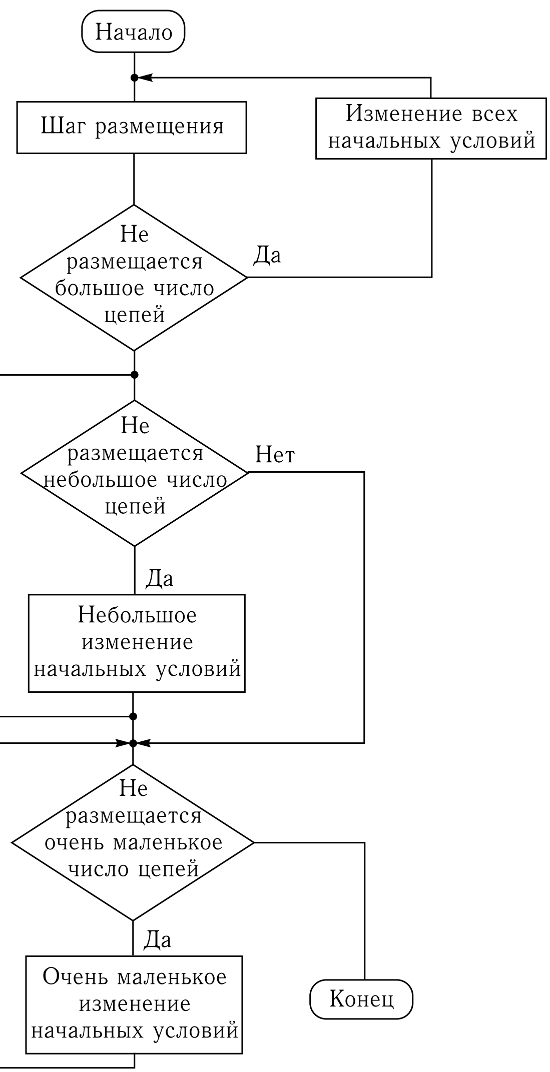


Рис. 10.19. Структурная схема расплывчатого алгоритма размещения

3°. Если не трассируется «очень маленькое» число связей, то сделать «очень маленький шаг» и перейти к 3°.

4°. Конец работы алгоритма.

Рассмотрим пример размещения БИС на подложке. Структурная схема одного из возможных алгоритмов такого типа показана на рис. 10.19. В терминах нечетких высказываний структурная схема алгоритма может быть переведена в расплывчатые инструкции.

Следует отметить, что конструктор хорошо оперирует расплывчатыми понятиями «большое», «маленькое», «очень маленькое» и эффективно реализует данный алгоритм в интерактивном режиме связи с ЭВМ. При определении степени принадлежности можно формализовать алгоритм и реализовать его на ЭВМ. Отметим, что операция «сделать шаг» в алгоритме (рис. 10.18) может быть сложным алгоритмом перетрассировки всего кристалла или небольшой его области, или простого удаления нескольких трасс и т. п. Реальные варианты нечетких алгоритмов представляют сложную структуру и необходимо разбиение его на части меньшей размерности. Важным в разработке и применении таких алгоритмов является то, что они могут служить эффективным средством распространения и изучения опыта пользователя.

Контрольные вопросы

1. Назовите основные виды универсальных алгоритмических моделей.
2. Что представляет собой первая алгоритмическая модель?
3. В чем заключается преобразование слов в произвольных абстрактных алфавитах?
4. Что такое алфавитный оператор?
5. Дайте различные определения алгоритма на основе первой алгоритмической модели.
6. Какие алгоритмы являются равными, а какие — эквивалентными?
7. Определите вторую алгоритмическую модель.
8. Каким образом строятся рекурсивные алгоритмы?
9. Что такое алгоритмы, сопутствующие рекурсивным функциям?
10. Что такое рекурсивные и частично рекурсивные функции?
11. Определите операторы суперпозиции, подстановки и рекурсии.
12. Сформулируйте тезис Чёрча.
13. В чем заключается метод «разделяй и властвуй»?

14. Постройте третью алгоритмическую модель.

15. Опишите принцип работы машины Тьюринга.

16. Приведите правила построения алгоритмов Ван-Хао.

17. Приведите основные правила построения логических схем алгоритмов (ЛСА).

18. В чем заключаются правила построения алгоритмов Маркова?

19. Определите основную идею графических схем алгоритмов (ГСА).

20. Приведите основные правила построения структурных схем алгоритмов.

21. В чем основная идея словесного описания алгоритма?

22. Что представляют собой «жадные» алгоритмы?

23. Для каких задач целесообразно применять «жадные» алгоритмы?

24. Дайте определение нечеткого (расплывчатого) алгоритма.

25. Что такое расплывчатая и лингвистическая переменные?

Задания для самостоятельной работы

1. Постройте пример первой алгоритмической модели.

2. Постройте пример второй алгоритмической модели.

3. Постройте пример третьей алгоритмической модели.

4. Запишите словесный алгоритм упорядочивания массива из n чисел в порядке возрастания.

5. Запишите словесный алгоритм для вычисления среднего арифметического и наибольшего значений элементов массива целых чисел размерностью до 70.

6. Запишите алгоритм Ван-Хао для вычисления среднего арифметического значения элементов массива чисел размерностью до 20.

7. Составить словесный алгоритм определения элементов множества Z, если $Z = X \backslash Y$, где X, Y — множества ($|X| = N$, $|Y| = M$).

8. Составить ЛСА поиска наибольшего элемента массива A_{ij}, где $i = 1, 2, \ldots, 10$; $j = 1, 2, \ldots, 20$.

9. Составить ГСА вычисления по формуле $K!$

10. Составить ГСА нахождения одинаковых чисел в одномерном массиве длины n.

11. Запишите ГСА вычисления по формуле $C = \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m} (A_i + B_j)/(n + m)$.

12. Составить структурную схему алгоритма вычисления по формуле $C = \sum\limits_{i=1}^{n} (A_i - B_i)$.

13. Составить ЛСА вычисления по формуле $D = \dfrac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m} C_{ij}$.

14. Составить структурную схему алгоритма вычисления по формуле: $D = 1/N \sum\limits_{i=1}^{n} (X_i - X_0)^2$, где $X_0 = 1/N \sum\limits_{i=1}^{n} X_i$.

15. Составить ЛСА алгоритма вычисления по формуле: $C = \sum\limits_{j=1}^{m} \sum\limits_{i=1}^{n} (A_i - B_j)$.

16. Составьте логическую схему алгоритма сортировки для массива $A = (3, 41, 52, 26, 38, 57, 9, 49)$.

17. Приведите пример использования «жадного» алгоритма для решения задачи нахождения наибольшего числа в массиве $A = (13, 41, 89, 26, 38, 57, 9, 49)$.

18. Составьте нечеткий алгоритм перехода перекрестка с односторонним движением.

19. Составьте нечеткий алгоритм перехода перекрестка с двусторонним движением.

Модель — это копия или аналог изучаемого явления или процесса, отражающая существенные свойства моделируемого объекта с точки зрения исследователя. Математическая модель — это приближенное описание какого-либо класса явлений внешнего мира, выраженное с помощью математической символики. Использование таких моделей позволяет строить эффективные схемы алгоритмов.

Глава 11
СЛОЖНОСТЬ АЛГОРИТМОВ

> Не рассуждайте о том, почему происходит какое-то явление, — описывайте его количественно.
>
> *Галилей*

Временная сложность алгоритма, полиномиальные и экспоненциальные алгоритмы, анализ алгоритмов

ЦЕЛИ

Освоив эту главу, учащиеся должны:

- уметь определять временную сложность алгоритмов;
- уметь определять полиномиальные и экспоненциальные алгоритмы;
- уметь анализировать алгоритмы.

11.1. Анализ алгоритмов

Важнейшая практическая задача теории и практики проектирования устройств различного типа — это анализ алгоритма. Кроме того, анализ алгоритмов — одна из важнейших задач дискретной математики. С анализом алгоритма связано время работы алгоритма. Оно также связано с ограничениями на характеристики работы ЭВМ и со сложностью решаемой задачи. Для выбора лучшего алгоритма необходимо получение оценок или границ для объема памяти и времени работы ЭВМ, которое потребуется алгоритму для обработки конкретных данных. Для этого необходимо сформулировать количественный критерий для сравнения двух алгоритмов, решающих одну и ту же задачу. В этой связи желательно определить, в каком случае решение задачи считается оптимальным.

При анализе алгоритма определяется количество времени, необходимое для его выполнения на ЭВМ. Это приблизительное количество времени, затраченное на выполнение стандартных или приведенных к ним операций, выполняемых алгоритмом. Тогда вычислительная сложность алгоритма обычно называется «временем работы алгоритма».

Важная характеристика конкретного алгоритма — зависимость числа операций и времени работы алгоритма от размера его входных данных. Например, входными данными в алгоритме, анализирующем графовую модель, могут быть число вершин и ребер графа. Тогда временная сложность алгоритма — это зависимость времени решения задачи от числа вершин или ребер графа. В информационной или управляющей модели входными данными может быть количество блоков. При проектировании печатных плат входными данными может быть число цепей или микросхем и т. п. Тогда временная сложность алгоритма проектирования печатных плат будет, например, представлять зависимость времени проектирования платы от числа микросхем и числа цепей схемы.

Известно, что последовательность действий алгоритма определяется входными данными, структурой модели, целевой функцией задачи, форматом выходных данных и другими данными.

В настоящее время алгоритмы сравниваются по скорости роста числа операций. Это связано с тем, что при небольшом размере ($n < 50$) входных данных одному алгоритму потребуется меньше операций, чем другому. При росте объема входных данных ($n > 50$) для указанных алгоритмов ситуация может поменяться на противоположную. На основе анализа алгоритмов оценивается, насколько быстро решается задача на массиве входных данных размера n. Следовательно, анализ алгоритма позволяет пользователю выбирать эффективный, с его точки зрения, алгоритм.

При анализе алгоритмов решения сложных задач ($n > 1000$) для упрощения различные входные множества разбиваются на классы в зависимости от поведения алгоритма на каждом множестве. Такое разбиение позволяет уменьшить число рассматриваемых возможностей. Например, число различных расстановок множества из 20 неповторяющихся чисел есть $20! = 2432902008176640000$.

Применим к данному множеству из 20 чисел алгоритм поиска наименьшего элемента. Как уже было сказано, имеется 2432902008176640000 вариантов расстановки чисел в начальном множестве. Их все можно поместить в один класс. Количество вариантов, когда наименьшее число стоит на втором месте, в 20 раз меньше, а именно $19! = 121645100408832000$. Их можно отнести к другому классу. Таким образом, можно разбить все входные множества на N разных классов. Для описанного алгоритма разбиение основывается на местоположении элемента, имеющего наименьшее значение. В результате получается 20 классов. После выделения классов можно проанализировать поведение алгоритма на одном множестве из каждого класса. Это разбиение выполняется на основе принципа «разделяй и властвуй», приведенного выше. Если разбиение выбрано правильно с точки зрения пользователя, то на всех множествах входных данных одного класса алгоритм производит одинаковое количество операций, а на множествах из другого класса это количество операций будет другим.

В алгоритмах обычно выделяют операции двух типов:

— сравнение;

— арифметические операции:

• аддитивные операторы;

• мультипликативные операторы.

Существует *тезис Тьюринга*: любой алгоритм может быть реализован на машине Тьюринга. Математически доказано, что если задача не может быть решена на машине Тьюринга, то она не может быть решена ни на какой другой автоматической вычислительной машине.

Один из основных результатов Тьюринга — разделение всех задач, решаемых с помощью алгоритмов, на два типа (класса).

1 класс. Задачи, для которых алгоритмы никогда не могут быть написаны, т. е. задачи в принципе не решаемые. Например, задача о квадратуре круга или построение универсального решателя (алгоритма) для решения всех задач.

Иногда часть проблем, для решения которых нет алгоритма, можно сделать разрешимыми. Это можно осуществить, например, за счет расширения набора допустимых операций, упрощения задачи, снижения требований, введения ограничений, разбиения ее на подзадачи и т. п.

2 класс. Это класс решаемых задач, т. е. задач, для которых могут быть написаны и реализованы алгоритмы. К ним относятся задачи, решаемые на основе:

• полиномиальных алгоритмов;

• экспоненциальных алгоритмов.

Экспоненциальными алгоритмами называются алгоритмы, у которых время решения экспоненциально растет по мере увеличения размера входных данных. К ним относятся алгоритмы типа 2^n, $n!$, $n!!$ и т. п. Здесь n — количество входов алгоритма.

К экспоненциальным алгоритмам принадлежат алгоритмы полного перебора при нахождении оптимального решения.

Пример 1. Найти все одинаковые числа в массиве натуральных чисел размера $n \times n$.

Пример 2. Задача о коммивояжере. Пусть некоему коммивожеру необходимо посетить n городов и вернуться в исходную точку, причем каждый город он может посетить только один раз. Найти такой маршрут коммиявожера, чтобы пройденный им путь был минимальным.

Очевидно, что в обоих задачах для нахождения оптимального решения необходимо перебрать все возможные варианты. Следовательно, эти задачи будут иметь экспоненциальную временную сложность.

Полиномиальные алгоритмы — это такие алгоритмы, при реализации которых на ЭВМ время решения изменяется в зависимости от размера входных данных (например, числа вершин или ребер графа). Эта зависимость представляется в виде полиномов, где число входов n принимает значения n, n^2, $n^3, \ldots$ Если число входов равно n, то имеем дело с «линейным» алгоритмом. *Линейный алгоритм* — это алгоритм, у которого зависимость времени решения от числа входных данных носит линейный характер. Другими словами, полиномиальные алгоритмы — это алгоритмы, у которых время решения находится в полиномиальной зависимости от изменения размера входных данных.

Пример 3. Пусть время решения произвольного алгоритма находится в следующей зависимости от размера входных данных: $t = an + b$, где n — это числа из множества $A = \{1, 5, 10, 15, 20, 25, 30\}$; a, b — постоянные величины ($a = 20$, $b = 10$). График временной сложности алгоритма показан на рис. 11.1 (функция 1).

Пример 4. Пусть время решения произвольного алгоритма находится в следующей зависимости от размера входных данных: $t = an^2 - bn + c$, где n — числа из множества A (пример 3); a, b, c — постоянные величины ($a = 1$, $b = 7$, $c = 10$). График временной сложности алгоритма представлен на рис. 11.1 (функция 2).

График функции 1 отражает линейную зависимость времени работы алгоритма от числа входных данных, а график функции 2 — квадратичную.

Анализируя графики, видно, что существует некоторая область для входных данных небольшого размера, где временная сложность «квадратичных» алгоритмов лучше, чем линейных. Так, например, на рис. 11.1 видно, что в области, где n меняется от 1 до приблизительно 27, квадратичный алгоритм работает быстрее линейного. Затем ($n > 27$) линейный алгоритм работает значительно быстрее.

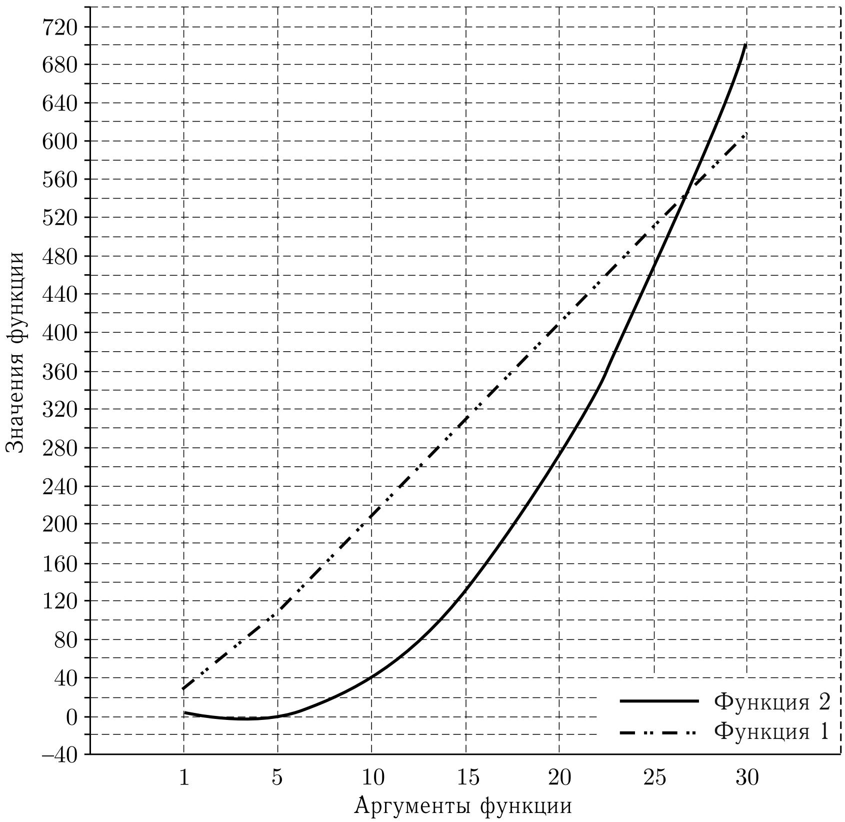


Рис. 11.1. Сравнение временной сложности функций

Аппроксимированные графики временной сложности полиномиальных алгоритмов различных классов показаны на рис. 11.2.

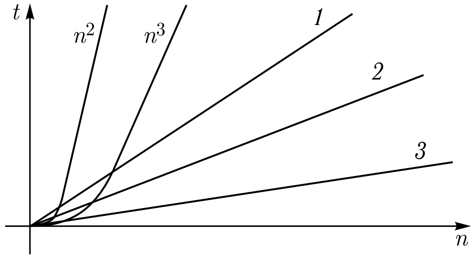


Рис. 11.2. Графики временной сложности различных классов алгоритмов

Существует классификация задач, для которых могут быть написаны алгоритмы. С учетом временной сложности этих алгоритмов они могут быть разделены на два класса. На рис. 11.3 представлен пример такой классификации алгоритмов.

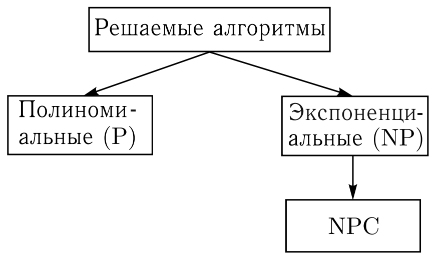


Рис. 11.3. Классификация алгоритмов

Алгоритмы, применяемые на практике, обычно разбивают на два класса: детерминированные и недетерминированные. Детерминированные алгоритмы состоят из операций, результаты каждой из которых определены однозначно, в противном случае — это недетерминированный алгоритм.

Задачей на допустимость называется задача определения хотя бы одного возможного решения.

Задачей на оптимальность считается задача определения допустимого решения, дающего экстремум функции.

Множество всех задач на допустимость, которые могут быть решены на основе детерминированного алгоритма за полиномиальное время, относятся к *подклассу* P (полиномиальных алгоритмов).

Согласно классическому учебнику Кормена, Лейзерсона и Ривеста «Алгоритмы: построение и анализ» запись $f(n) = \Theta(g(n))$ включает в себя две оценки временной сложности алгоритма: верхнюю и нижнюю. Их разделяют на $f(n) = O(g(n))$ и $f(n) = \Omega(g(n))$.

Говорят, что $f(n) = O(g(n))$, если найдется такая константа $c > 0$ и такое число входных данных n_0, что $0 \leqslant f(n) \leqslant cg(n)$ для всех $n \geqslant n_0$.

Говорят, что $f(n) = \Omega(g(n))$, если найдется такая константа $c > 0$ и такое число n_0, что $0 \leqslant cg(n) \leqslant f(n)$ для всех $n \geqslant n_0$.

Здесь предполагается, что функции f и g неотрицательны для достаточно больших значений аргумента.

Асимптотически обозначают, например, $8n^4 + 5n^3 + 13n^2 + 9n + 4 \approx 8n^4 + \Theta(n^3) + \Theta(n^2) + \Theta(n) \approx \Theta(n^4)$. То есть какой бы ни была функция $k(n) = \Theta(n^3), \Theta(n^2), \Theta(n)$ в левой части, сумма $8n^4 + \Theta(n^3) + \Theta(n^2) + \Theta(n)$ есть $\Theta(n^4)$. Тогда время работы полиномиальных алгоритмов такого типа составляет не более $\Theta(n^k)$ для некоторой константы, не зависящей от длины входа.

В теории встречаются задачи, для которых существует решающий их алгоритм, но любой такой алгоритм работает «долго» — время его работы не есть $O(n^k)$ ни для одного фиксированного числа k. Согласно работе Кормена, Лейзерсона и Ривеста «Алгоритмы: построение и анализ» полиномиальные алгоритмы представляют собой некую границу между «практическими» алгоритмами и алгоритмами, представляющими лишь теоретический интерес.

Отдельный класс составляют задачи, называемые «NP-полными». Для них не найдены полиномиальные алгоритмы, однако и не доказано, что таких алгоритмов не существует. Изучение NP-полных задач связано с нерешенными проблемой $P = \mathrm{NP}$ или $P \neq \mathrm{NP}$. Сокращение NP происходит от английских слов Nondeterministic Polynomial time, что переводится как недетерминированное полиномиальное время.

Большинство специалистов полагают, что NP-полные задачи нельзя решить за полиномиальное время. Очевидно, что если хотя бы для одной NP-полной задачи существует решающий ее полиномиальный алгоритм, то и для всех NP-полных задач такие алгоритмы существуют. В настоящее время известно много NP-полных задач. На практике считают, что если для некоторой задачи удается доказать ее NP-полноту, то она является практически неразрешимой.

В теории алгоритмов при исследовании NP-полноты рассматриваются в основном *задачи разрешения*, т. е. задачи, в которых требуется дать ответ «да» или «нет» на некоторый вопрос. Формально задачу разрешения можно рассматривать как функцию, отображающую множество условий I в множество решений $S = \{0, 1\}$, где $1 =$ «да», $0 =$ «нет». В этой связи для определения NP-полноты многие задачи преобразуют к указанному виду.

Перед исследованием NP-полноты оптимизационных задач на графах их также преобразуют в задачу разрешения. В оптимизационных задачах обычно требуется минимизировать или максимизировать значение некоторой величины. Прежде чем ставить вопрос о NP-полноте таких задач, их следует преобразовать в задачу разрешения. Обычно в качестве такой задачи разрешения рассматривают задачу проверки, является ли некоторое число верхней (или нижней) границей для оптимизируемой величины.

Поэтому данные задачи выделяют в особый класс оптимизационных комбинаторных задач на графах. В задачах такого класса общее число вариантов решения равно числу перестановок из n вершин графов, т. е. $C_n = n!$, а с учетом ограничений на формирование подмножеств (частей графа в задаче разбиения) — числу сочетаний из n вершин по m частей, т. е. $C_n^m = n!/m!(n-m)!$.

Если после этого получается NP-полная задача, то тем самым установлена трудность исходной задачи. В самом деле, если для оптимизационной задачи имеется быстрый алгоритм, то и полученную из нее задачу разрешения можно решить быстро (надо просто сравнить ответ этого алгоритма с заданной границей). Таким образом, теорию NP-полноты можно использовать и для исследования задач оптимизации.

Говорят, что алгоритм решает задачу за время $O(T(n^2))$, если на входном данном, например, на битовой строке i длины n, алгоритм работает за время $O(T(n^2))$.

Говорят, что функция $f : \{0,1\} \to \{0,1\}$ вычислима за полиномиальное время, если существует полиномиальный алгоритм, который для любого элемента $y \in \{0,1\}$ выдает результат $f(y)$.

Интуитивно класс P можно представлять себе как класс задач, которые можно «быстро» решить, а класс NP — как класс задач, которые нельзя решить за полиномиальное время. Известно, что ни для одной NP-полной задачи не найден полиномиальный разрешающий алгоритм.

Конечно, нет уверенности, что однажды не будет найден полиномиальный алгоритм для решения NP-полной задачи, это и будет доказательством, что $P = \mathrm{NP}$.

11.2. Сложность алгоритмов

Изменение алгоритма во времени или время, затраченное алгоритмом как функция размерности задачи, называется *временной сложностью алгоритма* (ВСА).

ВСА — это количество единиц времени, требуемое для обработки входных данных задачи.

Обозначается ВСА $O[f_A(n)]$. Функция $f_A(n)$ дает верхнюю границу для максимального числа операций, которые должен выполнить алгоритм для решения любой задачи размерности n.

Основные операции — это сдвиг, сравнение, сложение и тому подобное.

Существует теорема, что для достаточно больших n линейные алгоритмы всегда эффективнее квадратичных, кубичных и т. д. алгоритмов.

Четкой границы между экспоненциальными и полиномиальными алгоритмами нет.

Все алгоритмы, у которых ВСА $O(n)$, $O(n^2)$, $O(n^3)$, считаются эффективными и применяются при решении практических задач.

Все алгоритмы, у которых ВСА $O(n^5)$ и выше, считаются не эффективными для использования на практике.

Кроме ВСА важной характеристикой является сложность алгоритма (СА). *Сложность алгоритма* характеризуется числом выполняемых операций N и общим объемом информации.

Поскольку выполнение разных типов операций требует разного времени, СА зависит не только от общего количества операций, но и от количества типов операций и количества операций каждого типа, используемых рассматриваемым алгоритмом.

С числом операций алгоритма связано время его выполнения на конкретной ЭВМ. Объем информации связан с объемом памяти ЭВМ, необходимой для реализации этого алгоритма. Значит, время реализации алгоритма есть функция $T = f(N, V)$.

Но выполнение различных операций требует различного времени. Поэтому для определения сложности алгоритма необходимо знать не только общее количество операций, но и количество используемых в алгоритме типов операций и количество операций каждого типа в отдельности.

Количество операций для алгоритма получается следующим образом. На основании анализа алгоритма можно получить вектор встречаемости операций в алгоритме

$$N = \{Q_1, Q_2, \ldots, Q_r\},$$

где Q_i — количество операций в исследуемом классе задач, $N = \sum_{i=1}^{r} Q_i$, $P_i = Q_i/N$ — относительная частота операций.

Умножая время выполнения каждой операции на количество этих операций и суммируя результаты по всем операциям, получим время реализации алгоритма:

$$T = \sum_{i=1}^{r} Q_i t_i.$$

Принимая за эталон операцию сложения, можно определить общее количество приведенных операций. Для этого необходимо знать относительные веса различных операций по отношению к эталонной операции:

$$B_i = t_i / t_э,$$

где t_i — время реализации i-й операции; $t_э$ — время реализации эталонной операции сложения.

Тогда количество условных элементарных операций составит

$$N = \sum_{i=1}^{r} B_i Q_i,$$

а общее время их реализации — $T_э = N_э t_э$.

Объем информации, используемой алгоритмом, определяется как суммарный объем, необходимый для размещения программной, исходной, промежуточной и выходной информации.

$$И = И_{п} + И_{и} + И_{в} + И_{пр},$$

где И — общий объем информации; $И_{п}$ — объем программной информации; $И_{и}$ — объем исходной информации; $И_{в}$ — объем выходной информации; $И_{пр}$ — объем промежуточной информации.

Тогда обобщенный коэффициент сложности $К_{сл} = N_э / И$, т. е. он характеризуется числом эталонных операций, выполняемых при обработке единицы информации.

В современных ЭВМ распределением памяти занимается операционная система, и пользователь зачастую не знает, какую память он использует.

В современных ЭВМ объем внешней памяти значительно опережает объем оперативной памяти:

$$V_{в.п.} \gg V_{о.п.},$$

поэтому $T = f(N, V_{в.п.})$.

Основной путь повышения скорости реализации алгоритма — это параллельное выполнение большого числа операций.

Рассматривая различные алгоритмы решения одной и той же задачи, полезно проанализировать, сколько вычислительных ресурсов они требуют (время работы, память), и выбрать наиболее эффективный.

Обычно изучают зависимость времени работы от размера входной последовательности. Способ измерения размера входа зависит от конкретной задачи. В одних случаях, как например, для задачи сортировки, под размером понимают число элементов на входе. В других размером принято считать общее число битов, необходимое для представления всех входных данных.

Иногда размер входа измеряется не одним числом, а несколькими (например, число вершин и число ребер графа).

Временем работы алгоритма называется число элементарных шагов, которые он выполняет. Каждая строка псевдокода требует фиксированного числа операций (если только это не словесное описание каких-то сложных действий типа «отсортировать все точки по координате x»). Также необходимо различать *вызов* процедуры (исполняемый за фиксированное число операций) и ее *выполнение*, которое может быть долгим.

Из вышесказанного следует, что время работы в худшем случае и в лучшем случае может сильно различаться. Причем наибольший интерес представляет время работы в худшем случае, которое определяется как максимальное время работы для входов данного размера. Это обусловлено следующими причинами:

• зная время работы в худшем случае, можно гарантировать, что выполнение алгоритма для всех входов данной размерности займет не более определенного промежутка времени вне зависимости от конкретного входа;

• на практике ситуации, когда время работы алгоритма близко к максимуму, встречаются довольно часто.

В некоторых случаях представляет интерес также среднее время работы алгоритма на входах заданной длины. Конечно, эта величина зависит от выбранного распределения вероятностей, и на практике реальное распределение входов может отличаться от предполагаемого, которое обычно считают равномерным. Иногда можно добиться равномерности распределения, используя датчик случайных чисел.

Время работы в среднем может быть довольно близко к времени работы в худшем случае.

Для анализа рекурсивных алгоритмов необходимо учесть время, затрачиваемое на рекурсивные вызовы, и затем оценить время работы исходя из учета этого параметра.

Предположим, что алгоритм разбивает задачу размера n на a подзадач, каждая из которых имеет в b раз меньший размер. Считаем, что разбиение требует времени $D(n)$, а соединение полученных решений — времени $C(n)$. Тогда получаем соотношение для времени работы $T(n)$ на задачах размера n (в худшем случае): $T(n) = aT(n/b) + D(n) + C(n)$. Это соотношение справедливо для достаточно больших n, когда задачу имеет смысл разбивать на подзадачи. Для малых n, когда такое разбиение невозможно или не нужно, применяется какой-либо прямой метод решения задачи. Так как n ограничено, время работы тоже не превосходит некоторой константы.

Можно сказать, что разработка эффективных алгоритмов не менее важна, чем разработка аппаратного обеспечения.

Основной метод повышения скорости работы алгоритма — это параллельные вычисления. В соответствии с гипотезой Минского параллельные машины, выполняющие последовательную программу над n множествами данных, повышают производительность, по крайней мере, на множитель $O(\log n)$.

Следовательно, можно получить выигрыш на порядок величины, если реализовать алгоритм в параллельной модификации. Это достигается при добавлении n процессоров, увеличении числа выполняемых операций и уменьшении коэффициента использования процессоров.

Контрольные вопросы

1. Дайте определение понятия алгоритма.
2. Назовите основные свойства алгоритма.
3. Перечислите виды алгоритмов.
4. Приведите классификацию алгоритмов.
5. Назовите основные виды универсальных алгоритмических моделей.
6. Что представляет собой первая алгоритмическая модель?
7. Каким образом строятся рекурсивные алгоритмы?
8. Опишите принцип работы машины Тьюринга.
9. Приведите основные правила построения ЛСА.
10. Приведите основные правила построения ГСА.
11. Приведите основные правила построения структурных схем алгоритмов.
12. Что представляют собой «жадные» алгоритмы?
13. Для каких задач наиболее целесообразно применять «жадные» алгоритмы?
14. Как определяется временная сложность алгоритма?
15. Перечислите параметры, характеризующие эффективность алгоритма.
16. Поясните, в чем состоят отличия между полиномиальными и экспоненциальными алгоритмами.
17. Определите алгоритмы подклассов P, NP, NPC.
18. Поясните принципы анализа алгоритмов.
19. Каким образом определяется время работы алгоритмов?

20. Перечислите методы математического программирования.

21. Приведите примеры и дайте определение задач на допустимость и оптимальность.

Задания для самостоятельной работы

1. Дана последовательность чисел $x_1, x_2, \ldots, x_n$. Покажите, что за время $O(n \log n)$ можно определить, есть ли в этой последовательности три одинаковых числа.

2. Как записать выражение $n^3/1000 - 100n^2 - 100n + 3$ с помощью $O(n)$-обозначений?

3. Дан массив S из n действительных чисел, а также число x. Как за время $O(n \log n)$ определить, можно ли представить x в виде суммы трех элементов массива S?

4. При каком наименьшем значении n алгоритм, требующий $1000n^2$ операций, эффективнее алгоритма, требующего 3^n операций?

5. Пусть имеется алгоритм, решающий задачу размера n за время $f(n)$ микросекунд. Каков максимальный размер задачи, которую он может решить за время t? Заполнить таблицу.

	1 секунда	1 минута	1 час	1 день	1 месяц	1 год	1 век
$\log n$							
$\sqrt{n}$							
n							
$n \log n$							
n^2							
n^3							
n^4							
2^n							
3^n							
$n!$							

6. Следуя примеру, покажите работу сортировки слиянием для массива $A = (3, 41, 52, 26, 38, 57, 9, 49)$.

7. Напишите текст процедуры СЛИЯНИЕ(A, p, q, r).

8. Сортировку вставками можно оформить как рекурсивную процедуру: желая отсортировать $A[1, \ldots, n]$, мы рекурсивно сортируем $A[1, \ldots, n-1]$, а затем ставим $A[n]$ на нужное место

в отсортированном массиве $A[1, \ldots, n-1]$. Напишите рекуррентное соотношение для времени работы такой процедуры.

9. Для задачи о выборе заявок возможно несколько способов жадного выбора. Покажите на примере, что правила «на каждом шаге выбирать заявку наименьшей длительности, совместную с уже выбранными», а также «на каждом шаге выбирать заявку, совместную с наибольшим количеством остающихся», не годятся.

Тестовые задания к модулю 2

1. Сколько типов основных универсальных алгоритмических моделей Вам известно?

- 2
- 3
- 4
- 1

2. Какие из перечисленных ниже понятий относятся к свойствам алгоритмов?

- детерминированность
- понятность
- результативность
- эквивалентность

3. Рекурсия — это:

- функция
- свойство алгоритмов
- класс алгоритмов
- способ задания функций

4. Какой (или какие) из перечисленных ниже терминов не относятся к понятию «сложность алгоритма»?

- экспоненциальные алгоритмы
- эмпирические алгоритмы
- полиномиальные алгоритмы
- случайные алгоритмы

5. Какие из перечисленных ниже понятий не относятся к характеристикам алгоритмов?

- число операций
- временная сложность
- объем информации
- корректность

6. Какого рода задачи решаются, как правило, с помощью детерминированных полиномиальных алгоритмов?

- задачи на допустимость
- задачи на оптимальность
- NP-полные задачи
- NP-сложные задачи

7. Какие из перечисленных ниже понятий не являются универсальными алгоритмическими схемами?

- алгоритм Маркова
- алгоритм Ляпунова

- алгоритм Чена
- алгоритм Ван-Хао
- является надмножеством множества B

8. Алгоритм — это ...

- набор правил
- код программы
- вычислительная процедура
- программа на машинном носителе
- псевдокод

9. Установите соответствие (свойство–название).

- А) свойство алгоритма
- Б) сложность алгоритма
- В) способ задания функции
- Г) набор правил
- 1) рекурсия
- 2) полиномиальная
- 3) алгоритм
- 4) понятность

КРИТЕРИИ ОЦЕНКИ

Для оценки уровня полученных знаний при выполнении разработанных тестовых заданий по модулю 2 предлагается использовать следующую шкалу:

85–100 % правильных ответов — оценка «отлично»;

70–85 % правильных ответов — оценка «хорошо»;

55–70 % правильных ответов — оценка «удовлетворительно»;

менее 55 % правильных ответов — оценка «неудовлетворительно».

При анализе алгоритма определяется количество времени, необходимое для его выполнения на ЭВМ. Это приблизительное количество времени, затраченное на выполнение стандартных или приведенных к ним операций, выполняемых алгоритмом.

Для выбора «лучшего» алгоритма необходимо получение оценок или границ для объема памяти и времени работы ЭВМ, которое потребуется алгоритму для обработки конкретных данных.

Глоссарий к модулю 2

Краткость символики способствует постижению сути дела, в то время как словесное выражение перегружает разум.

Т. Клайн

Абстрактный алфавит — это конечная совокупность объектов, называемых буквами или символами этого алфавита.

Алгоритм — это набор правил, указывающих определенные действия, в результате которых входные данные преобразуются в выходные.

Алгоритм — это преобразование слов в абстрактном алфавите.

Алгоритм — это алфавитный оператор, задаваемый с помощью конечной системы правил.

Алгоритм — это алфавитный оператор с правилами, определяющими его действие.

Алгоритмический процесс — это последовательность действий в алгоритме.

Алгоритмический язык — это язык, на основании которого формулируется алгоритм.

Алфавитный оператор — функция, которая задает соответствие между словами входного и выходного абстрактных алфавитов.

Временная сложность алгоритма — это изменение алгоритма во времени или время, затраченное алгоритмом как функция размерности задачи.

Вторая универсальная алгоритмическая модель — это числовая функция, которая каждому алгоритму ставит в соответствие вычисляемую им функцию.

Вычислимая функция — это функция, для которой имеется способ получения ее значений.

Вычислительная сложность алгоритма обычно называется временем работы алгоритма.

Графические схемы алгоритмов (ГСА) представляют собой графическую интерпретацию алгоритмов Ляпунова.

Детерминированность — это свойство алгоритма выполнять на каждом шаге однозначные и независимые от исходных данных действия.

Детерминированные алгоритмы состоят из операций, результаты каждой из которых определены однозначно.

Длина слова в абстрактном алфавите определяется числом символов.

Жадный алгоритм на каждом шаге делает локально оптимальный выбор в предположении, что итоговое решение окажется оптимальным.

Задачей на допустимость называется задача определения хотя бы одного возможного решения.

Задачей на оптимальность считается задача определения допустимого решения, дающего экстремум функции.

Имитирующие алгоритмы создаются на основе описаний действий объектов, наблюдений, зависимости исходных данных от изменяющихся условий.

Интуиция — это знание, полученное на основе опыта, но не подвергнутое научному анализу.

Корректность — это свойство алгоритма при решении определенного класса задач давать правильные результаты для всех допустимых исходных данных.

Конфигурацией машины Тьюринга называется ее полное состояние, по которому можно однозначно определить дальнейшее поведение машины.

Лингвистическая переменная — это кортеж $\text{Л} = \langle \beta, T, X \rangle$, где β — наименование; T — множество значений переменной, представляющих наименования, областью определения которых является X.

Линейный алгоритм — это алгоритм, у которого зависимость времени решения от числа входов носит линейный характер.

Логическими схемами алгоритмов (ЛСА) называют выражения, составленные на основе абстрактного алфавита и состоящие из последовательной цепочки операторов и логических условий.

Массовость — это свойство алгоритма быть применимым для множества исходных данных.

Машина Тьюринга (МТ) — детерминированное устройство, состоящее из трех частей: ленты, считывающего устройства (головки) и управляющего устройства (УУ).

Многозначный алфавитный оператор каждому входному слову может поставить в соответствие некоторое множество выходных слов.

NP-*полные* (Nondeterministic Polynomial — недетерминированные полиномиальные) *задачи* — задачи, для которых не найдены полиномиальные алгоритмы, однако и не доказано, что таких алгоритмов не существует.

Недетерминированные алгоритмы состоят из операций, результаты каждой из которых определены неоднозначно.

Неправильный алгоритм — алгоритм, который для некоторого входа не останавливается или дает неправильный результат.

Нечеткий (*расплывчатый*) *алгоритм* — это упорядоченное множество расплывчатых инструкций, которые при их реализации позволяют получать приближенное решение задач.

Нечеткий алгоритм принятия решений служит для получения приближенного описания стратегии и решающего правила.

Нормальные алгоритмы Маркова (НАМ) основаны на использовании абстрактных алфавитов и марковских подстановок. В качестве исходных данных и конечных результатов используют строки символов, т. е. слова абстрактных алфавитов.

Область определения алфавитного оператора — это множество слов, на котором он определен.

Однозначный алфавитный оператор ставит в соответствие каждому входному слову одно выходное слово.

Операнды — это объекты, над которыми выполняются операции, предписываемые алгоритмами.

Операторный алгоритм Ван-Хао задается последовательностью приказов специального вида. Каждый приказ имеет определенный номер и указание, какую операцию необходимо выполнить над заданным объектом и с каким номером следует далее выполнять действие над результатом этой операции.

Определенность — это свойство алгоритма быть точным и полностью задает все его действия.

Первая универсальная алгоритмическая модель определяется на основе соответствия между словами в абстрактном алфавите.

Подкласс P (*полиномиальные алгоритмы*) — множество всех задач на допустимость, которые могут быть решены на основе детерминированного алгоритма за полиномиальное время.

Поиск в глубину — это алгоритм нахождения решения, когда в каждой исследуемой вершине дерева решений выбирается один из возможных путей и исследуется до конца; при этом другие пути не рассматриваются, пока сохраняется возможность получить конкретный результат. Если такая возможность отсутствует, процесс поиска продолжается от ближайшей вершины дерева решений, имеющей неисследованные пути.

Поиск в ширину предусматривает ветвление на дереве решений от уровня к уровню, причем все подзадачи текущего уровня исследуются раньше, чем любая подзадача следующего уровня.

Полиномиальные алгоритмы — это алгоритмы, у которых время решения находится в полиномиальной зависимости от изменения размера входных данных.

Порочный круг — это определение, в котором новое понятие выводится либо из самого себя, либо из другого понятия, которое было из него выведено.

Правильный алгоритм — это алгоритм, который на любом допустимом (для данной задачи) входе заканчивает работу и выдает результат, удовлетворяющий требованиям задачи.

Примитивно-рекурсивные функции — это функции, полученные без использования оператора построения по первому нулю.

Псевдокод — это описание представляет симбиоз из основных операторов различных языков программирования с использованием комментариев для описания действий алгоритма.

Равные алгоритмы — это алгоритмы, имеющие одинаковые АО и совпадающие системы правил, задающих действие этих алгоритмов на входные слова.

Разветвляющийся алгоритм — это алгоритм, который предусматривает выбор одной из нескольких альтернатив последовательностей действий в зависимости от исходных и промежуточных данных.

Расплывчатая переменная — это кортеж $\Pi = \langle \alpha, X, \widetilde{A} \rangle$, где α — наименование расплывчатой переменной; $X = \{x_1, x_2, \ldots, x_n\}$ — область ее определения; $A = \{\mu_{\widetilde{A}}(x_i) \,|\, x_i\} x_i \in X$ — расплывчатое множество в X, определяющее ограничения на возможные значения Π.

Результативность — это свойство алгоритма, обеспечивающее получение результата через конечное число шагов.

Рекурсия — это способ задания функций, когда ее значения для произвольных значений аргументов выражаются через значения определяемой функции для меньших значений аргументов.

Рекурсивная процедура — это прямое или косвенное обращение к себе в процессе выполнения алгоритма.

Рекурсивные определения состоят из двух частей: 1-я — циклическая, 2-я — прямая часть определения, которая является входом в циклическую.

Рекурсивные функции представляют собой частичный класс вычислимых функций.

Самоизменяющийся алгоритм — это алгоритм, который не только перерабатывает входные слова, но и сам изменяется в процессе такой переработки.

Символы абстрактного алфавита — это буквы, цифры, любые знаки, слова, предложения, любые тексты.

Словесная схема алгоритма (СА) задается через описание и перечисление операторов, аналогичных блокам, используемым структурной схемой алгоритма. Таким образом, словесная запись алгоритма представляет собой перечень операторов, пронумерованный в порядке их исполнения алгоритмом.

Случайный алгоритм — это алгоритм, в процессе выполнения которого предусматривается возможность случайного выбора отдельных действий.

Сортировка вставками основана на выборе очередного числа и вставки его в нужное место, сравнивая с имеющимися и передвигаясь справа налево.

Структурные схемы алгоритмов (ССА) изображаются графически в виде геометрических фигур (блоков). Внутри блока в виде формул или словесно записывается информация, характеризующая выполняемые им функции. Блоки соединяются линиями, символизирующими межблочные связи.

Тезис Тьюринга: любой алгоритм может быть реализован на машине Тьюринга.

Тезис Чёрча: каков бы ни был алгоритм, перерабатывающий наборы целых неотрицательных чисел, всегда существует алгоритм, представляющий рекурсивную функцию, который ему эквивалентен.

Циклический алгоритм — это алгоритм, который выполняется путем многократного повторения некоторой последовательности действий.

Частично-рекурсивные функции — это функции, определенные не для всех возможных значений аргумента.

Численные алгоритмы — это алгоритмы, в которых решения поставленных задач сводятся к арифметическим действиям.

Шаг алгоритма — это каждое отдельное действие алгоритма.

Эвристические алгоритмы (или эвристики) — это алгоритмы, теоретически не обоснованные, но позволяющие сократить количество переборов в пространстве поиска.

Эквивалентные алгоритмы — это алгоритмы с совпадающими АО, но не совпадающими способами их задания.

Экспоненциальными алгоритмами называются алгоритмы, имеющие свойства экспоненциального роста. Экспоненциальные алгоритмы — это алгоритмы, у которых время решения экспоненциально растет по мере увеличения размера входных данных.

Эмпирические алгоритмы — это алгоритмы, основанные на опыте, изучении фактов и опирающиеся на непосредственное наблюдение и эксперимент.

Язык операндов — это язык, предложения которого являются допустимыми для алгоритма вариантами исходных данных.

МОДУЛЬ 3.
АЛГЕБРА ЛОГИКИ (1,5 КРЕДИТА)

Логика — это искусство ошибаться с уверенностью в своей правоте.

Д. Крач

Комплексная цель и задачи изучения модуля

Цель модуля 3 — дать представление о фундаментальных понятиях, базовых принципах и законах таких важных разделов дискретной математики, как алгебра логики, рассмотреть постановку основных задач и проблем, изучить вопросы методологии решаемой проблемы.

В результате освоения модуля 3 студент должен быть готов продемонстрировать следующие *компетенции* и *уровень подготовки*:

- умение применять на практике законы и тождества алгебры логики, доказывать справедливость тождеств алгебры логики;
- знание основных законов алгебры логики;
- навыки решения практических задач алгебры логики, умение представить решаемую конкретную практическую задачу в виде абстрактной алгоритмической модели, пригодной для дальнейшего формального решения на основе существующих языков программирования средствами вычислительной техники.

Самостоятельная работа предусматривает проработку лекций (1,2 часа в неделю), тестирование, а также изучение литературы, формулировку цели работы, объекта и задач исследования, методов, источников и средств библиографического поиска, использованных для достижения поставленной цели.

Модуль включает в себя формулировку цели, проблемное изложение программного лекционного материала, тестовые вопросы для самоконтроля и список литературы. В процессе самостоятельного изучения представленных методических материалов происходит формирование указанных компетенций.

Глава 12
ЭЛЕМЕНТЫ АЛГЕБРЫ ЛОГИКИ

Математика устанавливает отношения между идеями, вскрывая необходимые связи между ними, а такие связи разум постигает лучше всего.

Д. Локк

Алгебра логики, основные логические функции, основные логические операции, основные логические тождества и законы

ЦЕЛИ

Освоив эту главу, учащиеся должны:

- уметь определять основные логические функции;
- уметь строить таблицы истинности логических функций и операций;
- уметь доказывать истинность логических тождеств.

12.1. Логические функции

Логика — это искусство рассуждений. Создатель логики — Аристотель. Наука о законах и формах мышления, так определил логику Аристотель. Лейбниц предложил ввести в логику математическую символику и использовать ее для общих логических построений. Эту идею реализовал в прошлом столетии Джордж Буль и заложил основы математической (символьной) логики.

Основная цель применения в логике символики — это свести операции с логическими заключениями к формальным действиям над символами. При этом исходные положения записываются формулами, которые преобразуются по определенным законам, а полученные результаты истолковываются в определенных понятиях.

Логика — наука, изучающая с формальной точки зрения понятия, методы их определения и преобразования, суждения о них и структуры доказательных рассуждений. Аристотелева логика называется философской или формальной. Она основана на силлогизмах типа:

Все люди смертны.

Сократ — человек.

Следовательно, Сократ смертен.

В XIX веке Пирс сформулировал абдуктивную логику. В ее основу он положил три основных принципа (Непейвода Н.Н.):

— индукция — получение общего закона по множеству частных случаев;

— дедукция — получение из общего утверждения другого общего либо частного;

— абдукция — получение нового частного случая из множества частных случаев.

Основная задача математической логики (МЛ): заменить рассуждения вычислениями. Основа математической логики была заложена в трудах де Моргана, Буля, Пирса, Кантора, Геделя, Маркова, Клини, Ляпунова и многих других ученых. Марков говорил: «Математическая логика — логика по предмету, математика по методу». Язык математической логики, ставший символическим языком современной математики, возник, когда неудобство математического языка для нужд математики было окончательно осознано. Следовательно, математическая логика — наука, изучающая саму математику математическими средствами.

Математическая логика изучает формальную структуру рассуждений математическими средствами. Язык математической логики — исторически первый, точно определенный формальный язык.

Существует двузначная и многозначная логика. Двузначная логика имеет дело с объектами, которые принимают одно из двух возможных значений $(0, 1)$. Объекты, содержащие и принимающие значения из конечного множества, содержащего больше двух элементов, называют многозначными, они обрабатываются аппаратом многозначной логики или сводятся к двузначным объектам.

Высказывание — утверждение об объектах, имеющих однозначный, точно определенный смысл. Например, в году 365 дней; $2 \times 2 = 5$; $3 \times 3 = 9$ и т. д. Некоторые высказывания являются истинными, некоторые ложными. Выделяют безусловно истинные высказывания ($3 \times 3 = 9$) и безусловно ложные ($2 \times 2 = 5$). Высказывания, говорящие о множестве утверждений, называются общими. Атомарными элементарными высказываниями считают такие, которые сообщают единичный факт. Сложные образуются из атомарных применением трех операций:

— логические связки применяются к высказываниям и, в результате, получается новое высказывание. Например, это «A и B»; «не только A, но и B и C»;

— модальности применяются к высказываниям и изменяют наше отношение к ним. Например, «говорят, что $A > B$»; «по словам ученых...»;

— кванторные конструкции применяются к совокупности однородных (отличающихся лишь значениями некоторых параметров) высказываний либо выражений и дают единое высказывание либо выражение, не зависящее от указанных параметров. Например, «большинство...»; «все...»; «для любого...».

Множество предложений, не имеющих четкой и однозначной интерпретации, называются квазивысказываниями.

Определение 1. *Единственными логическими значениями являются истина и ложь, обозначаемые* 1 *или* 0, И *или* Л.

Определение 2. *Логическое значение сложного утверждения зависит лишь от логических значений его компонентов, а не от его смысла.*

Простейшие из выражений, обозначающих предметы, т. е. их конкретные имена, называются константами. Выражение, обозначающее предмет, называется термом.

Для образования высказывания из предметов их соединяют отношением. Например, $a = b$ — двуместное отношение, сопоставляющее двум числам a и b высказывание $a = b$, в частности $1 = 1$ — истинное высказывание, а $1 = 9$ — ложное.

В математической логике часто применяют запись $P(t_1, t_2, \ldots, t_n)$, чтобы обозначить высказывание, образованное применением n-местного отношения P к предметам $t_1, t_2, \ldots, t_n$. Символ P, изображающий отношение, называется предикатом, а $t_1, t_2, \ldots, t_n$ — термы.

Следовательно, для задания элементарных формул необходимо определить предикаты, используемые в теории, и ее термы. В совокупности предикаты, константы, операции, термы составляют словарь или сигнатуру логики.

Одним из основных понятий алгебры логики является понятие *логического высказывания*. Под высказыванием в логике понимают любое утверждение, в отношении которого можно сказать, что оно либо истинно, либо ложно. Например, высказывание «два больше одного» — истинно, а высказывание «пять принадлежит множеству отрицательных чисел» — ложно. Высказывания бывают *простые*, т. е. не зависящие от других высказываний, и *составные*, т. е. состоящие из двух или более простых высказываний. Составные высказывания образуются из простых с помощью *логических операций*. Простые высказывания называют еще *логическими переменными*, а составные — *логическими функциями* (ЛФ) этих переменных.

Особенность ЛФ в том, что они принимают значения в конечных множествах. Область значений ЛФ — это конечная совокупность чисел, символов, любых объектов. ЛФ могут зависеть от одной, двух и любого числа переменных (аргументов). Аргументы могут принимать значения из элементов как конечных, так и бесконечных множеств.

Логическая функция $Y = f(x_1, x_2, \ldots, x_n)$ — это отображение множества наборов (n-мерных векторов, кортежей вида $(x_1, \ldots, x_n)$) на множество ее значений $N = \{\lambda_1, \ldots, \lambda_2\}$. Если аргументы принимают значения из того же множества, что и сама функция, то ее называют однородной.

Объекты с двумя возможными состояниями характеризуются булевыми переменными, которые принимают лишь два различных значения. Отношения между булевыми переменными представляются *булевыми функциями* $y = f(x)$, но y и x принимают свои значения из двухэлементного множества $\{0, 1\}$. Множество всех БФ вместе с операциями отрицания, конъюнкции и дизъюнкции образует *булеву алгебру* (БА). Основные в БА три функции: отрицание, дизъюнкция и конъюнкция. Более сложные формулы получаются замещением входящих в них переменных другими логическими формулами, которые обычно заключаются в скобки, например:

Логическое отрицание (операция НЕ, инверсия) — это операция, результат которой является истинным тогда, когда исходное высказывание ложно, и наоборот.

Обозначается как $\overline{x}$. Таблица истинности этой операции будет иметь следующий вид:

x	$\overline{x}$
0	1
1	0

Логическое сложение (операция ИЛИ, дизъюнкция) — это операция, которая истинна, когда истинно хотя бы одно из составляющих ее высказываний, и ложна — в случае, если оба высказывания ложны. Обозначается как $x \vee y$. Данная операция является многоместной. Таблицу истинности этой операции можем записать так:

x	y	$x \vee y$
0	0	0
1	0	1
0	1	1
1	1	1

Логическое умножение (операция И, конъюнкция) — это операция, которая истинна тогда и только тогда, когда истинны оба составляющих ее высказывания, и ложна во всех остальных случаях. Записывается как $x \,\&\, y$, или $x \wedge y$, или $x\,y$. Данная операция является многоместной. Таблица истинности этой операции будет выглядеть следующим образом:

x	y	$x \,\&\, y$
0	0	0
1	0	0
0	1	0
1	1	1

Импликация — это операция, которая ложна тогда, когда исходное высказывание истинно, а конечное — ложно, и истинна — во всех остальных случаях. Записывается в виде $x \to y$. Таблица истинности этой операции будет иметь следующий вид:

x	y	$x \to y$
0	0	1
1	0	0
0	1	1
1	1	1

Эквивалентность — это операция, которая истинна в том случае, когда значения составляющих ее высказываний совпадают, и ложна — в противном случае. Обозначается следующим образом: $x \leftrightarrow y$. Таблица истинности будет иметь следующий вид:

x	y	$x \leftrightarrow y$
0	0	1
1	0	0
0	1	0
1	1	1

Порядок выполнения логических операций в составных логических высказываниях следующий: сначала инверсия, затем конъюнкция, дизъюнкция, импликация и эквивалентность.

Выражения, с помощью которых записываются высказывания, называются логическими формулами или формулами.

Связки $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ называются связками исчисления высказываний или пропорциональными связками.

Принцип бритвы Оккама: «Не умножай сущностей без необходимости».

Формулы в математической логике — это выражения, обозначающие высказывания. Сложные формулы строятся из более простых при помощи логических связок. Логические связки применяются к высказыванию и в результате дают высказывание.

Логические связки делятся на связки исчисления высказываний (пропорциональные), которые задаются таблицами истинности, и кванторы, которые заставляют переменную пробежать весь универс.

Квантор $\forall$ сочетается со связкой $\Rightarrow$, а квантор $\exists$ — со связкой $\wedge$. Например, утверждение «Некоторые хорошие (Х(x)) студенты (С(x)) становятся учеными (У(x))» можно записать в виде

$$\exists x(\text{Х}(x) \wedge \text{С}(x) \wedge \text{У}(x)).$$

Например, утверждение «Все сторожевые (СС(x)) собаки (С(x)) — злые (Зл(x))»:

$$\forall x(\text{С}(x) \wedge \text{СС}(x)) \Rightarrow \text{Зл}(x).$$

Перевод утверждения «для всякого целого числа есть большее» можно записать:

$$\forall x(x \in \mathbb{Z} \Rightarrow \exists y(y \in \mathbb{Z} \wedge y > x)),$$

где $\mathbb{Z}$ — множество целых чисел.

Основной закон равенства: если A — произвольная формула языка формальной теории, то

$$\forall x, y(x = y \Rightarrow (A(x) \Rightarrow A(y))).$$

Другими словами, свойства равных объектов эквивалентны.

Определение Лейбница: два предмета равны, если они обладают одинаковыми свойствами.

Тождественно истинные формулы в математической логике называются тавтологии. Если формула A — тавтология, то она обозначается $| = A$.

В математической индукции имеется базис — утверждение, что свойство выполнимо для самого маленького из рассматриваемых чисел, и шаг — обоснование перехода от числа n к числу $n + 1$. На языке математической логики метод математической индукции запишется

$$A(0) \wedge \forall n(A(n) \Rightarrow A(n+1)) \Rightarrow \forall n A(n).$$

Итак, отметим, что высказывания принимают логические значения. Единственными логическими значениями будем считать истину и ложь. Высказывания о предметах образуются при помощи отношений или предикатов. Выражения, обозначающие предметы, называются термами. Термы строятся из переменных и констант при помощи операций, или функциональных символов, которые применяются к предметам и в результате дают новый предмет. Отношения (предикаты) применяются к термам и в результате дают высказывание (элементарную формулу).

Примеры решения задач

Пример 12.1. Определить значение следующего составного высказывания:

$$x \wedge (\overline{x} \vee \overline{y} \vee z \wedge y \to \overline{z}) \vee \overline{x} \wedge y,$$

при условии, что $x = 1$; $y = 0$; $z = 0$.

Решение. Поскольку рассматриваемое высказывание состоит из двух частей, соединенных между собой операцией дизъюнкции, то, возможно, окажется достаточно определить значение только одной из составляющих его частей (в случае, если окажется, что эта составляющая истинна). Рассмотрим ту часть, которая содержит меньше переменных. Для этого подставим заданные значения переменных и получим выражение $0 \wedge 0$. Очевидно, что данное выражение ложно. В таком случае продолжим рассмотрение второй части исходного выражения. Здесь у нас имеет место операция конъюнкции. Поскольку переменная x истинна, рассмотрим выражение в скобках:

$$\overline{x} \vee \overline{y} \vee z \wedge y \to \overline{z}.$$

Подставляем в рассматриваемое высказывание значения логических переменных и получаем выражение:

$$0 \vee 1 \vee 0 \wedge 0 \to 1.$$

Как видно, одна из дизъюнкций составляющих данное выражение содержит единицу, следовательно, все выражение истинно и мы можем на этом остановиться. Таким образом, можно сказать, что исходное высказывание истинно.

Пример 12.2. Для составного логического высказывания, заданного в примере 12.1, определить такой набор значений логических переменных x, y, z, при котором значение составного высказывания было бы ложно.

Решение. Поскольку в исходном высказывании имеется операция дизъюнкции, то для того, чтобы это высказывание было ложным, необходимо, чтобы одновременно были ложны высказывания: $x \wedge (\overline{x} \vee \overline{y} \vee z \wedge y \to \overline{z})$ и $\overline{x} \wedge y$. Рассмотрим эти высказывания по отдельности, причем поскольку второе высказывание содержит меньше переменных, начнем именно с него. Очевидно, что для того чтобы было ложно высказывание $\overline{x} \wedge y$, необходимо, чтобы либо $\overline{x}$, либо y, либо обе переменные одновременно были ложны. Таким образом, с учетом инверсии переменной x получили три возможных набора значений переменных x и y:

$$(x = 1, y = 1;\ x = 1, y = 0;\ x = 0, y = 0).$$

Теперь рассмотрим вторую составляющую:

$$x \wedge (\overline{x} \vee \overline{y} \vee z \wedge y \to \overline{z}).$$

Для этого будем поочередно подставлять полученные ранее наборы в составное высказывание, заключенное в скобки.

Возьмем первый набор: $x = 1$, $y = 1$. Поскольку x истинно, то значение высказывания в скобках должно быть ложным. Подставив значения x и y в высказывание в скобках, получим

$$0 \vee 0 \vee z \wedge 1 \to \overline{z}.$$

Это выражение может быть ложно в двух случаях. Во-первых, если переменная равна нулю, в таком случае вне зависимости от результата импликации данное выражение будет ложно. А во-вторых, оно будет ложно, если будет ложным результат импликации. Как известно, операция импликации ложна только в случае, если левая половина импликации истинна, а правая — ложна. Из данного высказывания видно, что результат импликации будет ложным в случае, если $z = 1$.

Таким образом, мы получили два набора значений переменных, при котором исходное высказывание ложно: $x = 1$, $y = 1$, $z = 1$; $x = 1$, $y = 1$, $z = 0$.

Проанализировав аналогичным образом остальные варианты, получим еще два набора логических переменных, при которых исходное высказывание ложно:

$$1)\ x = 0, \quad y = 0, \quad z = 0;$$

$$2)\ x = 0, \quad y = 0, \quad z = 1.$$

Ответ. Полученный результат можно записать в виде таблицы истинности (см. табл. 12.1) логической функции Y:

$$Y = x \wedge (\overline{x} \vee \overline{y} \vee z \wedge y \to \overline{z}) \vee \overline{x} \wedge y.$$

Таблица 12.1

x	y	z	Y
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	0
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	0

12.2. Основные логические тождества и законы

Приведем основные тождества и законы булевой алгебры.

Переместительный закон (*коммутативность*):

$$x \vee y = y \vee x; \quad x \wedge y = y \wedge x. \tag{12.1}$$

Сочетательный закон (*ассоциативность*):

$$\begin{aligned} (x \vee y) \vee z = x \vee (y \vee z) = x \vee y \vee z; \\ (x \wedge y) \wedge z = x \wedge (y \wedge z) = x \wedge y \wedge z. \end{aligned} \tag{12.2}$$

Распределительный закон (*дистрибутивность*):

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z); \qquad (12.3)$$
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

Закон де Моргана:

$$\overline{x \vee y} = \overline{x} \wedge \overline{y}; \quad \overline{x \wedge y} = \overline{x} \vee \overline{y}. \qquad (12.4)$$

Закон двойного отрицания:

$$\overline{\overline{x}} = x. \qquad (12.5)$$

Закон повторяемости:

$$x \vee x = x; \quad x \wedge x = x. \qquad (12.6)$$

Тождества:

$$x \vee 0 = x; \quad x \vee 1 = 1; \quad x \vee \overline{x} = 1; \qquad (12.7)$$
$$x \wedge 0 = 0; \quad x \wedge 1 = x; \quad x \wedge \overline{x} = 0.$$

Правила поглощения:

$$x \vee (x \wedge y) = x; \quad x \wedge (x \vee y) = x. \qquad (12.8)$$

Правила склеивания:

$$(x \wedge y) \vee (x \wedge \overline{y}) = x; \quad (x \vee y) \wedge (x \vee \overline{y}) = x. \qquad (12.9)$$

Следствия из закона де Моргана:

$$x \vee y = \overline{\overline{x} \wedge \overline{y}}; \quad x \wedge y = \overline{\overline{x} \vee \overline{y}}. \qquad (12.10)$$

Тождества:

$$x \to y = \overline{x} \vee y, \qquad (12.11)$$
$$x \to y = \overline{y} \to \overline{x}, \qquad (12.12)$$
$$(x \to y) \wedge (y \to x) = x \leftrightarrow y, \qquad (12.13)$$
$$\overline{x} \to y = \overline{y} \to x = x \vee y. \qquad (12.14)$$

Справедливость всех вышеуказанных законов и тождеств доказывается с помощью таблиц истинности.

Примеры решения задач

Пример 12.3. Доказать с помощью таблиц истинности справедливость распределительного закона:

$$(A \vee B) \wedge C = A \wedge C \vee B \wedge C.$$

Решение. Для доказательства справедливости данного закона приведем таблицы истинности для каждой из частей тождества:

Таблица 12.2

A	B	C	$A \vee B = P_1$	$P_1 \wedge C = Q_1$
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	0	0
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Как видно из табл. 12.2, сначала записываются все возможные наборы значений переменных A, B, C. Так как в рассматриваемом примере имеется три переменных, то существует восемь различных вариантов таких наборов. Затем последовательно выполняются все необходимые логические операции для каждого такого набора. Так, например, в четвертом столбце табл. 12.2 отражен результат промежуточной операции дизъюнкции между переменными A и B. Пятый столбец табл. 12.2 представляет собой результат выполнения логических операций, описанных левой частью исходного выражения. Аналогичное действие производится и для правой части исходного логического выражения.

Таблица 12.3

A	B	C	$A \wedge C = P_1$	$B \wedge C = P_2$	$P_1 \vee P_2 = Q_2$
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	0
1	1	0	0	0	0
0	0	1	0	0	0
1	0	1	1	0	1
0	1	1	0	1	1
1	1	1	1	1	1

Сравнив итоговые выражения Q_1 и Q_2 в табл. 12.2 и 12.3, можно увидеть, что они совпадают, $Q_1 = Q_2$, и, следовательно, исходное тождество справедливо.

Пример 12.4. Доказать с помощью законов и тождеств алгебры логики равенство функций X_1 и X_2, где:

$$X_1 = \overline{A}\,B\,C \vee A\,\overline{B}\,C \vee A\,B\,\overline{C} \vee A\,\overline{B}\,\overline{C} \vee \overline{A}\,B\,\overline{C} \vee A\,B\,C;$$

$$X_2 = A\,\overline{B} \vee B.$$

Примечание. Здесь и далее для краткости будем использовать запись $A\,B$ вместо записи $A \wedge B$.

Решение. Минимизируем выражение X_1. Для этого сгруппируем конъюнкции таким образом, чтобы конъюнкции одной группы отличались друг от друга только на одну составляющую, например:

$$(\overline{A}\,B\,C \vee \overline{A}\,B\,C) \vee (A\,\overline{B}\,C \vee A\,\overline{B}\,\overline{C}) \vee (A\,B\,\overline{C} \vee A\,B\,C).$$

Теперь выносим за скобки одинаковые составляющие:

$$(\overline{A}\,B\,(C \vee \overline{C})) \vee (A\,\overline{B}\,(C \vee \overline{C})) \vee (A\,B\,(C \vee \overline{C})).$$

Так как согласно тождествам алгебры логики $C \vee \overline{C} = 1$, то функцию X_1 можно записать в следующем виде:

$$X_1 = \overline{A}\,B \vee A\,\overline{B} \vee A\,B.$$

Продолжаем группировать составляющие функцию X_1 высказывания. После группировки $\overline{A}\,B$ и $A\,B$ получим:

$$\overline{A}\,B \vee A\,B = B\,(\overline{A} \vee A) = B.$$

Таким образом, мы пришли к выражению: $X_1 = A\,\overline{B} \vee B$.
Следовательно, $X_1 = X_2$, что и требовалось доказать.

12.3. Булевы функции одной и двух переменных

Для представления функций используются таблицы истинности.

Областью определения БФ от n переменных служит множество слов длины n. Они представляют собой все возможные наборы из n двоичных цифр и их общее количество $\nu = 2^{(2^n)}$. При $n = 3$ число возможных наборов $2^{(2^3)} = 256$, а при $n = 5$ число наборов $2^{(2^5)} > 4$ миллиардов. В табл. 12.4 представлена БФ одной переменной:

Таблица 12.4

Y	x		Обозначение	Название
	0	1		
y_0	0	0	0	const 0
y_1	0	1	x	переменная x
y_2	1	0	$\overline{x}$	инверсия x
y_3	1	1	1	const 1

Функции $y_0 = 0$ и $y3 = 1$ — функции константы (тождественный нуль и тождественная единица).

Приведем таблицу булевых функций для двух переменных (табл. 12.5).

Таблица 12.5

Y	x_1 0011 x_2 0101	СДНФ СКНФ	Обозначение	Название
y_0	0000	$(x_1 \vee x_2)(x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2)(\overline{x_1} \vee \overline{x_2})$	0	const 0
y_1	0001	$x_1\ x_2$ $(x_1 \vee x_2)(x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2)$	$x_1\ x_2$	Конъюнкция
y_2	0010	$x_1\ \overline{x_2}$ $(x_1 \vee x_2)(x_1 \vee \overline{x_2})(\overline{x_1} \vee \overline{x_2})$	$x_1 \leftarrow x_2$	Отрицание импликации
y_3	0011	$x_1\ \overline{x_2} \vee x_1\ \overline{x_2}$ $(x_1 \vee x_2)(x_1 \vee \overline{x_2})$	x_1	Переменная x_1
y_4	0100	$\overline{x_1}\ x_2$ $(x_1 \vee x_2)(\overline{x_1} \vee x_2)(\overline{x_1} \vee \overline{x_2})$	$x_2 \leftarrow x_1$	Отрицание обратной импликации
y_5	0101	$\overline{x_1}\ x_2 \vee x_1\ x_2$ $(x_1 \vee x_2)(\overline{x_1} \vee x_2)$	x_2	Переменная x_2
y_6	0110	$\overline{x_1}\ x_2 \vee x_1\ \overline{x_2}$ $(x_1 \vee x_2)(\overline{x_1} \vee \overline{x_2})$	$x_1 \oplus x_2$	Неравнозначность
y_7	0111	$\overline{x_1}\ x_2 \vee x_1\overline{x_2} \vee x_1\ x_2$ $(x_1 \vee x_2)$	$x_1 \vee x_2$	Дизъюнкция
y_8	1000	$\overline{x_1}\ \overline{x_2}$ $(x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2)(\overline{x_1} \vee \overline{x_2})$	$x_1 \downarrow x_2$	Стрелка Пирса
y_9	1001	$\overline{x_1}\ \overline{x_2} \vee x_1\ x_2$ $(x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2)$	$x_1 \sim x_2$	Эквивалентность
y_{10}	1010	$\overline{x_1}\ \overline{x_2} \vee x_1\ \overline{x_2}$ $(x_1 \vee \overline{x_2})(\overline{x_1} \vee \overline{x_2})$	$\overline{x_2}$	Инверсия x_2
y_{11}	1011	$\overline{x_1}\ \overline{x_2} \vee x_1\ \overline{x_2} \vee x_1\ x_2$ $(\vee \overline{x_2})$	$x_2 \rightarrow x_1$	Обратная импликация
y_{12}	1100	$\overline{x_1}\ \overline{x_2} \vee \overline{x_1}\ x_2$ $(\overline{x_1} \vee x_2)(\overline{x_1} \vee \overline{x_2})$	$\overline{x_1}$	Инверсия x_1
y_{13}	1101	$\overline{x_1}\ \overline{x_2} \vee \overline{x_1}\ x_2 \vee x_1\ x_2$ $(\overline{x_1} \vee x_2)$	$x_1 \rightarrow x_2$	Импликация
y_{14}	1110	$\overline{x_1}\ \overline{x_2} \vee \overline{x_1}\ x_2 \vee x_1\ \overline{x_2}$ $(\overline{x_1} \vee \overline{x_2})$	$x_1 \mid x_2$	Штрих Шеффера
y_{15}	1111	$\overline{x_1}\ \overline{x_2} \vee \overline{x_1}\ x_2 \vee x_1\ \overline{x_2} \vee x_1\ x_2$	1	const 1

Как видно, в табл. 12.5 встречаются некоторые неизвестные ранее понятия. Опишем их.

Стрелка Пирса приводится к виду $\overline{x_1 \vee x_2}$ и называется операцией ИЛИ–НЕ.

Штрих Шеффера можно привести к виду $\overline{x_1 \wedge x_2}$ и называется операцией И–НЕ.

Функции отрицания импликации (прямой и обратной) называются иначе *функциями запрета* по x_1 и x_2 соответственно и читаются «не x_2, а x_1» и наоборот.

Всякая функция от меньшего числа переменных содержится среди функций большего числа переменных. Между функциями имеются зависимости, на основании которых можно записать соотношения для констант $0 = \overline{1}$ и $1 = \overline{0}$, для функции одной переменной $x = \overline{\overline{x}}$ и для функций двух переменных: $x_1 x_2 = \overline{x_1 \leftarrow x_2}$, $\overline{x_1 \leftarrow x_2} = \overline{x_1 \rightarrow x_2}$, $x_1 \vee x_2 = \overline{x_1 \sim x_2}$.

Любая функция двух переменных (включая константы) выражается в аналитической форме через совокупность шести функций, содержащей отрицание $\overline{x}$ и любую из каждой пары функций:

$$\{y_0, y_{15}\},\ \{y_1, y_{14}\},\ \{y_2, y_{13}\},\ \{y_6, y_9\},\ \{y_7, y_8\}.$$

Например, такой совокупностью могут быть функции: константа 0, отрицание $\overline{x}$, конъюнкция $x_1 \wedge x_2$, дизъюнкция $x_1 \vee x_2$, эквивалентность $x_1 \sim x_2$, импликация $x_1 \rightarrow x_2$.

Выбранная совокупность шести функций является избыточной. Покажем, что эквивалентность и импликация выражаются через остальные функции этой совокупности:

$$x_1 \rightarrow x_2 = \overline{x_1} \vee x_2, \quad x_1 \sim x_2 = (\overline{x_1} \vee x_2)(x_1 \vee \overline{x_2}).$$

Для этого построим табл. 12.6 и сравним ее с основной таблицей.

Таблица 12.6

x_1	0	0	1	1	
x_2	0	1	0	1	
$\overline{x_1}$	1	1	0	0	
$\overline{x_2}$	1	0	1	0	
$\overline{x_1} \vee x_2$	1	1	0	1	$x_1 \rightarrow x_2$
$x_1 \vee \overline{x_2}$	1	0	1	1	
$(\overline{x_1} \vee x_2)(x_1 \vee \overline{x_2})$	1	0	0	1	$x_1 \sim x_2$

Итак, комплект элементарных функций сокращается до четырех:

- константа 0,
- конъюнкция $x_1 \wedge x_2$,
- отрицание $\overline{x}$,
- дизъюнкция $x_1 \vee x_2$.

Его можно сократить за счет законов де Моргана: $x_1 \vee x_2 = \overline{\overline{x_1} \wedge \overline{x_2}}$, $x_1 \wedge x_2 = \overline{\overline{x_1} \vee \overline{x_2}}$.

Булевы функции могут быть выражены через отрицание и конъюнкцию или через отрицание и дизъюнкцию. Более того, для записи любой БФ достаточно одной из 2 элементарных функций: *стрелки Пирса* или *штриха Шефера*

$$\overline{x} = x \downarrow x = x \,|\, x, \quad x_1 \wedge x_2 = (x_1 \,|\, x_2) \,|\, (x_1 \,|\, x_2),$$

$$x_1 \vee x_2 = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2).$$

С помощью суперпозиции функций, т. е. подстановок в логические формулы вместо переменных, некоторых БФ получают более сложные функции от любого числа переменных.

Например, подставляя в формулу $a \cdot b$ выражение $a = x_1 \vee x_2$, $b = x_2 \to c$, $c = \overline{x_3}$, получим $(x_1 \vee x_2)(x_2 \to \overline{x_3})$.

Таблица 12.7 для сложения формул записывается на основании общей таблицы для элементарных функций.

Например:

Таблица 12.7

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$x_1 \vee x_2$	0	0	1	1	1	1	1	1
$\overline{x_3}$	1	0	1	0	1	0	1	0
$x_2 \to \overline{x_3}$	1	1	1	0	1	1	1	0
$(x_1 \vee x_2)(x_2 \to \overline{x_3})$	0	0	1	0	1	1	1	0

Область определения БФ от n переменных $y = f(x_1, \ldots, x_n)$ рассматривают как совокупность n-мерных векторов, компонентами которого являются буквы 0 и 1.

При $n = 3$ каждый вектор представляется вершиной единичного куба в трехмерном пространстве.

В общем случае совокупность векторов $(x_1, \ldots, x_n)$ отображается на множество вершин n-мерного куба.

БФ отображается на n-мерном кубе путем выделения вершин, соответствующих векторам $(x_1, \ldots, x_n)$, на которых БФ принимает значение 1.

Контрольные вопросы

1. Поясните термин «логическое высказывание».
2. Какие логические операции вы знаете?
3. Назовите порядок выполнения логических операций в составном логическом высказывании.
4. Что называется логической формулой?
5. Каким образом задается логическая формула?
6. Как устанавливается равенство двух логических формул?
7. Как доказывается истинность логических формул, тождеств, законов?
8. Что такое булева функция?
9. Запишите все булевы функции «двух переменных».
10. Определите булевы функции «штрих Шеффера» и «стрелка Пирса».
11. Что такое функция отрицания импликации?
12. Приведите основные логические тождества и законы.
13. В чем заключается закон де Моргана?
14. Каким образом доказывается справедливость основных логических законов и тождеств?

Задания для самостоятельной работы

1. Определить значения следующих составных высказываний:

а) $((x \vee \overline{y} \vee \overline{z})\, \overline{x}\, y) \to (x \vee \overline{y})\, z$, если $x = 0$; $y = 1$; $z = 1$.
б) $\overline{x}\, y \to (\overline{y}\, z \vee x\, y\, \overline{z} \to (x \vee z))$, если $x = 0$; $y = 1$; $z = 0$.
в) $x\, z \to (x \vee \overline{x}\, \overline{y}\, z \vee y\, z) \to y\, \overline{z}$, если $x = 0$; $y = 0$; $z = 1$.
г) $(\overline{x}\, y \to z) \vee (\overline{z} \to x\, \overline{y}\, z)\, x\, z$, если $x = 1$; $y = 0$; $z = 1$.
д) $y(\overline{z} \vee \overline{x}\, y \to x\, z) \vee (x\, \overline{y} \to (\overline{x} \vee z)$, если $x = 1$; $y = 1$; $z = 1$.
е) $((x \vee \overline{y}\, z) \to (\overline{x}\, z \vee y\, z))(x\, \overline{y} \vee y\, \overline{z})$, если $x = 1$; $y = 1$; $z = 1$.
ж) $((x\, \overline{y}) \to (x\, \overline{z} \vee \overline{y}\, z)) \vee ((\overline{x}\, \overline{y} \vee \overline{x}\, z) \to (x\, y\, \overline{z} \vee \overline{x}\, y))$, если $x = 1$; $y = 1$; $z = 0$.

2. Для составных логических высказываний, заданных в п. 1, определить набор значений логических переменных x, y, z, при котором значение составного высказывания было бы: а) ложно; б) истинно.

Ответ записать в виде таблицы истинности.

3. Используя таблицы истинности, доказать справедливость следующих тождеств:

а) $x \vee (y\,z) = (x \vee y)(x \vee z)$;
б) $x\,(y \vee z) = (x\,y) \vee (x\,z)$;
в) $x\,(x \vee z) = x$;
г) $x \vee (x\,y) = x$;
д) $(x\,y) \vee (x\,\overline{y}) = x$;
е) $(x \vee y)(x \vee \overline{y}) = x$;
ж) $x \to y = \overline{x} \vee y$;
з) $x \to y = \overline{y} \to \overline{x}$;
и) $(x \to y)(y \to x) = (x \leftrightarrow y)$;
к) $\overline{x} \to y = \overline{y} \to x = x \vee y$;
л) $x \vee y = \overline{\overline{x}\,\overline{y}}$;
м) $x\,y = \overline{\overline{x} \vee \overline{y}}$;
н) $\overline{x \vee y} = \overline{x}\,\overline{y}$;
о) $\overline{x\,y} = \overline{x} \vee \overline{y}$.

4. Используя законы и тождества алгебры логики, доказать или опровергнуть справедливость следующих тождеств:

а) $\overline{x\,y} \to y = y$;
б) $\overline{x \to y} \to \overline{z \vee \overline{y}} = x \to y$;
в) $\overline{(((x \vee \overline{y})\,\overline{z}) \vee x) \to \overline{y}} = x\,y$;
г) $\overline{(x\,y) \to \overline{\overline{(\overline{(x \vee z)}\,y)}}} = 0$;
д) $\overline{(\overline{((x \vee \overline{y})\,\overline{z})} \vee x) \to y} = \overline{x}\,\overline{y}\,\overline{z}$.

Логика — это искусство рассуждений, наука о законах и формах мышления. Основная цель применения в логике символики — это свести операции с логическими заключениями к формальным действиям над символами. При этом исходные положения записываются формулами, которые преобразуются по определенным законам, а полученные результаты истолковываются в определенных понятиях.

Глава 13

НОРМАЛЬНЫЕ ФОРМЫ БУЛЕВЫХ ФУНКЦИЙ

Логика — это искусство приходить к непредсказуемому выводу.

С. Джонсон

Нормальные формы булевых функций, дизъюнктивная и конъюнктивная нормальные формы, совершенная нормальная форма, способы перехода от нормальных к совершенным нормальным формам, алгебра Жегалкина, функциональная полнота булевых функций

ЦЕЛИ

Освоив эту главу, учащиеся должны:

- знать основные способы перехода от нормальных к совершенным нормальным формам;
- знать основные операции алгебры Жегалкина;
- уметь строить нормальные формы булевых функций;
- уметь определять функционально полные системы булевых функций.

13.1. Дизъюнктивная и конъюнктивная нормальные формы

В булевой алгебре, как и в любой алгебре, существует принцип двойственности. Так, например, взаимно двойственными являются операции дизъюнкции и конъюнкции.

Дизъюнктивная (конъюнктивная) нормальная форма (ДНФ (КНФ)) — это дизъюнкция (конъюнкция) конечного числа различных членов, каждый из которых представляет собой конъюнкцию (дизъюнкцию) отдельных переменных или их отрицаний, входящих в данный член не более одного раза.

За основу для построения *нормальных* (*безынверсных*) *форм БФ* взяты операции конъюнкции и дизъюнкции. Соответственно различают дизъюнктивную нормальную форму (ДНФ) и конъюнктивную нормальную форму (КНФ) БФ.

Дизъюнктивной нормальной формой БФ называется функция, представляющая собой дизъюнкцию конечного числа конъюнкций, каждая из которых содержит простые переменные в прямой или инверсной форме; но не более одного раза для каждой конъюнкции.

Конъюнктивную нормальную форму БФ, по аналогии, можем определить как конъюнкцию некоторого числа дизъюнкций, содержащих не повторяющиеся в рамках каждой конъюнкции переменные в прямой или инверсной форме.

Любая формула приводится к нормальной форме следующим образом:

1) с помощью законов де Моргана формула преобразуется к такому виду, чтобы знаки отрицания относились только к отдельным переменным;

2) на основе первого (второго) дистрибутивного закона формула сводится к дизъюнкции конъюнкций (конъюнкции дизъюнкций);

3) полученное выражение упрощается в соответствии с тождествами (12.7).

Любое высказывание может быть приведено к нормальной (безынверсной) форме. Для этого можно воспользоваться следующими методами.

1. Законами де Моргана пользуются для избавления от знаков инверсии, распространяющихся на несколько переменных сразу.

2. Законы алгебры логики (дистрибутивность и т. д.) используют для приведения высказывания к ДНФ или КНФ.

3. Тождества алгебры логики используют для упрощения полученных выражений.

Элементарной конъюнкцией (*дизъюнкцией*) или *конъюнктивным* (*дизъюнктивным*) *термом* называется выражение, представляющее собой конъюнкцию (дизъюнкцию) любого конечного множества попарно различимых букв или состоящее из одной буквы.

Члены ДНФ (КНФ), представляющие собой элементарные конъюнкции (дизъюнкции) k букв, называются минитермами (макстермами) k-го ранга. Так, например, $x\,y$ — в форме выше минитерм второго ранга, $(x \vee \overline{y})$ — макстерм второго ранга.

Введем также понятие совершенной нормальной формы БФ. Если в каждом члене нормальной формы представлены все переменные (либо в прямом, либо в инверсном виде), то она называется совершенной нормальной формой (СНФ).

Совершенной конъюнктивной нормальной формой БФ (СКНФ) называется логическая функция, представленная в конъюнктивной нормальной форме, причем каждая ее элементарная дизъюнкция содержит все возможные переменные в прямой или инверсной форме.

Соответственно *совершенной дизъюнктивной нормальной формой БФ* (СДНФ) называется логическая функция, представленная в дизъюнктивной нормальной форме, при условии, что каждая составляющая ее элементарная конъюнкция содержит все возможные переменные в прямой или инверсной форме.

Любая БФ, не являющаяся тождественным нулем (единицей), имеет одну и только одну совершенную дизъюнктивную (конъюнктивную) нормальную форму (СДНФ (СКНФ)). Если какой-либо член в ДНФ не содержит переменной x_i, то она вводится тождественным преобразованием:

$$\varphi = \varphi(x_i \vee \overline{x_i}) = \varphi x_i \vee \varphi \overline{x_i}.$$

Совершенные нормальные формы можно записать для любой БФ, исходя из таблицы истинности этой функции, например, так, как показано в таблице БФ двух переменных.

В этом случае СДНФ записывается для тех значений наборов переменных, при которых БФ истина, т. е. принимает значение, равное единице. При этом, если значение переменной в строке равно 1, то переменная входит в конъюнкцию в прямом виде, а если 0 — то в обратном.

В свою очередь, СКНФ записывается для тех наборов переменных, при которых БФ обращается в 0.

Составляющие СКНФ элементарные дизъюнкции записываются следующим образом: если значение переменной в строке таблицы равно 0, то она записывается в прямом виде, если 1 — то в инверсном.

Для совокупности переменных $x_1, \ldots, x_n$ выражение $\widetilde{x}_1\, x_2 \ldots x_n$ называют конституентой единицы, а выражение $\widetilde{x}_1 \vee \widetilde{x}_2 \vee \ldots \vee \widetilde{x}_n$ — конституентой нуля ($\widetilde{x}_i$ означает либо x_i, либо $\overline{x_i}$).

Данная конституента единицы (нуля) образуется при одном соответствующем ей наборе значений переменных, который получается, если все переменные принять равными единице (нулю), а их отрицания — нулю (единице). Например, конституенте единицы $x_1\,\overline{x_2}\,x_3\,x_4$ соответствует набор (1011), а конституенте нуля $\overline{x_1} \vee x_2 \vee \overline{x_3} \vee \overline{x_4}$ — (1011).

В некоторых учебниках на основе конституент единицы (нуля) составляются канонические дизъюнктивные (конъюнктивные) нормальные формы булевых функций. ДНФ (КНФ) называется *канонической* или *совершенной*, если все ее элементарные конъюнкции (дизъюнкции) являются конституентами единицы (нуля) (П.С. Довгий, В.И. Поляков).

Так как СДНФ (СКНФ) является дизъюнкцией (конъюнкцией) конституент единицы (нуля), то можно утверждать, что представляемая ею БФ $f(x_1, x_2, \ldots, x_n)$ обращается в единицу (нуль) только при наборах значений переменных $x_1, x_2, \ldots, x_n$, соответствующих этим конституентам. На остальных наборах эта функция обращается в нуль (единицу), как показано, например, в таблице:

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y	0	1	1	0	1	0	0	1

$$y = \overline{x_1}\,\overline{x_2}\,x_3 \vee \overline{x_1}\,x_2\overline{x_3} \vee x_1\overline{x_2}\,\overline{x_3} \vee x_1\,x_2\,x_3 =$$
$$= (x_1 \vee x_2 \vee x_3)(x_1 \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee x_2 \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee x_3).$$

Примеры решения задач

Пример 13.1. Привести нижеследующие функции к нормальной (безынверсной) форме:

а) $Y_1 = (\overline{A\,B \vee \overline{B}\,C})\,\overline{A\,B}$.

Решение. Применив правило де Моргана, преобразуем функцию Y_1 к виду
$$Y_1 = (\overline{A\,B \vee \overline{B}\,C \vee A\,B}).$$

После применения сочетательного закона можем записать
$$Y_1 = (\overline{A\,B \vee \overline{B}\,C}).$$

Раскрыв инверсию, получим конъюнктивную нормальную форму (КНФ) исходной функции:
$$Y_{1\text{КНФ}} = (\overline{A\,B})(\overline{\overline{B}\,C}) = (\overline{A} \vee \overline{B})(B \vee \overline{C}).$$

Для перехода от КНФ к ДНФ воспользуемся распределительным законом:

$$Y_{1\text{ДНФ}} = (\overline{A} \vee \overline{B})(B \vee \overline{C}) = \overline{A}(B \vee \overline{C}) \vee \overline{B}(B \vee \overline{C}) = \\ = \overline{A}\,B \vee \overline{A}\,\overline{C} \vee \overline{B}\,B \vee \overline{B}\,\overline{C} = \overline{A}B \vee \overline{A}\,\overline{C} \vee \overline{B}\,\overline{C}.$$

Ответ.

$$Y_{1\text{КНФ}} = (\overline{A} \vee \overline{B})(B \vee \overline{C});$$

$$Y_{1\text{ДНФ}} = \overline{A}\,B \vee \overline{A}\,\overline{C} \vee \overline{B}\,\overline{C}.$$

б) $Y_2 = \overline{\overline{(\overline{A} \vee B)}\,\overline{(B \vee C)}} \to \overline{A\,\overline{C}}$.

Решение. Используя закон де Моргана и тождество (12.11), раскрываем скобки и избавляемся от инверсии:

$$\overline{(\overline{A} \vee B)}\,\overline{(B \vee C)} = (A\,\overline{B})(\overline{B}\,\overline{C}) = A\,\overline{B}\,\overline{C}.$$

$$\overline{\overline{(\overline{A} \vee B)}\,\overline{(B \vee C)}} \to \overline{A\,\overline{C}} = \overline{\overline{A\,\overline{B}\,\overline{C}}} \vee \overline{A\,\overline{C}} = A\,\overline{B}\,\overline{C} \vee (\overline{A} \vee C).$$

Ответ.

$$Y_{2\text{ДНФ}} = A\,\overline{B}\,\overline{C} \vee \overline{A} \vee C.$$

Пример 13.2. Пусть задана таблица булевой функции трех переменных. Записать для этой функции СДНФ и СКНФ.

Таблица 13.1

x_1	x_2	x_2	Y
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	1
1	1	1	0

Решение. Для записи совершенной дизъюнктивной нормальной формы (СДНФ) выбираем те наборы переменных x_1, x_2, x_3, при которых функция Y принимает значение, равное 1:

$$Y_{\text{СДНФ}} = x_1\,\overline{x_2}\,\overline{x_3} \vee \overline{x_1}\,x_2\,x_3 \vee x_1\,x_2\,x_3 \vee \overline{x_1}\,x_2\,x_3.$$

Соответственно для записи совершенной конъюнктивной нормальной формы (СКНФ) выбираются наборы переменных, при которых функция Y принимает значение, равное 0:

$$\text{СКНФ} = (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \overline{x_3})(\overline{x_1} \vee x_2 \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}).$$

Полученные выражения можно преобразовать к другому виду на основании свойств булевой алгебры.

Приведем функцию $x\,\overline{y}\,z \vee x\,\overline{z}$ к СДНФ:

$$x\,\overline{y}\,z \vee x\,\overline{z} = x\,\overline{y}\,z \vee x\,\overline{z}\,(y \vee \overline{y}) = x\,\overline{y}\,z \vee x\,y\,\overline{z} \vee x\,\overline{y}\,\overline{z}.$$

Приведем к СКНФ:

$$\begin{aligned}
&\overline{(x \vee y\overline{z})}(x \vee z) = \overline{x}\,\overline{(y\overline{z})}(x \vee z) = \\
&= \overline{x}(\overline{y} \vee z)(x \vee z) = (\overline{x} \vee y\,\overline{y})(\overline{y} \vee z \vee x\,\overline{x})(x \vee z \vee y\overline{y}) = \\
&= (\overline{x} \vee y)(\overline{x} \vee \overline{y})(\overline{y} \vee z \vee x)(\overline{y} \vee z \vee \overline{x})(x \vee z \vee y) \times \\
&\times (x \vee z \vee \overline{y}) = (\overline{x} \vee y \vee z\,\overline{z})(\overline{x} \vee \overline{y} \vee z\,\overline{z})(x \vee \overline{y} \vee z)(\overline{x} \vee \overline{y} \vee z) \times \\
&\times (x \vee y \vee z)(x \vee \overline{y} \vee z) = (\overline{x} \vee y \vee z)(\overline{x} \vee y \vee \overline{z})(\overline{x} \vee \overline{y} \vee z) \times \\
&\times (\overline{x} \vee \overline{y} \vee \overline{z})(x \vee \overline{y} \vee z)(\overline{x} \vee \overline{y} \vee z)(x \vee y \vee z)(x \vee \overline{y} \vee z) = \\
&= (\overline{x} \vee y \vee z)(\overline{x} \vee y \vee \overline{z})(\overline{x} \vee \overline{y} \vee \overline{z})(\overline{x} \vee \overline{y} \vee z)(x \vee \overline{y} \vee z)(x \vee y \vee z).
\end{aligned}$$

13.2. Способы перехода от нормальных к совершенным нормальным формам

Существует два способа перехода от нормальных к совершенным нормальным формам булевых функций.

1. Аналитический способ. Данный способ состоит в пошаговом дополнении каждой компоненты, составляющей нормальную форму до полной, т. е. содержащей все возможные переменные. Для СДНФ при этом производится операция последовательного умножения на логическое выражение вида $(x_i \vee \overline{x_i})$, где x_i — одна из переменных набора $x_1, x_2, x_3, \ldots, x_n$, которые не входят в рассматриваемую компоненту.

В свою очередь, для СКНФ производится последовательное суммирование с логическим выражением вида $x_i\,\overline{x_i}$.

Причем количество проводимых операций умножения или суммирования зависит от разницы между числом переменных, входящих в рассматриваемую компоненту, и общим числом задействованных переменных.

2. Графический способ. Наиболее наглядным способом перехода к совершенным нормальным формам являются *карты*

Карно. Ниже представлены карты Карно для двух, трех и четырех переменных соответственно.

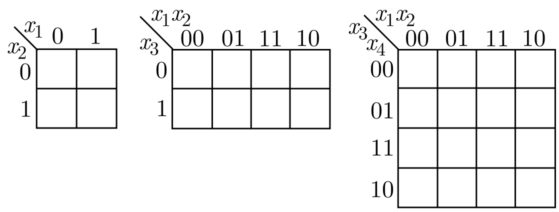


Рис. 13.1. Карты Карно

Как видно, карты Карно составляются таким образом, чтобы значение переменных в соседних клетках отличались друг от друга не более чем на единицу.

Алгоритм перехода к СДНФ можно сформулировать следующим образом:

1) для заданной функции строится карта Карно;

2) в клетках соответствующих компонентам, составляющим ДНФ, ставятся единицы;

3) затем для каждой клетки, имеющей единицу, записываются конъюнкции, содержащие полный набор переменных;

4) полученные конъюнкции соединяются символами дизъюнкции.

Для получения с помощью карт Карно СКНФ необходимо проинвертировать исходную БФ.

Затем для полученного инверсного выражения выполняется алгоритм, приведенный для записи СДНФ.

Полученную в п. 4 вышеуказанного алгоритма СДНФ еще раз инвертируем и получаем в результате искомую СКНФ.

Примеры решения задач

Пример 13.3. Преобразовать в СДНФ и СКНФ следующие переключательные функции:

а) $Y = AB \vee C$.

Решение. Как видно, исходная функция имеет две составляющие, соединенные между собой символом дизъюнкции. Используя законы и тождества алгебры логики, преобразуем эти составляющие до полных. Согласно тождествам алгебры логики, значение выражения типа $B \vee \overline{B}$ всегда равно единице, поэтому добавление такого выражения в исходное не изменяет его значения. Применив к полученному выражению распределительный закон, перейдем к записи СДНФ:

$$A\,B\,(C \vee \overline{C}) = A\,B\,C \vee A\,B\,\overline{C};$$

$$C\,(A \vee \overline{A}) = C\,A \vee C\,\overline{A} = (C\,A \vee C\,\overline{A})(B \vee \overline{B}) = \\ = A\,B\,C \vee \overline{A}\,B\,C \vee A\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,C.$$

Ответ. $Y_{\text{СДНФ}} = A\,B\,C \vee A\,B\,\overline{C} \vee \overline{A}\,B\,C \vee A\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,C.$

б) $Y = A\,(B \vee \overline{C})$.

Решение. Исходная функция состоит из двух составляющих, соединенных символом конъюнкции. Для того чтобы записать исходную функцию в виде СКНФ, необходимо, чтобы каждая составляющая включала в себя полный набор переменных, соединенных знаком конъюнкции. Согласно тождествам алгебры логики значение выражения типа $B\,\overline{B}$ всегда равно нулю, поэтому добавление такого выражения в исходное не изменяет его значения. Применив после этого к полученному выражению распределительный закон, мы легко перейдем к записи СКНФ:

$$A{=}A \vee B\,\overline{B}{=}(A \vee B)(A \vee \overline{B}){=}(A \vee B \vee C\,\overline{C}) \vee (A \vee \overline{B} \vee C\,\overline{C}) = \\ = (A \vee B \vee C)(A \vee B \vee \overline{C})(A \vee \overline{B} \vee C)(A \vee \overline{B} \vee \overline{C});$$

$$B \vee C = B \vee C \vee A\,\overline{A} = (A \vee B \vee C)(\overline{A} \vee B \vee C).$$

Ответ.

$$Y_{\text{СКНФ}} = \\ = (A \vee B \vee C)(A \vee B \vee \overline{C})(A \vee \overline{B} \vee C)(A \vee \overline{B} \vee \overline{C})(\overline{A} \vee B \vee C).$$

в) $Y = A\,(\overline{B} \vee C)(\overline{A} \vee B \vee C)$.

Решение.

$$A = A \vee B\,\overline{B} = (A \vee B)(A \vee \overline{B}) = \\ = (A \vee B \vee C\,\overline{C}) \vee (A \vee \overline{B} \vee C\,\overline{C}) = \\ = (A \vee B \vee C)(A \vee B \vee \overline{C})(A \vee \overline{B} \vee C)(A \vee \overline{B} \vee \overline{C});$$

$$\overline{B} \vee C = \overline{B} \vee C \vee A\,\overline{A} = (A \vee \overline{B} \vee C)(\overline{A} \vee \overline{B} \vee C).$$

Ответ.

$$Y_{\text{СКНФ}} = \\ = (A \vee B \vee C)(A \vee B \vee \overline{C})(A \vee \overline{B} \vee C)(A \vee \overline{B} \vee \overline{C})(\overline{A} \vee \overline{B} \vee C).$$

г) $Y = A \vee \overline{B}C \vee \overline{A}\,B\,C$.

Решение.

$$\overline{B}\,C(A \vee \overline{A}) = A\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,C;$$

$$A\,(C \vee \overline{C}) = C\,A \vee \overline{C}\,A = (C\,A \vee \overline{C}\,A)(B \vee \overline{B}) = \\ = A\,B\,C \vee A\,B\,\overline{C} \vee A\,\overline{B}\,C \vee A\,\overline{B}\,\overline{C}.$$

Ответ. $Y_{\text{СДНФ}} = A\,B\,C \vee A\,B\,\overline{C} \vee A\,\overline{B}\,C \vee A\,\overline{B}\,\overline{C} \vee \overline{A}\,\overline{B}\,C$.

Пример 13.4. Преобразовать с помощью карт Карно переключательную функцию:

а) $Y = A\,B \vee C$ в СДНФ.

Решение. Исходная функция содержит две составляющие: $A\,B$ и C. Строим карту Карно. Конъюнкция $A\,B$ соответствует столбцу 11 в карте Карно. Проставляем единицы во всех ячейках этого столбца. Вторая составляющая C исходной функции соответствует строке 1 в карте Карно. Во всех ячейках данной строки также проставляем единицы. В результате этой операции мы получили карту Карно, изображенную ниже.

$C\backslash A\,B$	00	01	11	10
0	0	0	1	0
1	1	1	1	1

Данная карта имеет пять ячеек, содержащих единицы. Следовательно, СДНФ исходной функции должна содержать пять конъюнкций. Запишем теперь исходную функцию в совершенной дизъюктивной нормальной форме:

$$Y_{\text{СДНФ}} = A\,B\,C \vee A\,B\,\overline{C} \vee \overline{A}\,B\,C \vee A\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,C.$$

б) $Y = (A \vee B \vee C)(A \vee \overline{B} \vee D)$ в СКНФ.

Решение. Найдем инверсное значение функции Y:

$$\overline{Y} = \overline{A}\,\overline{B}\,\overline{C} \vee \overline{A}\,B\,\overline{D}.$$

Построим карту Карно аналогично описанному в предыдущем примере способу.

$CD \backslash AB$	00	01	11	10
00	1	1	0	0
01	1	0	0	0
11	0	0	0	0
10	0	1	0	0

Полученная карта Карно содержит четыре ненулевые ячейки. После этого запишем выражение, являющееся инверсией СКНФ:

$$\overline{Y} = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \vee \overline{A}\,\overline{B}\,\overline{C}\,D \vee \overline{A}\,B\,\overline{C}\,\overline{D} \vee \overline{A}\,B\,C\,\overline{D}.$$

Теперь для того чтобы получить СКНФ, достаточно проинвертировать последнее выражение:

$$Y_{\text{СКНФ}} = (A \vee B \vee C \vee D)(A \vee B \vee C \vee \overline{D}) \times \\ \times (A \vee \overline{B} \vee C \vee D)(\overline{A} \vee B \vee C \vee \overline{D}).$$

13.3. Алгебра Жегалкина

Другая алгебра булевых функций строится на основе операций сложения по модулю 2 и конъюнкции. Она называется алгеброй Жегалкина (АЖ).

Свойства АЖ:

1) коммутативность $x \oplus y = y \oplus x$, $xy = yx$;

2) ассоциативность $x \oplus (y \oplus z) = (x \oplus y) \oplus z$, $x(y\,z) = (x\,y)\,z$;

3) дистрибутивность умножения относительно сложения $x(y \oplus z) = x\,y \oplus x\,z$;

4) свойства констант $x\,1 = x$, $x\,0 = 0$, $x \oplus 0 = x$.

Справедливы также соотношения:

$$\overline{x} = 1 \oplus x, \quad x_1 \vee x_2 = x_1 \oplus x_2 \oplus x_1\,x_2, \quad x_1 \oplus x_2 = x_1\,\overline{x_2} \vee \overline{x_1}\,x_2.$$

Используя эти формулы, можно перейти от АЖ к БА и обратно.

Например:

1) $x\,(\overline{x} \vee y) = x[(1 \oplus x) \oplus y \oplus (1 \oplus x)y] = x(1 \oplus x \oplus y \oplus y \vee x\,y) = x(1 \oplus x \oplus x\,y) = x \oplus x\,x \oplus x\,x\,y = x\,y$.

2) $1 \oplus x \oplus y \oplus x\,y = (1 \oplus x)(1 \oplus y) = \overline{x}\,\overline{y}$.

Через операции алгебры Жегалкина можно выразить все другие БФ. Пример показан в табл. 13.2.

Таблица 13.2

$x_1 \to x_2 = \overline{x_1} \vee x_2 = 1 \oplus x_1 \oplus x_1\,x_2$
$x_1 \sim x_2 = (\overline{x_1} \vee x_2)(x_1 \vee \overline{x_2}) = 1 \oplus x_1 \oplus x_2$
$x_1 \leftarrow x_2 = \overline{x_1 \to x_2} = x_1 \oplus x_1\,x_2$
$x_1 \mid x_2 = 1 \oplus x_1\,x_2$
$x_1 \downarrow x_2 = 1 \oplus x_1 \oplus x_2 \oplus x_1\,x_2$

13.4. Функциональная полнота булевых функций

В алгебре логики из множества $\nu = 2^{2^n}$ различных БФ n переменных $y = f(x_1, \ldots, x_n)$ выделяют пять типов БФ.

1. Функции, сохраняющие константу 0, т. е. такие $f(x_1, \ldots, x_n)$, что $f(0, \ldots, 0) = 0$, их число $\dfrac{1}{2}\upsilon = \dfrac{1}{2}2^{2^n}$.

2. Функции, сохраняющие константу 1, т. е. такие $f(x_1, \ldots, x_n)$, что $f(1, \ldots, 1) = 1$, их число равно половине общего числа всех функций n переменных, т. е. $\dfrac{1}{2}\upsilon$.

3. Самодвойственные функции, т. е. такие, которые принимают противоположные значения на любых двух противоположных наборах. Их число $2^{\frac{1}{2}^{2^n}}$.

4. Линейные функции, т. е. такие, которые представляются в алгебре Жегалкина каноническим многочленом, не содержащим произведений переменных $a_0 + a_1x_1 + \ldots + a_nx_n$, где коэффициенты $a_0, \ldots, a_n$ принимают значения 0 или 1. Так как всего коэффициентов $n + 1$, то число различных линейных функций будет 2^{n+1}.

5. Монотонные функции, т. е. такие, которые для любых двух наборов из множества значений переменных, частично упорядоченного соотношением $(\alpha_1, \ldots, \alpha_n) \leqslant (\beta_1, \ldots, \beta_n)$, при $\alpha_i \leqslant \beta_i$ $(i = 1, 2, \ldots, n)$, удовлетворяют неравенству $f(\alpha_1, \ldots, \alpha_n) \leqslant f(\beta_1, \ldots, \beta_n)$.

Система функций, суперпозицией которых представляется любая функция из некоторого множества БФ, называется функционально полной. Если в такой системе допускаются константы 0 и 1, то ее называют ослаблено функционально полной. Говорят, что ФПС функций образует базис в логическом пространстве. Система функций называется минимально полным базисом, если удаление из нее любой функции превращает эту систему в неполную.

Теорема о функциональной полноте. *Для того чтобы БФ была полной, необходимо и достаточно, чтобы она включала хотя бы одну функцию: несохраняющую константу* 0, *несохраняющую константу* 1, *несамодвойственную, нелинейную, немонотонную.*

Теорему понимают так: одна и та же функция может представлять в функционально полной системе (ФПС) одно или несколько требуемых свойств, если она обладает этими свойствами. Таблица характеризует свойства БФ с позиций функциональной полноты (знаком $\times$ отмечены свойства, которыми обладает данная функция).

БФ	Формулы	Свойства				
		Несохраняющие 0	Несохраняющие 1	Несамодвойственность	Нелинейность	Немонотонность
Константа 0	0		$\times$	$\times$		
Константа 1	1	$\times$		$\times$		
Отрицание	$\overline{x}$	$\times$	$\times$			$\times$
Конъюнкция	$x_1\, x_2$			$\times$	$\times$	
Дизъюнкция	$x_1 \vee x_2$			$\times$	$\times$	
Импликация	$x_1 \to x_2$	$\times$		$\times$	$\times$	$\times$
Эквивалентность	$x_1 \sim x_2$	$\times$		$\times$		$\times$
Отрицание импликации	$x_1 \leftarrow x_2$		$\times$	$\times$	$\times$	$\times$
Сумма по модулю 2	$x_1 \oplus x_2$		$\times$	$\times$		$\times$
Штрих Шеффера	$x_1 \mid x_2$	$\times$	$\times$	$\times$	$\times$	$\times$
Стрелка Пирса	$x_1 \downarrow x_2$	$\times$	$\times$	$\times$	$\times$	$\times$

Из таблицы видно, что операции дизъюнкция и отрицание, конъюнкция и отрицание, штрих Шеффера, стрелка Пирса удовлетворяют теореме о функциональной полноте.

Система операций АЖ (сумма по модулю 2 и конъюнкция) вместе с константой 1 образуют ослабленно-функционально полную систему. Выбрав любую, неэлементарную функцию и дополнив ее одной или несколькими другими функциями так, чтобы они вместе удовлетворяли теореме о функциональной полноте, можно выразить через них все другие БФ.

Контрольные вопросы

1. Что называется безынверсной (нормальной) формой логической функции?

2. Дайте определение дизъюнктивной нормальной формы логической функции.

3. Дайте определение конъюнктивной нормальной формы логической функции.

4. Укажите способы приведения логической функции к нормальной форме.

5. Что такое минитерм k-го ранга?

6. Что называется совершенной дизъюнктивной нормальной формой логической функции?

7. Что называется совершенной конъюнктивной нормальной формой логической функции?

8. Перечислите способы перехода от нормальных к совершенным нормальным формам БФ.

9. В чем смысл аналитического способа перехода к совершенной нормальной форме логической функции?

10. Что представляет собой графический способ перехода к совершенной нормальной форме логической функции?

11. Опишите принципы составления карт Карно.

12. Что представляет собой алгебра Жегалкина?

13. Перечислите пять типов логических функций.

14. Какая система функций называется функционально полной?

15. Дайте определение функциональной полноты.

16. Приведите теорему о функциональной полноте.

Задания для самостоятельной работы

1. Приведите заданную булеву функцию к нормальной (безынверсной) форме:

а) $Y_1 = \overline{x_1 \to (x_2 \vee x_3)}$;

б) $Y_2 = \overline{x_1 \to x_2} \to \overline{x_3\overline{x_1}}$;

в) $Y_3 = \overline{(x_1 \vee x_2) \to \overline{x_1}x_2}$;

г) $Y_4 = \overline{\overline{(\overline{x_1x_2} \vee x_3)} \to \overline{x_2x_3}}$;

д) $Y_5 = \overline{(((x_1\overline{x_2}) \vee x_3)x_1) \to \overline{x_2}}$.

2. Преобразовать следующие логические функции в совершенную нормальную форму аналитическим способом:

а) $Y = A \vee \overline{B}\,C \vee \overline{A}\,B\,\overline{C}$;

б) $Y = \overline{A}\,B \vee A\,\overline{C} \vee \overline{B}\,C$;

в) $Y = A\,\overline{D} \vee B\,C \vee \overline{A}\,\overline{B}\,\overline{C} \vee B\,\overline{C}\,D$;

г) $Y = (A \vee \overline{B} \vee \overline{C})(\overline{A} \vee C \vee D)(\overline{A} \vee B \vee \overline{D})$;

д) $Y = (A \vee \overline{C})(\overline{B} \vee C \vee \overline{D})(B \vee D)$;

е) $Y = \overline{B}(A \vee D)(\overline{A} \vee \overline{B} \vee \overline{D})$.

3. Преобразовать логические функции, заданные в предыдущем пункте, к совершенной нормальной форме с использованием карт Карно.

За основу для построения нормальных форм булевых функций взяты операции конъюнкции и дизъюнкции. Соответственно различают дизъюнктивную и конъюнктивную нормальную формы булевых функций.

Система функций, суперпозицией которых представляется любая функция из некоторого множества булевых функций, называется функционально полной. Функционально полная систем функций образует базис в логическом пространстве.

Глава 14
ЛОГИЧЕСКИЕ СХЕМЫ

Бог сотворил мир на основе
математических принципов.

И. Ньютон

Методы минимизации логических функций, основные элементы, реализующие логические функции, карты Карно, метод Квайна, метод Блейка–Порецкого

ЦЕЛИ

Освоив эту главу, учащиеся должны:

- знать типы логических элементов, реализующих булевы функции;
- уметь применять методы минимизации логических функций;
- уметь строить логические схемы булевых функций.

14.1. Реализация булевых функций

Устройства, реализующие элементарные БФ, называются логическими элементами. Их входы соответствуют булевым переменным, а выходы — реализуемой функции.

На рис. 14.1 приведены логические элементы, реализующие элементарные БФ, и примеры соответствия логических элементов и БФ.

По европейским и американским стандартам основные логические элементы обозначаются несколько иначе. Основные типы вентилей с точки зрения логических функций представлены в табл. 14.1.

В таблице наряду с названием логического элемента и логическим символом приведена его функция в виде логического выражения. Следует обратить внимание на то, что кружок означает логическое отрицание. Три типа логических схем — И, ИЛИ, НЕ, представленные в табл. 14.1, соответствуют операциям булевой алгебры — конъюнкции, дизъюнкции и инверсии. С точ-

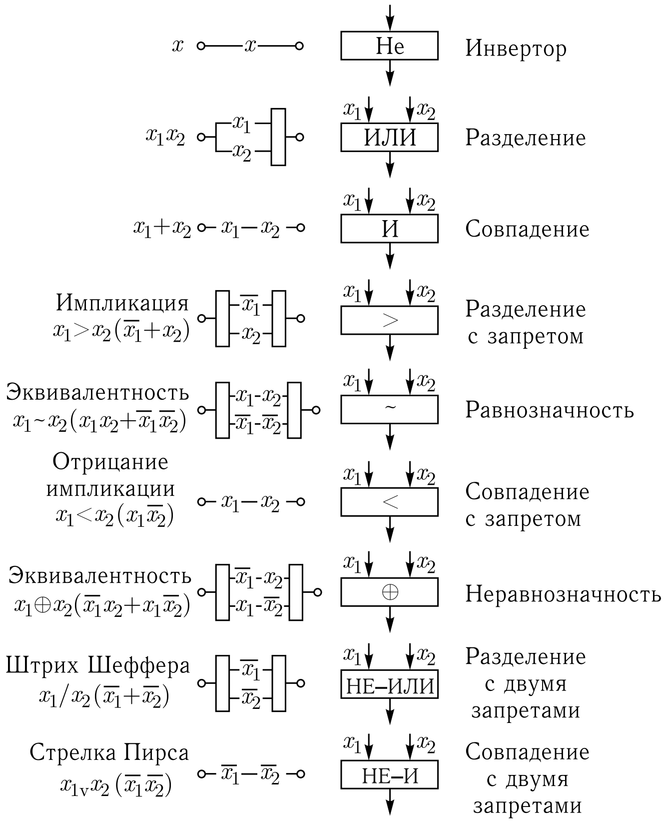


Рис. 14.1. Соответствие логических элементов и булевых функций

ки зрения возможности реализации произвольной логической функции эти три типа элементов образуют полный набор. Как следует из рис. 14.2, используя только элементы И и НЕ, можно получить элемент ИЛИ, поэтому сами по себе элементы И и НЕ составляют полный набор.

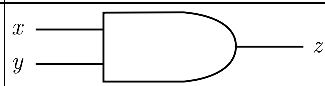
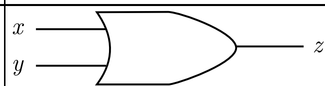
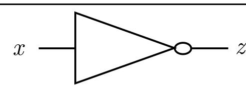
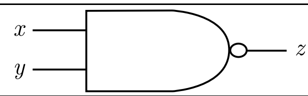
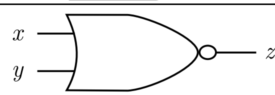
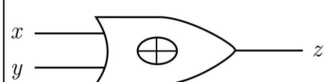
Подобно суперпозиции функций логические схемы образуются суперпозицией элементов посредством объединения их внешних узлов.

Задачи применения ЛС состоят в том, чтобы обеспечить экономичную реализацию БФ в некотором базисе.

Например, ЛС, реализующая функцию

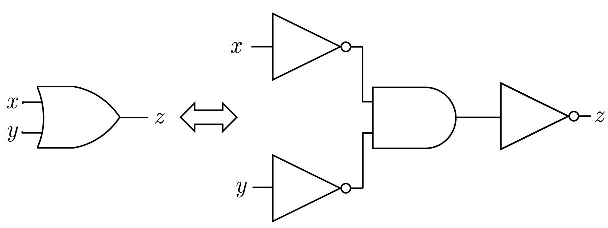
$$Y = (x_1 \mid x_2) \oplus (\overline{x_3} \to x_1).$$

Таблица 14.1

Название вентиля	Символ логической схемы	Логическая функция
И		$z = x_1 \cdot x_2$
ИЛИ		$z = x + y$
НЕТ		$z = \overline{x}$
И–НЕ		$z = \overline{x \cdot y} = \overline{x} + \overline{y}$
ИЛИ–НЕ		$z = \overline{x + y} = \overline{x} \cdot \overline{y}$
ИСКЛЮЧАЮЩЕЕ ИЛИ		$z = x \oplus y =$ $= x \cdot \overline{y} + \overline{x} \cdot y$
СОЕДИНИТЕЛЬ		$z = x$

Переведем в базис (НЕ, И, ИЛИ) (рис. 14.3):

$$y = \overline{x_1\, x_2} \oplus (x_3 \vee x_1) = (x_1\, x_2 (x_3 \vee x_1)) \vee ((\overline{x_1\, x_2})(\overline{x_3 \vee x_1})).$$



$z = x + y$ $\qquad$ $z = \overline{\boldsymbol{x} \cdot \boldsymbol{y}} = x + y$

Рис. 14.2. Пример получения логического элемента ИЛИ

Та же ЛС, но упрощенная (рис. 14.4).

Всегда можно построить много различных ЛС, соответствующих данной ЛФ, они все эквивалентны. Из множества эквивалентных ЛС можно выделить экономичную.

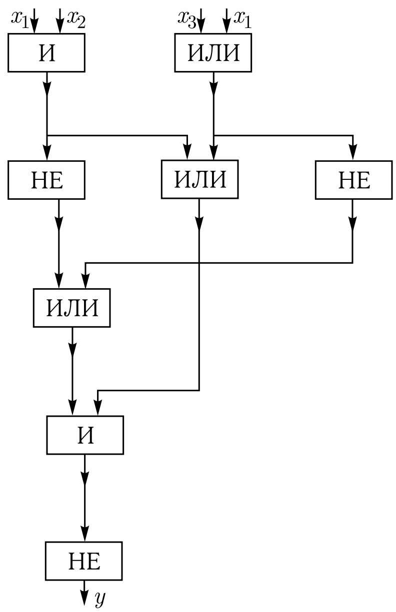


Рис. 14.3. Схема логической функции

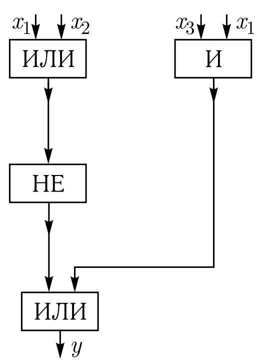


Рис. 14.4. Логическая схема функции после минимизации

Каноническая задача синтеза (т. е. построения) ЛС в булевом базисе сводится к минимизации БФ, т. е. к представлению их ДНФ, содержащей наименьшее число букв. Такие формы называются минимальными.

Логические функции, заданные булевыми выражениями, которые содержат операции логического произведения, суммы и исключающей суммы, а также операцию логического отрицания, легко могут быть реализованы комбинациями логических элементов. Выражения булевой алгебры $(x + y \cdot \overline{z}) \oplus w$, как показано на рис. 14.5, могут быть подвергнуты разложению по «дереву вычислений» в соответствии со скобками и приоритетом операций. Здесь и далее следует понимать, что в булевой алгебре операция $x + y \cdot \overline{z}$ аналогична операции $x \vee y \cdot \overline{z}$.

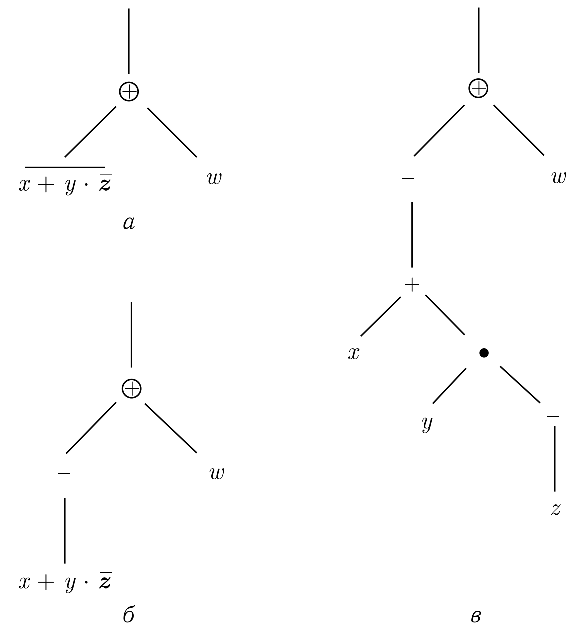


Рис. 14.5. Пример синтеза логической схемы по булевой функции

Разложение по дереву вычислений начинается с последней вычисляемой операции булевого выражения. В примере, приведенном на рис. 14.5, последней в булевом выражении вычисляется операция исключающей суммы ($\oplus$), поэтому можно получить частичное разложение, показанное на рис. 14.5, *а*. Поскольку операция $\oplus$ представляет собой двухместную операцию, на дереве вычислений имеются 2 операнда. Один из этих двух операндов — ω — представляет собой переменную и дальнейшему разложению не подлежит. Другой операнд — $\overline{x + y \cdot \overline{z}}$ — представляет собой булево выражение и может быть разложен дальше. Последней вычисляемой операцией в выражении $\overline{x + y \cdot \overline{z}}$ является операция логического отрицания, и разложение принимает вид, показанный на рис. 14.5, *б*. Дальнейшие аналогичные операции разложения приводят к виду, показанному на рис. 14.5, *в*. Когда операнд логического отрицания ($-$) представляет собой

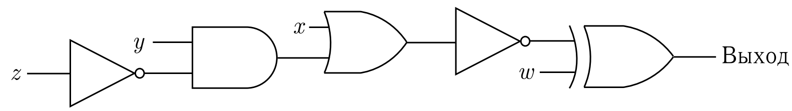


Рис. 14.6. Реализация логической схемы по булевой функции

переменную и дополнение (отрицание) этой переменной может быть использовано в схеме, то в разложении нет необходимости.

Для того чтобы по дереву разложения, соответствующему булеву выражению, сформировать цепь логических элементов, на дереве следует произвести замену операции логического произведения на элемент И, логической суммы — на элемент ИЛИ, исключающей логической суммы — на элемент ИСКЛЮЧАЮЩЕЕ ИЛИ и операции логического отрицания — на элемент НЕ. На рис. 14.6 показан результат такой подстановки, выполненной по дереву (рис. 14.5, *г*). Корень исходного дерева становится выходом, а его ветви — входами.

14.2. Минимизация булевых функций

«При проектировании логической схемы возникает задача минимизации оборудования, используемого в схеме. Простейшей оценкой затрат оборудования на построение схемы, реализующей заданную функцию, является *цена схемы по Квайну* S_Q, которая представляет собой суммарное количество входов во все логические элементы схемы. Задача минимизации цены схемы связана с задачей минимизации булевой функции, описывающей закон функционирования этой схемы. Обычно задача минимизации булевых функций решается с применением нормальных форм (ДНФ или КНФ). Нормальная форма булевой функции, содержащая минимальное количество букв, называется *минимальной нормальной формой* (МДНФ или МКНФ). Задача минимизации булевых функций сводится к нахождению их минимальных покрытий, т. е. покрытий, обладающих минимальной ценой» (П.С. Довгий, В.И. Поляков).

Под *минимизацией* понимают процесс упрощения булевых функций, сведения их к минимально возможной форме. *Минимальной* называется такая форма БФ, которая не допускает уже никаких сокращений. Минимизация БФ производится с помощью различных методов, среди которых можно выделить три.

1. Аналитический (метод последовательного исключения переменных).

2. Метод карт Карно.

3. Метод Куайна–Макласки.

Рассмотрим аналитический метод минимизации булевых функций.

В этом случае для минимизации используют известные законы и тождества алгебры логики. Процесс минимизации заключается в последовательных операциях группировки и выноса за скобки общих переменных с последующим их исключением.

Формула, представленная в ДНФ, упрощается многократным применением операций:

• склеивания $a\,b \vee a\,\overline{b} = a$;

• поглощения $a \vee a\,b = a$, $a \vee \overline{a}\,b = a \vee b$, $(a \vee b)(a \vee \overline{b}) = 0$, $a(a \vee b) = a$, $a(\overline{a} \vee b) = a\,b$.

В результате приходим к аналитическому выражению, когда дальнейшие преобразования невозможны, т. е. получаем тупиковую форму.

Среди тупиковых форм находится и минимальная ДФ. Причем она может быть не единственной. Для проверки, что тупиковая форма минимальна, надо найти все тупиковые формы и сравнить их по числу входящих в них букв.

Например, функция задана СДНФ:

$$Y = \overline{x_1}\,x_2\,\overline{x_3} \vee \overline{x_1}\,x_2\,x_3 \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,x_3 \vee x_1\,x_2\,x_3.$$

Группируя члены и применяя операцию склеивания, получим

$$Y = (\overline{x_1}\,x_2\,\overline{x_3} \vee \overline{x_1}\,x_2\,x_3) \vee (x_1\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,x_3) \vee x_1\,x_2\,x_3 = \\ = \overline{x_1}\,x_2 \vee x_1\,\overline{x_2} \vee x_1\,x_2\,x_3.$$

Но эта форма не минимальна (7 букв).

При повторении в исходной формуле одного члена (это возможно, так как $x \vee x = x$) получим:

$$Y = \overline{x_1}\,x_2 \vee x_1\,\overline{x_2} \vee (x_1\,x_2\,x_3 \vee \overline{x_1}\,x_2\,x_3) = \\ = \overline{x_1}\,x_2 \vee x_1\overline{x_2} \vee x_2\,x_3 \quad \text{(6 букв)}.$$

Каждой вершине n-мерного куба можно поставить в соответствие конституенту единицы. Подмножество отмеченных вершин является отображением на n-мерном кубе БФ от n переменных в СДНФ. На рис. 14.7 показан пример отображения БФ трех переменных на трехмерном кубе.

Для отображения функции от n переменных, представленной в любой ДНФ, необходимо установить соответствие между ее *минитермами* (конституентами единицы) и элементами n-мерного куба.

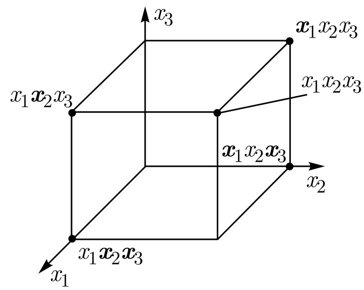


Рис. 14.7. Пример отображения БФ трех переменных на трехмерном кубе

Минитерм $(n-1)$-го ранга φ_{n-1} — это результат склеивания двух минитермов n-го ранга, т. е. $\varphi_{n-1} = \varphi_{n-1}x_i \vee \varphi_{n-1}$. На n-мерном кубе это соответствует замене двух вершин, отличающихся только значениями координаты x_i, *соединяющим эти вершины ребром*.

Говорят, что ребро покрывает инцидентные ему вершины. Таким образом, минитермам $(n-1)$-го порядка соответствуют ребра n-мерного куба. Аналогично устанавливается соответствие минитермов $(n-2)$-го порядка граням n-мерного порядка, каждая из которых покрывает четыре вершины и четыре ребра.

Например: $Y = \overline{x_1}\,x_2 \vee x_1\,\overline{x_2} \vee x_3$. На рис. 14.8 показана графическая реализация данной функции.

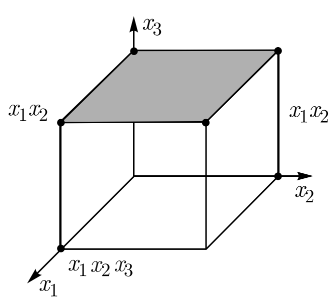


Рис. 14.8. Графическая реализация функции y

Покрытие, соответствующее минимальной форме, называется *минимальным покрытием*.

Например, неминимальная форма $Y = \overline{x_1}\,x_2 \vee x_1\,\overline{x_2} \vee x_1\,x_3 \vee$ $\vee\, x_2\,x_3$ соответствует покрытию куба (рис. 14.9).

А покрытию $Y = \overline{x_1}\,x_2 \vee x_1\,\overline{x_2} \vee x_1\,x_3$ соответствует минимальная форма (рис. 14.10).

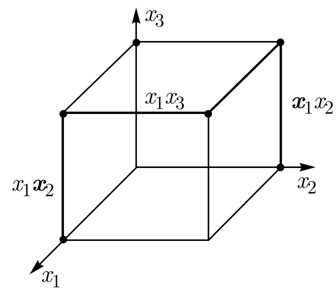


Рис. 14.9. Вариант покрытия куба

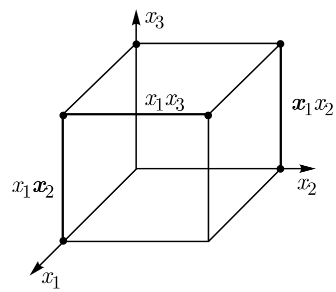


Рис. 14.10. Покрытие куба минимальной формой

Отображение функции на n-мерном кубе наглядно при $n \leqslant 3$. При $n > 4$ требуется много сложных построений.

Примеры решения задач

Пример 14.1. Используя законы алгебры логики, минимизировать логическую функцию:

а) $Y = x_1\,x_2 \vee \overline{x_2}\,\overline{x_3} \vee x_1\,x_2\,x_3 \vee x_1\,\overline{x_2}\,\overline{x_3}$.

Решение. Применив метод группировки, описанный выше, получим:

$$x_1\,x_2 \vee x_1\,x_2\,x_3 = x_1\,x_2(1 \vee x_3) = x_1\,x_2;$$
$$\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,\overline{x_3} = \overline{x_2}\,\overline{x_3}\,(1 \vee x_1) = \overline{x_2}\,\overline{x_3}.$$

Ответ. $Y = x_1\,x_2 \vee \overline{x_2}\,\overline{x_3}$.

б) $Y = x_1\,x_3 \vee x_1\,x_2\,x_3 \vee \overline{x_1}\,x_3 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3} \vee x_2\,x_3$.

Решение. После группировки получим:

$$\overline{x_1}\,x_3 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3} = \overline{x_1}\,(x_3 \vee \overline{x_2}\,\overline{x_3});$$
$$x_1\,x_3 \vee x_1\,x_2\,x_3 = x_1\,x_3(1 \vee x_2) = x_1\,x_3.$$

Применим распределительный закон для логического умножения:

$$\overline{x_1}\,(x_3 \vee \overline{x_2}\,\overline{x_3}) = \overline{x_1}\,(\overline{x_2} \vee x_3)(x_3 \vee \overline{x_3}) = \overline{x_1}\,(\overline{x_2} \vee x_3) = \overline{x_1}\,x_2 \vee \overline{x_1}\,x_3.$$

После этого получим следующее выражение:

$$Y = \overline{x_1}\,\overline{x_2} \vee \overline{x_1}\,x_3 \vee x_2\,x_3 \vee x_1\,x_3.$$

Продолжая процесс группировки, получим:

$$x_3(x_1 \vee \overline{x_1}) \vee \overline{x_1}\,\overline{x_2} \vee x_2\,x_3 = x_3 \vee \overline{x_1}\,\overline{x_2} \vee x_2\,x_3 = \\ = x_3(1 \vee x_2) \vee \overline{x_1}\,\overline{x_2} = x_3 \vee \overline{x_1}\,\overline{x_2}.$$

Ответ. $Y = x_3 \vee \overline{x_1}\,\overline{x_2}$.

в) $Y = A\,\overline{B}\,\overline{C} \vee A\,B\,\overline{C} \vee A\,B\,C$.

Решение. Воспользуемся для минимизации исходной функции знакомым нам методом группировки:

$$Y = A\,\overline{B}\,\overline{C} \vee A\,B\,\overline{C} \vee A\,B\,C = A\,\overline{B}\,\overline{C} \vee A\,B(\overline{C} \vee C) = \\ A\,\overline{B}\,\overline{C} \vee A\,B = A(\overline{B}\,\overline{C} \vee B) = A(\overline{B} \vee B)(\overline{C} \vee B) = A(\overline{C} \vee B).$$

Ответ. $Y = A(\overline{C} \vee B)$.

14.3. Карты Карно

«Карты Карно являются одним из способов таблично-графического представления булевых функций. Они используются для минимизации булевых функций от небольшого числа переменных (как правило, от трех до шести). С использованием карт Карно достаточно просто выделяется минимальное покрытие функции, по которому составляется МДНФ (для единичного покрытия) или МКНФ (для нулевого). Для этой цели на карте выделяются максимальные кубы, представляемые прямоугольниками из клеток, отмеченных единицами или нулями. Две соседние клетки карты образуют 1-куб, четыре — 2-куб, восемь — 3-куб и т. д. Покрытие с минимальной ценой формируется, если каждая существенная вершина будет покрыта максимальным кубом наибольшей размерности и для покрытия всех существенных вершин будет использовано наименьшее число кубов» (П.С. Довгий, В.И. Поляков).

Минимизация с помощью карт Карно производится следующим образом.

1. Строится карта Карно.

2. В соответствии с алгоритмом, описанным выше, карта Карно заполняется единицами.

3. Производится минимизация. В основу процесса минимизации положен следующий принцип: две стоящие рядом единицы могут быть объединены.

При этом переменная, значение которой изменяется, исключается из дальнейшего рассмотрения.

Соответственно при объединении четырех расположенных рядом единиц одновременно исключаются из рассмотрения две переменных, восьми единиц — три и т. д.

4. Полученная в результате процесса минимизации функция будет являться минимальной.

Карты Карно представляют собой специальные таблицы соответствия. Столбцы и строки таблиц соответствуют всевозможным наборам значений не более двух переменных. Причем эти наборы расположены в таком порядке, что каждый последующий отличается от предыдущего значением только одной из переменных, поэтому и соседние клетки таблицы по горизонтали и вертикали отличаются значением только одной переменной. Клетки, расположенные по краям таблицы, также считаются соседними и обладают этим свойством. Клетки наборов, на которых функция принимает значение 1, заполняются единицами (рис. 14.11, 14.12, 14.13).

00	01	11	10

(x_1, x_2)

Рис. 14.11. Карта Карно для двух переменных

		00	01	11	10
(x_1)	0				
	1				
		(x_2, x_3)			

Рис. 14.12. Карта Карно для трех переменных

Например, для функции $Y = x_1\,\overline{x_3} \vee x_2\,x_4\overline{x_1} \vee x_3\,\overline{x_4}$ (НДФ) (рис. 14.14).

Для упрощения строки и столбцы, соответствующие значениям 1 для некоторой переменной, выделяются фигурной скобкой с обозначением этой переменной.

Покрытие соответствует минимальной дизъюнктивной форме.

		00	01	11	10
$(x_1,\, x_2)$	00				
	01				
	11				
	10				
		$(x_3,\, x_4)$			

Рис. 14.13. Карта Карно для четырех переменных

			x_3		
				1	x_2
x_1		1	1	1	
	1	1	1		
	1	1			
		x_4			

Рис. 14.14. Заполнение карты Карно для функции Y

Считывание минитермов осуществляется по правилу: клетки, образующие S-куб, дают минитерм $(n-s)$-го ранга, в который входят те $(n-s)$ переменные, которые сохраняют одинаковые значения на этом S кубе, причем значениям 1 соответствуют сами переменные, а значениям 0 — их отрицания. Переменные, которые не сохраняют свои значения на S-кубе, в минитермах отсутствуют.

Различные способы считывания приводят к различным представлениям функции в ДНФ. Например, по одной и той же карте Карно (рис. 14.15) после считывания можно записать несколько вариантов представления одной функции.

$$Y_1 = x_1\,\overline{x_2} \vee x_1\,x_3 \vee x_2\,x_3 \vee \overline{x_1}\,x_2;$$
$$Y_2 = x_1\,\overline{x_2} \vee x_1\,x_2\,x_3 \vee \overline{x_1}\,x_2;$$
$$Y_3 = x_1\,\overline{x_2} \vee x_2\,x_3 \vee \overline{x_1}\,x_2.$$

			x_2	
			1	1
x_1	1	1	1	
		x_3		

Рис. 14.15. Карта Карно для трех аргументов

Построим карту Карно для пяти аргументов: A, B, C, D, E, F (рис. 14.16).

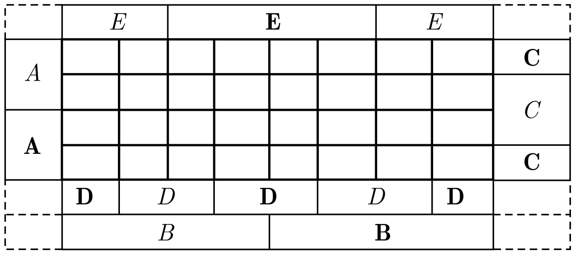


Рис. 14.16. Карта Карно для пяти аргументов

При большом числе переменных БФ представляется *комплексом кубов* (КК).

КК $K(y)$ функции $y = f(x_1, x_2, \ldots, x_n)$ определяется как объединение множеств $K^S(y)$ всех ее S-кубов ($S = 0, 1, \ldots, n$), т. е. $K(y) = \cup K^S(y)$, причем некоторые из $K^S(y)$ могут быть пустыми.

Для записи S-кубов и минитермов функции от n переменных используются слова длины n, буквы которых соответствуют всем n переменным.

Входящие в минитерм переменные называются связанными и представляются значениями, при которых минитерм равен 1 (1 для x_i и 0 для $\overline{x_i}$). Не входящие в минитерм переменные являются свободными и обозначаются x.

Например, 2-куб функции пяти переменных соответствующий минитерму $x_2\, x_5$, запишется:

$$x\,1\,0\,x\,1$$

$$(x_1\, x_2\, x_3\, x_4\, x_5).$$

0-кубы, соответствующие конституентам 1, представляются наборами значений переменных, на которых функция равна 1. Для удобства слова S-кубов располагают в столбцы, а их совокупность — в фигурные скобки.

Например, КК, соответствующий представлению функции на трехмерном кубе, выражается следующим образом:

$$K(y) = K^0 \cup K^1 \cup K^2.$$

$$K^0 = \begin{Bmatrix} 000011 \\ 001101 \\ 010110 \end{Bmatrix}, \quad K^1 = \begin{Bmatrix} 0000xx \\ 0xx101 \\ x01x10 \end{Bmatrix}, \quad K^2 = \begin{Bmatrix} 0 \\ x \\ x \end{Bmatrix}.$$

Графическая интерпретация данного выражения представлена на рис. 14.17, *а*, *б*.

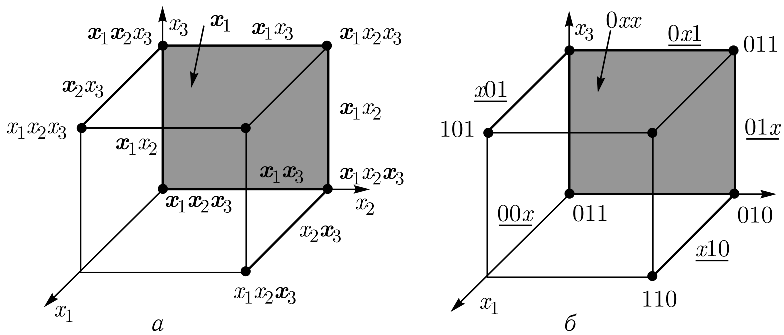


Рис. 14.17. Графическая интерпретация

Комплекс кубов образует максимальное покрытие функции. Исключая из него все те S-кубы, которые покрываются кубами высшей размерности, получаем покрытия, соответствующие тупиковым формам. Так, для примера имеем:

$$c = \begin{Bmatrix} xx0 \\ 01x \\ 10x \end{Bmatrix},$$

которое соответствует функции $y = \overline{x_2}\, x_3 \vee x_2 \vee \overline{x_1}$. В данном случае это покрытие является минимальным.

Приведем постановку задачи минимизации БФ. Минимизация БФ сводится к поиску минимальной ДФ, которой соответствует минимальное покрытие. Общее число букв, входящих в нормальную форму, выражается ценой покрытия:

$$C = \sum_s q_s(n - s),$$

где q_s — число S-кубов, образующих покрытие данной функции от n переменных.

Задача минимизации решается в 2 шага:

1 шаг. Сначала ищут сокращенное покрытие, которое включает все S-кубы максимальной размерности, но не содержит ни одного куба, покрывающего каким-либо кубом того покрытия. Соответствующую ДНФ называют *сокращенной*, а ее минитермы — *простыми импликантами*.

2 шаг. Переход от сокращенной ДНФ к тупиковым ДНФ. Из них выбираются минимальные.

Примеры решения задач

Пример 14.2. Минимизировать с помощью карт Карно БФ:

а) $Y = A\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,\overline{C} \vee \overline{A}\,B\overline{C} \vee A\,B\,C \vee A\,B\,\overline{C}$.

Решение. Строим карту Карно (рис. 14.18).

$C\backslash AB$	00	01	11	10
0	1	1	1	0
1	1	0	1	1

Рис. 14.18. Пример минимизации функции трех переменных (СДНФ)

В полученной карте Карно группируем соседние единицы по горизонтали и вертикали. Так, например, объединив единицы, как показано в вышеуказанной таблице, мы получим следующую минимальную БФ:

$$Y_{1\text{мин}} = \overline{A}\,\overline{B} \vee B\,\overline{C} \vee A\,C.$$

Выполнив группировку иначе, так, например, как показано на рис. 14.19, получим иной вариант минимальной БФ:

$$Y_{2\text{мин}} = \overline{B}\,C \vee \overline{A}\,\overline{C} \vee A\,B.$$

$C\backslash AB$	00	01	11	10
0	1	1	1	0
1	1	0	1	1

Рис. 14.19. Второй вариант группировки переменных

Очевидно, что эти два варианта равнозначны, поскольку содержат одинаковое число СНФ.

Ответ. $Y_{1\text{мин}} = \overline{A}\,\overline{B} \vee B\,\overline{C} \vee A\,C$ либо $Y_{2\text{мин}} = \overline{B}\,C \vee \overline{A}\,\overline{C} \vee A\,B$.

б) $X = A\,\overline{B}\,C\,D \vee A\,\overline{B}\,\overline{C}\,D \vee A\,B\,\overline{C}\,D \vee \overline{A}\,B\,C\,\overline{D} \vee \overline{A}\,B\,C\,D \vee$ $\vee\,\overline{A}\,B\,\overline{C}\,D \vee \overline{A}\,\overline{B}\,C\,D \vee \overline{A}\,\overline{B}\,\overline{C}\,D \vee \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} \vee A\,\overline{B}\,\overline{C}\,\overline{D}$.

Решение. Уже известным нам способом строим карту Карно (рис. 14.20).

$CD\backslash AB$	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	0	1	0	0

Рис. 14.20. Пример минимизации функции четырех переменных (СДНФ)

Затем группируем соседние единицы. В результате группировки получаем следующий результат:

$$Y_{\text{ДНФ мин}} = A \vee \overline{B}\,\overline{C} \vee \overline{A}\,B\,C.$$

Для того чтобы записать по полученной карте Карно $X_{\text{КНФ мин}}$, необходимо провести группировку по нулям (рис. 14.21):

$CD \backslash AB$	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	0	1	0	0

Рис. 14.21. Пример минимизации функции четырех переменных (СКНФ)

После группировки соседних единиц получим выражение:

$$B\,\overline{C}\,\overline{D} \vee \overline{B}\,C\,\overline{D} \vee A\,C\,\overline{D}.$$

Теперь для того чтобы получить $X_{\text{КНФ мин}}$, достаточно проинвертировать полученное выражение:

$$Y_{\text{КНФ мин}} = \overline{B\overline{C}\overline{D} \vee \overline{B}C\overline{D} \vee AC\overline{D}} = \\ = (\overline{B} \vee C \vee D)(B \vee \overline{C} \vee D)(A \vee \overline{C} \vee D).$$

14.4. Метод Куайна–Мак-Класки

Этот метод применяется для минимизации БФ, содержащих небольшое количество переменных (как правило, не более четырех), причем исходная функция должна быть записана в виде СДНФ.

Минимизация выполняется в два этапа.

1-й этап. Сначала производится выделение из СДНФ так называемых сокращенных нормальных форм. Для этого строится таблица, по строкам и столбцам которой записываются все составляющие БФ конъюнкции.

Затем производится анализ полученных конъюнкций и для тех конъюнкций, которые отличаются друг от друга на одну переменную, производится операция склеивания, а на пересечении соответствующей строки и столбца записывается результат операции склеивания, т. е. *сокращенная нормальная форма* (СНФ).

2-й этап. Составляется новая таблица, в которой строкам соответствуют полученные на предыдущем этапе сокращенные нормальные формы, а столбцам — конъюнкции, составляющие исходную функцию.

Затем проверяется условие вхождения сокращенных форм в каждую из имеющихся исходных конъюнкций.

Там, где данное условие выполняется, на пересечении соответствующей строки и столбца ставится знак «плюс».

После этого записывается минимальная функция, которая определяется как минимальный набор сокращенных форм, полностью покрывающий исходный набор конъюнкций.

Метод используется, когда функция задана в СДНФ или таблицей соответствия. Приведение к сокращенной форме осуществляется последовательным применением операции склеивания: $a\,x_i \vee a\,\overline{x_i} = a$.

Множеству конституент единицы, представленных в исходной форме, соответствует совокупность 0-кубов K^0, а операция склеивания — объединению двух 0-кубов, которые отличаются только одной координатой. Результатом такого объединения является 1-куб, в котором различные координаты исходных 0-кубов замещены символом x. Сравнивая попарно все 0-кубы, получаем множество 1-кубов K^1. Применяя к K^1 операцию склеивания, находим множество 2-кубов K^2 и т. д. Это продолжается до тех пор, пока получаемое из K^S очередное K^{S+1} не окажется пустым множеством. В результате множество K^0 преобразуется в комплекс кубов:

$$K = \{K^0, K^1, \ldots, K^S\}, \text{ причем } S \leqslant n.$$

Для выделения из K множества простых импликант $Z \subset K$ при каждой операции склеивания необходимо отмечать меткой (например Δ) те кубы, которые объединяются в кубы высшей размерности.

Очевидно, неотмеченные кубы и образуют множество простых импликант Z. Чтобы уменьшить число сравниваемых пар при операции объединения, целесообразно разбить множество K^S на классы, в каждом из которых содержатся S-кубы с одинаковым числом единиц, и упорядочить эти классы по возрастающему числу единиц. Так как объединяться могут только такие два S-куба, число единиц в которых точно на одну больше или меньше, то достаточно ограничиться попарным сравнением S-кубов одного класса с S-кубами соседнего класса.

На втором шаге для образования минимального покрытия используется таблица покрытий. Ее строки соответствуют импликантам, а столбцы конституентам единицы СДНФ данной функции, которые представляются 0-кубами (вершинами) комплекса кубов. В клетку таблицы записывается метка, если простая импликанта в данной строке покрывает вершину в данном столбце. Экстремалям соответствуют те строки таблицы, которые содержат единственную метку в каком-либо столбце. Удаляя строки экстремалей и все столбцы, в которых эти строки имеют метки, получаем более простую таблицу. На ее основе выбираем простые импликанты, которые дополняют выделенное множество экстремалей до минимального покрытия функции.

Пример 14.3. Задана функция четырех переменных $y = f(x_1, x_2, x_3, x_4)$ таблицей соответствия:

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
y	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	0

Ей соответствует СДНФ: $y = \overline{x_1}\,\overline{x_2}\,x_3\,x_4 \vee \overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4} \vee \overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4} \vee \overline{x_1}\,x_2\,x_3\,x_4 \vee x_1\,\overline{x_2}\,\overline{x_3}\,x_4 \vee x_1\,\overline{x_2}\,x_3\,x_4 \vee x_1\,x_2\,\overline{x_3}\,\overline{x_4} \vee x_1\,x_2\,\overline{x_3}\,x_4$. На рис. 14.22 показано графическое представление этой функции.

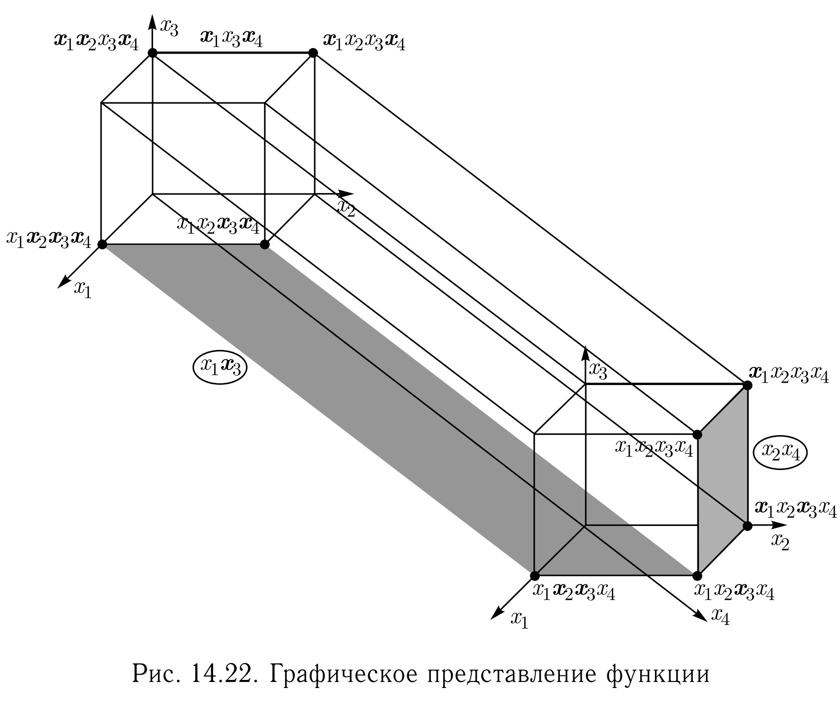


Рис. 14.22. Графическое представление функции

Множество 0-кубов после разбиения и упорядочения запишется так:

$$K^0 = \left|\begin{array}{cc|cccc|cc} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}\right|.$$

Объединяем кубы и отмечаем те из них, которые покрываются кубами большей размерности.

Отображение функции $y = x_1\,\overline{x_3} + x_2\,x_4 + \overline{x_1}\,x_3\,\overline{x_4}$.

$$K^0 = \left|\begin{array}{cc|ccccccc} 0 & x & 0 & 0 & x & 1 & x & 1 & 1 \\ 1 & 1 & x & 1 & 0 & 0 & 1 & x & 1 \\ 0 & 0 & 1 & x & 1 & x & 0 & 0 & 0 \\ x & 0 & 1 & 1 & 1 & 1 & 1 & 1 & x \end{array}\right|,$$

$$K^2 = \left|\begin{array}{c} x \\ 1 \\ 0 \\ x \end{array}\right|.$$

Простым импликантам соответствуют неотмеченные кубы. Составляем таблицу покрытия Z, которому соответствует сокращенная форма:

$$y = \overline{x_1}\,x_3\,x_4 \vee \overline{x_1}\,x_2\,x_4 \vee \overline{x_2}\,x_3\,x_4 \vee x_1\,\overline{x_2}\,x_4 \vee x_1\,\overline{x_3}\,x_4 \vee x_2\,x_3.$$

$Z\backslash K^0$	0100	0011	0101	1001	1100	0111	1011	1101	импликанта
$0x11$		+				+			A
$01x1$			+			+			B
$x011$		+					+		C
$10x1$				+			+		D
$1x01$				+				+	E
$x10x$	+		+		+				F

Извлекаем единственную экстремаль ($x10x$), которой соответствует минитерм $x_2\,\overline{x_3}$, и упрощаем таблицу к следующему виду:

$Z\backslash K^0$	0011	1001	0111	1011
$0x11$	$+$		$+$	
$01x1$			$+$	
$x011$	$+$			$+$
$10x1$		$+$		$+$
$1x01$		$+$		

В качестве дополнительных целесообразно выбрать кубы ($0x11$) и ($10x1$), так как они совместно с экстремалью ($x10x$) образуют покрытие функции, минимальная форма которой имеет вид: $y = \overline{x_1}\,x_3\,x_4 \vee x_1\,\overline{x_2}\,x_4 \vee x_2\,\overline{x_3}$. Отвечающее этой функции минимальное покрытие покажем на карте Карно (рис. 14.23).

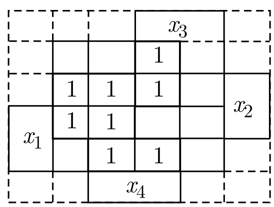


Рис. 14.23. Минимальное покрытие на основе карт Карно

При минимизации методом Куайна–Мак-Класки требуется предварительно представлять функцию в СДНФ, что часто приводит к дополнительным преобразованиям.

Примеры решения задач

Пример 14.4. Минимизировать БФ методом Куайна:

$$Y = A\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,C \vee \overline{A}\,\overline{B}\,\overline{C} \vee \overline{A}\,B\,\overline{C} \vee A\,B\,C \vee A\,B\,\overline{C}.$$

Решение. Находим сокращенные нормальные формы (СНФ):

	$A\,\overline{B}\,C$	$\overline{A}\,\overline{B}\,C$	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,B\,\overline{C}$	$A\,B\,\overline{C}$	$A\,B\,C$
$A\,\overline{B}\,C$	1	$\overline{B}\,C$				$A\,C$
$\overline{A}\,\overline{B}\,C$	$\overline{B}\,C$	1	$\overline{A}\,\overline{B}$			
$\overline{A}\,\overline{B}\,\overline{C}$		$\overline{A}\,\overline{B}$	1	$\overline{A}\,\overline{C}$		
$\overline{A}\,B\,\overline{C}$			$\overline{A}\,\overline{C}$	1	$B\,\overline{C}$	
$A\,B\,\overline{C}$				$B\,\overline{C}$	1	$A\,B$
$A\,B\,C$	$A\,C$				$A\,B$	1

Как видно из таблицы, для исходной функции можно выделить следующие СНФ:

$$Y = \overline{B}\,C \vee \overline{A}\,\overline{B} \vee \overline{A}\,\overline{C} \vee B\,\overline{C} \vee A\,B \vee A\,C.$$

Данное выражение представляет собой *сокращенную нормальную форму*, так как не поддается дальнейшему упрощению.

Теперь переходим ко второму этапу. Расставляем метки на пересечении строк и столбцов в случаях, если данная СНФ входит в данную конъюнкцию.

	$A\ \overline{B}\ C$	$\overline{A}\ \overline{B}\ C$	$\overline{A}\ \overline{B}\ C$	$\overline{A}\ B\ \overline{C}$	$A\ B\ \overline{C}$	$A\ B\ C$
$\overline{B}\ C$	+	+				
$\overline{A}\ \overline{B}$		+	+			
$\overline{A}\ \overline{C}$			+	+		
$B\ \overline{C}$				+	+	
$A\ B$					+	+
$A\ C$	+					+

Теперь для того чтобы записать минимизированную функцию X, необходимо решить задачу минимального покрытия всех столбцов полученной таблицы минимальным количеством строк.

Из таблицы видно, что такое перекрытие достигается двумя способами:

$$Y_{1\text{мин}} = \overline{B}\,C \vee \overline{A}\,\overline{C} \vee A\,B \quad \text{либо} \quad Y_{2\text{мин}} = \overline{A}\,\overline{B} \vee B\,\overline{C} \vee A\,C.$$

Если исходить из произвольной ДНФ, то для получения промежуточной сокращенной формы используется *метод Блейка–Порецкого*. Он основан на тождестве $ac \vee b\overline{c} = ac \vee b\overline{c} \vee ab$, называемом операцией обобщенного склеивания.

Действительно,

$$ac \vee b\overline{c} = ac \vee abc \vee b\overline{c} \vee ab\overline{c} = ac \vee b\overline{c} \vee ab(c \vee \overline{c}) = ac \vee b\overline{c} \vee ab.$$

Входящие в это тождество буквы могут представлять любые булевы формулы, в частности конъюнкции переменных.

Произвольная ДНФ приводится к сокращенной с применением всех возможных обобщенных склеиваний с последующим устранением минитермов на основе операции поглощения: $a \vee ab = a$.

14.5. Переход от булевых функций к простейшим комбинационным схемам

Проиллюстрируем взаимосвязь между булевыми функциями и простейшими комбинационными схемами.

Внимательно проанализировав предыдущий материал, можно уверенно сказать, что любую булеву функцию можно представить в виде элементарной логической (комбинационной) схемы.

В этом случае каждая переменная, входящая в состав булевой функции, будет отображена в логической схеме в виде переключателя.

Операция конъюнкции логических переменных в исходной булевой функции будет соответствовать последовательному соединению переключателей, отображающих данные логические переменные, а операция дизъюнкции, в свою очередь, соответствует параллельному соединению переключателей и ветвей логической схемы.

Мы также в случае необходимости вправе совершить и обратный переход от логической схемы к записи булевой функции.

Примеры решения задач

Пример 14.5. Упростить логическую схему, изображенную на рис. 14.24.

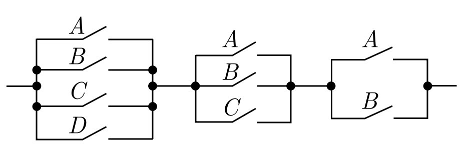


Рис. 14.24. Пример логической схемы

Решение. На основании вышеизложенных правил можем записать исходную логическую схему в виде следующей логической функции:

$$Y = (A \vee B \vee C \vee D)(A \vee B \vee C)(A \vee B).$$

В дальнейшем для упрощения этой функции воспользуемся уже известными нам законами и тождествами алгебры логики.

Легко заметить, что выражения в скобках отличаются друг от друга на одну переменную. Поэтому, по всей видимости, для минимизации данного выражения можно использовать правила поглощения (1.8).

Рассмотрим первые два выражения:

$$(A \vee B \vee C \vee D) \quad \text{и} \quad (A \vee B \vee C).$$

Так как в обеих частях присутствует выражение $A \vee B \vee C$, то будем считать это выражение равным X.

После этого выражение

$$(A \vee B \vee C \vee D)(A \vee B \vee C).$$

можно преобразовать к виду $(X \vee D)\,X$.

Теперь, применив тождество (1.8), мы получим

$$(X \vee D)\,X = X.$$

Следовательно, исходная функция принимает вид

$$Y = (A \vee B \vee C)(A \vee B).$$

Теперь вновь ищем выражение, повторяющееся в обеих частях функции Y. В данном случае это будет выражение $(A \vee B)$, поэтому принимаем его равным Z, и, подставив его в функцию Y, получим: $Y = (Z \vee C)\,Z$.

Применив к данному выражению тождество (1.8), получим: $(Z \vee C)\,Z = Z$.

Таким образом, в результате проведенных преобразований исходная функция свелась к следующему виду:

$$Y = A \vee B.$$

Тогда ее можно представить в виде логической схемы (рис. 14.25):

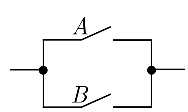


Рис. 14.25. Логическая схема после упрощения

Контрольные вопросы

1. Назовите основные логические элементы, реализующие элементарные булевы функции.

2. Каким образом основные логические элементы, реализующие элементарные булевы функции, изображаются по американским и европейским стандартам?

3. Приведите пример реализации булевой функции комбинациями логических элементов.

4. Перечислите способы минимизации булевых функций.

5. В чем заключается аналитический способ минимизации логических функций?

6. Поясните механизм минимизации логических функций с использованием карт Карно.

7. Поясните принцип минимизации БФ методом Куайна–Мак-Класки.

8. Что называется сокращенной нормальной формой булевой функции?

9. Что такое комплекс кубов?

10. В чем заключается задача минимизации булевых функций?

11. Каким образом осуществляется переход от логических схем к булевым функциям и обратно?

12. Приведите примеры взаимосвязи булевых функций с простейшими логическими схемами.

Задания для самостоятельной работы

1. Минимизировать следующие логические функции аналитическим способом:

а) $Y = A\,B\,C \vee \overline{A}\,\overline{B}\,C \vee A\,B\,\overline{C} \vee \overline{A}\,B\,C \vee \overline{A}\,\overline{B}\,\overline{C}$;

б) $Y = A\overline{B}\,\overline{C}\,D \vee \overline{A}\,\overline{B}\,C\,\overline{D} \vee \overline{A}\,\overline{B}\,C\,D \vee \overline{A}\,\overline{B}\,\overline{C}\,D \vee A\,\overline{B}\,C\,D \vee$
$\vee A\,B\,C\,D$;

в) $Y = A\,\overline{B}\,\overline{C}\,D \vee \overline{A}\,\overline{B}\,C\,\overline{D} \vee \overline{A}\,\overline{B}\,C\,D \vee \overline{A}\,\overline{B}\,\overline{C}\,D \vee A\,\overline{B}\,C\,D \vee$
$\vee A\,B\,C\,D \vee \overline{A}\,B\,C\,D \vee A\,\overline{B}\,\overline{C}\,\overline{D}$;

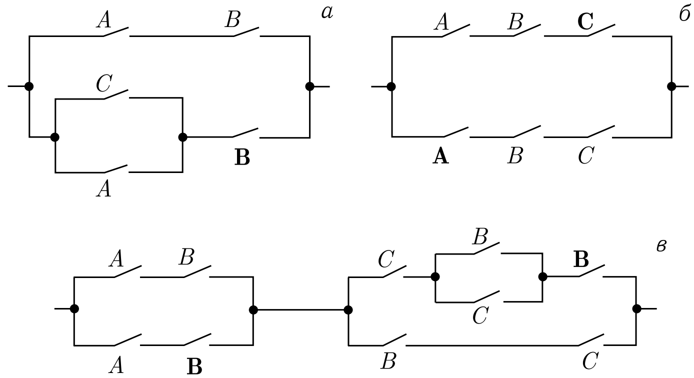
г) $Y = (\overline{A} \vee B \vee C)(A \vee \overline{B} \vee C)(A \vee \overline{B} \vee \overline{C})(\overline{A} \vee B \vee \overline{C}) \times$
$\times (\overline{A} \vee \overline{B} \vee \overline{C})$;

д) $Y = (\overline{A} \vee \overline{B} \vee C \vee \overline{D})(\overline{A} \vee \overline{B} \vee \overline{C} \vee D)(\overline{A} \vee \overline{B} \vee \overline{C} \vee \overline{D}) \times$
$\times (\overline{A} \vee B \vee C \vee D)(A \vee B \vee \overline{C} \vee D)(A \vee B \vee \overline{C} \vee \overline{D})(A \vee B \vee C \vee D)$.

2. Минимизировать логические функции, приведенные в предыдущем примере, с помощью карт Карно.

3. Минимизировать логические функции, приведенные в предыдущем примере, методом Куайна–Мак-Класки.

4. Упростить следующие логические схемы:



Устройства, реализующие элементарные БФ, называются логическими элементами. Их входы соответствуют булевым переменным, а выходы — реализуемой функции.

Под минимизацией понимают процесс упрощения булевых функций, сведения их к минимально возможной форме. Минимальной называется такая форма БФ, которая не допускает уже никаких сокращений.

Тестовые задания к модулю 3

1. Какие из перечисленных ниже понятий являются операциями булевой алгебры?

- транзитивность
- эквивалентность
- дистрибутивность
- импликация

2. Какие из перечисленных ниже понятий являются логическими функциями?

- цепь Гамильтона
- стрелка Пирса
- штрих Шеффера
- цикл Эйлера

3. Каким образом доказывается истинность логических формул, тождеств, законов?

- выполнением преобразований
- составлением таблицы истинности
- путем подстановок
- путем упрощения

4. Для того чтобы базис булевых функций был функционально полным, необходимо и достаточно, чтобы он включал в себя

- функцию, сохраняющую константу 0, несамодвойственную, линейную и монотонную
- функцию, не сохраняющую константу 1, самодвойственную, линейную и немонотонную
- функцию, не сохраняющую константу 1, самодвойственную, функцию, не сохраняющую константу 0 и немонотонную
- функцию, не сохраняющую константу 1, несамодвойственную, функцию, не сохраняющую константу 0 и немонотонную

5. Какие из перечисленных ниже методов не используют при минимизации логических выражений?

- метод Блейка–Порецкого
- метод Куайна–Мак-Класки
- метод неопределенных коэффициентов
- метод группировки

6. Карты Карно используются для

- минимизации БФ
- вычисления БФ

- записи совершенной нормальной формы БФ
- записи нормальной формы БФ

7. Укажите последовательность логических операций в составном логическом высказывании.

- дизъюнкция
- импликация
- инверсия
- эквивалентность
- конъюнкция

8. Какое из перечисленных ниже выражений не является записью функции Веба для двух переменных?

- $(x_1 \vee x_2)\,\&\,(x_1 \vee \overline{x_2})\,\&\,(\overline{x_1} \vee x_2)$
- $(x_1 \vee \overline{x_2})\,\&\,(\overline{x_1} \vee x_2)\,\&\,(\overline{x_1} \vee \overline{x_2})$
- $\overline{x_1}\,\&\,\overline{x_2}$
- $x_2 \downarrow x_2$

9. При каких значениях переменных x, y, z составное логическое высказывание $(x\,\&\,z \to (\overline{x}\,\&\,\overline{y}\,\&\,z) \to y\,\&\,\overline{z}\,)$ будет ложным?

- $x = 0,\ y = 1,\ z = 0$
- $x = 1,\ y = 1,\ z = 1$
- $x = 0,\ y = 0,\ z = 0$
- $x = 1,\ y = 1,\ z = 0$

10. Укажите, какие из нижеследующих логических тождеств неверны

- $\overline{X \cdot Y} \to Y = Y$
- $\overline{X \to Y} \to \overline{Z \vee \overline{Y}} = X \to Y$
- $\overline{(((X \vee \overline{Y}) \cdot \overline{Z}) \vee X) \to \overline{Y}} = X \cdot Y$
- $\overline{(\overline{((X \vee \overline{Y}) \cdot \overline{Z})} \vee X) \to Y} = \overline{X} \cdot \overline{Y} \cdot \overline{Z}$

11. Какие из перечисленных ниже выражений представляют собой нормальную форму логической функции?

- $(x_1 \vee x_2)\,\&\,(x_1 \vee \overline{x_2})\,\&\,(\overline{x_1} \vee x_2)$
- $(x_1 \vee \overline{x_2})\,\&\,(\overline{x_1} \vee x_2)\,\&\,(\overline{x_1} \vee \overline{x_2})$
- $\overline{(X \to Y)} \to \overline{Z \cdot \overline{X}}$
- $\overline{X \to (Y \vee Z)}$
- графиком

12. Установите соответствие (название–обозначение).

А) эквивалентность
Б) инверсия

В) дизъюнкция
Г) импликация
Д) конъюнкция
1) &
2) $\vee$
3) $\leftrightarrow$
4) $\rightarrow$
5) $^{-}$

13. Установите соответствие (название–обозначение).

А) стрелка Пирса
Б) неравнозначность
В) штрих Шеффера
Г) отрицание импликации
Д) импликация
1) $|$
2) $\leftarrow$
3) $\downarrow$
4) $\rightarrow$
5) $\oplus$

14. Установите соответствие (название–обозначение).

А) эквивалентность
Б) инверсия
В) неравнозначность
Г) импликация
Д) отрицание импликации
1) $\leftarrow$
2) $\oplus$
3) $\leftrightarrow$
4) $\rightarrow$
5) $^{-}$

15. Установите соответствие (название–обозначение).

А) конъюнктивная нормальная форма
Б) дизъюнктивная нормальная форма
В) произвольная форма
Г) совершенная дизъюнктивная нормальная форма
Д) совершенная конъюнктивная нормальная форма

1) $(x_1 \,\&\, x_3) \vee (x_1 \,\&\, \overline{x_2}) \vee (\overline{x_1} \,\&\, x_3)$

2) $(x_1 \vee x_2) \,\&\, (x_1 \vee \overline{x_2}) \,\&\, (\overline{x_1} \vee x_2)$

3) $(x_1 \,\&\, x_2) \vee (x_1 \,\&\, \overline{x_2}) \vee (\overline{x_1} \,\&\, x_2)$

4) $(x_1 \vee x_3) \,\&\, (x_1 \vee \overline{x_2}) \,\&\, (\overline{x_1} \vee x_3)$

5) $\overline{(X \to Y)} \vee \overline{Z \,\&\, \overline{\overline{X}}}$

КРИТЕРИИ ОЦЕНКИ

Для оценки уровня полученных знаний при выполнении разработанных тестовых заданий по модулю 3 предлагается использовать следующую шкалу:

85–100 % правильных ответов — оценка «отлично»;

70–85 % правильных ответов — оценка «хорошо»;

55–70 % правильных ответов — оценка «удовлетворительно»;

менее 55 % правильных ответов — оценка «неудовлетворительно».

Глоссарий к модулю 3

Булева алгебра образуется из множества всех БФ вместе с операциями отрицания, конъюнкции и дизъюнкции.

Булевы функции $y = f(x)$ представляют собой отношения между булевыми переменными, причем y и x принимают свои значения из двухэлементного множества $\{0, 1\}$.

Высказывание — это любое утверждение, в отношении которого можно сказать, что оно либо истинно, либо ложно.

Дизъюнктивной нормальной формой БФ называется функция, представляющая собой дизъюнкцию конечного числа конъюнкций, каждая из которых содержит простые переменные в прямой или инверсной форме; но не более одного раза для каждой конъюнкции.

Импликация — это операция, которая ложна тогда, когда исходное высказывание истинно, а конечное — ложно, и истинна во всех остальных случаях.

Конъюнктивная нормальная форма БФ — конъюнкция некоторого числа дизъюнкций, содержащих не повторяющиеся в рамках каждой конъюнкции переменные в прямой или инверсной форме.

Логические функции (ЛФ) — это составные логические высказывания на основе логических переменных.

Логическое отрицание — это операция, результат которой является истинным тогда, когда исходное высказывание ложно, и наоборот.

Логическое сложение — это операция, которая истинна, когда истинно хотя бы одно из составляющих ее высказываний, и ложна в случае, если все высказывания ложны.

Логическое умножение — это операция, которая истинна тогда и только тогда, когда истинны все составляющие их высказывания, и ложна во всех остальных случаях.

Минимальная форма БФ — это форма, которая не допускает никаких сокращений.

Минимизация — это процесс упрощения булевых функций, сведения их к минимально возможной форме.

Минимальным покрытием называется покрытие, соответствующее минимальной форме.

Минитерм — это конституента единицы.

Минитерм $(n-1)$-го ранга φ_{n-1} — это результат склеивания двух минитермов n-го ранга, т. е. $\varphi_{n-1} = \varphi_{n-1}x_i + \varphi_{n-1}\neg x_i$.

Операция обобщенного склеивания: $ac \vee b\overline{c} = ac \vee b\overline{c} \vee ab$.

Простые высказывания — это высказывания, не зависящие от других высказываний. Простые высказывания называют еще *логическими переменными*.

Простые импликанты — это минитермы сокращенной ДНФ.

Совершенной дизъюнктивной нормальной формой БФ (СДНФ) называется логическая функция, представленная в дизъюнктивной нормальной форме, при условии, что каждая составляющая ее элементарная конъюнкция содержит все возможные переменные в прямой или инверсной форме.

Совершенной конъюнктивной нормальной формой БФ (СКНФ) называется логическая функция, представленная в конъюнктивной нормальной форме, при условии, что каждая составляющая ее элементарная дизъюнкция содержит все возможные переменные в прямой или инверсной форме.

Сокращенной ДНФ называют сокращенное покрытие, которое включает все S-кубы максимальной размерности, но не содержит ни одного куба, покрывающего каким-либо кубом того покрытия.

Составные высказывания состоят из двух или более простых высказываний. Составные высказывания образуются из простых с помощью *логических операций*.

Стрелка Пирса — приводится к виду $\overline{x_1 \vee x_2}$ и называется операцией ИЛИ–НЕ.

Теорема о функциональной полноте — для того чтобы БФ была полной, необходимо и достаточно, чтобы она включала хотя бы одну функцию: не сохраняющую константу 0, не сохраняющую константу 1, несамодвойственную, немонотонную.

Штрих Шеффера — можно привести к виду $\overline{x_1 \wedge x_2}$ и называется операцией И–НЕ.

Эквивалентность — это операция, которая истинна в том случае, когда значения составляющих ее высказываний совпадают, и ложна — в противном случае.

МОДУЛЬ 4.
ОСНОВЫ ТЕОРИИ ГРАФОВ (2 КРЕДИТА)

> Вначале любая оригинальная теория признается абсурдной, потом — верной, потом — самоочевидной и незначительной и, наконец, столь важной и самобытной, что бывшие критики присваивают ее себе.
>
> *У. Джеймс*

Комплексная цель и задачи изучения модуля

Цель модуля 4 — дать представление о фундаментальных понятиях, базовых принципах и законах одного из важнейших разделов дискретной математики — теории графов, рассмотреть постановку основных задач и проблем, изучить вопросы методологии решаемой проблемы.

В результате освоения модуля 4 студент должен быть готов продемонстрировать следующие *компетенции* и *уровень подготовки*:

- знание основных понятий теории графов;
- умение оценивать и определять временную сложность алгоритмов решения задач на графах;
- знание основных определений и понятий теории графов;
- умение составлять формализованное описание и математическую постановку основных задач на графах;
- знание основных тождеств, теорем и законов теории графов;
- практические навыки использования основных графовых алгоритмов для решения различных оптимизационных задач.

Самостоятельная работа предусматривает проработку лекций (1,2 часа в неделю), тестирование, а также изучение литературы, формулировку цели работы, объекта и задач исследования, методов, источников и средств библиографического поиска, использованных для достижения поставленной цели.

Модуль включает в себя формулировку цели, проблемное изложение программного лекционного материала, тестовые вопросы для самоконтроля и список литературы. В процессе самостоятельного изучения представленных методических материалов происходит формирование указанных компетенций.

Глава 15

ВВЕДЕНИЕ В ТЕОРИЮ ГРАФОВ

> Математика — это не просто созданное человеком мощное оружие познания, а средство, которое позволяет нам осуществлять надежный контакт с высшей объективной реальностью, в огромной степени расширяя пределы информационных каналов, непосредственно связанных с нашими органами чувств.
>
> *М. Клайн*

Граф, *вершина*, *ребро*, *дуга*, *петля*, *мультиграф*, *матрица смежности*, *матрица инцидентности*, *полный граф*, *конечный граф*, *нуль-граф*, *двойственный граф*, *локальная степень*, *подграф*, *суграф*, *регулярный граф*, *операции над графами*, *маршруты*, *цепи*, *циклы*, *связность*, *компонента связности*, *разбиение графа*, *разрез*, *блок*, *точка сочленения*, *мост*, *неразделимый граф*, *алгоритм определения кратчайших маршрутов в графе*

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать способы задания графов;
- уметь выполнять операции над графами;
- уметь строить матрицы инцидентности, смежности и списки, а также переходить от одной формы представления графа к другой;
- уметь строить платоновы графы;
- уметь определять для любого вида графа его двойственный;
- знать определение нечеткого графа и различать нечеткие неориентированные графы первого и второго рода;
- уметь строить подграфы и суграфы на смешанном графе, мультиграфе;

• уметь определять маршруты, цепи, циклы в произвольном графе;

• уметь определять компоненты связности графа;

• уметь выполнять разбиение графа;

• уметь определять точки сочленения и мосты в графе;

• уметь определять кратчайшие пути в графе на основе алгоритмов Дейкстры и Форда.

15.1. Способы задания графов и виды графов

15.1.1. Способы задания графов. При изучении взаимосвязи между объектами (взаимоотношений людей в коллективах, связей в ЭВА, в молекулах, микросхемах и т. п.) исследуемый объект обозначается точками и соединяющими их линиями. Такое изображение ввиду наглядности позволяет определять наиболее существенные связи, находить оптимальное решение задачи, определять ошибки в рассуждениях и т. д. Описание, состоящее из точек и соединяющих их линий, оказывается удобной моделью любого объекта, в связи с чем представляет интерес исследование самого этого объекта.

Теория графов оперирует формальными моделями объектов, имеет дело со свойствами самих графов независимо от того, какова природа объектов, описываемых графами. Использование аппарата теории графов для разработки алгоритмов в науке и технике приводит к повышению эффективности и качества создаваемых объектов.

Понятие *графа* опирается на понятие множества. *Графом* можно назвать объект, состоящий из двух множеств: точек и линий, которые находятся между собой в некотором отношении. Множество точек графа обозначается $X = \{x_1, x_2, \ldots, x_n\}$, $|X| = n$ и называется *множеством вершин*. Множество линий, соединяющих любые пары вершин, $(x_i, x_j) \in X$, называется *множеством ребер или дуг* и обозначается $U = \{u_1, u_2, \ldots, u_m\}$, $|U| = m$.

Граф, у которого все вершины «помечены» целыми числами от 1 до n, называется помеченным. Числа $1, 2, \ldots, n$ называют метками графа и обозначают вершины графа G через $1, 2, \ldots, i, \ldots, n$, или $x_1, x_2, \ldots, x_i, \ldots, x_n$, или $x(1), x(2), \ldots, x(i), \ldots, x(n)$. Причем все вышеуказанные записи являются равносильными. Заметим, что для реальных задач необходима нумерация элементов. Поэтому в дальнейшем будем рассматривать помеченные графы. Аналогичным образом помечаются ребра.

Тогда *графом* можно считать математический объект, который обозначается $\mathsf{G} = (X, U)$ и состоит из множества вершин и множества ребер или дуг, находящихся между собой в некотором отношении.

В общем случае множество линий, соединяющих любые пары вершин U, можно представить в виде $U = \widetilde{U} \cup \overrightarrow{U} \cup \overset{\circ}{U}$, где $\widetilde{U}$ — подмножество неориентированных линий, для которых несуществен порядок соединения вершин. Подмножество называется *подмножеством ребер*. Причем каждое ребро $u_i \in \widetilde{U}$ определяется неупорядоченной парой вершин x_k, y_l, которые оно соединяет. Записывается $u_i = (x_k, y_l)$ или $u_i = (y_l, x_k)$.

$\overrightarrow{U}$ — подмножество ориентированных линий, для которых существен порядок соединения вершин. Подмножество называется *подмножеством дуг*. Причем каждая дуга $u_i \in \overrightarrow{U}$ определяется упорядоченной парой (кортежем длины два) вершин x_k, y_l, которые u_i соединяет. Записывается $u_i = \langle x_k, y_l \rangle$. Заметим, что $u_i = \langle x_k, y_l \rangle$ и $u_j = \langle y_l, x_k \rangle$ — это различные дуги в графе G.

$\overset{\circ}{U}$ — подмножество линий, каждая из которых выходит и входит в одну и ту же соответствующую этой линии вершину. Называется $\overset{\circ}{U}$ *подмножеством петель*. Каждая петля $u_i \in \overset{\circ}{U}$ может определяться упорядоченной или неупорядоченной парой, например вида $u_i = \langle x_k, x_k \rangle$, $u_i = (x_k, x_k)$.

Граф $\mathsf{G} = (X, U)$, у которого $\overrightarrow{U}$, $\widetilde{U}$, $\overset{\circ}{U} \neq \varnothing$, называется смешанным. На рис. 15.1 показан смешанный граф G. Здесь $|X| = 6$, $|U| = 13$; $U = \widetilde{U} \cup \overrightarrow{U} \cup \overset{\circ}{U}$; $\widetilde{U} = \{u_3, u_4, u_7, u_9, u_{10}, u_{11}\}$; $\overrightarrow{U} = \{u_1, u_2, u_8, u_{12}, u_{13}\}$; $\overset{\circ}{U} = \{u_5, u_6\}$.

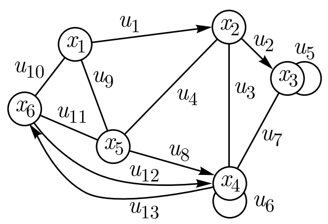


Рис. 15.1. Смешанный граф G

Граф $\mathsf{G} = (X, U)$, у которого $U = \overrightarrow{U}$, а $\widetilde{U} = \varnothing$ называется ориентированным, или орграфом. Орграф будем обозначать $\mathsf{D} = (X, U)$ или $\mathsf{G} = \{X, \overrightarrow{U}\}$.

Заметим, что подмножество петель можно рассматривать как ориентированные, так и неориентированные ребра. Обычно всегда оговаривают, когда рассматривается граф с петлями. Граф G, у которого $U = \widetilde{U}$, называется неориентированным графом, или неорграфом. Графы, не содержащие петель и кратных ребер, называются *простыми*.

Рассмотрим сначала неорграфы с петлями и без петель, которые будем называть графами. Граф G, у которого существует хотя бы одна пара вершин, соединяемых m ребрами $u_i \in U$, называется *мультиграфом*, а максимальное m — *мультичислом* графа G $(m \in \{2, 3, \ldots\})$. Ребра, соединяющие одну и ту же пару вершин, называются кратными. На рис. 15.2 показан пример мультиграфа $(m = 4)$.

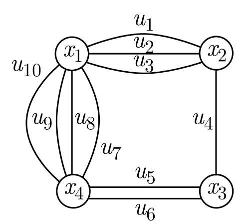


Рис. 15.2. Мультиграф G

Если ребро $u_k \in U$ графа $\mathsf{G} = (X, U)$ соединяет вершины x_i, $x_j \in X$, т. е. $u_k = (x_i, x_j)$, то говорят, что ребро u_k — *инцидентно* вершинам x_i, x_j. Верно и обратное, вершины x_i, x_j инцидентны ребру u_k. Любые две вершины x_i, $x_j \in X$ графа G называются смежными, если существует соединяющее эти вершины ребро $u_k \in U$, т. е. $u_k = (x_i, x_j)$. Если два ребра u_k, $u_i \in U$ инцидентны одной и той же вершине, то они также называются *смежными*.

Основными способами задания графа $\mathsf{G} = (X, U)$ являются геометрический, аналитический и матричный. Основа геометрического способа задания графа — рисунок, дающий изображение графа. Изображение графа в виде рисунка обладает наглядностью, раскрывает содержательный смысл представляемого объекта, но не формально.

Рассмотрим аналитический способ задания графа. Согласно Н. Бурбаки граф записывается как $\mathsf{G} \subset B = X \times X$. К. Берж предлагает задавать граф как соответствие Γ между подмножествами множества вершин. При этом граф можно задать в виде $\mathsf{G} = (X, \Gamma)$, где $\Gamma \subseteq X \times X$, $\Gamma = \{\Gamma x_1, \Gamma x_2, \ldots, \Gamma x_n\}$. Здесь Γ — это отображение множества X в множество X. Например, пусть задан граф $\mathsf{G} = (X, \Gamma)$ (рис. 15.3). Тогда $|X| = 6$,

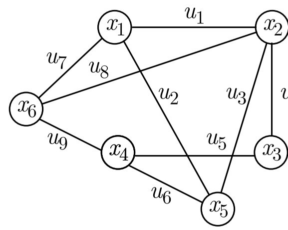


Рис. 15.3. Граф G

$X = \{x_1, x_2, \dots, x_6\}$, $\Gamma = \{\Gamma x_1, \Gamma x_2, \dots, \Gamma x_6\}$, $\Gamma x_1 = \{x_2, x_5, x_6\}$, $\Gamma x_2 = \{x_1, x_3, x_5, x_6\}$, $\Gamma x_3 = \{x_2, x_4\}$, $\Gamma x_4 = \{x_5, x_6\}$, $\Gamma x_5 = \{x_1, x_2, x_4\}$, $\Gamma x_6 = \{x_1, x_2, x_4\}$.

Учитывая вышесказанное, граф часто рассматривают как пару $\mathsf{G} = (X, \Gamma)$, образованную множеством X и отображением Γ множества X в себя.

Отметим, что для однозначного задания графов достаточно использовать $\Gamma x_1, \Gamma x_2, \dots, \Gamma x_{n-1}$ отображений. В этом случае элементы x_i, определяющие Γx_i, не включают в отображение Γx_{i+1}; в отображение Γx_{i+2} не включают элементы x_i, x_{i+1} и т. д. Соответственно в последнее отображение Γx_{n-1} не включают элементы $x_1, x_2, \dots, x_{n-2}$. Тогда рассматриваемый граф G (рис. 15.3), может быть задан следующим образом: $\Gamma x_1 = \{x_2, x_5, x_6\}$, $\Gamma x_2 = \{x_3, x_5, x_6\}$, $\Gamma x_3 = \{x_4\}$, $\Gamma x_4 = \{x_5, x_6\}, \Gamma x_5 = \varnothing$.

Такое представление графа позволяет экономить память ЭВМ и исключать избыточную информацию при представлении графа аналитическим способом.

Большинство задач информатики удобно решать при использовании матричного задания графов. Квадратная таблица $\mathrm{R} = ||r_{i,j}||_{nxn}$ называется *матрицей смежности*, если ее элементы образуются по правилу:

$$r_{i,j} = \begin{cases} 1, & \text{если вершина } x_i \text{ смежна с вершиной } x_j; \\ 0 & \text{в противном случае.} \end{cases}$$

Заметим, что для мультиграфа

$$r_{i,j} = \begin{cases} q, & \text{если вершина } x_i \text{ соединена } q\text{-ребрами с вершиной } x_j; \\ 0 & \text{в противном случае.} \end{cases}$$

Очевидно, что для рассматриваемых неорграфов $r_{i,j} = r_{j,i}$ и для задания графа можно использовать треугольную матрицу R. Для графа без петель $r_{i,i} = 0$, для графа с петлями $r_{i,i} \neq 0$. Для графа **G** (см. рис. 15.3) треугольная матрица смежности имеет вид

$$\mathrm{R} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \left| \begin{matrix} 0 & 1 & 0 & 0 & 1 & 1 \\ & 0 & 1 & 0 & 1 & 1 \\ & & 0 & 1 & 0 & 0 \\ & & & 0 & 1 & 1 \\ & & & & 0 & 0 \\ & & & & & 0 \end{matrix} \right| \end{array}.$$

Строки и столбцы матрицы R соответствуют вершинам графа **G**. На пересечении x_i строки и x_j столбца ставится элемент $r_{i,j}$, соответствующий числу ребер, соединяющих вершины x_i и x_j. Заметим, что строки и столбцы матрицы R также можно нумеровать числами натурального ряда, соответствующими индексам помеченных вершин графа. Петли в графе изображаются элементами $r_{i,j}$, расположенными по главной диагонали матрицы R.

Преимущество использования матриц смежности — это простота и формальность преобразований над графами.

Основной недостаток применения матрицы смежности заключается в том, что она занимает в ЭВМ память объемом $|X|^2$ даже тогда, когда граф содержит только $|X|$ ребер. Устранением этого недостатка является представление графа в виде списков. Список смежности для вершины x_i — совокупность всех вершин, смежных с x_i. Представление графа в виде списков смежности требует памяти ЭВМ порядка $|X| + |U|$. В основном таким представлением графа пользуются, когда $|U| \ll |X|^2$. Для графа **G** (см. рис. 15.3) список смежности имеет следующий вид:

$$\mathrm{R}_L = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \left| \begin{matrix} x_2 & x_5 & x_6 & 0 \\ x_1 & x_3 & x_5 & x_6 \\ x_2 & x_4 & 0 & 0 \\ x_3 & x_5 & x_6 & 0 \\ x_1 & x_2 & x_4 & 0 \\ x_1 & x_2 & x_4 & 0 \end{matrix} \right|,$$

а для треугольной матрицы

$$\mathrm{R}_L = \begin{array}{c|ccc|} x_1 & x_2 & x_5 & x_6 \\ x_2 & x_3 & x_5 & x_6 \\ x_3 & x_4 & 0 & 0 \\ x_4 & x_5 & x_6 & 0 \\ x_5 & 0 & 0 & 0 \\ x_6 & 0 & 0 & 0 \end{array} \quad \text{или} \quad \mathrm{R}_L = \begin{array}{c|ccc|} 1 & 2 & 5 & 6 \\ 2 & 3 & 5 & 6 \\ 3 & 4 & 0 & 0 \\ 4 & 5 & 6 & 0 \\ 5 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \end{array}.$$

Заметим, что рассмотренные списки составлены по наличию ребер, соединяющих вершины. Списки могут составляться также по отсутствию связей (по умолчанию).

Прямоугольная таблица вида $\mathrm{I} = ||i_{k,l}||_{n\times m}$ называется *матрицей инцидентности*, если ее элементы образуются по правилу:

$$i_{k,l} = \begin{cases} 1, & \text{если вершина } x_k \text{ инцидентна ребру } u_l; \\ 0 & \text{в противном случае.} \end{cases}$$

Матрица инцидентности для **G** (см. рис. 15.3) имеет вид

$$\mathrm{I} = \begin{array}{c|ccccccccc|} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ \hline x_1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ x_2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ x_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ x_5 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}.$$

Строки матрицы I соответствуют вершинам графа, столбцы — ребрам, а элемент $i_{k,l}$ указывает на инцидентность вершины x_k и ребра u_l. В каждом столбце матрицы I расположено по две единицы, так как каждое ребро соединяет ровно две вершины. При наличии в графе петель соответствующие им столбцы в матрице I будут иметь по одной единице, так как петля соединяет только одну вершину графа.

А. Кофман предлагает задавать булеву матрицу как шахматную доску. Заштрихованным клеткам в булевой матрице соответствуют нули, а незаштрихованным — единицы или наоборот. Такую булеву матрицу называют сотами, так как в различных задачах на графах необходимо размещать объекты по ячейкам, объединенным в соты.

Известно, что число ячеек в матрице (сотах) размером $m \times n$ равно 2^{mn}. Соты порядка n содержат n^2 ячеек. Тогда число ячеек матрицы с k запретными элементами (соответствующие элементы булевой матрицы — нули) равно

$$\sum_{k=0}^{n^2} c_{n^2}^k = 2^{n^2}.$$

Предполагается, что выполняются следующие условия:

• соты (матрица) имеют порядок n;

• в каждой ячейке матрицы располагается не более одного объекта (т. е. ноль или единица).

В этом случае ячейки матрицы разбиваются на два подмножества:

• ячейки, обладающие неким свойством;

• ячейки, не обладающие этим свойством.

Тогда, согласно А. Кофману, говорят, что граф (рис. 15.4)

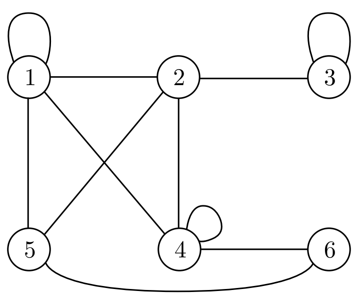


Рис. 15.4. Граф G

может быть реализован в виде матрицы:

$$\begin{array}{c|cccccc|}
1 & \blacksquare & \blacksquare & \square & \blacksquare & \blacksquare & \square \\
2 & \blacksquare & \square & \blacksquare & \blacksquare & \blacksquare & \square \\
3 & \square & \blacksquare & \blacksquare & \square & \square & \square \\
4 & \blacksquare & \blacksquare & \square & \blacksquare & \square & \blacksquare \\
5 & \blacksquare & \blacksquare & \square & \square & \square & \blacksquare \\
6 & \square & \square & \square & \blacksquare & \blacksquare & \square
\end{array},$$

где закрашенным элементам матрицы соответствуют ребра графа G.

Такая запись представляет интерес для нечеткого задания графов. При этом каждому ребру можно поставить в соответствие функцию принадлежности.

Кёниг и К. Берж под графом понимают разбиение теоретико-множественного произведения некоторого счетного множества на себя, на две части.

Графы иногда представляют в латинской матрице. Для графа G (рис. 15.4) такое представление имеет вид

$$\begin{array}{c|cccccc|}
1 & 1{,}1 & 1{,}2 & \varnothing & 1{,}4 & 1{,}5 & \varnothing \\
2 & 2{,}1 & \varnothing & 2{,}3 & 2{,}4 & 2{,}5 & \varnothing \\
3 & \varnothing & 3{,}2 & 3{,}3 & \varnothing & \varnothing & \varnothing \\
4 & 4{,}1 & 4{,}2 & \varnothing & 4{,}4 & \varnothing & 4{,}6 \\
5 & 5{,}1 & 5{,}2 & \varnothing & \varnothing & \varnothing & 5{,}6 \\
6 & \varnothing & \varnothing & \varnothing & 6{,}4 & 6{,}5 & \varnothing
\end{array}.$$

Латинская матрица удобна для задания как ориентированных, так и неориентированных графов (неорграфов). Причем в неорграфах значение элемента матрицы $i,\ j = j,\ i$ (в ориентированных графах наоборот: $i,\ j \neq j,\ i$).

15.1.2. Виды графов. Определим *смежностный* (*двойственный*) *граф* $\mathsf{G}_S = (U, V)$, (обозначают также $U(\mathsf{G})$) для графа $\mathsf{G} = (X, U)$. Вершинами G_S являются ребра G, а ребрами G_S — пары $(u_i,\ u_j)$. Для построения G_S по G на каждом ребре $u_i \in U$ графа G выбирают среднюю точку и считают ее вершиной $u_i \in U$ графа G_S (см. рис. 15.5). Таким образом, двойственный граф G_S будет иметь 12 вершин: $U = \{u_1, u_2, \ldots, u_{12}\}$.

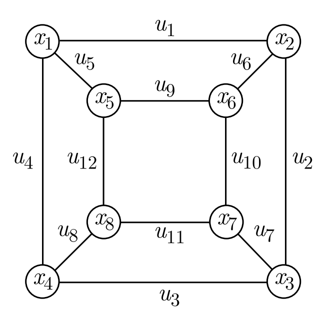


Рис. 15.5. Построение двойственного графа G_S

Затем пару $(u_i,\ u_j)$ соединяют ребром $v_k = (u_i, u_j)$, если ребра $u_i,\ u_j$ имеют общую вершину в G. То есть для графа G_S вершина u_1, например, будет иметь общие ребра с вершинами u_2, u_4, u_5, u_6, так как эти ребра в исходном графе G имели общие вершины (x_1 и x_2). В результате получим следующий двойственный граф G_S (рис. 15.6)

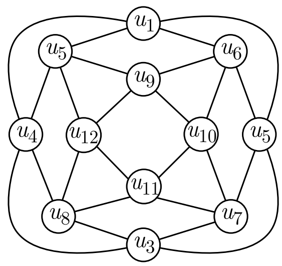


Рис. 15.6. Двойственный граф G_S

Следует отметить, что существуют различного рода матрицы и списки, производные от R и I, которые используются при записи алгоритмов.

Граф G называется *конечным*, если множества его вершин и ребер — конечные множества. Далее будем рассматривать конечные графы, так как модели объектов, представляемые графами, состоят из конечного числа элементов.

Граф G, у которого $X \neq \varnothing$, а $U = \varnothing$, называется *нуль-графом*, а его вершины — изолированными. Обозначается он G_0. Граф $\mathsf{G} = (X, U)$, $|X| = n$ называется *полным*, если между любой парой вершин x_i, $x_j \in X$ имеется ребро $u_k \in U$. Обозначается он K_n. Число ребер, инцидентных вершине $x_i \in X$ графа G, называется *локальной степенью* или просто степенью этой вершины и обозначается $\rho(x_i)$ или ρ_i. Тогда число ребер графа $\mathsf{G} = (X, U)$, $|X| = n$, $|U| = m$,

$$m = \frac{1}{2}\sum_{i=1}^{n} \rho(x_i). \tag{15.1}$$

Заметим, что если в графе имеются петли, то каждая из них должна считаться дважды. Так как для полного графа на n вершин $\rho(x_1) = \rho(x_2) = \ldots = \rho(x_n)$, то

$$m = n(n-1)/2. \tag{15.2}$$

Графы с одинаковыми числами степеней называются *регулярными*. Регулярные графы степени 3 называются кубическими (трехвалентными). Примером кубического графа является граф Петерсона (рис. 15.7).

Среди регулярных графов интерес представляют так называемые платоновы графы — графы, образованные вершинами и ребрами пяти правильных многогранников — платоновых тел:

- тетраэдра (рис. 15.8, *а*),
- куба (рис. 15.8, *б*),

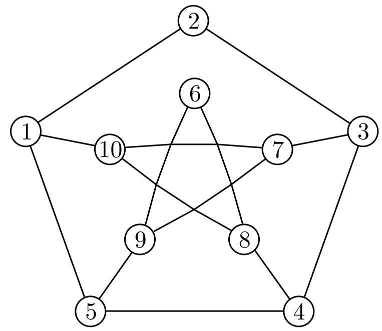


Рис. 15.7. Граф Петерсона

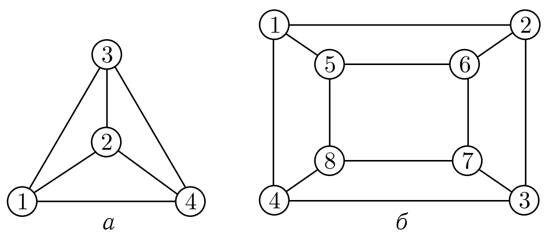


Рис. 15.8. Тетраэдр (*а*), куб (*б*)

- октаэдра (рис. 15.9, *а*),
- додекаэдра (рис. 15.9, *б*),
- икосаэдра (рис. 15.9, *в*).

Число ребер регулярного графа с локальной степенью $\rho(x_i)$ вершины x_i определяется как

$$m = n\rho(x_i)/2. \tag{15.3}$$

Заметим, что в графе каждое ребро соединяет две вершины и, следовательно, число вершин нечетной степени четно.

Полный граф $\mathsf{K}_{1,n}$ называется звездным. Он показан на рис. 15.10.

Над графами можно выполнять следующие операции: объединение ($\cup$), сложение ($+$) и произведение ($\times$).

1. Объединением графов $\mathsf{G}(a) = (X(a), U(a))$ и $\mathsf{G}(b) = (X(b), U(b))$ называется граф $\mathsf{G} = (X, U)$, у которого $X = (X(a) \cup X(b))$, $U = (U(a) \cup U(b))$ (рис. 15.11).

2. Суммой графов $\mathsf{G}(a) = (X(a), U(a))$ и $\mathsf{G}(b) = (X(b), U(b))$ называется граф $\mathsf{G} = (X, U)$, который определяется объединением $\mathsf{G}(a)$ и $\mathsf{G}(b)$, и каждая вершина $x_i \in X(a)$, не вошедшая

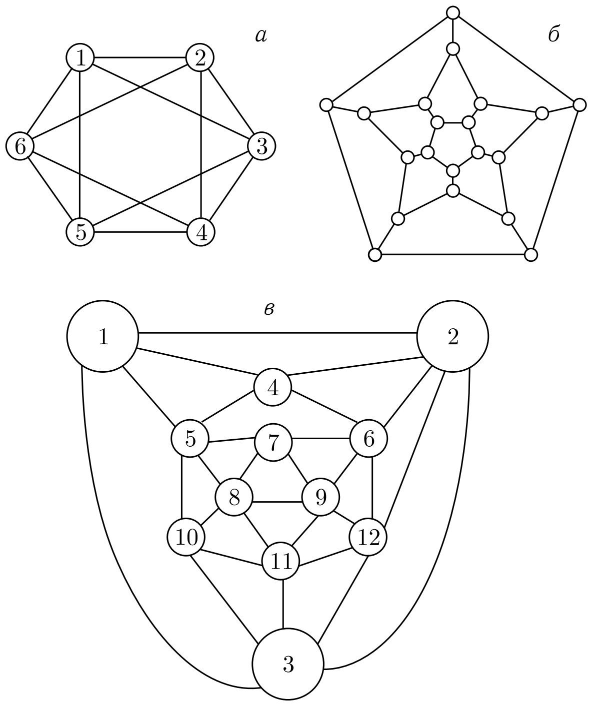


Рис. 15.9. Октаэдр (*а*), додекаэдр (*б*), икосаэдр (*в*)

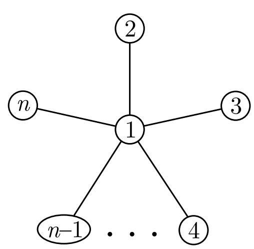


Рис. 15.10. Пример звездного графа

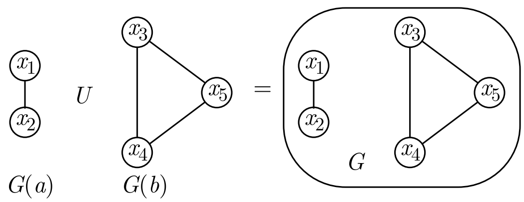


Рис. 15.11. Операция объединения графов

в пересечение $X(a) \cap X(b)$, соединяется со всеми вершинами $X(b) \backslash X(a) \cap X(b)$ и наоборот (рис. 15.12).

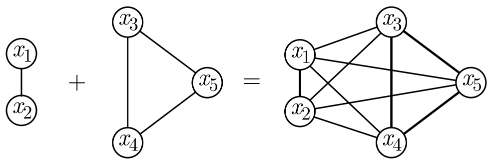


Рис. 15.12. Сумма графов

3. Произведением графов $\mathsf{G}(a) = (X(a), U(a))$ и $\mathsf{G}(b) = (X(b), U(b))$ называется граф $\mathsf{G} = (X, U)$, у которого $X = X(a) \times X(b)$, а ребра соответствуют связности подграфов (рис. 15.13).

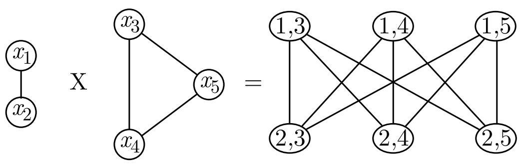


Рис. 15.13. Произведение графов

Подграфом графа G называется граф, у которого все вершины и ребра принадлежат G, т.е. $\mathsf{G}' = (X', U')$, подграф $\mathsf{G} = (X, U)$, если $X' \subseteq X$, $U' \subseteq U$ и ребра U' соединяют только вершины X'. Удаление из графа $\mathsf{G} = (X, U)$ вершины x_i со всеми инцидентными ей ребрами приводит к подграфу $\mathsf{G}'' = (X'', U'')$, $X'' = X \backslash x_i$. На рис. 15.14, *а–в* показаны граф $\mathsf{G} = (X, U)$ и два его подграфа, $\mathsf{G}1 = (X1, U1)$ и $\mathsf{G}2 = (X2, U2)$, где $|X| = 7$, $|U| = 10$, $X = \{x_1, x_2, \ldots, x_7\}$, $U = \{u_1, u_2, \ldots, u_{10}\}$; $|X1| = 3$, $|U1| = 3$, $X1 = \{x_1, x_2, x_4\}$, $U1 = \{u_1, u_7, u_6\}$; $|X2| = 5$, $|U2| = 6$, $X2 = \{x_2, x_3, x_5, x_6, x_7\}$, $U2 = \{u_2, u_3, u_4, u_8, u_9, u_{10}\}$. Заметим, что подграф $\mathsf{G}1$ можно получить, удалив из G вершины x_3, x_5, x_6, x_7 с инцидентными им ребрами, а подграф $\mathsf{G}2$, удалив из G вершины x_1, x_4 с инцидентными им ребрами.

Суграфом $\mathsf{G}' = (X', U')$ графа $\mathsf{G} = (X, U)$ называется граф, для которого $X' = X$, $U' \subseteq U$.

Граф называется *дополнением* графа G до полного, если он состоит из всех ребер полного графа K, не принадлежащих G, $\overline{\mathsf{G}} = \mathsf{K} \backslash \mathsf{G}$. На рис. 15.14, *г* показан суграф графа (см. рис. 15.14, *в*). На рис. 15.14, *д* приведено дополнение графа G (см. рис. 15.14, *в*) до полного графа K_7.

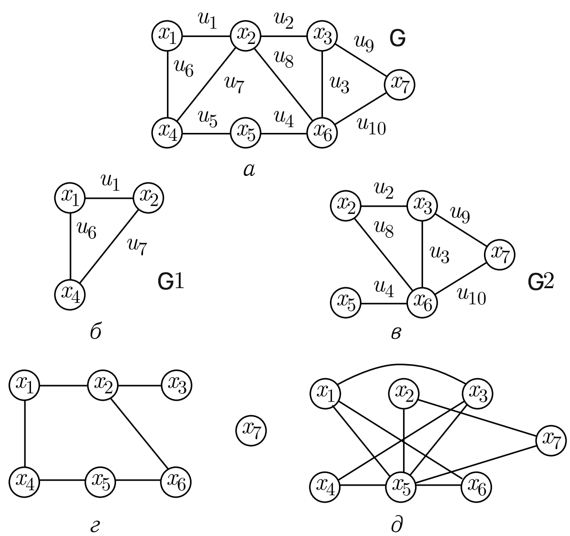


Рис. 15.14. Граф G (*а*) и его подграфы G1 (*б*) и G2 (*в*); суграф графа G (*г*); дополнение графа G (*д*)

15.1.3. Нечеткие неориентированные графы. В некоторых случаях для построения моделей сложных технических объектов, например электрических схем, используются нечеткие графы.

Граф $\mathsf{G} = (X, U)$ называется нечетким, если для каждой вершины $x_i \in X$ множество U является нечетким множеством. Множество U характеризуется функцией принадлежности μ_u, принимающей значения из отрезка $[0, 1]$. Значение $\mu(u)$ показывает степень принадлежности ребра u к множеству U. Очевидно, что если $\mu_u(x)$ для любых x, $y \in X$ принимает значение 0 или 1, то нечеткий граф G становится обыкновенным.

В данном разделе рассмотрим способы задания нечетких неориентированных графов двух видов. К первому виду относят графы, имеющие нечеткое множество ребер. Ко второму виду будем относить графы, имеющие, кроме того, нечеткое множество вершин.

Нечетким неориентированным графом первого рода называется и через $\widetilde{\mathsf{G}} = (X, \widetilde{U})$ обозначается пара множеств, у которого $X = \{x_i\}$, $i \in I = \{1, 2, \ldots, n\}$ — четкое множество вершин, а $\widetilde{U} = \{\langle \mu_U(x_i, x_k)/(x_i, x_k)\rangle\}$ — нечеткое множество ребер, где x_i, $x_k \in X$, $\mu_U(x_i, x_k)$ — значение функции принадлежности μ_U для ребра (x_i, x_k).

Как четкие, так и нечеткие неориентированные графы удобно задавать в виде нечетких матриц смежности $\mathrm{R}_x = ||r_{ik}||_n$, где $r_{ik} = \mu_U(x_i, x_k)$.

Нетрудно видеть, что для неориентированных графов первого рода нечеткая матрица смежности симметрична относительно главной диагонали.

Дадим определение нечеткого неориентированного графа второго вида. Пусть имеется некоторое универсальное множество A и задано нечеткое множество $\widetilde{X}$ в A, имеющее вид $\widetilde{X} = \{\langle \mu_x(x)/x \rangle\}$, $x \in A$.

Нечеткий граф $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U})$ является *нечетким неориентированным графом второго вида*, если $\widetilde{X} = \{\langle \mu_x(x)/x \rangle\}$, $x \in A$, $|\widetilde{X}| = n$, $\widetilde{U} = \{\langle \mu_U(x_i, x_k)/(x_i, x_k) \rangle\}$, $x_i, x_k \in X$, где X — носитель множества $\widetilde{X}$.

Примеры решения задач

Пример 15.1. Для графа, показанного на рис. 15.15, записать множества ориентированных и неориентированных ребер и дуг.

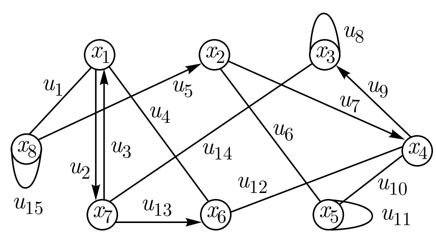


Рис. 15.15

Граф содержит 8 вершин и 15 ребер: $|X| = 8$; $|U| = 15$, тогда множество ребер $U = \widetilde{U} \cup \overrightarrow{U} \cap \overset{\circ}{U}$ включает в себя следующие подмножества: $\widetilde{U} = \{u_1, u_4, u_6, u_{10}, u_{12}, u_{14}\}$; $\overrightarrow{U} = \{u_2, u_3, u_5, u_7, u_9, u_{13}\}$; $\overset{\circ}{U} = \{u_8, u_{11}, u_{15}\}$.

Пример 15.2. Привести пример мультиграфа с мультичислом, равным 4.

Ответ. Мультиграф, удовлетворяющий заданным требованиям, приведен на рис. 15.16.

Пример 15.3. Пусть граф $\mathsf{G} = (X, U)$ (рис. 15.17) задан графическим способом. Задать тот же граф аналитическим и матричным способом.

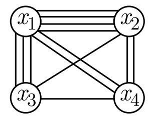


Рис. 15.16. Искомый мультиграф

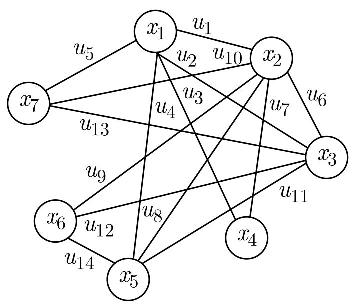


Рис. 15.17. Граф G

Решение. Граф, заданный на рис. 15.17, можно задать аналитическим способом следующим образом: $|X| = 7$; $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$; $\Gamma = \{\Gamma x_1, \Gamma x_2, \Gamma x_3, \Gamma x_4, \Gamma x_5, \Gamma x_6, \Gamma x_7\}$; $\Gamma x_1 = \{x_2, x_3, x_4, x_5, x_7\}$; $\Gamma x_2 = \{x_1, x_3, x_4, x_5, x_6, x_7\}$; $\Gamma x_3 = \{x_1, x_2, x_5, x_6, x_7\}$; $\Gamma x_4 = \{x_1, x_2\}$; $\Gamma x_5 = \{x_1, x_2, x_3, x_6\}$; $\Gamma x_6 = \{x_2, x_3, x_5\}$; $\Gamma x_7 = \{x_1, x_2, x_3\}$.

Очевидно, что такой способ задания содержит слишком много избыточной информации. Для устранения этого недостатка достаточно использовать Γx_{n-1}-образ и не включать в Γx_{i-1} элементы x_i, определяющие Γx_i.

Тогда получим: $\Gamma x_1 = \{x_2, x_3, x_4, x_5, x_7\}$; $\Gamma x_2 = \{x_3, x_4, x_5, x_6, x_7\}$; $\Gamma x_3 = \{x_5, x_6, x_7\}$; $\Gamma x_4 = \varnothing$; $\Gamma x_5 = \{x_6\}$; $\Gamma x_6 = \varnothing$; $\Gamma x_7 = \varnothing$.

Граф, изображенный на рис. 15.17, можно задать также в виде специальной матрицы (матрицы смежности):

$$\mathrm{R} = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{vmatrix}.$$

При этом для задания неориентированного графа вполне достаточно треугольной матрицы типа R':

$$\mathrm{R}' = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \left| \begin{array}{ccccccc} 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ & 0 & 1 & 1 & 1 & 1 & 1 \\ & & 0 & 0 & 1 & 1 & 1 \\ & & & 0 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 \\ & & & & & 0 & 0 \\ & & & & & & 0 \end{array} \right|.$$

Для любого графа, в том числе заданного на рис. 15.17, можно задать еще одну специальную матрицу — матрицу инцидентности. Любой элемент i_{pq} матрицы инцидентности будет равен 1, в случае если p-я вершина в исходном графе инцидентна q-му ребру. Таким образом, матрица инцидентности для графа, заданного на рис. 15.17, примет следующий вид:

$$\mathrm{I} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \left| \begin{array}{cccccccccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} & u_{14} \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} \right|.$$

Пример 15.4. Привести примеры подграфа, суграфа и дополнения до полного для графа, заданного на рис. 15.17.

На рис. 15.18 изображен подграф G' графа G, полученный удалением из графа G вершин x_2, x_3 и инцидентных им ребер.

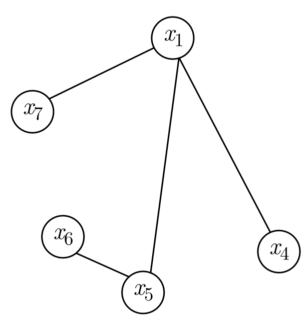


Рис. 15.18. Подграф графа G

На рис. 15.19 изображен граф $\mathsf{G}' = (X', U')$, являющийся суграфом графа G: $|X'| = 7$, $|U'| = 8$.

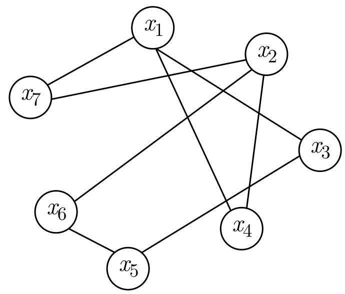


Рис. 15.19. Суграф графа G

На рис. 15.20 изображен граф $\overline{\mathsf{G}} = \mathsf{K} \backslash \mathsf{G}$, являющийся дополнением до полного графа G.

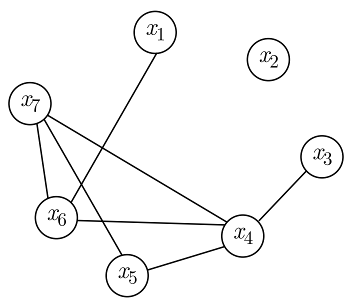


Рис. 15.20. Дополнение графа G

Пример 15.5. Для графа G, заданного на рис. 15.21, a, построить двойственный ему граф G_S.

Решение. Зададим смежностный (двойственный) граф $\mathsf{G}_S = (U, V)$ для графа $\mathsf{G} = (X, U)$. Для построения G_S по G на каждом ребре $u_i \in U$ графа G выбирают среднюю точку и считают ее вершиной $u_i \in U$ графа G_S. Граф G содержит 8 ребер, следовательно, двойственный ему граф G_S будет иметь 8 вершин.

Затем пару (u_i, u_j) соединяем ребром $v_k = (u_i, u_j)$, если ребра u_i, u_j имеют общую вершину в G. То есть для графа G_S вершина u_1, например, будет иметь общие ребра с вершинами u_2, u_3, u_4, u_7, u_8, так как эти ребра в исходном графе G имели общие вершины (x_1 и x_3). Далее продолжаем строить ребра для других вершин графа G_S.

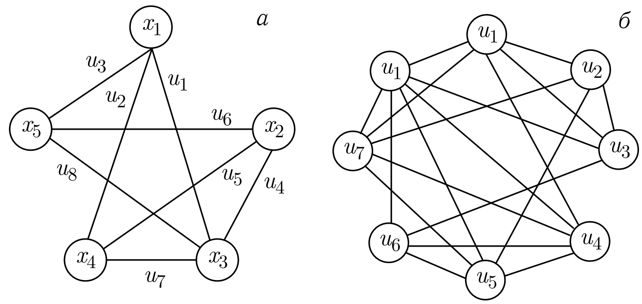


Рис. 15.21. Пример графа G (*а*); двойственный граф G_S (*б*)

Ответ. В результате получим следующий двойственный граф G_S (рис. 15.21, *б*).

Пример 15.6. Задать нечеткий граф $\widetilde{\mathsf{G}} = (X, \widetilde{U})$, у которого $X = \{x_1, x_2, x_3, x_4, x_5\}$, а $\widetilde{U} = \{\langle 0{,}3/(x_1, x_1)\rangle, \langle 0{,}7/(x_1, x_2)\rangle, \langle 1/(x_1,x_5)\rangle, \langle 0{,}6/(x_2,x_4)\rangle, \langle 0{,}4/(x_2,x_5)\rangle, \langle 0{,}9/(x_3,x_4)\rangle, \langle 0{,}8/(x_3,x_3)\rangle\}$.

Решение. Граф показан на рис. 15.22.

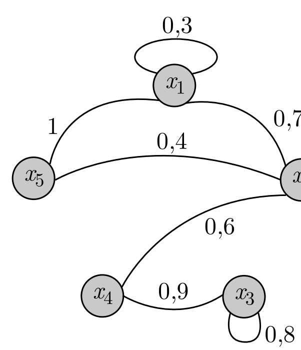


Рис. 15.22. Нечеткий граф

Пример 15.7. Записать матрицу смежности нечеткого неориентированного графа $\widetilde{\mathsf{G}} = (X, \widetilde{U})$, заданного в примере 15.1.

Решение. Матрица смежности R_x^1 нечеткого неориентированного графа $\widetilde{\mathsf{G}}$, рассмотренного в примере 15.1, запишется следующим образом:

$$\mathrm{R}_x^1 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left| \begin{array}{ccccc} 0{,}3 & 0{,}7 & 0 & 0 & 1 \\ 0{,}7 & 0 & 0 & 0{,}6 & 0{,}4 \\ 0 & 0 & 0{,}8 & 0{,}9 & 0 \\ 0 & 0{,}6 & 0{,}9 & 0 & 0 \\ 1 & 0{,}4 & 0 & 0 & 0 \end{array} \right| \end{array}.$$

Пример 15.8. Записать нечеткий неориентированный граф второго вида.

Решение. Зададим нечеткий неорграф второго вида $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U})$, у которого $\widetilde{X} = \{\langle 1/x_1\rangle, \langle 0{,}4/x_2\rangle, \langle 0{,}7/x_3\rangle, \langle 0{,}5/x_4\rangle, \langle 0{,}9/x_5\rangle\}$, $\widetilde{U} = \{\langle 0{,}3/(x_1, x_1)\rangle, \langle 0{,}7/(x_1, x_2)\rangle, \langle 1/(x_1, x_5)\rangle, \langle 0{,}6/(x_2, x_4)\rangle \langle 0{,}4/(x_2, x_5)\rangle, \langle 0{,}9/(x_3, x_4)\rangle, \langle 0{,}8/(x_3, x_3)\rangle\}$.

Вопросы для самоконтроля

1. Что такое граф? Приведите пример. Назовите известные вам типы графов.
2. В чем состоит различие между ориентированными и неориентированными графами?
3. Опишите известные вам способы задания графов.
4. Что называется подграфом и суграфом графа?
5. Что называется дополнением графа?
6. Какой граф называется двойственным?
7. Какие операции могут выполняться над графами?
8. Что такое мультиграф и как определяется его мультичисло?
9. Дайте определение регулярного графа.
10. Дайте определение нечеткого графа. В чем состоит отличие нечеткого неориентированного графа первого и второго рода?

Задания для самостоятельной работы

1. Приведите примеры задания графа различными способами: аналитическим, графическим, матричным.
2. Задайте графическим и матричным способами ориентированный, неориентированный и смешанный граф.
3. Задайте произвольный граф $\mathsf{G} = (X, U)$, $|X| = 9$, $|U| = 24$ и запишите для него матрицу смежности и инцидентности, а также его списковое представление.
4. Приведите словесный алгоритм описания и составьте структурную схему алгоритма перехода от матрицы инцидентности к матрице смежности и наоборот.

5. Определите количество ребер в полном графе K_{15} без петель.

6. Постройте Платоновы графы.

7. Задайте Платоновы графы на основе матриц смежности и инцидентности.

8. Задан произвольный граф $\mathsf{G} = (X, U)$, где $|X| = 6$, $|U| = 10$. Постройте двойственный ему граф G_S.

9. Приведите пример мультиграфа и определите его мультичисло.

10. Приведите примеры нечетких неориентированных графов первого и второго рода.

Понятие графа опирается на понятие множества. Графом можно назвать объект, состоящий из двух множеств: точек и линий, которые находятся между собой в некотором отношении.

Множество точек графа обозначается $X = \{x_1, x_2, \ldots, x_n\}$ и называется множеством вершин. Множество линий, соединяющих любые пары вершин $(x_i, x_j) \in X$, называется множеством ребер или дуг и обозначается $U = \{u_1, u_2, \ldots, u_m\}$.

15.2. Маршруты, цепи, циклы

Маршрутом в графе $\mathsf{G} = (X, U)$ называется некоторая конечная последовательность ребер вида $S = (x_0, x_1), (x_1, x_2), \ldots, (x_{l-1}, x_l)$, где x_0, x_l — соответственно его начальная и конечная вершины. Очевидно, что в конечном графе G можно выделить только конечное число маршрутов. Число ребер в маршруте S называется его *длиной*. Существует простой способ определения маршрутов длиной q по матрице R графа G путем возведения ее в q-ю степень.

Например, пусть задана матрица R графа G (рис. 15.23):

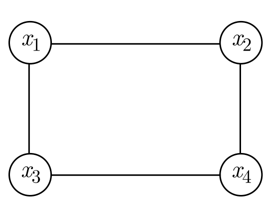


Рис. 15.23. Пример графа G

Возведем ее в квадрат по правилу:

$$r_{i,j}^2 = r_{1,i} \cdot r_{j,1} + r_{2,i} \cdot r_{j,2} + \ldots + r_{n,i} \cdot r_{j,n}. \qquad (15.4)$$

Тогда для рассматриваемого в данном примере графа, к примеру, элемент r_{11}^2 матрицы R^2 будет иметь следующее значение:

$$r_{1,1}^2 = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + a_{14}b_{41} = 0 + 1 \cdot 1 + 1 \cdot 1 + 0 = 2.$$

А например, $r_{3,3}^2 = r_{1,3}r_{3,1} + r_{2,3}r_{3,2} + r_{3,3}r_{3,3} + r_{4,3}r_{3,4} = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 2$.

$$\mathrm{R}^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left|\begin{array}{cccc} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{array}\right| \end{array}.$$

Каждый $r_{i,j}$ элемент R^2 равен числу маршрутов длиной 2, ведущих из вершины x_i в x_j. Например, $r_{3,2} = 2$ означает, что в графе два маршрута, ведущих из x_3 в x_2, длиной 2: $S_1 = (x_3, x_1), (x_1, x_2)$; $S_2 = (x_3, x_4), (x_4, x_2)$.

Возведение матрицы в степень производится путем поэлементного перемножения строки на столбец с суммированием полученных сомножителей.

Маршрут, в котором нет повторяющихся ребер, называется *цепью*. Замкнутая цепь, в которой $x_0 = x_l$, называется *циклом*. Соответственно цепи и циклы называются *простыми*, если они не содержат повторяющихся вершин, кроме, разумеется, первой и последней в случае цикла. В графе G (см. рис. 15.14, *а*):

$S_1 = (x_1, x_2), (x_2, x_3), (x_3, x_6), (x_6, x_2), (X_2, x_4)$ — маршрут,

$S_2 = (x_1, x_2), (x_2, x_3), (x_3, x_6), (x_6, x_2)$ — цепь,

$S_3 = (x_6,x_2), (x_2,x_4), (x_4,x_1), (x_1,x_2), (x_2,x_3), (x_3,x_6)$ — цикл,

$S_4 = (x_1, x_2), (x_2, x_3), (x_3, x_7), (x_7, x_6)$ — простая цепь,

$S_5 = (x_1,x_2), (x_2,x_3), (x_3,x_7), (x_7,x_6), (x_6,x_5), (x_5,x_4), (x_4,x_1)$ — простой цикл.

Цикл длиной 3 называется треугольником.

Связный регулярный граф степени 2 называется циклическим графом или циклом. Циклический граф с n вершинами обозначается C_n (рис. 15.24, *а*).

Соединение графов K_1 и C_{n-1} ($n \geqslant 3$) называется колесом с n вершинами и обозначается W_n. На рис. 15.24, *б* показано колесо W_6.

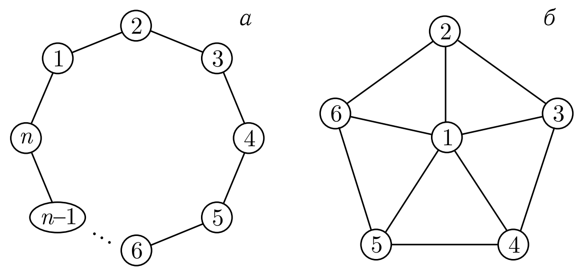


Рис. 15.24. Циклический граф (*а*); граф-колесо (*б*)

Две произвольные вершины $x_i, x_j \in X$ графа называются *связными*, если существует маршрут S, в котором концевыми будут вершины x_i, x_j. Граф G называется *связным*, если любые две его вершины связаны. В противном случае G не связен, а каждый из составляющих его подграфов G_1, $\mathsf{G}_2, \ldots, \mathsf{G}_l$ называется *компонентой связности*. Из определения связности следует, что в *связном* графе G вершина x_i связана сама с собой; если x_i связана с x_j, то x_j связана с x_i; если x_i связана с x_j, а x_j связана с x_k, то и x_i связана с x_k $(x_i, x_j, x_k \in X)$.

Следовательно, отношение связности является *отношением эквивалентности*. В этом случае множество вершин X графа $\mathsf{G} = (X, U)$, который моделирует любой объект, можно разбить на непересекающиеся классы G_i, причем ребра графа будут соединять только вершины внутри этих классов. Таким образом, получим *разбиение* графа G на связные подграфы (компоненты связности). Например, графы, изображенные на рис. 15.14, связные, а граф на рис. 15.25 состоит из четырех компонентов связности G_1–G_4, т. е. несвязен. Заметим, что связный граф состоит из единственной компоненты связности. Если граф имеет несколько компонент связности, то он несвязен, так как вершины из разных компонент связности нельзя соединить маршрутом.

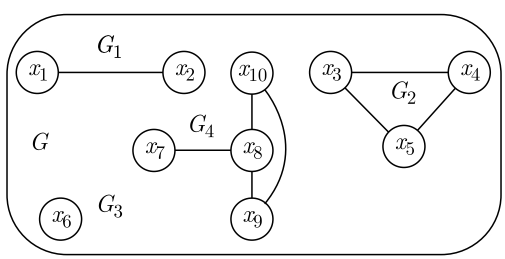


Рис. 15.25. Пример несвязного графа

Очевидно, что число ребер в связном графе

$$n-1 \leqslant m \leqslant n(n-1)/2. \tag{15.5}$$

Теорема 15.1. *Пусть* G — *простой граф с* n *вершинами и* k *компонентами. Тогда число* m *его ребер удовлетворяет неравенствам*

$$n-k \leqslant m \leqslant (n-k)(n-k+1)/2. \tag{15.6}$$

Следствие 15.1. *Любой простой граф с* n *вершинами и более чем* $(n-1)(n-2)/2$ *ребрами связен.*

При конструировании систем часто надо знать, какое наименьшее число связей необходимо удалить из схемы, чтобы она перестала быть связной. Если вершины связного графа $\mathsf{G}=(X,U)$ разбить на 2 подмножества X' и X'', где X'', $X' \subset X$, $X''=X\backslash X'$, $X' \cup X'' = X$, то подмножество ребер $U' \subset U$, у которых одни концевые вершины принадлежат X', а другие X'', называют *разрезом* графа G. Подграф $\mathsf{G}'=(X,U\backslash U')$, полученный из G удалением ребер разреза, будет несвязанным графом, состоящим, по крайней мере, из двух компонент связности. Множество ребер $U'' \subset U$, удаление которых дает несвязный подграф $\mathsf{G}'=(X,U\backslash U'')$, и не существует подмножества $U' \subset U''$ такого, что $\mathsf{G}''=(X,U\backslash U')$, несвязен, называют *правильным разрезом*. Из определения следует, что правильный разрез разбивает G точно на две компоненты связности. В этом случае разрез будет определяться как объединение правильных разрезов. На рис. 15.26, *а* показан пример разреза в графе $\mathsf{G}=(X,U)$. Разрез показан пунктирной линией.

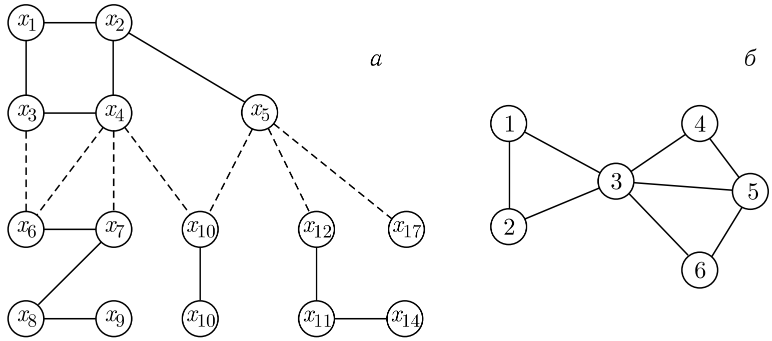


Рис. 15.26. Пример разреза в графе (*а*) и наличия точки сочленения (*б*)

Правильными разрезами будут следующие подмножества ребер: $U_1 = \{(x_3,x_6),(x_4,x_6),(x_4,x_7)\}$, $U_2=\{(x_4,x_10),(x_5,x_{10})\}$,

$U_3 = \{(x_5, x_{12})\}$, $U_4 = \{(x_5, x_{15})\}$, а разрез $U = U_1 \cup U_2 \cup U_3 \cup U_4$. Заметим, что разрез в графе всегда существует. Тривиальными разрезами G являются $U' = U$ и $U' = \varnothing$. Если разрез состоит из одного ребра $u_i \in U$, то u_i называется *перешейком*, *или мостом*.

Точкой сочленения называется вершина графа, после удаления которой граф распадается на связные компоненты. Например, в графе G на рис. 15.26, *б* вершина 3 является точкой сочленения.

Неразделимый граф — это связный, непустой, не имеющий точек сочленения граф. *Блок* графа — это его максимальный неразделимый подграф. Если G — неразделимый граф, то он часто сам называется блоком. Так, на рис. 15.27 приведен пример графа G, имеющего две точки сочленения x_5 и x_6, удаление которых разбивает граф G на 4 блока (рис. 15.28).

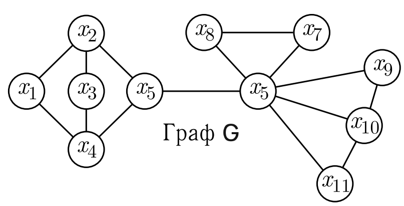


Рис. 15.27. Граф G

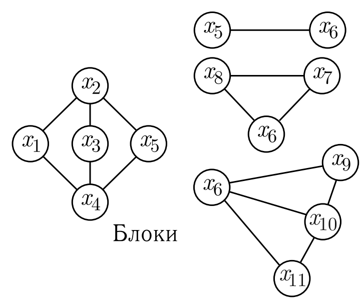


Рис. 15.28. Блоки графа G

Граф блоков $B(\mathsf{G})$ — это граф, в котором вершины-блоки и две вершины смежны, если соответствующие блоки имеют общую точку сочленения.

Вершинная связность $H = H(\mathsf{G})$ в G — это наименьшее число вершин, удаление которых приводит к несвязному графу. $H(\mathsf{G})$ в несвязном графе равна нулю. Связность в графе с точкой сочленения равна 1.

Реберная связность $\lambda = \lambda(\mathsf{G})$ — это наименьшее число ребер, удаление которых приводит к несвязному или тривиальному графу.

Для любого графа:

$$H(\mathsf{G}) \leqslant \lambda(\mathsf{G}) \leqslant \rho_{\min}(\mathsf{G}), \tag{15.7}$$

где $\rho_{\min}(\mathsf{G})$ — наименьшая степень графа.

Граф G называется n-связным, если $H(\mathsf{G}) \geqslant n$ и n-реберно-связным, если $\lambda(\mathsf{G}) \geqslant n$.

Теорема 15.2 (Дирак). *Если граф* G n*-связен*, $n \geqslant 2$, *то любое множество, содержащее* n *вершин графа* G, *принадлежит простому циклу.*

Теорема 15.3. *Граф 3 связен тогда и только тогда, когда он или совпадает с колесом, или получается из колеса с помощью двух типов операций:*

1) *добавление нового ребра;*

2) *замена вершины* V *с* $\rho(V)$, *по крайней мере, на две смежные вершины* V', V'', *где каждая вершина, которая была смежна с* V, *соединяется точно с одной из* V', V'' *так, чтобы в полученном графе* $\rho(V') \geqslant 3$ *и* $\rho(V'') \geqslant 3$.

Примеры решения задач

Пример 15.9. Пусть задан граф G, показанный на рис. 15.23. Подсчитать количество маршрутов длиной 3 в данном графе.

Решение. Запишем матрицу смежности графа G:

$$\mathrm{R} = \begin{array}{c|cccc|} & x_1 & x_2 & x_3 & x_4 \\ x_1 & 0 & 1 & 1 & 0 \\ x_2 & 1 & 0 & 0 & 1 \\ x_3 & 1 & 0 & 0 & 1 \\ x_4 & 0 & 1 & 1 & 0 \end{array}.$$

Как уже было показано ранее, чтобы найти число маршрутов длиной 2, в заданном графе необходимо возвести его матрицу смежности в квадрат. Алгебраическое произведение матриц определяется по формуле

$$r_{ij}^2 = \sum_{k=1}^{n} a_{ik}\, b_{ki},$$

где r_{ij}^2 — элемент матрицы R^2.

Вычислив, таким образом, значения всех элементов, мы придем в итоге к следующей матрице:

$$\mathrm{R}^2 = \begin{array}{c|cccc|} & x_1 & x_2 & x_3 & x_4 \\ x_1 & 2 & 0 & 0 & 2 \\ x_2 & 0 & 2 & 2 & 0 \\ x_3 & 0 & 2 & 2 & 0 \\ x_4 & 2 & 0 & 0 & 2 \end{array}.$$

Для того чтобы определить количество маршрутов длиной 3 в исходном графе, необходимо полученную на первом шаге матрицу R^2 умножить на R. Например,

$$r^3_{1,2} = r_{1,1}r^2_{1,2} + r_{1,2}r^2_{2,2} + r_{1,3}r^2_{3,2} + r_{1,4}r^2_{4,2} = \\ = 0 \cdot 0 + 1 \cdot 2 + 2 \cdot 3 + 0 \cdot 0 = 4.$$

Используя вышеуказанную формулу умножения матриц, получим:

$$\mathrm{R}^3 = \begin{array}{c|cccc|} & x_1 & x_2 & x_3 & x_4 \\ x_1 & 0 & 4 & 4 & 0 \\ x_2 & 4 & 0 & 0 & 4 \\ x_3 & 4 & 0 & 0 & 4 \\ x_4 & 0 & 4 & 4 & 0 \end{array}.$$

Из полученной матрицы R^3 следует, что в исходном графе между вершинами, например, x_1 и x_2, существует 4 маршрута длиной 3: $S_1 = (x_1, x_2)(x_2, x_1)(x_1, x_2)$; $S_2 = (x_1, x_2)(x_2, x_4)(x_4, x_2)$; $S_3 = (x_1, x_3)(x_3, x_1)(x_1, x_2)$; $S_4 = (x_1, x_3)(x_3, x_4)(x_4, x_2)$.

Пример 15.10. Для графа, изображенного на рис. 15.29, привести примеры маршрута, цепи, цикла, простой цепи и простого цикла.

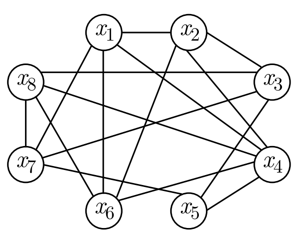


Рис. 15.29. Пример графа G

Ответ. Маршрутом в графе является любая конечная последовательность ребер, причем маршрут может проходить через одну и ту же вершину или ребро любое число раз. Тогда в качестве маршрута можно принять такую конечную последовательность: $S = (x_1, x_2), (x_2, x_6), (x_6 - x_2), (x_2 - x_7), (x_7 - x_6), (x_6 - x_2)$. Цепью называют маршрут, не имеющий повторяющихся ребер. Таким образом, последовательность $S = (x_1, x_2), (x_2, x_4), (x_4, x_6), (x_6, x_2), (x_2, x_7)$ можно считать цепью. Цикл представляет собой замкнутую цепь, т. е. цепь, у которой первая и последняя вершины совпадают. Тогда последовательность $C = (x_1, x_5), (x_5, x_3), (x_3, x_1), (x_1, x_8), (x_8, x_5), (x_5, x_1)$ является циклом. При этом, как мы видим, цепь и цикл могут проходить несколько раз через одну и ту же вершину. Простая цепь (цикл) не может включать в себя ни повторяющихся ребер, ни вершин. Примером простой цепи может служить последовательность $S = (x_1, x_2), (x_2, x_4), (x_4, x_6), (x_6, x_8), (x_8, x_3)$; а простой цикл $C = (x_1, x_5), (x_5, x_3), (x_3, x_1)$.

Пример 15.11. Для графа, заданного на рис. 15.30, привести пример разреза, правильного разреза.

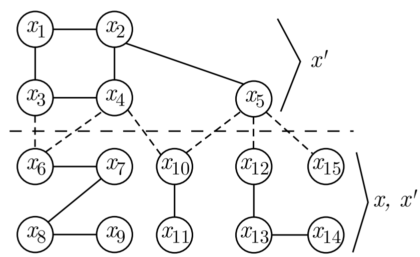


Рис. 15.30. Пример графа G

Ответ. Правильными разрезами будут: $U_1 = \{(x_3, x_6), (x_4, x_6), (x_4, x_7)\}$; $U_2 = \{(x_4, x_{10}), (x_5, x_{10})\}$; $U_3 = \{(x_5, x_{12})\}$; $U_4 = \{(x_5, x_{15})\}$. А разрез U определяется как их объединение: $U = U_1 \cup U_2 \cup U_3 \cup \cup U_4$.

Вопросы для самоконтроля

1. Что из себя представляет маршрут в графе?
2. Как определяется длина маршрута?
3. Что такое маршрут в графе? Как определяется длина маршрута?
4. Дайте определение цепи, цикла, простого цикла и простой цепи в графе.
5. Какой граф называется связным?
6. Что такое компонента связности?
7. Что такое блок графа?
8. Дайте определение разреза, правильного разреза графа. Приведите примеры.
9. Дайте определение вершинной связности. Приведите примеры.
10. Дайте определение реберной связности. Приведите примеры.

Задания для самостоятельной работы

1. Задайте произвольный граф $\mathsf{G} = (X, U)$, $|X| = 5$, $|U| = 7$. Определите количество маршрутов длиной 2, 3, 4.

2. Составьте словесный алгоритм определения маршрутов в графе.

3. Составьте структурную схему алгоритма определения маршрутов в графе.

4. Задайте произвольный граф $\mathsf{G} = (X, U)$, где $|X| = 6$; $|U| = 10$. Постройте маршрут, цепь, цикл, простую цепь, простой цикл. Покажите на примере отличия между этими видами маршрутов.

5. Составьте структурную схему алгоритма или словесный алгоритм определения связности произвольного неориентированного графа.

6. Задайте произвольный граф $\mathsf{G} = (X, U)$, где $|X| = 6$; $|U| = 18$. Приведите пример правильного разреза и разреза графа.

7. Приведите пример графа, имеющего точку сочленения.

8. Приведите пример неразделимого графа.

9. Приведите пример графа блоков.

10. Постройте граф G и покажите на его примере взаимосвязь между числом реберной и вершинной связности и минимальной локальной степенью вершин графа.

При моделировании систем модели и алгоритмы нахождения маршрутов, цепей, циклов, разрезов, связности графов позволяют строить эффективные алгоритмы преобразования графов.

15.3. Нахождение кратчайших маршрутов (цепей)

15.3.1. Алгоритм Форда. Приведем алгоритм Форда нахождения маршрутов с минимальной суммарной стоимостью ребер. Будет анализироваться помеченный граф с весами на ребрах. Граф с весами на ребрах называется *взвешенным*.

1°. Строим матрицу R взвешенного графа G.

Если вершины не смежны между собой, то в матрице ставим M. Фиксируем одну вершину x_k и приписываем ей вес $V(k) = 0$, всем остальным вершинам приписываем вес $V(j) = \mathrm{M}$ или ∞.

2°. Берем произвольную вершину $x_j \neq x_k$ и находим смежную с x_j вершину x_i, имеющую вес $r_{i,j}$. В начальный момент в качестве вершины x_i берем фиксированную x_k. Для x_j и x_i проверяется выполнение неравенства

$$V(j) > V(i) + r_{i,j}. \tag{15.8}$$

Если оно выполняется, то прежний вес $V(j)$ вершины x_j заменяется суммой $V(i) + r_{i,j}$. В противном случае вес $V(j)$ остается неизменным.

3°. Повторяем п. 1, 2, пока не рассмотрены все вершины.

4°. Конец работы алгоритма.

Для уменьшения объема вычислений операцию пересчета индексов сначала выполняют для всех x_j, смежных с фиксированной x_k, а затем продолжают это для вершин, смежных с выбранными ранее.

Операция пересчета производится для всех вершин графа до тех пор, пока ни одна из вершин не сможет принимать нового значения $V(j)$. Тогда длинами кратчайших цепей от x_j до x_k будут величины $V(j)$, полученные в результате замен переменных весов $V(j)$ на новые.

Например, для графа **G** рис. 15.31 выполним алгоритм Форда.

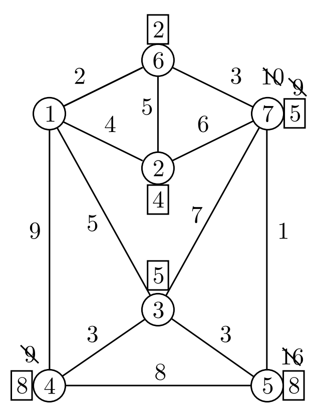


Рис. 15.31. Пример графа **G**

В матрице R по главной диагонали проставлены нули, так как в графе **G** нет петель. При отсутствии связи между вершинами проставлен знак М.

Матрица смежности этого графа запишется так:

$$\mathrm{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}\!\!\begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left|\begin{array}{ccccccc} 0 & 4 & 5 & 9 & \mathrm{M} & 2 & \mathrm{M} \\ 4 & 0 & \mathrm{M} & \mathrm{M} & \mathrm{M} & 5 & 6 \\ 5 & \mathrm{M} & 0 & 3 & 3 & \mathrm{M} & 7 \\ 9 & \mathrm{M} & 3 & 0 & 8 & \mathrm{M} & \mathrm{M} \\ \mathrm{M} & \mathrm{M} & 3 & 8 & 0 & \mathrm{M} & 1 \\ 2 & 5 & \mathrm{M} & \mathrm{M} & \mathrm{M} & 0 & 3 \\ \mathrm{M} & 6 & 7 & \mathrm{M} & 1 & 3 & 0 \end{array}\right| \end{array}.$$

Принимаем вес вершины x_1 равным $V(1) = 0$, $V(2) = V(3) = \ldots = V(7) = \mathrm{M} = \infty$.

1. Фиксированная вершина $x_k = x_1$.

2. Просматриваем все пары x_i, x_j и проверяем выполнение неравенства (15.8).

Пусть $i = 1$, $j = 2$, тогда имеем $V(1) = 0$, $V(2) = \mathrm{M}$, $r_{1,2} = 4$.

Условие неравенства $V(2) = \mathrm{M} > V(1) + r_{1,2} = 0 + 4 = 4$. Поэтому вершине x_2 присваиваем новый вес $V(2)^1 = 4$.

Для вершин, у которых $r_{i,j} = \mathrm{M}$, неравенство (15.8) выполняться не будет и вес $V(j)$ останется прежним.

Далее для x_3: $V(3) = \mathrm{M} > V(1) + r_{1,3} = 0 + 5 = 5$ и $V(3)^1 = 5$.

Для x_4: $V(4) = \mathrm{M} > V(1) + r_{1,4} = 0 + 9 = 9$ и $V(4)^1 = 9$.

Для x_6: $V(6) = \mathrm{M} > V(1) + r_{1,6} = 0 + 2 = 2$ и $V(6)^1 = 2$.

Теперь произведем пересчет индексов для вершин, смежных с x_1, с учетом пути через другие смежные вершины.

Для x_2: $V(2) = 4 < V(6) + r_{2,6} = 2 + 5 = 7$, значит индекс вершины остается без изменения.

Для x_4: $V(4) = 9 > V(3) + r_{4,3} = 5 + 3 = 8$, следовательно, меняем значение $V(4)$ с 9 на 8 (см. рис. 15.31).

Для x_6: $V(6) = 2 < V(2) + r_{2,6} = 4 + 5 = 9$, здесь индекс остается без изменения.

Произведем пересчет индексов для вершин, не смежных с фиксированной вершиной x_1.

Вычислим индекс вершины x_5, его можно найти через x_3, x_4 или x_7. Индекс x_7 равен M, поэтому для нее неравенство (15.8) не выполняется.

Вычислим индекс вершины x_5 через x_4. Получим $V(5)^1_4 = \mathrm{M} > V(4)^1 + r_{4,5} = 8 + 8 = 16$.

Приняв $V(5)^1 = 16$ за старое значение индекса, пересчитаем его через x_3:

$$V(5)^1_3 = 16 > V(3)^1 + r_{3,5} = 5 + 3 = 8.$$

Так как условие (15.8) выполняется, то значение индекса $V(5)^1 = 16$ заменяем на $V(5)^2 = 8$.

Индекс x_7 можно определить через x_2, x_3, x_5, x_6.

$$V(7)_2^1 = \mathrm{M} > V(2)^1 + r_{2,7} = 4 + 6 = 10,$$
$$V(7)_5^1 = \mathrm{M} > V(5)^1 + r_{5,7} = 8 + 1 = 9,$$
$$V(7)_3^1 = \mathrm{M} > V(3)^1 + r_{3,7} = 5 + 7 = 12,$$
$$V(7)_6^1 = \mathrm{M} > V(6)^1 + r_{6,7} = 2 + 3 = 5.$$

Итак, $V(1) = 0$, $V(2) = 4$, $V(3) = 5$, $V(4) = 8$, $V(5) = 8$, $V(6) = 2$, $V(7) = 5$.

Алгоритм Форда дает кратчайшее расстояние от выбранных вершин до фиксированной вершины, а сами цепи не определяет.

Для определения цепи от выбранной вершины до фиксированной воспользуемся равенством $V(j) - V(i) = r_{i,j}$.

Например, найдем путь от x_7 до x_1. Ищется предпоследний шаг при пересчете индекса $V(7)^1$. Это x_7–x_6. Затем для x_6 определяется вершина пересчета. Это x_6–x_1. Следовательно, кратчайший путь x_7–x_6–x_1, а кратчайшее расстояние между вершинами x_1 и x_7 определено ранее $V(7)^1 = 5$.

Предположение, что при выборе на каждом шаге ближайшей вершины ожидаемая длина всего маршрута будет меньше, чем при любом другом выборе, не очевидно.

На рис. 15.32 приведен пример, когда необходимо определить путь из вершины A в B с минимальной стоимостью. При выборе ближайшей вершины 1 из вершины A получим длину маршрута $AB = 13$. Например, длины маршрутов A–1–6–7–B или A–1–2–7–8–B равны 13. В то же время все другие маршруты (через вершину 3) дают длину маршрута $AB = 8$ (например, A–3–4–5–B).

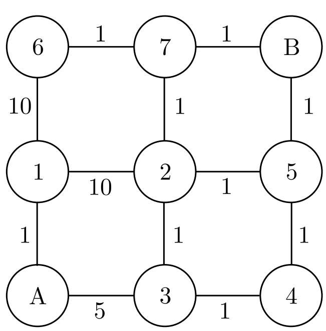


Рис. 15.32. Пример нахождения минимального маршрута

Отметим, что здесь необходима стратегия просмотра длины маршрута в глубину на несколько шагов. Хотя в общем случае этот метод может свестись к полному перебору вариантов.

15.3.2. Алгоритм Дейкстры. Рассмотрим алгоритм Дейкстры нахождения кратчайшей цепи между двумя заданными вершинами x_s и x_t.

Метод основан на приписывании вершинам временных пометок, причем пометка вершины x_i дает верхнюю границу длины пути от вершины x_s к этой вершине. Значения этих пометок постепенно уменьшаются на основе итерационной процедуры и на каждом шаге итерации точно одна из временных пометок становится постоянной. Это говорит о том, что пометка дает длину кратчайшей цепи от вершины x_s к рассматриваемой вершине.

Алгоритм.

Пусть $l(x_i)$ — пометка x_i (блок присвоения начальных значений).

1°. Пусть $l(s) = 0$. Положим $l(x_i) = \infty$ для всех $x_i \neq s$ и считаем их пометки временными; пусть $p = s$ (блок обновления пометок).

2°. Для всех $x_i \in \Gamma(p)$, пометки которых временные, изменить пометки в соответствии с выражением

$$l(x_i) \leftarrow \min\left[l(x_i);\ l(p) + r(p, x_i)\right] \qquad (15.9)$$

(блок превращения пометки в постоянную).

3°. Среди всех вершин с временными пометками найти такую, что $l(x_i^*) \min\left[l(x_i)\right]$.

4°. Считать пометку x_i^* постоянной и положить $p = x_i^*$.

5°.

а) *Если надо построить цепь от вершины x_s к вершине x_t.*

Если $p = t$, то $l(p)$ — длина кратчайшей цепи, останов.

Если $p \neq t$, переход к п. 2°.

б) *Если надо построить цепь от вершины x_s ко всем остальным вершинам.*

Если все вершины отмечены как постоянные, то эти пометки дают длины кратчайших цепей, останов.

Если некоторые вершины являются временными, переход к п. 2°.

Рассмотрим последовательность выполнения алгоритма Дейкстры на примере построения кратчайших цепей в графе (см. рис. 15.31).

Матрица смежности графа **G** имеет следующий вид:

$$
\mathrm{R} = \begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & 4 & 5 & 9 & 0 & 2 & 0 \\
2 & 4 & 0 & 0 & 0 & 0 & 5 & 6 \\
3 & 5 & 0 & 0 & 3 & 3 & 0 & 7 \\
4 & 9 & 0 & 3 & 0 & 8 & 0 & 0 \\
5 & 0 & 0 & 3 & 8 & 0 & 0 & 1 \\
6 & 2 & 5 & 0 & 0 & 0 & 0 & 3 \\
7 & 0 & 6 & 7 & 0 & 1 & 3 & 0
\end{array}.
$$

Постоянные пометки будем обозначать *, остальные считаются временными.

Вначале нам необходимо выбрать начальную вершину x_s, которую мы будем использовать в дальнейшем в качестве фиксированной. Примем в качестве такой вершины x_1. Метки всех остальных вершин, отличных от вершины x_1, считаем равными ∞, а значение $p = x_1$.

1°. $L(x_1) = 0^*$, $L(x_i) = \infty$, $\forall x_i \neq x_1$, $p = x_1$.

Теперь приступим к выполнению алгоритма нахождения кратчайших по стоимости путей.

Первая итерация.

2°. Для всех вершин, смежных с начальной вершиной x_1 и имеющих временные метки $\Gamma(x_1) = \{x_2, x_3, x_4, x_6\}$, пересчитываем в соответствии с выражением (15.9) действующие значения их меток.

Для $l(x_2) \leftarrow \min\,[\infty; 0^* + 4] = 4$.
Для $l(x_3) \leftarrow \min\,[\infty; 0^* + 5] = 5$.
Для $l(x_4) \leftarrow \min\,[\infty; 0^* + 9] = 9$.
Для $l(x_6) \leftarrow \min\,[\infty; 0^* + 2] = 2$.

3°. Выбираем из списка вершин с временными метками вершину с временной меткой, имеющей минимальное значение:

$$
\begin{array}{cccccccc}
\min & [4, & 5, & 9, & \infty, & 2, & \infty] \\
 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7
\end{array}
$$

Очевидно, что такой вершиной является вершина x_6.

4°. Вершина x_6 получает постоянную пометку $l(x_6) = 2^*$, значение $p = x_6$.

5°. Поскольку не все вершины имеют постоянные пометки, переходим к п. 2°.

Вторая итерация.

2°. Для всех вершин, смежных с текущей вершиной x_6 и имеющих временные метки $\Gamma(x_6) = \{x_2, x_7\}$, пересчитываем в соответствии с выражением (15.9) действующие значения их меток.

Для $l(x_2) \leftarrow \min[4; 2^* + 5] = 4$.

Для $l(x_7) \leftarrow \min[\infty; 2^* + 3] = 5$.

Новое значение метки для вершины x_2 оказалось больше текущего, поэтому оставляем текущее значение временной метки, а для вершины x_7 заменяем значение $l(x_7) = \infty$ на полученное значение $l(x_7) = 5$.

3°. Выбираем из списка вершин с временными метками вершину с временной меткой, имеющей минимальное значение:

$$\min \begin{array}{ccccc} [4, & 5, & 9, & \infty, & 5] \\ x_2 & x_3 & x_4 & x_5 & x_7 \end{array}$$

Очевидно, что такой вершиной является вершина x_2.

4°. Вершина x_2 получает постоянную пометку $l(x_2) = 4^*$, значение $p = x_2$.

5°. Поскольку не все вершины имеют постоянные пометки, переходим к п. 2°.

Третья итерация.

2°. Для всех вершин, смежных с текущей вершиной x_2 и имеющих временные метки $\Gamma(x_2) = \{x_7\}$, пересчитываем в соответствии с выражением (15.9) действующие значения их меток.

Для $l(x_7) \leftarrow \min[5; 4^* + 6] = 5$.

Новое значение метки для вершины x_7 оказалось больше текущего, поэтому оставляем текущее значение временной метки $l(x_7) = 5$.

3°. Выбираем из списка вершин с временными метками вершину с временной меткой, имеющей минимальное значение:

$$\min \begin{array}{cccc} [5, & 9, & \infty, & 5] \\ x_3 & x_4 & x_5 & x_7 \end{array}$$

В данном случае две вершины имеют одинаковые метки $l(x_3) = l(x_7) = 5$ и мы можем выбрать любую из них. Пусть это будет вершина x_3.

4°. Вершина x_3 получает постоянную пометку $l(x_3) = 5^*$, значение $p = x_3$.

5°. Поскольку не все вершины имеют постоянные пометки, переходим к п. 2°.

Четвертая итерация.

2°. Для всех вершин, смежных с текущей вершиной x_3 и имеющих временные метки $\Gamma(x_3) = \{x_4, x_5, x_7\}$, пересчитываем в соответствии с выражением (15.9) действующие значения их меток:

для $l(x_4) \leftarrow \min\,[9; 5^* + 3] = 8$,
для $l(x_5) \leftarrow \min\,[\infty; 5^* + 3] = 8$,
для $l(x_7) \leftarrow \min\,[5; 5^* + 7] = 5$.

Новое значение метки для вершины x_7 оказалось больше текущего, поэтому оставляем текущее значение временной метки, а для вершин x_4 и x_5 заменяем текущие значения меток на новые.

3°. Выбираем из списка вершин с временными метками вершину с временной меткой, имеющей минимальное значение:

$$\begin{array}{cccc} \min & [8, & 8, & 5] \\ & x_4 & x_5 & x_7 \end{array}$$

Очевидно, что такой вершиной является вершина x_7.

4°. Вершина x_7 получает постоянную пометку $l(x_7) = 5^*$, значение $p = x_7$.

5°. Не все вершины имеют постоянные пометки, переходим к п. 2°.

Пятая итерация.

2°. Для всех вершин, смежных с текущей вершиной x_7 и имеющих временные метки $\Gamma(x_7) = \{x_5\}$, пересчитываем в соответствии с выражением (15.9) действующие значения их меток.

Для $l(x_5) \leftarrow \min\,[8; 5^* + 1] = 6$.

3°. Выбираем из списка вершин с временными метками вершину с временной меткой, имеющей минимальное значение:

$$\begin{array}{ccc} \min & [8, & 6] \\ & x_4 & x_5 \end{array}$$

Такой вершиной является вершина x_5.

4°. Вершина x_5 получает постоянную пометку $l(x_5) = 6^*$, значение $p = x_5$.

5°. Поскольку не все вершины имеют постоянные пометки, переходим к п. 2°.

Шестая итерация.

2°. Из вершин, смежных с текущей вершиной x_3 и имеющих временные метки, осталась только вершина x_4: $\Gamma(x_1) = \{x_4\}$.

Пересчитываем значение ее метки в соответствии с выражением (15.9):

для $l(x_4) \leftarrow \min[8; 6^* + 8] = 8$.

Новое значение метки для вершины x_4 оказалось больше текущего, поэтому оставляем текущее значение временной метки.

3°. В списке вершин с временными метками только один элемент:

$$\min_{x_4} \;[8]$$

4°. Вершина x_4 получает постоянную пометку $l(x_4) = 8^*$, значение $p = x_4$.

5°. Поскольку все вершины графа имеют постоянные метки, конец алгоритма.

Таким образом, мы построили кратчайшие пути от вершины x_1 до вершин x_2, x_3, x_6 напрямую:

$$x_1 \to x_2(4^*),\; x_1 \to x_3(5^*),\; x_1 \to x_6(2^*),$$

а до вершин x_4, x_5, x_7 через промежуточные:

$$x_1 \to x_3 \to x_4(8^*);$$
$$x_1 \to x_6 \to x_7 \to x_5(6^*);$$
$$x_1 \to x_6 \to x_7(5^*).$$

Длину цепи $x_s \to x_i$, т. е. суммарную стоимость весов пройденных ребер, можно получить при помощи рекурсивной процедуры:

$$l(x_s) + r(x_s, x_j) = l(x_i),$$

т. е. проверяется, какая из вершин, смежных вершине x_s, дает вместе с весом ребра $r(x_s, x_j)$ сумму, равную $l(x_i)$.

Например: $l(x_4) = 8$. Поскольку вершина x_4 напрямую не связана с вершиной x_1, то определим кратчайший путь из x_1 в x_4:

$$l(x_1) = l(x_4) - r(x_3, x_4) = 8-3=5=l(x_3) - r(x_1, x_3) = 5 - 5 = 0.$$

Таким образом, построена кратчайшая цепь между вершинами x_1 и x_4: $x_1 \to x_3 \to x_4$.

Примеры решения задач

Пример 15.12. Пусть задан граф $\mathsf{G} = (X, U)$. Найти с помощью алгоритма Форда кратчайшие пути в заданном графе G от фиксированной вершины до всех остальных.

Решение. Рассмотрим матрицу смежности графа **G**:

$$
\mathrm{R} = \begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 0 & 10 & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & 3 & 6 & 12 \\
2 & 10 & 0 & 18 & \mathrm{M} & \mathrm{M} & \mathrm{M} & 2 & \mathrm{M} & 13 \\
3 & \mathrm{M} & 18 & 0 & 25 & \mathrm{M} & 20 & \mathrm{M} & \mathrm{M} & 7 \\
4 & \mathrm{M} & \mathrm{M} & 25 & 0 & 5 & 16 & 4 & \mathrm{M} & \mathrm{M} \\
5 & \mathrm{M} & \mathrm{M} & \mathrm{M} & 5 & 0 & 10 & \mathrm{M} & 23 & \mathrm{M} \\
6 & \mathrm{M} & \mathrm{M} & 20 & 16 & 10 & 0 & 17 & 15 & 9 \\
7 & 3 & 2 & \mathrm{M} & 4 & \mathrm{M} & 17 & 0 & \mathrm{M} & 24 \\
8 & 6 & \mathrm{M} & \mathrm{M} & \mathrm{M} & 23 & 15 & \mathrm{M} & 0 & 5 \\
9 & 12 & 13 & 7 & \mathrm{M} & \mathrm{M} & 9 & 24 & 5 & 0
\end{array}.
$$

1.1. В качестве фиксированной вершины выбираем вершину x_1. Тогда ее вес $V(1) = 0$, а веса всех остальных вершин: $V(2) = V(3) = \ldots = V(9) = \infty = \mathrm{M}$.

1.2. Просматриваем первую строку матрицы смежности и выбираем поочередно все вершины, не помеченные литерой M. В результате мы сформировали следующий список вершин: x_2, x_7, x_8, x_9.

1.3. Для каждой из вершин, входящих в полученный список, проверяем истинность неравенства (15.8):

$$V(2) = \infty > 0 + 10 = 10; \quad V(7) = \infty > 0 + 3 = 3;$$

$$V(8) = \infty > 0 + 6 = 6; \quad V(9) = \infty > 0 + 12 = 12.$$

Значения весов для вершин x_3, x_4, x_5, x_6 не изменяются.

Теперь произведем пересчет индексов для вершин, смежных с x_1, с учетом пути через другие смежные вершины.

1.4. Производим пересчет весов для сформированного списка вершин.

Для вершины x_2:

$V(2) > V(7) + r_{27} \to 10 > 3 + 2$ — неравенство истинно, следовательно, вес вершины изменяется: $V(2)^1 = 5$;

$V(2) > V(9) + r_{29} \to 10 > 1 + 13$ — неравенство ложно, следовательно, вес вершины не изменяется;

для вершины x_7:

$V(7) > V(2) + r_{27} \to 3 > 10 + 2$ — неравенство ложно, следовательно, вес вершины не изменяется;

$V(7) > V(9) + r_{79} \to 3 > 12 + 24$ — неравенство ложно, следовательно, вес вершины не изменяется;

для вершины x_8:

$V(8) > V(9) + r_{89} \to 6 > 12 + 5$ — неравенство ложно, следовательно, вес вершины не изменяется;

для вершины x_9:

$V(9) > V(2) + r_{29} \to 12 > 10 + 13$ — неравенство ложно, следовательно, вес вершины не изменяется;

$V(9) > V(7) + r_{79} \to 12 > 3 + 24$ — неравенство ложно, следовательно, вес вершины не изменяется;

$V(9) > V(8) + r_{89} \to 12 > 6 + 5$ — неравенство истинно, следовательно, вес вершины изменяется: $V(9)^1 = 11$.

1.5. Проверка условия: все ли вершины помечены.

Произведем пересчет индексов для вершин, не смежных с фиксированной вершиной x_1.

2.1. Вычислим значение веса вершины x_3, его можно найти через x_2, x_4, x_6 или x_9. Однако значения весов вершин x_4 и x_6 нам пока не известны, поэтому производим расчет через вершины x_2 и x_9:

$V(3) > V(2) + r_{23} \to \infty > 5 + 18$ — неравенство истинно, следовательно, вес вершины изменяется: $V(3)^1 = 23$;

$V(3) > V(9) + r_{39} \to 23 > 11 + 7$ — неравенство истинно, следовательно, вес вершины изменяется: $V(3)^2 = 18$.

2.2. Вычислим значение веса вершины x_4, его можно найти через x_3, x_5, x_6 или x_7. Однако значения весов вершин x_5 и x_6 нам пока не известны, поэтому производим расчет через вершины x_3 и x_7:

$V(4) > V(3) + r_{34} \to \infty > 18 + 25$ — неравенство истинно, следовательно, вес вершины изменяется: $V(4)^1 = 43$;

$V(4) > V(7) + r_{47} \to 43 > 3 + 4$ — неравенство истинно, следовательно, вес вершины изменяется: $V(4)^2 = 7$.

2.3. Вычислим значение веса вершины x_5, его можно найти через x_4, x_6 или x_8. Однако значение веса вершины x_6 нам пока не известно, поэтому производим расчет через вершины x_4 и x_8:

$V(5) > V(4) + r_{45} \to \infty > 7 + 5$ — неравенство истинно, следовательно, вес вершины изменяется: $V(5)^1 = 12$;

$V(5) > V(8) + r_{58} \to 12 > 6 + 23$ — неравенство ложно, следовательно, вес вершины не изменяется.

2.4. Вычислим значение веса вершины x_6, его можно найти через x_3, x_4, x_5, x_7, x_8 или x_9:

$V(6) > V(3) + r_{36} \to \infty > 18 + 20$ — неравенство истинно, следовательно, вес вершины изменяется: $V(6)^1 = 38$;

$V(6) > V(4) + r_{46} \to 38 > 7 + 16$ — неравенство истинно, следовательно, вес вершины изменяется: $V(6)^2 = 23$;

$V(6) > V(5) + r_{56} \to 23 > 12 + 10$ — неравенство истинно, следовательно, вес вершины изменяется: $V(6)^3 = 22$;

$V(6) > V(7) + r_{67} \to 22 > 3 + 17$ — неравенство истинно, следовательно, вес вершины изменяется: $V(6)^4 = 20$;

$V(6) > V(8) + r_{68} \to 20 > 6 + 15$ — неравенство ложно, следовательно, вес вершины не изменяется;

$V(6) > V(9) + r_{69} \to 20 > 11 + 9$ — неравенство ложно, следовательно, вес вершины не изменяется.

3. Конец работы алгоритма.

Алгоритм Форда дает кратчайшее расстояние от вершин до фиксированной вершины, а сами цепи не определяет.

Для определения цепи от выбранной вершины до фиксированной воспользуемся равенством $V(j) - V(i) = r_{i,j}$.

Например, найдем путь от x_6 до x_1. Ищется предпоследний шаг при пересчете индекса $V(6)$. Это x_6–x_7. Затем для x_7 определяется вершина пересчета. Это x_6–x_1. Следовательно, кратчайший путь x_6–x_7–x_1, а кратчайшее расстояние между вершинами x_1 и x_6 определено ранее: $V(6) = 20$.

Пример 15.13. Пусть задан граф $\mathsf{G} = (X, U)$. Найти с помощью алгоритма Дейкстры кратчайшие пути в заданном графе G от фиксированной вершины до всех остальных.

Решение. Рассмотрим матрицу смежности графа G:

$$\mathrm{R} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9 \end{array} \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \left| \begin{matrix} 0 & 10 & 0 & 0 & 0 & 0 & 3 & 6 & 12 \\ 10 & 0 & 18 & 0 & 0 & 0 & 2 & 0 & 13 \\ 0 & 18 & 0 & 25 & 0 & 20 & 0 & 0 & 7 \\ 0 & 0 & 25 & 0 & 5 & 16 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 10 & 0 & 23 & 0 \\ 0 & 0 & 20 & 16 & 10 & 0 & 17 & 15 & 9 \\ 3 & 2 & 0 & 4 & 0 & 17 & 0 & 0 & 24 \\ 6 & 0 & 0 & 0 & 23 & 15 & 0 & 0 & 5 \\ 12 & 13 & 7 & 0 & 0 & 9 & 24 & 5 & 0 \end{matrix} \right| \end{array}.$$

Постоянные пометки будем обозначать *, остальные временные.

1°. $l(x_1) = 0^*$. $l(x_i) = \infty$, $\forall x_i \neq x_1$, $p = x_1$.

В качестве начальной вершины принимаем вершину x_1. Всем остальным вершинам графа присваиваем метки $l(x_i) = \infty$.

Первая итерация.

2°. $\Gamma(p) = \Gamma(x_1) = \{x_2, x_7, x_8, x_9\}$. Возьмем вершину x_2 и из выражения (15.9) получим:

$$l(x_2) \leftarrow \min\,[l(x_2), l(x_1) + r(x_1, x_2)] = \min\,[\infty, 0^* + 10] = 10.$$

Аналогично вычисляем значения меток для других вершин, смежных с вершиной x_1: $l(x_7) = 3$, $l(x_8) = 6$, $l(x_9) = 12$.

3°. В списке вершин графа выберем вершину, имеющую минимальное значение метки: $\min\,[x_2, x_7, x_8, x_9, x_3, x_4, x_5, x_6] = \min\,[10, 3, 6, 12, \infty, \infty, \infty, \infty] = 3$ — соответствует вершине x_7.

4°. Вершина x_7 получает постоянную пометку $l(x_7) = 3^*$, $p = x_7$.

5°. Не все вершины имеют постоянные пометки, переходим к п. 2°.

Вторая итерация.

2°. $\Gamma(p) = \Gamma(x_7) = \{x_2, x_4, x_6, x_9\}$ — все пометки временные. Из выражения (15.9) вычисляем:

$$l(x_2) = \min\,[l(x_2), l(x_7) + r(x_7, x_2)] = \min\,[10, 3^* + 2] = 5.$$

Аналогично $l(x_4) = 7$, $l(x_6) = 17$, $l(x_9) = 12$.

3°. $\min\,[x_2, x_4, x_6, x_8, x_9, x_3, x_5] = \min\,[5, 7, 17, 6, 12, \infty, \infty] = 5$ — соответствует вершине x_2.

4°. Вершина x_2 получает постоянную пометку $l(x_2) = 5^*$, $p = x_2$.

5°. Перейдем к п. 2°, так как не все вершины имеют постоянные пометки.

Третья итерация.

2°. $\Gamma(p) = \Gamma(x_2) = \{x_1, x_3, x_7, x_9\}$, только x_3 и x_9 имеют временные пометки. Из выражения (15.9) получим $l(x_3) = \min\,[\infty, 5^* + 18] = 23$, $l(x_9) = \min\,[12, 5^* + 13] = 12$.

3°. $\min\,[x_3, x_4, x_8, x_6, x_9, x_5] = \min\,[23, 7, 6, 17, 12, \infty] = 6$ — соответствует вершине x_8.

4°. Вершина x_8 получает постоянную пометку $l(x_8) = 6^*$, $p = x_8$.

5°. Перейдем к п. 2°, так как не все вершины имеют постоянные пометки.

Четвертая итерация.

2°. $\Gamma(p) = \Gamma(x_8) = \{x_9, x_6, x_5\}$, x_5 и x_9 имеют временные пометки. Из выражения (15.9) получим:

$$l(x_5) = \min\,[\infty, 6^* + 23] = 29,$$

$$l(x_9) = \min[12, 6^* + r(x_8, x_9)] = 11.$$

3°. $\min[x_5, x_9, x_3, x_4, x_5] = \min[29, 11, 23, 7, 29] = 7$ — соответствует вершине x_4.

4°. Вершина x_4 получает постоянную пометку $l(x_4) = 7^*$, $p = x_4$.

5°. Переход к п. 2°.

Пятая итерация.

2°. $\Gamma(p) = \Gamma(x_4) = \{x_3, x_6, x_5\}$, $l(x_3) = 23$, $l(x_6) = 17$, $l(x_5) = 12$.

3°. $\min[x_5, x_6, x_9, x_3] = \min[12, 17, 11, 23] = 11$.

4°. Вершина x_9 получает постоянную пометку $l(x_9) = 11^*$, $p = x_9$.

5°. Переход к п. 2°.

Шестая итерация.

2°. $\Gamma(p) = \Gamma(x_9) = \{x_3, x_6\}$ (берем только непомеченные вершины).

$$(x_6) = \min[l(x_6), l(x_9) + r(x_6, x_9)] = [17, 11^* + 9] = 17,\ l(x_3) = 18.$$

3°. $\min[x_5, x_6, x_3] = \min[12, 17, 18] = 12$.

4°. Вершина x_5 получает постоянную пометку $l(x_5) = 12^*$, $p = x_5$.

5°. Переход к п. 2°.

Седьмая итерация.

2°. $\Gamma(p) = \Gamma(x_5) = \{x_6\}$.

$$l(x_6) = \min[l(x_6), l(x_5) + r(x_5, x_6)] = [17, 12^* + 10] = 17.$$

3°. $\min[x_6, x_3] = \min[17, 18] = 17$.

4°. Вершина x_6 получает постоянную пометку $l(x_6) = 17^*$, $p = x_6$.

5°. Переход к п. 2°.

Восьмая итерация.

2°. $\Gamma(p) = \Gamma(x_3) = \varnothing$, $l(x_3) = \min[18] = 18$, $p = x_3$. Все вершины помечены.

Итак, получили кратчайшие пути от вершины x_1 до всех остальных: $x_1 \to x_2(5^*)$, $x_1 \to x_3(18^*)$, $x_1 \to x_4(7^*)$, $x_1 \to x_5(12^*)$, $x_1 \to x_6(17^*)$, $x_1 \to x_7(3^*)$, $x_1 \to x_8(6^*)$, $x_1 \to x_9(11^*)$;

$x_1 \to x_7 \to x_2 \to x_3(18^*)$, $x_1 \to x_7 \to x_2(5^*)$, $x_1 \to x_8(6^*)$, $x_1 \to x_8 \to x_9(11^*)$, $x_1 \to x_7 \to x_6(17^*)$, $x_1 \to x_7 \to x_4 \to x_5(12^*)$, $x_1 \to x_7 \to x_4(7^*)$.

Длину цепи, т. е. суммарную стоимость весов пройденных ребер, можно получить при помощи описанной ранее рекурсивной процедуры.

Например: $x_1 \to x_3(18^*)$, сумму 18 дает вершина x_9 с ребром $\{x_9{-}x_3\}$;

имеем $x_3 \to x_9(11^*)$, сумму 11 дает вершина x_8 с ребром $\{x_8{-}x_9\}$;

имеем $x_3 \to x_9 \to x_8(6^*)$, сумму 6 дает вершина x_1 с ребром $\{x_1{-}x_8\}$.

Тогда имеем кратчайший путь $x_3 \to x_9 \to x_8 \to x_1$ длиной 18.

Результат построения кратчайших путей в графе G показан на рис. 15.33.

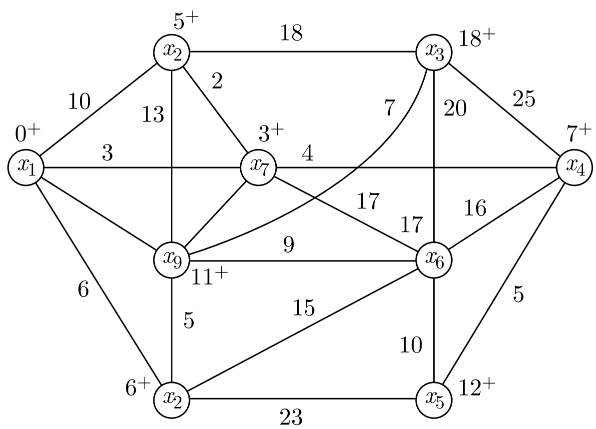


Рис. 15.33. Граф G

Вопросы для самоконтроля

1. Какой граф называется взвешенным?
2. Как определяется кратчайшая цепь в графе?
3. Что представляет собой алгоритм Форда?
4. Для чего используется алгоритм Форда?
5. Какова сложность алгоритма Форда?
6. На чем основан метод Дейкстры?
7. Для чего используется алгоритм Дейкстры?
8. Какова сложность алгоритма Дейкстры?

9. Как определяется последовательность прохождения вершин в кратчайшей цепи?

10. В каком случае алгоритмы Форда и Дейкстры не позволяют найти кратчайший по стоимости путь?

Задания для самостоятельной работы

1. Задан произвольный взвешенный неориентированный граф: $\mathsf{G} = (X, U)$, $|X| = 9$; $|U| = 20$. Найдите кратчайшие пути между всеми вершинами заданного графа, используя метод Форда.

2. Запишите словесный алгоритм нахождения кратчайших путей в графе (по методу Форда).

3. Постройте структурную схему алгоритма Форда нахождения кратчайших путей в графе.

4. Задан произвольный взвешенный неориентированный граф: $\mathsf{G} = (X, U)$, $|X| = 9$; $|U| = 20$. Найдите кратчайшие пути между всеми вершинами заданного графа, используя метод Дейкстры.

5. Запишите словесный алгоритм нахождения кратчайших путей в графе (по методу Дейкстры).

6. Постройте структурную схему алгоритма Дейкстры нахождения кратчайших в графе.

7. Приведите псевдокод алгоритма Форда.

8. Приведите псевдокод алгоритма Дейкстры.

9. Определите временную сложность алгоритма Форда. Проведите сравнительную оценку с алгоритмом Дейкстры по времени работы и объему занимаемой памяти.

10. Определите временную сложность алгоритма Дейкстры. Проведите сравнительную оценку с алгоритмом Форда по времени работы и объему занимаемой памяти.

Алгоритмы Форда и Дейкстры являются эффективным способом определения кратчайшей по стоимости цепи между двумя вершинами графа.

Глава 16

СПЕЦИАЛЬНЫЕ ЦИКЛЫ И МЕТРИКА ГРАФОВ

Математика позволяет познавать многое в окружающем нас реальном мире и овладевать им.

М. Клайн

Эйлеров цикл, *эйлерова цепь*, *гамильтонов цикл*, *гамильтонова цепь*, *жорданова кривая*, *алгоритм построения гамильтонова цикла в графе*, *задача коммивояжера*, *метрика графа*, *расстояние между вершинами*, *матрица расстояний*, *матрица геометрии*, *диаметр*, *максимальное удаление*, *центр графа*, *протяженность*, *число протяженности*, *координатная решетка*, *дерево*, *корень*, *покрывающее дерево*, *кратчайшее покрывающее дерево*, *алгоритм построения кратчайшего покрывающего дерева*, *дерево Штейнера*

ЦЕЛИ

Освоив эту главу, студенты должны:

- знать определение и условия существования эйлерова цикла;
- уметь строить эйлеров цикл в графе;
- решать задачу построения гамильтонова цикла в графе с помощью алгоритма Робертса–Флореса;
- уметь строить гамильтонов цикл в графе алгебраическим методом;
- решать задачу коммивояжера с помощью алгоритма Хелда–Карпа;
- решать задачу коммивояжера геометрическим методом;
- иметь представление о метрике графа;
- знать, каким образом определяется расстояние между вершинами графа;
- уметь строить матрицу расстояний и матрицу геометрии;
- знать, каким образом определяется диаметр и центр графа;
- знать определение дерева и покрывающего дерева в графе;

• уметь строить кратчайшее покрывающее дерево в графе на основе метода Прима;

• уметь строить кратчайшее покрывающее дерево в графе на основе алгоритма Краскала;

• уметь строить кратчайшее покрывающее дерево Штейнера в графе, отображенном в координатную решетку.

16.1. Эйлеровы и гамильтоновы цепи и циклы

При решении задач науки и техники важное значение имеют графы специального вида — эйлеровы и гамильтоновы.

Постановка задачи о кенигсбергских мостах знаменует начало создания теории графов. На рис. 16.1 показан условный план города, а на рис. 16.2 — его модель в виде графа. Здесь A, B, C, D (рис. 16.1) — суша, a, b, c, d, e, f, g — мосты, соединяющие сушу между собой.

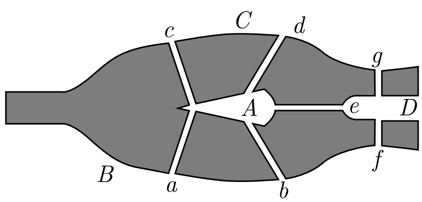


Рис. 16.1. Условный план города

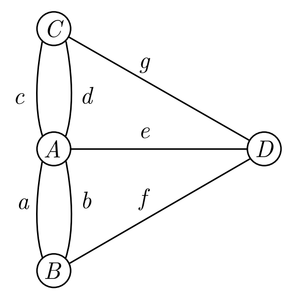


Рис. 16.2. Модель плана города в виде графа

Требуется пройти каждый мост по одному разу и вернуться в исходную часть города. На рис. 16.2 вершины графа — части города, а ребра графа — мосты.

Все вершины графа на рис. 16.2 имеют нечетную степень, следовательно, решения нет.

Теорема 16.1 (Л. Эйлер, 1736 г.) *Связный неориентированный граф содержит эйлеров цикл* (*эйлерову цепь*) *тогда и только тогда*, *когда число вершин нечетной степени равно* 0 (*или* 2).

Доказательство для цикла:

Необходимость. Любой Эйлеров цикл (ЭЦ) должен прийти в вершину по одному ребру и покинуть ее по другому, так как любое ребро должно использоваться только один раз. Поэтому если G содержит ЭЦ, то степени всех вершин должны быть четными.

Достаточность. Пусть G — связный неорграф, все вершины которого имеют четную степень. Начнем путь из произвольной вершины x_0 и пойдем по некоторому ранее не использованному ребру к следующей вершине и так до тех пор, пока не вернемся в x_0 и не замкнем цикл. Если все ребра использованы, то ЭЦ построен. Если некоторые ребра не использованы, то процесс продолжается. Пусть Ф — построенный цикл (рис. 16.3) и Ф1 — цикл с неиспользованными ребрами.

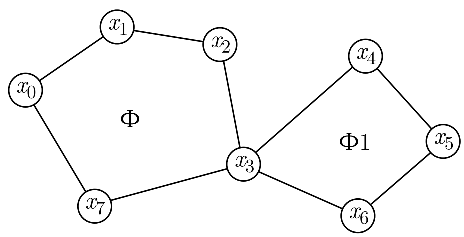


Рис. 16.3. Пример построения эйлерова цикла

Так как G связен, то Ф должен проходить через вершину, являющуюся концом неиспользованного ребра.

Если удалить ребра Ф, то в оставшемся графе вершины по-прежнему будут иметь четную степень, так как в Ф должно быть четное число ребер, инцидентных каждой вершине. Начиная с x_i получаем цикл Ф1. Если все ребра использованы, то ЭЦ построен путем $\Phi \cup \Phi 1$. Если остались неиспользованные ребра, то процесс снова продолжается до тех пор, пока не будут использованы все ребра и построен ЭЦ Ф (рис. 16.4). Это доказывает теорему.

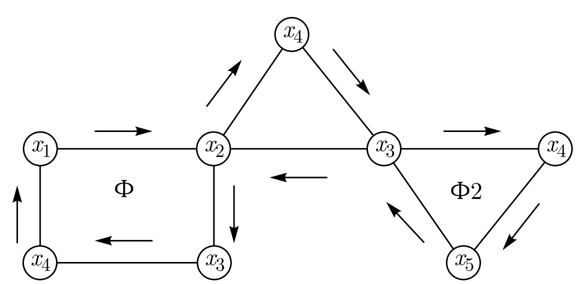


Рис. 16.4. Пример эйлерова цикла в графе

Граф называется *полуэйлеровым*, если существует незамкнутая цепь, проходящая через каждое ребро графа только один раз. Конечный граф **G** является *эйлеровым*, если он связен и все его локальные степени четны. Если эйлеров цикл $C_э$ существует, то, проходя по его ребрам, можно нарисовать эйлеров граф на бумаге, не отрывая карандаша. На рис. 16.5–16.7 показаны неэйлеров (рис. 16.5), полуэйлеров (рис. 16.6) и эйлеров (рис. 16.7) графы. Порядок обхода полуэйлерова и эйлерова графов показан стрелками.

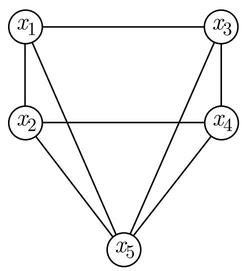


Рис. 16.5. Неэйлеров граф

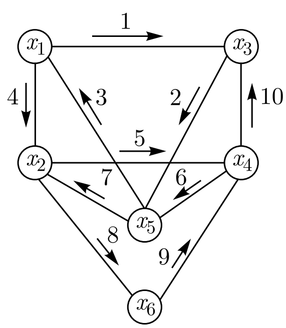


Рис. 16.6. Полуэйлеров граф

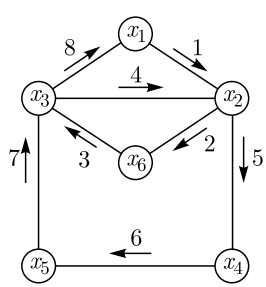


Рис. 16.7. Эйлеров граф

Например, граф **G** (рис. 16.6) не является эйлеровым, так как степени вершин x_1, x_3 нечетны. Граф **G** также не является эйлеровым, если он не связен, хотя его компоненты могут являться эйлеровыми графами.

Заметим, что связный граф **G** является полуэйлеровым, когда в нем не более двух вершин имеют нечетные локальные степени. Причем одна из этих вершин будет начальной, а другая — конечной.

Запишем алгоритм Флери построения $C_э$ в связном графе $\mathsf{G} = (X, U)$, локальные степени которого четны.

1°. Пусть начальная вершина цикла $x_k = x_1$. Число ребер U графа G равно $|U| = n$, а x_s — произвольно выбранная вершина графа. Эйлеров цикл $c_i = \varnothing$.

2°. Определяем множество ребер, смежных вершине x_s.

3°. Выбираем ребро, удаление которого не приведет к разбиению графа на два компонента связности (кроме изолированных вершин). Пусть это будет ребро $u_k = (x_s, x_p)$.

4°. Тогда добавляем это ребро в эйлеров цикл $C_k = C_k \cup u_k$ и удаляем его из графа G.

5°. Если выполняется условие $|C_k| = n$, то переход к п. 7°, иначе переход к п. 6°.

6°. Присваиваем $k = k + 1$, $s = p$ и переходим к п. 2°.

7°. Эйлеров цикл построен, конец работы алгоритма.

Также можно привести еще один вариант записи алгоритма Флери.

1°. Выбирается произвольная вершина x_i, $i \in I = \{1, 2, \ldots, n\}$.

2°. Определяется произвольное ребро $(x_i, x_j) j \neq i$, $j \in I$, инцидентное вершине x_i, ему присваивается номер 1. Это ребро заносится в список $L_э$ и вычеркивается из графа G.

3°. Рассматривается вершина x_j, если есть возможность другого выбора, не выбираются ребра (x_j, x_k), ему присваивается номер, оно заносится в список $L_э$ и вычеркивается из G.

4°. Производится проверка $|L_э| = m$. Если да, то переход к п. 5°, если нет, то п. 3° повторяется для вершин $x_b, \ldots, x_n$.

5°. Построен эйлеров цикл $C_э$, конец работы алгоритма.

На рис. 16.7 стрелками показан пример работы алгоритма $C_э = (x_1, x_2), (x_2, x_6), (x_6, x_3), (x_3, x_2), (x_2, x_4), (x_4, x_5), (x_5, x_3), (x_3, x_1)$.

Цикл, проходящий по всем вершинам графа G один раз, называется *гамильтоновым*, а G называется *гамильтоновым графом*. Например, граф G (рис. 16.7) не имеет гамильтонова цикла (ГЦ), а граф G (рис. 16.5) имеет $\mathsf{G}_г = (x_1, x_2), (x_2, x_5), (x_5, x_4), (x_4, x_3), (x_3, x_1)$. В отличие от эйлеровых циклов для ГЦ неизвестен общий критерий существования. В основном известны только теоремы, дающие достаточные условия существования ГЦ.

Теорема 16.2. *Если в графе G с n вершинами для любой пары несмежных вершин x_i, x_j верно: $\rho(x_i) + \rho(x_j) \geqslant n$, то G имеет ГЦ.*

Из теоремы следует результат Дирака, что граф имеет ГЦ, если для каждой его вершины

$$\rho(x_i) \geqslant n/2. \qquad (16.1)$$

16.1.1. Связь между эйлеровыми и гамильтоновыми графами. На рис. 16.8 показан ряд классических графов. Причем на рис. 16.8, *а* — граф и эйлеров, и гамильтонов, на рис. 16.8, *б* — граф эйлеров, но не гамильтонов, на рис. 16.8, *в* — граф гамильтонов, но не эйлеров и на рис. 16.8, *г* — граф не эйлеров и не гамильтонов.

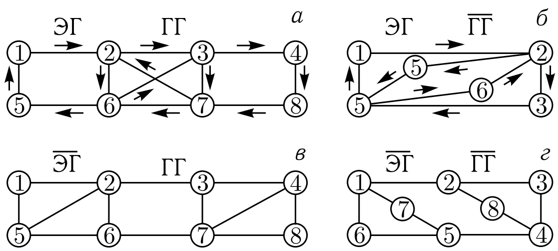


Рис. 16.8. Примеры эйлеровых и гамильтоновых графов

Теорема 16.3. *Пусть* G *имеет* $n \geqslant 3$ *вершин. Если для всякого* p $1 \leqslant p < (n-1)/2$ *число вершин со степенями, не превосходящими* p, *меньше чем* p, *и для нечетного* n *число вершин степени* $(n-1)/2$ *не превосходит этого числа, то* G *— гамильтонов граф* (*ГГ*).

Теорема 16.4. *Если* G *— эйлеров граф, то граф* $U(G) = G_s$, *двойственный графу* G, *является эйлеровым и гамильтоновым графом. Если* G *— гамильтонов граф, то* G_s *также гамильтонов граф.*

Можно привести контрпримеры обратным утверждениям. Так, например, граф G (рис. 16.9) является неэйлеровым, но га-

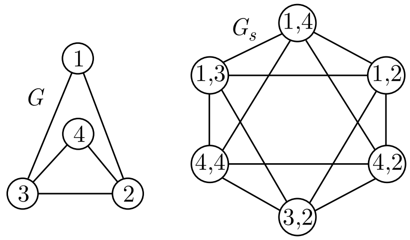


Рис. 16.9. Примеры графов

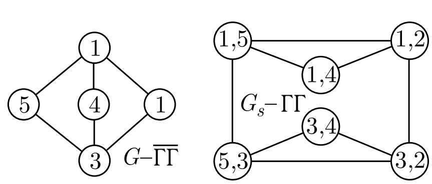


Рис. 16.10. Примеры графов

мильтоновым, а двойственный ему граф G_s является как эйлеровым, так и гамильтоновым.

Например, граф G — неэйлеров и негамильтонов граф, а граф G_s — неэйлеров и гамильтонов граф $\mathsf{G}_s = \mathsf{G}(U) = U(\mathsf{G})$ (рис. 16.10).

Если каждое ребро графа G подразбито путем введения дополнительной вершины, то граф G называется *графом подразбиений* и обозначается $S(\mathsf{G})$ (рис. 16.11).

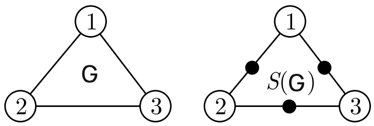


Рис. 16.11. Пример графа подразбиений

Теорема 16.5. *Для того чтобы граф $U(S(\mathsf{G}))$ был гамильтоновым графом, достаточно, чтобы G был гамильтоновым графом, и необходимо, чтобы $U(\mathsf{G})$ был гамильтоновым графом.*

Тотальным графом $T(\mathsf{G})$ называется граф, у которого множеством вершин является $(X \cup U)$ и две вершины смежны тогда и только тогда, когда они *соседние* в G. Вершины и ребра *соседние*, если они смежны и инцидентны. На рис. 16.12, *а* показан граф G, а на рис. 16.12, *б* — его тотальный граф. Видно, что $T(\mathsf{G})$ содержит в качестве порожденных подграфов G и $U(\mathsf{G})$.

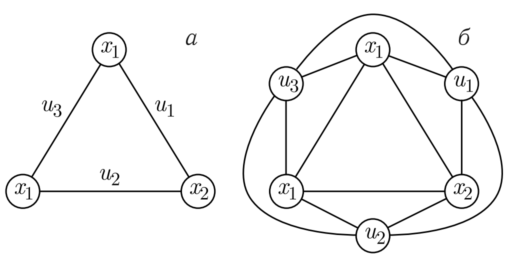


Рис. 16.12. Пример тотального графа

Теорема 16.6. *Если G — нетривиальный связный граф с n вершинами, не являющийся простой цепью, то граф $U^p(\mathsf{G})$ — гамильтонов граф, для всех $p \geqslant n - 3$. $U^n(\mathsf{G})$ — итерированный реберный граф графа G.*

Примеры, подтверждающие теорему 16.6, показаны на рис. 16.13. Как видно из рисунка, начальный граф G удовлетворяет требованиям теоремы, т. е. не является простой цепью. Число вершин n графа G равно 6. Тогда $p = 3$ и, следовательно, из всех двойственных графу G гамильтонов только граф $U^p(\mathsf{G}) = U^3(\mathsf{G})$, в то время как предшествующие ему графы $U(\mathsf{G})$ и $U^2(\mathsf{G})$ не являются гамильтоновыми.

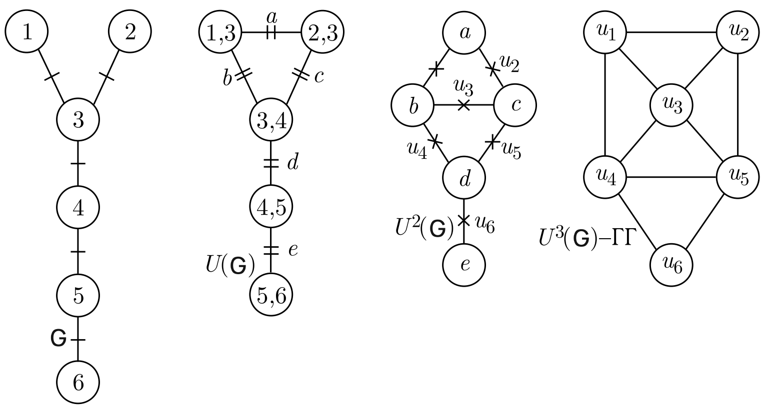


Рис. 16.13. Примеры нетривиальных и реберных графов

Процесс построения двойственных графов аналогичен описанному выше алгоритму. На каждом ребре в графе, для которого строится двойственный граф, ставим точку, которая преобразуется в вершину двойственного графа. После того как таким образом определили все вершины двойственного графа, мы соединяем их между собой ребрами. Причем ребром в двойственном графе соединяются вершины, соответствующие ребрам, смежным в исходном графе. Так, например, ребро u_{13} в графе G (рис. 16.13) смежно ребрам u_{23} и u_{34}. Следовательно, вершина $x_{1,3}$ в двойственном графе $U(\mathsf{G})$ будет соединена ребрами с вершинами $x_{2,3}$ и $x_{3,4}$. Аналогично каждое ребро $(a, b, \ldots, e)$ графа $U(\mathsf{G})$ дает одну вершину двойственного ему графа $U^2(\mathsf{G})$, а каждое ребро (u_1, u_2, u_6) графа $U^2(\mathsf{G})$ преобразуется в вершину графа $U^3(\mathsf{G})$, который является уже гамильтоновым. Таким образом, мы получили практическое подтверждение справедливости теоеремы 16.6.

Приведем известные утверждения, упрощающие алгоритмы определения ГЦ.

Утверждение 16.1. Если граф $\mathsf{G} = (X, U)$ имеет ГЦ, тогда для всех $x_i \in X$ $\rho(x_i) \geqslant 2$.

Утверждение 16.2. Если $x_i \in X$ и $\rho(x_i) = 2$, тогда 2 ребра, инцидентные вершине x_i, должны находиться в ГЦ, если он существует, для графа G.

Утверждение 16.3. Если $x_i \in X$ и $\rho(x_i) > 2$ и при построении ГЦ мы уже прошли через вершину x_i, то оставшиеся ребра, инцидентные x_i, исключаются из дальнейшего рассмотрения. На рис. 16.14 показан граф G из класса графов, не содержащих ГЦ.

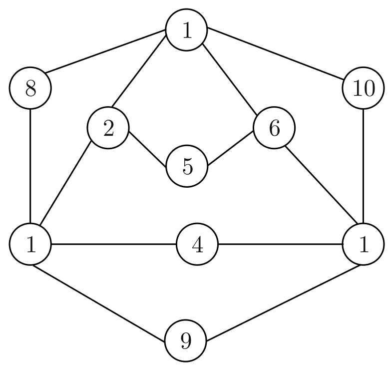


Рис. 16.14. Пример графа, не содержащего гамильтонов цикл

Введем понятие жордановой кривой и приведем один из вариантов теоремы Жордана. *Жордановой кривой* на плоскости называется непрерывная кривая, не имеющая самопересечений. Соответственно *замкнутой жордановой кривой* называется жорданова кривая, начало и конец которой совпадают.

Теорема 16.7 (Жордана). *Если L — замкнутая жорданова кривая, а x_i, x_j — две различные точки, расположенные на ней, то любая жорданова кривая, соединяющая x_i и x_j, должна лежать целиком внутри L или вне L (за исключением точек x_i, x_j), или пересекать L в некоторой точке, отличной от точек x_i, x_j.*

Если вершины графа $\mathsf{G} = (X, U)$ расположить так, чтобы ГЦ представлял собой самонепересекающуюся кривую, аналогичную замкнутой жордановой кривой, то ГЦ разделит плоскость на две области — внутреннюю и внешнюю. Естественно, внутренняя и внешняя области взаимно обратимы. Вообще говоря, согласно теореме 16.7 любой простой цикл, расположенный на плоскости, разбивает ее на внешнюю и внутреннюю области. Причем если имеются две вершины: x_i (расположенная во внутренней области) и x_j (расположенная во внешней области), то соединить их ребром без пересечения ребер цикла невозможно. Пример такого расположения вершин графа показан на рис. 16.15.

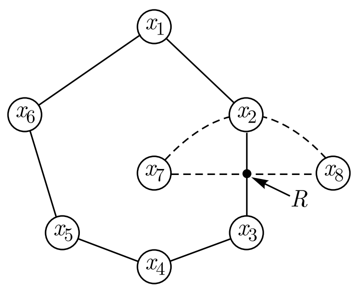


Рис. 16.15. Построение жордановой кривой

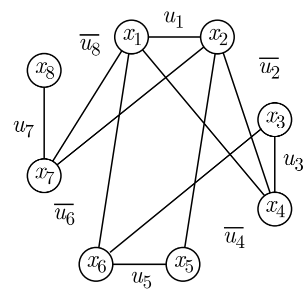


Рис. 16.16. Гамильтонов псевдоцикл

Здесь и далее под пересечением понимается соединение двух ребер графа в точке, не соответствующей никакой вершине. Если множество вершин графа $\mathsf{G} = (X, U)$ расположить на предполагаемой замкнутой самонепересекающейся кривой, то ребра, соединяющие соседние вершины графа, образуют гамильтонов псевдоцикл (ГПЦ), содержащий как ребра $u_i \in U$, так и фиктивные ребра $\overline{u_i} \notin U$, которые на самом деле отсутствуют. Пример гамильтонова псевдоцикла показан на рис. 16.16.

Здесь $C_{\text{ГПЦ}} = \{(x_1,x_2), (x_2,x_3), (x_3,x_4), (x_4,x_5), (x_5,x_6), (x_6,x_7), (x_7, x_8), (x_8, x_1)\}$, $u_1 = (x_1, x_2)$, $u_3 = (x_3, x_4)$, $u_5 = (x_5, x_6)$, $u_7 = (x_7, x_8)$ — существующие ребра, а фиктивными ребрами являются $\overline{u_2}$, $\overline{u_4}$, $\overline{u_6}$, $\overline{u_8}$. Очевидно, что полный граф всегда содержит ГЦ.

Примеры решения задач

Пример 16.1. Привести примеры графов:
а) содержащих одновременно ЭЦ и ГЦ,
б) содержащих ГЦ и не содержащих ЭЦ,
в) содержащих ЭЦ и не содержащих ГЦ,
г) не содержащих ни ЭЦ, ни ГЦ.

На рис. 16.17 приведены примеры графов, соответствующих заданным в данном примере условиям. Для эйлеровых графов на рисунке стрелками показан порядок обхода ребер графа.

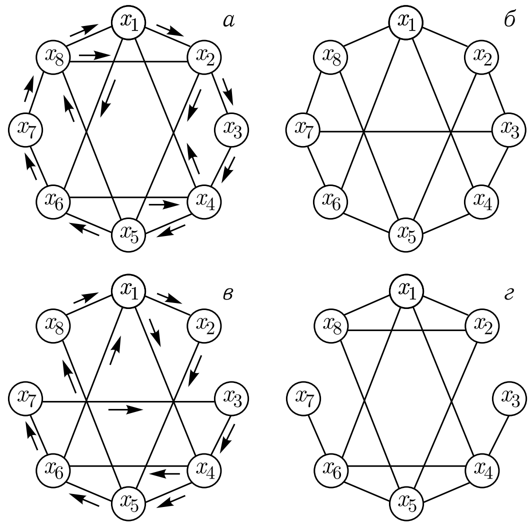


Рис. 16.17. Примеры эйлеровых и гамильтоновых графов

Пример 16.2. Для графа $\mathsf{G} = (X, U)$, заданного на рис. 16.17, *в*, построить ЭЦ, используя алгоритм Флери. (Нумерация в примере представляет собой двухуровневый список, первая цифра которого соответствует номеру итерации алгоритма, а вторая — шагу алгоритма на текущей итерации.)

Решение. Из рис. 16.17, *в* видно, что граф G — связный, поэтому переходим к алгоритму построения ЭЦ в графе.

1.1. Производим подсчет локальных степеней графа G. Все вершины имеют четные степени. Переходим к п. 1.2.

1.2. Выбираем вершину x_1.

1.3. Выбираем ребро $u(x_1{-}x_4)$. Здесь и далее в алгоритме примем следующее обозначение: $u(x_1{-}x_4) = u_{14}$. Заносим его в список L и вычеркиваем из матрицы смежности R.

1.4. $|L| \neq m$, где m — число ребер в графе G, переходим к п. 1.5.

1.5. Переходим к вершине x_4.

2.3. Выбираем ребро u_{45}. Заносим его в список L и вычеркиваем из матрицы смежности.

2.4. $|L| \neq m$, переходим к п. 2.5.

2.5. Переходим к вершине x_5.

3.3. Выбираем ребро u_{56} и заносим его в список L.

3.4. $|L| \neq m$, переходим к п. 3.5.

3.5. Переходим к вершине x_6.

4.3. Выбираем ребро u_{67} (ребро u_{61} не выбираем, поскольку в данный момент имеется возможность выбрать другое ребро).

4.4. $|L| \neq m$, переходим к п. 4.5.

4.5. Переходим к вершине x_7.

5.3. Выбираем ребро u_{73}. Заносим его в список L.

5.4. $|L| \neq m$, переходим к п. 5.5.

5.5. Выбираем вершину x_3.

6.3. Выбираем ребро u_{34} и заносим его в список L.

6.4. $|L| \neq m$, переходим к п. 6.5.

6.5. Выбираем вершину x_4.

7.3. Выбираем ребро u_{46}, заносим его в список L.

7.4. $|L| \neq m$, переходим к п. 7.5.

7.5. Выбираем вершину x_6.

8.3. Поскольку иных вариантов нет, выбираем ребро u_{61}.

8.4. $|L| \neq m$, переходим к п. 8.5.

8.5. Выбираем вершину x_1.

9.3. Выбираем ребро u_{12}, заносим его в список L.

9.4. $|L| \neq m$, переходим к п. 9.5.

9.5. Выбираем вершину x_2.

10.3. Выбираем ребро u_{25}, заносим его в список L.

10.4. $|L| \neq m$, переходим к п. 10.5.

10.5. Выбираем вершину x_5.

11.3. Выбираем ребро u_{58}, заносим его в список L.

11.4. $|L| \neq m$, переходим к п. 11.5.

11.5. Выбираем вершину x_8.

12.3. Выбираем ребро u_{81}, заносим его в список L.

12.4. $|L| = m$, переходим к п. 12.6.

12.6. Построен ЭЦ $(u_{14}-u_{45}-u_{56}-u_{67}-u_{73}-u_{34}-u_{46}-u_{61}-u_{12}-u_{25}-u_{58}-u_{81})$.

Поставленная задача выполнена. Работа алгоритма закончена.

Пример 16.3. Привести пример гамильтонова графа и показать на нем порядок обхода вершин.

Решение. На рис. 16.18 показан граф G, в котором имеется гамильтонов цикл. Ребра, входящие в гамильтонов цикл, выделены жирной линией. Последовательность обхода вершин графа в гамильтоновом цикле может быть следующей: ГЦ $=$ $= (x_1 - x_2 - x_{15} - x_{13} - x_{18} - x_{16} - x_{14} - x_6 - x_7 - x_{17} - x_{20} - x_{19} - x_{11} - x_{12} - x_3 - x_4 - x_{10} - x_9 - x_8 - x_5 - x_1)$.

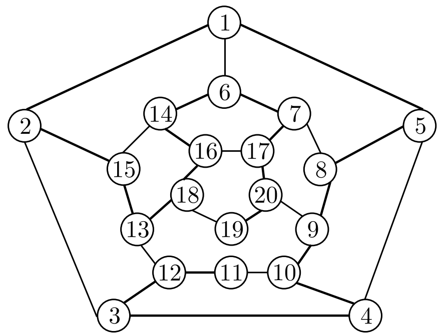


Рис. 16.18. Пример гамильтонова цикла в графе

Вопросы для самоконтроля

1. Какой цикл в графе называется эйлеровым? Что такое полуэйлеров граф?
2. Какие условия существования эйлерова цикла в графе?
3. Что такое гамильтонов цикл в графе?
4. Существуют ли условия, позволяющие однозначно определить наличие гамильтонова цикла в произвольном связном графе?
5. В чем заключается основная идея алгоритма Флери?
6. Что такое граф подразбиений?
7. Какой граф называется тотальным?
8. В каком случае ребра и вершины графа называются соседними?
9. Поясните понятие жордановой кривой.
10. Что такое замкнутая жорданова кривая?

Задания для самостоятельной работы

1. Для произвольного неориентированного графа определите, можно ли построить в этом графе эйлеров цикл.
2. Для произвольного связного неориентированного графа постройте с помощью алгоритма Флери эйлеров цикл.
3. Постройте структурную схему алгоритма Флери для построения эйлерова цикла.
4. Запишите псевдокод алгоритма Флери.
5. Приведите пример графа подразбиений.
6. Составьте описание алгоритма построения реберных графов.
7. Приведите псевдокод алгоритма построения реберных графов.

8. Приведите пример тотального графа.

9. Приведите пример связи эйлеровых и гамильтоновых графов, словесное описание алгебраического метода.

10. Приведите пример построения гамильтонова псевдоцикла.

Эйлеровы и гамильтоновы циклы в графе связаны с решением практических задач логистики, искусственного интеллекта, проектирования при нахождении циклов и цепей, проходящих по всем ребрам или через все вершины.

Связь между эйлеровыми и гамильтоновыми графами позволяет строить модели и производить преобразования для построения эффективных алгоритмов.

16.2. Алгоритмы построения гамильтонова цикла

16.2.1. Алгоритм Робертса–Флореса. Рассмотрим алгоритм Робертса и Флореса для построения гамильтонова цикла на графе. Алгоритм основан на переборе вершин, входящих в анализируемые цепи. Данная цепь непрерывно продлевается до момента, когда построен гамильтонов цикл или если эта цепь не приводит к его построению. Тогда рассматривается другая цепь и процесс продолжается сначала, пока не будет найден гамильтонов цикл или рассмотрены все цепи и окажется, что этот цикл не найден.

Исходной информацией является специальная матрица $\mathrm{M} = |m_{ij}|_{k\times n}$. Здесь k — это параметр, определяющий число строк матрицы. Величина m_{ij} определяет метку первой вершины, для которой есть ребро $u_{i,j}$. Число строк матрицы равно наибольшей локальной степени в анализируемом графе. Например, задан граф **G** (рис. 16.19) и его матрица:

$$\mathrm{M} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \left|\begin{array}{cccccc} b & a & a & c & b & a \\ c & c & b & e & c & b \\ f & e & d & - & d & c \\ - & f & e & - & - & \\ - & - & f & - & - & \end{array}\right| \end{array}.$$

Рассмотрим работу алгоритма на примере. Алгоритм начинается с рассмотрения элементов первого столбца матрицы М, если не задана иная начальная вершина гамильтонова цикла.

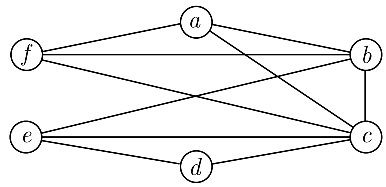


Рис. 16.19. Граф G

1.1. Выбираем вершину a. В списке вершин графа G, смежных с вершиной a (столбец a матрицы M), первой стоит вершина b.

1.2. Заносим в последовательность обхода вершин вершину b: ГЦ $= [a{-}b]$, переходим к столбцу b матрицы M.

1.3. Так как гамильтонов цикл еще не построен, продолжаем вычисления.

2.1. В списке вершин графа G, смежных с вершиной b, первой стоит вершина a, однако она уже задействована в цикле. Поэтому выбираем следующую вершину в столбце. Это вершина c.

2.2. Заносим в последовательность обхода вершин вершину c: ГЦ $= [a{-}b{-}c]$, переходим к столбцу c матрицы M.

2.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

3.1. В списке вершин графа G, смежных с вершиной c, вершины a и b мы пропускаем, так как они использованы в цикле. Выбираем вершину d.

3.2. Заносим в последовательность обхода вершин вершину d: ГЦ $= [a{-}b{-}c{-}d]$, переходим к столбцу d матрицы M.

3.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

4.1. В списке вершин графа G, смежных с вершиной d, пропускаем вершину c и выбираем вершину e.

4.2. Заносим в последовательность обхода вершин вершину e: ГЦ $= [a{-}b{-}c{-}d{-}e]$, переходим к столбцу e матрицы M.

4.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

5.1. В списке вершин графа G, смежных с вершиной e, все вершины уже задействованы в цикле.

5.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}c{-}d]$.

6.1. В списке вершин графа G, смежных с вершиной d, все вершины уже задействованы в цикле.

6.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}c]$.

7.1. В списке вершин графа G, смежных с вершиной c, пропускаем вершину d и выбираем вершину e.

7.2. Заносим в последовательность обхода вершин вершину e: ГЦ $= [a{-}b{-}c{-}e]$, переходим к столбцу e матрицы M.

7.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

8.1. В списке вершин графа G, смежных с вершиной e, пропускаем вершины b и c, после чего выбираем вершину d.

8.2. Заносим в последовательность обхода вершин вершину d: ГЦ $= [a{-}b{-}c{-}e{-}d]$, переходим к столбцу d матрицы M.

8.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

9.1. В списке вершин графа G, смежных с вершиной d, все вершины уже задействованы в цикле.

9.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}c{-}e]$.

10.1. В списке вершин графа G, смежных с вершиной e, все вершины уже задействованы в цикле.

10.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}c]$.

11.1. В списке вершин графа G, смежных с вершиной c, пропускаем вершину e и выбираем вершину f.

11.2. Заносим в последовательность обхода вершин вершину f: ГЦ $= [a{-}b{-}c{-}f]$, переходим к столбцу f матрицы M.

11.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

12.1. В списке вершин графа G, смежных с вершиной f, все вершины уже задействованы в цикле.

12.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}c]$.

13.1. В списке вершин графа G, смежных с вершиной c, все вершины уже задействованы в цикле.

13.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b]$.

14.1. В списке вершин графа G, смежных с вершиной b, пропускаем вершину c и выбираем вершину e.

14.2. Заносим в последовательность обхода вершин вершину e: ГЦ $= [a{-}b{-}e]$, переходим к столбцу c матрицы M.

14.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

15.1. В списке вершин графа G, смежных с вершиной e, пропускаем вершину b и выбираем вершину c.

15.2. Заносим в последовательность обхода вершин вершину c: ГЦ $= [a{-}b{-}e{-}c]$, переходим к столбцу c матрицы M.

15.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

16.1. В списке вершин графа G, смежных с вершиной c, выбираем вершину d.

16.2. Заносим в последовательность обхода вершин вершину d: ГЦ $= [a{-}b{-}e{-}c{-}d]$, переходим к столбцу d матрицы M.

16.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

17.1. В списке вершин графа G, смежных с вершиной d, все вершины уже задействованы в цикле.

17.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}e{-}c]$.

18.1. В списке вершин графа G, смежных с вершиной c, выбираем вершину f.

18.2. Заносим в последовательность обхода вершин вершину f: ГЦ $= [a{-}b{-}e{-}c{-}f]$, переходим к столбцу f матрицы M.

18.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

19.1. В списке вершин графа G, смежных с вершиной f, все вершины уже задействованы в цикле.

19.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}e{-}c]$.

20.1. В списке вершин графа G, смежных с вершиной c, все вершины уже задействованы в цикле.

20.2. Возвращаемся на шаг назад и пытаемся найти другой вариант продолжения обхода вершин графа: ГЦ $= [a{-}b{-}e]$.

21.1. В списке вершин графа G, смежных с вершиной e, пропускаем вершину c и выбираем вершину d.

21.2. Заносим в последовательность обхода вершин вершину d: ГЦ $= [a{-}b{-}e{-}d]$, переходим к столбцу c матрицы M.

21.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

22.1. В списке вершин графа G, смежных с вершиной d, выбираем вершину c.

22.2. Заносим в последовательность обхода вершин вершину c: ГЦ $= [a{-}b{-}e{-}d{-}c]$, переходим к столбцу c матрицы M.

22.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

23.1. В списке вершин графа G, смежных с вершиной c, выбираем вершину f.

23.2. Заносим в последовательность обхода вершин вершину f: ГЦ $= [a{-}b{-}e{-}d{-}c{-}f]$, переходим к столбцу c матрицы M.

23.3. Гамильтонов цикл еще не построен, продолжаем вычисления.

24.1. В списке вершин графа G, смежных с вершиной f, выбираем вершину a, поскольку это завершающее ребро гамильтонова цикла.

23.2. Заносим в последовательность обхода вершин вершину a: ГЦ $= [a{-}b{-}e{-}d{-}c{-}f{-}a]$.

23.3. Гамильтонов цикл построен, работа алгоритма завершена.

Главное преимущество алгоритма — простота, а его основным недостатком является сложность в случае необходимости перебора всех путей в процессе построения гамильтонова цикла. В случае же, когда гамильтонова цикла в графе не существует, алгоритм и вовсе сведется к полному перебору всех возможных вариантов построения пути.

Если в графе можно построить несколько гамильтоновых циклов, то алгоритм позволяет последовательно определить все гамильтоновы циклы. Кроме того, алгоритм позволяет определять и гамильтоновы цепи.

16.2.2. Алгебраический метод. Идея алгоритма состоит в построении всех простых цепей в графе путем последовательного перемножения матриц графа. В качестве исходных матриц для перемножения используются модифицированная матрица смежности графа B и матрица перемножений P_L. Модифицированная матрица смежности $\mathrm{B} = ||b_{ij}||$ представляет собой матрицу, в которой если x_i и x_j смежны, то в ячейку матрицы вместо единицы заносится x_j. На первом этапе матрица перемножений $\mathrm{P}_L = ||p_{L(i,j)}||$ идентична матрице B, а элемент матрицы $p_{L(i,j)}$ представляет собой сумму внутренних произведений всех простых цепей длиной l. Выражение вида $x_2 \times x_3 \times \ldots \times x_{k-1}$, получаемое в ячейке на пересечении строки x_1 и столбца x_k, представляет собой цепь $x_1, x_2, x_3, \ldots, x_{k-1}, x_k$.

Алгебраическое произведение матриц $\mathrm{P}'_{L+1} = \mathrm{B} \times \mathrm{P}_L$ представляет собой матрицу $\mathrm{P}'_{L+1} = p_{L+1(s,t)}$, где элемент $p_{L+1(s,t)} = \sum\limits_k b_{(s,k)} \times p_{L(k,t)}$.

Из полученной таким образом матрицы алгебраического произведения P'_{L+1} исключаются элементы, расположенные на главной диагонали. Кроме того, из ячеек, содержащих суммы внутренних произведений, исключаются слагаемые, содержащие начальный элемент простой цепи в качестве сомножителя. Полученная после подобного преобразования матрица обозначается нами как P_{L+1}, после чего процесс повторяется. По достижении $n-1$ шага все непустые элементы матрицы P_{n-1} представляют собой гамильтоновы цепи, которые при наличии ребра $u(x_k - x_1)$ образуют гамильтоновы циклы.

Примеры решения задач

Пример 16.4. Пусть задан граф $\mathsf{G} = (X, U)$, изображенный на рис. 16.20. Построить ГЦ для заданного графа, используя алгоритм Робертса–Флореса.

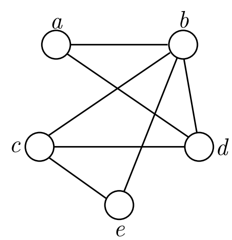


Рис. 16.20. Исходный граф

Решение. Построим матрицу M для заданного графа. Она будет иметь следующий вид:

$$\mathrm{M} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \left|\begin{array}{ccccc} b & a & b & a & a \\ d & c & d & b & b \\ e & d & e & c & c \\ & e & & & \end{array}\right| \end{array}.$$

Теперь воспользуемся описанным ранее алгоритмом.

1.1. Выбираем из матрицы M вершину a и заносим ее в список: $L = \{a\}$.

2.1. По матрице смежности выбираем из столбца с индексом a вершину, расположенную в первой ячейке. Это вершина b: $L = \{a, b\}$.

2.2. Проверяем условие $|L| = n$. Условие не выполняется, поэтому переходим к п. 3.1.

3.1. Выбираем столбец b, рассматриваем вершину из первой ячейки — это вершина a. Такая вершина уже есть в списке L, поэтому из второй ячейки столбца b берем вершину c. Переходим к следующему шагу.

3.2. Вершины c в списке L нет. Заносим ее в список: $L = \{a, b, c\}$.

3.3. Проверяем условие $|L| = n$. Условие не выполняется, поэтому переходим к п. 4.1.

4.1. В столбце c рассматриваем вершину из первой ячейки — это вершина b. Такая вершина уже есть в списке L, поэтому из второй ячейки столбца c берем вершину d. Переходим к следующему шагу.

4.2. Вершины d в списке L нет. Заносим ее в список: $L = \{a, b, c, d\}$.

4.3. Проверяем условие $|L| = n$. Условие не выполняется, поэтому переходим к п. 5.1.

5.1. В столбце d все вершины уже рассмотрены. Поэтому возвращаемся на шаг назад и пытаемся построить другой вариант построения пути. Вершина d вычеркивается из списка $L = \{a, b, c\}$. Переходим к п. 6.1.

6.1. В столбце c из третьей ячейки столбца берем вершину e. Переходим к следующему шагу.

6.2. Вершины e в списке L нет. Заносим ее в список: $L = \{a, b, c, e\}$.

6.3. Проверяем условие $|L| = n$. Условие не выполняется, поэтому переходим к п. 7.1.

7.1. В столбце e все вершины уже рассмотрены. Поэтому возвращаемся на шаг назад и пытаемся построить другой вариант построения пути. Вершина e вычеркивается из списка $L = \{a, b, c\}$. Переходим к п. 8.1.

8.1. В столбце c все вершины уже рассмотрены. Поэтому возвращаемся на шаг назад и пытаемся построить другой вариант построения пути. Вершина c вычеркивается из списка $L = \{a, b\}$. Переходим к п. 9.1.

9.1. В столбце b из третьей ячейки столбца берем вершину d. Переходим к следующему шагу.

9.2. Вершины d в списке L нет. Заносим ее в список: $L = \{a, b, d\}$.

9.3. Проверяем условие $|L| = n$. Условие не выполняется, поэтому переходим к п. 10.1.

10.1. В столбце d из третьей ячейки столбца берем вершину c. Переходим к следующему шагу.

10.2. Вершины c в списке L нет. Заносим ее в список: $L = \{a, b, d, c\}$.

10.3. Проверяем условие $|L| = n$. Условие не выполняется, поэтому переходим к п. 11.1.

11.1. В столбце c из третьей ячейки столбца берем вершину e. Переходим к следующему шагу.

11.2. Вершины e в списке L нет. Заносим ее в список: $L = \{a, b, d, c, e\}$.

11.3. Проверяем условие $|L| = n$. Условие выполняется. Следовательно, в графе G построена гамильтонова цепь и нам необходимо проверить, есть ли в исходном графе замыкающее ребро, чтобы завершить гамильтонов цикл.

11.4. В столбце e имеется вершина a, следовательно, поставленная задача выполнена, в графе G построен гамильтонов цикл:

$$\{a–b–d–c–e–a\}.$$

11.5. Конец работы алгоритма.

Пример 16.5. Пусть задан граф $\mathsf{G} = (X, U)$, изображенный на рис. 16.21. Построить ГЦ для данного графа алгебраическим методом.

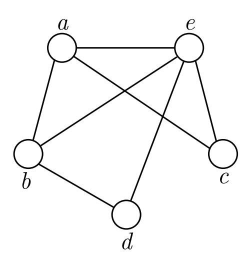


Рис. 16.21. Исходный граф

Решение.

1. Запишем матрицу смежности B и матрицу перемножений P_1 для заданного графа. Матрица P_1 на первом этапе совпадает с матрицей

$$\mathrm{B} = \begin{array}{c|ccccc|} a & 0 & b & c & 0 & e \\ b & a & 0 & 0 & d & e \\ c & a & 0 & 0 & 0 & e \\ d & 0 & b & 0 & 0 & e \\ e & a & b & c & d & 0 \end{array}.$$

2. После перемножения матриц B и P_1 получим матрицу:

$$
P_2 = \begin{array}{c|c|c|c|c|c|}
 & a & b & c & d & e \\ \hline
a & b+c+e & e & e & b+e & b+c \\ \hline
b & e & a+d+e & a+e & e & a+d \\ \hline
c & e & a+e & a+e & e & a \\ \hline
d & b+e & e & e & b+e & b \\ \hline
e & b+c & a+d & a & b & a+b+c+d \\ \hline
\end{array}.
$$

После исключения из матрицы P_2' элементов, находящихся на главной диагонали, получим матрицу:

$$
P_2' = \begin{array}{c|c|c|c|c|c|}
 & a & b & c & d & e \\ \hline
a & 0 & e & e & b+e & b+c \\ \hline
b & e & 0 & a+e & e & a+d \\ \hline
c & e & a+e & 0 & e & a \\ \hline
d & b+e & e & e & 0 & b \\ \hline
e & b+c & a+d & a & b & 0 \\ \hline
\end{array}.
$$

3. После перемножения матриц B и P_2 получим матрицу:

$$
P_3' = \begin{array}{c|c|c|c|c|c|}
 & a & b & c & d & e \\ \hline
a & be+ce+eb+ec & \mathbf{ca}+ce+\mathbf{ea}+ed & \mathbf{ba}+be+\mathbf{ea} & be+ce+eb & \mathbf{ba}+bd+\mathbf{ca} \\ \hline
b & de+\mathbf{db}+\mathbf{eb}+ec & ae+de+ea+ed & ae+de+ea & \mathbf{ab}+ae+\mathbf{eb} & \mathbf{ab}+ac+\mathbf{db} \\ \hline
c & eb+\mathbf{ec} & ae+ea+ed & ae+ea & ab+ae & ab+\mathbf{ac} \\ \hline
d & be+eb+ec & ea+\mathbf{ed} & ba+be+ea & be+eb & ba+\mathbf{bd} \\ \hline
e & \mathbf{be}+\mathbf{ce}+db+\mathbf{de} & \mathbf{ae}+ca+\mathbf{ce}+\mathbf{de} & \mathbf{ae}+ba+\mathbf{be}+\mathbf{de} & ab+\mathbf{ae}+\mathbf{be}+\mathbf{ce} & ab+ac+ba+bd+ca+db \\ \hline
\end{array}.
$$

В полученной матрице сначала обнуляем ячейки главной диагонали, а затем из всех ячеек матрицы исключаем элементы, в которые в качестве сомножителя входят концевые вершины цикла. Например, в строке a исключаем все элементы, включающие в качестве сомножителя вершину a. Для наглядности такие элементы выделены в матрице жирным шрифтом. После исключения из матрицы P_2' всех вышеуказанных элементов получим матрицу:

$$\mathrm{P}_3 = \begin{array}{c|c|c|c|c|c|}
 & a & b & c & d & e \\ \hline
a & 0 & ce+ed & be & be+ce+eb & bd \\ \hline
b & de+ec & 0 & ae+de+ea & ae & ac \\ \hline
c & eb & ae+ea+ed & 0 & ab+ae & ab \\ \hline
d & be+eb+ec & ea & ba+be+ea & 0 & ba \\ \hline
e & db & ca & ba & ab & 0 \\ \hline
\end{array}\,.$$

4. После перемножения матриц B и P_3 получим матрицу:

$$\mathrm{P}_4' = \begin{array}{c|c|c|c|c|c|}
 & a & b & c & d & e \\ \hline
a & bde+bec+ceb+edb & cae+cea+ced+eca & bde+bae+bea+eba & bae+cab+cae+ceb+eab & bac+cab \\ \hline
b & dbe+deb+dec+edb & ace+aed+dea+eca & abe+dba+dbe+dea+eba & abe+ace+aeb+eab & abd+dba \\ \hline
c & edb & ace+eca+aed & abe+eba & abe+ace+aeb+eab & abd \\ \hline
d & bde+edb+bec & eca & bae+bde+bea+eba & bae+eab & bac \\ \hline
e & bde+bec+ceb+dbe+deb+dec & ace+aed+cae+cea+ced & abe+bae+bde+bea+dba+dbe+dea & abe+ace+aeb+bae+cab+cae+ceb & abd+bac+cab+dba \\ \hline
\end{array}$$

После аналогичного преобразования матрицы P_4' можем записать матрицу:

$$\mathrm{P}_4 = \begin{array}{c|c|c|c|c|c|}
 & a & b & c & d & e \\ \hline
a & 0 & ced & bde & ceb & 0 \\ \hline
b & dec & 0 & dea & ace & 0 \\ \hline
c & edb & aed & 0 & abe+aeb+eab & abd \\ \hline
d & bec & eca & bae+bea+eba & 0 & bac \\ \hline
e & 0 & 0 & dba & cab & 0 \\ \hline
\end{array}\,.$$

Все ненулевые ячейки матрицы P_4 представляют собой гамильтоновы цепи. Концами этих цепей являются вершины, на пересечении которых располагается рассматриваемая ячейка. Например, ячейка на пересечении строки a и столбца d представляет собой гамильтонову цепь a—c—e—b—d. Те из построенных цепей, для которых существует замыкающее ребро, образуют гамильтонов цикл. Анализ полученной матрицы P_4 позволяет сделать вывод о наличии в исходном графе **G** следующего ГЦ:

$$a—c—e—d—b.$$

Все остальные цепи либо не имеют замыкающего ребра, либо являются вариантами обнаруженного цикла. Например, c—e—d—b—a или d—b—a—c—e.

Очевидно, что различные эвристические алгоритмы определения гамильтонова цикла, допустим, алгоритм Робертса–Флореса, не могут дать точного решения задачи, но обеспечивают нахождение результата за полиномиальное время. В худшем случае, например, когда гамильтонова цикла в графе не существует, они могут свестись к полному перебору всех возможных вариантов. При анализе всех цепей можно получить точный результат, но для задач практически большой размерности такой вариант неприемлем. Алгоритм перемножения матриц позволяет всегда найти гамильтонов цикл в графе, если он существует, однако этот алгоритм сложно реализовать с вычислительной точки зрения.

Вопросы для самоконтроля

1. В чем состоит идея алгоритма Робертса–Флореса?
2. Опишите последовательность построения гамильтонова цикла в графе с помощью алгоритма Робертса–Флореса.
3. Какова сложность алгоритма Робертса–Флореса?
4. В чем состоит идея алгебраического метода?
5. Опишите последовательность построения гамильтонова цикла в графе с помощью алгебраического метода.
6. Какова сложность алгебраического метода при построении одного гамильтонова цикла?
7. Какова сложность алгебраического метода при построении всех гамильтоновых циклов в графе?
8. В чем заключается отличие стратегии работы алгоритмов Робертса–Флореса и алгебраического?
9. Для решения каких задач следует применять алгоритм Робертса–Флореса?
10. В каких случаях применение алгебраического метода эффективно?

Задания для самостоятельной работы

1. Для произвольного графа постройте, используя алгоритм Робертса–Флореса, гамильтонов цикл.

2. Запишите структурную схему алгоритма Робертса–Флореса для построения гамильтонова цикла.

3. Приведите словесное описание метода Робертса–Флореса.

4. Для произвольного графа постройте гамильтонов цикл, используя алгебраический метод.

5. Запишите структурную схему алгебраического метода для построения гамильтонова цикла.

6. Приведите словесное описание алгебраического метода.

7. Задайте граф на 7 вершин и постройте все гамильтоновы циклы методом Робертса–Флореса.

8. Задайте граф на 7 вершин и постройте все гамильтоновы циклы алгебраическим методом.

9. Запишите псевдокод алгоритма Робертса–Флореса.

10. Запишите псевдокод алгебраического метода.

Эвристические алгоритмы определения гамильтонова цикла являются быстродействующими, но не точными. При анализе всех цепей можно получить точный результат. Алгоритм перемножения матриц позволяет всегда найти гамильтонов цикл в графе, если он существует.

16.3. Задача о коммивояжере и алгоритмы ее решения

16.3.1. Алгоритм Хелда и Карпа. С задачей нахождения гамильтонова цикла в графе тесно связана задача о коммивояжере. *Задача о коммивояжере* формулируется следующим образом: имеется n городов, расстояния между которыми известны. Коммивояжеру необходимо посетить каждый город по одному разу и вернуться в исходный пункт, пройдя при этом минимально возможное расстояние.

Из определения задачи о коммивояжере следует, что ее можно свести к задаче нахождения гамильтонова цикла во взвешенном полном графе с минимальным суммарным весом ребер, составляющих гамильтонов цикл. Известно несколько алгоритмов решения задачи о коммивояжере.

Идея алгоритма Хелда–Карпа состоит в следующем. Сначала определяются все кратчайшие пути, соединяющие вершину x_1 с остальными вершинами. Затем определяются кратчайшие пути от вершины x_1 до всех остальных вершин графа, через одну промежуточную вершину, после этого через две и т. д., до $n-1$ шага. Затем к полученным результатам добавляется длина пути для возвращения в вершину x_1. Из всех полученных вариантов пути выбирается тот, которому соответствует наименьшая сумма весов ребер. Основным недостатком данного метода является необходимость выполнить полный перебор всех возможных вариантов, что для задач большой размерности практически невозможно.

16.3.2. Геометрический метод решения. Опишем геометрический метод решения задачи о коммивояжере. Решение задачи геометрическим методом возможно при условии, что в графе любые две вершины смежны и выполняется правило треугольника.

Сформулируем «*правило треугольника*»: длина оптимального маршрута коммивояжера не может быть меньше удвоенной величины максимального веса ребра, выбранного среди множества всех ребер графа:

$$L1 + L2 > 2r_{i,j}^{\max}. \tag{16.2}$$

Приведем словесное описание алгоритма. Примем по умолчанию, что $L1$ — расстояние между вершинами x_i и x_k, а $L2$ — расстояние между вершинами x_k и x_j.

1°. Определим максимальный элемент матрицы R и вершины x_k, x_h, расстояние между которыми max.

2°. Строятся треугольники с основанием $k{-}h$ с вершинами $i \in I = \{1, 2, \ldots, n\}$, $i \neq k$, $i \neq h$. Таких треугольников будет $n-2$. Вычисляем их периметры по формуле:

$$\Pi_{kih} = r(k,i) + r(i,h) + r(k,h). \tag{16.3}$$

3°. Находим $\Pi_{\max} = \Pi_{klh}$. Фиксируем вершины x_k, x_l, x_h и выделяем звенья $L1$ — $(k{-}h)$, $L2$ — $(k{-}l), (l{-}h)$. Далее оставшиеся $n-3$ вершины включаются либо в $L1$, либо в $L2$.

4°. Для каждого фиксированного $i \neq l$, $i \neq k$, $i \neq h$ вычисляем

$$\Delta R(k,h) = r(k,i) + r(I,h) - r(k,h),$$

$$\Delta R(k,l) = r(k,i) + r(I,l) - r(k,l),$$

$$\Delta R(l,h) = r(k,i) + r(I,h) - r(l,h).$$

Наименьшая ΔR определит принадлежность x_i соответствующему отрезку, и x_i принадлежит $k{-}l$.

5°. Отрезки $a{-}b$ заменяем согласно 4° отрезками $a{-}c$ и $c{-}b$. Если все вершины просмотрены, то к 7°.

6°. Если две вершины x_a, x_b отнесены к одному отрезку $k{-}h \to k{-}a$, $a{-}h$ и делают $k{-}b{-}a$ или $a{-}b{-}h$.

7°. Определяем длину маршрута.

Например, пусть имеется 6 городов, расстояния между которыми известны. Матрица смежности графа запишется следующим образом:

$$
\mathrm{R} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\left|\begin{array}{cccccc}
\infty & 30 & 65 & 40 & 62 & 28\\
30 & \infty & 40 & 20 & 40 & 60\\
65 & 40 & \infty & 30 & 55 & 60\\
40 & 20 & 30 & \infty & 30 & 32\\
62 & 40 & 55 & 30 & \infty & 36\\
28 & 60 & 60 & 32 & 36 & \infty
\end{array}\right| \end{array}.
$$

Согласно 1° определим пару вершин, расстояние между которыми максимально (наибольший элемент матрицы R).

$$r_{i,j}^{\max} = \max\left(r(i,j)\right) = r_{1,3} = 65.$$

Фиксируем вершины x_1, x_3 и для отрезка 1–3 строим треугольники с вершинами в точках 2, 4, 5 и 6.

2°. Вычисляем периметры построенных треугольников (рис. 16.22) по формуле (16.3). При этом слагаемое $r_{1,3}$ можно пропустить, поскольку оно одинаково для всех выражений.

$$\Pi_{123} = 30 + 40 = 70,$$

$$\Pi_{143} = 40 + 30 = 70,$$

$$\Pi_{153} = 62 + 55 = 117,$$

$$\Pi_{163} = 28 + 60 = 88.$$

3°. Выбираем треугольник с максимальной величиной периметра Π_{153} и фиксируем вершины x_1, x_5, x_3. Выделяем два звена: $L1$ — ($x_1{-}x_3$) и $L2$ — ($x_1{-}x_5$ и $x_5{-}x_3$) (см. рис. 16.22).

4°. Оставшиеся незадействованными вершины x_2, x_4, x_6 отнесем по принадлежности к отрезкам 1–3, 1–5, 5–3. Для этого с учетом правила треугольника, гласящего, что сумма длин двух сторон треугольника всегда больше длины третьей стороны,

$$r(i,j)+r(j,k)>r(i,k),\ r(i,k)+r(k,j)>r(i,j),\ r(k,i)+r(i,j)>r(k,j),$$

используем следующую формулу:

$$\Delta R(l,h) = r(l,i) + r(i,h) - r(l,h). \tag{16.4}$$

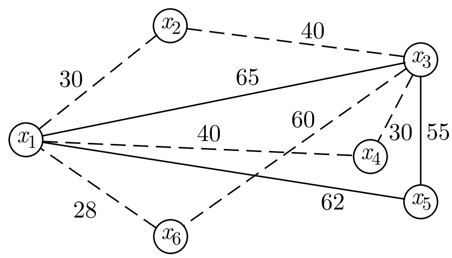


Рис. 16.22. Построение пути коммивояжера

Начнем с вершины x_2. При этом получим, что $k = 1$, $h = 3$, $l = 5$, $i = 2$. Тогда имеем:

$$\Delta R_{1,3} = r_{1,2} + r_{2,3} - r_{1,3} = 30 + 40 - 65 = 5,$$

$$\Delta R_{1,5} = r_{1,2} + r_{2,5} - r_{1,5} = 30 + 40 - 28 = 42,$$

$$\Delta R_{3,5} = r_{3,2} + r_{2,5} - r_{3,5} = 40 + 40 - 55 = 25.$$

Так как из всех полученных значений $\Delta R_{1,3} = 5$ является наименьшим, то вершину x_2 отнесем к отрезку 1–3.

Аналогичные вычисления выполняем для вершин x_4 и x_6. Так, для вершины x_4: $k = 1$, $h = 3$, $l = 5$, $i = 4$. Тогда

$$\Delta R_{1,3} = r_{1,4} + r_{4,3} - r_{1,3} = 40 + 30 - 65 = 5,$$

$$\Delta R_{1,5} = r_{1,4} + r_{4,5} - r_{1,5} = 40 + 30 - 28 = 42,$$

$$\Delta R_{3,5} = r_{3,4} + r_{4,5} - r_{3,5} = 30 + 30 - 55 = 5.$$

Вершину x_4 можно отнести как к отрезку 1–3, так и к отрезку 3–5. Отнесем ее к отрезку 3–5, так как к отрезку 1–3 мы уже отнесли вершину x_2.

Для вершины x_6: $k = 1$, $h = 3$, $l = 5$, $i = 6$. Тогда

$$\Delta R_{1,3} = r_{1,6} + r_{6,3} - r_{1,3} = 28 + 60 - 65 = 23,$$

$$\Delta R_{1,5} = r_{1,6} + r_{6,5} - r_{1,5} = 28 + 36 - 62 = 2,$$

$$\Delta R_{3,5} = r_{3,6} + r_{6,5} - r_{3,5} = 60 + 36 - 55 = 41.$$

Следовательно, вершину x_6 отнесем к отрезку 1–5. Теперь производим включение каждой вершины в маршрут.

5°. Отрезок 1–3 заменяем двумя отрезками 1–2 и 2–3. Аналогично отрезок 3–5 заменяем отрезками 3–4 и 4–5, а отрезок 1–5 — отрезками 1–6 и 6–5.

Если все вершины включены в маршрут, то алгоритм закончен, надо вычислить только длину маршрута и перейти к п. 7°. В противном случае переход к п. 6°.

Если в процессе работы окажется, что к одному отрезку одновременно отнесено несколько вершин, то в этом случае оставляем вершину, для которой значение $\Delta R(k, h)$ минимально.

6°. Все вершины отнесены к отрезкам $k{-}l$, $k{-}h$ и $l{-}h$. Пусть, например, для отрезка $k{-}h$ зафиксированы две вершины: x_i и x_j. Тогда надо отнести вершину x_j к одному из отрезков $k{-}i$ или $i{-}h$.

Для этого вычисляем значения

$$\Delta R_{ki} = r(k, j) + r(j, i) - r(k, i),$$

$$\Delta R_{ih} = r(i, j) + r(j, h) - r(i, h).$$

Минимальная из этих двух величин определяет принадлежность вершины x_j к отрезку $k{-}i$ или отрезку $i{-}h$. Аналогично определяем принадлежность всех остальных вершин для других отрезков и переходим к п. 5°.

7°. Построен следующий маршрут коммивояжера:

$$x_1{-}x_2{-}x_3{-}x_4{-}x_5{-}x_6{-}x_1.$$

Его длина равна 194.

На рис. 16.23 показан ГЦ с весами на ребрах, когда сумма весов пройденных ребер минимальна.

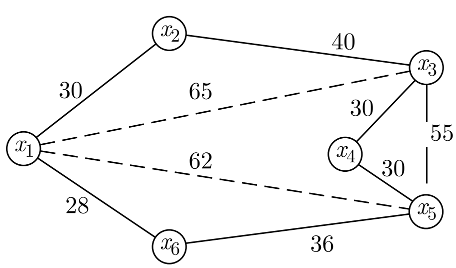


Рис. 16.23. Решение задачи коммивояжера

Алгоритм справедлив только для графа, у которого выполняется правило треугольника.

Примеры решения задач

Пример 16.6. Пусть задан граф G, изображенный на рис. 16.24. Решить задачу коммивояжера для заданного графа, используя алгоритм Хелда–Карпа.

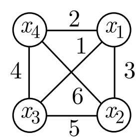


Рис. 16.24. Исходный граф

Решение. Построим матрицу расстояний заданного графа:

$$R = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 1 & 2 \\ 2 & 3 & 0 & 5 & 6 \\ 3 & 1 & 5 & 0 & 4 \\ 4 & 2 & 6 & 4 & 0 \end{array}.$$

1. В соответствии с заданным алгоритмом определяем прямые пути из вершины x_1 в вершину x_j, где $j = (2, 3, 4) : d_{12} = 3$; $d_{13} = 1$; $d_{14} = 4$.

2. Теперь определяем пути из вершины x_1 в вершину x_j через одну промежуточную вершину:

$$d_{12}^3 = d_{13} + d_{32} = 1 + 5 = 6; \quad d_{14}^3 = d_{13} + d_{34} = 1 + 4 = 5;$$
$$d_{13}^2 = d_{12} + d_{23} = 3 + 5 = 8; \quad d_{14}^2 = d_{12} + d_{24} = 3 + 6 = 9;$$
$$d_{12}^4 = d_{14} + d_{42} = 2 + 6 = 8; \quad d_{13}^4 = d_{14} + d_{43} = 2 + 4 = 6.$$

3. Определяем пути из вершины x_1 в вершину x_j через две промежуточные вершины:

$$d_{12}^{3,4} = d_{14}^3 + d_{42} = d_{13} + d_{34} + d_{42} = 1 + 4 + 6 = 11;$$
$$d_{13}^{2,4} = d_{14}^2 + d_{43} = d_{12} + d_{24} + d_{43} = 3 + 6 + 4 = 13;$$
$$d_{14}^{2,3} = d_{13}^2 + d_{34} = d_{12} + d_{23} + d_{34} = 3 + 5 + 4 = 12;$$
$$d_{12}^{4,3} = d_{13}^4 + d_{32} = d_{14} + d_{43} + d_{32} = 2 + 4 + 5 = 11;$$
$$d_{13}^{4,2} = d_{12}^4 + d_{23} = d_{14} + d_{42} + d_{23} = 2 + 6 + 5 = 13;$$
$$d_{14}^{3,2} = d_{12}^3 + d_{24} = d_{13} + d_{32} + d_{24} = 1 + 5 + 6 = 12.$$

4. Теперь к полученной длине пути добавляем длину возвращения в исходный пункт:

$$d_{12} = d_{12}^{3,4}\ (\text{или}\ \ d_{12}^{43}) + d_{21} = 11 + 3 = 14;$$

$$d_{13} = d_{13}^{24}\ (\text{или}\ \ d_{13}^{42}) + d_{31} = 13 + 1 = 14;$$

$$d_{14} = d_{14}^{23}\ (\text{или}\ \ d_{14}^{32}) + d_{41} = 12 + 2 = 14.$$

Таким образом, в результате работы алгоритма выяснилось, что длина пути коммивояжера при любом варианте обхода вершин графа одинакова и равна 14.

Пример 16.7. Пусть задан граф $\mathsf{G} = (X, U)$. Решить задачу о коммивояжере для заданного графа геометрическим методом.

Решение. Зададим граф $\mathsf{G} = (X, U)$ матрицей расстояний, а поскольку граф неориентированный, то для его однозначного задания вполне достаточно треугольной матрицы

$$\mathrm{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left|\begin{array}{cccccc} \infty & 40 & 29 & 66 & 52 & 36 \\ & \infty & 41 & 63 & 37 & 50 \\ & & \infty & 38 & 48 & 42 \\ & & & \infty & 28 & 56 \\ & & & & \infty & 32 \\ & & & & & \infty \end{array}\right| \end{array}.$$

1. Определяем по матрице R элемент с максимальным весом. Это элемент $r_{14} = 66$. Вершины x_1 и x_4 фиксируем.

2. Строим треугольники с основанием $x_1 - x_3$ (рис. 16.25, *а*).

Вычисляем периметры получившихся треугольников:

$$\Pi_{124} = 40 + 63 = 103;$$

$$\Pi_{134} = 29 + 38 = 67;$$

$$\Pi_{154} = 52 + 28 = 80;$$

$$\Pi_{164} = 36 + 56 = 92.$$

3. Выбираем треугольник Π_{124}, имеющий максимальную длину. Фиксируем вершины x_1, x_2, x_4. Выделяем два звена $L1 = (x_1 - x_4)$ и $L2 = (x_1 - x_2$ и $x_2 - x_4)$ (рис. 16.25, *б*).

4. Подсчитываем значения ΔR для вершин x_3, x_5, x_6.

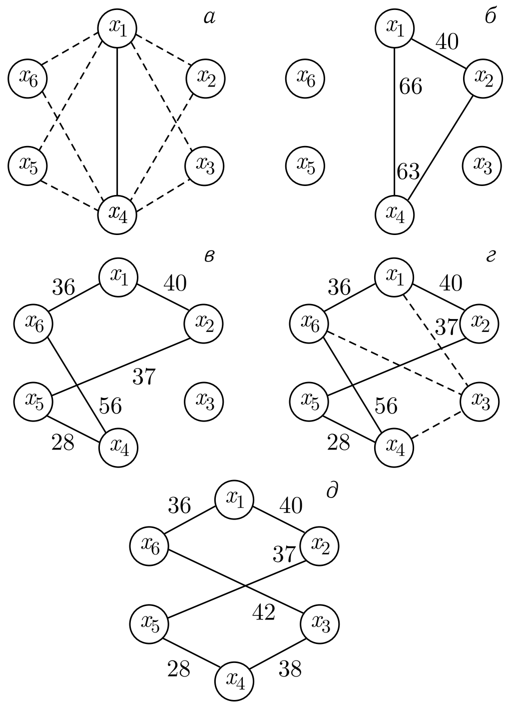


Рис. 16.25. Построение пути коммивояжера: *а* — шаг 1; *б* — шаг 2; *в* — шаг 3; *г* — шаг 4; *д* — результат

а) Для вершины x_3:

$$\Delta R_{12} = r_{13} + r_{32} - r_{12} = 29 + 41 - 40 = 30;$$
$$\underline{\Delta R_{14} = r_{13} + r_{34} - r_{14} = 29 + 38 - 66 = 1};$$
$$\Delta R_{24} = r_{23} + r_{34} - r_{24} = 41 + 38 - 63 = 16.$$

б) Для вершины x_5:

$$\Delta R_{12} = r_{15} + r_{52} - r_{12} = 52 + 37 - 40 = 49;$$
$$\Delta R_{14} = r_{15} + r_{54} - r_{14} = 52 + 28 - 66 = 24;$$
$$\underline{\Delta R_{24} = r_{25} + r_{54} - r_{24} = 37 + 28 - 63 = 2}.$$

в) Для вершины x_6:

$$\Delta R_{12} = r_{16} + r_{62} - r_{12} = 36 + 50 - 40 = 46;$$
$$\underline{\Delta R_{14} = r_{16} + r_{64} - r_{14} = 36 + 56 - 66 = 26};$$
$$\Delta R_{24} = r_{26} + r_{64} - r_{24} = 50 + 56 - 63 = 43.$$

5. В результате проведенных подсчетов получилось, что вершина x_5 должна быть отнесена к отрезку $x_2{-}x_4$, вершины x_3 и x_6 относятся к отрезку $x_1{-}x_4$. Следовательно, заменяем отрезок $x_2{-}x_4$ двумя отрезками $x_2{-}x_5$ и $x_5{-}x_4$. Сравниваем полученные значения ΔR_{14} для вершин x_3 $(\Delta R_{14} = 1)$ и x_6 $(\Delta R_{14} = 26)$. Вершину x_3, соответствующую меньшему из двух значению ΔR_{14}, оставляем, а через вершину x_6 строим обход $x_1{-}x_6$ и $x_6{-}x_4$ (рис. 16.25, *в*). После чего переходим к п. 6.

6. Для вершины x_3 подсчитываем значения ΔR_{16} и ΔR_{64}, для того чтобы определить, к какому из этих двух отрезков она должна быть отнесена (рис. 16.25, *г*).

Для вершины x_3:

$$\Delta R_{16} = r_{13} + r_{36} - r_{16} = 29 + 42 - 36 = 35;$$
$$\underline{\Delta R_{64} = r_{63} + r_{34} - r_{64} = 42 + 38 - 56 = 24}.$$

Вершину x_3 отнесем к отрезку $x_4{-}x_6$. После этого шага все вершины графа рассмотрены.

7. В результате мы получили решение задачи о коммивояжере (рис. 16.25, *д*).

Длина построенного маршрута коммивояжера равна:

$$r_{12} + r_{25} + r_{54} + r_{43} + r_{36} + r_{61} = 40 + 37 + 28 + 38 + 42 + 36 = 221.$$

Вопросы для самоконтроля

1. В каких задачах науки и практики может быть использована задача коммивояжера?
2. Дайте определение задачи коммивояжера.
3. В чем состоит идея алгоритма Хелда–Карпа?
4. На чем основан геометрический метод решения задачи коммивояжера?
5. Какие методы решения задачи коммивояжера вы знаете?
6. Какова сложность алгоритма Хелда–Карпа?
7. В чем состоит «правило треугольника»? Сформулируйте его.
8. Каким образом строятся треугольники в графе при решении задачи коммивояжера?
9. В чем заключаются недостатки и преимущества алгоритма Хелда–Карпа?
10. В чем заключаются недостатки и преимущества геометрического метода решения задачи коммивояжера?

Задания для самостоятельной работы

1. Для произвольного связного неориентированного графа решите задачу коммивояжера на основе алгоритма Хелда–Карпа.
2. Для произвольного связного неориентированного графа решите задачу коммивояжера геометрическим методом.
3. Запишите структурную схему алгоритма Хелда–Карпа.
4. Постройте структурную схему геометрического метода.
5. Приведите словесное описание алгоритма Хелда–Карпа.
6. Приведите словесное описание геометрического метода.
7. Запишите псевдокод алгоритма Хелда–Карпа.
8. Запишите псевдокод геометрического метода.
9. Вычислите ВСА алгоритма Хелда–Карпа.
10. Вычислите ВСА геометрического метода.

Задача о коммивояжере является классической в теории графов. Модель задачи коммивояжера можно эффективно использовать для решения большого числа оптимизационных задач.

16.4. Расстояния на графах

Рассмотрим метрику графов. *Метрика* графа основана на понятии расстояния. *Расстоянием* $d(x_i, x_j)$ между вершинами x_i, $x_j \in X$ графа $\mathsf{G} = (X, U)$ называется длина кратчайшей цепи, соединяющей эти вершины. Под *длиной цепи* понимается число входящих в нее ребер. Функция $d(x_i, x_j)$, определенная на множестве ребер графа G, называется метрикой графа.

Функцию расстояний для графа G удобно задавать *матрицей расстояний* $\mathrm{D} = ||d_{i,j}||_{n\times n}$ или ее списком. Элемент матрицы определяется следующим образом:

$$d_{i,j} = \begin{cases} 0, & x_i = x_j, \\ d_{i,j}, & x_i \neq x_j. \end{cases} \qquad (16.5)$$

Например, для графа G (рис. 16.26)

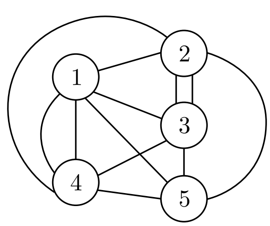


Рис. 16.26. Граф G

матрица расстояний имеет вид

$$\mathrm{D} = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 1 & 2 & 1 \\ 2 & 1 & 0 & 2 & 1 & 1 \\ 3 & 1 & 2 & 0 & 1 & 1 \\ 4 & 2 & 1 & 1 & 0 & 1 \\ 5 & 1 & 1 & 1 & 1 & 0 \end{array}.$$

Список расстояний можно записать в виде множества, состоящего из кортежей длины три: $L_D = \{\langle 1,2,1\rangle, \langle 1,3,1\rangle, \langle 1,4,2\rangle, \langle 1,5,1\rangle, \langle 2,3,2\rangle, \langle 2,4,1\rangle, \langle 2,5,1\rangle, \langle 3,4,1\rangle, \langle 3,5,1\rangle, \langle 4,5,1\rangle\}$.

Первый элемент в кортеже соответствует вершине x_i графа, второй элемент — вершине x_j, а третий элемент кортежа — расстоянию $d_{i,j}$.

Если известны расстояния между любой парой вершин графа, то можно определить его *диаметр* $d(\mathsf{G})$ как максимальное расстояние между его вершинами:

$$d(\mathsf{G}) = \max d_{i,j}, \quad i, j \in I = \{1, 2, \ldots, n\}. \tag{16.6}$$

Кратчайшие простые цепи, связывающие вершины i, $j \in X$ с максимальным расстоянием между ними, называются *диаметрально простыми цепями*.

Пусть v — произвольная вершина $\mathsf{G} = (X, U)$. *Максимальным удалением* в графе G от вершины v называется величина:

$$r(v) = \max d_{v,x}, \quad v, x \in X. \tag{16.7}$$

Согласно Ф. Харари эксцентриситет вершины v в связном графе G определяется как $\max d_{v,x}$, $v, x \in X$ по всем вершинам x.

Тогда *радиусом* графа $r(\mathsf{G})$ называется наименьший из эксцентриситетов вершин. Любая кратчайшая цепь от центра v до максимально удаленной от него вершины x — *радиальной цепью*:

$$r(\mathsf{G}) = \min r(x), \quad x \in X. \tag{16.8}$$

Вершина v называется *центром графа*, если среди всех вершин с максимальным удалением от нее выбирается удаление, принимающее минимальное значение.

Центр не обязательно должен быть единственным. Например, в полном графе K_n радиус равен 1 и любая вершина является центром.

Пусть G — конечный граф $\mathsf{G} = (X, U)$, $|U| = m$. Количество последовательностей ребер этого графа без повторений конечно и равно $m!$. Тогда конечно и число простых цепей, в которых ребра не повторяются. *Протяженностью* $g(v', v'')$ между вершинами v', $v'' \in X$ называется максимальная из длин, связывающих эти вершины.

Если исключить из этих простых цепей циклы, то для любой вершины $v \in X$, $g(v, v) = 0$. В этом случае протяженность также удовлетворяет аксиомам метрики.

Пусть $L(v', v''')$ — самая длинная простая цепь, соединяющая вершины v'–v'''. Пусть $L^*(v'', v')$ — некоторая простая цепь, соединяющая вершины v'' и v'. Пусть (v'', v^4) — участок последней цепи до первого пересечения с L.

Тогда v^4 делит цепь на 2 участка L', L''. Участки L' и $\widetilde{L}$ составляют простую цепь с началом v' и концом v'', а L'' и $\widetilde{L}$ — простую цепь, соединяющую v'' и v''', причем сумма их длин не меньше длины цепи L. Значит, и сумма максимальных длин путей с теми же началами и концами, равная $g(v', v'') + g(v'', v''')$, не меньше длины цепи L, которая равна $g(v', v''')$ (рис. 16.27).

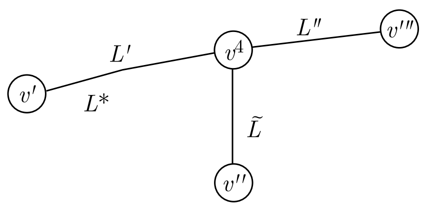


Рис. 16.27. Длины простых цепей

В графе G существуют *диаметральные по протяженности* или длиннейшие простые цепи. Их длина l_0 называется *диаметром протяженности*. Для каждой вершины v существуют самые длинные простые цепи с концами в этой вершине. Их длина $g(v) = \max g(v, v')$, $v' \in v$ называется *числом протяженности* для вершины v.

Для логических задач представляет интерес нахождение функции расстояний для графов G_r частного вида, называемых координатной решеткой. В графе $\mathsf{G}_r = (X_r, U_r)$ множество вершин X_r соответствует узлам решетки (сетки), а множество ребер U_r — горизонтальным и вертикальным отрезкам, соединяющим узлы решетки. Пример графа G_r, т. е. координатной решетки, показан на рис. 16.28. Расстояние между двумя смежными вершинами в G_r, называемое шагом решетки, будем считать равным единице.

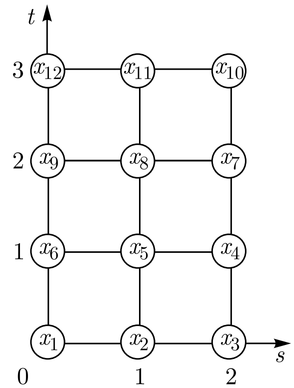


Рис. 16.28. Пример координатной решетки G_r

Расстояние $d_{i,j}$ между двумя произвольными вершинами в G_r можно определить по формуле

$$d_{i,j} = |s_i - s_j| + |t_i - t_j|, \tag{16.9}$$

где s_i, s_j и t_i, t_j — координаты вершин x_i, $x_j \in X$. Обычно задаются размеры решетки $p \times q$, где p — число узлов решетки по оси s, а q — по оси t.

Например, для графа G_r (рис. 16.28) расстояние

$$d_{6,10} = |s_6 - s_{10}| + |t_6 - t_{10}| = |0 - 2| + |1 - 3| = 4,$$

т. е. из вершины x_6 необходимо пройти четыре ребра G_r, чтобы попасть в вершину x_{10}.

Если произвольный граф G отображается в G_r так, что любые вершины G размещаются в узлах решетки, то расстояние между вершинами G определяется как расстояние между соответствующими узлами решетки. Если расстояние $d_{i,j}$ определять как длину кратчайшей прямой, соединяющей вершины x_i, x_j, то

$$d_{i,j} = \sqrt{(s_i - s_j)^2 + (t_i - t_j)^2}\,. \tag{16.10}$$

Заметим, что первый способ определения $d_{i,j}$ проще, так как при этом $d_{i,j}$ принимает только целочисленные значения. Любой граф G может быть отображен в решетку G_r. Для подсчета суммарной длины $L(\mathsf{G})$ ребер графа $\mathsf{G} = (X, U)$, отображенного в решетку G_r, введем понятие матрицы геометрии $\mathrm{D}(\gamma)$.

Матрица геометрии $\mathrm{D}(\gamma)$ представляет собой часть матрицы расстояний D, в которой исключены элементы $d_{i,j}$, если вершины x_i, $x_j \in X$ не смежны в графе G. Для построения матрицы геометрии $\mathrm{D}(\gamma)$ графа G необходимо каждый элемент матрицы D умножить на соответствующий элемент матрицы смежности R: $\mathrm{D}(\gamma) = ||r_{i,j}, d_{i,j}||_{n\times n}$.

Например, граф G без учета весов ребер отобразим в решетку G_r размера 3×2 (рис. 16.29).

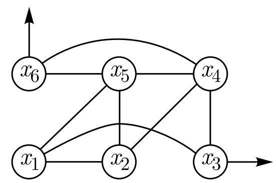


Рис. 16.29. Пример графа, отображенного в решетку

Матрицы R и D графа G примут вид:

$$\mathrm{R} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 & 1 & 0 \\ 3 & 1 & 0 & 0 & 1 & 0 & 0 \\ 4 & 0 & 1 & 1 & 0 & 1 & 1 \\ 5 & 1 & 1 & 0 & 1 & 0 & 1 \\ 6 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}, \qquad \mathrm{D} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 2 & 3 & 2 & 1 \\ 2 & 1 & 0 & 1 & 2 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 5 & 2 & 1 & 2 & 1 & 0 & 1 \\ 6 & 1 & 2 & 3 & 2 & 1 & 0 \end{array}.$$

Тогда можно определить матрицу геометрии $\mathrm{D}(\gamma) = \mathrm{R} \times \mathrm{D}$:

$$\mathrm{D}(\gamma) = \begin{array}{c|cccccc|c} & 1 & 2 & 3 & 4 & 5 & 6 & \rho(x_i) \\ 1 & 0 & 1 & 2 & 0 & 2 & 0 & 5 \\ 2 & 1 & 0 & 0 & 2 & 1 & 0 & 4 \\ 3 & 2 & 0 & 0 & 1 & 0 & 0 & 3\,. \\ 4 & 0 & 2 & 1 & 0 & 1 & 2 & 6 \\ 5 & 2 & 1 & 0 & 1 & 0 & 1 & 5 \\ 6 & 0 & 0 & 0 & 2 & 1 & 0 & 3 \end{array}$$

Сумма элементов матрицы $\mathrm{D}(\gamma)$ определяет удвоенную суммарную длину ребер графа G при данном отображении его в решетку G_r. Для рассмотренного примера суммарная длина ребер графа $L(\mathsf{G})$ составляет 13 условных единиц. При другом отображении графа в решетку суммарная длина $L(\mathsf{G})$ изменяется, хотя в частном случае может совпадать с предыдущим значением.

Примеры решения задач

Пример 16.8. Для графа $\mathsf{G} = (X, U)$, изображенного на рис. 16.30, записать матрицу расстояний и список расстояний.

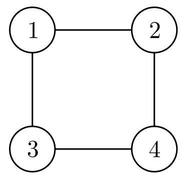


Рис. 16.30. Граф G

Решение. Запишем матрицу расстояний для заданного графа. Матрица расстояний будет иметь вид:

$$\mathrm{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left| \begin{array}{cccc} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{array} \right| \end{array}.$$

Таким образом, первая часть задания выполнена. Теперь построим список расстояний. Его можно записать следующим образом:

$$\mathrm{D} = \{\langle 1, 2, 1\rangle; \langle 1, 3, 1\rangle; \langle 1, 4, 2\rangle; \langle 2, 3, 2\rangle; \langle 2, 4, 1\rangle; \langle 3, 4, 1\rangle\}.$$

Пример 16.9. Для графа $\mathsf{G} = (X, U)$, заданного на рис. 16.31, построить его отображение в координатную решетку и матрицу

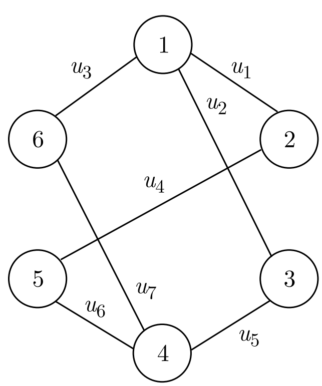


Рис. 16.31. Граф G

геометрии. Подсчитать суммарную длину ребер $L(\mathsf{G})$ графа, отображенного в координатную решетку.

Решение. После отображения в координатную решетку графа G получим граф G_r, показанный на рис. 16.32.

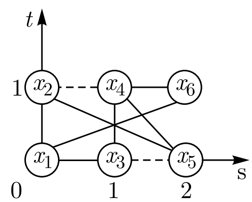


Рис. 16.32. Отображение графа G в координатную решетку

Теперь для полученного графа G_r запишем матрицу геометрии. Как уже говорилось ранее, матрица геометрии получается путем перемножения матрицы расстояний D и матрицы смежности:

$$
\mathrm{D} = \begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 1 & 1 & 2 & 2 & 3 \\
2 & 1 & 0 & 2 & 1 & 3 & 2 \\
3 & 1 & 2 & 0 & 1 & 1 & 2 \\
4 & 2 & 1 & 1 & 0 & 2 & 1 \\
5 & 2 & 3 & 1 & 2 & 0 & 1 \\
6 & 3 & 2 & 2 & 1 & 1 & 0
\end{array}, \qquad
\mathrm{R} = \begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 \\
2 & 1 & 0 & 0 & 0 & 1 & 0 \\
3 & 1 & 0 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 1 & 0 & 1 & 1 \\
5 & 0 & 1 & 0 & 1 & 0 & 0 \\
6 & 1 & 0 & 0 & 1 & 0 & 0
\end{array},
$$

$$
\mathrm{D}_\gamma = \mathrm{D} \times \mathrm{R} = \begin{array}{c|cccccc|c}
 & 1 & 2 & 3 & 4 & 5 & 6 & \sum \\
1 & 0 & 1 & 1 & 0 & 0 & 3 & 5 \\
2 & 1 & 0 & 0 & 0 & 3 & 0 & 4 \\
3 & 1 & 0 & 0 & 1 & 0 & 0 & 2 \\
4 & 0 & 0 & 1 & 0 & 2 & 1 & 4 \\
5 & 0 & 3 & 0 & 2 & 0 & 0 & 5 \\
6 & 3 & 0 & 0 & 1 & 0 & 0 & 4
\end{array}.
$$

В результате перемножения матриц мы получили матрицу геометрии D_γ. Сумма элементов данной матрицы равна 24. Следовательно, суммарная длина ребер $L(\mathsf{G})$ графа G_r равна 12.

Вопросы для самоконтроля

1. Что такое метрика графа?

2. Как определяется расстояние между двумя вершинами графа?

3. Каким образом формируется матрица расстояний?

4. В чем состоит отличие между матрицей расстояний и матрицей геометрии?

5. Каким образом определяется диаметр графа?

6. Каким образом граф отображается в координатную решетку?

7. Как построить список расстояний?

8. Как определяется радиус графа?

9. Что такое число протяженности?

10. Каким образом определяется расстояние между узлами решетки?

Задания для самостоятельной работы

1. Для графа $\mathsf{G} = (X, U)$ построить матрицу и список расстояний.

2. Для произвольного графа $\mathsf{G} = (X, U)$ построить его отображение в координатную решетку G_r, матрицу расстояний, матрицу геометрии, а также подсчитать суммарную длину ребер.

3. Для произвольного графа определить диаметр и радиус графа.

4. Для произвольного графа определите центр графа.

5. Определите в произвольном графе G длиннейшие простые цепи и число протяженности.

6. Постройте произвольный граф на 10 вершин, отобразите его в решетку и определите расстояние между его вершинами.

7. Постройте словесный алгоритм определения матрицы геометрии.

8. Запишите псевдокод алгоритма определения матрицы геометрии.

9. Приведите структурную схему алгоритма определения матрицы геометрии по рисунку графа.

10. Приведите словесный алгоритм отображения графа в координатную решетку.

Понятия расстояния в графе, а также связанные с ним понятия метрики графа, длины цепи, матриц расстояний и геометрии являются базовыми понятиями при решении широкого круга оптимизационных задач, связанных с топологией графа.

16.5. Деревья

Связный граф без циклов называется *деревом* и обозначается $\mathsf{T} = (X, U)$, $|X| = n$. Любое дерево T имеет $n-1$ ребро. Начальная вершина называется *корнем*, из которого выходят ребра — *ветви* дерева. Очевидно, что в дереве любые две вершины x_i, x_j связаны единственной цепью. В любом связном графе G можно выделить произвольное дерево T. В любом дереве порядка $n \geqslant 2$ имеется не менее двух концевых вершин.

Теорема 16.8. *Пусть* T *— граф из* n *вершин. Следующие утверждения эквивалентны*:

- T *является деревом*,
- T *не содержит циклов и имеет* $n-1$ *ребер*,
- T *связен и имеет* $n-1$ *ребер*,
- T *связен и каждое его ребро является мостом*,
- *любые две вершины графа* T *соединены ровно одной простой цепью*,
- T *не содержит циклов*, *но*, *добавляя к нему любое новое ребро*, *получаем ровно один цикл*.

Для задач конструкторского и технологического проектирования наибольший интерес представляют деревья, число вершин которых равно числу вершин графа, из какого выделено это дерево. Такие деревья называются *покрывающими* (ПД), или *остовными*. На рис. 16.33 показан полный граф K_4 и все его 16 ПД.

Отметим, что здесь всего 4 различных покрывающих дерева. Остальные графы в каждом ряду одинаковые, только нарисованы по-разному.

В связном графе G удаление одного ребра, принадлежащего некоторому выбранному циклу, не нарушает связности оставшегося графа. Можно применять эту процедуру до тех пор, пока в G не останется ни одного цикла. В результате получим

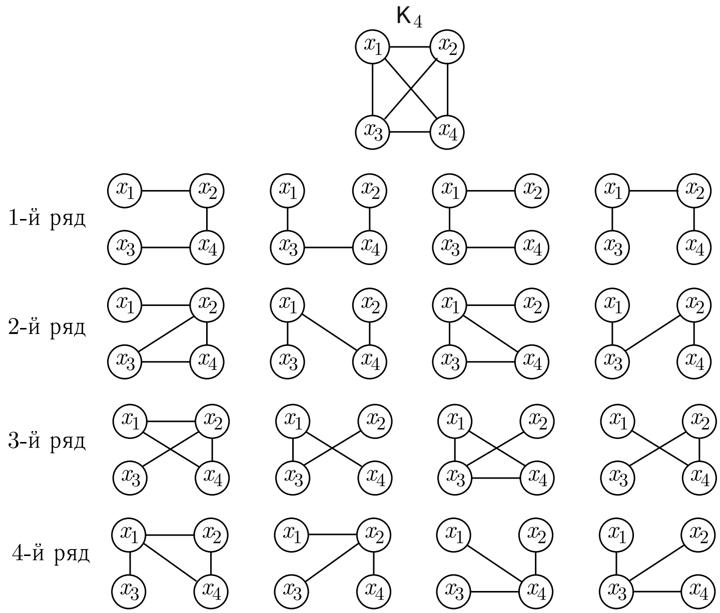


Рис. 16.33. Граф K_4 и его покрывающие деревья

дерево, связывающее все вершины G. Оно является остовным (покрывающим) деревом, или остовом или каркасом графа.

Граф, в котором число ребер не меньше чем число вершин, содержит цикл. Число ребер m произвольного графа $\mathsf{G} = (X, U)$, $|X| = n$, $|U| = m$, которые необходимо удалить для получения каркаса, не зависит от последовательности их удаления.

Например, на рис. 16.34, *а* показан граф G, а на рис. 16.34, *б* — его каркас (остовное дерево).

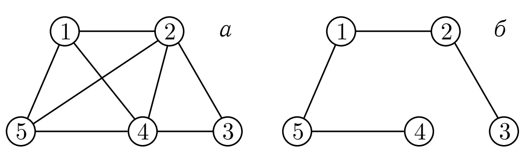


Рис. 16.34. Граф G (*а*) и его остовное дерево (*б*)

Утверждение 16.4. Всякий ациклический подграф $\mathsf{G}' = (X', U')$ произвольного графа $\mathsf{G} = (X, U)$, $X' \subseteq X$, $U' \subseteq U$ содержится в некотором каркасе O графа G ($\mathsf{O} = (X, U')$, $\mathsf{G} = (X, U)$, $U' \subseteq U$).

Утверждение 16.5. Если $\mathsf{O}_1 = (X_1, U_1)$ и $\mathsf{O}_2 = (X_2, U_2)$ — два каркаса графа $\mathsf{G} = (X, U)$ ($X = X_1 = X_2$, $U_1 \subseteq U$, $U_2 \subseteq U$), то для любого ребра $u_i \in U_1$ графа O_1 существует такое ребро $u_j \in U_2$ графа O_2, что граф $\mathsf{O}_1 - u_i + u_j$ также является каркасом. Например, имеем граф G (рис. 16.35, *а*) и два каркаса этого графа O_1 (рис. 16.35, *б*) и O_2 (рис. 16.35, *в*). Пусть ребро $u_i = u_{3,4}$, а ребро $u_j = u_{4,5}$, ребро $u_i \in U_1$ графа O_1 и ребро $u_j \in U_2$ графа O_2. Тогда граф $\mathsf{O}_1 - u_i + u_j = \mathsf{O}_1 - u_{3,4} + u_{4,5}$ также будет каркасом (рис. 16.35, *г*).

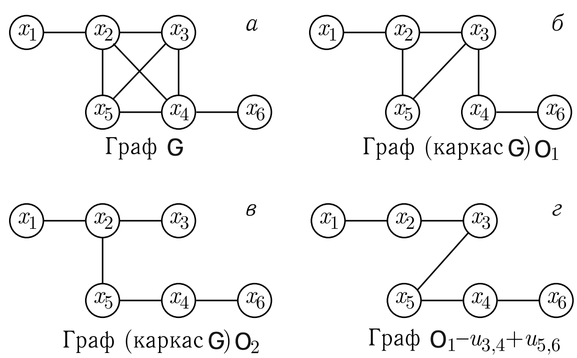


Рис. 16.35. Граф G и его каркасы

Теорема 16.9 (Кэли). *Существует ровно n^{n-2} различных помеченных деревьев с n вершинами.*

Следствие 16.1. *Число остовных деревьев в K_n равно n^{n-2}.*

Множество деревьев графа называется *лесом*. На рис. 16.36 показан лес, состоящий из 3 деревьев T_1–T_3.

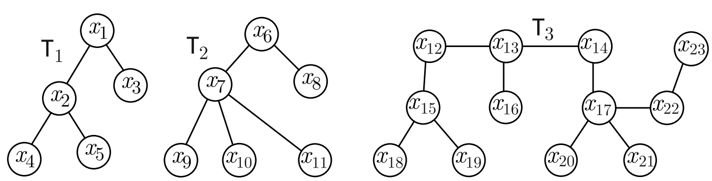


Рис. 16.36. Множество деревьев графа

Задачи выделения эйлеровых и гамильтоновых циклов и покрывающих деревьев связаны с задачами о лабиринте, коммивояжере и с построением цепей, циклов, маршрутов минимальной стоимости.

Задача о лабиринте в терминах теории графов формулируется как задача отыскания в связном графе $\mathsf{G} = (X, U)$ такого маршрута, который начинается в заданной вершине $x_i \in X$ и приводит в искомую вершину $x_j \in X$, причем маршрут должен содержать кратчайшее число ребер.

Пусть нам необходимо проложить сеть проводов, связывающих n терминалов вычислительной аппаратуры, причем так, чтобы из одного терминала можно было связаться с любым другим. Если из экономических соображений требуется, чтобы количество затраченного провода было минимально, то граф, вершины которого соответствуют терминалам, а ребра — соединяющим их проводам, должен быть деревом. Задача состоит в определении одного из n^{n-2} возможных деревьев, соединяющих терминалы, с минимизацией числа проводов. На языке теории графов эту задачу можно сформулировать следующим образом. Пусть $\mathsf{G} = (X, U)$ — связный граф и каждому ребру $u_i \in U$ ставится в соответствие некоторое неотрицательное $\nu(u_i)$, называемое его мерой или весом. Необходимо найти алгоритм построения покрывающего дерева T, сумма мер которого, взятая по всем ребрам, минимальна. Такие деревья будем называть кратчайшими покрывающими деревьями (КПД). Графы с весами на ребрах или вершинах назовем взвешенными.

Запишем алгоритм Краскала для построения кратчайшего покрывающего дерева.

1. Пусть задан граф $\mathsf{G} = (X, U)$, для которого нужно построить кратчайшее покрывающее дерево. Строим граф $\mathsf{G}_1 = \mathsf{O}_{\text{п}} + u_1$, присоединяя к пустому графу $\mathsf{O}_{\text{п}}$ на множестве вершин X ребро $u_1 \in U$ минимального веса.

2. Если граф G_i уже построен и $i < n - 1$, то строим граф $\mathsf{G}_{i+1} = \mathsf{G}_i + u_{i+1}$, где u_{i+1} — ребро графа G, имеющее минимальный вес среди ребер, не входящих в G_i и не составляющих циклов с ребрами из G_i. При $i = n - 1$ переход к п. 3.

3. Конец работы алгоритма.

Теорема 16.10. *При $i < n - 1$ граф G_{i+1} можно построить. Граф G_{i-1} является каркасом с минимальным весом ребер в графе G.*

Например, пусть задан граф с G весами на ребрах (рис. 16.37).

Согласно алгоритму Краскала выбираем сначала ребро $u_{1,4}$ с весом 1. Далее берем ребро $u_{1,5}$ с весом 1. Затем среди оставшихся ребер с минимальным весом есть ребро $u_{4,5}$, однако оно непригодно для каркаса, так как образует цикл с ребрами $u_{1,4}$ и $u_{1,5}$. Далее выбираем ребра $u_{3,6}$ и $u_{5,6}$ с весом 2. Итак, каркас с минимальным весом, равным 8, построен (рис. 16.37, *б*).

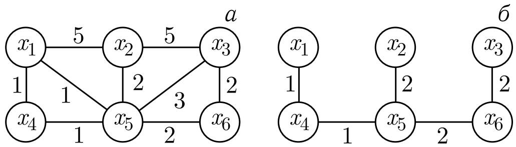


Рис. 16.37. Взвешенный граф G (*а*); каркас графа G (*б*)

Приведем логическую схему алгоритма (ЛСА) Краскала решения данной задачи:

$$A_0 A_1 \downarrow^2 \downarrow^1 A_2 p_1 \uparrow^1 p_2 \uparrow^2 A_{\text{к}},$$

здесь A_0 — оператор начала; A_1 — оператор, устанавливающий $i = 1$; A_2 — оператор, выбирающий ребро u_i с наименьшим весом; p_1 — проверка логического условия несовпадения ребра u_i с предыдущим и необразования цикла с предыдущими ребрами. При $p_1 = 0$, $i = i + 1$ переход к A_2, при $p_1 = 1$ переход к p_2; p_2 — проверка логического условия, что мощность выбранных ребер равна $n - 1$. При $p_2 = 0$, $i = i + 1$ переход к A_2, при $p_2 = 1$ переход к $A_{\text{к}}$; $A_{\text{к}}$ — оператор конца алгоритма.

Опишем алгоритм Прима построения кратчайшего покрывающего дерева (КПД) на взвешенном по ребрам графе. Алгоритм основан на разрастании поддеревьев. Дерево $\mathsf{T}' = (X', U')$ является поддеревом $\mathsf{T} = (X, U)$, если $X' \subseteq X$, $U' \subseteq U$. Причем поддерево T' может состоять и из одной вершины. Поддерево T' последовательно разрастается за счет прибавления ребер (x_i, x_j), $(x_i \in \mathsf{T}$, $x_j \notin \mathsf{T}')$. Каждое добавляемое ребро должно иметь наименьший вес. Процесс продолжается, пока число ребер в T' не станет равным $n - 1$. Покажем работу алгоритма на построении кратчайшего покрывающего дерева в графе G. Пусть граф задан матрицей смежности

$$\mathrm{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left| \begin{array}{cccccc} 0 & 5 & 8 & 0 & 37 & 0 \\ 5 & 0 & 0 & 7 & 10 & 0 \\ 8 & 0 & 0 & 3 & 0 & 0 \\ 0 & 7 & 3 & 0 & 11 & 20 \\ 37 & 10 & 0 & 11 & 0 & 12 \\ 0 & 0 & 0 & 20 & 12 & 0 \end{array} \right| \end{array}.$$

Выбираем вершину x_1 и считаем ее на первом шаге поддеревом $\mathsf{T}' = (X', U')$, $X' = \{x_1\}$, $U' = \varnothing$. Строим упорядоченный список ребер, инцидентных вершине x_1. Упорядочивание производится по возрастанию веса ребер:

$$L = \begin{vmatrix} u_i & \nu(u_i) \\ (1,2) & 5 \\ (1,3) & 8 \\ (1,5) & 37 \end{vmatrix}.$$

Из списка выбирается ребро $(1,2)$ (т. е. (x_1, x_2)). Производится разрастание поддерева $\mathsf{T}' = (X', U')$, $X' = \{x_1, x_2\}$, $U' = \{(1,2)\}$. В список L вносятся ребра, инцидентные вершине x_2. После этого ребра снова упорядочиваются, и список примет вид

$$L = \begin{vmatrix} u_i & \nu(u_i) \\ (2,4) & 7 \\ (1,3) & 8 \\ (2,5) & 10 \\ (1,5) & 37 \end{vmatrix}.$$

Выбирается ребро $(2,4)$ с наименьшим весом и добавляется к поддереву T', если оно не образует цикла с ребрами T'. Остальные ребра, объединяющие вершины X' между собой, удаляются из L; $X' = \{x_1, x_2, x_4\}$, $U' = \{(1,2),(2,4)\}$. Проверяется $|X'| = |X|$; если нет, то выбирается вершина x_4 и ребра, инцидентные ей, пополняют список L:

$$L = \begin{vmatrix} u_i & \nu(u_i) \\ (4,3) & 3 \\ (1,3) & 8 \\ (2,5) & 10 \\ (4,5) & 11 \\ (4,6) & 20 \\ (1,5) & 37 \end{vmatrix}.$$

Продолжая, получаем $U' = \{(1,2),(2,4),(4,3),(2,5),(5,6)\}$, $|X| = |X'|$, процесс окончен, $\sum\limits_{1}^{n-1} \nu(u_i) = 32$. На рис. 16.38 кратчайшее покрывающее дерево показано жирными линиями.

Алгоритмы построения покрывающих деревьев с минимальной суммарной длиной ребер на взвешенных графах основаны на двух принципах:

• каждая изолированная вершина соединяется с ближайшей соседней;

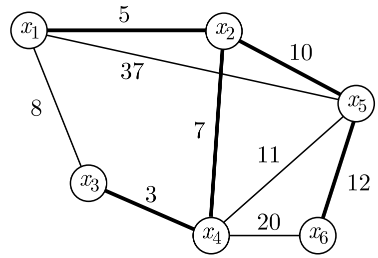


Рис. 16.38. Исходный граф **G** и его кратчайшее покрывающее дерево

• каждое изолированное поддерево соединяется с ближайшей соседней вершиной (или поддеревом) кратчайшим ребром (или цепью).

Задача построения КПД усложняется, если при соединении множества вершин (контактов цепи) разрешается использование дополнительных точек соединения. В общем виде данная задача формируется так: для заданных $x_1, x_2, \ldots, x_n$ точек плоскости построить КПД с $n' \geqslant n$ вершинами — и называется *задачей Штейнера* (ЗШ). Обычно решение ЗШ рассматривают в областях прямоугольной конфигурации. Для $n \leqslant 5$ известно решение ЗШ, в общем же случае известны условия, которым должны удовлетворять деревья Штейнера (ДШ) и эвристические алгоритмы. Дополнительные точки, вводимые при построении КПД, называют *точками Штейнера* (ТШ).

Приведем ряд лемм, полученных Фридманом и Меноном, дающих необходимые условия существования и построения ДШ.

Лемма 16.1. *Для n вершин, соединенных между собой, всегда можно построить ДШ, в котором каждая ТШ g_i соединена по крайней мере с тремя другими.*

Лемма 16.2. *Для множества вершин $\{x_1, x_2, x_3\}$, которые должны быть соединены, координаты ТШ g, минимизирующей $\sum\limits_{i=1}^{3} d(g, n_i)$, будут $S_{ср}$, $T_{ср}$ и справедливо выражение $\sum\limits_{i=1}^{3} d(g, n_i) = 1/2P(x_1, x_2, x_3)$, где $S_{ср}$, $T_{ср}$ — средние значения S_i $(i = 1, 2, 3)$ и T_j $(j = 1, 2, 3)$; $P(x_1, x_2, x_3)$ — длина периметра прямоугольника, построенного на вершинах x_1, x_2, x_3; $d(g, n_i)$ — прямоугольное расстояние между g и n_i.*

На рис. 16.39 показан пример, иллюстрирующий условия леммы 16.2 при построении ДШ, соединяющего вершины x_1, x_2, x_3. Точка Штейнера g имеет координаты $Sg = 3$, $Tg = 5$.

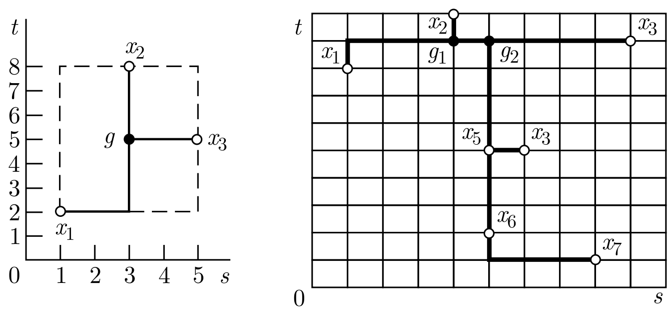


Рис. 16.39. Построения дерева Штейнера

Лемма 16.3. *Дано ДШ, состоящее из x_i точек плоскости. Пусть ТШ g соединена только с тремя вершинами x_1, x_2, x_3. Тогда g находится внутри прямоугольника, построенного на вершинах x_1, x_2, x_3, и ТШ единственная.*

Следствие 16.2. *Для заданного множества вершин x_1, x_2, x_3 ДШ содержит одну ТШ g с координатами (S_{cp}, T_{cp}), если она не совпадает ни с одной из вершин x_1, x_2, x_3 и суммарная длина ребер ДШ равна $1/2P(x_1, x_2, x_3)$.*

Лемма 16.4. *Дано ДШ на множестве $X = \{x_1, x_2, \ldots, x_n\}$, а ТШ этого дерева образуют множество $G = \{g_1, g_2, \ldots, g_k\}$. Если $g_1 \subset G$ и $g_i \in I$, то g_1 содержит элемент g_{1k}, соединенный не менее чем с двумя вершинами из I, в противном случае g_1 пусто. Здесь I — множество точек пересечения линий координатной сетки.*

На основе приведенных лемм сформулирована известная теорема.

Теорема 16.11. *Дано ДШ на множестве вершин $X = \{x_1, x_2, \ldots, x_n\}$ с множеством ТШ $G = \{g_1, g_2, \ldots, g_k\}$. Тогда для этого множества существует такое другое ДШ с множеством ТШ G', что $(\forall g'_i \in G')(g'_i \in I)$.*

Из рассмотрения теоремы следует метод определения ДШ на основе полного перебора в качестве ТШ всех возможных подмножеств точек I. Очевидно, что такой метод применим только при построении ДШ для небольшого числа ТШ. В этой связи разрабатываются эвристические процедуры построения квазиоптимальных ДШ.

Процедура 1.

1°. Все вершины $x_i \in X$ проектируются на ось $S(T)$.

2°. Расстояние от наименьшей до наибольшей координаты делится пополам, и из этой точки $i \in I$ проводится перпендикуляр (столб Штейнера).

3°. Из каждой вершины $x_i \in X$ опускается перпендикуляр до пересечения со столбом Штейнера.

4°. Построено ДШ. Конец работы алгоритма.

На основе данной процедуры можно предложить следующий алгоритм построения дерева Штейнера.

Итак, во-первых, мы отображаем заданный граф G в координатную решетку, а затем находим координаты всех вершин полученного графа G_r и вычисляем, по какой из осей координат разность максимального и минимального значений координат будет максимальной.

После этого проводим перпендикуляр к той оси, относительно которой разность координат меньше. При этом перпендикуляр проводим либо на равном удалении от крайних точек, либо, если количество вершин с одной стороны перпендикуляра значительно превышает число вершин с другой, немного сдвигаем перпендикуляр по оси координат в сторону большего числа вершин.

Полученный таким образом перпендикуляр представляет собой ось или ствол будущего дерева, который затем соединяют отрезками с остальными вершинами графа.

Данный алгоритм очень прост, ВСА равна $O(n)$, но суммарная длина проводников дерева Штейнера далеко не оптимальна.

Процедура 2 (алгоритм Ханана).

1°. Все множество вершин $x_i \in X$ разбивается на классы $S_0, S_1, \ldots, S_i$ в порядке возрастания координаты S_i так, чтобы у всех вершин одного класса была одинаковая координата S_i.

2°. Анализируется множество вершин класса S_0, и соединяются перпендикуляром все вершины этого множества с точкой $S_{\min}$ оси S.

3°. Образуется множество I', состоящее из всех точек I, которые были до этого соединены с произвольной вершиной x_i, включая и вершины x_i строящегося ДШ.

4°. Выбирается следующее множество S_{i+1}, имеющее наименьшую координату S. Определяются точка $i_m \in I'$ и вершина $x_j \in S_{i+1}$, для которых

$$d(i_m, x_j) \geqslant d(i_q, x'_j), \quad \forall i_q \in I', \quad \forall x'_j \in S_{i+1}.$$

Другими словами, x_i находится на кратчайшем расстоянии от рассматриваемого фрагмента ДШ в классе S_{i+1}. Точка i_m с координатами (s_m, t_m) соединяется с вершиной x_j с координатами (s_i, t_i) двухзвенной линией. Причем звенья параллельны координатным осям S и T.

5°. Для каждой вершины $x_k \in S_{i+1}$ производятся операции, аналогичные 4°. При этом вершина x_k соединяется с ближайшей в I'.

6°. Пункты 4° и 5° повторяются для всех по порядку множеств S_j до построения ДШ.

Отметим, что существует большое число модификаций описанных процедур. Можно проводить упорядочивание координат не по увеличению $S(T)$, а по уменьшению $S(T)$. Временная сложность алгоритма процедуры 2 составляет $O(n^2)$. На рис. 16.39, *б* показана реализация процедуры 2 с упорядочиванием вершин в 1° в порядке убывания координат S. Суммарная длина ДШ составляет 22 условные единицы, при этом число ТШ равно 2.

Для построения ДШ можно использовать методы поиска или ветвей и границ в зависимости от требований на ВСА.

Заметим, что на основе модели дерева можно описывать любое алгебраическое выражение, например, дерево (рис. 16.40) может моделировать следующие записи:

$$(2+y)^*7,\ ^*+2y7,\ ^*(2+y)7, 2y+7^*, (2+y)7^* \text{ и т. п.}$$

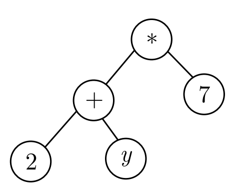


Рис. 16.40. Пример записи алгоритма на основе модели дерева

Запись, где символы операций предшествуют операндам, называется польским выражением, или польской записью. Это $^*+2y7$. Использование таких моделей позволяет разрабатывать эффективные программы и алгоритмы преобразования алгебраических выражений.

Примеры решения задач

Пример 16.10. Пусть задан граф $\mathsf{G} = (X, U)$, изображенный на рис. 16.41, a. Привести примеры произвольного и покрывающего деревьев для заданного графа.

Ответ. Пример покрывающего дерева приведен на рис. 16.41, *б*. На рис. 16.41, *в*, *г* показаны примеры выделения произвольных деревьев из исходного графа.

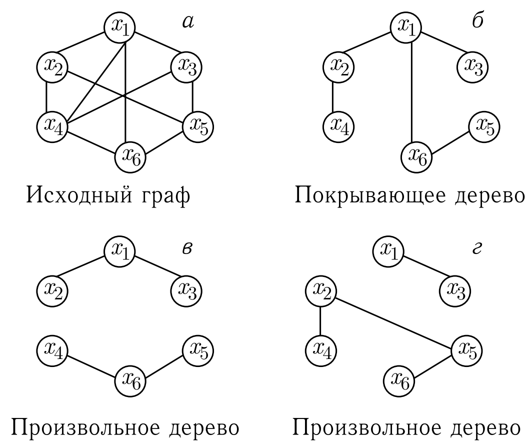


Рис. 16.41. Граф G и его деревья: *а* — исходный граф; *б* — покрывающее дерево; *в* — произвольное дерево; *г* — произвольное дерево

Пример 16.11. Записать все покрывающие деревья для полного графа на три вершины, заданного на рис. 16.42, a.

Решение. Согласно теореме Кэли число покрывающих деревьев полного графа равно: $t = n^{n-2}$.

Таким образом, подставив в формулу число вершин заданного графа, получим результат: $t = 3^{3-2} = 3^1 = 3$.

Отсюда следует, что для данного графа можно построить три покрывающих дерева, как это показано на рис. 16.42, *б*.

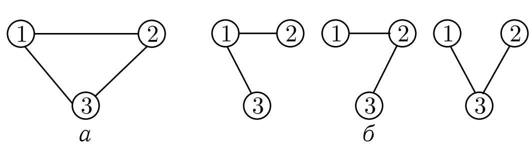


Рис. 16.42. Граф (*а*) и множество его покрывающих деревьев (*б*)

Пример 16.12. Пусть задан граф $\mathsf{G} = (X, U)$, изображенный на рис. 16.43, *а*. Построить кратчайшее покрывающее дерево (КПД), используя алгоритм Краскала.

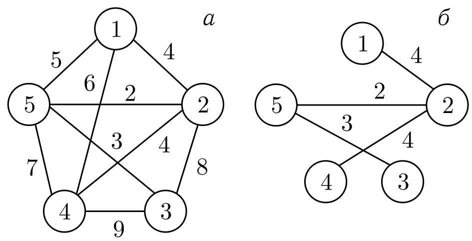


Рис. 16.43. Граф G (*а*); кратчайшее покрывающее дерево для графа G (*б*)

(Нумерация в примере представляет собой двухуровневый список, первая цифра которого соответствует номеру итерации алгоритма, а вторая — шагу алгоритма на текущей итерации.)

Решение. Запишем треугольную матрицу смежности R заданного графа:

$$\mathrm{R} = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 4 & 0 & 6 & 5 \\ 2 & & 0 & 8 & 4 & 2 \\ 3 & & & 0 & 9 & 3 \\ 4 & & & & 0 & 7 \\ 5 & & & & & 0 \end{array}.$$

1.1. Выбираем по матрице смежности минимальный элемент. Это ребро u_{25}. Заносим его в массив $M = \{u_{25}\}$.

1.2. Выбранное ребро единственное, поэтому переходим к п. 3.

1.3. $M \neq n - 1$, переходим к п. 1.

2.1. Выбираем ребро u_{35}, заносим его в массив $M = \{u_{25}, u_{35}\}$.

2.2. Цикл не образуется, поскольку в массиве всего два ребра.

2.3. $|M| \neq n - 1$, переход к п. 1.

3.1. Выбираем ребро u_{12}, заносим его в список $M = \{u_{25}, u_{35}, u_{12}\}$.

3.2. Проверяем выбранные ребра. Цикл не образуется, поэтому переходим к п. 3.

3.3. $M \neq n-1$, переходим к п. 1.

4.1. Выбираем ребро u_{24}, заносим его в список $M = \{u_{25}, u_{35}, u_{12}, u_{24}\}$.

4.2. Проверяем выбранные ребра. Цикл не образуется, поэтому переходим к п. 3.

4.3. Мощность $|M| = 4$. Так как $4 = 5 - 1$, алгоритм закончен. Кратчайшее покрывающее дерево (КПД) построено (рис. 16.43, *б*). Его суммарный вес равен $\sum\limits_{\text{КПД}} = 2 + 3 + 4 + 4 = 13$.

Пример 16.13. Построить КПД для графа G, изображенного на рис. 16.44, используя алгоритм Прима.

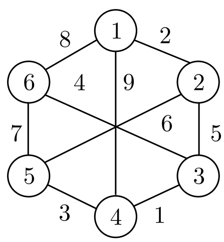


Рис. 16.44. Исходный граф

Решение.

1.1. Выбираем произвольную вершину. Пусть это будет вершина x_1. Составляем список:

$$L_1 = \begin{vmatrix} (1,2)2 \\ (1,6)8 \\ (1,4)9 \\ (U_{ij})V_4 \end{vmatrix}, \quad \text{где } V_4 \text{ — вес ребра.}$$

1.2. Из списка L_1 выбираем ребро $(1,2)$ и включаем его в поддерево $\mathsf{T}' = \{(1,2)\}$.

1.3. Так как цикл не образуется, переходим к п. 4.

1.4. Как видно $|X'| \neq |X|$, следовательно, переходим к п. 5.

1.5. Добавляем в список L_1 ребра, инцидентные вершине x_2 и после переупорядочивания списка L_1 получаем новый список L_2:

$$L_2 = \begin{vmatrix} (2,3)5 \\ (2,5)6 \\ (1,6)8 \\ (1,4)9 \end{vmatrix}.$$

2.2. Выбираем из списка L_2 ребро $(2,3)$ и включаем его в поддерево $\mathsf{T}' = \{(1,2);(2,3)\}$.

2.3. Цикл не образуется, переход к п. 4.

2.4. $|X'| \neq |X|$, переходим к п. 5.

2.5. Добавляем в список L ребра, инцидентные вершине x_3, и переупорядочиваем список. Переходим к п. 2.

$$L_3 = \begin{vmatrix} (3,4)1 \\ (3,6)4 \\ (2,5)6 \\ (1,6)8 \\ (1,4)9 \end{vmatrix}.$$

3.2. Из списка L_3 выбираем ребро $(3,4)$ и заносим его в поддерево $\mathsf{T}' = \{(1,2);(2,3);(3,4)\}$.

3.3. Цикл не образуется, переходим к п. 4.

3.4. $|X'| \neq |X|$, переходим к п. 5.

3.5. Добавляем в список L ребра, инцидентные вершине x_4, и переупорядочиваем список. Ребро $(1,4)$ исключено из списка L_4, так как оно образует цикл с уже выбранными ребрами. Переходим к п. 2.

$$L_4 = \begin{vmatrix} (4,5)3 \\ (3,6)4 \\ (2,5)6 \\ (1,6)8 \end{vmatrix}.$$

4.2. Из списка L_4 выбираем ребро $(4,5)$ и включаем его в поддерево $\mathsf{T}' = \{(1,2);(2,3);(3,4);(4,5)\}$.

4.3. Цикл не образуется, переходим к п. 4.

4.4. $|X'| \neq |X|$, переходим к п. 5.

4.5. Добавляем в список L ребра, инцидентные вершине x_5, и переупорядочиваем список. Ребро $(2,5)$ исключено из списка L_5, так как оно образует цикл с уже выбранными ранее ребрами. Переходим к п. 2.

$$L_5 = \begin{vmatrix} (3,6)4 \\ (5,6)7 \\ (1,6)8 \end{vmatrix}.$$

5.2. Из списка L_5 выбираем ребро $(3,6)$ и включаем его в поддерево $\mathsf{T}' = \{(1,2);(2,3);(3,4);(4,5);(3,6)\}$.

5.3. Цикл не образуется, переход к п. 4.

5.4. $|X'| = |X|$, переход к п. 6.

5.6. Построено КПД (рис. 16.45): $\mathsf{T}' = \{(1{,}2); (2{,}3); (3{,}4); (4{,}5); (3,6)\}$ с суммарным весом, равным $\sum\limits_{\text{КПД}} = 2 + 5 + 1 + 3 + 4 = 15$.

Работа алгоритма закончена.

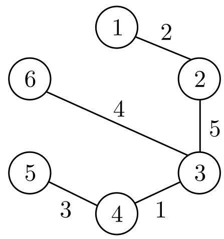


Рис. 16.45. Кратчайшее покрывающее дерево

Пример 16.14. Для графа G_r, заданного на рис. 16.46, построить кратчайшее покрывающее дерево.

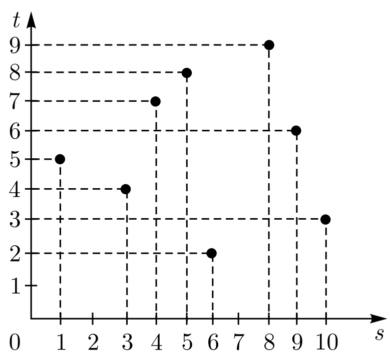


Рис. 16.46. Задача построения дерева Штейнера

Решение. Подсчитаем разность координат по обеим осям:

$$\text{ось } s\colon \quad 10 - 1 = 9,$$

$$\text{ось } t\colon \quad 9 - 2 = 7.$$

Перепад координат по оси t меньше, следовательно, проводим перпендикуляр к оси t.

Теперь определим точку, через которую мы проведем ось дерева. Поскольку число вершин слева и справа от предполагаемой оси равно, проводим перпендикуляр, к примеру, через точку с координатой $t = 6$. Таким образом, мы провели ось дерева Штейнера.

И теперь, соединив с полученной осью все вершины графа, не находящиеся на оси, мы получим в итоге дерево Штейнера (рис. 16.47). Суммарная длина дерева Штейнера находится путем простого сложения длин отрезков, составляющих дерево:

$$L = 1 + 2 + 1 + 2 + 4 + 3 + 3 + 9 = 25.$$

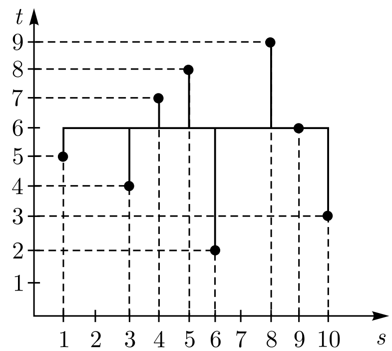


Рис. 16.47. Дерево Штейнера

Вопросы для самоконтроля

1. Приведите механизмы определения деревьев в произвольном графе.
2. Приведите определение произвольного, покрывающего и кратчайшего покрывающего деревьев в графе.
3. Что в теории графов подразумевается под терминами «корень», «ветвь», «лес»?
4. Каким образом определяется число покрывающих деревьев в произвольном полном графе?
5. Сформулируйте задачу о лабиринте.
6. Приведите теорему для определения дерева.
7. Запишите теорему Кэли и следствие из нее.
8. Приведите теорему о каркасе минимального веса в графе.
9. В чем заключаются алгоритмы Краскала и Прима?
10. Приведите леммы о деревьях Штейнера.

Задания для самостоятельной работы

1. Постройте все покрывающие деревья для полного графа на 4 вершины.
2. Для произвольного взвешенного графа $\mathsf{G} = (X, U)$ постройте кратчайшее покрывающее дерево, используя алгоритм Краскала.

3. Для произвольного взвешенного графа $\mathsf{G} = (X, U)$ постройте кратчайшее покрывающее дерево, используя алгоритм Прима.

4. Постройте структурную схему алгоритма Прима для построения кратчайшего покрывающего дерева в произвольном неориентированном взвешенном графе.

5. Постройте структурную схему алгоритма Краскала для построения кратчайшего покрывающего дерева в произвольном неориентированном взвешенном графе.

6. Постройте словесный алгоритм Краскала.

7. Постройте словесный алгоритм Прима.

8. Постройте структурную схему алгоритма Ханаана.

9. Запишите структурную схему алгоритма Штейнера построения кратчайшего покрывающего дерева.

10. Основываясь на информации, полученной в ходе изучения материалов данного пособия и конспекта лекций, предложите эвристические процедуры, улучшающие результаты работы алгоритма Штейнера. Для произвольного графа, отображенного в координатную решетку G_r, постройте кратчайшее покрывающее дерево.

Деревья являются важнейшим классом графов, они позволяют моделировать исследования электрических цепей, проектирование объектов, процессы в химии, биологии, генетике.

Глава 17
ГРАФОВЫЕ ИНВАРИАНТЫ

> Математика устанавливает отношения между идеями, вскрывая необходимые связи между ними, а такие связи разум постигает лучше всего.
>
> *Д. Люкк*

Цикломатическое число графа, матрица фундаментальных циклов хроматическое число графа, раскраска вершин графа, последовательный метод раскраски, двудольный граф, число внутренней устойчивости, число внешней устойчивости, независимое подмножество, планарность графа, критерии планарности, максимальный планарный граф, толщина графа, число пересечений графа, минимизация числа пересечений

ЦЕЛИ

Освоив эту главу, студенты должны:

- уметь определять цикломатическое число графа;
- уметь строить матрицу фундаментальных циклов;
- знать, каким образом граф отображается в координатную решетку;
- уметь определять хроматическое число графа;
- уметь строить раскраску вершин графа;
- уметь определять число внутренней и внешней устойчивости;
- знать определения планарного и плоского графа;
- уметь определять планарность графа исходя из приведенных критериев и эвристик;
- знать, как определяется число пересечений графа;
- уметь определять минимальное число пересечений графа;
- уметь строить графы с минимальным числом пересечений.

17.1. Цикломатическое и хроматическое числа графа

Заметим, что реализация графовых алгоритмов упрощается при использовании таких числовых инвариантных характеристик графа, как числа цикломатическое, хроматическое и др.

Наименьшее число ребер, которое необходимо удалить из графа G, чтобы он стал ациклическим (деревом), называется *цикломатическим числом* графа. Для графа $\mathsf{G} = (X, U)$, $|X| = n$, $|U| = m$ цикломатическое число

$$\gamma(\mathsf{G}) = m - n + k, \tag{17.1}$$

где k — число компонентов связности графа, ($k = 1, 2, \ldots$). Величину $\rho(\mathsf{G}) = n - k$ называют коцикломатическим числом. Тогда $\gamma(\mathsf{G}) = m - \rho(\mathsf{G})$.

Циклы, получаемые добавлением какого-нибудь ребра из графа G к ребрам дерева и отличающиеся один от другого хотя бы одним ребром, называют *фундаментальными циклами*. Очевидно, для графа, состоящего из одной компоненты связности,

$$\gamma(\mathsf{G}) = m - n + 1. \tag{17.2}$$

Например, для графа G (рис. 16.38) $\gamma(\mathsf{G}) = 9 - 6 + 1 = 4$, т. е. после удаления четырех ребер он становится деревом. В примере это могут быть ребра (x_1, x_3), (x_1, x_5), (x_4, x_5), (x_4, x_6). Тогда получаем дерево T, выделенное жирным шрифтом.

Заметим, что цикломатическое число графа не зависит от ориентации его ребер и формула его определения справедлива для мультиграфов. Из определения $\gamma(\mathsf{G})$ следует, что оно может быть только положительным или равным нулю. Граф (мультиграф) не имеет циклов тогда и только тогда, когда $\gamma(\mathsf{G}) = 0$. Очевидно, что граф (мультиграф) имеет единственный цикл тогда и только тогда, когда $\gamma(\mathsf{G}) = 1$.

Цикломатическое число $\gamma(\mathsf{G})$ показывает только число ребер, которые необходимо удалить из графа. Для получения конкретных ребер необходимо каждый раз удалять ребро, которое разрушает хотя бы один цикл.

Матрица $\mathrm{R}_c = ||r_c(i, j)||_{\gamma(\mathsf{G}) \times m}$, состоящая из $\gamma(\mathsf{G})$ строк и m столбцов, называется матрицей фундаментальных циклов, если

$$r_c(i, j) = \begin{cases} 1, & \text{если ребро } u_j \text{ принадлежит циклу } c_i; \\ 0 & \text{в противном случае.} \end{cases}$$

Для графа G (рис. 16.38) матрица R_c запишется следующим образом:

$$\mathrm{R}_c = \begin{array}{c|ccccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ c_1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ c_2 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ c_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ c_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}.$$

$u_1 = (x_1, x_2)$, $u_2 = (x_1, x_3)$, $u_3 = (x_1, x_5)$, $u_4 = (x_2, x_4)$, $u_5 = (x_2, x_5)$, $u_6 = (x_3, x_4)$, $u_7 = (x_4, x_5)$, $u_8 = (x_4, x_6)$, $u_9 = (x_5, x_6)$.

Раскраской вершин графа называется разбиение множества вершин X на l непересекающихся классов (подмножеств):

$$X_1, X_2, \ldots, X_l; \quad X = \bigcup_{i=1}^{l} X_i; \quad X_i \bigcap X_j = \varnothing; \quad i, j = I\{1, 2, \ldots, l\}, \tag{17.3}$$

таких, что внутри каждого подмножества X_i не должно содержаться смежных вершин. Если каждому подмножеству X_i поставить в соответствие определенный цвет, то вершины этого подмножества можно окрасить в один цвет, вершины другого подмножества X_j — в другой цвет, и т. д. до раскраски всех подмножеств. В этом случае граф называется l-раскрашиваемым. Наименьшее число подмножеств, на которое разбивается граф при раскраске, называется *хроматическим числом* $\chi(\mathsf{G})$.

Очевидно, что полный граф K_n можно раскрасить только n цветами, следовательно, $\chi(\mathsf{K}_n) = n$. Для связного графа $\mathsf{G} = (X, U)$ с $(n-1) \leqslant m \leqslant n(n-1)/2$ верхняя оценка хроматического числа

$$\chi(\mathsf{G}) = \frac{3 + \sqrt{9 + 8(m-n)}}{2}. \tag{17.4}$$

Известно утверждение, что для любого графа хроматическое число превышает максимальную локальную степень вершины не более чем на единицу, т. е.

$$\chi(\mathsf{G}) \leqslant 1 + \max \rho(x_i), \quad x_i \in X, \quad i \in I = \{1, 2, \ldots, n\}. \tag{17.5}$$

Если известно число вершин в наибольшем полном подграфе K_i графа G, то

$$\chi(\mathsf{G}) = n/n_i,$$

где n_i — число вершин в подграфе K_i. Известны еще такие оценки:

$$\chi(\mathsf{G}) \geqslant \frac{n^2}{n^2 - m^2}, \quad \chi(\mathsf{G}) \leqslant n + 1 - \chi(\overline{\mathsf{G}}), \tag{17.6}$$

где $\chi(\overline{\mathsf{G}})$ — хроматическое число графа $\overline{\mathsf{G}} = \mathsf{K} \backslash \mathsf{G}$.

Хроматическое число обычно определяют аналитически с помощью методов линейного программирования. Рассмотрим один из возможных методов определения $\chi(\mathsf{G})$, равного g.

Пусть справедливы следующие утверждения:

$$\xi(j,p) = \begin{cases} 1, \text{ если вершина } x_j \text{ окрашена цветом } p \ (p = 1, 2, \ldots, q); \\ 0 \text{ в противном случае} \end{cases}$$

и

$$i(j,l) = \begin{cases} 1, & \text{если вершина } x_j \text{ инцидентна ребру } u; \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда можно составить следующую систему линейных соотношений:

$$\begin{gathered} \sum_{p=1}^{q} \xi(j,p) = 1, \quad j = 1, 2, \ldots, n, \\ \sum_{p=1}^{q} i(j,l)\xi(j,p) \leqslant 1, \quad l = 1, 2, \ldots, m, \\ \sum_{p=1}^{q} \xi(j,p) \geqslant 0, \quad p = 1, 2, \ldots, q. \end{gathered} \tag{17.7}$$

Первая система n уравнений говорит о том, что каждая вершина окрашена только одним цветом из множества всех цветов. Система неравенств показывает, что любые две смежные вершины графа не должны быть окрашены одним цветом.

Каждому способу раскраски вершин графа g цветами (при обязательном условии разноцветности смежных вершин) соответствует система целых чисел $\xi(j,p)$, удовлетворяющая условиям (17.3), и наоборот, каждому целочисленному решению системы (17.3) соответствует один из способов раскраски. Наименьшее q, для которого система (17.7) имеет целочисленное решение, будет искомым хроматическим числом графа.

Опишем последовательный метод раскраски графа $\mathsf{G} = (X, U)$. Сначала составляется упорядоченный в порядке не возрастания степеней вершин список. Первая вершина окрашивается в цвет 1 и удаляется из списка. При просмотре списка в цвет 1 раскрашиваются и удаляются из него все вершины, не смежные с первой выбранной и между собой. Далее снова выбирается первая вершина из списка, она окрашивается в цвет 2 и удаляется. Процесс продолжается, пока не будут окрашены все вершины.

Например, пусть задан граф G, показанный на рис. 17.1.

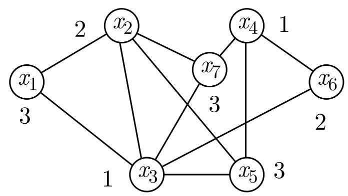


Рис. 17.1. Граф G

Построим список смежности

x_i	x_3	x_2	x_4	x_5	x_7	x_1	x_6
$\rho(x_i)$	5	4	3	3	3	2	2

Вершины x_3, x_4 окрашиваем в цвет 1 и удаляем из списка. Получаем новый список

x_i	x_2	x_5	x_7	x_1	x_6
$\rho(x_i)$	4	3	3	2	2

Вершины x_2, x_6 окрашиваем в цвет 2 и удаляем из списка:

x_i	x_5	x_7	x_1
$\rho(x_i)$	3	3	2

Вершины x_5, x_7, x_1 окрашиваем в цвет 3. На этом процесс окончен, так как раскрашены все вершины графа.

Отметим, что такого типа последовательные методы могут давать плохие результаты раскраски при быстром получении результата. Существует много модификаций последовательных алгоритмов, связанных с переупорядочиванием списков после удаления вершин, анализа вершин с одинаковой локальной степенью и выбора из них вершин, через которые проходит большее число маршрутов, и т. п. Временная сложность таких алгоритмов составляет величину $O(n) - O(n^2)$.

Важное практическое применение имеет частный вид n-раскрашиваемых графов — *двудольные* (*бихроматические*) графы. Обозначается двудольный граф $\mathsf{G}(n_1, n_2) = (X_1, X_2, U)$. Причем $X_1 \cup X_2 = X$, $X_1 \cap X_2 = \varnothing$, а ребра соединяют только подмножества X_1 и X_2 между собой. На рис. 17.2 показан пример двудольного графа $\mathsf{G} = (X_1, X_2, U)$, $|X_1| = 3 = n_1$, $|X_2| = 2 = n_2$, $X_1 = \{x_1, x_2, x_3\}$, $X_2 = \{x_4, x_5\}$.

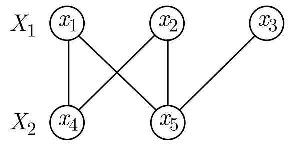


Рис. 17.2. Двудольный граф

Граф $\mathsf{K}(n_1, n_2)$ называется *полным двудольным*, если любая вершина $x_i \in X_1$ смежна с каждой вершиной $x_j \in X_2$ и наоборот ($i = 1, 2, \ldots, n_1$; $j = n_1 + 1, n_1 + 2, \ldots, n_1 + n_2$).

Например, при добавлении к графу $\mathsf{G}(n_1, n_2)$ (рис. 17.2) ребра (x_3, x_4) он становится полным двудольным графом. Очевидно, что в графе $\mathsf{G} = (X_1, X_2, U)$ подмножество X_1 можно раскрасить одним цветом, а подмножество X_2 — другим. Число ребер m графа $\mathsf{K}(n_1, n_2)$ определяется выражением $m = n_1 \cdot n_2$.

При разработке алгоритмов конструкторского и технологического проектирования ЭВА часто возникает необходимость определения двудольности произвольного графа схемы или выделения максимальных непересекающихся двудольных частей.

Теорема 17.1. *Граф* G *является двудольным тогда и только тогда, когда он не имеет простых циклов нечетной длины.*

Примеры решения задач

Пример 17.1. Определить цикломатическое число графа $\mathsf{G} = (X, U)$, изображенного на рис. 17.3.

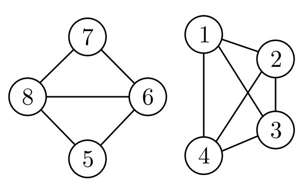


Рис. 17.3. Граф G

Решение. Из рис. 17.3 видно, что исходный граф состоит из двух компонент связности, причем $|X| = 8$; $|U| = 11$, т. е. граф G имеет 8 вершин и 11 ребер. Подставив эти числа в формулу (17.1), получим $\gamma(\mathsf{G}) = 11 - 8 + 2 = 5$. Следовательно, из исходного графа G необходимо удалить пять ребер для того, чтобы получилось дерево. Теперь необходимо определить, какие именно ребра мы должны удалить.

Как уже было сказано выше, на каждом шаге удаляется ребро, образующее хотя бы один цикл. В заданном графе присутствуют две компоненты связности. Проанализировав каждое ребро первой компоненты, приходим к выводу, что в данном случае удаление любого ребра приведет к разрыву одного из имеющихся циклов, поэтому можно удалить любое ребро. Пусть это будет ребро $(x_1 – x_2)$. Удалим шаг за шагом, например, ребра $(x_2 – x_3)$; $(x_3 – x_4)$. Как видим, в первой компоненте не осталось больше ни одного цикла. Теперь рассмотрим вторую компоненту. Здесь можем удалить, к примеру, ребра $(x_5 – x_6)$; $(x_6 – x_7)$. Мы удалили из исходного графа пять ребер. В результате получили лес, состоящий из двух деревьев, изображенных на рис. 17.4.

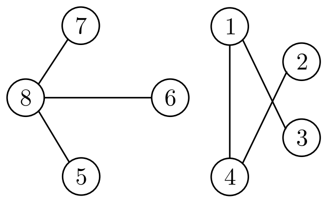


Рис. 17.4. Лес, состоящий из двух покрывающих деревьев на основе графа **G**

Пример 17.2. Определить хроматическое число и раскраску графа, изображенного на рис. 17.5.

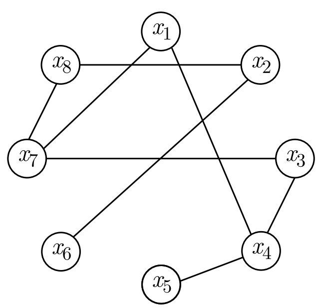


Рис. 17.5. Граф **G**

Решение. Для решения поставленной задачи воспользуемся последовательным алгоритмом раскраски графа.

1. Подсчитываем локальные степени вершин графа **G** и составляем список вершин в порядке невозрастания их локальных степеней:

$$\begin{array}{c|c|c|c|c|c|c|c|c|} \rho(X_i) & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 1 \\ \hline X_i & 4 & 7 & 8 & 1 & 2 & 3 & 5 & 6 \end{array}.$$

2. Выбираем из списка вершину x_4: $\Pi_1 = \{x_4\}$.

3. Просматриваем список на предмет нахождения несмежных вершин. Выбираем ближайшую несмежную вершину. Это вершина x_7: $\Pi_1 = \{x_4, x_7\}$.

4. Продолжаем процесс просмотра. Выбираем вершину x_2 — несмежную вершинам x_4 и x_7, включенным в подмножество Π_1. Теперь $\Pi_1 = \{x_4, x_7, x_2\}$.

5. Просматриваем список дальше, и так как в списке больше нет вершин, несмежных вершинам подмножества Π_1, окрашиваем вершины этого подмножества в первый цвет и удаляем их из списка. После удаления список примет вид

$\rho(X_i)$	3	2	2	2	1
X_i	8	1	3	5	6

6. Выбираем вершину x_8: $\Pi_2 = \{x_8\}$.

7. Выбираем вершину x_1: $\Pi_2 = \{x_8, x_1\}$.

8. Выбираем вершину x_3: $\Pi_2 = \{x_8, x_1, x_3\}$.

9. Выбираем вершину x_6: $\Pi_2 = \{x_8, x_1, x_3, x_6\}$.

10. В списке нет больше вершин, несмежных вершинам подмножества Π_2. Окрашиваем вершины этого подмножества во второй цвет и удаляем их из списка.

11. В списке остается только одна вершина — x_5. Эта вершина окрашивается в третий цвет.

Таким образом, для раскраски графа нам потребовались три краски, следовательно, хроматическое число: $\chi(\mathsf{G}) = 3$.

Пример 17.3. Привести пример полного двудольного графа $\mathsf{G} = (X_1, X_2, U)$, при условии, что $|X_1| = 3$, а $|X_2| = 4$.

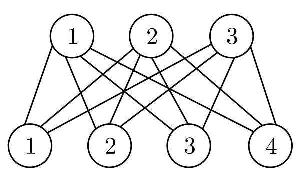


Рис. 17.6. Полный двудольный граф

Ответ. Пример полного двудольного графа, соответствующего исходным условиям, показан на рис. 17.6.

Вопросы для самоконтроля

1. Что такое цикломатическое число графа? Что показывает цикломатическое число графа?

2. Каким образом определяется цикломатическое число графа?

3. Что такое коцикломатическое число графа?

4. Дайте определение фундаментального цикла.

5. Каким образом можно построить матрицу фундаментальных циклов?

6. Что такое хроматическое число графа?

7. Какие оценки хроматического числа вы знаете? Какие методы нахождения хроматического числа обычно используются при решении практических задач?

8. Что такое раскраска вершин графа? Каким образом производится раскраска вершин графа?

9. Какой граф называется двудольным?

10. Дайте определение полного двудольного графа.

Задания для самостоятельной работы

1. Для произвольного $\mathsf{G} = (X, U)$ графа определить цикломатическое число.

2. Для произвольного графа $\mathsf{G} = (X, U)$ определить цикломатическое число и построить матрицу фундаментальных циклов.

3. Для произвольного $\mathsf{G} = (X, U)$ графа определить хроматическое число.

4. Опишите идею последовательного метода раскраски.

5. Для произвольного графа $\mathsf{G} = (X, U)$ построить раскраску последовательным методом и определить хроматическое число.

6. Задать произвольный граф G, который являлся бы двудольным.

7. Постройте структурную схему алгоритма раскраски графа.

8. Постройте структурную схему алгоритма определения двудольного графа.

9. Постройте псевдокод алгоритма раскраски графа.

10. Постройте псевдокод алгоритма определения двудольного графа.

Цикломатическое и хроматическое числа — это основные инварианты графа, которые не меняются при изменении его топологии.

17.2. Числа внутренней и внешней устойчивости графа

Рассмотрим теперь *число внутренней устойчивости*. Если две любые вершины подмножества X' графа $\mathsf{G} = (X, U)$, $X' \subseteq X$ не смежны, то оно называется *внутренне устойчивым*. Подмножество $\psi_i \subseteq X$ графа $\mathsf{G} = (X, U)$ называется *максимальным внутренне устойчивым подмножеством* (МВУП) или *независимым* (НП), если добавление к нему любой вершины $x_i \in X$ делает его не внутренне устойчивым. Подмножество ψ_i будет независимым, если

$$\forall x_i \in \psi_i (\Gamma x_i \cap \psi_i = \varnothing). \tag{17.8}$$

Независимые подмножества различаются по числу входящих в них элементов. В произвольном графе G можно выделить семейство всех НП вида $\psi = \{\psi_1, \psi_2, \ldots, \psi_s\}$. Независимые подмножества, содержащие наибольшее число элементов, называются *предельными*. Тогда *число внутренней устойчивости* $\eta(\mathsf{G})$ определяется мощностью предельного НП

$$\eta(\mathsf{G}) = |\max \psi_i|, \quad \psi_i \in \psi. \tag{17.9}$$

Число внутренней устойчивости $\eta(\mathsf{G})$ можно связать с хроматическим числом $\chi(\mathsf{G})$ следующим образом: $\eta(\mathsf{G})\chi(\mathsf{G}) \geqslant n$.

Рассмотрим вопросы построения семейства независимых подмножеств на основе матрицы смежности или ее списка. Данный алгоритм описан А. Кофманом на основе идей, предложенных К. Магу. Идея алгоритма построения семейства $\psi = \{\psi_1, \psi_2, \ldots, \psi_l\}$ заключается в следующем. В списке смежности графа G определяется строка, имеющая наибольшее число элементов. Если таких строк несколько, то выбирается любая. Для определенной строки i записывается элемент $C_i = (x_i + x_k \times x_l \times \ldots \times x_q)$, (где знак сложения «$x_i + x_k$» подразумевает операцию дизъюнкции, а знак умножения «$x_k \times x_l$» или «$x_k x_l$» — операцию конъюнкции). Здесь x_i — вершина графа, соответствующая выделенной строке i, а $x_k, x_l, \ldots, x_q$ — вершины, смежные с x_i. Далее в списке смежности исключается строка i, а также элементы с индексом i из всех оставшихся строк списка. Указанная операция соответствует выделению из графа G звездного подграфа с вершиной x_i. В списке снова определяется строка j с наибольшим числом элементов и записывается элемент $C_j = (x_j + x_p \times x_r \times \ldots \times x_v)$. Далее процесс выполняется аналогично до тех пор, пока в списке для оставшихся строк не будет содержаться ни одного элемента. Затем составляется произведение $\Pi = C_i \times C_j \times \ldots \times C_k$, в котором раскрываются скобки, и в полученной сумме выполняется минимизация с учетом выражений

$$x_i^n = x_i, x_i + x_i + \ldots + x_i = x_i; \quad x_i + 1 = 1; \quad x_i + x_i x_j = x_i. \tag{17.10}$$

Для выделения семейства ψ необходимо в выражении, полученном после минимизации, для каждого элемента суммы найти не включенные в него вершины графа, которые образуют максимальные НП.

Методика определения МНП используется для раскраски вершин графа **G**.

Из рассмотренного алгоритма следует, что раскраску можно интерпретировать как разбиение всех вершин графа на независимые подмножества. Если бы граф имел систему МНП, состоящих из различных вершин, то раскраска соответствовала бы покрытию вершин **G** системой МНП. Но одни и те же вершины графа могут принадлежать различным МНП. В этой связи существует конечное множество различных допустимых раскрасок, так как вершину, принадлежащую разным МНП, можно окрасить в разные цвета.

Если получено семейство МНП, то можно построить матрицу $\mathrm{M} = ||m_{i,j}||_{n\times w}$, где

$$m_{i,j} = \begin{cases} 1, & \text{если } x_i \in \psi_j; \\ 0 & \text{в противном случае,} \end{cases}$$

w — число МНП в семействе ψ. Тогда задачу раскраски сводят к нахождению наименьшего числа строк, покрывающих все ее столбцы единицами.

Например, для графа **G** (рис. 17.1) матрицу M можно записать следующим образом:

$$\mathrm{M} = \begin{array}{c} \\ \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \end{array} \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \left|\begin{matrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{matrix}\right| \end{array}.$$

Проанализировав данную матрицу, можно отметить, что минимальное число строк, покрывающих все столбцы матрицы, равно трем. Это строки ψ_1 (покрываются столбцы x_1, x_5, x_6

и x_7), ψ_5 (покрываются столбцы x_3, x_4), а также либо строка ψ_3 (покрываются столбцы x_2, x_6), либо строка ψ_4 (покрываются столбцы x_2, x_4).

Таким образом, можно выполнить следующую раскраску графа: вершины x_1, x_5, x_6 и x_7 — первый цвет; вершины x_3, x_4 — второй цвет; вершина x_2 — третий цвет.

Рассмотрим алгоритм определения внешне устойчивого подмножества.

Пусть исследуемый объект задан в виде неориентированного графа $\mathsf{G} = (X, \Gamma x)$, который можно рассматривать как пару, образованную множеством X и многозначным отображением Γx множества X в себя. $X = \{x_1, x_2, \ldots, x_n\}$ — множество вершин графа, $\Gamma x_i = \{x_k, x_e, \ldots, x_p\}$ определяет множество вершин $x_k, x_e, \ldots, x_p$, смежных с вершиной x_i, другими словами, определяет ребра графа (x_i, x_k), $(x_i, x_e), \ldots, (x_i, x_p)$; $\bigcup\limits_{i=1}^{n} \Gamma x_i = \Gamma x$.

Подмножество вершин $M_i \subseteq X$ графа G называется внешне устойчивым, если

$$(\forall x \notin M_i)(\Gamma x \cap M_i \neq \varnothing). \tag{17.11}$$

Внешне устойчивое подмножество (ВУП) называется *минимальным*, если удаление из него произвольной вершины делает его внешне неустойчивым. В каждом графе можно выделить семейство $M = \{M_1, M_2, \ldots, M_i\}$ минимально внешне устойчивых подмножеств (МВУП). Элементы семейства M отличаются друг от друга по виду и количеству входящих в них элементов. Мощность МВУП, содержащего наименьшее число вершин, называется *числом внешней устойчивости* ($\beta(\mathsf{G})$).

Рассмотрим алгоритм получения минимального внешне устойчивого подмножества и наметим путь получения семейства МВУП.

Для задания графов будем использовать структурные числа. Система А, не содержащая одинаковых столбцов a_i и одинаковых элементов $a(q, i)$ в каждом столбце, называется структурным числом.

$$\mathrm{A} = \{a_1, a_2, \ldots, a_n\}, \quad a_i \neq a_j,$$
$$a_q = \{a(1, q), a(2, q), \ldots, a(m, q)\}; \quad a(i, q) \neq a(j, q).$$

Столбцам структурного числа (СЧ) можно поставить в соответствие покрывающие деревья графа, а элементам столбцов СЧ — номера соответствующих деревьев. СЧ А имеет геометрическое изображение в виде графа $\mathsf{G} = (X, \Gamma x)$, $X = \{x_1, x_2, \ldots, x_n\}$, если существует разложение А на простые сомножители $P_1, P_2, \ldots, P_{n-1}$ и произвольный элемент $a_i \in P_i$ встречается не более чем в двух сомножителях. Если каждый P_i записать в виде номеров ребер, смежных вершине x_i, то СЧ графа **G** примет вид $\mathrm{A} = P_1, P_2, \ldots, P_{n-1}$. Произведение элементов P_i позволит получить СЧ в виде матрицы, где каждый столбец соответствует номерам ребер покрывающего дерева.

Граф **G** будем задавать в виде преобразованного структурного числа (ПСЧ).

$$\mathrm{A} = Q_1, Q_2, \ldots, Q_n; \quad Q_i = [i][j_1, j_2, \ldots, j_l] = [i][Q'_i].$$

Сомножитель Q_i соответствует вершине x_i графа **G**; i — индекс x_i; $Q'_i = j_1, j_2, \ldots, j_l$ — последовательность индексов вершин, смежных x_i.

Для графа **G** (рис. 17.7)

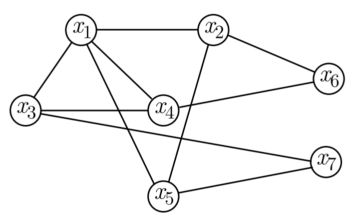


Рис. 17.7. Граф **G**

преобразованное структурное число запишется следующим образом:

$$\mathrm{A} = \begin{vmatrix} [1] & [2,3,4,5] \\ [2] & [1,5,6] \\ [3] & [1,4,7] \\ [4] & [1,3,6] \\ [5] & [1,2,7] \\ [6] & [2,4] \\ [7] & [3,5] \end{vmatrix}.$$

Очевидно, что такая запись идентична той, что получилась бы, если бы мы использовали списочное задание графа. Таким образом, преобразованное структурное число A (ПСЧ) — это не что иное, как список смежности рассматриваемого графа. При этом для задания графа достаточно $n-1$ строк вида Q_i. В связи с вышесказанным в дальнейшем при решении практических задач будем считать равнозначным обозначение A (т. е. преобразованное структурное число) и R_L (т. е. список смежности).

Введем понятие расширенной алгебраической производной ПСЧ, которую запишем следующим образом: $\dfrac{\partial_p \mathrm{A}}{\partial Q_i} = \mathrm{A}'$ — исключена строка Q_i, а также элементы i и $j_1, j_2, \ldots, j_l$ из Q'_k частей оставшихся строк A.

Указанная операция соответствует удалению из графа G звездного подграфа с вершиной x_i, а также ребер, соединяющих вершины, смежные x_i. Вершины, удаленные после взятия производной, будем записывать так: $X' = \{i, j_1, j_2, \ldots, j_l\}$. Причем $X' \subseteq X$.

На основе введенной операции взятия расширенной алгебраической производной сформулируем алгоритм построения минимального внешне устойчивого подмножества $M_i \subseteq M$. Основная идея алгоритма заключается в следующем. В ПСЧ определяется строка Q_i, у которой $|Q'_i|$ максимальна. Если таких строк несколько, то выбирается любая. Далее определяется расширенная алгебраическая производная $\dfrac{\partial_p \mathrm{A}}{\partial Q_i} = \mathrm{A}'$ и записывается подмножество $X' = \{x_i, x_k, x_l, \ldots, x_q\}$ (вершины x_k, $x_l, \ldots, x_q$ соответствуют элементам Q'_i). Затем производится сравнение X с X'. Если $X' = X$, то вершина x_i образует внешне устойчивое подмножество M_1. В противном случае в ПСЧ A' снова определяется строка Q_j, у которой $|Q'_j|$ максимальна. Если таких строк несколько, то выбирается строка Q_s, которая имеет максимальное число элементов, не вошедших в подмножество X' на предыдущем шаге. Находится $\dfrac{\partial_p \mathrm{A}}{\partial Q_i}$ и записывается подмножество X''. Если $X'' = X$, то вершины x_i и x_j образуют внешне устойчивое подмножество $M_1 = \{x_i, x_j\}$. Если нет, то процесс повторяется аналогично, пока на некотором l шаге $X^{(l)} = X$. Для построения семейства M необходимо аналогичные операции над ПСЧ выполнять последовательно с каждой строкой с просмотром всех ответвлений.

Запишем алгоритм построения минимального внешне устойчивого подмножества множества вершин G.

1°. Запись по графу G преобразованного структурного числа A.

2°. Выбор в A строки Q_i с $|Q'_i| = \max$.

3°. Определение $\dfrac{\partial_p \mathrm{A}}{\partial Q_i} = \mathrm{A}'$ и получение X'.

4°. Сравнение X' с X. Если $|X'| = |X|$, то $M_1 = \{x_i\}$ и переход к 6°. Иначе к 5°.

5°. Выбор в ПСЧ строки j с максимальным Q'_j, $j = i$ и переход к 3°.

6°. Выделено внешне устойчивое подмножество. Конец работы алгоритма.

Работу алгоритма покажем на примере графа **G** (рис. 17.7). В ранее записанном ПСЧ A выделим строку Q_1, так как часть Q'_1 содержит наибольшее число элементов. Определим $\dfrac{\partial_p \mathrm{A}}{\partial Q_i}$ и получим новое ПСЧ:

$$\mathrm{A}' = \begin{vmatrix} [2] & [6] \\ [3] & [7] \\ [4] & [6] \\ [5] & [7] \\ [6] & [\varnothing] \\ [7] & [\varnothing] \end{vmatrix}.$$

Выделенное подмножество — $X' = \{x_1, x_2, x_3, x_4, x_5\}$. Граф, соответствующий A', показан на рис. 17.8.

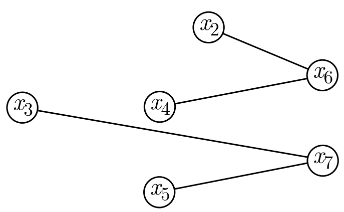


Рис. 17.8. Преобразованный граф **G**

Мощность X' не равна мощности X. Следовательно, вершину x_1 относим в M_1 и продолжаем процесс получения внешне устойчивого подмножества. На втором шаге определяем строки Q_2, Q_3, Q_4, Q_5. Так как мощность этих строк одинакова, мы можем выбрать любую строку, например, Q_2. После этого вычисляем $\dfrac{\partial_p \mathrm{A}'}{\partial Q_i}$ и получаем:

$$\mathrm{A}'' = \begin{vmatrix} [3] & [7] \\ [4] & [\varnothing] \\ [5] & [7] \\ [6] & [\varnothing] \\ [7] & [\varnothing] \end{vmatrix}.$$

Теперь $X'' = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. Величина $X'' \neq X$, следовательно, вершину x_2 относим к M_1 и алгоритм продолжает работу. Рассматриваем строки Q_3 и Q_5. Выбираем любую из них, например Q_3, и определяем значение $\dfrac{\partial_p \mathrm{A}''}{\partial Q_i}$. В результате получим

$$\mathrm{A}'' = \begin{vmatrix} [4] & [\varnothing] \\ [5] & [\varnothing] \\ [6] & [\varnothing] \\ [7] & [\varnothing] \end{vmatrix}.$$

$X''' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} = X$. Следовательно, получили внешне устойчивое подмножество $M = \{x_1, x_2, x_3\}$.

Теорема 17.2. *Подмножество M_i, полученное по алгоритму, является минимальным внешне устойчивым подмножеством.*

Доказательство. Покажем, что M_i есть внешне устойчивое подмножество. Предположим, что M_i не является внешне устойчивым, т. е. элементы подмножества вершин M_i совместно не смежны оставшимся вершинам графа. Это говорит о том, что на конечном l шаге работы алгоритма $\mathrm{X}(l)$ $X^{(l)} \neq X$, что противоречит условию алгоритма. Так как алгоритм заканчивает работу, когда $X^{(l)} \neq X$, то M_i является внешне устойчивым подмножеством.

Покажем теперь, что $M_i \subset M$ — минимальное внешне устойчивое подмножество. Предположим, что существует M_j такое, что $M_j \subset M_i$, т. е. существует, по крайней мере, один элемент $x_b \in X$, после удаления которого из M_i подмножество M_j будет внешне устойчивым. Другими словами, алгоритм после получения внешне устойчивого подмножества продолжит работу и выделит $M_i \supset M_j$, что противоречит структуре алгоритма. Следовательно, M_i есть минимальное внешне устойчивое подмножество.

Для нахождения числа внешней устойчивости необходимо выделить семейство минимальных внешне устойчивых подмножеств и определить в нем наименьшее по мощности подмножество.

Существуют графы, которые содержат некоторое подмножество $X' \subseteq X$ вершин, причем это подмножество является одновременно внутренним и внешне устойчивым. Подмножества такого типа называются *ядрами* графа.

Теорема 17.3. *Если граф $\mathsf{G} = (X, U)$ имеет ядро, то его числа устойчивости удовлетворяют условию*

$$n(\mathsf{G}) \geqslant \beta(\mathsf{G}). \tag{17.12}$$

Например, в графе G (рис. 17.7) подмножество вершин $\{x_3, x_5, x_6\}$ является ядром, так как оно одновременно и внешне и внутренне устойчиво. Причем $\eta(\mathsf{G}) = 3 = \beta(\mathsf{G})$.

Примеры решения задач

Пример 17.4. Выделить семейство независимых подмножеств для графа, заданного списком смежности R_L:

$$\mathrm{R}_L = \left.\begin{array}{c|l} 1 & 2,4,5 \\ 2 & 1,3,6,7 \\ 3 & 2,5,6,7,8 \\ 4 & 1,5,7,8 \\ 5 & 1,3,4 \\ 6 & 2,3 \\ 7 & 2,3,4 \\ 8 & 3,4 \end{array}\right| .$$

Решение. Для решения поставленной задачи воспользуемся вышеописанным алгоритмом.

1. Выбираем из списка смежности R_L строку с наибольшим числом элементов. В данном случае это строка с индексом 3.

Теперь можем записать следующее выражение:

$$C_3 = (x_3 + x_2\, x_5\, x_6\, x_7\, x_8).$$

Удаляем из списка 3-ю строку и из всех остальных строк — элементы с индексом 3. В результате получим модифицированный список смежности R_{L_1}.

$$\mathrm{R}_{L_1} = \left.\begin{array}{c|l} 1 & 2,4,5 \\ 2 & 1,6,7 \\ 4 & 1,5,7,8 \\ 5 & 1,4 \\ 6 & 2 \\ 7 & 2,4 \\ 8 & 4 \end{array}\right| .$$

2. В списке R_{L_1} выбираем строку с индексом 4. Записываем выражение: $C_4 = (x_4 + x_1\, x_5\, x_7\, x_8)$. После удаления четвертой строки и элементов с соответствующими ей индексами получим

список смежности R_{L_2}:

$$\mathrm{R}_{L_2} = \left.\begin{array}{r|l} 1 & 2,5 \\ 2 & 1,6,7 \\ 5 & 1 \\ 6 & 2 \\ 7 & 2 \\ 8 & 0 \end{array}\right| .$$

3. Выбираем строку 2 и записываем выражение $C_2 = (x_2 + x_1\, x_6\, x_7)$. После удаления второй строки и элементов с соответствующими ей индексами получим список R_{L_3}:

$$\mathrm{R}_{L_3} = \left.\begin{array}{r|l} 1 & 5 \\ 5 & 1 \\ 6 & 0 \\ 7 & 0 \\ 8 & 0 \end{array}\right| .$$

4. Из двух оставшихся ненулевых строк выберем строку 1 и запишем $C_1 = (x_1 + x_5)$.

5. Запишем произведение полученных выражений:

$$\Pi = (x_1 + x_5)(x_2 + x_1\, x_6\, x_7)(x_3 + x_2\, x_5\, x_6\, x_7\, x_8)(x_4 + x_1\, x_5\, x_7\, x_8).$$

В результате последовательного перемножения и минимизации на основе тождеств алгебры логики получим выражение:

$$\Pi = x_1\, x_2\, x_3\, x_4 + x_1\, x_2\, x_3\, x_5\, x_7\, x_8 + x_1\, x_3\, x_4\, x_6\, x_7 + \\ + x_2\, x_3\, x_4\, x_5 + x_1\, x_3\, x_5\, x_6\, x_7\, x_8 + x_1\, x_2\, x_5\, x_6\, x_7\, x_8.$$

6. Обозначим каждую конъюнкцию в произведении следующим образом: $K_1, K_2, K_3, \ldots, K_6$. Теперь «проинвертируем» каждый элемент полученного произведения, т. е. определим элементы, не входящие в каждую из конъюнкций:

$$\overline{\Pi} = \overline{K}_1 + \overline{K}_2 + \overline{K}_3 + \overline{K}_4 + \overline{K}_5 + \overline{K}_6 = \\ = x_5\, x_6\, x_7\, x_8 + x_4\, x_6 + x_2\, x_5\, x_8 + x_1\, x_6\, x_7\, x_8 + x_2\, x_4 + x_3\, x_4.$$

Каждая из полученных конъюнкций соответствует независимому подмножеству. Таким образом, мы получили семейство независимых подмножеств:

$$\psi = [\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6],$$

где $\psi_1 = \{x_5, x_6, x_7, x_8\}$; $\psi_2 = \{x_4, x_6\}$; $\psi_3 = \{x_2, x_5, x_8\}$; $\psi_4 = \{x_1, x_6, x_7, x_8\}$; $\psi_5 = \{x_2, x_4\}$; $\psi_6 = \{x_3, x_4\}$.

Поскольку мощность максимального независимого подмножества равна четырем, то число внутренней устойчивости заданного графа $\eta(\mathsf{G})$ также равно 4.

Данный алгоритм можно использовать также для решения задачи раскраски графа. Для этого необходимо для каждой вершины определить независимые подмножества, в которые она не входит.

Пример 17.5. Выполнить раскраску графа, заданного в примере 17.4, на основе алгоритма Кофмана. Выпишем для каждой вершины графа те конъюнкции, в которые эти вершины не входят.

$$x_1 \notin K_4; \quad x_2 \notin K_3, K_5; \quad x_3 \notin K_6; \quad x_4 \notin K_2, K_5, K_6;$$
$$x_5 \notin K_1, K_3; \quad x_6 \notin K_1, K_2, K_4; \quad x_7 \notin K_1, K_4; \quad x_8 \notin K_1, K_3, K_4.$$

Теперь запишем полученное выражение в виде произведения этих конъюнкций:

$$\Pi' = K_4 \times (K_3 + K_5) \times K_6 \times (K_2 + K_5 + K_6) \times (K_4 + K_3) \times \times (K_1 + K_2 + K_4) \times (K_1 + K_4) \times (K_1 + K_3 + K_4).$$

После перемножения элементов данного выражения и его минимизации получим следующее выражение:

$$\Pi' = K_1 K_2 K_3 K_4 K_6 + K_1 K_4 K_5 K_6.$$

Число элементов K_i в каждой конъюнкции соответствует количеству цветов, которые необходимы для раскраски графа, а каждый K_i определяет подмножество вершин, которые можно раскрасить одним цветом.

Таким образом, из полученного выражения следует, что данный граф можно раскрасить двумя способами, причем для этого потребуется либо четыре краски, либо пять.

Рассмотрим, например, конъюнкцию, содержащую четыре элемента. Это элементы K_1, K_4, K_5, K_6. Как мы уже определили ранее, конъюнкции K_1 соответствуют элементы $\{x_5, x_6, x_7, x_8\}$. Следовательно, эти вершины мы окрашиваем в первый цвет.

Для определения вершин, которые будут окрашены во второй цвет, необходимо определить разность:

$$K_4 \backslash K_1 = \{x_1, x_6, x_7, x_8\} \backslash \{x_5, x_6, x_7, x_8\} = \{x_1\}.$$

Окрашиваем вершину x_1 во второй цвет.

Действуя аналогичным образом, мы определяем, что вершины x_2, x_4 окрашиваются в третий цвет, а вершина x_3 — в четвертый.

Таким образом, задача раскраски графа решена, граф раскрашен четырьмя красками.

Задачу раскраски графа можно решить также решением *задачи нахождения минимального покрытия* независимыми подмножествами всех элементов исходного набора, как показано в следующем примере.

Пример 17.6. Раскрасить граф из примера 17.4 через решение задачи покрытия.

Решение. Составляем таблицу, в которой строки соответствуют независимым подмножествам, а столбцы — вершинам графа.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
ψ_1	0	0	0	0	1	1	1	1
ψ_2	0	0	0	1	0	1	0	0
ψ_3	0	1	0	0	1	0	0	1
ψ_4	1	0	0	0	0	1	1	1
ψ_5	0	1	0	1	0	0	0	0
ψ_6	0	0	1	1	0	0	0	0

.

Выбираем первую строку этой таблицы. Она покрывает элементы x_5, x_6, x_7, x_8. Вторая строка перекрывает вершину x_4. Третья строка — вершину x_2. Четвертая — вершину x_1. И, наконец, шестая — вершину x_3. Таким образом, мы получили раскраску исходного графа пятью цветами.

1 вариант: 1-я краска — x_5, x_6, x_7, x_8; 2-я краска — x_4; 3-я — x_2; 4-я — x_1; 5-я — x_3.

Очевидно, что существует и второй, более оптимальный вариант раскраски данного графа.

2 вариант: 1-я краска — x_1, x_6, x_7, x_8 (4-я строка таблицы); 2-я краска — x_3, x_4 (6-я строка таблицы); 3-я — x_2, x_5 (3-я строка таблицы).

Вопросы для самоконтроля

1. Что такое число внутренней устойчивости графа?
2. Что такое независимое подмножество вершин графа?
3. Каким образом выделяется независимое подмножество вершин графа?

4. Какое подмножество вершин графа называется предельным?

5. Приведите механизм определения внутренне устойчивых подмножеств.

6. Опишите идею алгоритма выделения независимых подмножеств в графе.

7. Что такое число внешней устойчивости?

8. Приведите механизм определения внешне устойчивых подмножеств.

9. Что такое минимальное внешне устойчивое подмножество?

10. Опишите основную идею алгоритма выделения минимальных внешне устойчивых подмножеств.

Задания для самостоятельной работы

1. Для произвольного $\mathsf{G} = (X, U)$ графа определить число внутренней устойчивости.

2. Определить число внутренней устойчивости произвольного графа $\mathsf{G} = (X, U)$ и построить семейство независимых подмножеств.

3. Для произвольного графа $\mathsf{G} = (X, U)$ построить раскраску на основе алгоритма выделения независимых подмножеств.

4. Для произвольного графа $\mathsf{G} = (X, U)$ определить число внешней устойчивости и применить алгоритм выделения минимальных внешне устойчивых подмножеств.

5. Постройте структурную схему алгоритма нахождения семейства независимых подмножеств.

6. Запишите псевдокод алгоритма построения семейства независимых подмножеств.

7. Постройте структурную схему алгоритма определения семейства доминирующих подмножеств в графе.

8. Постройте псевдокод алгоритма определения семейства доминирующих подмножеств в графе.

9. Запишите псевдокод алгоритма раскраски графа путем нахождения минимального покрытия независимыми подмножествами.

10. Постройте структурную схему алгоритма раскраски графа путем нахождения минимального покрытия независимыми подмножествами.

Числа внутренней и внешней устойчивости относятся к инвариантам графа. Они позволяют определить специальные группы вершин в графах.

17.3. Планарность графов

Методы теории графов позволяют решать многие технические и теоретические задачи, такие как расчет электронных схем, задачи сетевого планирования, микроэлектроники, искусственного интеллекта и др. Это объясняется тем, что граф, сохраняя всю наглядность и содержательность объекта, позволяет строить формальные модели живой и неживой природы, которые легко обрабатываются на ЭВМ. Данный раздел посвящен обзору важнейшего раздела теории графов — планарности. Проблемы планарности графов связаны с возможностью плоского (без пересечений ребер) изображения графа, минимизации пересечений ребер, разбиения непланарного графа на плоские и планарные подграфы и суграфы.

17.3.1. Плоские и планарные графы. Изучение планарных графов начал Эйлер в его исследованиях полиэдров. С каждым полиэдром связан граф, состоящий из точек и линий полиэдра. Говорят, что граф укладывается на поверхность S, если его можно нарисовать на S так, что никакие два его ребра не пересекаются. Граф называется *планарным*, если его можно уложить на плоскости без пересечений. *Плоский* граф — это граф, уже расположенный на плоскости без пересечений.

Области, определяемые плоским графом (ПЛГ), называются его *внутренними гранями*, или просто *гранями*. Если границей грани ПЛГ является простой цикл, то иногда под гранью понимают этот цикл. ПЛГ (рис. 17.9) имеет две внутренние грани: f_1, f_2 и одну внешнюю — f_3. Неограниченная область называется *внешней гранью*.

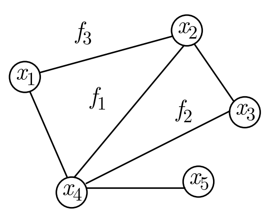


Рис. 17.9. Плоский граф с тремя гранями

Плоской картой называется связный ПЛГ вместе со всеми его гранями. Уравнение (теорема) Эйлера для плоской карты с n-вершинами, m-ребрами и f-гранями запишется как

Теорема 17.4.

$$n - m + f = 2. \tag{17.13}$$

Следствие 17.1. *Если* G — *плоская* (n, m)*-карта, в которой каждая грань является* C*-циклом* (*т. е. циклом длиной* C, *состоящим из* C *ребер*), *то*

$$m = \frac{C(n-2)}{C-2} \tag{17.14}$$

и

$$m = Cf/2. \tag{17.15}$$

Максимальным планарным графом (МПГ) называется граф, который при добавлении любого ребра перестает быть планарным.

Следствие 17.2. *Если* G — *МПГ, то каждая его грань является треугольником и*

$$m = 3n - 6. \tag{17.16}$$

Если G — *плоский граф, у которого любая грань есть 4-цикл, то*

$$m = 2n - 4. \tag{17.17}$$

Так как наибольшим числом ребер в ПЛГ обладает граф, у которого каждая грань есть треугольник, то получаем необходимое условие планарности графа в терминах числа ребер.

Следствие 17.3. *Графы* K_5 *и* $\mathsf{K}_{3,3}$ *не являются планарными.*

На рис. 17.10, *а* показан граф K_5, а на рис. 17.10, *б* — $\mathsf{K}_{3,3}$.

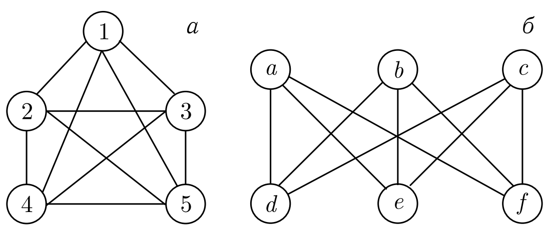


Рис. 17.10. Полный граф K_5 (*а*); полный двудольный граф $\mathsf{K}_{3,3}$ (*б*)

Доказательство. K_5 есть граф с 5 вершинами и 10 ребрами, и он не может быть планарным, так как $m = 10 > 3n - 6 = 9$. Для $\mathsf{K}_{3,3}$ имеем $m = 9 > 2n - 4 = 8$, что и доказывает следствие 17.3.

Легко видеть, что каждый подграф ПЛГ планарен, а любой граф, содержащий в качестве подграфа непланарный граф (НПГ), сам не может быть планарным. Отсюда вытекает, что K_5 и $\mathsf{K}_{3,3}$ — по существу, единственные непланарные графы в том смысле, что любой НПГ содержит один из них.

Два графа *гомеоморфны* (или тождественны с точностью до вершины степени 2), если они могут быть получены из одного и того же графа включением (добавлением) в его ребра новых вершин степени 2.

Л.С. Понтрягин в 1927 г. доказал, но не опубликовал критерий планарности.

К. Куратовский в 1930 г. независимо от Л.С. Понтрягина получил тот же результат.

Теорема 17.5. (Понтрягин–Куратовский). *Граф планарен тогда и только тогда, когда он не содержит подграфа, гомеоморфного графам* K_5 *или* $\mathsf{K}_{3,3}$.

Граф G называется стягиваемым к графу H, если H можно получить из G с помощью некоторой последовательности элементарных стягиваний. Например, на рис. 17.11 показано, как граф Петерсона стягивается в K_5. Здесь произошло стягивание в новую вершину w_i любого из пяти ребер (u_i, v_i), соединяющего пятиугольник с пентаграммой (звездой), т. е. K_5 получается из графа Петерсона стягиванием 5 ребер внутреннего цикла с внешним (рис. 17.12).

Теорема 17.6. (Харари–Татт). *Граф планарен тогда и только тогда, когда он не содержит подграфов, стягиваемых в* K_5 *или* $\mathsf{K}_{3,3}$.

Уитни связал планарность графов с существованием двойственных графов (ДГ). Для данного плоского графа G его геометрический ДГ, т. е. ГДГ G^* строится так:

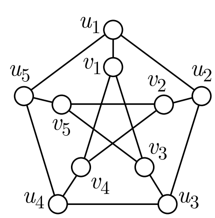


Рис. 17.11. Граф Петерсона

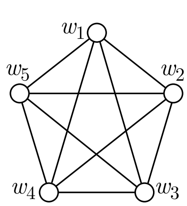


Рис. 17.12. Граф K_5, полученный из графа Петерсона

• поместим в каждую область G, включая внешнюю, по одной вершине G^*;

• если две области имеют общее ребро u, соединим помещенные в них вершины ребром u^*, пересекающим только u.

Данный процесс показан, например, для графа G (рис. 17.13).

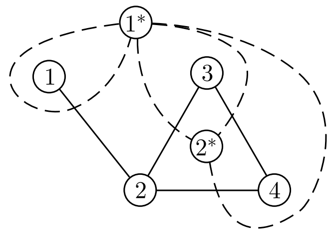


Рис. 17.13. Пример построения ГДГ

Здесь ребра ГДГ G^* — штриховые линии. G^* имеет петлю тогда, когда в G есть концевая вершина в примере с рис. 17.10, *в*. G^* имеет кратные ребра, когда две области G содержат по крайней мере два общих ребра.

Таким образом, двусвязный плоский граф имеет всегда в качестве двойственного или граф, или мультиграф. ГДГ плоского графа G также плоский, а ГДГ к ГДГ — это первоначальный граф G.

Толщина графа — наименьшее число планарных графов, объединение которых есть G. Крупность (шероховатость, зернистость) графа — наибольшее число непланарных подграфов в G, не пересекающихся по ребрам (т. е. подграфов, не имеющих общих ребер). *Число скрещиваний* (*пересечений*) графа — наименьшее число пересечений ребер, которое должно быть при расположении G на плоскости.

17.3.2. Эвристики для определения планарности. Все существующие методы определения планарности разбивают на два класса. В первый класс входит методы, основанные на проверке критериев:

а) Понтрягина–Куратовского;

б) Харари–Татта;

в) Уитни;

г) существования абстрактно-двойственного графа;

д) Мак-Лейна.

Во второй класс входят эвристические методы, в той или иной степени использующие критерии первого класса, — это:

а) алгоритм Бадера;

б) циклические методы;

в) матричные методы;

г) комбинированные методы.

Так как неорграф планарен тогда и только тогда, когда все его связные компоненты планарны, то достаточно рассматривать лишь связные неорграфы. Очевидно, что неорграф планарен тогда и только тогда, когда все его двусвязные компоненты планарны. Поэтому если неорграф (далее граф) является разделимым, можно разложить его на двусвязные компоненты и рассматривать их отдельно. Поскольку кратные ребра и петли всегда можно добавить к графу или удалить из него без нарушения свойств планарности, достаточно рассматривать только простые графы.

Итак, для определения планарности будем предполагать, что исходный граф неориентированный, простой и двухсвязный. Согласно теореме Понтрягина–Куратовского неплоский граф имеет по крайней мере 9 ребер.

Сущность алгоритма заключается в следующем. Записывается матрица смежности, анализируя которую определяются пресечения ребер графа. Затем строится граф (матрица) пересечений. Далее определяется двудольность (бихроматичность) графа пересечения.

Теорема 17.7 (Бадер). *Если граф пересечений двудольный, то исходный граф планарный.*

Доказательство следует из того, что в двудольном графе можно выделить два подмножества несмежных вершин Y', Y'' таких, что $Y' \cup Y'' = Y$ и $Y' \cap Y'' = \varnothing$. Граф пересечений $\mathsf{G}' = (Y, V)$ для графа $\mathsf{G} = (X, U)$ определяется так, что $Y \leftrightarrow U'$, где U' — подмножество пересекающихся ребер.

Алгоритм рассмотрим на примере. Пусть дан $\mathsf{G} = (Y, V)$, имеющий известный гамильтонов цикл (ГЦ). Расположим его на плоскости (рис. 17.14). По известной методике определим пересечения ребер внутри ГЦ. Далее строится граф пересечений $\mathsf{G}' = (Y, V)$ (рис. 17.15). Теперь необходимо воспользоваться теоремой Бадера и определить, является ли граф G' двудольным или нет.

Согласно теореме Кёнига граф двудолен, если в нем нет циклов нечетной длины. Поэтому для определения двудольности графа G' можно предложить несколько основных эвристик.

Э1. Определить все циклы графа пересечений G', если среди них нет нечетных, то исходный граф G планарен.

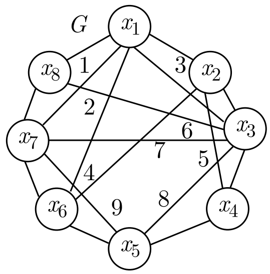


Рис. 17.14. Граф G

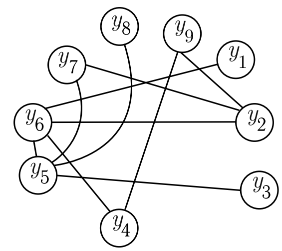


Рис. 17.15. Граф пересечений G'

Э2. Проверить, является ли граф пересечений G' деревом. Если да, то G' — двудольный, так как всякое дерево двудольный граф. Следовательно, если G' — дерево, то G — планарный граф.

Э3. Определить систему МВУП (независимых подмножеств) графа G'. Если среди семейства МВУП найдутся такие два подмножества Π_1, Π_2, что

$$\Pi_1 \cup \Pi_2 = Y, \quad \Pi_1 \cap \Pi_2 = \varnothing,$$

то граф G' — двудольный, а G — планарный.

Например, для графа G' (рис. 17.15)

$$\Pi_1 = \{y_1, y_2, y_4, y_5\}, \quad \Pi_2 = \{y_6, y_7, y_3, y_8, y_9\},$$
$$\mathsf{G}' = (Y, V), \quad \Pi_1 \cup \Pi_2 = Y, \quad \Pi_1 \cap \Pi_2 = \varnothing.$$

Граф G', представленный в виде двудольного, показан на рис. 17.16. Очевидно, что граф G — планарный (рис. 17.14). Так, ребра G, соответствующие Π_1, можно расположить без пересечений внутри ГЦ, а ребра, соответствующие Π_2, — внутри ГЦ, или наоборот. Плоское изображение графа (рис. 17.14) показано на рис. 17.17.

Основная стратегия определения планарности состоит в том, чтобы в графе G найти цикл C, разместить C на плоскости в виде простой замкнутой кривой, разложить оставшуюся часть G–C на непересекающиеся по ребрам пути и затем попытаться разместить внутри C либо целиком вне C. Если это удалось для всего графа G, то он планарен, в противном случае он не планарен. Трудность реализации алгоритма заключается в том, что при размещении пути можно выбрать либо внутренность, либо внешность C и необходимо обеспечить, чтобы неправильный выбор области размещения на ранней стадии не устранял возможности размещения последующих путей. Это может привести к неверному заключению, что планарный граф непланарен.

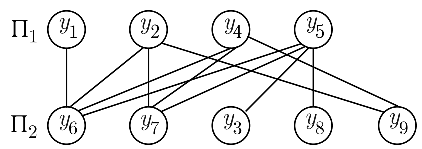


Рис. 17.16. Двудольный граф G

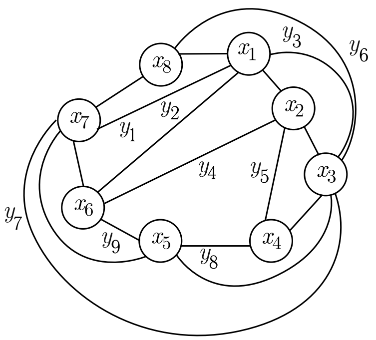


Рис. 17.17. Плоское преображение графа G

17.3.3. Минимизация пересечений ребер графов. В связи с тем, что полный граф K_n содержит максимальное число ребер, то минимальное число пересечений ребер K_n является верхней оценкой числа пересечений произвольных графов на n вершинах.

Пусть дан полный граф $\mathsf{K}_n = (X, U)$. Если его расположить на плоскости таким образом, чтобы образовалась выпуклая фигура, а вершины соединялись прямолинейными ребрами, тогда число пересечений в таких графах определится по формуле

$$P(\mathsf{K}_n) = \frac{n(n-1)(n-2)(n-3)}{24} \tag{17.18}$$

при $n \geqslant 4$. Причем в каждой точке пересекается не более двух ребер. Тогда для K_4: $P(\mathsf{K}_4) = 1$, $P(\mathsf{K}_5) = 5$, $P(\mathsf{K}_6) = 15$.

Любой планарный граф может быть расположен на плоскости таким образом, что все его ребра есть прямые линии.

Верхняя оценка числа скрещиваний (минимального числа пересечений полного графа) дана Гаем:

$$P_{\min}(\mathsf{K}_n) \leqslant \begin{cases} \dfrac{1}{64}(n-1)^2(n-3)^2, & \text{если } n \text{ нечетно,} \\ \dfrac{1}{64}(n-2)^2(n-4), & \text{если } n \text{ четно.} \end{cases} \tag{17.19}$$

Существует гипотеза Харари–Хилла. Верхняя оценка минимального числа пересечений в полном графе (формула Гая (17.19)) является точным значением.

Самый простой путь расположения полного графа для минимизации пересечений его ребер следующий. Сначала располагаются ребра K_n, образующие ГЦ в виде выпуклой фигуры, которая разбивает плоскость на две области. Все остальные ребра K_n, образующие множество $U\backslash U_{\text{г.ц}}$, проводятся во внешней и выпуклой областях путем чередования, т. е. подмножество U_1, инцидентное вершине x_1, проведем во внутренней области, подмножество U_2, инцидентное x_2, — во внешней, U_3, инцидентное x_3, — во внутренней и т. д. Например, для графа K_n, $n = 6$, получим рис. 17.18.

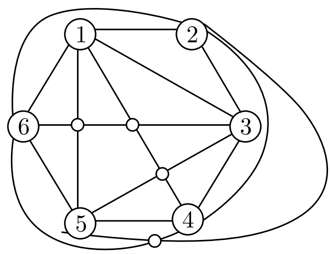


Рис. 17.18. Граф K_6

Тогда $P(\mathsf{K}_n)$ при таком расположении равно 4. А число скрещиваний, согласно (17.18), запишется:

$$P_{\min}(\mathsf{K}_6) = 3.$$

Опишем процедуру расположения полного графа с минимальным числом пересечений.

Выделим в K_n ГЦ, и все вершины и ребра ГЦ расположим в виде выпуклой фигуры. При этом n ребер из $n(n-1)/2$ ребер K_n расположено на плоскости. Оставшиеся $n(n-1)/2 - n$ ребер, т. е. $n(n-3)/2$ ребра необходимо расположить внутри и вне выпуклой фигуры с наименьшим числом пересечений.

Проведем сначала подмножества ребер:

$$U_i = \{u(i, i+2), u(i-1, i+3), \ldots, u(i-q+1, i+q+1)\},$$

где

$$q = \begin{cases} \dfrac{n-3}{2} + \dfrac{1}{2}, & \text{если } n \text{ четно,} \\ \dfrac{n-3}{2}, & \text{если } n \text{ нечетно,} \end{cases} \tag{17.20}$$

если $i = 1$, то $i - 1 = n$, $i - 2 = n - 1, \ldots$

Заметим, что ребра подмножества U_i располагаются «параллельно» друг другу, не пересекаясь между собой.

Далее аналогично проводим подмножество ребер U_i'. Очевидно, что в полном графе с любым числом вершин ребра подмножеств U_i и U_i' не пересекаются между собой. Увеличивая i на единицу, проведем новые подмножества ребер U_{i+1}, U_{i+1}', которые минимальным образом пересекаются с ребрами предыдущих множеств.

Следующие подмножества ребер, которые минимальным образом пересекаются с уже расположенными, проводятся аналогично до тех пор, пока число таких подмножеств во внутренней области не будет равно

$$L = \begin{cases} \dfrac{n}{2}, & \text{если } n \text{ четно,} \\ \dfrac{n}{2} + \dfrac{1}{2}, & \text{если } n \text{ нечетно.} \end{cases} \tag{17.21}$$

Остальные подмножества ребер, дополняющие граф до полного, проводятся во внешней области. То есть начиная с подмножества U_{L+1} аналогичные «параллельные» подмножества проводятся во внешней области ГЦ. При этом, если в графе K_n число вершин n-четно, то у него одинаковое число подмножеств «параллельных» ребер во внешней и внутренней областях ГЦ. Соответственно при n-нечетном в одной области будет на одно подмножество больше, чем в другой.

Информацию о расположении ребер в K_n запишем в видоизмененной матрице смежности.

Для K_n $(n = 8)$:

$\mathrm{R}_8 =$

	1	2	3	4	5	6	7	8	1	2	3	4	5	6
1	0	×					1	×						
2		0	×			1	1	1	×					
3			0	×	1	1	1	–	–	×				
4				0	×	1	–	–	–	–	×			
5					0	×	–	–	–	–		×		
6						0	×	–	–				×	
7							0	×						×
8								0						

а K_8, расположенный на основе матрицы R_8, имеет вид (рис. 17.19).

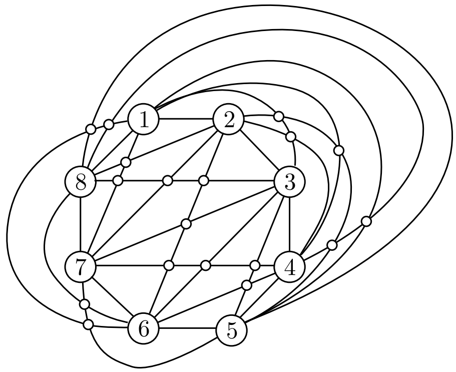


Рис. 17.19. Граф K_8

Подмножества параллельных ребер имеют вид:

$$\begin{array}{ll} U_1 - (1,7)(2,6)(3,5), & U_5 - (1,3)(5,7)(4,8), \\ U_2 - (2,7)(3,6), & U_6 - (1,4)(8,5), \\ U_3 - (2,8)(3,7)(4,6), & U_7 - (2,4)(1,5)(8,6), \\ U_4 - (3,8)(4,7), & U_8 - (1,6)(2,5). \end{array}$$

U_1, U_2, U_3, U_4 — подмножества параллельных ребер, проведенных во внутренней области. Согласно формуле (17.21) $L = 4$ и число подмножеств в обоих областях одинаково. U_5, U_6, U_7, U_8 — подмножества параллельных ребер, проведенных во внешней области.

Величина $P(\mathsf{K}_8) = 18$, это соответствует формуле Гая (17.19) (при этом 9 пересечений внутри ГЦ и 9 пересечений вне ГЦ).

Итак, для получения расположения графа на плоскости с $P_{\min}(\mathsf{G})$ необходимо единицы и черточки матрицы размещать в соответствующих диагоналях. Число диагоналей необходимо определять по формуле (17.21). Каждая диагональ должна быть заполнена элементами полностью. Заполнение диагоналей можно начинать с любой строки матрицы.

Построим, например R_{11}. По (17.21) определим число диагоналей, соответствующих «подмножествам параллельных ребер»:

$$L_{11}^{\text{внутр}} = \frac{n}{2} + \frac{1}{2} = 6, \quad L_{11}^{\text{внешн}} = \frac{n}{2} - \frac{1}{2} = 5.$$

$$
\mathrm{R}_{11} = \begin{array}{c|ccccccccccc||ccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & \times & & & & & & & & 1 & \times & & & & & \\
2 & & 0 & \times & & & & & & 1 & 1 & 1 & \times & & & & \\
3 & & & 0 & \times & & & & 1 & 1 & 1 & 1 & 1 & \times & & & \\
4 & & & & 0 & \times & & 1 & 1 & 1 & 1 & 1 & 1 & - & \times & & \\
5 & & & & & 0 & \times & 1 & 1 & 1 & 1 & 1 & - & - & - & \times & \\
6 & & & & & & 0 & \times & 1 & 1 & 1 & - & - & - & - & - & \times \\
7 & & & & & & & 0 & \times & 1 & - & - & - & - & - & & \\
8 & & & & & & & & 0 & \times & - & - & - & - & & & \\
9 & & & & & & & & & 0 & \times & - & - & & & & \\
10 & & & & & & & & & & 0 & \times & & & & & \\
11 & & & & & & & & & & & 0 & \times & & & & \\
\hline
\end{array}\,.
$$

Построение диагоналей начнем с первой строки матрицы. Тогда по формуле (17.19) для K_{11} получим:

$$P_{\min}(\mathsf{K}_{11}) \leqslant \frac{1}{64}(11-1)^2(11-3)^2 = \frac{100\cdot 64}{64} = 100.$$

Рассмотрим случай, когда число вершин K_n четно. Пусть имеется расположение K_n на плоскости с минимальным числом пересечений $P_{n,\min}$. Заметим, что в полном графе с четным n ребра, инцидентные вершине x_i, обладают одинаковыми свойствами по отношению к числу пересечений с ребрами, инцидентными любой другой вершине x_j $(i, j \in I = \{1, 2, \ldots, n\},\ i \notin j)$. Поэтому ребра, инцидентные x_i, имеют такое же число пересечений, как и ребра, инцидентные любой вершине x_j. Это вытекает из способа построения графа.

Каждое пересечение образуется наложением двух ребер, концы которых лежат в 4 вершинах. Поэтому каждое пересечение учитывается 4 раза. Общее число пересечений K_n равно

$$\sum_{i=1}^{n} P_i = 4P_{n,\min}. \tag{17.22}$$

При удалении из K_n любой вершины с инцидентными ей ребрами получим K_{n-1}, с минимальным числом пересечений.

Справедливость этих утверждений покажем построением видоизмененных матриц.

Например, в приведенной выше матрице R_8 удалим, например, третью строку и третий столбец. Получим:

$$
\mathrm{R}_7 = \begin{array}{c|ccccccc||ccccc|}
 & 1 & 2 & 4 & 5 & 6 & 7 & 8 & 1 & 2 & 4 & 5 & 6 \\
\hline
1 & 0 & \times & & & & 1 & \times & & & & & \\
2 & & 0 & \times & & 1 & 1 & 1 & \times & & & & \\
4 & & & 0 & \times & 1 & 1 & - & - & \times & & & \\
5 & & & & 0 & \times & - & - & - & - & \times & & \\
6 & & & & & 0 & \times & - & - & & & \times & \\
7 & & & & & & 0 & \times & & & & & \times \\
8 & & & & & & & 0 & \times & & & & \\
\hline
\end{array}.
$$

На основе R_7 построим оптимальное расположение этого графа K_7.

По формуле (17.19) имеем

$$P_{\min}(\mathsf{K}_7) \leqslant \frac{1}{64}(7-1)^2(7-3)^2 = \frac{36 \cdot 16}{64} = 9.$$

Тогда для любого K_n с четным n запишем

$$P_{n-1,\min} = P_{n,\min}\Big(1 - \frac{4}{n}\Big). \tag{17.23}$$

Из (17.23) получаем

$$P_{n,\min} = \frac{n}{n-4} P_{n-1,\min} \tag{17.24}$$

при $n \equiv 0 \pmod 2$ и $n \geqslant 6$.

Для K_n $n \neq 0 \pmod 2$ также имеет место выражение (17.22). Тогда

$$\sum_{i=1}^{n} P_i = P_{\max}(n-1). \tag{17.25}$$

Выражение (17.23) справедливо, так как удаление вершины x_i с $P_{\max}$ со всеми инцидентными ей ребрами из K_n позволяет получить K_{n-1} с четным числом вершин с минимумом пересечений.

На основе (17.25) и (17.22) получим

$$P_{n-1,\min} = P_{n,\min} - \frac{4P_{n,\min}}{n-1}. \tag{17.26}$$

После преобразования (17.26) имеем

$$P_{n,\min} = \frac{n-1}{n-5} P_{n-1,\min} \tag{17.27}$$

при $n \neq 0 \,(\mathrm{mod}\, 2)$ и $n \geqslant 7$.

Объединяя (17.24) и (17.27) в одно выражение, получим

$$P_{n,\min} = \frac{P_{n-1,\min}}{1 - \dfrac{8}{2n - 1 + (-1)^n}}. \tag{17.28}$$

После преобразования (3.28) можно получить для

$$n\text{-четного } P_{n,\min} = \frac{n(n-2)^2(n-4)}{64}, \tag{17.29}$$

$$n\text{-нечетного } P_{n,\min} = \frac{(n-1)^2(n-3)^2}{64}, \tag{17.30}$$

что доказывает верхнюю оценку Гая и гипотезу Харари–Хилла о том, что верхняя оценка (17.19) является точным значением.

Рассмотрим ряд эвристик, применяемых для минимизации пересечений ребер графа.

Э.1. Сначала производится сортировка вершин графа по возрастанию локальных степеней. Далее выбирается вершина с наименьшей локальной степенью и вокруг нее строятся треугольные грани, не имеющие пересечений, пока это возможно.

Э.2. В графе определяется ребро с наибольшим числом пересечений. Через него проводится секущая ось и весь граф перемещается по правую или левую часть от секущей оси. Далее процесс продолжается аналогично. При этом возможно несколько вариантов. Перед переносом графа проводить анализ: не приведет ли перемещение к увеличению числа пересечений. Тогда необходимо переходить к другой секущей оси и т. д.

Э.3. Минимизация пересечений при расположении ребер во внутренней и внешней областях ГЦ. Отсутствие ребер ГЦ позволит только уменьшить число пересечений за счет проведения некоторых ребер в том месте, где отсутствуют ребра ГЦ. Основная идея алгоритма следующая. Для обоих областей ГЦ строятся матрицы смежности. Далее определяется «отклонение» для каждого ребра G. Под отклонением ребра графа G понимается разность между числом пересечений этого ребра со всеми остальными, когда оно во внутренней и внешней областях ГЦ, затем строится матрица отклонений, из которой выбирается элемент, имеющий максимальное значение. Ребро, соответствующее этому элементу, выносится во внешнюю область ГЦ. Далее процесс продолжается аналогично, пока все элементы матрицы не станут отрицательными.

На основе эвристик разрабатываются эффективные алгоритмы минимизации пересечений ребер графовых моделей в любой области науки и производства.

Вопросы для самоконтроля

1. Дайте определение и приведите примеры плоских и планарных графов.
2. Дайте определение плоской карты.
3. Запишите теорему Эйлера о плоской карте.
4. Дайте определение максимально планарного графа.
5. Приведите теоремы о планарности графов.
6. Каким образом можно связать понятия двойственности и планарности графов?
7. Какие графы являются гомеоморфными?
8. Поясните понятия толщины графа и числа скрещиваний.
9. Приведите эвристики для определения планарности графа.
10. Сформулируйте основную идею алгоритмов минимизации пересечений ребер графа.

Задания для самостоятельной работы

1. Приведите примеры, подтверждающие справедливость теоремы 17.4 и ее следствий.
2. Запишите словесный алгоритм определения планарности на основе теоремы Понтрягина–Куратовского.
3. Запишите словесный алгоритм определения планарности на основе теоремы Харари–Татта.
4. Запишите словесный алгоритм определения планарности на основе теоремы Мак-Лейна.
5. Запишите словесный алгоритм определения планарности на основе теоремы Уитни.
6. Запишите словесный алгоритм определения планарности на основе теоремы Бадера.
7. Запишите словесный алгоритм определения планарности на основе циклических методов.

8. Запишите словесный алгоритм определения планарности на основе матричных методов.

9. Запишите словесный алгоритм определения планарности на основе комбинированных методов.

10. Запишите словесный алгоритм расположения полного графа на плоскости для минимизации пересечений ребер графа.

Планарность является важнейшим разделом теории графов. Определение планарных графов позволит строить модели объектов без пересечения ребер (связей).

17.4. Ориентированные графы

17.4.1. Способы задания. Часто при описании соединений в схемах необходимо учитывать направление прохождения сигнала.

Ориентированный граф будем обозначать $\mathsf{D} = (X, U)$ и называть графом. *Ориентированным графом* (*орграфом*) D, как уже было отмечено ранее, называется пара множеств $(X(\mathsf{D}), U(\mathsf{D}))$, где $X(\mathsf{D})$ — непустое конечное множество вершин графа D, а $U(\mathsf{D})$ — конечное множество упорядоченных пар элементов из $X(\mathsf{D})$, называемых дугами. К примеру, дуга, первым элементом которой является вершина x_i, а вторым — вершина x_j, называется дугой из x_i в x_j.

По аналогии с неориентированными графами определим некоторые базовые понятия для ориентированных графов. *Маршрутом* в орграфе D называется чередующаяся последовательность ребер и дуг $(x_0, u_1, x_1, \ldots, u_n, x_n)$, где каждая дуга $U_i = (x_{i-1}, x_i)$. *Путь* в орграфе — это маршрут, все вершины которого различны. *Контур* в орграфе D — это простой замкнутый маршрут, в котором совпадают только первая и последняя вершины. Если в орграфе D существует путь из вершины x_i в вершину x_j, то в этом случае говорят, что вершина x_j *достижима* из вершины x_i. Неориентированный граф, полученный из орграфа D путем замены всех дуг на ребра, называется *основанием* графа D.

Орграф D может быть однозначно задан матрицей смежности $\mathrm{R}(\mathsf{D}) = ||r_{ij}||_{n\times n}$, причем

$$r_{ij} = \begin{cases} 1, & \text{если } \langle x_i, x_j \rangle \text{ — дуга } \mathsf{D}, \\ 0 & \text{в противном случае.} \end{cases}$$

Например, пусть задан граф D, показанный на рис. 17.20.

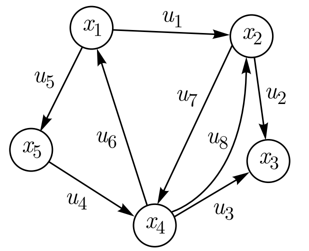


Рис. 17.20. Граф D

Тогда его матрица смежности $\mathrm{R}(\mathsf{D})$ имеет вид

$$\mathrm{R}(\mathsf{D}) = \begin{array}{c|ccccc|c|} & 1 & 2 & 3 & 4 & 5 & \rho^+(x_j) \\ 1 & 0 & 1 & 0 & 0 & 1 & 2 \\ 2 & 0 & 0 & 1 & 1 & 0 & 2 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 1 & 1 & 0 & 0 & 3 \\ 5 & 0 & 0 & 0 & 1 & 0 & 1 \\ \rho^-(x_j) & \boxed{1} & \boxed{2} & \boxed{2} & \boxed{2} & \boxed{1} & \end{array}.$$

Так как $r_{ij} \neq r_{ji}$, то матрица не симметрична относительно главной диагонали. То есть в общем случае матрица смежности орграфа не будет симметрична относительно главной диагонали.

Дуга $u_i = \langle x_i, x_j \rangle$ считается положительно инцидентной ее конечной вершине x_j. Число дуг, положительно инцидентных вершине x_j (т. е. сумма единиц строки матрицы D), называется *полустепенью захода* и обозначается $\rho^+(x_j)$. Число дуг, отрицательно инцидентных x_j, т. е. выходящих из x_j (сумма единиц столбца матрицы D), называется *полустепенью исхода* и обозначается через $\rho^-(x_j)$. Сумма вышеуказанных величин определяет *локальную степень* $\rho(x_i)$ вершины x_i:

$$\rho(x_j) = \rho^+(x_j) + \rho - (x_j).$$

Так как любая дуга положительно инцидентна некоторой вершине x_j и также отрицательно инцидентна той же самой вершине x_j, то

$$\sum_{x_j \in X} \rho^+(x_j) = \sum_{x_j \in X} \rho^-(x_j) = |U|.$$

Ориентированный граф D также можно задать матрицей инцидентности $\mathrm{I}(\mathsf{D})$. С учетом вышеупомянутых определений элемент x_i матрицы инцидентности $\mathrm{I}(\mathsf{D})$ может принимать одно из трех значений:

1) элемент равен нулю, если не существует дуги, инцидентной рассматриваемой вершине (вершина не инцидентна дуге);

2) элемент равен $+1$, если существует дуга, исходящая из вершины;

3) элемент равен -1, если существует входящая в вершину дуга.

Для графа D на рис. 17.20 матрица инцидентности имеет вид

$$\mathrm{I}(\mathsf{D}) = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{array} \\ \left| \begin{array}{cccccccc} +1 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & 0 & +1 & -1 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 & +1 & -1 & +1 \\ 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 \end{array} \right| \end{array}.$$

17.4.2. Решение стандартных графовых задач с использованием орграфов. Как уже отмечалось выше, любой неориентированный граф G можно преобразовать в ориентированный граф D путем замены каждого ребра двумя противоположно направленными дугами. Отсюда следует, что многие рассмотренные нами ранее алгоритмы решения задач для неориентированных графов применимы и для орграфов. К числу таких алгоритмов относятся задачи нахождения гамильтонова цикла, задача коммивояжера и другие. Мы не будем в данном случае останавливаться на рассмотрении вышеперечисленных задач, поскольку они идентичны алгоритмам, описанным ранее для неориентированных графов.

Маршрутом графа D считается чередующаяся последовательность вершин и дуг $(x_0, u_1, x_1, u_n, x_n)$, в которой каждая дуга u_i есть кортеж $u_i = \langle x_{i-1}, x_i \rangle$. Маршрут, в котором все вершины различны, называется *путем*. Замкнутый маршрут, у которого все вершины различны, за исключением первой и последней, называется *контуром*. Если существует путь из x_i в x_j, то говорят, что x_j достижима из x_i.

Два орграфа называются изоморфными, если можно установить изоморфизм между их основаниями при сохранении порядка стрелок на каждой дуге.

Неорграф G называется ориентируемым, если каждое его ребро можно ориентировать так, что полученный граф D будет сильно связным. Такой процесс обычно называют заданием ориентации графа G. Очевидно, что произвольный эйлеров граф может быть ориентируем, так как достаточно пройти по любой эйлеровой цепи, ориентируя ребра графа в направлении движения.

Граф D называется эйлеровым, если в нем существует замкнутая цепь, содержащая каждую его дугу.

Необходимым условием существования эйлерова орграфа является его сильная связность.

Связный граф $\mathsf{D} = (X, U)$ является эйлеровым, когда $\forall x_i \in X(\rho^+(x_j) = \rho^-(x_j))$.

Орграф называется гамильтоновым, если в нем существует контур, содержащий каждую вершину орграфа.

Теорема 17.8. *Пусть* D *— сильно связный граф. Если* $\forall x_i \in X(\rho^+(x_j) \geqslant n/2$ *и* $\rho^-(x_j) \geqslant n/2) \Rightarrow \mathsf{D}$, *то* D *— гамильтонов граф.*

Рассмотрим алгоритм построения дерева на ориентированном графе $\mathsf{D} = (X, U)$.

1. Выбираем произвольную вершину x_i и по матрице смежности определяем все смежные ей вершины (т. е. определяем вершины, соответствующие стрелкам, исходящим из x_i). В результате образуется подмножество вершин $X' = \{x_i, x_j, \ldots, x_v\}$, смежных с x_i. В случае, если $X' = X$, то переходим к п. 4, в противном случае переходим к п. 2.

2. Выбираем произвольную вершину $x_j \in X'$ такую, что $x_j \neq x_i$. Снова определяем все смежные с x_j вершины, за исключением вершин, уже вошедших в X'.

3. Проверка условия $X' = X$, если условие выполняется, то переход к п. 4, в противном случае переход к п. 2.

4. Конец работы алгоритма, построено дерево T.

17.4.3. Выделение сильносвязных компонент. Пусть задан ориентированный граф $\mathsf{D} = (X, U)$, где X, U — соответственно множество вершин и множество ребер графа. Граф D называется *сильносвязным*, если любые две его вершины взаимно достижимы. *Подграфом* D' графа D называется граф, все вершины которого и инцидентные им ребра принадлежат D, т. е. $\mathsf{D}' = (X', U')$, $X' \subseteq X$ и $U' \subseteq U$, и ребра D' инцидентны только вершинам из X'.

Граф G, полученный из графа D заменой каждой дуги $u_i = \langle x_k, x_l \rangle$ на соответствующее ребро $u_i = (x_k, x_l)$, т. е. устранением стрелок, называется основанием D.

Тогда максимально сильносвязный подграф графа D будем называть *сильной компонентой* графа D. Отметим, что в ориентированном графе существует одна и только одна сильная компонента, содержащая данную вершину x_i.

Множество вершин $\mathrm{R}(x_i)$ графа D, достижимых из некоторой вершины x_i, состоит из таких элементов x_j, для которых (i, j) элемент в *матрице достижимости* $\mathrm{R} = ||r_{ij}||$ равен 1. *Матрица контрдостижимости* определяется как $\mathrm{Q} = \mathrm{R}^t$, где R^t — матрица, транспонированная к матрице достижимости R.

Если вершина x_i одновременно является начальной и конечной вершиной пути, то множество вершин некоторого цикла, содержащего x_i, совпадает с пересечением $[\mathrm{R}(x_i) \cap \mathrm{Q}(x_i)]$ и все они *взаимно достижимы*.

В случае удаления этих вершин из исходного графа в оставшемся *порожденном* графе $\mathsf{D}' = \langle x - \mathrm{R}(x_i) \cap \mathrm{Q}(x_i)\rangle$ таким же способом выделяется новая сильная компонента, содержащая $x_j \in \langle x - \mathrm{R}(x_i) \cap \mathrm{Q}(x_i)\rangle$. Этот процесс продолжается до тех пор, пока весь исходный граф не будет разбит на сильные компоненты.

Граф $\mathsf{D}^* = (X^*, U^*)$ называется *конденсацией* графа D и определяется следующим образом: каждая его вершина представляет множество вершин некоторой сильной компоненты графа D, дуга $\langle x_i^*, x_j^*\rangle$ существует в D^* тогда и только тогда, когда в графе D существует дуга $\langle x_i, x_j\rangle$ такая, что x_i принадлежит компоненте, соответствующей вершине x_i^*, а x_j — компоненте, соответствующей вершине x_j^*. Очевидно, что конденсация графа D не содержит циклов, поскольку наличие цикла означает, что любые вершины этого цикла взаимно достижимы.

Определим прямое и обратное транзитивные замыкания. *Прямым транзитивным замыканием* $\Gamma^+(x_i)$ называют подмножество вершин $X' \subseteq X$, в которые можно попасть из вершины x_i по некоторому пути. *Обратным транзитивным замыканием* называют подмножество вершин, из которых можно попасть в x_i по некоторому пути, и обозначают $\Gamma^-(x_i)$.

Прямое и обратное транзитивные замыкания можно найти из соотношений:

$$\begin{aligned}
&\Gamma^+(x_i) = \{x_i\} \cup \Gamma^+(x_i) \cup \Gamma^{+2}(x_i) \cup \Gamma^{+3}(x_i) \cup \ldots,\\
&\Gamma^-(x_i) = \{x_i\} \cup \Gamma^-1(x_i) \cup \Gamma^{-2}(x_i) \cup \Gamma^{-3}(x_i) \cup \ldots,\\
&\Gamma^{+2}(x_i) = \Gamma^+\{\Gamma(x_i)\},\\
&\Gamma^{+3}(x_i) = \Gamma^+\{\Gamma^{+2}(x_i)\} = \Gamma^+\big\{\Gamma^+\{\Gamma^+(x_i)\}\big\}.
\end{aligned}$$

Аналогично определяем для $\Gamma^{-2}(x_i)$, $\Gamma^{-3}(x_i)$ и т. д.

Опишем *алгоритм Мальгранжа* разбиения графа D на максимально связные подграфы.

Идея алгоритма заключается в следующем. Выбирается произвольная вершина $x_i \subseteq X$ в графе D, для которой определяются прямое и обратное транзитивные замыкания $\Gamma^+(x_i)$, $\Gamma^-(x_i)$. Их пересечение $C(x_i) = \Gamma^+ x_i \cap \Gamma^- x_i$ является сильной компонентой графа D, т. е. максимально связным подграфом. Далее выбирается следующая вершина $x_j \notin C(x_i)$ и процесс продолжается.

Заметим, что любой неорграф G можно перевести в орграф путем замены каждого ребра двумя противоположно направленными дугами. В этой связи многие результаты для неорграфов могут быть интерпретированы на орграфы.

17.4.4. Нечеткие ориентированные графы. Рассмотрим нечеткие ориентированные графы. Они, так же как и нечеткие неориентированные графы, могут быть двух видов. К первому виду относят графы, имеющие нечеткое множество ребер. Ко второму виду относят графы, имеющие, кроме того, нечеткое множество вершин.

Зададим нечеткий ориентированный граф первого рода $\widetilde{\mathsf{G}} = (X, \widetilde{U})$. При этом $X = \{x_i\}$, $i \in I = \{1, 2, \ldots, n\}$, а $\widetilde{U} = \{\langle \mu_U \langle x_i, x_k \rangle / \langle x_i, x_k \rangle \rangle\}$, $\langle x_i, x_k \rangle \in X^2$, $\mu_U \langle x_i, x_k \rangle$ — степень принадлежности ориентированного ребра $\langle x_i, x_k \rangle$ нечеткому множеству ориентированных ребер $\widetilde{U}$.

Нечеткий ориентированный граф первого рода можно также удобно задавать в виде $\widetilde{\mathsf{G}} = (X, \widetilde{\Gamma})$, где $X = \{x_i\}$, $i \in I = \{1, 2, \ldots, n\}$, а $\widetilde{\Gamma}$ — нечеткое многозначное отображение множества вершин X в себя, т. е. $\widetilde{\Gamma}$: $X \to X$, задаваемое в виде системы нечетких образов элементов $x \in X$ при этом отображении, т. е. $\widetilde{\Gamma}(x_i) = \{\langle \mu_\Gamma(x_j)/x_j \rangle\}$, $x_j \in \Gamma(x_i)$, здесь $\Gamma(x_i)$ — четкое множество образов вершины $x_i \in X$.

Как нечеткий неориентированный, так и ориентированный графы удобно задавать в виде нечетких матриц смежности $\mathrm{R}_x = ||r_{ik}||_n$, где $r_{ik} = \mu_U(x_i, x_k)$ для неориентированных графов и $\mu_U \langle x_i, x_k \rangle$ для ориентированных графов.

Нечетким ориентированным графом второго вида называется граф $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U}$, где $\widetilde{X}$ — множество вершин является нечетким множеством в некотором универсальном множестве A, т. е. $\widetilde{X} = \{\langle \mu_x(x)/x \rangle\}$, $x \in A$, $|\widetilde{X}| = n$, $\widetilde{U}$ — нечеткое множество ориентированных ребер определяется как $\widetilde{U} = \{\langle \mu_U \langle x_i, x_k \rangle / \langle x_i, x_k \rangle \rangle\}$, $\langle x_i, x_k \rangle \in X^2$, где X — носитель множества $\widetilde{X}$.

Нечеткий ориентированный граф второго вида $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U})$ при необходимости можно однозначно преобразовать в нечеткий ориентированный граф первого вида $\widetilde{\mathsf{G}}' = (X, \widetilde{U}')$ следующим образом. В качестве множества вершин X принимается носитель множества $\widetilde{X}$, а нечеткое множество ориентированных ребер $\widetilde{U}'$ принимает вид: $\widetilde{U}' = \{\langle \mu'_U \langle x_i, x_k \rangle / \langle x_i, x_k \rangle \rangle\}$, $\langle x_i, x_k \rangle \in X^2$, где функция $\mu'_U \langle x_i, x_k \rangle = \mu_U \langle x_i, x_k \rangle \,\&\, \mu_X(x_i) \,\&\, \mu_X(x_k)$, в минимаксном базисе — $\mu'_U \langle x_i, x_k \rangle = \min\left(\mu_U \langle x_i, x_k \rangle, \mu_X(x_i), \mu_X(x_k)\right)$, и в вероятностном базисе — $\mu'_U \langle x_i, x_k \rangle = \mu_U \langle x_i, x_k \rangle \times \mu_X(x_i) \times \times \mu_X(x_k)$. Графы $\widetilde{\mathsf{G}}$ и G' будем называть сопряженными.

Каждому ориентированному нечеткому графу второго вида соответствует единственный ориентированный нечеткий граф первого вида, в то же время каждому ориентированному нечеткому графу первого вида соответствует бесконечно много нечетких графов второго вида.

Аналогичные преобразования можно выполнить и для преобразования нечеткого неориентированного графа второго вида к нечеткому неориентированному графу первого вида.

Примеры решения задач

Пример 17.7. Пусть задан орграф D, изображенный на рис. 17.21, *а*. Построить основание, матрицу смежности и матрицу инцидентности заданного графа.

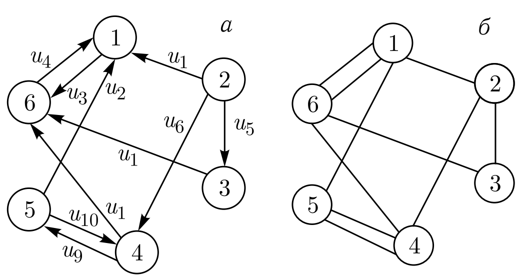


Рис. 17.21. Ориентированный граф D (*а*); основание графа D (*б*)

Решение. Основание исходного графа D, полученное путем замены дуг соответствующими им ребрами, показано на рис. 17.21, *б*.

Матрица смежности исходного графа D будет иметь следующий вид:

$$
\mathrm{R}(\mathsf{D}) = \begin{array}{c|cccccc|c|}
 & 1 & 2 & 3 & 4 & 5 & 6 & \rho^+(x_j) \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
2 & 1 & 0 & 1 & 1 & 0 & 0 & 3 \\
3 & 0 & 0 & 0 & & 0 & 1 & 1 \\
4 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\
5 & 1 & 0 & 0 & 1 & 0 & 0 & 2 \\
6 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
\rho^-(x_j) & 3 & 0 & 1 & 2 & 1 & 3 &
\end{array}.
$$

Таким образом, для вершины x_1, например, полустепень исхода $\rho^+(x_1)$ равна единице, а полустепень захода $\rho^-(x_1)$ равна трем.

Матрица инцидентности для того же графа запишется следующим образом:

$$
\mathrm{I}(\mathsf{D}) = \begin{array}{c|cccccccccc|}
 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} \\
x_1 & -1 & -1 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
x_2 & +1 & 0 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 \\
x_3 & 0 & 0 & 0 & 0 & -1 & 0 & +1 & 0 & 0 & 0 \\
x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 \\
x_5 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \\
x_6 & 0 & 0 & -1 & +1 & 0 & 0 & -1 & -1 & 0 & 0
\end{array}.
$$

Пример 17.8. Пусть задан ориентированный граф $\mathsf{D} = (X, U)$ (рис. 17.22). Построить дерево, используя вышеописанный алгоритм.

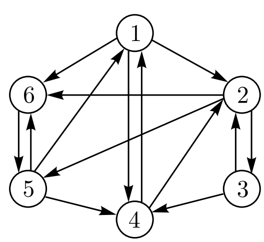


Рис. 17.22. Орграф D

Решение. 1. Выбираем вершину x_1. Находим смежные ей вершины и получаем подмножество $X' = \{x_1, x_2, x_4, x_6\}$. Поскольку $X' \neq X$, переходим к п. 2.

2. Из подмножества X' выбираем произвольную вершину, например x_2. Находим смежные ей вершины. Это вершины x_3, x_5, x_6. Добавляем в подмножество X' вершины x_3, x_5. Вершина x_6 уже входит в подмножество X', поэтому дугу $\langle x_2, x_6\rangle$ мы игнорируем.

3. После выполнения п. 2 настоящего алгоритма получим подмножество $X' = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, таким образом, условие $X' = X$ выполнено, переходим к п. 4.

4. Построено дерево $\mathsf{T} = \{\langle x_1, x_2\rangle, \langle x_1, x_4\rangle, \langle x_1, x_6\rangle, \langle x_2, x_3\rangle, \langle x_2, x_5\rangle\}$. Алгоритм окончен.

Ответ. Построено дерево T (рис. 17.23).

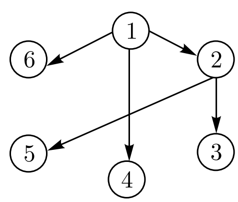


Рис. 17.23. Дерево T

Пример 17.9. Разбить граф $\mathsf{G} = (X, U)$, изображенный на рис. 17.24, на максимально связные подграфы и построить его конденсацию.

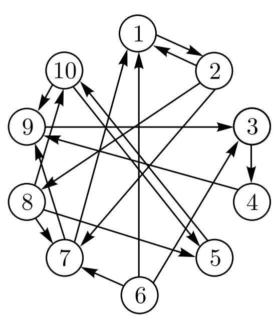


Рис. 17.24. Граф G

Решение. Используем алгоритм Мальгранжа.

Выбираем произвольную вершину. Пусть это будет вершина x_1. Строим матрицу смежности заданного графа.

$$
\mathrm{R} = \begin{array}{r|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & & 1 & & & & & & & & \\
2 & 1 & & & & & & 1 & 1 & & \\
3 & & & & 1 & & & & & & \\
4 & & & & & & & & & 1 & \\
5 & & & & & & & & & & 1 \\
6 & 1 & & 1 & & & & 1 & & & \\
7 & 1 & & & & & & & & 1 & \\
8 & & & & & 1 & & 1 & & & 1 \\
9 & & & 1 & & & & & & & \\
10 & & & & & 1 & & & & 1 &
\end{array} .
$$

По матрице смежности определяем вершины, достижимые из вершины x_1:

$$\Gamma^+(x_1) = \{x_1\} \cup \{x_2\}.$$

Теперь определяем вершины, достижимые из x_2:

$$\Gamma^+(x_1) = \{x_1\} \cup \{x_2\} \cup \{x_1, x_7, x_8\}.$$

Определяем вершины, достижимые из x_7 и x_8:

$$\Gamma^+(x_1) = \{x_1\} \cup \{x_2\} \cup \{x_1, x_7, x_8\} \cup \{x_1, x_9, x_5, x_7, x_{10}\}.$$

Определяем вершины, достижимые из вершин x_5, x_9, x_{10}:

$$\Gamma^+(x_1) = \{x_1\} \cup \{x_2\} \cup \{x_1, x_7, x_8\} \cup \\ \cup \{x_1, x_5, x_7, x_9, x_{10}\} \cup \{x_3, x_5, x_9, x_{10}\}.$$

Определяем вершины, достижимые из вершины x_3:

$$\Gamma^+(x_1) = \{x_1\} \cup \{x_2\} \cup \{x_1, x_7, x_8\} \cup \\ \cup \{x_2, x_5, x_7, x_9, x_{10}\} \cup \{x_3, x_5, x_9, x_{10}\} \cup \{x_4\}.$$

Поскольку из вершины x_4 достижима только вершина x_9, а она уже включена в список рассмотренных вершин, то в итоге, после выполнения операции объединения данных подмножеств вершин, получим $\Gamma^+(x_1) = \{x_1, x_2, x_3, x_4, x_5, x_7, x_8, x_9, x_{10}\}$.

Теперь нам необходимо найти вершины, из которых достижима вершина x_1:

$$\Gamma^-(x_1) = \{x_1\} \cup \{x_2, x_6, x_7\} \cup \{x_1\} \cup \{x_2, x_6, x_8\} \cup \{x_2\} = \\ = \{x_1, x_2, x_6, x_7, x_8\}.$$

Теперь для того, чтобы выделить максимально связную компоненту, нам необходимо найти пересечение данных множеств:

$$\Gamma^{+}(x_1) \cap \Gamma^{-}(x_1) = \{x_1, x_2, x_7, x_8\}; \quad X_1^* = \{x_1, x_2, x_7, x_8\}.$$

Исключаем эти вершины из рассмотрения и продолжаем процесс выделения максимально связных компонент. Берем вершину x_3 и определяем для нее прямое и обратное транзитивные замыкания:

$$\Gamma^{+}(x_3) = \{x_3\} \cup \{x_4\} \cup \{x_9\} \cup \{x_3\} = \{x_3, x_4, x_9\};$$

$$\Gamma^{-}(x_3) = \{x_3\} \cup \{x_6, x_9\} \cup \{x_4, x_{10}\} \cup \{x_3, x_5\} \cup \{x_{10}\} = \\ = \{x_3, x_4, x_5, x_6, x_9, x_{10}\}.$$

Тогда сильная компонента:

$$X_2^* = \Gamma^{+}(x_3) \cap \Gamma^{-}(x_3) = \{x_3, x_4, x_9\}.$$

Продолжаем процесс. Выбираем из оставшихся вершин x_5, x_6, x_{10} вершину x_5:

$$\Gamma^{+}(x_5) = \{x_5\} \cup \{x_{10}\} \cup \{x_5\} = \{x_5, x_{10}\};$$

$$\Gamma^{-}(x_5) = \{x_5\} \cup \{x_{10}\} \cup \{x_5\} = \{x_5, x_{10}\}.$$

Таким образом, получена сильная компонента:

$$X_3^* = \{x_5, x_{10}\}.$$

У нас осталась всего одна невыделенная вершина — это вершина x_6. Следовательно, вершина x_6 образует последнюю сильную компоненту в графе G: $X_4^* = \{x_6\}$.

Итак, мы выделили в заданном графе все возможные сильные компоненты.

Теперь строим конденсацию графа G. В качестве вершин графа конденсации выступают сильные компоненты, при этом вершины, составляющие каждую сильную компоненту, стягиваются в одну. Все ребра, соединяющие стягиваемые вершины с вершинами, принадлежащими другим сильным компонентам, сохраняются. Таким образом, мы построили конденсацию D^* (рис. 17.25) исходного графа G.

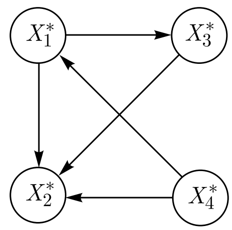


Рис. 17.25. Конденсация D^*

Пример 17.10. Для графа $\mathsf{D} = (X, U)$ на рис. 17.26 определить прямое и обратное транзитивные замыкания для вершины x_7.

Решение.

$$\Gamma^+ x_7 = \{x_7\} \cup \Gamma^+ x_7 \cup \Gamma^+ 2x_7 \cup \Gamma^+ 3x_7;$$

$$\Gamma^+ x_7 = \{x_7, x_4, x_6\};$$

$$\Gamma^{+2} x_7 = \Gamma^+\{\Gamma^+ x_7\} = \Gamma^+\{x_7, x_4, x_6\} = \{x_7, x_4, x_6, x_2, x_5\};$$

$$\Gamma^{+3} x_7 = \Gamma^+\{\Gamma^{+2} x_7\} = \Gamma^+\{x_7, x_4, x_6, x_2, x_5\} = \\ = \{x_7, x_6, x_4, x_5, x_1, x_2, x_3\};$$

$$\Gamma^- x_7 = \{x_7\} \cup \Gamma^{-1} x_7 \cup \Gamma^{-2} x_7;$$

$$\Gamma^- x_7 = \{x_7, x_2\};$$

$$\Gamma^{-2} x_7 = \Gamma^-\{\Gamma^- x_7\} = \{x_7, x_2, x_4\},$$

тогда $\Gamma^- x_7 = \{x_2, x_4, x_7\}$, а $\Gamma x_7 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} = X$.

Пример 17.11. Построить разбиение на основе метода Мальгранжа ориентированного графа $\mathsf{D} = (X, U)$, заданного матрицей смежности R.

$$\mathrm{R} = \begin{array}{c|ccccccc||c|}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \Gamma^+ x_1 \\
x_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
x_2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & - \\
x_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
x_4 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & - \\
x_5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & - \\
x_6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & - \\
x_7 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & - \\
\hline
\Gamma^- x_1 & 0 & 1 & 1 & 2 & 2 & 3 & 3 &
\end{array}.$$

Решение. Столбец Γ^+x_1 строится следующим образом. В клетке столбца против стрелки x_1 ставится 0. В клетку столбца против строки x_3 ставится 1, так как строка x_1 содержит 1 на месте x_3. Строка x_3 содержит 1 только на уже отмеченном месте x_1. Тогда в оставшиеся клетки столбца Γ^+x_1 проставляются знаки «–». Числа в клетках Γ^+x_1 — это количество путей из вершины x_1 в другие вершины графа D. Тогда $\Gamma^+x_1 = \{x_1, x_3\}$.

Аналогично заполняются клетки строки Γ^-x_1. Против столбца x_1 в клетке Γ^-x_1 ставится 0. В клетках строки Γ^-x_1 против x_2 и x_3 ставится 1, так как столбец имеет 1 против x_2 и x_3. В столбце x_2 имеется 1 против строки x_4, следовательно, в клетку Γ^-x_1 против x_4 записывается 2. В столбце x_3 имеем единицы против строк x_1 и x_5. Строка x_1 уже рассмотрена, а в клетке Γ^-x_1, расположенной против x_5, ставится 2. Далее аналогично заполняются оставшиеся клетки Γ^-x_1. Тогда $\Gamma^-x_1 = \{x_1, x_2, x_3, x_5, x_6, x_7\}$, $C(x_1) = \Gamma^+x_1 \cap \Gamma^-x_1 = \{x_1, x_3\}$. Теперь удалим из графа D вершины x_1 и x_3. Получим матрицу R' оставшегося графа D':

$$\mathrm{R}' = \begin{array}{c|ccccc||c|}
 & x_2 & x_4 & x_5 & x_6 & x_7 & \Gamma^+x_2 \\
x_2 & 0 & 0 & 0 & 0 & 1 & 0 \\
x_4 & 1 & 1 & 0 & 0 & 0 & 2 \\
x_5 & 0 & 0 & 0 & 1 & 0 & 3 \\
x_6 & 0 & 0 & 1 & 0 & 0 & 2 \\
x_7 & 0 & 1 & 0 & 1 & 1 & 1 \\
\Gamma^-x_2 & 0 & 1 & - & - & 2 &
\end{array}.$$

Аналогично, заполняя клетки Γ^+x_2 и Γ^-x_2, определяем $\Gamma^+x_2 = \{x_2, x_4, x_5, x_6, x_7\}$, $\Gamma^-x_2 = \{x_2, x_4, x_7\}$. Тогда $C(x_2) = \Gamma^+x_2 \cap \Gamma^-x_2 = \{x_2, x_4, x_7\}$. Снова удалим из D' вершины x_2, x_4, x_7, получим граф D''. Запишем его матрицу смежности:

$$\mathrm{R}'' = \begin{array}{c|cc||c|}
 & x_5 & x_6 & \Gamma^+x_5 \\
x_5 & 0 & 1 & 0 \\
x_6 & 1 & 0 & 1 \\
\Gamma^-x_5 & 0 & 1 &
\end{array}.$$

Получим $\Gamma^+x_5 = \{x_5, x_6\}$, $\Gamma^-x_5 = \{x_5, x_6\}$, $C(x_5) = \{x_5, x_6\}$. Следовательно, в результате работы алгоритма получили разло-

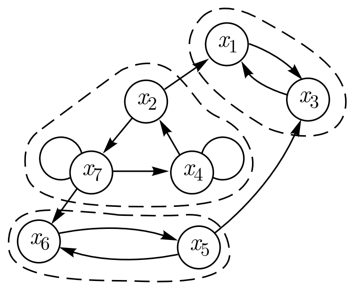


Рис. 17.26. Разложение графа на максимально связные подграфы

жение графа на три части, показанные на рис. 17.26 штриховыми линиями.

Пример 17.12. Зададим нечеткий ориентированный граф первого рода $\widetilde{\mathsf{G}} = (X, \widetilde{U})$, где $X = \{x_1, x_2, x_3, x_4, x_5\}$, а $\widetilde{U} = \{\langle 1/\langle x_1, x_4\rangle\rangle,\ \langle 0{,}5/\langle x_1, x_5\rangle\rangle,\ \langle 0{,}1/\langle x_2, x_2\rangle\rangle,\ \langle 0{,}9/\langle x_2, x_3\rangle\rangle,$ $\langle 0{,}8/\langle x_4, x_2\rangle\rangle,\ \langle 0{,}2/\langle x_4, x_1\rangle\rangle,\ \langle 0{,}6/\langle x_5, x_5\rangle\rangle\}$.

Граф показан на рис. 17.27.

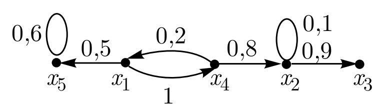


Рис. 17.27. Нечеткий ориентированный граф первого рода

Пример 17.13. Зададим нечеткий граф из примера 17.7 в виде $\widetilde{\mathsf{G}} = (X, \widetilde{\Gamma})$.

Решение. Получим $X = \{x_1, x_2, x_3, x_4, x_5\}$, $\widetilde{\Gamma}(x_1) = \{\langle 1/x_4\rangle, \langle 0{,}5/x_5\rangle\}$, $\widetilde{\Gamma}(x_2) = \{\langle 0{,}1/x_2\rangle, \langle 0{,}9/x_3\rangle\}$, $\widetilde{\Gamma}(x_3) = \varnothing$, $\widetilde{\Gamma}(x_4) = \{\langle 0{,}2/x_1\rangle, \langle 0{,}8/x_2\rangle\}$, $\widetilde{\Gamma}(x_5) = \{\langle 0{,}6/x_5\rangle\}$.

Пример 17.14. Записать матрицу смежности R_x^2 графа, рассмотренного в примере 17.7.

Решение. Матрица смежности запишется следующим образом:

$$\mathrm{R}_x^2 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left|\begin{array}{ccccc} 0 & 0 & 0 & 1 & 0{,}5 \\ 0 & 0{,}1 & 0{,}9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0{,}2 & 0{,}8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0{,}6 \end{array}\right| \end{array}.$$

Пример 17.15. Задать нечеткий ориентированный граф второго вида $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U})$, $\widetilde{X} = \{\langle 0{,}9/x_1\rangle, \langle 0{,}8/x_2\rangle, \langle 0{,}7/x_3\rangle, \langle 0{,}6/x_4\rangle, \langle 0{,}4/5\rangle\}$, $\widetilde{U} = \{\langle 0{,}4/\langle x_1, x_2\rangle\rangle, \langle 0{,}8/\langle x_1, x_3\rangle\rangle, \langle 1/\langle x_1, x_4\rangle\rangle, \langle 0{,}5/\langle x_3, x_4\rangle\rangle, \langle 0{,}9/\langle x_3, x_5\rangle\rangle\}$.

Решение. Граф $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U})$ приведен на рис. 17.28.

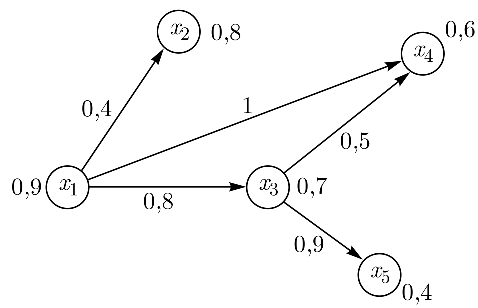


Рис. 17.28. Нечеткий ориентированный граф второго вида $\widetilde{\mathsf{G}} = (\widetilde{X}, \widetilde{U})$

Вопросы для самоконтроля

1. Дайте определение ориентированного графа.
2. Что понимается под достижимостью?
3. Что такое основание орграфа?
4. Что называется сильной компонентой орграфа?
5. Поясните принцип формирования матриц достижимости и контрдостижимости.
6. В каком случае говорят, что вершины орграфа взаимнодостижимы?
7. Как определяется прямое и обратное транзитивное замыкания?
8. Что такое конденсация графа?
9. Опишите алгоритм разбиения графа на максимально связные подграфы.
10. Определите полустепени исхода и захода.

Задания для самостоятельной работы

1. Для произвольного ориентированного графа $\mathsf{D} = (X, U)$ запишите его матрицу смежности, матрицу инцидентности и постройте его основание.

2. Для произвольного ориентированного графа $\mathsf{D} = (X, U)$ постройте дерево.

3. Разбейте произвольный ориентированный граф $\mathsf{D} = (X, U)$ на сильные компоненты и постройте его конденсацию.

4. Для произвольного ориентированного графа запишите алгоритм определения прямого транзитивного замыкания.

5. Для произвольного ориентированного графа запишите алгоритм определения обратного транзитивного замыкания.

6. Постройте алгоритм определения конденсации ориентированного графа.

7. Определите способы задания нечетких орграфов.

8. Постройте нечеткие орграфы первого и второго рода.

9. Запишите алгоритм определения прямого транзитивного замыкания в нечетком орграфе.

10. Запишите алгоритм определения обратного транзитивного замыкания в нечетком орграфе.

17.5. Гиперграфы

В описанных выше графах использовались бинарные отношения на множестве вершин. Однако необходимо отметить, что в общем случае на множестве вершин X графа $\mathsf{D} = (X, U)$ можно задать k-арные отношения с помощью гиперграфов. Такое обобщение понятия графа позволяет строить объекты, в которых каждое ребро может соединять не только две вершины, но и любое подмножество множества вершин, что является важным при решении многих технических задач. Это обстоятельство позволяет также при построении математических моделей более точно отражать некоторые свойства реальных объектов.

Дадим определение гиперграфа. Пусть $X = \{x_1, x_2, \ldots, x_n\}$ — конечное множество и $E = \{l_1, l_2, \ldots, l_m\}$ — семейство подмножеств X такое, что $(\forall l_i) l_i \neq 0$ и $\bigcup\limits_{i \in I} l_i = X$, $I = \{1, 2, \ldots, m\}$. Говорят, что объект $\mathsf{H} = (X, E)$ — *гиперграф* на множестве X, если он состоит из множества вершин X и множества ребер E, причем каждое ребро $l_i \in E$ представляет собой некоторое подмножество множества вершин, т. е. $l_i \subseteq X$. Если $\forall l_i \in E$ $(|l_i| = 2)$, то гиперграф H является графом G без изолированных вершин. На рис. 17.29, *а* показан пример гиперграфа $\mathsf{H} = (X, E)$, $|X| = 7$, $|E| = 5$. Заметим, что ребро l_5 с $|l_5| = 1$ есть петля. Ребра гиперграфа — $l_1 = \{x_1, x_2, x_3, x_4\}$; $l_2 = \{x_2, x_3, x_6, x_7\}$; $l_3 = \{x_5, x_4\}$; $l_4 = \{x_5, x_7\}$; $l_5 = \{x_1\}$.

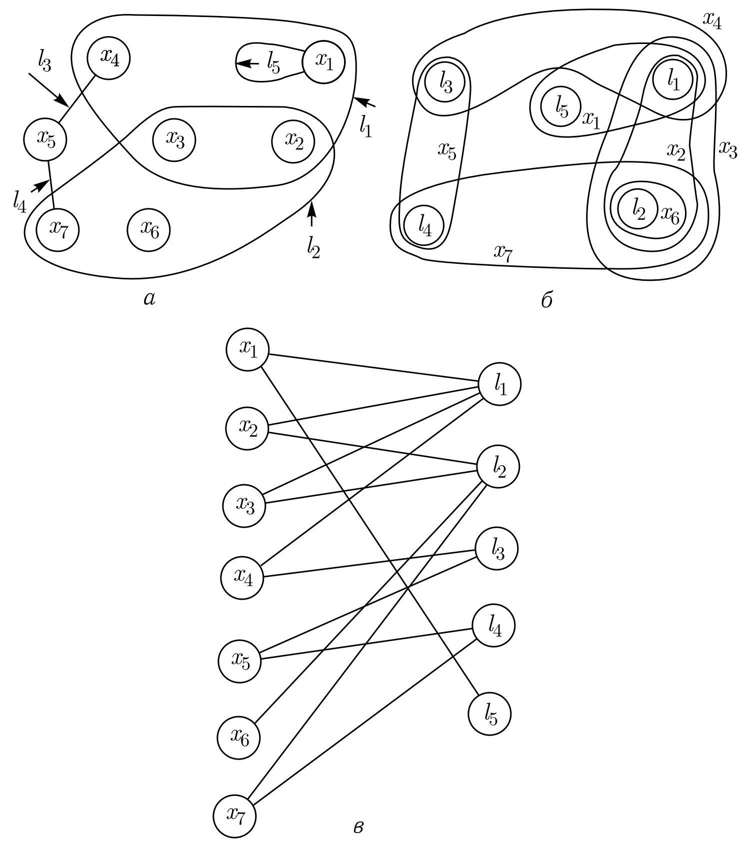


Рис. 17.29. Гиперграф $\mathsf{H} = (X, E)$ (*а*); двойственный гиперграф H^* (*б*); граф Кёнига $\mathsf{K}(\mathsf{H})$ (*в*)

В гиперграфе $\mathsf{H} = (X, E)$ две вершины называются *смежными* в том случае, если существует ребро e_i, содержащее эти вершины. Ребра гиперграфа называются *смежными* в случае, если выполняется следующее условие: $e_i \cap e_j \neq 0$, т. е. если их пересечение представляет собой непустое подмножество.

Матрицей инцидентности гиперграфа H называется матрица $I(\mathsf{H}) = ||h_{ij}||_{m\times n}$, причем

$$h_{ij} = \begin{cases} 1, & \text{если } x_j \in l_i; \\ 0, & \text{если } x_j \notin l_i. \end{cases}$$

Для гиперграфа H на рис. 17.29, *а* матрица инцидентности примет вид

$$\mathrm{I}(\mathsf{H}) = \begin{array}{c|ccccccc|} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \hline l_1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ l_2 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ l_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ l_4 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ l_5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}.$$

Любому гиперграфу $\mathsf{H} = (X, E)$ соответствует гиперграф $\mathsf{H}^* = (E, X)$, вершинами которого являются ребра $E = \{l_1, l_2, \ldots, l_m\}$, а ребрами — вершины $x_1, x_2, \ldots, x_n$. Гиперграф H^* называется *двойственным* H. Матрица инцидентности гиперграфа H^* есть транспонированная матрица гиперграфа H. Если два ребра $l_i, l_j \in E$ в H смежны, то соответствующие вершины $l_i, l_j \in E$ в H^* также смежны, то же выполняется и для вершин $x_i, x_j \in X$ в H, когда они становятся ребрами H^*. Гиперграф H^*, двойственный H, показан на рис. 17.29, *б*.

А его матрица инцидентности запишется в виде:

$$\mathrm{I}(\mathsf{H})^* = \begin{array}{c|ccccc|} & l_1 & l_2 & l_3 & l_4 & l_5 \\ \hline x_1 & 1 & 0 & 0 & 0 & 1 \\ x_2 & 1 & 1 & 0 & 0 & 0 \\ x_3 & 1 & 1 & 0 & 0 & 0 \\ x_4 & 1 & 0 & 1 & 0 & 0 \\ x_5 & 0 & 0 & 1 & 1 & 0 \\ x_6 & 0 & 1 & 0 & 0 & 0 \\ x_7 & 0 & 1 & 0 & 1 & 0 \end{array}.$$

В гиперграфе $\mathsf{H} = (X, E)$ *цепь* длины q определяется как последовательность вида $s(\mathsf{H}) = x_1, l_1, x_2, l_2, \ldots, l_q, x_{q+1}$. Если $q > 1$ и $x_{q+1} = x_1$, то цепь с такими параметрами называется циклом длины q.

Гиперграф H с n вершинами, m ребрами и k связными компонентами считается *деревом* в случае, когда его ребра имеют не более одной общей вершины, т. е. тогда, когда

$$\sum_{i=1}^{n} (|l_i| - 1) = n - k.$$

При решении задач автоматизации конструирования возникает необходимость сопоставлять гиперграфу граф.

Гиперграф $\mathsf{H} = (X, E)$ можно представить в виде двудольного графа Кёнига $\mathsf{K}(\mathsf{H}) = (X \cup E, V)$, где X — первое подмножество вершин двудольного графа, соответствующее множеству вершин гиперграфа H; E — второе подмножество вершин двудольного графа, соответствующее множеству ребер гиперграфа H; V — множество ребер двудольного графа, устанавливающее смежность между двумя множествами вершин X и E в графе $\mathsf{K}(\mathsf{H})$ в соответствии с инцидентностью вершин и ребер гиперграфа H.

Заметим, что в двудольном графе $\mathsf{K}(\mathsf{H})$ вершины внутри подмножеств X и E не смежны, причем вершины $x_i \in X$ и $l_j \in E$ в $\mathsf{K}(\mathsf{H})$ смежны тогда и только тогда, когда в гиперграфе H вершина x_i принадлежит ребру l_j. На рис. 17.29, *в* приведен граф Кёнига для гиперграфа H.

Отметим, что не только любой конечный гиперграф представляется в виде $\mathsf{K}(\mathsf{H})$, но и каждый двудольный граф является представлением некоторого гиперграфа.

Под раскраской вершин гиперграфа $\mathsf{H} = (X, E)$ понимается некоторое разбиение множества его вершин на классы, не содержащие пустого подмножества, когда инцидентные вершины должны быть окрашены в разные цвета, причем каждому классу $X_i (X_1 \cup X_2 \cup \ldots X_q = X; X_1 \cap X_2 \cap \ldots X_q = \varnothing)$ отвечает определенный цвет, в который окрашены вершины, а q соответствует количеству различных цветов, которые можно использовать для раскраски.

Примеры решения задач

Пример 17.16. Построить матрицу инцидентности гиперграфа $\mathsf{H} = (X, E)$, изображенного на рис. 17.30.

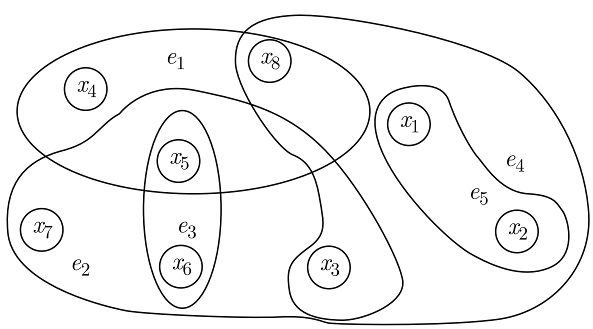


Рис. 17.30. Гиперграф H

Граф, изображенный на рис. 17.30, можно задать следующей матрицей инцидентности.

$$
\mathrm{I}(\mathsf{H}) = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{array} \\
\left\| \begin{array}{cccccccc}
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \right\|
\end{array}.
$$

Вопросы для самоконтроля

1. Дайте определение гиперграфа.
2. Какие две вершины либо два ребра гиперграфа считают смежными?
3. Как составляется матрица инцидентности гиперграфа?
4. Дайте определение цепи в гиперграфе.
5. Как составляется матрица смежности гиперграфа?
6. Каким образом для гиперграфа можно построить двойственный граф?
7. Какой гиперграф считается деревом?
8. Что такое раскраска гиперграфа?
9. Дайте определение двудольного графа Кёнига.
10. Как строится двудольный граф Кёнига?

Задания для самостоятельной работы

1. Приведите пример гиперграфа.
2. Для произвольного гиперграфа запишите матрицу инцидентости.
3. Для произвольного гиперграфа запишите матрицу смежности.
4. Для произвольного гиперграфа постройте двойственный граф.
5. Задайте произвольный гиперграф и покажите на нем примеры цепей различной длины.
6. Приведите примеры гиперграфов, являющихся деревьями.
7. Задайте произвольный гиперграф и постройте его раскраску.
8. Для произвольного гиперграфа постройте двудольный граф Кенига.
9. Составьте алгоритм задания матрицы смежности гиперграфа.
10. Составьте алгоритм задания матрицы инцидентности гиперграфа.

17.6. Задача определения паросочетаний в графе

Проведем ряд понятий и определений о паросочетаниях. Два ребра графа называются независимыми, если они не имеют общей вершины. Тогда паросочетание (ПС) — это множество независимых ребер. Паросочетание с наибольшим числом ребер называется максимальным (МПС).

Рассмотрим подробнее паросочетание в двудольном графе $\mathsf{G} = (X, Y)$, где $X = A \cup B$, $A \cap B = \varnothing$. Можно также записать $\mathsf{G} = (A \cup B, U)$ или $\mathsf{G} = (A, B; U)$.

Подмножество ребер $C \subseteq U$ двудольного графа $\mathsf{G} = (A, B; U)$ называется паросочетанием (ПС), если никакие два ребра из C не имеют общей вершины. Другими словами, ребра в C не являются идентичными друг другу. Тогда по аналогии с вышесказанным паросочетание C в двудольном графе $\mathsf{G} = (A, B; U)$ называется максимальным ($C \subseteq U$), если никакое другое паросочетание в G не содержит ребер больше, чем C.

Паросочетание $C \subseteq U$ в двудольном графе $\mathsf{G} = (A, B; U)$ называется полным, если $(\forall x \in A,\ \exists y \in B)\ ((x, y) \in C)$. Другими словами, для любого x из A существует y из B и обязательно существует ребро (xy) из C.

Пример 17.17. Пусть задан граф $\mathsf{G} = (A, B; U)$, $|A| = 4$, $|B| = 4$, $|U| = 12$, $A = \{1, 2, 3, 4\}$, $B = \{5, 6, 7, 8\}$ (рис. 17.31).

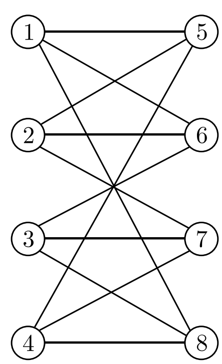


Рис. 17.31. Двудольный граф $\mathsf{G} = (A, B; U)$

Полное паросочетание имеет вид $C_n = \{(1,5),(2,6),(3,7),(4,8)\}$. Соответственно одно из возможных паросочетаний может иметь вид

$$C = \{(1, 6), (3, 8)\}.$$

Максимальных паросочетаний может быть несколько. В нашем примере они совпадают по мощности с полным паросочетанием. Приведем МПС для двудольного графа (рис. 17.31):

$$\text{МПС1} = \{(1,6),(2,7),(3,8),(4,5)\},$$

$$\text{МПС2} = \{(2,5),(3,6),(4,7),(1,8)\}.$$

В двудольном графе $\mathsf{G} = (A, B; U)$ при $C \subseteq A$ вводится подмножество $R(C) = \{y | y \in B \wedge b$ смежная с вершиной $x_i \in C\}$.

Существование полных паросочетаний можно определять на основе теоремы Холла.

Теорема 17.9. *Двудольный граф* $\mathsf{G} = (A, B; U)$ *имеет полное паросочетание тогда и только тогда, когда для каждого подмножества* $C \subseteq A$ *справедливо* $|C| \leqslant |R(C)|$.

Доказательство теоремы следует из определения и способа построения $R(C)$.

Пусть в $\mathsf{G} = (A, B; U) C \subseteq A$ и $\Gamma(C) \subseteq B$ — подмножество вершин из C. Предположим, что $|C| > |\Gamma(C)|$. Тогда очевидно, что не существует полного паросочетания и справедливо выражение $|\text{ПС}| \leqslant |A| - (|C| - |\Gamma(C)|)$.

Пример 17.18. Например, имеем двудольный граф, изображенный на рис. 17.32.

Здесь $\mathsf{G} = (A, B; U)$, $A = \{1,2,3,4\}$, $|A| = 4$, $B = \{5,6,7\}$, $|B| = 3$, $U = \{(1,7),(2,7),(3,7),(3,5),(3,6),(4,7)\}$, $|U| = 6$. Имеем произвольное паросочетание ПС $\{(3,6),(4,7)\}$, $|\text{ПС}| = 2$.

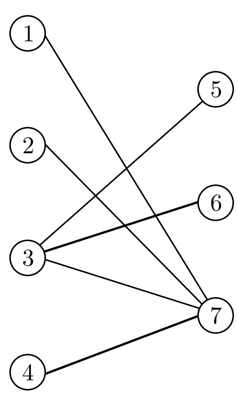


Рис. 17.32. Двудольный граф

Пусть $C \subseteq A$, $C = \{1,2\}$. Тогда $\Gamma(C) = \{7\}$ и $|C| = 2$, $|\Gamma(C)| = 1$. $|\text{ПС}| \leqslant 4 - (2 - 1)$, $|\text{ПС}| \leqslant 3$. Пусть $C = \{1,2,4\}$, тогда $\Gamma(C) = \{7\}$ и $|C| = 2$, $|\Gamma(C)| = 1$, тогда $|\text{ПС}| \leqslant 4 - (3 - 1)$, $|\text{ПС}| \leqslant 2$ совпадает с $|\text{ПС}| = 2$.

Дефицитом $\delta(C)$ подмножества C множества A называется выражение

$$\delta(C) = |C| - |\Gamma(C)|.$$

Дефицит $\delta(C)$ графа $\mathsf{G} = (A, B; U)$ определяется следующим образом:

$$\delta(C) = \max_{c \subseteq C}(|C| - |\Gamma(C)|).$$

Тогда справедливо выражение

$$|\text{ПС}| \leqslant |A| - \beta(G).$$

На этой формуле базируется теорема Кёнига.

Теорема 17.10. *Число ребер в максимальном паросочетании двудольного графа* $\mathsf{G} = (A, B; U)$ *равно*

$$|\text{МПС}| \leqslant |A| - \beta(G).$$

Из теоремы следует, что минимальное число вершин двудольного графа, покрывающих все ребра, равно числу ребер в любом максимальном паросочетании этого графа.

Опишем модифицированную эвристику построения максимального паросочетания в двудольном графе. Исходные данные — двудольный граф, представленный в виде двух частей $\mathsf{G} = (A, B; U)$, и произвольное построение. Оно может состоять, в частности, из одного ребра.

Произведем разбиение подмножества A на две части. В первую включаются вершины, в которые не входят ребра паросочетания. Во вторую включаются вершины, в которые входят ребра паросочетания. Если первая часть пуста, то исходное паросочетание максимально. Это следует из приведенных выше теорем.

Далее среди вершин второй части выбирается вершина x_i с наименьшей локальной степенью. Если таких вершин несколько, то выбирается любая. Для выбранной вершины ищем ребро $(x_i,\ x_j)$, которое не является ребром паросочетания, такое, что $(\exists\, x_i,\ x_j)$ $(x_j \in B)$. Затем продолжаем строить цепь назад по ребру паросочетания. Получаем цепь $(x_i,\ x_j)$ $(x_j,\ x_k)$, далее продолжаем аналогично. Работа заканчивается, когда нет возврата по ребру паросочетания. Далее из исходного паросочетания удаляются ребра, имеющиеся в цепи, и добавляются ребра, которые в нем отсутствуют.

Пример 17.19. Имеем двудольный граф $\mathsf{G} = (A, B; U)$, представленный на рис. 17.33. Задано исходное паросочетание ПС $=$ $= \{(1, 7), (2, 6)\}$. Разобьем подмножество $A = \{1, 2, 3, 4, 5\}$ на две части: $A1 = \{1, 2\}$, $A2 = \{3, 4, 5\}$. Начинаем с вершины 4, имеющей наименьшую локальную степень. Строим цепь $4 \to 6 \leftarrow 2 \to 10$. В этой цепи нет ребер, которые можно добавить в паросочетание. Оставшиеся вершины имеют одинаковую локальную степень, поэтому можем выбрать любую. Выбираем вершину 5 и строим цепь: $5 \to 7 \leftarrow 1 \to 8 \to 5$. После анализа в паросочетание добавляем ребро (5, 8). Строим новую цепь $3 \to 10$. После анализа получаем, что ребро (3, 10) можно добавить к ПС. В результате построено МПС $= \{(1, 7), (2, 6), (5, 8), (3, 10)\}$, показанное жирными линиями на рис. 17.33.

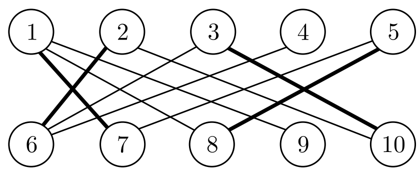


Рис. 17.33. Максимальное паросочетание в двудольном графе $\mathsf{G} = (A, B; U)$

Рассмотрим эвристику построения МПС на основе анализа специальной матрицы смежности и построение в ней «параллельных диагоналей».

Пример 17.20. Пусть задан граф $\mathsf{G} = (A, B; U)$ (рис. 17.34).

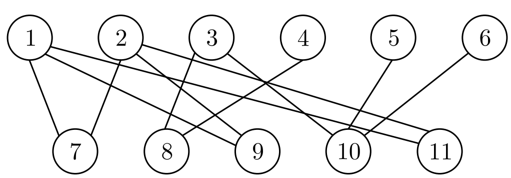


Рис. 17.34. Пример графа $\mathsf{G} = (A, B; U)$

Специальная матрица смежности этого графа запишется:

$$\mathrm{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccc} 7 & 8 & 9 & 10 & 11 \end{array} \\ \left\| \begin{array}{c|c|c|c|c} 1 & & 1 & & 1 \\ \hline 1 & & 1 & & 1 \\ \hline & 1 & & 1 & \\ \hline & 1 & & & \\ \hline & & & 1 & \\ \hline & & & 1 & \end{array} \right\| \end{array}.$$

В матрице можно выделить семь «параллельных диагоналей»:

$$\text{ПС}_1 = \{(1,7)\}, \qquad \text{ПС}_5 = \{(2,9),(3,10)\},$$
$$\text{ПС}_2 = \{(1,9)\}, \qquad \text{ПС}_6 = \{(2,11)\},$$
$$\text{ПС}_3 = \{(1,11)\}, \qquad \text{ПС}_7 = \{(4,8),(6,10)\}.$$
$$\text{ПС}_4 = \{(2,7),(3,8),(5,10)\},$$

Каждая из таких диагоналей является паросочетанием исследуемого графа. Очевидно, при наличии всех единиц по главной диагонали матрицы мы получим максимальное паросочетание. Полученные диагонали можно представить в виде амплитуды (рис. 17.35).

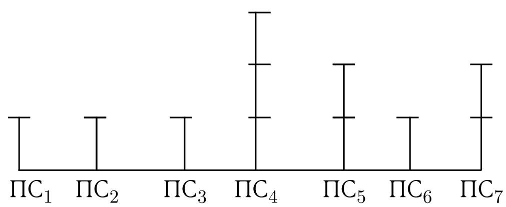


Рис. 17.35. Амплитуда паросочетаний

Для дальнейших исследований выберем амплитуду ПС_4, и на основе суперпозиции с другими диагоналями (ПС_i) будет построено максимальное паросочетание.

Для получения большего числа диагоналей с максимальным числом элементов произведем расширение матрицы:

$$R = \begin{array}{c|ccccc|ccccc|}
 & 7 & 8 & 9 & 10 & 11 & 7 & 8 & 9 & 10 & 11 \\
\hline
1 & 1 & & 1 & & 1 & 1 & & 1 & & 1 \\
2 & 1 & & 1 & & 1 & 1 & & 1 & & 1 \\
3 & & 1 & & 1 & & & 1 & & 1 & \\
4 & & 1 & & & & & & & & \\
5 & & & & 1 & & & & & 1 & \\
6 & & & & 1 & & & & & 1 &
\end{array}.$$

Как видно, получено новое паросочетание

$$\text{ПС}_8 = \text{ПС}_7 \cup \text{ПС}_3 = \{(1,11),(2,7),(3,8),(5,10)\},$$

которое является максимальным.

Заметим, что матрицу можно расширить справа, слева, сверху и снизу. Например, для графа G, заданного матрицей

$$\mathrm{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \!\!\begin{array}{c} \begin{array}{cccc} 5 & 6 & 7 & 8 \end{array} \\ \left\|\begin{array}{c|c|c|c} 1 & 1 & & 1 \\ \hline 1 & 1 & 1 & \\ \hline & 1 & 1 & 1 \\ \hline 1 & & 1 & 1 \end{array}\right\| \end{array},$$

построим расширенные матрицы сверху и снизу. Это дает возможность получить две новые диагонали с максимальным количеством элементов.

$$\mathrm{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \\ 1 \\ 2 \\ 3 \\ 4 \\ \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \!\!\begin{array}{c} \begin{array}{cccc} 5 & 6 & 7 & 8 \end{array} \\ \left\|\begin{array}{c|c|c|c} & & & \\ \hline & & & \\ \hline & & & \\ \hline 1 & & & \end{array}\right\| \\ \\ \left\|\begin{array}{c|c|c|c} 1 & 1 & & \\ \hline 1 & 1 & 1 & \\ \hline & 1 & 1 & 1 \\ \hline & & 1 & 1 \end{array}\right\| \\ \\ \left\|\begin{array}{c|c|c|c} & & & 1 \\ \hline & & & \\ \hline & & & \\ \hline & & & \end{array}\right\| \end{array}.$$

Такое расширение соответствует следующим операциям суперпозиции:

$$\text{ПС}_1 = \{(2,5),(3,6),(4,7)\} \cup \{(1,8)\}, \quad |\text{ПС}_1| = 4;$$
$$\text{ПС}_2 = \{(1,6),(2,7),(3,8)\} \cup \{(4,5)\}, \quad |\text{ПС}_2| = 4.$$

Это позволяет получить два новых максимальных паросочетания ПС_1 и ПС_2.

Тогда, с учетом описанных выше эвристик, алгоритм определения паросочетания в двудольном графе можно записать в следующем виде.

Исходные данные: двудольный граф $\mathsf{G} = (A, B; U)$, $|A|$, $|B|$, матрица смежности.

1. Строится специальная матрица смежности.

2. Определяются диагонали матрицы, соответствующие паросочетаниям графа.

3. Строится амплитуда, и из нее выбирается элемент с максимальным значением.

4. Производится суперпозиция максимальной диагонали с другими путем расширения матрицы справа, слева, сверху и снизу.

5. Определяются максимальные паросочетания.

6. Конец работы алгоритма.

Время работы алгоритма зависит от размера анализируемой матрицы: чем больше размер матрицы, тем больше может потребоваться времени. В задачах на графах размер входа часто измеряется не одним числом, а двумя (число вершин n и число ребер m графа). Для задачи определения паросочетания в двудольном графе необходимо определить время работы алгоритма в худшем случае, среднее время работы алгоритма и время работы алгоритма в лучшем случае.

Вопросы для самоконтроля

1. В каком случае два ребра графа называются независимыми?

2. Что такое паросочетание?

3. Дайте определение максимального паросочетания.

4. Что такое паросочетание в двудольном графе?

5. Что такое полное паросочетание?

6. При каких условиях двудольный граф имеет полное паросочетание?

7. Дайте определение дефицита подмножества или графа.

8. Как определяется число ребер в максимальном паросочетании двудольного графа?

9. В чем состоит основная идея эвристического алгоритма, используемого для построения максимального паросочетания в двудольном графе?

10. В чем состоит основная идея эвристического алгоритма построения МПС на основе «параллельных диагоналей»?

Задания для самостоятельной работы

1. Приведите пример паросочетания в графе.

2. Приведите пример паросочетания в двудольном графе.

3. Для произвольного двудольного графа постройте максимальное паросочетание.

4. Постройте для произвольного графа пример, подтверждающий справедливость теоремы Холла.

5. Приведите пример двудольного графа и покажите на этом примере справедливость теоремы Кёнига.

6. Запишите алгоритм построения максимального паросочетания в двудольном графе на основе разбиения.

7. Для произвольного двудольного графа приведите пример работы алгоритма построения максимального паросочетания на основе разбиения.

8. Для произвольного двудольного графа запишите его матрицу смежности, постройте «параллельные диагонали» и нарисуйте график амплитуды паросочетаний.

9. Запишите алгоритм построения максимального паросочетания в двудольном графе на основе построения «параллельных диагоналей».

10. Для произвольного двудольного графа приведите пример работы алгоритма построения максимального паросочетания на основе «параллельных диагоналей»

17.7. Изоморфизм графов

Большое число комбинаторно-логических и оптимизационных задач на графах требует установления *изоморфизма* или изоморфного вложения между заданными структурами. Данная проблема, как и многие из рассмотренных выше задач, является NP-полной. Поэтому разрабатываются различные эвристики для получения приемлемых практических результатов.

Задача распознавания изоморфизма графов (РИГ) состоит в следующем. Для заданных графов $\mathsf{G}_1 = (X_1, U_1)$ и $\mathsf{G}_2 = (X_2, U_2)$ требуется определить, существует ли взаимно-однозначное отображение φ: $X_1 \to X_2$ такое, что ребро $u = (x, y) \in U_1$ тогда и только тогда, когда $(\varphi(x), \varphi y)) \in U_2$.

В настоящее время известны *полиномиальные алгоритмы* для следующих классов графов: графы ограниченной степени; графы с ограниченной кратностью собственных значений; k-разделимые графы; k-стягиваемые графы и т. д.

Особое внимание заслуживают *сильнорегулярные графы* (СГ). Граф $\mathsf{G} = (X, \rho, n_{11}^1, n_{11}^0)$, $|X| = n$, называется n-вершинным регулярным графом степени ρ, если в нем любая пара смежных вершин имеет n_{11}^1 общих соседей, а любая пара несмежных вершин — n_{11}^0. Известно, что СГ образуют класс наиболее трудных задач для РИГ.

Задача изоморфного вложения графа является *NP-полной задачей*. Она имеет много сходства и в то же время существенно (по сложности) отличается от РИГ. Например, для решения задачи изоморфизма подграфа G_1 с использованием известных алгоритмов РИГ необходимо разработать процедуру выделения в графе G подмножества $X_1 \subset X$, равномощного с множеством вершин X_2 графа G_2.

Данная процедура включает k_1 действий, где $n = |X|$, $n_2 = |X_2|$. Следовательно, k_1 раз надо применять и алгоритм РИГ: $k_1 = \binom{n}{n_2}$.

Поэтому при выделении каждого подграфа G_1 в графе G необходимо выполнять k_2 действий:

$$k_2 = \binom{m_1}{m_2},$$

где m_1 — количество ребер в подграфе G_1, m_2 — в графе G_2, $m_1 > m_2$.

Следовательно, даже если есть полиномиальный алгоритм РИГ, с его помощью невозможно решить задачу изоморфного вложения за полиномиальное время. Она может быть решена за полиномиальное время, если G_1 — лес, а G_2 — дерево.

Наибольшую трудоемкость представляет установление изоморфизма однородных графов, имеющих автоморфные подграфы. Для решения таких задач используются методы разбиения исследуемых графов на различные уровни. При этом ВСА алгоритма снижается с $\mathsf{O}(n!)$ до $\mathsf{O}(k!)$, где n — число элементов в графе, а k — число элементов в наибольшем автоморфном подграфе, т. е. в таком подграфе, где не имеет значения выбор вершин для установления соответствия.

Основная идея таких алгоритмов заключается в следующем. В графах, исследуемых на изоморфизм, выбираются две предполагаемо изоморфные вершины. Относительно них производится разбиение всех оставшихся вершин на два класса. В первый класс включаются вершины, смежные предполагаемо изоморфным, а во второй нет.

Пример 17.21. На рис. 17.36 показаны два графа: G и G'. $\mathsf{G} = (X, U)$, $\mathsf{G}' = (X', U')$, $|X| = |X'| = 6$, $|U| = |U'| = 9$, локальные степени всех вершин графа равны трем. Матрицы смежности графов G и G' приведены ниже.

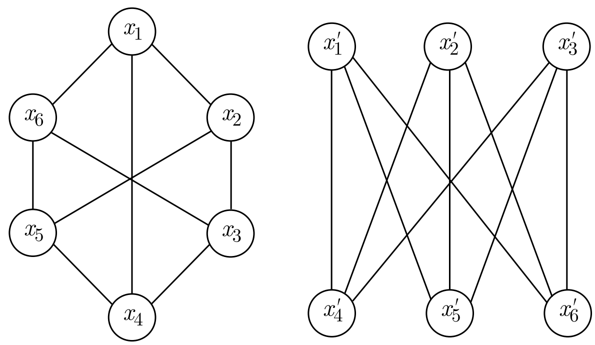


Рис. 17.36. Графы G и G′

$$
\mathrm{R}_{\mathsf{G}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{|c|c|c|c|c|c|}
\multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} \\ \hline
0 & 1 & & 1 & & 1 \\ \hline
1 & 0 & 1 & & 1 & \\ \hline
 & 1 & 0 & 1 & & 1 \\ \hline
1 & & 1 & 0 & 1 & \\ \hline
 & 1 & & 1 & 0 & 1 \\ \hline
1 & & 1 & & 1 & 0 \\ \hline
\end{array}
\quad
\begin{array}{|c|}
\multicolumn{1}{c}{} \\ \hline
3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline
\end{array}\ ,
$$

$$
\mathrm{R}_{\mathsf{G}'} = \begin{array}{c} \\ x_1^1 \\ x_2^1 \\ x_3^1 \\ x_4^1 \\ x_5^1 \\ x_6^1 \end{array}
\begin{array}{|c|c|c|c|c|c|}
\multicolumn{1}{c}{x_1^1} & \multicolumn{1}{c}{x_2^1} & \multicolumn{1}{c}{x_3^1} & \multicolumn{1}{c}{x_4^1} & \multicolumn{1}{c}{x_5^1} & \multicolumn{1}{c}{x_6^1} \\ \hline
0 & & & 1 & 1 & 1 \\ \hline
 & 0 & & 1 & 1 & 1 \\ \hline
 & & 0 & 1 & 1 & 1 \\ \hline
1 & 1 & 1 & 0 & & \\ \hline
1 & 1 & 1 & & 0 & \\ \hline
1 & 1 & 1 & & & 0 \\ \hline
\end{array}
\quad
\begin{array}{|c|}
\multicolumn{1}{c}{} \\ \hline
3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline 3 \\ \hline
\end{array}\ .
$$

Необходимо установить изоморфизм графов G и G′. Другими словами, необходимо определить, существует ли отношение эквивалентности $X \Leftrightarrow X'$, $U \Leftrightarrow U'$ такое, что если ребро $(x_i, x_j) \Leftrightarrow$ ребру (x_i', x_j'), то $x_i \Leftrightarrow x_i'$, $x_j \Leftrightarrow x_j'$, $x_i, x_j \in X$, $x_i', x_j' \in X'$.

Покажем на примере реализацию эвристики разбиения для установления изоморфизма однородных графов.

Пример 17.22. Выберем одну вершину в графе G и одну в графе G′ и относительно них выполним указанное выше разбиение:

$$\{x_1\}\{x_2, x_4, x_6\}_{1+}\{x_3, x_5\}_{1-} \quad (\mathsf{G}),$$
$$\{x'_1\}\{x'_4, x'_5, x'_6\}_{1'+}\{x'_2, x'_3\}_{1'-} \quad (\mathsf{G}').$$

В данном разбиении вершины x_2, x_4, x_6 смежны вершине x_1, а вершины x_3, x_5 не смежны вершине x_1 графа G. Аналогичное разбиение выполнено для графа G$'$. Здесь считается, что вершины x_1 и x'_1 предполагаемо изоморфны (П-изоморфны). Далее, выбираются подмножества меньшей мощности и внутри них проверяется смежность вершин. В нашем примере вершины x_3, x_5 и x'_2, x'_3 не смежны. Поэтому процесс разбиения продолжается.

$$\{x_1\}\{x_3\}_{1-}\{\{x_2, x_4, x_6\}_{3+}\}_{1+}\{\{x_5\}_{3-}\}_{1-},$$
$$\{x'_1\}\{x'_2\}_{1'-}\{\{x'_4, x'_5, x'_6\}_{2'+}\}_{1'+}\{\{x'_3\}_{2'-}\}_{1'-}.$$

Продолжая процесс аналогично, получим, что граф G изоморфен графу G$'$. Подстановка вершин запишется

$$t = \begin{Bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1' & 4' & 2' & 5' & 3' & 6' \end{Bmatrix}.$$

В рассматриваемом примере подмножества вершин $\{x_2, x_4, x_6\}$ и $\{x'_4, x'_5, x'_6\}$ являются автоморфными, т. е. каждая вершина в одном подграфе может быть изоморфна любой вершине второго подграфа.

G$'$ преобразовали в G на основе подстановки

$$\mathrm{R}_{\mathsf{G}\Pi} = \begin{array}{c|c|c|c|c|c|c|c|c|}
 & 1' & 4' & 2' & 5' & 3' & 6' & & \\
\hline
1(1') & 0 & 1 & & 1 & & 1 & & 3 \\
\hline
2(4') & 1 & 0 & 1 & & 1 & & & 3 \\
\hline
3(2') & & 1 & 0 & 1 & & 1 & & 3 \\
\hline
4(5') & 1 & & 1 & 0 & 1 & & & 3 \\
\hline
5(3') & & 1 & & 1 & 0 & 1 & & 3 \\
\hline
6(6') & 1 & & 1 & & 1 & 0 & & 3 \\
\hline
\end{array}.$$

Следовательно, G изоморфен G$'$.

В алгоритмах такого типа необходим перебор между элементами автоморфных подграфов. Причем, как следует из рассмотрения примера, такой перебор здесь выполняется последовательно для всех вершин, кроме последней в подграфе.

Пример 17.23. На рис. 17.37 показаны нетривиальные, неоднородные графы G и G$'$. В этих графах $|X| = |X'| = 7$, $|U| = |U'| = 8$, $\rho_1 = 1$, $\rho_2 = 3$, $\rho_3 = 3$, $\rho_4 = 3$, $\rho_5 = 3$, $\rho_6 = 2$, $\rho_7 = 1$;

$\rho_a = 1$, $\rho_b = 3$, $\rho_c = 3$, $\rho_d = 3$, $\rho_e = 3$, $\rho_f = 2$, $\rho_g = 1$. Вершина 1 может быть П-изоморфна вершине a или g. В первом случае нас ждет тупик, а во втором успех. Мы предположили, что вершина 1 П-изоморфна вершине a. Тогда на основе связности графа и эвристики разбиения следует, что вершина 7 П-изоморфна вершине g. Из этого вытекает, что вершина 6 П-изоморфна вершине f.

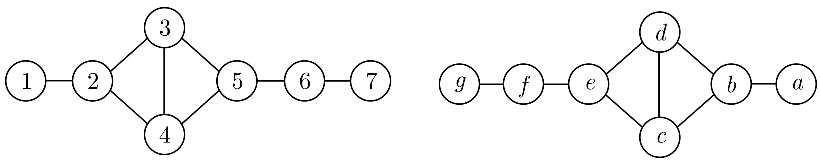


Рис. 17.37. Нетривиальные графы G и G'

$$
\mathrm{R}_{\mathsf{G}} = \begin{array}{c|c|c|c|c|c|c|c|c|c|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & & \\
\cline{2-8}\cline{10-10}
1 & 0 & 1 & & & & & & & 1 \\
\cline{2-8}\cline{10-10}
2 & 1 & 0 & 1 & 1 & & & & & 3 \\
\cline{2-8}\cline{10-10}
3 & & 1 & 0 & 1 & 1 & & & & 3 \\
\cline{2-8}\cline{10-10}
4 & & 1 & 1 & 0 & 1 & & & & 3 \\
\cline{2-8}\cline{10-10}
5 & & & 1 & 1 & 0 & 1 & & & 3 \\
\cline{2-8}\cline{10-10}
6 & & & & & 1 & 0 & 1 & & 2 \\
\cline{2-8}\cline{10-10}
7 & & & & & & 1 & 0 & & 1 \\
\cline{2-8}\cline{10-10}
\end{array}\;,
$$

$$
\mathrm{R}_{\mathsf{G}'} = \begin{array}{c|c|c|c|c|c|c|c|c|c|}
 & a & b & c & d & e & f & g & & \\
\cline{2-8}\cline{10-10}
a & 0 & 1 & & & & & & & 1 \\
\cline{2-8}\cline{10-10}
b & 1 & 0 & 1 & 1 & & & & & 3 \\
\cline{2-8}\cline{10-10}
c & & 1 & 0 & 1 & 1 & & & & 3 \\
\cline{2-8}\cline{10-10}
d & & 1 & 1 & 0 & 1 & & & & 3 \\
\cline{2-8}\cline{10-10}
e & & & 1 & 1 & 0 & 1 & & & 2 \\
\cline{2-8}\cline{10-10}
f & & & & & 1 & 0 & 1 & & 2 \\
\cline{2-8}\cline{10-10}
g & & & & & & 1 & 0 & & 1 \\
\cline{2-8}\cline{10-10}
\end{array}\;.
$$

Далее необходимо установить, существует ли соответствие между подмножествами автоморфных вершин $\{2, 3, 4, 5\}$ и $\{b, c, d, e\}$. В этом случае, как отмечалось выше, в общем, необходимо выполнить полный перебор между этими подмножествами вершин. Для этих подмножеств это 4!

Выполняя такие преобразования, получим, что вершина 2 П-изоморфна вершине b, вершина 3 П-изоморфна вершине d, 4 — c и 5 — e. Отсюда следует, что графы G и G' (рис. 17.37) изоморфны, а подстановка изоморфизма, переводящая один граф G в граф G', запишется

$$t = \begin{Bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ a & b & d & c & e & f & g \end{Bmatrix}.$$

Следовательно, G изоморфен G'.

Как видно, наличие вершин с различными локальными степенями упрощает процесс РИГ. Это позволяет выбирать начальные условия для эвристики разбиения. В примере (рис. 17.37) всего два способа для выбора разбиения, в то время как в графах (рис. 17.36) шесть возможных способов разбиения. В однородных графах все степени вершин одинаковы, поэтому начальные условия выбираются произвольно случайным образом.

В этой связи при установлении изоморфизма в однородных графах ВСА в самом лучшем случае $\mathsf{O}(n)$, в самом худшем случае $\mathsf{O}(n!)$. Если графы неизоморфны, то за одну итерацию установить результат невозможно. Необходимо провести сравнение на изоморфизм одной случайно выбранной вершины из одного графа со всеми остальными вершинами другого графа.

Пример 17.24. Пусть заданы два графа (рис. 17.38): G и G'. При этом $\mathsf{G} = (X, U)$, $\mathsf{G}' = (X', U')$, $|X| = |X'| = 6$, $|U| = |U'| = 9$, локальные степени всех вершин графа равны трем.

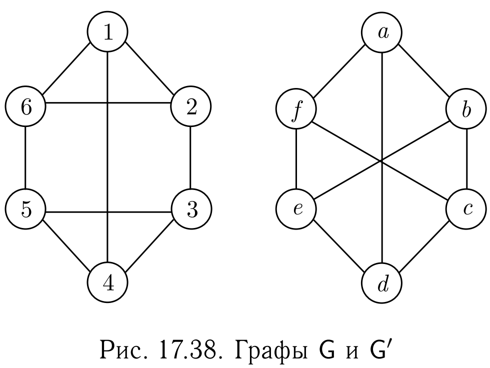


Рис. 17.38. Графы G и G'

Для РИГ применим эвристику разбиения. Предположим, что вершина 1 графа G П-изоморфна вершине a графа G'. Тогда получим

$$\{1\} \quad \{2,4,6\}_{1+} \quad \{3,5\}_{1-} \quad (\mathsf{G}),$$

$$\{a\} \quad \{b,d,f\}_{1'+} \quad \{e,c\}_{1'} \quad (\mathsf{G}').$$

Для дальнейшего анализа выбираем соответствующие подмножества наименьшей мощности $\{3,5\}$ и $\{e,c\}$. Вершины (3, 5) в графе G смежны, а вершины (c,e) в графе G′ — нет. Следовательно, вершина 1 не может быть изоморфна вершине a. В этой связи необходимо провести аналогичные операции для проверки изоморфизма вершины 1 с остальными вершинами графа G′. Далее получим

$$\{1\} \quad \{2,4,6\}_{1+} \quad \{3,5\}_{1-} \quad (\mathsf{G}),$$

$$\{b\} \quad \{a,c,e\}_{1'+} \quad \{d,f\}_{1'-} \quad (\mathsf{G}').$$

Вершины $(3,5)$ в графе G смежны, а вершины (d,f) в графе G′ — нет. Следовательно, вершина 1 не может быть изоморфна вершине b. Продолжая аналогичные построения, имеем, что вершина 1 не может быть изоморфна вершинам c, d, e. Наконец, предположим, что вершина 1 графа G П-изоморфна вершине f графа G′. Тогда получим

$$\{1\} \quad \{2,4,6\}_{1+} \quad \{3,5\}_{1-} \quad (\mathsf{G}),$$

$$\{f\} \quad \{a,c,e\}_{1'+} \quad \{b,d\}_{1'-} \quad (\mathsf{G}').$$

Вершины $(3,5)$ в графе G смежны, а вершины (b,d) в графе G′ — нет. Следовательно, вершина 1 не может быть изоморфна вершине b. Проанализированы все вершины графа G′ на предмет изоморфизма с вершиной 1 графа G. Во всех случаях ответ отрицательный. Поэтому граф G неизоморфен графу G′.

Пример 17.25. Рассмотрим пример работы алгоритма распознавания изоморфизма для графов G и G′ (рис. 17.39). $\mathsf{G} = (X,U)$, $\mathsf{G}' = (X',U')$, $|X| = |X'| = 10$, $|U| = |U'| = 15$, локальные степени всех вершин графа равны трем. Предположим, что вершина 1 графа G П-изоморфна вершине a графа G′. Тогда получим

$$\{1\} \quad \{2,5,6\}_{1+} \quad \{3,4,7,8,9,10\}_{1-} \quad (\mathsf{G}),$$

$$\{a\} \quad \{b,e,f\}_{1'+} \quad \{c,d,h,g,k,i\}_{1'-} \quad (\mathsf{G}').$$

Для дальнейшего анализа выбираем соответствующие подмножества наименьшей мощности $\{2,5,6\}$ и $\{b,e,f\}$. Предпо-

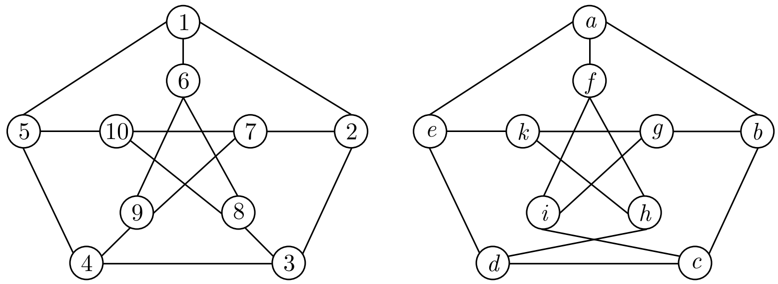


Рис. 17.39. Графы G и G″

ложим, что вершина 2 П-изоморфна вершине b. В этом случае получим

$$\{1\} \quad \{2\}_{1+}[\{5,6\}_{1+}]_{1-} \quad [\{7,3\}_{1+}, \{4,8,9,10\}_{1-}]_{1-} \quad (\mathsf{G}),$$

$$\{a\} \quad \{b\}_{1'+}[\{e,f\}_{1'+}]_{1'-} \quad [\{g,c\}_{1'+}, \{d,h,g,k,i\}_{1'-}]_{1'-} \quad (\mathsf{G}').$$

Как видно, П-изоморфизм сохраняется. Продолжая далее, запишем

$$\{1\} \quad \{2\}_{1+}[\{5,6\}_{1+}]_{1-}$$
$$[[\{7\}_{1+}, \{3\}_{1+}]_{1+}, [\{9,10\}_{1-}, \{4,8\}_{1-}]_{1-}]_{1-} \quad (\mathsf{G}),$$
$$\{a\} \quad \{b\}_{1'+}[\{e,f\}_{1'+}]_{1'-}$$
$$[[\{g\}_{1'+}, \{c\}_{1'+}]_{1'+}, [\{i,k\}_{1'-}, \{h,d,\}_{1'-}]_{1'-}]_{1'-} \quad (\mathsf{G}').$$

Вершины (h,d) в графе G′ смежны, а вершины $(4,8)$ в графе G — нет. Следовательно, вершина 1 не может быть изоморфна вершине a, вершина 2 — b, вершина 7 — g и вершина 3 — c. Продолжая аналогично, получим, что граф G не изоморфен графу G′.

Определение изоморфизма графов позволяет определять одинаковые графы, которые расположены на плоскости по-разному.

Вопросы для самоконтроля

1. В каком случае два графа называются изоморфными?
2. В чем состоит задача распознавания изоморфизма графов?
3. К какому классу задач относятся задачи распознавания изоморфизма?
4. Какова временная сложность алгоритмов распознавания изоморфизма?
5. Каким образом решается задача определения изоморфизма двух графов?

6. Что означает запись о том, что два графа П-изоморфны?

7. Для чего может использоваться процедура определения изоморфизма графов?

8. В чем состоит основная идея приведенных алгоритмов определения изоморфизма?

9. Каким образом выбираются вершины в алгоритме определения изоморфизма для однородных графов?

10. Что такое изоморфное вложение графов?

Задания для самостоятельной работы

1. Приведите пример изоморфных графов.
2. Приведите пример неизоморфных графов.
3. Приведите пример изоморфного вложения графов.
4. Приведите пример однородных графов.
5. Запишите алгоритм определения изоморфизма графов.
6. Для двух изоморфных графов запишите подстановку вершин.
7. Приведите пример автоморфных подмножеств вершин графов.
8. Запишите алгоритм определения изоморфизма однородных графов.
9. Приведите пример работы алгоритма определения изоморфизма для двух произвольных графов.
10. Запишите алгоритм построения подстановки вершин.

Тестовые задания к модулю 4

1. Множество линий, соединяющих любые пары вершин графа — это ...

а) множество ребер
б) множество вершин
в) множество петель
г) множество дуг
д) множество элементов

2. Совокупность вершин, смежных с вершиной x_i графа G, — это ...

а) список связности
б) матрица инцидентности
в) список смежности
г) матрица связности
д) матрица смежности

3. Когда вершины x_i и x_j связаны между собой ребром u_{ij}, это означает следующее:

а) вершина x_i инцидента ребру u_{ij}
б) вершина x_i смежна ребру u_{ij}
в) вершина x_i смежна с вершиной x_j
г) вершина x_i инцидента вершине x_j
д) вершина x_i связана с ребром u_{ij}

4. Когда два ребра инцидентны одной вершине x графа G, то справедливы высказывания:

1) вершина смежна с этими ребрами
2) эти ребра смежны с вершиной x
3) эти ребра смежны между собой
4) вершина x инцидента этим ребрам
5) ребра инциденты вершине x

5. Когда ребро u_{ij} соединяет вершины x_i и x_j графа G, то справедливы высказывания:

- вершины x_i и x_j инцидентны ребру u_{ij}
- вершины x_i и x_j смежны с ребром u_{ij}
- ребро u_{ij} смежно вершине x_j
- вершина x_j инцидента вершине x_i
- ребро u_{ij} смежно вершине x_i

6. Граф, у которого всем вершинам соответствуют некоторые целые числа, — это ...

а) числовой граф
б) целый граф

в) помеченный граф
г) строгий граф
д) регулярный граф

7. Мультичисло графа — это ...

а) число ребер
б) число вершин
в) максимальное число ребер, соединяющих две вершины
г) число ребер, соединяющих две вершины
д) число циклов

8. Пусть задан граф $\mathsf{G} = (X, U)$. При $X = \varnothing$ и $U = \varnothing$ можно сказать, что граф G — это ...

а) мультиграф
б) нуль-граф
в) пустой граф
г) полный граф
д) регулярный граф

9. Граф $\mathsf{G} = (X, U)$, в котором любые две вершины соединены ребром, — это ...

а) мультиграф
б) нуль-граф
в) пустой граф
г) полный граф
д) регулярный граф

10. Граф $\mathsf{G} = (X, U)$, в котором вершины имеют одинаковую локальную степень, — это ...

а) мультиграф
б) нуль-граф
в) пустой граф
г) полный граф
д) регулярный граф

11. Граф $\mathsf{G}' = (X', U')$, полученный путем удаления из графа $\mathsf{G} = (X, U)$ некоторого числа вершин вместе с инцидентными им ребрами, — это ...

а) подграф
б) разрез
в) суграф
г) дополнение
д) блок

12. Конечная последовательность ребер вида $S = (x_0, x_1), (x_1, x_2), \ldots, (x_{k-1}, x_k)$, где x_0 и x_k — соответственно начальная и конечная вершины, это ...

а) маршрут
б) цепь
в) цикл
г) простая цепь
д) простой цикл

13. Длиной маршрута в графе называется ...

а) расстояние между начальной и конечной вершинами
б) сумма стоимостей
в) число ребер маршрута
г) число вершин маршрута
д) число вершин графа

14. Для подсчета числа маршрутов длины n в графе необходимо выполнить следующие действия:

- возвести матрицу смежности графа в степень n
- возвести матрицу инцидентности графа в степень n
- перебрать возможные варианты
- перемножить ребра
- перемножить вершины

15. Цикл, не содержащий повторяющихся вершин, кроме первой и последней, — это ...

а) цепь
б) цикл
в) простая цепь
г) простой цикл
д) эйлеров цикл

16. Две вершины графа G являются связными, если выполняются следующие условия:

а) существует ребро инцидентное этим вершинам
б) существует маршрут, где эти вершины являются конечными
в) вершины соединены n ребрами
г) вершины являются смежными
д) граф полный

17. Граф, между любыми двумя вершинами которого можно построить маршрут, — это ...

а) связный граф
б) регулярный граф

в) полный граф
г) смежностный граф
д) простой граф

18. Цикл, проходящий по всем ребрам графа один раз, — это ...

а) простой цикл
б) полный цикл
в) эйлеров цикл
г) гамильтонов цикл
д) фундаментальный цикл

19. Задача нахождения в полном графе с весами на ребрах гамильтонова цикла, у которого сумма ребер минимальна, носит название ...

- задача Петри
- задача Дирака
- задача коммивояжера
- транспортная задача
- задача о назначениях

20. Начальная вершина графа T — это ...

а) дерево
б) лес
в) ветвь
г) кустарник
д) корень

21. Ребра, выходящие из начальной вершины графа T, — это ...

а) дерево
б) лес
в) ветви
г) кустарник
д) корни

22. Число покрывающих деревьев в полном графе K_4 равно ...

а) 4
б) 8
в) 16
г) 2
д) 32

23. Для графа T, являющегося деревом, верны следующие утверждения:

• граф T имеет $n-1$ ребро

• граф T не содержит циклов

• граф T не связен

• любые две вершины графа T соединены единственным циклом

• любые две вершины графа T соединены циклом

24. Длина кратчайшей цепи, соединяющей вершины x_i и x_j графа G, — это ...

а) протяженность вершин

б) удаление вершин

в) расстояние между вершинами

г) последовательность вершин

д) удаленность вершин

25. Наименьшее число ребер, которые нужно удалить из графа, чтобы он стал деревом, — это ...

а) цикломатическое число

б) хроматическое число

в) число внутренней устойчивости

г) число внешней устойчивости

д) число внутренней полноты

26. Цикломатическое число определяется по формуле $\gamma(\mathbf{G}) = m - n + k$, где m — число вершин графа G, n — число его ребер, а k — это ...

а) число компонент связности

б) число циклов

в) число цепей

г) число простых цепей

д) число простых циклов

27. Разбиение множества вершин X на k непересекающихся классов таких, что внутри каждого класса не должно содержаться смежных вершин, — это ...

а) разрез графа

б) сечение графа

в) раскраска графа

г) пересечение графа

д) разбиение графа

28. При раскраске графа одним цветом красятся вершины, отвечающие следующим требованиям:

а) они являются сильно связными

б) они являются слабо связным

в) они являются связными

г) они являются не связными
д) они являются частично связными

29. Подмножество, любые две вершины которого являются несмежными, может называться ...

а) внутренне устойчивое
б) внешне устойчивое
в) максимально внутренне устойчивое
г) независимое
д) полное

30. Подмножество, вершины которого смежны со всеми остальными вершинами графа, — это подмножество ...

а) полное
б) внутренне устойчивое
в) внешне устойчивое
г) независимое
д) несвязное

31. Для определения числа внешней устойчивости необходимо определить ...

- мощность минимального внешне устойчивого подмножества
- число внутренней устойчивости
- число минимальных внешне устойчивых подмножеств
- число внешне устойчивых подмножеств
- число ребер графа

32. Подмножество, являющееся одновременно внутренне и внешне устойчивым, — это ...

а) клика
б) ядро
в) доминирующее подмножество
г) независимое подмножество
д) базовое подмножество

КРИТЕРИИ ОЦЕНКИ

Для оценки уровня полученных знаний при выполнении разработанных тестовых заданий по модулю 4 предлагается использовать следующую шкалу:

85–100 % правильных ответов — оценка «отлично»;

70–84 % правильных ответов — оценка «хорошо»;

55–69 % правильных ответов — оценка «удовлетворительно»;

менее 55 % правильных ответов — оценка «неудовлетворительно».

Глоссарий к модулю 4

Глава 15

Блок графа — это его максимальный неразделимый подграф.

Вершинная связность графа — это наименьшее число вершин, удаление которых приводит к несвязному графу. Связность в графе с точкой сочленения равна 1. Связность в несвязном графе равна 0.

Взвешенный граф — это граф с весами на ребрах.

Граф — это объект, состоящий из двух множеств: точек и линий, которые находятся между собой в некотором отношении. Понятие *графа* опирается на понятие множества.

Граф — математический объект, который состоит из множества вершин и множества ребер или дуг, находящихся между собой в некотором отношении.

Длина маршрута определяется числом ребер, входящих в маршрут.

Если ребро соединяет две вершины графа, то говорят, что ребро *инцидентно* этим вершинам или вершины *инцидентны* данному ребру.

Конечный граф — граф, у которого множества вершин и ребер представляют собой конечные множества.

Локальная степень или просто *степень* вершины графа — это число ребер, инцидентных данной вершине. Локальная степень вершины x_i обозначается $\rho(x_i)$ или ρ_i.

Маршрут в графе — это некоторая конечная последовательность ребер, имеющая начальную и конечную вершину.

Граф также может быть задан с помощью *матрицы инцидентности*. *Матрица инцидентности* — это прямоугольная таблица, состоящая из нулей и единиц. Строкам матрицы соответствуют вершины, а столбцам — ребра графа. Причем на пересечении строки столбца ставится единица, если вершина, соответствующая данной строке, инцидентна ребру, соответствующему данному столбцу.

При матричном способе задания графов можно использовать *матрицу смежности*. *Матрица смежности* — это квадратная таблица, состоящая из нулей и единиц. Строкам и столбцам матрицы соответствуют вершины графа, причем на пересечении строки столбца ставится единица, если вершины, соответствующие данной строке и столбцу, являются смежными (т. е. соединены ребром).

Множество вершин — это множество точек графа.

Множество ребер или *дуг* — множество линий, соединяющих любые пары вершин графа.

Мультиграф — граф, у которого существует хотя бы одна пара вершин, соединяемых несколькими ребрами. Ребра, соединяющие одну и ту же пару вершин, называются *кратными*. Максимальное число кратных ребер в мультиграфе называется его *мультичислом*.

Неориентированный граф (*неорграф*) — граф, состоящий из подмножества вершин и подмножества неориентированных ребер.

Неразделимый граф — это связный, непустой, не имеющий точек сочленения граф.

Нуль-граф — граф, у которого множество ребер является пустым, а все его вершины изолированные.

Ориентированный граф (*орграф*) — это граф, состоящий из подмножества вершин и подмножества ориентированных ребер (дуг).

Отношение связности является отношением эквивалентности.

Перешеек (*мост*) в графе — это разрез, состоящий из одного ребра.

Подграф — часть графа, у которой все вершины и ребра принадлежат исходному графу, а ребра подграфа соединяют только вершины подграфа. Подграф можно получить, удалив из исходного графа одну или несколько вершин с инцидентными им ребрами.

Подмножество дуг — подмножество ориентированных линий, для которых существен порядок соединения вершин. Каждая дуга определяется упорядоченной парой (кортежем длины два) вершин, которые она соединяет.

Подмножество петель — подмножество линий, каждая из которых выходит и входит в одну и ту же соответствующую этой линии вершину. Каждая петля может определяться как упорядоченной, так и неупорядоченной парой вершин.

Подмножество ребер графа — подмножество неориентированных линий, для которых несуществен порядок соединения вершин. Каждое ребро определяется неупорядоченной парой вершин, которые оно соединяет.

Полный граф — граф, между любой парой вершин которого имеется ребро. Полный граф обозначается K_n.

Правильный разрез графа — это разрез, который разбивает исходный граф точно на две компоненты связности.

Тривиальными разрезами считают разрез, в который входят все ребра графа, и разрез, не содержащий ни одного ребра графа.

Простая цепь (*цикл*) — это цепь (цикл), которая не содержит повторяющихся вершин, кроме первой и последней.

Простой граф — это граф, не содержащий петель и кратных ребер.

Разбиение графа представляет собой разбиение на непересекающиеся классы при условии, что ребра графа будут соединять только вершины внутри этих классов.

Разрез графа — это такое разбиение графа на два подмножества, при котором во множестве ребер графа одни концевые вершины принадлежат первому подмножеству вершин, а другие — второму.

Реберная связность — это наименьшее число ребер, удаление которых приводит к несвязному или тривиальному графу.

Регулярный граф — граф, все вершины которого имеют одинаковые локальные степени.

Две произвольные вершины графа называются *связными*, если существует маршрут, в котором концевыми будут эти вершины.

Граф является *связным*, если любые две его вершины связаны. В противном случае граф не связен, а каждый из составляющих его подграфов называется *компонентой связности*.

Смежностный (*двойственный*) *граф* — граф, вершинами которого являются ребра исходного графа, а ребрами — пары ребер исходного графа, имеющие общую вершину. Для построения двойственного графа на каждом ребре исходного графа выбирают среднюю точку и считают ее вершиной.

Любые две вершины графа называются *смежными*, если существуют соединяющие эти вершины ребра. Аналогично если два ребра инцидентны одной и той же вершине, то они также называются *смежными*.

Смешанный граф — это граф, содержащий подмножества ребер, дуг и петель.

Основные *способы задания графа* — геометрический, аналитический и матричный.

Суграф — это часть графа, содержащая обязательно все вершины исходного графа, а также все либо часть его ребер.

Точка сочленения — вершина графа, после удаления которой граф распадается на связные компоненты.

Треугольник — цикл длины 3.

Цепь — это маршрут, в котором нет повторяющихся ребер.

Цикл — это замкнутая цепь, в которой первая и последняя вершины совпадают.

Циклический граф (*цикл*) — связный регулярный граф степени 2.

Глава 16

Гамильтонов цикл — это цикл, проходящий по всем вершинам графа один раз. При этом сам граф называют *гамильтоновым графом*.

Граф подразбиений $S(\mathsf{G})$ — это граф, каждое ребро которого подразбито путем введения дополнительной вершины.

Дерево — связный граф без циклов. В дереве любые две вершины связаны единственной цепью. Любое дерево имеет $n-1$ ребро.

Диаметр графа — это максимальное расстояние между любыми двумя его вершинами.

Диаметрально простые цепи — это кратчайшие простые цепи, связывающие две вершины графа с максимальным расстоянием между ними.

Длиннейшие простые цепи в графе называют *диаметральными по протяженности*. Их длина называется *диаметром протяженности*. Для каждой вершины существуют самые длинные простые цепи с концами в этой вершине. Их длина называется *числом протяженности* для данной вершины графа.

Длина цепи — это число входящих в нее ребер.

Жорданова кривая — непрерывная кривая на плоскости, не имеющая самопересечений. *Замкнутая жорданова кривая* — это жорданова кривая, начало и конец которой совпадают.

Жордана теорема. Если L *— замкнутая жорданова кривая, а* x_i, x_j *— две различные точки, расположенные на ней, то любая жорданова кривая, соединяющая* x_i *и* x_j, *должна лежать целиком внутри* L *или вне* L *(за исключением точек* x_i, x_j*) или пересекать* L *в некоторой точке, отличной от точек* x_i, x_j.

Задача о коммивояжере формулируется следующим образом. Имеется n городов, расстояния между которыми известны. Коммивояжеру необходимо посетить каждый город по одному разу и вернуться в исходный пункт, пройдя при этом минимально возможное расстояние.

Задача о лабиринте в терминах теории графов формулируется как задача отыскания в связном графе такого маршрута, который начинается в заданной вершине и приводит в искомую вершину, причем маршрут должен содержать минимальное число ребер.

Задача построения кратчайшего покрывающего дерева, когда при соединении множества вершин на плоскости (обычно в областях прямоугольной конфигурации) разрешается использование дополнительных точек соединения, называется *задачей Штейнера* (ЗШ). Дополнительные точки, вводимые в процессе построения кратчайшего покрывающего дерева, называют *точками Штейнера* (ТШ).

Корень — это начальная вершина, из которой выходят ребра — *ветви* дерева.

Кэли теорема. Существует ровно n^{n-2} различных помеченных деревьев с n вершинами.

Лес — множество деревьев графа.

Максимальным удалением в графе от некоторой произвольной вершины называется максимальное расстояние между рассматриваемой вершиной и любой другой вершиной графа.

Матрица геометрии — это часть матрицы расстояний, из которой исключаются элементы, если вершины, которые им соответствуют, являются не смежными в рассматриваемом графе. Для построения матрицы геометрии необходимо каждый элемент матрицы расстояний умножить на соответствующий элемент матрицы смежности.

Функцию расстояний для графа обычно задают *матрицей расстояний* или ее списком.

Метрика графа основана на понятии расстояния.

Покрывающее или *остовное дерево* — дерево, число вершин которого равно числу вершин графа, из какого выделено это дерево. Число остовных деревьев в полном графе равно n^{n-2}.

Полуэйлеров граф — это граф, в котором существует незамкнутая цепь, проходящая через каждое ребро графа только один раз. Связный граф G является *полуэйлеровым*, когда в нем не более двух вершин имеют нечетные локальные степени.

Протяженность между вершинами графа — это максимальная из длин, связывающих эти вершины.

Радиальная цепь — любая кратчайшая цепь от центра до максимально удаленной от него вершины графа.

Радиус графа — это максимальное удаление от центра графа.

Расстояние между вершинами графа — это длина кратчайшей цепи, соединяющей эти вершины.

Соседними называют вершины и ребра графа в случае, если они смежны и инцидентны.

Тотальный граф — граф, у которого множеством вершин является объединение множества вершин и множества ребер исходного графа, и две вершины считаются смежными тогда и только тогда, когда они являются соседними в исходном графе.

Центром графа называется вершина графа, для которой величина максимального удаления принимает минимальное значение.

Эйлеров граф — это конечный граф, который является связным, и все локальные степени его вершин четные.

Эйлеров цикл — это цикл, который проходит по всем ребрам графа один раз. Если в графе существует эйлеров цикл, то, проходя по его ребрам, можно нарисовать эйлеров граф на бумаге, не отрывая карандаша.

Глава 17

Внешне устойчивое подмножество — это подмножество вершин графа, которые смежны со всеми остальными вершинами графа. Внешне устойчивое подмножество называется *минимальным*, если удаление из него произвольной вершины делает его внешне неустойчивым.

Внутренне устойчивое подмножество графа — это подмножество, любые две вершины которого являются не смежными.

Гиперграф $\mathsf{H} = (X, E)$ — это объект, который состоит из множества вершин X и множества ребер E, причем каждое ребро $l_i \in E$ представляет собой некоторое подмножество множества вершин, т. е. $l_i \subseteq X$.

Области, определяемые плоским графом, называются его *внутренними гранями* или просто *гранями*. Неограниченная область называется *внешней гранью*.

Два графа *гомеоморфны* или тождественны с точностью до вершины степени 2, если они могут быть получены из одного и того же графа включением (добавлением) в его ребра новых вершин степени 2.

Гиперграф H^* называется *двойственным* гиперграфу $\mathsf{H} = (X, E)$, если вершинами H^* являются ребра H, а ребрами — вершины H.

Двудольный (*бихроматический*) *граф* — это граф, раскрашиваемый двумя красками, причем множество вершин такого графа разбито на два непересекающихся подмножества, а ребра графа соединяют между собой только вершины, принадлежащие разным подмножествам.

Гиперграф $\mathsf{H} = (X, E)$ можно представить в виде *двудольного графа Кёнига* $\mathsf{K}(\mathsf{H}) = (X \cup E, V)$, где X — первое подмножество вершин двудольного графа, соответствующее множеству вершин гиперграфа H; E — второе подмножество вершин двудольного графа, соответствующее множеству ребер гиперграфа H; V — множество ребер двудольного графа, устанавливающее смежность между двумя множествами вершин X и E в графе $\mathsf{K}(\mathsf{H})$ в соответствии с инцидентностью вершин и ребер гиперграфа H.

Гиперграф $\mathsf{H} = (X, E)$ считается *деревом* в случае, когда его ребра имеют не более одной общей вершины.

Крупность (*шероховатость*, *зернистость*) графа — наибольшее число непланарных подграфов в исходном графе, не пересекающихся по ребрам (не имеющих общих ребер).

Максимальное внутренне устойчивое подмножество (*МВУП*) *или независимое* (*НП*) *подмножество* — это подмножество графа такое, что добавление к нему любой новой вершины делает его внутренне неустойчивым.

Максимально планарный граф — это граф, который при добавлении любого ребра перестает быть планарным.

Планарный граф — это граф, который можно уложить на плоскости так, что никакие два его ребра не будут иметь пересечений. *Плоский граф* — это граф, уже уложенный на плоскости.

Плоской картой называется связный плоский граф вместе со всеми его гранями.

Граф называется *полным двудольным*, если любая его вершина, принадлежащая одному подмножеству, смежна с каждой вершиной другого подмножества и наоборот.

Понтрягина–Куратовского теорема. Граф планарен тогда и только тогда, *когда он не содержит подграфа*, *гомеоморфного графам* K_5 *или* $\mathsf{K}_{3,3}$.

Предельные подмножества — это независимые подмножества, содержащие наибольшее число элементов.

Раскраска вершин графа — это разбиение множества вершин на k непересекающихся классов (подмножеств) таких, что внутри каждого подмножества не должно содержаться смежных вершин.

Под *раскраской вершин гиперграфа* $\mathsf{H} = (X, E)$ понимается некоторое разбиение множества его вершин на классы, не содержащие пустого подмножества, когда инцидентные вершины должны быть окрашены в разные цвета, причем каждому классу назначается определенный цвет, в который окрашены вершины, а величина q соответствует количеству различных цветов, которые можно использовать для раскраски.

Граф G называется *стягиваемым* к графу H, если граф H можно получить из графа G с помощью некоторой последовательности элементарных стягиваний.

В гиперграфе $\mathsf{H} = (X, E)$ две вершины называются *смежными* в том случае, если существует ребро, содержащие эти вершины.

Ребра гиперграфа называются *смежными* в случае, если их пересечение представляет собой непустое подмножество.

Толщина графа — наименьшее число планарных подграфов, объединение которых дает исходный граф.

Фундаментальные циклы — это циклы, получаемые добавлением какого-нибудь ребра из графа к ребрам дерева и отличающиеся один от другого хотя бы одним ребром.

Харари–Татта теорема. Граф планарен тогда и только тогда, когда он не содержит подграфов, стягиваемых в K_5 *или* $\mathsf{K}_{3,3}$.

Хроматическое число графа $\chi(\mathsf{G})$ — наименьшее число подмножеств, на которое разбивается граф при раскраске.

В гиперграфе $\mathsf{H} = (X, E)$ *цепь* длины q определяется как последовательность вида $s(\mathsf{H}) = x_1, l_1, x_2, l_2, \ldots, l_q, x_{q+1}$.

Цикломатическое число графа — это наименьшее число ребер, которые необходимо удалить из графа, чтобы он стал ациклическим (деревом). Цикломатическое число определяется по формуле: $\gamma(\mathsf{G}) = m - n + k$, где n — число вершин, m — число ребер, k — число компонент связности графа.

Число внутренней устойчивости графа определяется мощностью предельного независимого подмножества.

Число внешней устойчивости графа определяется мощностью минимального внешне устойчивого подмножества, содержащего наименьшее число вершин.

Число скрещиваний (*пересечений*) *графа* — наименьшее число пересечений ребер, которое получается при расположении графа на плоскости.

Ядро графа — это некоторое подмножество вершин графа, являющееся одновременно внутренне и внешне устойчивым.

Подмножество ребер $C \subseteq U$ двудольного графа $\mathsf{G} = (A, B; U)$ называется *паросочетанием*, если никакие два ребра из C не имеют общей вершины. Паросочетание с наибольшим числом ребер называется *максимальным паросочетанием*.

Два графа $\mathsf{G}_1 = (X_1, U_1)$ и $\mathsf{G}_2 = (X_2, U_2)$ называются *изоморфными*, если существует взаимно-однозначное отображение φ: $X_1 \to X_2$ такое, что ребро $u = (x, y) \in U_1$ тогда и только тогда, когда $(\varphi(x), \varphi(y)) \in U_2$.

Заключение

> В любой науке столько истины, сколько в ней математики.
>
> *И. Кант*

В данном учебнике приводятся материалы интенсивно развивающейся области дискретной математики. Рассмотрены элементы теории множеств, алгоритмов и алгебры логики, теории графов и гиперграфов. Использование теории множеств позволяет абстрактно и формально описывать основные модели информатики и вычислительной техники, доказывать тождества с множествами и строить оптимальные схемы устройств цифровой техники. Теория множеств является первым, но основным звеном в общей теории дискретной математики.

Вместе с теорией множеств математическая логика и теория алгоритмов образуют теоретический фундамент современных вычислительных наук. Знание элементов теории алгоритмов и оценки временной сложности позволяют строить эффективные схемы алгоритмов, на основе которых создаются современные программные комплексы. Это дает возможность выполнять анализ и синтез устройств автоматики и вычислительной техники.

Решение задачи минимизации булевых функций позволяет решать задачи синтеза комбинационных устройств и логических схем, проблемы представления булевых функций в виде логических элементов.

Теория графов и гиперграфов — основа моделирования и разработки современной цифровой вычислительной техники и информационных систем.

В настоящее время применение дискретной математики является повсеместным во всех областях науки и техники. Поэтому она служит не только фундаментом современной математики, но и основным звеном современного математического образования.

Авторы стремились показать широкие возможности применения дискретной математики, теории алгоритмов и алгебры логики, их универсальность и специализированность для решения задач науки и техники. Авторы тщательно подобрали примеры и рисунки, которые сопровождаются подробными решениями с пояснениями.

Основная задача учебника вместо простого получения фактических знаний — обеспечить развитие навыков распознавания, формулирования и решения проблем. Использование учебника позволит организовать обучение в пространстве знаний, упорядоченных по направлению развития качества предметной области. Творческое построение процессов получения знаний позволит обеспечить практическую реализацию изученного материала.

Знание аппарата теории множеств, алгоритмов и алгебры логики позволяет студенту с помощью языка математики формировать новые понятия и повышает интеллектуальный потенциал студента. Практически все последующие курсы направлений «Информатика и вычислительная техника» и «Информационные системы» используют основные понятия и положения дискретной математики.

Библиографический комментарий

Книги — корабли мысли, странствующие по волнам времени и бережно несущие свой драгоценный груз от поколения к поколению.

Ф. Бэкон

Модуль 1

Задачам теории множеств посвящена обширная библиография, привести которую авторы не считают необходимым. Авторы при написании данного учебного пособия использовали концепции построения и описания множеств, приведенные в книге Шихановича [1]. Энциклопедической монографией по модулю 1 является классический труд группы известных французских математиков под псевдонимом Бурбаки [2]. Она может быть использована для углубления знаний, полученных при изучении данного пособия.

Различным аспектам теории множеств посвящены книги [3–10]. Всевозможные операции над множествами и большое количество примеров приведено в книгах [1, 3, 8–10]. Более подробный материал по упорядоченным множествам и кортежам можно найти в книгах [1, 2, 8–13]. Сведения об отношениях в том или ином объеме включены в множество книг по теоретико-множественной тематике. Различные способы представления отношений описаны в книгах [1, 2, 8–11].

Наиболее полно вопросы представления соответствий и примеры представления и преобразования соответствий даны в монографии Шихановича [1]. Материал по бесконечным множествам наиболее удачно для студентов изложен в книгах [1, 2, 12, 13, 23]. Понятие мультимножества является относительно новым и более полно описано в монографии Петровского [14].

Основателем теории нечетких множеств является Заде [15]. Для подробного изучения теории нечетких множеств рекомендуем следующую литературу [11, 16–20]. Различные разделы дискретной математики представлены на доступном языке в учебнике [21, 22].

Вопросы, посвященные приближенным множествам, подробно рассмотрены в книге [24].

Модуль 2

Задачам теории алгоритмов посвящена обширная библиография. Авторы при написании данного учебного пособия использовали концепции классического учебника Т. Кормена, Ч. Лейзерсона, Р. Ривеста «Алгоритмы: построение и анализ». Это энциклопедическая монография по дискретной математике. Кроме того, этот учебник может быть использован для углубления знаний, полученных при изучении данного пособия [25]. Авторы советуют также книги О.П. Кузнецова «Дискретная математика для инженеров» [7, 26], В.П. Сигорского «Математический аппарат инженера», а также методические разработки по теории алгоритмов авторов данного пособия [27, 42–44].

Различным аспектам теории алгоритмов посвящены книги [28–31, 35, 45]. Всевозможные описания типов универсальных алгоритмических моделей приведены в учебном пособии Л.А. Гладкова, В.В. Курейчика, В.М. Курейчика «Основы теории алгоритмов» [27]. Более подробный материал по нечетким моделям и алгоритмам можно найти в книгах [17–20, 32].

Наиболее полно примеры и задачи с решениями приведены в книге Шапорева [33] и учебном пособии авторов [27].

В книге [34] изложены методы и средства теории алгоритмов. Освещены основные математические свойства теории алгоритмов, необходимые для реальной практической деятельности.

В учебнике Дж. Макконелла [35] обсуждаются алгоритмы решения распространенных задач программирования: сортировки, сравнения с образцом, на графах, поиска и выборки и др.

Фундаментальный учебник Д.А. Андерсона [8] по дискретной математике подробно рассматривает вопросы логики, исчисления предикатов, алгоритмов и рекурсии. Особое внимание уделено теории доказательств. Книга содержит большое количество примеров и упражнений.

Сведения о комбинаторике в том или ином объеме включены в множество книг по дискретной математике.

Модуль 4

В книге [34] приведены методы и средства дискретной математики как инструментарий при обработке информации на ЭВМ. Книга удобна для студентов, так как содержит обширный материал по решению задач, особенно в разделе по теории графов.

В книге [35] широко обсуждаются практические алгоритмы на графах. Книга будет особенно интересна студентам, так как содержит описание реальных программ для решения графовых задач.

Книга [8] — это современный классический учебник по дискретной математике. Особое внимание уделено теории доказательств. Материал сопровождается многочисленными примерами и упражнениями, особенно по теории графов.

Книгу [30] можно рассматривать как справочник по графовым и сортировочным алгоритмам. Для студентов интерес может представлять наличие программ.

Пособие [47] можно рекомендовать для студентов с углубленной математической подготовкой. В нем в сжатой форме приведены основные разделы теории графов с задачами.

В книге [48] представлено систематизированное введение в теорию графов с доказательствами теорем и примерами.

Учебник [3] является единым методически взаимосвязанным курсом. Особый интерес для студентов может представлять раздел по графам и мографам, ориентированный на практическое применение в области информационных технологий.

Книга [49] интересна для студентов тем, что авторы показали связь общей теории сетей с прикладными задачами на графах различного вида.

В учебном пособии [50] с методической точки зрения излагается теория графов. Представлено сведение прикладных практических задач к задачам теории графов.

Учебное пособие [51] ориентировано на студентов технических вузов с хорошей математической подготовкой.

Книга [24] предназначена инженерам в области информационных технологий. Особое внимание уделено оптимизационным задачам на графах, имеющим практическое применение.

В монографии [52] изложены фундаментальные основы ИКТ с применением теории графов. Интерес для студентов может представлять наличие программных кодов для решения основных графовых задач.

В книге [53] с математической точки зрения рассматривается решение важнейшей теоретической и практической задачи: «сколько существует графов». Для студентов важен обзор решенных и нерешенных задач, перечисления графов.

Книга [54] дает полное представление о направлениях исследований в теории графов, приводятся упражнения и нерешенные задачи.

В книге [55] дано множество интересных приложений теории графов в различных областях науки и техники.

В учебном пособии [56] рассматриваются алгоритмические методы теории графов.

Основное внимание в учебнике [57] уделено алгебраическим методам анализа графов, ориентированным на практические инженерные задачи.

Книгу [58] можно использовать как справочное руководство по современной теории графов.

В книге [59] описаны основы теории графов и ее применение к сетям с сосредоточенными параметрами в электро- и вычислительной технике.

Также при изучении теории графов студентам могут быть полезны книги, приведенные в дополнительном списке литературы.

Список литературы

Книга — немой учитель.

Платон

Некоторые книги незаслуженно забываются, но нет ни одной, которую незаслуженно бы помнили.

У. Оден

1. *Шиханович Ю. А.* Введение в дискретную математику. — М.: Наука, 1965.
2. *Бурбаки Н.* Теория множеств / Под ред. В. А. Успенского. — М.: Мир, 1965.
3. *Горбатов В.А. и др.* Дискретная математика. Учебник для студентов втузов. — М., 2003.
4. Прикладная комбинаторная математика. Сб. статей под ред. Э. Беккенбаха. — М.: Мир, 1968.
5. *Хаггарти Р.* Дискретная математика для программистов: Учеб. пособ. — М.: Техносфера, 2003.
6. *Яблонский С. В.* Введение в дискретную математику: Учеб. пособ. — М.: Наука, 1979.
7. *Кузнецов О. П.*, *Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988.
8. *Андерсон Д.* Дискретная математика и комбинаторика. — М.: Издательский дом «Вильямс», 2003.
9. *Новиков Ф. А.* Дискретная математика для программистов. — М.: Издательский дом «Питер», 2000.
10. *Стол Р.* Множества. Логика. Аксиоматические теории. — М.: Просвещение, 1968.
11. *Мелихов А. Н.*, *Берштейн Л. С.* Конечные четкие и расплывчатые множества. Ч. 1, 2. — Таганрог: ТРТУ, 1981.
12. *Александров П. С.* Введение в теорию множеств и общую топологию. — М.: Наука, 1977.
13. *Куратовский К.*, *Мостовский А.* Теория множеств. — М.: Мир, 1970.
14. *Петровский А. Б.* Пространства множеств и мультимножеств. — М.: УРСС, 2003.
15. *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений. — М.: Мир, 1976.
16. *Борисов А. Н. и др.* Обработка нечеткой информации в системах принятия решений. — М.: Радио и связь, 1989.

17. *Ярушкина Н. Г.* Основы теории нечетких и гибридных систем. Учебное пособие. — М.: Финансы и статистика, 2004.
18. Нечеткие множества и теория возможностей / Под ред. В. Б. Кузьмина. — М.: Радио и связь, 1986.
19. *Рыжов А. П.* Элементы теории нечетких множеств и измерения нечеткости. — М.: Диалог-МГУ, 1998.
20. *Кофман А.* Введение в теорию нечетких множеств. — М.: Радио и связь, 1982.
21. *Спирина М. С.*, *Спирин М. А.* Дискретная математика. — М.: Издательский центр «Академия», 2004.
22. *Микони С. В.* Теория и практика рационального выбора. — М.: Маршрут, 2004.
23. *Хаусдорф Ф.* Теория множеств. — М.: КомКнига, 2006.
24. *Вагин В. Н.*, *Головина Е. Ю.*, *Загорянская А. А.*, *Фомина М. В.* Достоверный и правдоподобный вывод в интеллектуальных системах. — М.: Физматлит, 2004.
25. *Кормен Т.*, *Лейзерсон Ч.*, *Ривест Р.* Алгоритмы: построение и анализ. — М.: Издательский дом «Вильямс», 2000.
26. *Кузнецов О. П.* Дискретная математика для инженера. — СПб: Лань, 2004.
27. *Гладков Л. А.*, *Курейчик В. В.*, *Курейчик В. М.* Основы теории алгоритмов: Учеб. пособ. — Таганрог: Изд-во ТРТУ, 2003.
28. *Ерусалимский Я. М.* Дискретная математика: Теория, задачи, приложения. — М.: Вузовская книга, 2002.
29. *Судоплатов С. В.*, *Овчинникова Е. В.* Дискретная математика: Учебник. — М.: ИНФРА-М, 2005.
30. *Иванов Б. Н.* Дискретная математика. Алгоритмы и программы: Учеб. пособ. — М.: Лаборатория базовых знаний, 2001.
31. *Плотников А. Д.* Дискретная математика: Учеб. пособ. — М.: Новое знание, 2005.
32. *Берштейн Л. С.*, *Боженюк А. В.* Нечеткие модели принятия решений: дедукция, индукция, аналогия: Монография. — Таганрог: Изд-во ТРТУ, 2001.
33. *Шапорев С. Д.* Дискретная математика: Курс лекций и практических занятий. — СПб.: БХВ–Петербург, 2006.
34. *Капитонова Ю. В. и др.* Лекции по дискретной математике. — СПб.: БХВ–Петербург, 2004.
35. *Макконелл Дж.* Анализ алгоритмов: Краткий курс. — М.: Техносфера, 2002.
36. *Непейвода Н. Н.* Прикладная логика. — Новосибирск: Изд-во Новосибирского университета, 2000.
37. *Клини С. К.* Математическая логика. — М.: Мир, 1973.
38. *Ершов Ю. Л.*, *Палютин Е. А.* Математическая логика. — М.: Наука, 1979.

39. *Гиндикин С. Г.* Алгебра логики в задачах. — М.: Наука, 1972.
40. *Макоха А. Н.*, *Сахнюк П. А.*, *Червяков Н. И.* Дискретная математика. — М.: Физматлит, 2005.
41. *Акимов О. Е.* Дискретная математика: логика, группы, графы, фракталы. — М.: Издатель Акимова, 2005.
42. *Сигорский В. П.* Математический аппарат инженера. — Киев: Техника, 1997.
43. *Курейчик В. М.* Учеб. пособ. по дискретной математике. Часть 1. — Таганрог: Изд-во ТРТУ, 1997.
44. *Курейчик В. М.*, *Гладков Л. А.*, *Лисяк Н. К.*, *Курейчик В. В.* Учеб. пособ. для практических и лекционных занятий по курсу «Математические основы дискретной техники». Часть 1. — Таганрог: Изд-во ТРТУ, 1997.
45. *Пападимитриу Х.*, *Стайниц К.* Комбинаторная оптимизация. Алгоритмы и сложность. — М.: Мир, 1983.
46. *Довгий П. С.*, *Поляков В. И.* Синтез комбинационных схем. — СПб.: Изд-во СПбНИУ ИТМО, 2006.
47. *Редькин Н. П.* Дискретная математика: Курс лекций для студентов-механиков. — СПб.: Лань, 2003.
48. *Зыков А. А.* Основы теории графов. — М.: Вузовская книга, 2004.
49. *Френк Г.*, *Фриш И.* Сети, связи и потоки. — М.: Связь, 1978.
50. *Емеличев В. А. и др.* Лекции по теории графов: Учеб. пособ. — М.: Книжный дом «ЛИБРОКОМ», 2009.
51. *Белов В. В. и др.* Теория графов: Учеб. пособ. для втузов. — М.: Высшая школа, 1976.
52. *Касьянов В. Н.*, *Евстигнеев В. А.* Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ–Петербург, 2003.
53. *Харари Ф.*, *Палмер Э.* Перечисление графов. — М.: Мир, 1977.
54. *Оре О.* Теория графов. — М.: Книжный дом «ЛИБРОКОМ»/ УРСС, 2009.
55. *Басакер Р.*, *Саати Т.* Конечные графы и сети. — М.: Наука, 1973.
56. *Асеев Г. Г.*, *Абрамов О. М.*, *Ситников Д. Э.* Дискретная математика: Учеб. пособ. — Ростов-на-Дону: Феникс, 2003.
57. *Белоусов А. И.*, *Ткачев С. Б.* Дискретная математика: Учеб. пособ. для вузов. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2004.
58. *Татт У.* Теория графов. — М.: Мир, 1988.
59. *Свами М.*, *Тхуласираман К.* Графы, сети и алгоритмы. — М.: Мир, 1984.
60. *Абондо-Бодино Д.* Применение в экономике теории графов. — М.: Прогресс, 1966.

61. *Берштейн Л. С.*, *Боженюк А. В.* Введение в теорию нечетких графов. — Таганрог: Изд-во ТРТУ, 1995.
62. *Галкина В. А.* Дискретная математика: комбинаторные методы оптимизации. — М.: Гелиос АРВ, 2003.
63. *Гэри М.*, *Джонсон Д.* Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982.
64. *Кирсанов М. Н.* Графы в MAPLE. Задачи, алгоритмы, программы. — М.: Физматлит, 2007.
65. *Костюкова Н. И.* Графы и их применение. Комбинаторные алгоритмы для программистов: Учеб. пособ. — М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007.
66. *Кофман А.* Введение в прикладную комбинаторику. — М.: Наука, 1975.
67. *Кристофидес Н.* Теория графов. Алгоритмический подход. — М.: Мир, 1978.
68. *Курейчик В. М.* Математическое обеспечение КТП с применением САПР. — М.: Радио и связь, 1990.
69. *Курейчик В. М.* Дискретная математика: Учеб. пособ. Ч. 2. Элементы теории графов. — Таганрог: Изд-во ТРТУ, 1998.
70. *Курейчик В. М.*, *Гладков Л. А.*, *Лисяк Н. К.*, *Курейчик В. В.* Учеб. пособ. для практических и лекционных занятий по курсу «Математические основы дискретной техники». Ч. 2. — Таганрог: Изд-во ТРТУ, 1998.
71. *Мелихов А. Н.*, *Берштейн Л. С.*, *Курейчик В. М.* Применение графов для проектирования дискретных устройств. — М.: Наука, 1974.
72. *Мелихов А. Н.*, *Карелин В. П.* Методы распознавания изоморфизма и изоморфного вложения четких и нечетких графов. — Таганрог: Изд-во ТРТУ, 1995.
73. *Микони С. В.* Элементы дискретной математики. — СПб.: Изд-во ПГУПС, 1999.
74. *Палий И. А.* Дискретная математика: Курс лекций. — М.: Эксмо, 2008.
75. *Уилсон Р.* Введение в теорию графов. — М.: Мир, 1977.

ПРИЛОЖЕНИЕ

Основные положения ГОСТ 19.701-90 (ИСО 5807-85) «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения»

1. Введение

Схема программы отображает последовательность операций в программе и состоит:

1) из символов процесса, указывающих фактические операции обработки данных (включая символы, определяющие путь, которого следует придерживаться с учетом логических условий);

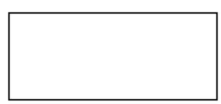
2) линейных символов, указывающих поток управления;

3) специальных символов, используемых для облегчения написания и чтения схемы.

2. Символы

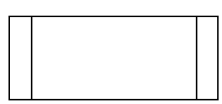
2.1. Процесс

Символ отображает функцию обработки данных любого вида.



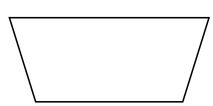
2.2. Предопределенный процесс

Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, в модуле).



2.3. Ручная операция

Символ отображает любой процесс, выполняемый человеком.

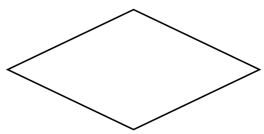


2.4. Подготовка

Символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы).

2.5. Решение

Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.

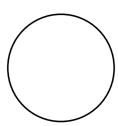


2.6. Линия

Символ отображает поток данных или управления.

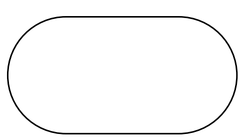
2.7. Соединитель

Символ используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.



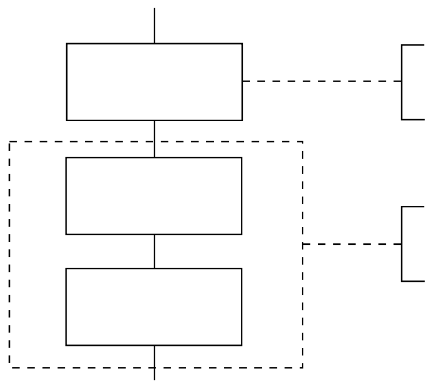
2.8. Терминатор

Символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец программы, внешнее использование и источник или пункт назначения данных).



2.9. Комментарий

Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры.



3. Правила применения символов

3.1. Символ предназначен для графической идентификации функции, которую он обозначает, независимо от текста внутри этого символа.

3.2. Символы в схеме должны быть расположены равномерно. Следует придерживаться разумной длины соединений и минимального числа длинных линий. Символы должны быть по возможности одного размера.

3.3. Текст для чтения должен записываться слева направо и сверху вниз независимо от направления потока. Если объем текста, помещаемого внутри символа, превышает его размеры, следует использовать символ комментария.

4. Правила выполнения соединений

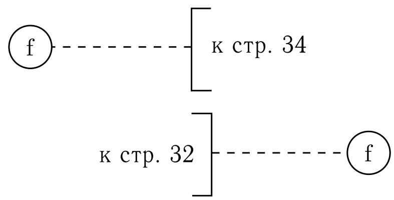
4.1. Направление потока слева направо и сверху вниз считается стандартным.

4.2. В схемах следует избегать пересечения линий. Изменение направления в точках пересечения не допускается.

4.3. Линии в схемах должны подходить к символу либо слева, либо сверху, а исходить либо справа, либо снизу. Линии должны быть направлены к центру символа.

4.4. При необходимости линии в схемах следует разрывать для избежания излишних пересечений или слишком длинных линий, а также, если схема состоит их нескольких страниц. Соединитель в начале разрыва называется внешним соединителем, а соединитель в конце разрыва — внутренним соединителем.

Ссылки к страницам могут быть приведены совместно с символом комментария для их соединителей.



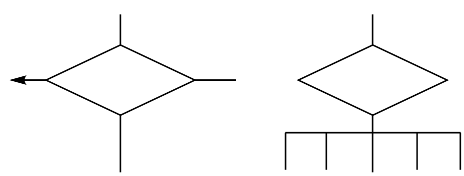
5. Специальные условные обозначения

Несколько выходов из символа следует показывать:

1) несколькими линиями от данного символа к другим символам;

2) одной линией от данного символа, которая затем разветвляется в соответствующее число линий.

Каждый выход из символа должен сопровождаться соответствующими значениями условий, чтобы показать логический путь, который он представляет, с тем чтобы эти условия и соответствующие ссылки были идентифицированы.



Об авторах

Курейчик Виктор Михайлович — доктор технических наук, профессор, заведующий кафедрой дискретной математики и методов оптимизации Южного федерального университета. Заслуженный деятель науки РФ. Академик Российской академии естественных наук, Академии инженерных наук РФ, Нью-Йоркской академии наук (1994–1995) США, старший член корпорации IEEE США (1992–н/вр).

Сфера научных интересов: физическое проектирование БИС и СБИС, искусственный интеллект в САПР, генетические алгоритмы и их применение в САПР.

Автор более 500 публикаций, включая 20 монографий и 40 изобретений. Под его руководством подготовлены 60 кандидатов и 10 докторов наук. Президент Российской ассоциации САПР; член Американского библиографического института.

В. М. Курейчик — руководитель секции отделения радиоэлектроники и систем управления Северо-Кавказского научного центра высшей школы, в рамках которой создал научную школу по генетическим методам оптимизации. Научный руководитель школы «Эволюционное моделирование, генетические алгоритмы и интеллектуальные САПР». Награжден серебряной и бронзовой медалями ВДНХ за разработку промышленных САПР. В 1999 г. награжден почетным знаком РАЕН «За заслуги в развитии науки и экономики», серебряной медалью Российской академии естественных наук за открытие в области генетических алгоритмов, в 2002 г. — медалью ордена «За заслуги перед Отечеством» II степени.

Курейчик Владимир Викторович — доктор технических наук, профессор, заведующий кафедрой систем автоматизированного проектирования Южного федерального университета.

Занимается разработкой интеллектуальных систем на основе САПР, информационных технологий, эволюционных вычислений, роевых алгоритмов для решения оптимизационных задач на графах.

Автор более 350 научных работ, в том числе 18 монографий, 14 учебников и учебных пособий, 16 изобретений. Под его руководством подготовлены 14 кандидатов и 1 доктор наук.

Является членом 2 диссертационных советов по защите докторских диссертаций в ЮФУ, Ученого Совета ЮФУ, экспертом 2 научных фондов (РФФИ и РНФ), членом редколлегии 5 ведущих научных журналов из перечня ВАК, председателем оргкомитета Международного научного конгресса «Интеллектуальные системы и информационные технологии», председателем Северо-Кавказского отделения Российской ассоциации искусственного интеллекта (РАИИ), членом-корреспондентом Российской академии естественных наук и действительным членом Академии инженерных наук им. А. М. Прохорова.

Гладков Леонид Анатольевич — кандидат технических наук, доцент кафедры САПР Южного федерального университета. Преподает дисциплины: «Методы оптимизации», «Дискретная математика», «Теоретические основы САПР», «Теория игр и комбинаторика», «Модели и методы анализа проектных решений».

Автор более 100 публикаций, в том числе более 10 монографий, 26 учебников и учебных пособий.

Научные интересы: теория графов, методы оптимизации, эволюционное моделирование, генетические алгоритмы, искусственный интеллект, гибридные интеллектуальные системы, системы поддержки принятия решений.

Является членом оргкомитета Международного научного конгресса «Интеллектуальные системы и информационные технологии», ученым секретарем Северо-Кавказского отделения Российской ассоциации искусственного интеллекта (РАИИ), членом совета Российской ассоциации нечетких вычислений и мягких множеств.

Учебное издание

ГЛАДКОВ Леонид Анатольевич
КУРЕЙЧИК Владимир Викторович
КУРЕЙЧИК Виктор Михайлович

ДИСКРЕТНАЯ МАТЕМАТИКА

Редактор *В.Р. Игнатова*
Оригинал-макет: *Е.В. Сабаева*
Оформление переплета: *В.Ф. Киселев*

Подписано в печать 01.09.2014. Формат 60×90/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 31. Уч.-изд. л. 34,1. Тираж 500 экз.
Заказ №

Издательская фирма «Физико-математическая литература»
МАИК «Наука/Интерпериодика»
117342, Москва, ул. Бутлерова, 17 Б
E-mail: fizmat@maik.ru, fmlsale@maik.ru;
http://www.fml.ru

Отпечатано с электронных носителей издательства
в ГУП Чувашской Республики
«ИПК «Чувашия» Мининформполитики Чувашии,
428019, г. Чебоксары, пр-т И. Яковлева, 13

ISBN 978-5-9221-1575-9

9 785922 115759