

Министерство образования и науки Российской Федерации
Московский государственный институт электронной техники
(технический университет)

Т.А. Олейник

**Основы дискретной математики:
теория и практика**

Учебное пособие

Утверждено редакционно-издательским советом института

Москва 2010

УДК 519.7(075.8)

О53

Рецензенты: канд. физ.-мат. наук, доц. *Е.Ю. Кулькова*;

канд. экон. наук, доц. *И.В. Платонова*

Олейник Т.А.

О53 **Основы дискретной математики: теория и практика:** уч. пособие. - М.:
МИЭТ, 2010. - 252 с.: ил.

ISBN 978-5-7256-0594-5

Рассматриваются с разной степенью полноты несколько разделов дискретной математики: комбинаторика, алгебра логики, теория графов и теория конечных автоматов. По каждому разделу изложены теоретические сведения и приведены многочисленные примеры решения стандартных задач. Содержится большой набор задач повышенной сложности, самостоятельное решение которых поможет студентам более глубоко усвоить материал.

Предназначено для студентов младших курсов.

ISBN 978-5-7256-0594-5

© МИЭТ, 2010

Учебное пособие

Олейник Татьяна Анатольевна

Основы дискретной математики: теория и практика

Редактор *Е.Г. Кузнецова*. Технический редактор *Л.Г. Лосякова*. Корректор *Л.Г. Лосякова*.
Верстка автора.

Подписано в печать с оригинал-макета 15.11.2010. Формат 60×84 1/16. Печать офсетная. Бумага офсетная. Гарнитура Times New Roman. Усл. печ. л. 14,63. Уч.-изд. л. 12.6. Тираж 250 экз. Заказ 141.

Отпечатано в типографии ИПК МИЭТ.

124498, Москва, Зеленоград, проезд 4806, д. 5, МИЭТ.

Оглавление

Предисловие

Глава 1. Множества, бинарные отношения, комбинаторика

- § 1.1. Множества и бинарные отношения.
 - Базовые понятия и утверждения
 - 1. Множества и операции над ними
 - 2. Бинарные отношения на множестве
 - Теоретические обоснования
 - Задачи повышенной сложности
- § 1.2. Элементы комбинаторики
 - Базовые понятия и утверждения
 - 1. Правило произведения и правило суммы
 - 2. Сочетания и размещения
 - 3. Некоторые комбинаторные соотношения
 - Теоретические обоснования
 - Задачи повышенной сложности

Глава 2. Функции алгебры логики

- § 2.1. Булевы функции и способы их задания
 - Базовые понятия и утверждения
 - 1. Булевы векторы и булевы функции от n переменных
 - 2. Элементарные булевы функции
 - 3. Задание булевых функций формулами
 - Теоретические обоснования
 - Задачи повышенной сложности
- § 2.2. Совершенные дизъюнктивные и конъюнктивные нормальные формы
 - Базовые понятия и утверждения
 - 1. Принцип двойственности.
 - 2. Задание функции совершенной дизъюнктивной нормальной формой
 - 3. Задание функции совершенной конъюнктивной нормальной формой
 - Теоретические обоснования
 - Задачи повышенной сложности
- § 2.3. Минимизация дизъюнктивных нормальных форм
 - Базовые понятия и утверждения
 - 1. Постановка задачи минимизации ДНФ
 - 2. Понятие о сокращенной и тупиковой ДНФ
 - 3. Построение минимальных ДНФ
 - Теоретические обоснования
 - Задачи повышенной сложности
- § 2.4. Классы Поста и замыкание
 - Базовые понятия и утверждения
 - 1. Функции, сохраняющие 0; функции, сохраняющие 1
 - 2. Самодвойственные функции
 - 3. Монотонные функции
 - 4. Линейные функции
 - 5. Замыкание системы булевых функций
 - Теоретические обоснования
 - Задачи повышенной сложности
- § 2.5. Полнота системы булевых функций
 - Базовые понятия и утверждения

1. Понятие о полноте системы булевых функций

2. Критерий полноты системы булевых функций

Теоретические обоснования

Задачи повышенной сложности

Глава 3. Теория графов

§ 3.1. Основные определения

Базовые понятия и утверждения

1. Общие понятия

2. Изоморфные графы

3. Виды графов

4. Матрица смежности и матрица инцидентности

5. Подграфы и операции над ними

Задачи повышенной сложности

§ 3.2. Достижимость и компоненты связности, циклы и мосты, цикломатическое число

Базовые понятия и утверждения

1. Пути, цепи, циклы на графе

2. Достижимость и компоненты связности графа

3. Мосты и циклы графа

4. Цикломатическое число графа

Теоретические обоснования

Задачи повышенной сложности

§ 3.3. Деревья

Базовые понятия и утверждения

1. Определение и основные свойства деревьев

2. Остовы графа

3. Построение минимального остова

4. Кодирование деревьев

Теоретические обоснования

Задачи повышенной сложности

§ 3.4. Планарность

Базовые понятия и утверждения

1. Укладка графа в трехмерном пространстве

2. Планарные графы

Теоретические обоснования

Задачи повышенной сложности

§ 3.5. Обходы графов

Базовые понятия и утверждения

1. Эйлеров цикл и эйлерова цепь

2. Гамильтонов цикл и гамильтонова цепь

Теоретические обоснования

Задачи повышенной сложности

§ 3.6. Раскраска графов

Базовые понятия и утверждения

Теоретические обоснования

Задачи повышенной сложности

§ 3.7. Фундаментальная система циклов графа

Базовые понятия и утверждения

Задачи повышенной сложности

§ 3.8. Ориентированные графы

Базовые понятия и утверждения

1. Общие понятия

2. Изоморфные орграфы
3. Матрица смежности и матрица инцидентности орграфа
4. Ориентированные пути, цепи, циклы на орграфе
5. Ориентированные деревья

Задачи повышенной сложности

§ 3.9. Отыскание кратчайших путей. Алгоритм Дейкстры

Базовые понятия и утверждения

Теоретические обоснования

§ 3.10. Задача о максимальном потоке в сети

Базовые понятия и утверждения

Теоретические обоснования

§ 3.11. Реализация булевых функций с помощью схем из функциональных элементов

§ 3.12. Реализация булевых функций с помощью упорядоченных бинарных диаграмм решений

Базовые понятия и утверждения

1. Основные понятия

2. Построение минимальных УБДР функции относительно заданного порядка переменных

3. Построение сокращенных УБДР по формулам

Задачи повышенной сложности

Глава 4. Элементы теории автоматов

§ 4.1. Ограниченно-детерминированные функции

Базовые понятия и утверждения

1. Детерминированные функции

2. Ограниченно-детерминированные функции

§ 4.2. Реализация ограниченно-детерминированных функций конечными автоматами

Базовые понятия и утверждения

1. Конечный автомат Мили, способы его задания

2. Продолжение функций переходов и выходов на слова

3. Приведенный автомат

Задачи повышенной сложности

Ответы и указания к упражнениям

Литература

Предисловие

Данная книга представляет собой начальное пособие по дискретной математике и содержит материал, предусмотренный стандартной программой технических высших учебных заведений. Предварительные знания, необходимые для изучения этого пособия, исчерпываются школьным курсом математики и начальными сведениями по теории матриц.

Пособие состоит из четырех глав. В первой главе приведены сведения из элементарной теории множеств и бинарных отношений, начальные сведения из комбинаторики. Вторая глава посвящена теории булевых функций и знакомит с различными способами их задания, со специальными классами формул для представления булевых функций, такими как совершенные дизъюнктивные, конъюнктивные нормальные формы и полином Жегалкина. Достаточно подробно обсуждается понятие полноты системы булевых функций. В третьей главе излагаются основы теории графов, в том числе рассматриваются несколько оптимизационных задач. Последние два параграфа третьей главы знакомят с двумя представлениями булевых функций: схемами из функциональных элементов и упорядоченными бинарными диаграммами решений. Четвертая глава может рассматриваться как введение в теорию автоматов.

Материал первой главы используется при изложении последующих глав, вторая и третья главы практически независимы и могут изучаться в любом порядке относительно друг друга. Знакомство с четвертой главой предполагает знакомство со всеми предшествующими главами.

Построение книги имеет ряд особенностей. Главы делятся на параграфы, ориентированные на изучение отдельной темы. Параграфы разбиты на части: «Базовые понятия и утверждения», «Теоретические обоснования» и «Задачи повышенной сложности».

В первой части каждого параграфа приводятся основные понятия, излагаются без обоснования важные утверждения. Понятия и утверждения иллюстрируются многочисленными примерами. Приводятся решения типовых задач, а также даются простые упражнения для самостоятельного решения. В конце пособия приводятся ответы к упражнениям, ко многим из них даны указания и решения.

Во второй части каждого параграфа тема излагается на более глубоком уровне, в частности, разбираются доказательства значительной части утверждений и нетривиальных алгоритмов из первой части.

Третья часть каждого параграфа содержит набор нетиповых задач по теме, предназначенных для студентов, проявляющих повышенный интерес к предмету.

Если студент не видит необходимости в глубокой проработке материала, то он может ограничиться изучением лишь первых частей параграфов. Это позволит ему познакомиться с достаточно широким кругом понятий дискретной математики, набором алгоритмов решения стандартных задач и поможет в дальнейшем самостоятельно изучать прикладную литературу.

Студенту, заинтересованному в усвоении курса, советуем изучить пособие в полном объеме: познакомиться с обоснованиями утверждений и алгоритмов, а также не пожалеть времени на поиск решений задач повышенной сложности.

Глава 1. Множества, бинарные отношения, комбинаторика

§ 1.1. Множества и бинарные отношения

Множество, способы задания множеств. Мощность конечного множества. Подмножество. Операции над множествами: дополнение, объединение, пересечение, разность, декартово произведение. Правило суммы, формула включений и исключений. Бинарное отношение на множестве. Свойства бинарных отношений: рефлексивность, симметричность, транзитивность. Отношение эквивалентности и отношение порядка.

Базовые понятия и утверждения

1. Множества и операции над ними. Под *множеством* понимают объединение в единое целое определенных вполне различаемых объектов. Объекты при этом называют *элементами* образуемого ими множества.

Для обозначения множеств используют прописные буквы, а для обозначения элементов множеств - строчные буквы латинского алфавита.

Запись $x \in M$ означает, что x является элементом множества M ; в противном случае пишут $x \notin M$.

Множество называют *конечным*, если оно содержит конечное число элементов, и *бесконечным*, если оно содержит бесконечное число элементов. Множество, не содержащее элементов, называют *пустым* и обозначают символом \emptyset .

Число элементов конечного множества M называют его *мощностью* и обозначают $|M|$.

Множество можно описать, указав свойство, присущее элементам только этого множества. Множество всех объектов, обладающих свойством $P(x)$, обозначают $\{x | P(x)\}$. Конечное множество можно задать путем перечисления его элементов, т.е. $M = \{x_1, x_2, \dots, x_n\}$.

Например, запись $M = \{x | x^2 - 16 = 0, x \in \mathbf{R}\}$ означает, что множество M содержит два элемента - числа -4 и 4 .

Если каждый элемент множества A есть элемент множества B , то говорят, что A есть *подмножество* B , и пишут: $A \subseteq B$.

Заметим, что пустое множество \emptyset считают подмножеством любого множества.

Если $A \subseteq B$ и $B \subseteq A$, то говорят, что множества A и B *равны*, и пишут: $A = B$.

Если $A \subseteq B$ и $A \neq B$, то A называют *собственным подмножеством* B и, чтобы подчеркнуть это, применяют запись $A \subset B$.

Множество всех подмножеств множества M называют его *булеаном* и обозначают 2^M .

Например, если $M = \{a, b, c\}$, то

$$2^M = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}.$$

Вводят целый ряд **операций над множествами**, позволяющих получать из одних множеств другие.

1. Множество, состоящее из тех и только тех элементов, которые принадлежат хотя бы одному из множеств A и B , называют *объединением* A и B и обозначают $A \cup B$, т.е.

$$A \cup B = \{x | x \in A \text{ или } x \in B\}.$$

2. Множество, состоящее из тех и только тех элементов, которые принадлежат как множеству A , так и множеству B , называют *пересечением* A и B и обозначают $A \cap B$, т.е. $A \cap B = \{x | x \in A \text{ и } x \in B\}$.

Если $A \cap B = \emptyset$, то множества A и B называют *непересекающимися*.

3. Множество, состоящее из всех элементов множества A , не принадлежащих множеству B , называют *разностью* A и B и обозначают $A \setminus B$, т.е.

$$A \setminus B = \{x | x \in A \text{ и } x \notin B\}.$$

4. Обычно в конкретных рассуждениях всякое множество рассматривают как подмножество некоторого достаточно широкого множества, которое называют *универсальным*. Множество элементов универсального множества I , не принадлежащих множеству A , называют *дополнением* A и обозначают \bar{A} , т.е.

$$\bar{A} = \{x | x \in I \text{ и } x \notin A\}. \text{ Из определения следует, что } \bar{\bar{A}} = I \setminus A.$$

5. Множество, состоящее из упорядоченных пар (x, y) , в которых x - элемент множества A , а y - элемент множества B , называют *декартовым произведением множеств A и B* и обозначают $A \times B$, т.е. $A \times B = \{(x, y) | x \in A \text{ и } y \in B\}$.

Удобным приемом наглядного изображения операций являются диаграммы Эйлера - Венна. На них множества представлены плоскими фигурами (чаще всего кругами). Области, соответствующие множествам, полученным в результате операции, обычно выделяют цветом. На рис. 1.1 приведены диаграммы Эйлера - Венна, иллюстрирующие некоторые из введенных операций.

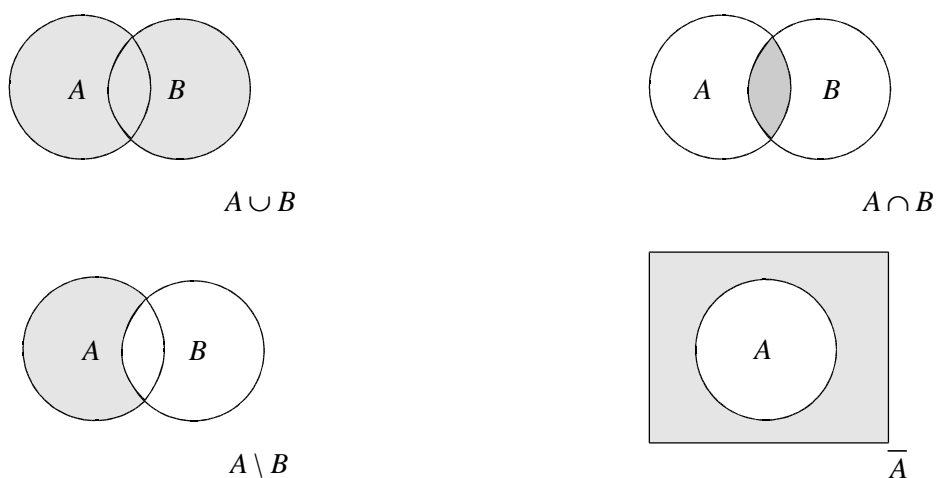


Рис. 1.1.

В качестве примера найдем объединение, пересечение, разность и декартово произведение множеств $A = \{x | x^3 - 9x = 0, x \in \mathbf{R}\}$ и $B = \{x | x^2 < 16, x \in \mathbf{N}\}$.

Поскольку $A = \{-3, 0, 3\}$, $B = \{1, 2, 3\}$, то $A \cup B = \{-3, 0, 1, 2, 3\}$, $A \cap B = \{3\}$, $A \setminus B = \{-3, 0\}$, $A \times B = \{(-3, 1), (-3, 2), (-3, 3), (0, 1), (0, 2), (0, 3), (3, 1), (3, 2), (3, 3)\}$.

Пусть задано универсальное множество I . Тогда для любых множеств $A, B, C \subset I$ выполняются следующие **свойства**:

коммутативные законы:

1. $A \cup B = B \cup A$;
2. $A \cap B = B \cap A$;

ассоциативные законы:

3. $A \cup (B \cap C) = (A \cup B) \cap C$;
4. $A \cap (B \cup C) = (A \cap B) \cup C$;

дистрибутивные законы:

$$5. A \cup (B \cap C) = (A \cup B) \cap (A \cup C);$$

$$6. A \cap (B \cup C) = (A \cap B) \cup (A \cap C);$$

законы идемпотентности:

$$7. A \cup A = A;$$

$$8. A \cap A = A;$$

законы де Моргана:

$$9. \overline{A \cup B} = \overline{A} \cap \overline{B};$$

$$10. \overline{A \cap B} = \overline{A} \cup \overline{B};$$

законы нуля:

$$11. A \cup \emptyset = A;$$

$$12. A \cap \emptyset = \emptyset;$$

законы единицы:

$$13. A \cup I = I;$$

$$14. A \cap I = A;$$

законы поглощения:

$$15. A \cup (A \cap B) = A;$$

$$16. A \cap (A \cup B) = A;$$

законы дополнения:

$$17. A \cup \overline{A} = I;$$

$$18. A \cap \overline{A} = \emptyset;$$

закон двойного дополнения:

$$19. \overline{\overline{A}} = A.$$

О том, как доказываются эти равенства, можно узнать во второй части данного параграфа.

Операции объединения, пересечения и декартова произведения можно обобщить на случай произвольного конечного числа участников.

Объединением множеств A_1, A_2, \dots, A_n называют множество, любой элемент которого является элементом хотя бы одного из данных множеств. Обозначение:

$$A_1 \cup A_2 \cup \dots \cup A_n \text{ или } \bigcup_{i=1}^n A_i.$$

Пересечением множеств A_1, A_2, \dots, A_n называют множество, любой элемент которого является элементом каждого из данных множеств. Обозначение: $A_1 \cap A_2 \cap \dots \cap A_n$ или $\bigcap_{i=1}^n A_i$.

Декартовым произведением множеств A_1, A_2, \dots, A_n называют множество

$$A_1 \times A_2 \times \dots \times A_n = \{(x_1, x_2, \dots, x_n) \mid x_1 \in A_1, x_2 \in A_2, \dots, x_n \in A_n\}.$$

В частном случае одинаковых сомножителей декартово произведение ~~$A \times A \times \dots \times A$~~
 n сомножителей

обозначают A^n .

Например, если $B = \{0, 1\}$, то

$$B^2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\},$$

$$B^3 = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}.$$

Приведем без доказательств **утверждения о числе элементов конечных множеств**.

1. Если между конечными множествами A и B существует взаимно-однозначное соответствие, то $|A| = |B|$.

2. Если A_1, A_2, \dots, A_n - конечные множества, то множество $A_1 \times A_2 \times \dots \times A_n$ также конечно и

$$|A_1 \times A_2 \times \dots \times A_n| = |A_1| \cdot |A_2| \cdot \dots \cdot |A_n|.$$

Например, если $B = \{0, 1\}$, то множество B^{10} имеет мощность $|B^{10}| = |B|^{10} = 1024$.

3. Если A_1, A_2, \dots, A_n - конечные попарно-непересекающиеся множества, то множество $A_1 \cup A_2 \cup \dots \cup A_n$ также конечно и

$$|A_1 \cup A_2 \cup \dots \cup A_n| = |A_1| + |A_2| + \dots + |A_n|.$$

Это утверждение называют **правилом суммы**.

4. Если A_1, A_2, \dots, A_n - конечные множества, то множество $\bigcup_{i=1}^n A_i$ также конечно и

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| +$$

$$+ \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n-1} |A_1 \cap \dots \cap A_n|.$$

Последнее равенство называется **формулой включений и исключений**. В частных случаях двух и трех множеств она принимает вид:

$$\begin{aligned} |A_1 \cup A_2| &= |A_1| + |A_2| - |A_1 \cap A_2|; \\ |A_1 \cup A_2 \cup A_3| &= |A_1| + |A_2| + |A_3| - \\ &\quad - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3| + |A_1 \cap A_2 \cap A_3|. \end{aligned}$$

Заметим, что формула включений и исключений действует и в том случае, когда множества A_1, A_2, \dots, A_n попарно не пересекаются (в этом случае все слагаемые в правой части формулы, содержащие пересечения множеств, обнуляются и формула трансформируется в правило суммы).

Пусть, например, $|A| = 4$, $|B| = 2$, $|C| = 3$, причем $A \cap B = \emptyset$, а $|A \cap C| = 2$. Тогда $|A \cup B|$ можно найти по правилу суммы: $|A \cup B| = |A| + |B| = 4 + 2 = 6$, а для поиска $|A \cup C|$ нужно использовать формулу включений и исключений: $|A \cup C| = |A| + |C| - |A \cap C| = 4 + 3 - 2 = 5$.

Пример 1. В группе из 100 туристов 65 человек знают английский язык, 55 человек знают французский и 38 человек знают оба языка. Сколько туристов в группе знает хотя бы один из этих языков?

◀ Пусть A и B - множества туристов, знающих соответственно английский и французский язык. Тогда $A \cup B$ - множество туристов, знающих хотя бы один из этих языков. Число таких туристов находим по формуле включений и исключений $|A \cup B| = |A| + |B| - |A \cap B| = 65 + 55 - 38 = 82$. ▶

Упражнение 1.1. Из 100 студентов-лингвистов польский язык изучают 42, чешский - 25, венгерский - 36, польский и чешский - 15, польский и венгерский - 14, чешский и венгерский - 12, польский, чешский и венгерский - 5. Сколько студентов не изучают ни одного из перечисленных языков?

Совокупность непустых, попарно непересекающихся подмножеств A_1, A_2, \dots, A_r

множества A называют *разбиением* A , если $A = \bigcup_{k=1}^r A_k$.

Например, для множества $M = \{a, b, c\}$ совокупность подмножеств $\{a\}, \{b, c\}$ разбиением является, а совокупность подмножеств $\{a, b\}, \{b, c\}$ не является.

Упражнение 1.2. Найти все разбиения множества $B = \{0,1\}$ и множества B^2 .

2. Бинарные отношения на множестве. Бинарные отношения -простой и вместе с тем очень важный объект дискретной математики.

Определение. *Бинарным отношением* на множестве M называется подмножество декартова произведения $M \times M$.

Для обозначения бинарных отношений, как правило, будем использовать строчные буквы греческого алфавита: $\alpha, \beta, \gamma, \tau$ и т.п.

Пусть τ - некоторое бинарное отношение на множестве M . Если $(a, b) \in \tau$, то говорят, что a и b связаны бинарным отношением τ и пишут $a\tau b$.

Пример 2. Пусть $M = \{-1, 0, 1\}$. Тогда

$$M \times M = \left\{ \begin{array}{lll} (-1, -1), & (-1, 0), & (-1, 1), \\ (0, -1), & (0, 0), & (0, 1), \\ (1, -1), & (1, 0), & (1, 1) \end{array} \right\}$$

и следующие множества могут служить примерами бинарных отношений на множестве M :

$$\alpha = \{(a, b) \mid a \geq b\} = \{(-1, -1), (0, -1), (0, 0), (1, -1), (1, 0), (1, 1)\};$$

$$\beta = \{(a, b) \mid a > b\} = \{(0, -1), (1, -1), (1, 0)\};$$

$$\gamma = \{(a, b) \mid a = b\} = \{(-1, -1), (0, 0), (1, 1)\};$$

$$\varphi = \{(a, b) \mid a^2 = b^2\} = \{(-1, -1), (-1, 1), (0, 0), (1, 1), (1, -1)\}.$$

Перечислим ряд важных **свойств**, которыми могут обладать бинарные отношения.

Определенное на множестве M бинарное отношение τ :

рефлексивно, если для $\forall a \in M$ выполняется $a\tau a$;

симметрично, если для $\forall a, \forall b \in M$ из $a\tau b$ следует $b\tau a$;

антисимметрично, если для $\forall a, \forall b \in M$ из $a\tau b$ и $b\tau a$ следует $a = b$;

транзитивно, если для $\forall a, \forall b, \forall c \in M$ из $a\tau b$ и $b\tau c$ следует $a\tau c$.

Определение. Если бинарное отношение рефлексивно, симметрично и транзитивно одновременно, то оно называется *отношением эквивалентности*.

Например, бинарное отношение α из примера 2 рефлексивно, антисимметрично и транзитивно, β - антисимметрично и транзитивно, γ - рефлексивно, симметрично, антисимметрично и транзитивно, ϕ - рефлексивно, симметрично и транзитивно. Следовательно, бинарные отношения γ и ϕ являются отношениями эквивалентности, а α и β - нет.

Определение. Пусть τ - отношение эквивалентности на множестве M и a - элемент M . *Классом эквивалентности* элемента a по бинарному отношению τ называют множество $[a]_{\tau} = \{b \in M \mid a\tau b\}$.

Например, множества $[-1]_{\gamma} = \{-1\}$, $[0]_{\gamma} = \{0\}$, $[1]_{\gamma} = \{1\}$ - классы эквивалентности элементов M по отношению γ , а $[-1]_{\tau} = \{-1, 1\}$, $[0]_{\tau} = \{0\}$, $[1]_{\tau} = \{-1, 1\}$ - классы эквивалентности элементов M по ϕ .

Упражнение 1.3. На множестве $M = \{1, 2, 3, 4\}$ определены бинарные отношения $\chi = \{(a, b) \mid a + b - \text{четно}\}$ и $\varepsilon = \{(a, b) \mid a \cdot b - \text{четно}\}$. Задать эти бинарные отношения перечислением элементов, указать свойства этих бинарных отношений, определить, являются ли они отношениями эквивалентности (если являются, то найти классы эквивалентности их элементов).

Перечислим **свойства классов эквивалентности**, присущие любому отношению эквивалентности, определенному на произвольном множестве M .

1. *Класс эквивалентности любого элемента множества M - непустое множество.*
2. *Классы эквивалентности любых двух элементов множества M либо не пересекаются, либо совпадают.*
3. *Объединение классов эквивалентности всех элементов множества M совпадает с самим множеством M .*

Доказательство этих свойств приведено во второй части параграфа.

Из свойств классов эквивалентности следует утверждение: *всякое отношение эквивалентности, заданное на множестве M , порождает разбиение множества M на классы эквивалентности этого отношения.*

Для иллюстрации этого утверждения вновь обратимся к бинарным отношениям γ и ϕ из примера 2.

Очевидно, что классы эквивалентности $[-1]_\gamma = \{-1\}$, $[0]_\gamma = \{0\}$, $[1]_\gamma = \{1\}$ элементов множества $M = \{-1, 0, 1\}$ по отношению γ не пусты, попарно не пересекаются, а их объединение совпадает с самим множеством M . Следовательно, γ порождает разбиение множества M на три подмножества: $\{-1\}$, $\{0\}$, $\{1\}$.

Для классов эквивалентности $[-1]_\phi = \{-1, 1\}$, $[0]_\phi = \{0\}$, $[1]_\phi = \{-1, 1\}$ элементов M по отношению ϕ имеем: классы эквивалентности элементов -1 и 1 совпадают и при этом не имеют общих элементов с классом эквивалентности элемента 0 , объединение всех классов совпадает с множеством M . Следовательно, отношение ϕ порождает разбиение множества M на два подмножества: $\{-1, 1\}$, $\{0\}$.

Рассмотрим еще один важный класс бинарных отношений.

Определение. Бинарное отношение называется *отношением порядка*, если оно рефлексивно, антисимметрично и транзитивно.

Пусть τ - отношение порядка на M . Если для любых двух элементов a и b множества M верно, что либо $a\tau b$, либо $b\tau a$, то τ называют отношением *линейного порядка*. В противном случае говорят, что τ - отношение *частичного порядка*.

Например, отношениями порядка являются отношения α и γ из примера 2 (α - линейного, γ - частичного).

Пример 3. Рассмотрим на множестве $B = \{2, 3, 6\}$ бинарное отношение ϕ , определяемое условием $a\phi b \Leftrightarrow (a \text{ делится нацело на } b)$. Это отношение рефлексивно, антисимметрично и транзитивно, и, значит, является отношением порядка, причем частичного, поскольку элемент 2 не связан с элементом 3 и элемент 3 не связан с элементом 2 .

Теоретические обоснования

Вернемся к свойствам (1) – (19) операций над множествами. Все они представляют собой утверждения о равенстве двух множеств. Стандартный способ доказательства равенства двух множеств $M_1 = M_2$ состоит в доказательстве двух включений: $M_1 \subseteq M_2$ и $M_2 \subseteq M_1$.

Доказательство каждого такого включения (пусть для определенности это будет $M_1 \subseteq M_2$) проводится по следующей схеме: рассматривается произвольный элемент множества M_1 и устанавливается, что он также является элементом множества M_2 .

В качестве примера докажем один из дистрибутивных законов:

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

1) Пусть x - произвольный элемент из $A \cap (B \cup C)$. Тогда по определению операции \cap имеем $x \in A$ и $x \in (B \cup C)$. Во втором случае из определения операции \cup выводим, что $x \in B$ или $x \in C$. Если $x \in B$, то с учетом того, что $x \in A$, получаем $x \in A \cap B$. Если $x \in C$, то с учетом того, что $x \in A$, получаем $x \in A \cap C$. Таким образом, $x \in A \cap B$ или $x \in A \cap C$. Следовательно, по определению операции \cup имеем $x \in (A \cap B) \cup (A \cap C)$. Тем самым установлено, что $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.

2) Пусть x - произвольный элемент из $(A \cap B) \cup (A \cap C)$. Тогда по определению операции \cup имеем $x \in A \cap B$ или $x \in A \cap C$. В первом случае из определения операции \cap выводим, что $x \in A$ и $x \in B$. Во втором случае - $x \in A$ и $x \in C$. Таким образом, $x \in B$ или $x \in C$, значит, $x \in B \cup C$. Кроме того, в обоих случаях $x \in A$. Следовательно, согласно определению операции \cap , имеем $x \in A \cap (B \cup C)$. Тем самым установлено, что $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$.

Действуя по такой же схеме, можно доказать и другие свойства операций над множествами (советуем проделать это самостоятельно).

В первой части данного параграфа были сформулированы свойства классов эквивалентности. Докажем эти свойства, предварительно записав их с использованием математической символики.

Теорема 1.1 (о свойствах классов эквивалентности). Пусть τ - отношение эквивалентности на множестве M . Тогда

1. $\forall a \in M \quad [a]_\tau \neq \emptyset$;
2. $\forall a, \forall b \in M \quad ([a]_\tau \cap [b]_\tau \neq \emptyset \Rightarrow ([a]_\tau = [b]_\tau))$;
3. $\bigcup_{a \in M} [a]_\tau = M$.

Доказательство. 1. τ - отношение эквивалентности, следовательно, τ является рефлексивным, т.е. $\forall a \in M$ выполняется $a\tau a$. Но тогда $a \in [a]_\tau$ и, значит, $[a]_\tau \neq \emptyset$.

2. Пусть $[a]_\tau \cap [b]_\tau \neq \emptyset$, т.е. $\exists c \in ([a]_\tau \cap [b]_\tau)$. Тогда $c \in [a]_\tau$ и $c \in [b]_\tau$, откуда $a\tau c$ и $b\tau c$, и, следовательно, в силу симметричности τ $a\tau c$ и $c\tau b$, и, наконец, поскольку τ транзитивно, получим $a\tau b$.

Возьмем любой элемент d множества $[b]_\tau$, тогда $b\tau d$. Так как $a\tau b$ и $b\tau d$, то в силу транзитивности τ $a\tau d$, т.е. $d \in [a]_\tau$. Таким образом, $[b]_\tau \subseteq [a]_\tau$.

Аналогично получим $[a]_\tau \subseteq [b]_\tau$. Следовательно, $[a]_\tau = [b]_\tau$.

3. Докажите это утверждение самостоятельно. ■

Задачи повышенной сложности

1.1. При голосовании в городскую думу в бюллетене в списке из трех кандидатов требовалось оставить не более одного. При подведении итогов оказалось, что кандидатов A и B вычеркнули 60% избирателей, кандидатов B и C - 80% избирателей, а кандидатов A и C - 70% избирателей. Какой процент избирателей проголосовал против всех кандидатов и какой кандидат набрал наибольшее число голосов?

1.2. Сколько бинарных отношений можно определить на множестве из n элементов?

1.3. На множестве $A = \{1, 2, 3, 4\}$ задано бинарное отношение $\alpha = \{(m, n) \mid m = n \text{ или } |m - n| = 2\}$. Какими свойствами обладает это бинарное отношение? Является ли оно отношением эквивалентности? порядка? В том случае, если α - отношение эквивалентности, указать разбиение множества A на классы эквивалентности.

1.4. Рассмотрим на множестве действительных чисел \mathbb{R} бинарные отношения $\beta, \gamma, \sigma, \rho$, определенные условиями: $\beta = \{(a, b) \mid a^2 = b^2\}$, $\gamma = \{(a, b) \mid a^2 < b^2\}$, $\sigma = \{(a, b) \mid a^2 \leq b^2\}$, $\rho = \{(a, b) \mid a = b\}$. Выяснить, какими свойствами обладают перечисленные бинарные отношения.

1.5. Привести пример рефлексивного, транзитивного, но не симметричного бинарного отношения на множестве из четырех элементов.

1.6. Привести пример рефлексивного, симметричного, но не транзитивного бинарного отношения на множестве из четырех элементов.

1.7. Привести пример транзитивного, симметричного, но не рефлексивного бинарного отношения на множестве из четырех элементов.

1.8. Пусть $M = \{x_1, x_2, \dots, x_n\}$ - конечное множество. Сопоставим каждому бинарному отношению τ на M матрицу размера $n \times n$, называемую *матрицей отношения*, в которой на пересечении i -й строки и j -го столбца стоит 1, если $(x_i, x_j) \in \tau$, и 0 - в противном случае. Определить, что представляет собой матрица отношений τ в случае, если τ :

а) рефлексивно;

б) симметрично;

в) антисимметрично.

1.9. Пусть A - конечное множество. Определим на булеане 2^A бинарное отношение χ :

$B\chi C \Leftrightarrow (B \subseteq C)$ (здесь $B \subseteq 2^A$, $C \subseteq 2^A$). Является ли отношение χ отношением эквивалентности? отношением частичного (линейного) порядка?

1.10. Доказать, что если бинарное отношение одновременно симметрично и антисимметрично, то оно транзитивно.

§ 1.2. Элементы комбинаторики

Выборки. Использование правила произведения и правила суммы для подсчета числа выборов. Сочетания и размещения без повторений и с повторениями. Перестановки. Формулы подсчета числа сочетаний, размещений, перестановок. Приемы доказательства комбинаторных тождеств. Метод математической индукции. Бином Ньютона. Биномиальные коэффициенты. Треугольник Паскаля.

Базовые понятия и утверждения

1. Правило произведения и правило суммы. В комбинаторике изучаются способы подсчета числа элементов конечных множеств. Рассмотрим схемы рассуждений, которые при этом используются.

Пример 1. Сборная команда легкоатлетов университета, в состав которой входят преподаватели Мишин и Петров, аспиранты Виктор, Иван, Борис и Андрей, а также студенты Тимур, Олег и Лев, принимает участие в межвузовской эстафете. По условиям соревнований на первом этапе соревнуются преподаватели, на втором - аспиранты, на третьем - студенты. Сколькими способами можно сформировать команду для участия в эстафете?

◀ Сформируем команду за три шага. Вначале выберем преподавателя для участия в первом этапе эстафеты, затем аспиранта для участия во втором этапе, и, наконец, студента для участия в третьем этапе. Представим все варианты формирования команды с помощью схемы (рис. 1.2). В первой строке схемы укажем варианты выбора преподавателя (М - Мишин, П - Петров), во второй строке - варианты выбора аспиранта (В - Виктор, И - Иван, Б - Борис, А - Андрей), в третьей - варианты выбора студента (Т - Тимур, О - Олег, Л - Лев).

Заметим, что число вариантов выбора на втором шаге не зависит от того выбора, который сделан на первом шаге (на схеме эта независимость выражается в том, что из букв М и П первой строки выходит одинаковое число стрелок во вторую строку), а число вариантов выбора на третьем шаге не зависит от того выбора, который сделан на первых двух шагах.

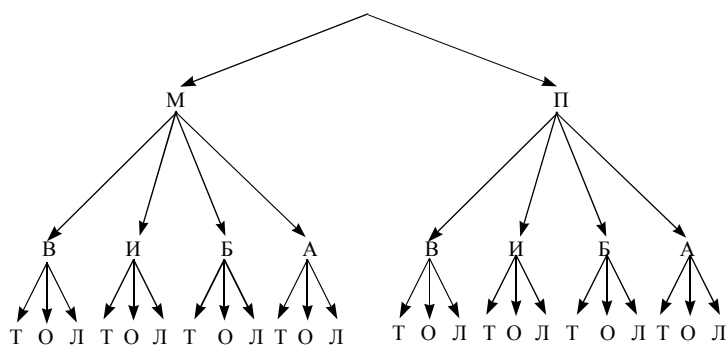


Рис. 1.2.

Каждому варианту состава команды на схеме соответствует путь, идущий из верхней строки в нижнюю (например, $M \rightarrow I \rightarrow L$). Поэтому число всех возможных вариантов формирования команды для участия в эстафете равно числу таких путей. Так как в первой строке две буквы, во второй - вчетверо больше, чем в первой, а третьей - втрое больше, чем во второй, то общее число путей равно произведению $2 \cdot 4 \cdot 3 = 24$. Таким образом, имеем 24 варианта формирования команды. ►

Вопрос: сколькими способами можно сформировать такой-то объект, является типичным для комбинаторных задач. При этом в большинстве случаев его можно заменить вопросом: сколько элементов содержится в таком-то множестве? Поэтому решение практически любой комбинаторной задачи начинается с описания множества, число элементов которого требуется подсчитать.

Во многих случаях множества удобно описывать как множества выборок. Под *выборками* понимают наборы элементов некоторого исходного множества M . Выборка - понятие широкое. В выборках элементы в соответствии с условиями задачи могут как повторяться, так и не повторяться. Порядок следования элементов в выборке может учитываться (в этом случае выборку называют упорядоченной), а может не учитываться. Вообще в соответствии с условиями задачи рассматриваются выборки, удовлетворяющие самым разным ограничениям. Например, может оказаться, что некоторые элементы множества M должны обязательно входить в выборку, а другие элементы могут как входить, так и не входить. Или может оказаться, что нужно рассматривать только такие упорядоченные выборки, в которые некоторые элементы множества M не только обязательно входят, но и идут друг за другом. При составлении математической модели комбинаторной задачи ее условия должны быть переформулированы как описания характерных особенностей выборки.

Объемом выборки называют число входящих в нее элементов (с учетом их повторения).

Когда выборки описаны, переходят к следующему шагу - подсчету числа выборок, соответствующих описанию. При этом используют ряд правил, одно из которых - **правило произведения**.

Пусть нам нужно подсчитать число упорядоченных выборок объема k , удовлетворяющих некоторым условиям. Предположим, что построение такой произвольной выборки можно разбить на k последовательных шагов так, чтобы на каждом шаге выбирать один из элементов выборки. При этом на первом шаге имеется выбор из n_1 возможностей; независимо от результата первого шага на втором шаге есть выбор из n_2 возможностей; независимо от результата первых двух шагов на третьем шаге есть выбор из n_3 возможностей и т.д., и, наконец, независимо от того, какой выбор был сделан на всех предыдущих шагах, на последнем шаге у нас имеется выбор из n_k возможностей. Тогда общее число интересующих нас упорядоченных выборок равно $n_1 \cdot n_2 \cdot n_3 \cdot \dots \cdot n_k$.

Вернемся к обсуждению примера. Проанализируем ситуацию, используя понятие выборки. Рассмотрим множество, элементами которого являются члены команды легкоатлетов. Чтобы сформировать команду, нужно отобрать из этого множества три элемента, т.е. составить выборку объема 3. В этой выборке элементы должны идти в определенном порядке: первым - преподаватель, вторым - аспирант, третьим - студент. Таким образом, команда - это выборка упорядоченная, и вопрос, сколькими способами можно сформировать команду, можно заменить вопросом, сколько имеется упорядоченных выборок объема 3, в которых первым элементом является преподаватель, вторым - аспирант, третьим - студент.

Любую такую выборку можно сформировать за три шага: вначале выбрать первый элемент, затем - второй и, наконец, третий. На первом шаге у нас есть 2 возможности, на втором - 4 возможности, на третьем - 3 возможности. Согласно правилу произведения, общее число интересующих нас выборок равно произведению $2 \cdot 4 \cdot 3 = 24$.

Пример 2. Сколько имеется четырехзначных чисел, составленных из цифр 1,2,3,4,5,6,7,8,9, в десятичной записи которых все цифры различны?

◀ Четырехзначное число, удовлетворяющее условию задачи, можно рассматривать как упорядоченную выборку объема 4, составленную из цифр 1,2,3,4,5,6,7,8,9, в которой элементы не повторяются. Построение такой выборки можно осуществить за четыре шага. На первом шаге выбирается первая цифра числа. Этот выбор можно осуществить 9

способами. На втором шаге выбирается вторая цифра числа. Так как по условию вторая цифра не должна совпадать с первой, то ее выбор зависит от выбора первой цифры, однако эта зависимость не распространяется на число возможностей выбора второй цифры (при каждом выборе, сделанном на первом шаге, имеется 8 вариантов выбора второй цифры). На третьем шаге выбирается третья цифра. Так как она должна быть отлична от первых двух, то на третьем шаге независимо от выбора, сделанного на первых двух шагах, у нас есть выбор из 7 возможностей. На четвертом шаге наш выбор сужается до 6 возможностей. Следовательно, согласно правилу произведения, количество чисел, удовлетворяющих условию, равно $9 \cdot 8 \cdot 7 \cdot 6 = 3024$. ►

Пример 3. Сколько подмножеств имеет n -элементное множество?

◀ Обозначим множество буквой A и занумеруем его элементы номерами от 1 до n :

$A = \{a_1, a_2, \dots, a_n\}$. Рассмотрим упорядоченные наборы из 0 и 1 длины n . Между

множеством таких наборов и множеством подмножеств A (булеаном A) установим

соответствие, сопоставив каждому подмножеству набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$, в котором на i -м месте стоит 1, если элемент a_i входит в подмножество, и 0, если не входит.

Например, если $A = \{a_1, a_2, a_3, a_4\}$, то подмножеству $\{a_1, a_4\}$ соответствует набор

$(1, 0, 0, 1)$. Пустому подмножеству соответствует набор из одних нулей, а самому A - набор из одних единиц.

Соответствие, которое мы установили между подмножеством A и множеством упорядоченных наборов из 0 и 1 длины n , является взаимно-однозначным, значит, мощности этих множеств совпадают (см. § 1.1). Таким образом, исходная задача свелась к определению числа упорядоченных наборов из 0 и 1 длины n . Каждый такой набор, по сути, является упорядоченной выборкой объема n из множества $\{0, 1\}$, которую можно построить за n этапов. На первом этапе выбирается значение α_1 (0 или 1), на втором - α_2 (0 или 1) и т.д. Следовательно, на каждом этапе есть выбор из двух возможностей и по правилу произведения число наборов равно $\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_n = 2^n$. Таким образом, n -элементное

множество имеет 2^n подмножеств. ►

Упражнение 1.4. У Олега десять книг, а у Ивана - двадцать. Сколькими способами можно осуществить обмен одной книги Олега на одну книгу Ивана?

Упражнение 1.5. Подсчитать число четных четырехзначных чисел, составленных из цифр 1, 2, 3, 4, 5, 6, 7, 8, 9, в десятичной записи которых соседние цифры различны.

Упражнение 1.6. Из города A в город B ведут четыре дороги. Сколькими способами можно съездить из A в B и обратно, если:

- а) путешествие туда и обратно совершается по разным дорогам;
- б) дороги туда и обратно выбираются независимо друг от друга?

При подсчете числа элементов в конечных множествах также широко используют **правило суммы**, впервые упомянутое в § 1.1. Напомним его формулировку.

Если A_1, A_2, \dots, A_n - конечные попарно-непересекающиеся множества, то множество $A_1 \cup A_2 \cup \dots \cup A_n$ также конечно и

$$|A_1 \cup A_2 \cup \dots \cup A_n| = |A_1| + |A_2| + \dots + |A_n|.$$

Пример 4. Сколько имеется натуральных чисел, меньших 10000, в десятичной записи которых все цифры различны?

◀ Множество натуральных чисел, удовлетворяющих условию, можно разбить на четыре подмножества: четырехзначных, трехзначных, двухзначных и однозначных чисел. Подсчитаем по отдельности количество чисел, входящих в каждое из этих подмножеств.

Каждое четырехзначное число, в десятичной записи которого все цифры различны, можно выписать за четыре шага: на первом шаге выбрать первую цифру числа, на втором - вторую и т.д. На первом шаге можно выбрать любую цифру, за исключением 0, т.е. имеется выбор из 9 возможностей. На втором шаге можно выбрать любую цифру, за исключением уже выбранной первой, значит, вновь имеем 9 возможностей. На третьем шаге число вариантов выбора сокращается до 8, а на четвертом - до 7. Следовательно, количество четырехзначных чисел, в десятичной записи которых все цифры различны, равно $9 \cdot 9 \cdot 8 \cdot 7 = 4536$. Рассуждая аналогично, находим, что количество трехзначных чисел, удовлетворяющих условию, равно $9 \cdot 9 \cdot 8 = 648$, количество двухзначных равно $9 \cdot 9 = 81$. Очевидно, что количество однозначных натуральных чисел равно 9.

Количество натуральных чисел, меньших 10000, в десятичной записи которых все цифры различны, найдем по правилу суммы: $4536 + 648 + 81 + 9 = 5274$. ▶

Упражнение 1.7. Сколько имеется шестизначных чисел, в десятичной записи которых «четные» (0, 2, 4, 6, 8) и «нечетные» (1, 3, 5, 7, 9) цифры чередуются?

Упражнение 1.8. Сколько имеется пятизначных чисел, в десятичной записи которых встречаются одинаковые цифры?

2. Сочетания и размещения. Напомним, что выборка называется *упорядоченной*, если порядок следования элементов в ней имеет значение (две упорядоченные выборки, имеющие одинаковый состав, но отличающиеся порядком следования элементов, считаются различными).

Если порядок следования элементов в выборке не является существенным, то выборка называется *неупорядоченной* (две неупорядоченные выборки считаются различными только тогда, когда они отличаются составом).

В выборках могут допускаться или не допускаться повторения элементов. Если повторения элементов не допускаются, то выборка называется *выборкой без повторений*; если повторения элементов допускаются - *выборкой с повторениями*. В выборках без повторений все элементы попарно различны.

Комбинируя эти свойства выборок, получают четыре основные разновидности выборок:

- 1) упорядоченная выборка объема k без повторений называется *размещением из n элементов по k* ;
- 2) неупорядоченная выборка объема k без повторений называется *сочетанием из n элементов по k* ;
- 3) упорядоченная выборка объема k с повторениями называется *размещением с повторением из n элементов по k* ;
- 4) неупорядоченная выборка объема k с повторениями называется *сочетанием с повторением из n элементов по k* .

Договоримся для обозначения упорядоченных выборок использовать круглые скобки, а для обозначения неупорядоченных - фигурные.

Проиллюстрируем введенные понятия на примере. Пусть множество M состоит из трех элементов $M = \{a, b, c\}$. Рассмотрим различные виды выборок объема 2. Имеем:

- 1) шесть упорядоченных выборок без повторений, т.е. размещений из 3 по 2:

$$(a, b), (b, a), (a, c), (c, a), (b, c), (c, b);$$

- 2) три неупорядоченные выборки без повторений, т.е. сочетаний из 3 по 2:

$$\{a, b\}, \{a, c\}, \{b, c\};$$

- 3) девять упорядоченных выборок с повторениями, т.е. размещений с повторениями из 3 по 2:

$$(a, b), (b, a), (a, c), (c, a), (b, c), (c, b), (a, a), (b, b), (c, c);$$

- 4) шесть неупорядоченных выборок с повторениями, т.е. сочетаний с повторениями из 3 по 2:

$$\{a,b\}, \{a,c\}, \{b,c\}, \{a,a\}, \{b,b\}, \{c,c\}.$$

Особый интерес представляет частный случай размещений, когда объем выборки совпадает с числом элементов множества, т.е. размещений из n элементов по n . Они называются *перестановками* n элементов.

В качестве примера перечислим все перестановки множества $A = \{a,b,c\}$:

$$(a,b,c), (b,a,c), (a,c,b), (c,a,b), (b,c,a), (c,b,a).$$

Отметим, что размещения из n элементов по k по сути являются упорядоченными последовательностями, состоящими из k различных элементов заданного множества M . Сочетания из n элементов по k являются подмножествами из k элементов заданного n -элементного множества.

Рассмотрим формулы подсчета числа сочетаний и размещений.

Число размещений. Обозначим число размещений из n по k через A_n^k (другое обозначение - $(n)_k$). Для подсчета числа A_n^k используем правило произведения. При построении конкретного размещения первым элементом в нем можно взять любой из n элементов множества A , вторым элементом - любой из $n-1$ оставшихся в A элементов и т.д. Поэтому

$$A_n^k = n(n-1)\dots(n-k+1) = \frac{n!}{(n-k)!} \text{ при } 1 \leq k \leq n.$$

При $k > n$ не существует размещений из n по k , следовательно, $A_n^k = 0$; при $k = 0$ полагаем $A_0^0 = A_n^0 = 1$.

Поскольку перестановки из n элементов - частный случай размещений из n элементов по k , когда $k = n$, то их число равно $A_n^n = n \cdot (n-1) \cdot \dots \cdot 2 \cdot 1 = n!$ Заметим, что число перестановок можно интерпретировать как число способов, которыми можно упорядочить множество, состоящее из n элементов.

Например, если в группе 20 человек, то список группы можно написать 20! способами.

Число сочетаний. Обозначим число сочетаний из n по k через C_n^k (другое обозначение - $\binom{n}{k}$). Поскольку в сочетании, в отличие от размещения, порядок следования

элементов не учитывается, то из одного сочетания из n элементов по k путем упорядочивания элементов можно получить $k!$ размещений. Значит,

$$C_n^k = \frac{A_n^k}{k!} = \frac{n!}{k!(n-k)!}, \quad 1 \leq k \leq n.$$

Из последней формулы следует

$$C_n^n = 1 \text{ и } C_n^k = C_n^{n-k}.$$

При $k=0$ полагают $C_0^0 = C_n^0 = 1$, а при $k > n$ - $C_n^k = 0$ (поскольку при $k > n$ не существует сочетаний из n элементов по k).

Число размещений с повторениями. Для числа размещений с повторениями из n элементов по k используется обозначение A_n^k . Для подсчета числа размещений с повторениями из n элементов по k используем правило произведения. При построении конкретного размещения первым элементом в нем можно взять любой из n элементов множества A , вторым элементом - также любой из n элементов множества A и т.д. Таким образом, число размещений из n элементов по k равно

$$\underbrace{n \cdot n \cdot \dots \cdot n}_{k \text{ раз}} = n^k.$$

Число сочетаний с повторениями. Для числа сочетаний с повторениями из n по k используется обозначение C_n^k . Можно показать, что $C_n^k = C_{n+k-1}^k$. Доказательство этой формулы опустим.

Рассмотрим несколько примеров использования комбинаторных формул.

Пример 5. а) Сколько пятибуквенных «слов» можно составить в алфавите из 9 букв, если буквы в «словах» не должны повторяться?

б) Сколько пятибуквенных «слов» можно составить в алфавите из 9 букв, если буквы в словах могут повторяться?

◀ а) Прежде всего заметим, что под «словом» в комбинаторных задачах понимают любую конечную последовательность букв. Каждому пятибуквенному «слову» можно сопоставить упорядоченную выборку, элементами которой являются буквы алфавита: первый элемент выборки - первая буква в «слове», второй элемент - вторая буква в «слове» и т.д. В этих упорядоченных выборках элементы не повторяются (так как буквы в «словах» не должны по условию быть одинаковыми), следовательно, мы имеем дело с размещениями из 9 элементов по 5. Всего таких размещений $A_9^5 = \frac{9!}{4!} = 15120$.

б) Каждому пятибуквенному «слову» можно сопоставить упорядоченную выборку, элементами которой являются буквы алфавита: первый элемент выборки - первая буква в «слове», второй элемент - вторая буква в «слове» и т.д. В отличие от ситуации а), элементы в этих выборках могут повторяться (буквы в «словах» по условию могут быть одинаковыми), следовательно, мы имеем дело с размещениями с повторениями из 9 элементов по 5. Всего таких размещений $A_9^5 = 9^5$. ►

Пример 6. В магазине продаются воздушные шары 7 цветов. Игорь решил купить для праздника 3 шара.

а) Сколькими способами Игорь может выбрать шары, если он хочет, чтобы шары отличались по цвету?

б) Сколькими способами Игорь может выбрать шары, если ему все равно, будут они отличаться по цвету или нет?

◀ В случае а) каждый выбор Игоря можно интерпретировать как неупорядоченную выборку без повторений из 7 цветов по 3, т.е. сочетание без повторений из 7 по 3. Число таких сочетаний $C_7^3 = \frac{7!}{3!4!} = 35$.

Отличие случая б) от случая а) состоит в том, что Игорь может купить 2 или 3 шара одного цвета. Значит, каждый выбор Игоря можно интерпретировать как неупорядоченную выборку с повторениями из 7 цветов по 3, т.е. сочетание с повторениями из 7 по 3. Число таких сочетаний $C_7^3 = C_{7+3-1}^3 = C_9^3 = \frac{9!}{3!6!} = 84$. ►

Пример 7. Имеется 10 карточек, на которых написаны цифры от 0 до 9. Сколькими способами можно выложить их в ряд таким образом, чтобы карточки, на которых записаны цифры 1,2,3, лежали рядом?

◀ Построение произвольной перестановки, удовлетворяющей условию задачи, разобьем на три шага: на первом выберем 3 последовательных места для карточек с цифрами 1,2,3; на втором расставим эти 3 карточки по выбранным местам; на третьем расположим остальные 7 карточек на оставшихся 7 местах.

На первом шаге имеется выбор из 8 возможных вариантов (выбрать места с 1 по 3 или места со 2 по 4 и т.д. или, наконец, места с 8 по 10). На втором шаге имеется выбор из 3! вариантов. На третьем шаге имеется выбор из 7! вариантов расположения оставшихся карточек на 7 свободных местах. Согласно правилу произведения, ответом на вопрос задачи является число $8 \cdot 3! \cdot 7! = 6 \cdot 8!$. ►

Пример 8. Сколько различных шестизначных чисел можно получить, выкладывая в ряд карточки с цифрами от 1 до 9 так, чтобы на первых трех местах стояли четные, а на последних трех - нечетные цифры?

◀ Каждое число, удовлетворяющее условию задачи, можно выложить за два шага: на первом положить в ряд 3 карточки с четными цифрами, на втором - 3 карточки с нечетными цифрами. Число возможностей на первом шаге равно числу упорядоченных выборок объема 3 без повторений, элементами которых являются четные цифры, т.е. числу размещений из 4 по 3: A_4^3 . Число возможностей на втором шаге равно числу упорядоченных выборок объема 3 без повторений, элементами которых являются нечетные цифры, т.е. числу размещений из 5 по 3: A_5^3 . Согласно правилу произведения, общее число чисел, удовлетворяющих условию задачи, равно $A_4^3 \cdot A_5^3 = \frac{4!}{1!} \cdot \frac{5!}{2!} = 24 \cdot 60 = 1440$. ▶

Пример 9. У Саши 10 марок, а у Вани - 20. Сколькими способами можно осуществить обмен 3 Сашиних марок на 3 Ванины?

◀ Для каждого обмена Саша должен отобрать 3 марки из 10. Он может это сделать C_{10}^3 способами, поскольку каждый результат отбора можно интерпретировать как неупорядоченную выборку без повторений 3 элементов из 10, т.е. сочетание из 10 по 3. В свою очередь Ваня может отобрать 3 марки для обмена C_{20}^3 способами. Каждому обмену можно однозначно сопоставить упорядоченную пару, первый элемент которой - набор марок, приготовленный для обмена Сашей, а второй - набор марок, приготовленный для обмена Ваней. Согласно правилу произведения, число таких пар, а значит, и число способов обмена равно произведению $C_{10}^3 \cdot C_{20}^3$. ▶

Упражнение 1.9. а) Сколькими способами можно разместить 5 пронумерованных шаров по 9 пронумерованным коробкам, если в 1 коробку можно положить не более 1 шара?

б) Сколькими способами можно разместить 5 пронумерованных шаров по 9 пронумерованным коробкам, если в 1 коробку можно положить неограниченное число шаров?

Упражнение 1.10. Сколько разных шестибуквенных слов можно получить, переставляя буквы в слове «фартук»?

Упражнение 1.11. а) Сколькими способами на одной полке можно разместить 6 книг по физике и 6 книг по математике так, чтобы книги по физике стояли правее книг по математике?

б) Сколькими способами на одной полке можно разместить 6 книг по физике и 6 книг по математике так, чтобы книги по физике чередовались с книгами по математике?

Упражнение 1.12. а) Сколькими способами Маша может выбрать 2 предмета из 8 для сдачи экзамена по выбору?

б) Маша должна сдать 2 экзамена за 8 дней. Сколькими способами она может составить расписание экзаменов, если нельзя сдавать больше одного экзамена в день?

Упражнение 1.13. В киоске продаются 10 видов рождественских поздравительных открыток. Тане нужно купить 8 открыток. Сколькими способами Таня сможет это сделать, если:

- а) она решила купить открытки только разных видов;
- б) она решила купить по две открытки четырех видов;
- в) Тане все равно, какие открытки покупать;
- г) одна из открыток понравилась Тане больше других, и она решила купить хотя бы одну такую открытку?

Упражнение 1.14. Собрание из 50 человек выбирает председателя, секретаря и трех членов счетной комиссии. Сколькими способами это можно сделать?

3. Некоторые комбинаторные соотношения. Числа C_n^k называются также *биномиальными коэффициентами*, поскольку они фигурируют в функциональном тождестве, называемом формулой бинома Ньютона:

$$(1+x)^n = C_n^0 + C_n^1 x + \dots + C_n^k x^k + \dots + C_n^n x^n.$$

Доказательство формулы приведено во второй части параграфа.

Для чисел C_n^k выполняется тождество $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$. Действительно,

$$\begin{aligned} C_{n-1}^k + C_{n-1}^{k-1} &= \frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!} = \\ &= \frac{(n-1)!}{(k-1)!k(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k-1)!(n-k)} = \\ &= \frac{(n-1)!}{(k-1)!(n-k-1)!} \left(\frac{1}{k} + \frac{1}{n-k} \right) = \\ &= \frac{(n-1)!}{(k-1)!(n-k-1)!} \cdot \frac{n}{k(n-k)} = \frac{n!}{k!(n-k)!} = C_n^k. \end{aligned}$$

Из этого тождества вытекает эффективный способ рекуррентного вычисления биномиальных коэффициентов, который можно представить в виде так называемого *треугольника Паскаля*:

2) *индуктивный переход*, состоящий в том, что, полагая справедливым утверждение $P(n_0)$ ($n_0 \geq N_0$), доказывают справедливость утверждения $P(n_0 + 1)$.

Пример 10. Доказать методом математической индукции формулу бинома Ньютона

$$(1+x)^n = C_n^0 + C_n^1 x + \dots + C_n^n x^n.$$

◀ 1. Базис индукции. При $n=1$ имеем $(1+x)^1 = C_1^0 + C_1^1 x$. Поскольку $C_n^0 = C_1^1 = 1$, формула верна.

2. Индуктивный переход. Докажем, что из предположения о том, что верно равенство

$$(1+x)^{n_0} = C_{n_0}^0 + C_{n_0}^1 x + \dots + C_{n_0}^{n_0-1} x^{n_0-1} + C_{n_0}^{n_0} x^{n_0},$$

следует, что верно равенство

$$(1+x)^{n_0+1} = C_{n_0+1}^0 + C_{n_0+1}^1 x + \dots + C_{n_0+1}^{n_0} x^{n_0} + C_{n_0+1}^{n_0+1} x^{n_0+1},$$

полученное из формулы бинома Ньютона заменой n на $n_0 + 1$.

В самом деле, имеем

$$\begin{aligned} (1+x)^{n_0+1} &= (1+x)^{n_0} (1+x) = \\ &= \left(C_{n_0}^0 + C_{n_0}^1 x + C_{n_0}^2 x^2 + \dots + C_{n_0}^{n_0-1} x^{n_0-1} + C_{n_0}^{n_0} x^{n_0} \right) (1+x) = \\ &= C_{n_0}^0 + C_{n_0}^1 x + C_{n_0}^2 x^2 + \dots + C_{n_0}^{n_0-1} x^{n_0-1} + C_{n_0}^{n_0} x^{n_0} + \\ &+ C_{n_0}^0 x + C_{n_0}^1 x^2 + C_{n_0}^2 x^3 + \dots + C_{n_0}^{n_0-1} x^{n_0} + C_{n_0}^{n_0} x^{n_0+1} = \\ &= C_{n_0}^0 + \left(C_{n_0}^0 + C_{n_0}^1 \right) x + \dots + \left(C_{n_0}^{n_0-1} + C_{n_0}^{n_0} \right) x^{n_0} + C_{n_0}^{n_0} x^{n_0+1} =^* \\ &= C_{n_0+1}^0 + C_{n_0+1}^1 x + C_{n_0+1}^2 x^2 + \dots + C_{n_0+1}^{n_0} x^{n_0} + C_{n_0+1}^{n_0+1} x^{n_0+1} \end{aligned}$$

(при переходе, помеченном (*), были использованы тождество $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$ и равенства $C_{n_0}^0 = 1$, $C_{n_0+1}^0 = 1$, $C_{n_0}^{n_0} = 1$, $C_{n_0+1}^{n_0+1} = 1$).

Таким образом, формула бинома Ньютона справедлива для любого $n \geq 1$. ►

Приведенная формулировка принципа математической индукции допускает равносильные варианты. В некоторых случаях мы будем использовать вариант, в котором индуктивный переход состоит в предположении справедливости утверждения $P(n)$ для $n \leq n_0$ и доказательстве его справедливости для $n = n_0 + 1$.

Для доказательства комбинаторных соотношений также пользуются аппаратом математического анализа. Проиллюстрируем это на примере.

Пример 11. Доказать, что при любом натуральном n выполняется равенство

$$C_{2n}^n = (C_n^0)^2 + (C_n^1)^2 + (C_n^2)^2 + \dots + (C_n^n)^2.$$

◀ Воспользуемся тем, что при всех действительных значениях x выполняется равенство $(1+x)^{2n} = (1+x)^n \cdot (1+x)^n$. Заменим $(1+x)^{2n}$ и $(1+x)^n$ их биномиальными разложениями:

$$\begin{aligned} C_{2n}^0 + \dots + C_{2n}^n x^n + \dots + C_{2n}^{2n} x^{2n} = \\ = (C_n^0 + C_n^1 x + \dots + C_n^n x^n) (C_n^0 + C_n^1 x + \dots + C_n^n x^n). \end{aligned}$$

Поскольку мы имеем тождество относительно x , коэффициенты при одинаковых степенях x в обеих его частях должны быть одинаковыми. Рассмотрим коэффициенты при x^n . В левой части интересующий нас коэффициент равен C_{2n}^n . Чтобы понять, чему равен коэффициент при x^n в правой части, перепишем ее, изменив порядок слагаемых во втором множителе:

$$\begin{aligned} (C_n^0 + C_n^1 x + C_n^2 x^2 + \dots + C_n^{n-1} x^{n-1} + C_n^n x^n) \times \\ \times (C_n^n x^n + C_n^{n-1} x^{n-1} + C_n^{n-2} x^{n-2} + \dots + C_n^1 x^1 + C_n^0). \end{aligned}$$

Теперь видно, что слагаемые, содержащие x^n , получаются при перемножении друг на друга первых, вторых, третьих и т.д. слагаемых, стоящих в скобках. Следовательно, верно равенство

$$C_{2n}^n = C_n^0 \cdot C_n^n + C_n^1 \cdot C_n^{n-1} + C_n^2 \cdot C_n^{n-2} + \dots + C_n^n \cdot C_n^0.$$

Учитывая, что $C_n^k = C_n^{n-k}$, окончательно получаем:

$$C_{2n}^n = (C_n^0)^2 + (C_n^1)^2 + (C_n^2)^2 + \dots + (C_n^n)^2. \blacktriangleright$$

Задачи повышенной сложности

1.11. Сколько имеется четных четырехзначных чисел, в десятичной записи которых все числа различны?

1.12. Сколько имеется четырехзначных чисел, в десятичной записи которых встречается цифра 5?

1.13. Сколько различных каруселей можно сделать, расположив по окружности фигурки десяти зверей (карусели считаются одинаковыми, если фигурки идут друг за другом в одинаковом порядке)?

1.14. а) Сколько разных шестибуквенных слов можно получить, переставляя буквы в слове «физика»?

б) Сколько разных восьмибуквенных слов можно получить, переставляя буквы в слове «черчение»?

1.15. Сколькими способами можно разложить по 8 пронумерованным коробкам трех синих и пяти красных шаров так, чтобы в каждой коробке оказалось ровно по одному шару?

1.16. Доказать, что число упорядоченных разбиений n -элементного множества на k подмножеств, первое из которых содержит n_1 элемент, второе - n_2 элемента, ..., k -е - n_k элементов, равно $\frac{n!}{n_1!n_2!\dots n_k!}$.

1.17. а) Сколько имеется пятизначных чисел, в десятичной записи которых цифры расположены в порядке убывания?

б) Сколько имеется пятизначных чисел, в десятичной записи которых каждая следующая цифра меньше либо равна предыдущей?

1.18. В студенческой группе учатся 9 девушек и 11 юношей. Сколькими способами можно сформировать команду из 7 человек для участия в соревнованиях, если в нее должно войти не менее 3 юношей?

1.19. а) В классе учатся 18 человек. Сколькими способами можно составить график дежурств по классу на 6 дней так, чтобы каждый день дежурили по 3 человека, причем никто не дежурил дважды?

б) В классе учатся 18 человек. Сколькими способами можно разбить его учеников на 6 групп?

1.20. а) Сколькими способами можно разложить 8 одинаковых шаров по 3 пронумерованным коробкам так, чтобы ни одна из коробок не осталась пустой?

б) Сколькими способами можно разложить 8 одинаковых шаров по 3 пронумерованным коробкам?

1.21. а) Сколькими способами можно представить число k в виде упорядоченной суммы n положительных целых чисел?

б) Сколькими способами можно представить число k в виде упорядоченной суммы n неотрицательных целых чисел?

1.22. Вывести формулу $\mathcal{C}_n^k = C_{n+k-1}^k$ для числа сочетаний с повторениями из n элементов по k .

1.23. а) Сколько существует рефлексивных бинарных соотношений на множестве из n элементов?

б) Сколько существует симметричных бинарных соотношений на множестве из n элементов?

в) Сколько существует антисимметричных бинарных соотношений на множестве из n элементов?

1.24. Используя бином Ньютона, доказать тождества:

а) $1 + C_n^1 + C_n^2 + \dots + C_n^k = 2^n$;

б) $1 - C_n^1 + C_n^2 - \dots + (-1)^n \cdot C_n^k = 0$;

в) $C_n^0 + C_n^2 + \dots + C_n^n = 2^{n-1}$ (n - четно);

г) $C_n^1 + C_n^3 + \dots + C_n^{n-1} = 2^{n-1}$ (n - четно);

д) $C_n^0 + C_n^2 + \dots + C_n^{n-1} = 2^{n-1}$ (n - нечетно);

е) $C_n^1 + C_n^3 + \dots + C_n^n = 2^{n-1}$ (n - нечетно).

1.25. Найти такое число k , при котором число сочетаний из n элементов по k наибольшее, при условии, что:

а) n - четное число; б) n - нечетное число.

Глава 2. Функции алгебры логики

§ 2.1. Булевы функции и способы их задания

Булев вектор. Функции алгебры логики (булевы функции). Задание булевых функций таблицей истинности и вектором значений. Элементарные булевы функции одной и двух переменных. Формулы над множеством функций, задание функций формулами, равносильные формулы. Доказательство равносильности формул с использованием таблиц истинности. Основные равносильности над множеством $\{\vee, \wedge, \neg, 0, 1\}$. Упрощение формул методом равносильных преобразований. Фиктивные и существенные переменные, равные функции, алгоритм удаления и введения фиктивных переменных.

Базовые понятия и утверждения

1. Булевы векторы и булевы функции от n переменных. Булевым вектором называется упорядоченный набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$, где α_i принимают значения 0 или 1.

Числа α_i называют *координатами* вектора, число n - его *длиной*.

Для краткого обозначения вектора $(\alpha_1, \alpha_2, \dots, \alpha_n)$ используют символ α .

Например, $(0,0), (0,1), (1,0), (1,1)$ - булевы векторы длины 2; $(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)$ - булевы векторы длины 3.

Число булевых векторов длины n равно 2^n . Действительно, с точки зрения комбинаторики булев вектор длины n - упорядоченная выборка элементов множества $\{0,1\}$, которую можно построить за n шагов (на первом шаге выбрать первую координату, на втором - вторую и т.д.). Для выбора каждой координаты есть два варианта (0 или 1). Следовательно, согласно правилу произведения (см. § 1.2), последовательный выбор n координат можно сделать $\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_n = 2^n$ способами.

Множество булевых векторов обозначают B^n и называют *единичным n -мерным кубом*.

Пример 1. Найти число булевых векторов длины 16, у которых

$$\alpha_1 = \alpha_2, \alpha_3 = \alpha_4, \dots, \alpha_{15} = \alpha_{16}.$$

◀ Для подсчета используем правило произведения. Чтобы составить требуемый вектор, нам достаточно последовательно выбрать восемь координат $\alpha_1, \alpha_3, \alpha_5, \dots, \alpha_{15}$. В

качестве каждой из них можно взять одно из двух чисел: 0 или 1. Таким образом, общее число векторов требуемого вида равно $\underbrace{2 \cdot 2 \cdot 2 \cdot 2 \cdot 2}_8 = 2^8 = 256$. ►

Упражнение 2.1. Сколько существует булевых векторов длины 8, которые одинаково читаются слева направо и справа налево?

Число $v(\mathbf{a}) = \alpha_1 \cdot 2^{n-1} + \alpha_2 \cdot 2^{n-2} + \dots + \alpha_k \cdot 2^{n-k} + \dots + \alpha_n \cdot 2^0$ называют *номером* булева вектора $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_n)$.

Пример 2. Вектор $\mathbf{a} = (1, 1, 1, 0, 1)$ имеет номер

$$v(\mathbf{a}) = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 29.$$

В дальнейшем нам часто придется перечислять булевы векторы от фиксированного числа переменных в порядке возрастания их номеров. Например, для булевых векторов длины 2 этот порядок таков: (0,0), (0,1), (1,0), (1,1). В этом легко убедиться, вычислив номера векторов:

$$v(0,0) = 0 \cdot 2^1 + 0 \cdot 2^0 = 0, \quad v(0,1) = 0 \cdot 2^1 + 1 \cdot 2^0 = 1,$$

$$v(1,0) = 1 \cdot 2^1 + 0 \cdot 2^0 = 2, \quad v(1,1) = 1 \cdot 2^1 + 1 \cdot 2^0 = 3.$$

Упражнение 2.2. Перечислить в порядке возрастания номеров булевы векторы длины 3.

Введем понятие функции алгебры логики или, что то же, булевой функции.

Определение. Функция, определенная на B^n и принимающая значение 0 или 1, называется *булевой функцией* от n переменных.

Для обозначения булевых функций используют символы f, g, h , а для обозначения аргументов - символы x, y, z, t (часто с индексами). Таким образом, запись $f(x_1, x_2, \dots, x_n)$ воспринимается как обозначение функции от n переменных (аргументов).

Чтобы задать булеву функцию f от n переменных, достаточно указать ее значение (0 или 1) на каждом наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ значений аргументов x_1, x_2, \dots, x_n . Для задания булевой функции удобно использовать *таблицу истинности* (табл. 2.1). В каждой строке такой таблицы вначале идет набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$ значений переменных

x_1, x_2, \dots, x_n , а затем значение функции на этом наборе. Булевы векторы перечисляются сверху вниз в порядке возрастания номеров.

Таблица 2.1

| x_1 | x_2 | \vdots | x_{n-1} | x_n | $f(x_1, x_2, \dots, x_{n-1}, x_n)$ |
|----------|----------|----------|-----------|----------|------------------------------------|
| 0 | 0 | \vdots | 0 | 0 | $f(0, 0, \dots, 0, 0)$ |
| 0 | 0 | \vdots | 0 | 1 | $f(0, 0, \dots, 0, 1)$ |
| 0 | 0 | \vdots | 1 | 0 | $f(0, 0, \dots, 1, 0)$ |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| 1 | 1 | \vdots | 1 | 1 | $f(1, 1, \dots, 1, 1)$ |

Таблицы истинности булевых функций от одного числа аргументов отличаются лишь последним столбцом. Поэтому булеву функцию можно также задать *вектором значений*, выписав его по правому столбцу таблицы истинности. Поскольку помимо строки заголовков таблица содержит 2^n строк (столько, сколько имеется булевых векторов длины n), то вектор значений булевой функции от n переменных имеет длину 2^n .

В частности, чтобы задать какую-нибудь булеву функцию от двух переменных, нужно каждому элементу множества $B^2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ поставить в соответствие либо 0, либо 1.

Таблица 2.2 Пример 3. Рассмотрим функцию $f(x, y)$ такую, что $f(0, 0) = 1$,

| x | y | f |
|-----|-----|-----|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$f(0, 1) = 1$, $f(1, 0) = 1$, $f(1, 1) = 0$. Эту же функцию зададим таблицей истинности (табл. 2.2).

По последнему столбцу таблицы выпишем вектор значений функции $f(x, y) = (1, 1, 1, 0)$.

Множество булевых функций от n переменных обозначают $P_2(n)$, а множество всех булевых функций - P_2 .

Число булевых функций от n переменных $|P_2(n)| = 2^{2^n}$. Действительно, каждая булева функция от n переменных однозначно определяется вектором значений длины 2^n , поэтому булевых функций от n переменных столько, сколько булевых векторов длины 2^n , т.е. 2^{2^n} .

В частности, число булевых функций от одной переменной равно 2, от двух переменных - 4, от трех переменных - 16.

2. Элементарные булевы функции. В табл. 2.3, 2.4 представлены все булевы функции от одной и двух переменных.

Таблица 2.3

| x | g_0 | g_1 | g_2 | g_3 |
|-----|-------|-------|-------|-------|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Таблица 2.4

| x | y | f_0 | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 |
|-----|-----|-------|-------|----------|----------|----------|----------|----------|----------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | | | | | | | | |
| x | y | f_8 | f_9 | f_{10} | f_{11} | f_{12} | f_{13} | f_{14} | f_{15} |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Ниже перечислены булевы функции, которые наиболее часто употребляются и потому считаются «элементарными». Эти функции имеют собственные обозначения и названия:

- 1) g_0 называется *тождественным нулем* и обозначается 0;
- 2) g_1 называется *тождественной функцией* и обозначается по имени переменной (в нашем случае x);
- 3) g_2 называется *отрицанием x* , обозначается \bar{x} или $\neg x$ и читается «не x »;
- 4) g_3 называется *тождественной единицей* и обозначается 1;
- 5) f_1 называется *конъюнкцией*, обозначается $x \wedge y$ или $x \& y$, или $x \cdot y$ и читается « x и y »;
- 6) f_6 называется *сложением по модулю 2*, обозначается $x \oplus y$ и читается « x плюс y »;
- 7) f_7 называется *дизъюнкцией*, обозначается $x \vee y$ и читается « x или y »;
- 8) f_8 называется *стрелкой Пирса*, обозначается $x \downarrow y$;
- 9) f_9 называется *эквивалентностью*, обозначается $x \leftrightarrow y$ и читается « x эквивалентно (равносильно) y »;
- 10) f_{13} называется *импликацией*, обозначается $x \rightarrow y$ и читается « x имплицирует (влечет) y ».
- 11) f_{14} называется *штрихом Шеффера* и обозначается $x|y$.

Символы $\neg, \wedge, \vee, \oplus, \downarrow, \leftrightarrow, \rightarrow, |$, с помощью которых обозначают элементарные функции, называются *логическими связками*.

Для удобства дальнейшего изложения приведем таблицы истинности элементарных функций с учетом введенных обозначений (табл. 2.5 и 2.6).

Таблица 2.5

| x | 0 | 1 | \bar{x} | x |
|-----|---|---|-----------|-----|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Таблица 2.6

| x | y | $x \wedge y$ | $x \oplus y$ | $x \vee y$ | $x \downarrow y$ | $x \leftrightarrow y$ | $x \rightarrow y$ | $x y$ |
|-----|-----|--------------|--------------|------------|------------------|-----------------------|-------------------|-------|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

3. Задание булевых функций формулами. Для задания булевых функций, помимо таблиц, используют формулы, которые обычно строят по «принципу матрешек», вкладывая друг в друга символические обозначения функций. Говоря о формуле, часто указывают, с использованием каких функций она строилась. В некоторых случаях нужно указывать и то, какие использовались переменные.

Например, $((x_1 \vee x_2) \rightarrow (x_2 | x_3))$ - формула над множеством функций, состоящим из дизъюнкции, импликации и штриха Шеффера с переменными из множества $\{x_1, x_2, x_3\}$, а $((x_2 \wedge (\neg x_1)) \leftrightarrow x_1)$ - формула над множеством функций, состоящим из конъюнкции, отрицания и эквивалентности, с переменными из множества $\{x_1, x_2\}$.

При составлении формулы над множеством функций \mathbf{B} с переменными из множества X можно в качестве аргументов функций из \mathbf{B} использовать любые переменные из X . Также на места аргументов функций из \mathbf{B} можно подставлять любые ранее составленные над \mathbf{B} формулы (их принято называть подформулами конечной формулы).

Например, $((x_1 \wedge x_1) \leftrightarrow (x_2 \vee (\neg x_2)))$ - формула над множеством функций, состоящим из конъюнкции, дизъюнкции и эквивалентности, с переменными из множества $\{x_1, x_2\}$, а $(x_1 \wedge x_1)$, $(\neg x_2)$, $(x_2 \vee (\neg x_2))$ - подформулы этой формулы.

Формула каждому набору значений аргументов ставит в соответствие значение некоторой функции и может служить, наряду с таблицей, способом задания этой функции. При этом говорят, что *формула задает (представляет, реализует)* функцию. Покажем на примерах, как построить таблицу истинности функции, заданной формулой.

Пример 4. Построить таблицу истинности и выписать вектор значений функции, заданной формулой:

а) $f(x_1, x_2) = ((x_2 \wedge (\neg x_1)) \leftrightarrow x_1)$;

б) $f(x_1, x_2, x_3) = ((x_1 \vee x_2) \rightarrow (x_2 | x_3))$.

◀ а) Пошаговое построение таблицы истинности функции $f(x_1, x_2) = ((x_2 \wedge (\neg x_1)) \leftrightarrow x_1)$ приведено в табл. 2.7.

Таблица 2.7

| x_1 | x_2 | $\neg x_1$ | $x_2 \wedge (\neg x_1)$ | $((x_2 \wedge (\neg x_1)) \leftrightarrow x_1)$ |
|-------|-------|------------|-------------------------|---|
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

Вектор значений функции $f(x_1, x_2) = (1000)$.

б) Пошаговое построение таблицы истинности функции

$f(x_1, x_2, x_3) = ((x_1 \vee x_2) \rightarrow (x_2 | x_3))$ приведено в табл. 2.8.

Таблица 2.8

| x_1 | x_2 | x_3 | $x_1 \vee x_2$ | $x_2 x_3$ | $((x_1 \vee x_2) \rightarrow (x_2 x_3))$ |
|-------|-------|-------|----------------|-------------|--|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Вектор значений функции $f(x_1, x_2, x_3) = (1111 \ 0111)$. ►

Упражнение 2.3. Построить таблицу истинности и выписать вектор значений функции, заданной формулой:

а) $f(x_1, x_2, x_3) = (((x_2 \downarrow x_1) \vee x_3) \oplus x_1)$;

б) $f(x_1, x_2, x_3) = ((1|(x_1 \wedge x_3)) \oplus (0 \downarrow x_2))$.

Строго понятия формулы над множеством функций и функции, заданной формулой, введем во второй части параграфа.

Для обозначения формулы над множеством функций $\mathbf{B} = \{f_1, f_2, \dots, f_k\}$ используют запись $\Phi[f_1, f_2, \dots, f_k]$ или $\Phi[\mathbf{B}]$. Если хотят обратить внимание на то, что при написании формулы использовались переменные x_1, x_2, \dots, x_n , то пишут $\Phi(x_1, x_2, \dots, x_n)$. Запись $f = \Phi[\mathbf{B}]$ означает, что формула $\Phi[\mathbf{B}]$ задает функцию f .

Функция f , заданная формулой над множеством функций $\{f_1, f_2, \dots\}$, называется *суперпозицией функций* $\{f_1, f_2, \dots\}$.

Для упрощения записи формул разрешается опускать у формул внешние скобки, вместо $\neg f$ писать \bar{f} ; опускать символ конъюнкции « \wedge » (« $\&$ », « \cdot »), т.е. вместо $f \wedge g$ ($f \& g$, $f \cdot g$) писать fg . Чтобы сократить число скобок в формуле, среди операций вводят «иерархию». Самой «старшей» операцией считают отрицание, следующей за ней - конъюнкцию, затем - дизъюнкцию и, наконец, импликацию и эквивалентность (их не упорядочивают). Если скобки не предписывают противное, операции в формулах выполняют в порядке старшинства.

Например, согласно введенным договоренностям, формулу $(x_1 \leftrightarrow (x_2 \wedge (\neg x_1)))$ можно записать в виде $x_1 \leftrightarrow x_2 \bar{x}_1$. Составляя таблицу истинности функции, заданной этой формулой, сначала следует найти отрицание x_1 , затем конъюнкцию x_2 и \bar{x}_1 и только затем эквивалентность x_1 и $x_2 \bar{x}_1$.

Упражнение 2.4. Учитывая договоренности о порядке выполнения операций, опустить «лишние» скобки и знак « \wedge » в формулах:

а) $x \wedge (y \wedge (\bar{x} \vee y))$;

б) $((x \vee y) \vee (x \wedge (y \wedge z))) \rightarrow ((\bar{x} \vee \bar{y}) \wedge z)$.

Одну и ту же функцию можно задать разными формулами. Покажем это на примерах.

Пример 5. Показать, что формулы $\bar{x} \vee xy$ и $\bar{x} \vee y$ задают одну функцию.

◀ Обозначим функцию, заданную первой формулой, f_1 , а функцию, заданную второй формулой, - f_2 (табл. 2.9).

Таблица 2.9

| x | y | \bar{x} | xy | $f_1 = \bar{x} \vee xy$ | $f_2 = \bar{x} \vee y$ |
|-----|-----|-----------|------|-------------------------|------------------------|
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

Сравнив последние два столбца таблицы, убедимся, что функции f_1 и f_2 на одинаковых векторах принимают равные значения. ►

Упражнение 2.5. Показать, что формулы $x \rightarrow y$ и $\bar{x} \vee y$ задают одну функцию.

Если формулы Φ_1 и Φ_2 задают одну функцию, то их называют *равносильными* и пишут $\Phi_1 = \Phi_2$.

Например, $\bar{x} \vee xy = \bar{x} \vee y$, $x \rightarrow y = \bar{x} \vee y$ (см. пример 5 и упражнение 2.5).

Теорема 2.1. Для формул над множеством $\{0,1,\wedge,\vee,\neg\}$ имеют место следующие равносильности:

1. $x \vee y = y \vee x$;
2. $x \wedge y = y \wedge x$;
3. $x \vee (y \vee z) = (x \vee y) \vee z$;
4. $x \wedge (y \wedge z) = (x \wedge y) \wedge z$;
5. $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$;
6. $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$;
7. $x \vee x = x$;
8. $x \wedge x = x$;

$$9. \overline{x \vee y} = \bar{x} \wedge \bar{y}; \quad 10. \overline{x \wedge y} = \bar{x} \vee \bar{y};$$

$$11. x \vee 0 = x; \quad 12. x \wedge 0 = 0;$$

$$13. x \wedge 1 = x \quad 14. x \vee 1 = 1;$$

$$15. x \vee (x \wedge y) = x; \quad 16. x \wedge (x \vee y) = x;$$

$$17. x \wedge \bar{x} = 0; \quad 18. x \vee \bar{x} = 1;$$

$$19. \overline{\bar{x}} = x.$$

Доказательство. Чтобы доказать любую из равносильностей 1 - 19, нужно убедиться в равенстве функций, которые задаются формулами, записанными в ее левой и правой частях. Для этого достаточно построить таблицы истинности этих функций. В качестве примера докажем равносильность под номером 10.

Из табл. 2.10 и 2.11 видно, что формулы $\overline{x \wedge y}$ и $\bar{x} \vee \bar{y}$ задают равные функции, следовательно, они равносильны.

Таблица 2.10

| x | y | $x \wedge y$ | $\overline{x \wedge y}$ |
|-----|-----|--------------|-------------------------|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Таблица 2.11

| x | y | \bar{x} | \bar{y} | $\bar{x} \vee \bar{y}$ |
|-----|-----|-----------|-----------|------------------------|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

В справедливости остальных равносильностей рекомендуем убедиться самостоятельно.

■

Равносильности 1 - 19 характеризуют свойства дизъюнкции, конъюнкции и отрицания:

1 и 2 - коммутативность, 3 и 4 - ассоциативность, 5 - дистрибутивность конъюнкции относительно дизъюнкции, 6 - дистрибутивность дизъюнкции относительно конъюнкции, 7 и 8 - идемпотентность. Равносильности 9 и 10 называют законами де Моргана, 11 и 12 - законами нуля, 13 и 14 - законами единицы, 15 и 16 - законами поглощения, 17 и 18 - законами отрицания, 19 - законом двойного отрицания.

Ассоциативность конъюнкции и дизъюнкции позволяет дополнить договоренности об упрощенной записи формул договоренностью опускать скобки в случае многократного последовательного применения только дизъюнкции или только конъюнкции.

Например, формулу $x \vee (y \vee z)$ можно записать в виде $x \vee y \vee z$. В дальнейшем будем также употреблять следующие обозначения:

$$\bigwedge_{i=1}^n x_i = x_1 \wedge x_2 \wedge \dots \wedge x_n \quad \text{и} \quad \bigvee_{i=1}^n x_i = x_1 \vee x_2 \vee \dots \vee x_n.$$

До сих пор мы доказывали равносильность формул следующим образом: по каждой формуле восстанавливали таблицу функции, после чего полученные таблицы сравнивали. Есть и другой подход - использование равносильных преобразований. Он основан на том, что результат вычисления значения функции по формуле не зависит от того, как получены значения переменных, входящих в формулу (брались они произвольно, как значения независимых переменных, или были получены в результате каких-то вычислений). В частности, равносильности 1 - 19 (см. теорему 2.1) остаются справедливыми при подстановке вместо переменных любых формул. Важно лишь соблюдать следующее правило: при подстановке в равносильность вместо некоторой переменной формулы Φ все вхождения этой переменной в исходное равенство должны быть одновременно заменены формулой Φ . Например, используя равносильность $\bar{x} \vee xy = \bar{x} \vee y$ (см. пример 5), можно записать новые равносильности $x \vee \bar{x}y = x \vee y$ (x заменено на \bar{x}) и $y \vee \bar{y}x = y \vee x$ (x заменено на \bar{y} , а y на x), а используя равносильность $\overline{x \vee y} = \bar{x} \wedge \bar{y}$ - равносильность $\overline{x \vee \bar{y}} = x \wedge y$ (x заменено на \bar{x} , а y на \bar{y}).

Таким образом, действует принцип: *если A - подформула формулы Φ , то при замене в формуле Φ любого вхождения A на равносильную ей формулу A_1 формула Φ переходит в равносильную ей формулу Φ_1 .*

Применяя этот принцип, можно находить для заданной формулы равносильные ей формулы, что во многих случаях приводит к упрощению исходной формулы.

Пример 6. Применяя равносильные преобразования, упростить формулу:

$$\text{а) } \bar{y} \vee xyz \vee xy\bar{z}; \quad \text{б) } (x \rightarrow y) \vee (x \rightarrow \bar{y}).$$

$$\blacktriangleleft \text{ а) } \bar{y} \vee xyz \vee xy\bar{z} = \bar{y} \vee (xyz \vee xy\bar{z}) =$$

$$= \bar{y} \vee xy \left(\underbrace{z \vee \bar{z}}_{=1} \right) = \bar{y} \vee xy = \bar{y} \vee x;$$

$$\text{б) } (x \rightarrow y) \vee (x \rightarrow \bar{y}) = (\bar{x} \vee y) \vee (\bar{x} \vee \bar{y}) =$$

$$= \bar{x} \vee y \vee \bar{x} \vee \bar{y} = (\bar{x} \vee \bar{x}) \vee (y \vee \bar{y}) = \bar{x} \vee 1 = 1. \blacktriangleright$$

Упражнение 2.6. Применяя равносильные преобразования, упростить формулу:

$$\text{а) } (x \rightarrow \bar{y}) \vee \overline{x \vee y}; \quad \text{б) } \overline{x \vee y} (x \vee \overline{xy}).$$

Теоретические обоснования

Есть булевы функции, которые не меняют своих значений при изменении значений некоторых своих переменных. Например, на значение функции f_3 (см. табл. 2.4) не влияет значение переменной y , поскольку $f_3(0,0) = f_3(0,1)$, $f_3(1,0) = f_3(1,1)$, а на значение функции f_5 не влияет значение переменной x , поскольку $f_5(0,0) = f_5(1,0)$, $f_5(0,1) = f_5(1,1)$. Переменные, значения которых не влияют на значения функции, называют фиктивными. Определим это понятие строго.

Определение. Переменная x_i называется *фиктивной* переменной функции $f(x_1, \dots, x_n)$, если на всех наборах $\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n$ значений переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ выполняются равенства

$$f(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) = f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n).$$

В противном случае переменная x_i называется *существенной*, т.е. переменная x_i называется *существенной* переменной функции $f(x_1, \dots, x_n)$, если найдется такой набор $\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n$ значений переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$, что

$$f(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) \neq f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n).$$

Чтобы определения фиктивной и существенной переменных легче было усвоить, адаптируем их к каким-нибудь частным случаям. Например, определим y как

фиктивную переменную функции $f(x, y, z)$: переменная y называется фиктивной переменной функции $f(x, y, z)$, если выполняются четыре равенства: $f(0,0,0) = f(0,1,0)$, $f(0,0,1) = f(0,1,1)$, $f(1,0,0) = f(1,1,0)$, $f(1,0,1) = f(1,1,1)$.

Теперь определим y как существенную переменную функции $f(x, y, z)$: переменная y называется существенной переменной функции $f(x, y, z)$, если хотя бы одно из равенств $f(0,0,0) = f(0,1,0)$, $f(0,0,1) = f(0,1,1)$, $f(1,0,0) = f(1,1,0)$, $f(1,0,1) = f(1,1,1)$ не выполняется.

Фиктивные переменные не влияют на значения функции и с этой точки зрения являются «лишними», поэтому естественно поставить вопрос об их удалении.

Операция удаления (введения) фиктивных переменных. Пусть для функции $f(x_1, x_2, \dots, x_n)$ переменная x_i является фиктивной. Возьмем таблицу истинности функции f . Вычеркнем из нее все строки, в которых $x_i = 1$, а также вычеркнем столбец переменной x_i . Полученная таким образом таблица будет задавать

некоторую функцию $g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$, причем на любом наборе

$\alpha_1, \dots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \dots, \alpha_n$ значений переменных $x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n$ для функций f и g выполнено равенство $f(\alpha_1, \dots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \dots, \alpha_n) = g(\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n)$. О функции g говорят, что она получена из функции f путем удаления фиктивной переменной, а о функции f говорят, что она получена из функции g путем введения фиктивной переменной.

Если одну функцию можно получить из другой путем введения или удаления фиктивных аргументов, то они считаются *равными*.

Пример 7. Найти фиктивные переменные функции и удалить их:

а) $f(x, y) = (0011)$; б) $f(x, y, z) = (11011101)$.

◀ а) Для наглядности рассуждений построим таблицу истинности функции (табл. 2.12).

$x: f(0,0) \neq f(1,0)$, следовательно, x - существенная;

$y: f(0,0) = f(0,1)$ и $f(1,0) = f(1,1)$, следовательно, y - фиктивная.

Чтобы удалить фиктивную переменную y , нужно вычеркнуть из таблицы истинности функции второй столбец, а также третью и пятую строки (они закрашены серым цветом). В результате получим таблицу истинности функции $g(x) = (01)$, равной функции $f(x, y) = (0011)$.

б) Построим таблицу истинности функции (табл. 2.13).

Таблица 2.12

| x | y | f |
|-----|-----|-----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Таблица 2.13 $x: f(0,0,0) = f(1,0,0)$, $f(0,0,1) = f(1,0,1)$,

| x | y | z | f |
|-----|-----|-----|-----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

$f(0,1,0) = f(1,1,0)$, $f(0,1,1) = f(1,1,1)$, следовательно, x - фиктивная;

$y: f(0,0,0) \neq f(0,1,0)$, следовательно, y - существенная;

$z: f(0,0,0) = f(0,0,1)$, $f(0,1,0) \neq f(0,1,1)$, значит, z - существенная.

Чтобы удалить фиктивную переменную x , нужно вычеркнуть из таблицы истинности функции первый столбец и последние четыре строки (они закрашены серым цветом). В результате получим таблицу истинности функции $g(y, z) = (1101)$, равной функции $f(x, y, z) = (11011101)$. ►

Упражнение 2.7. Найти фиктивные переменные функции

и удалить их:

а) $f(x, y, z) = (01011111)$; б) $f(x, y, z) = (00110011)$.

Пример 8. Из функции $g(x, y) = (0100)$ путем введения фиктивной переменной t получить функции $f_1(t, x, y)$, $f_2(x, t, y)$, $f_3(x, y, t)$.

◀ Заготовим шаблон таблицы истинности функции $f_1(t, x, y)$ (табл. 2.14), отведя первый столбец под значения переменной t , второй - под значения x , третий - под значения y , четвертый - под значения самой функции.

Значение функции $f_1(t, x, y)$ на каждой паре наборов вида $(0, a, b)$ и $(1, a, b)$ должны совпадать со значением функции $g(x, y)$ на наборе (a, b) . Значит, в строках,

соответствующих наборам $(0,0,0)$ и $(1,0,0)$, надо проставить значение $g(0,0)=0$, в строках, соответствующих наборам $(0,0,1)$ и $(1,0,1)$, - $g(0,1)=1$, в строках, соответствующих наборам $(0,1,0)$ и $(1,1,0)$, - $g(1,0)=0$ и, наконец, в строках, соответствующих наборам $(0,1,1)$ и $(1,1,1)$, - $g(1,1)=1$.

Заготовим шаблон таблицы истинности функции $f_2(x, t, y)$ (табл. 2.15), отведя первый столбец под значения переменной x , второй - под значения t , третий - под значения y , четвертый - под значения самой функции. Столбец значений функции $f_2(x, t, y)$ заполним, исходя из того, что на каждой паре наборов вида $(a, 0, b)$ и $(a, 1, b)$ значение функции f_2 должны совпадать со значением функции $g(x, y)$ на наборе (a, b) .

Заготовим шаблон таблицы истинности функции $f_3(x, y, t)$ (табл. 2.16), отведя первый столбец под значения переменной x , второй - под значения y , третий - под значения t , четвертый - под значения самой функции. Столбец значений функции $f_3(x, y, t)$ заполним, исходя из того, что на каждой паре наборов вида $(a, b, 0)$ и $(a, b, 1)$ значение функции f_3 должны совпадать со значением функции $g(x, y)$ на наборе (a, b) . ►

Таблица 2.14

| t | x | y | f_1 |
|-----|-----|-----|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Таблица 2.15

| x | t | y | f_2 |
|-----|-----|-----|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Таблица 2.16

| x | y | t | f_3 |
|-----|-----|-----|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Упражнение 2.8. Из функции $g(x, y) = (0010)$ путем введения фиктивной переменной t получить функции $f_1(t, x, y)$, $f_2(x, t, y)$, $f_3(x, y, t)$.

Упражнение 2.9. Найти функцию, зависящую существенным образом от всех своих аргументов и равную функции $f(x, y, z, t) = (1100 \ 0011 \ 1100 \ 0011)$.

Удаление фиктивных переменных упрощает дальнейшую работу с функцией. Однако в некоторых случаях фиктивные переменные полезно не удалять, а вводить. Дело в том, что, выполнив такую операцию неоднократно, можно перейти от данной функции к функции, которая ей равна и при этом зависит от любого наперед заданного числа переменных (большого исходного). Благодаря этому, рассматривая одновременно несколько функций, можно считать их зависящими от одного и того же множества переменных (в качестве такого множества можно взять объединение множеств переменных всех этих функций).

Имея в виду вышесказанное, будем в дальнейшем придерживаться следующих договоренностей.

1. Если число переменных специально не оговаривается, функции рассматриваются с точностью до фиктивных переменных, т.е. предполагается, что с заданием некоторой функции f заданы все равные ей функции, и для обозначения равных функций используется один и тот же функциональный символ.

2. При рассмотрении любой системы функций будем предполагать (если не оговорено противное), что все функции данной системы зависят от одного числа переменных. Такой подход позволит избежать громоздких рассуждений.

Например, при желании вместо системы функций $f_1(x) = (10)$, $f_2(y) = (11)$, $f_3(x, y, z) = (10011101)$ можно рассматривать систему равных им функций $f_1(x, y, z) = (11110000)$, $f_2(x, y, z) = (11111111)$, $f_3(x, y, z) = (10011101)$.

В первой части параграфа мы ввели на описательном уровне понятия формулы над множеством функций и функции, заданной формулой. Теперь формализуем эти понятия.

Пусть B - некоторое множество булевых функций, X - некоторое счетное множество переменных. Используя индукцию, определим *множество формул* над B с переменными из X . Одновременно будем определять числовую характеристику $dep(\Phi)$ формулы Φ , называемую ее *глубиной*, а также множество ее *подформул*.

Определение. Базис индукции. Каждая переменная x_i из X и каждая константа c из B является формулой глубины 0, т.е. $dep(x_i) = dep(c) = 0$. Множество подформул формулы глубины 0 состоит из нее самой.

Индуктивный переход. Пусть $f(x_1, x_2, \dots, x_n)$ - функция из \mathbf{B} , A_1, A_2, \dots, A_n - формулы и $\max\{dep(A_1), dep(A_2), \dots, dep(A_n)\} = k$. Тогда выражение $\Phi = (f(A_1, A_2, \dots, A_n))$ является формулой, ее глубина $dep(\Phi)$ равна $k + 1$, а множество подформул Φ включает саму формулу Φ и все подформулы A_1, A_2, \dots, A_n .

Используя индукцию по глубине формулы, определим *функцию, заданную формулой*.

Определение. Базис индукции. Пусть $dep(\Phi) = 0$. Тогда Φ имеет вид $\Phi = x_i$, где x_i - переменная из X , или $\Phi = c$, где c - константа из \mathbf{B} . В первом случае Φ задает тождественную функцию x_i , во втором - функцию, тождественно равную константе c .

Индуктивный переход. По предположению индукции каждой формуле глубины k сопоставлена некоторая функция. Пусть $\Phi(x_1, x_2, \dots, x_n)$ - произвольная формула глубины $dep(\Phi) = k + 1$. Следовательно, она имеет вид $\Phi = (f(A_1, A_2, \dots, A_n))$, где f - функция из \mathbf{B} , A_1, A_2, \dots, A_n - формулы, для которых $\max\{dep(A_1), dep(A_2), \dots, dep(A_n)\} = k$, и, значит, по предположению индукции этим формулам уже сопоставлены некоторые функции $f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n)$. Тогда формула Φ задает функцию $g(x_1, x_2, \dots, x_n) = (f(f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n)))$, значение которой на каждом наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ находится как значение функции f на наборе $(f_1(\alpha_1, \alpha_2, \dots, \alpha_n), \dots, f_n(\alpha_1, \alpha_2, \dots, \alpha_n))$.

Заметим, что, определяя индуктивно функцию, заданную формулой, мы воспользовались договоренностью рассматривать функции с точностью до фиктивных переменных и считали все функции системы \mathbf{B} зависящими от n переменных.

Зададимся вопросом: можно ли по виду формулы, которой задана функция, определить, какие из переменных этой функции существенны, а какие фиктивны? Очевидно, если функция задана формулой, в которую некоторая переменная не входит, то эта переменная фиктивная. Обратное утверждение в общем случае неверно, т.е. в формуле, задающей функцию, могут присутствовать как существенные, так и фиктивные переменные этой функции. Убедимся в этом на примерах.

Пример 9. Указать существенные и фиктивные переменные функции

$$f(x_1, x_2, x_3) = (x_3 \rightarrow (x_3 \vee x_2)) \rightarrow x_1.$$

◀ Вначале упростим формулу, которой задается функция, используя равносильность $a \rightarrow b = \bar{a} \vee b$:

$$f(x_1, x_2, x_3) = (x_3 \rightarrow (x_3 \vee x_2)) \rightarrow x_1 =$$

$$= (\bar{x}_3 \vee (x_3 \vee x_2)) \rightarrow x_1 = (1 \vee x_2) \rightarrow x_1 = 1 \rightarrow x_1.$$

Таким образом, значение функции определяется исключительно значением переменной x_1 , причем $f(1, 0, 0) = 1 \rightarrow 1 = 1$, $f(0, 0, 0) = 0 \rightarrow 1 = 0$. Значит, x_1 - существенная, x_2, x_3 - фиктивные. ►

Упражнение 2.10. Указать существенные и фиктивные переменные функции

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3.$$

Задачи повышенной сложности

Назовем *весом* булева вектора число его координат, равных 1.

2.1. Сколько булевых векторов длины n имеет вес k ?

Булевы векторы часто называют вершинами булева куба.

Расстоянием

(Хэмминга) между вершинами α и β куба B^n называют число $\rho(\alpha, \beta)$, равное числу

координат, в которых они различаются, т.е. $\rho(\alpha, \beta) = \sum_{i=1}^n |\alpha_i - \beta_i|$. Вершины (векторы) α

и β называют *соседними*, если $\rho(\alpha, \beta) = 1$, и *противоположными*, если $\rho(\alpha, \beta) = n$.

Например, расстояние между векторами $(1, 0, 0, 1, 0)$ и $(0, 0, 0, 1, 1)$ равно 2. Расстояние между векторами $(0, 0, 1, 0)$ и $(1, 0, 1, 0)$ равно 1 и, значит, они соседние. Расстояние между векторами $(1, 0, 0, 1, 0, 0)$ и $(0, 1, 1, 0, 1, 1)$ равно длине этих векторов, и, следовательно, они противоположные.

2.2. Найти число неупорядоченных пар вершин булева куба B^n , удаленных друг от друга на расстояние k ($0 \leq k \leq n$).

2.3. Показать, что для любых противоположных наборов α и β выполняется равенство $v(\alpha) + v(\beta) = 2^n - 1$.

2.4. Подсчитать число функций от n переменных, у которых:

а) одна переменная фиктивная, остальные существенные;

б) k ($1 \leq k \leq n$) переменных фиктивные, остальные - существенные.

2.5. Пусть $B = \{f, g, h\}$ - подмножество булевых функций, причем $f \in P_2(1)$, $g \in P_2(2)$, $h \in P_2(3)$ и ни одна из функций не является тождественной константой, $X = \{x, y, z\}$ - алфавит переменных. Определить, является ли выражение A формулой над множеством B независимо от вида функций f, g, h :

- а) $A = h(x, x, f(y))$; б) $A = g(f(z), 0)$;
 в) $A = h(g(x, y), x, f(z))$; г) $A = g(g(x, f(y)), x)$;
 д) $A = h(f(x), g(y, z))$; е) $A = h(g(x), g(y), 1)$.

2.6. По функциям $f(x_1, x_2)$ и $g(x_3, x_4)$, заданным векторами значений, построить векторы значений функции h :

- а) $f = (0111)$, $g = (0100)$, $h(x_2, x_3, x_4) = f(g(x_4, x_2), x_3)$;
 б) $f = (1110)$, $g = (0001)$,
 $h(x_1, x_2, x_3, x_4) = g(g(x_1, x_4), f(x_3, x_2))$.

2.7. Указать существенные и фиктивные переменные функции

$$f(x_1, \dots, x_n) = (x_1 \vee x_2) \oplus (x_2 \vee x_3) \oplus \dots \oplus (x_{n-1} \vee x_n) \oplus (x_n \vee x_1),$$

где $n \geq 3$.

2.8. Выяснить, на скольких векторах булева куба B^6 обращается в нуль функция:

- а) $f(x_1, x_2, \dots, x_6) = x_1 x_2 x_3 \oplus x_4 x_5 x_6$;
 б) $f(x_1, x_2, \dots, x_6) = x_1 x_2 \vee x_3 x_4 \vee x_5 x_6$.

§ 2.2. Совершенные дизъюнктивные и конъюнктивные нормальные формы

Двойственная функция. Принцип двойственности. Разложение булевой функции по переменным. Совершенная дизъюнктивная нормальная форма (СДНФ). Совершенная конъюнктивная нормальная форма (СКНФ).
Представление булевой функции в виде СДНФ и СКНФ.

Базовые понятия и утверждения

1. Принцип двойственности. Функция, заданная формулой $\bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$, называется двойственной к функции $f(x_1, x_2, \dots, x_n)$.

Функцию, двойственную к f , обозначают f^* , таким образом,

$$f^*(x_1, x_2, \dots, x_n) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n).$$

Пример 1. Используя определение, найти функции, двойственные к дизъюнкции, конъюнкции и функциям одной переменной.

◀ Пусть $f(x_1, x_2) = x_1 \vee x_2$. Тогда

$$f^*(x_1, x_2) = \overline{x_1 \vee x_2} = \bar{x}_1 \wedge \bar{x}_2 = x_1 \wedge x_2.$$

Пусть $f(x_1, x_2) = x_1 \wedge x_2$. Тогда

$$f^*(x_1, x_2) = \overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2 = x_1 \vee x_2.$$

Пусть $f(x) = x$. Тогда $f^*(x) = \overline{(x)} = x$.

Пусть $f(x) = \bar{x}$. Тогда $f^*(x) = \overline{(\bar{x})} = x$.

Пусть $f(x) = 0$. Тогда $f^*(x) = \bar{0} = 1$.

Пусть $f(x) = 1$. Тогда $f^*(x) = \bar{1} = 0$. ▶

Сравним таблицы истинности произвольной функции $f(x_1, x_2)$ и двойственной к ней функции $f^*(x_1, x_2)$ (табл. 2.17).

Таблица 2.17

| x_1 | x_2 | $f(x_1, x_2)$ | $f^*(x_1, x_2)$ |
|-------|-------|---------------|---|
| 0 | 0 | $f(0, 0)$ | $\bar{f}(\bar{0}, \bar{0}) = \bar{f}(1, 1)$ |
| 0 | 1 | $f(0, 1)$ | $\bar{f}(\bar{0}, \bar{1}) = \bar{f}(1, 0)$ |
| 1 | 0 | $f(1, 0)$ | $\bar{f}(\bar{1}, \bar{0}) = \bar{f}(0, 1)$ |
| 1 | 1 | $f(1, 1)$ | $\bar{f}(\bar{1}, \bar{1}) = \bar{f}(0, 0)$ |

Нетрудно заметить, что столбец значений функции f^* можно получить из столбца значений функции f , действуя по следующему алгоритму:

- 1) столбец значений функции f переписать в обратном порядке (т.е. число, стоящее в первой строке, записать в последнюю строку; число, стоящее во второй строке, - в предпоследнюю строку и т.д.);
- 2) в получившемся в результате выполнения п. 1 столбце значений каждое число заменить его отрицанием (0 заменить 1 и 1 заменить 0).

Очевидно, что этот алгоритм можно использовать для построения двойственной функции в случае любого числа аргументов.

Если функция задана вектором значений, то роль столбца значений в алгоритме выполняет вектор значений.

Пример 2. Задать вектором значений функцию, двойственную к данной:

а) $f = (1101)$; б) $f = (1001 \ 0000)$.

◀ а) $f = (1101)$ а (1011) а $f^* = (0100)$;

б) $f = (1001 \ 0000)$ а $(0000 \ 1001)$ а $f^* = (1111 \ 0110)$. ▶

Упражнение 2.11. Задать вектором значений функцию, двойственную к данной:

а) $f = (0001 \ 0110)$; б) $f = (1001 \ 0000 \ 0101 \ 1110)$.

Упражнение 2.12. Найти функции, двойственные к сложению по модулю два, эквивалентности, штриху Шеффера и стрелке Пирса.

На примерах мы имели возможность убедиться, что отношение двойственности между функциями симметрично, т.е. если функция $g = f^*$, то функция $f = g^*$. Это утверждение несложно обосновать теоретически.

Действительно, пусть $g(x_1, x_2, \dots, x_n) = f^*(x_1, x_2, \dots, x_n)$. Тогда

$$\begin{aligned} g^*(x_1, x_2, \dots, x_n) &= (f^*(x_1, x_2, \dots, x_n))^* = (\overline{f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)})^* = \\ &= \overline{\overline{f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)}} = f(x_1, x_2, \dots, x_n). \end{aligned}$$

Пусть функция f задана формулой, и мы хотим построить формулу для двойственной к ней функции f^* . Согласно определению, это можно сделать, заменив в формуле, которой задана f , каждую переменную ее отрицанием и взяв отрицание от самой формулы (именно так мы поступили в примере 1).

Рассмотрим другой способ построения формулы для f^* по формуле, представляющей f . Этот способ основан на **принципе двойственности**: *если формула $\Phi[f_1, f_2, \dots, f_n]$ задает функцию g , то формула, полученная из нее заменой символов функций f_1, f_2, \dots, f_n на символы двойственных к ним функций $f_1^*, f_2^*, \dots, f_n^*$, задает функцию g^* , двойственную к функции g* . Полученную таким образом формулу будем называть *двойственной к $\Phi[f_1, f_2, \dots, f_n]$* и обозначать $\Phi^*[f_1, f_2, \dots, f_n]$.

Доказательство принципа двойственности приведено во второй части параграфа.

Если функция задана формулой над множеством $\{0, 1, x, \bar{x}, x \wedge y, x \vee y\}$, то, используя пример 1, принципу двойственности можно придать более конкретный вид: если в формуле Φ , представляющей функцию f , все конъюнкции заменить на дизъюнкции, дизъюнкции - на конъюнкции, 1 на 0, 0 на 1, то получим формулу Φ^* , представляющую функцию f^* , двойственную к f .

Пример 3. Задать формулой функцию, двойственную к данной:

а) $f = \overline{x \vee z \bar{y}}$; б) $f = (x \vee \bar{y}) yz \vee 1$.

◀ а) $f^* = (\overline{x \vee z \bar{y}})^* = \overline{x \wedge (z \vee \bar{y})}$.

Обратим внимание на следующее обстоятельство. В записи формулы, представляющей f , конъюнкция вследствие соглашения о краткой записи формул в скобки не взята.

Строя двойственную формулу, мы меняем конъюнкцию на дизъюнкцию. При этом следует учитывать, что связка « \vee » имеет меньший приоритет выполнения, чем « \wedge », значит, чтобы двойственная формула сохранила структуру исходной формулы, при переходе от конъюнкции к дизъюнкции нужно взять дизъюнкцию в скобки. В то же время при переходе от дизъюнкции к конъюнкции скобки можно опустить.

$$\text{б) } f^* = ((x \vee \bar{y}) yz \vee 1)^* = (\bar{x}\bar{y} \vee y \vee z) \wedge 0. \blacktriangleright$$

Упражнение 2.13. Задать формулой функцию, двойственную к данной:

$$\text{а) } f = x\bar{y}z \vee \bar{x}yz; \quad \text{б) } f = (x \vee \bar{y} \vee 0)\bar{x}y.$$

Если функции равны, то и двойственные к ним функции равны. В сочетании с принципом двойственности это очевидное обстоятельство позволяет получать новые равносильности, переходя от равенства $\Phi_1 = \Phi_2$ к равенству $\Phi_1^* = \Phi_2^*$. Например, из равносильности 1 (см. теорему 2.1) можно получить равносильность 2, а из равносильности 2 - равносильность 1.

Действительно, от равенства $x \vee y = y \vee x$ перейдем к равенству $(x \vee y)^* = (y \vee x)^*$ и, заменив дизъюнкцию на конъюнкцию, а конъюнкцию на дизъюнкцию, получим равенство $x \wedge y = y \wedge x$.

Примерами других пар, связанных законом двойственности, являются равносильности 3 и 4, 5 и 6, 7 и 8, 9 и 10, 11 и 12, 13 и 14, 15 и 16, 17 и 18.

Пример 4. Задать функцию, двойственную к функции $f(x, y, z) = \bar{x}x \vee (x \downarrow y)$:

а) вектором значений; б) формулой.

◀ а) Начнем с того, что зададим таблично (табл. 2.18) саму функцию

$$f(x, y, z) = \bar{x}x \vee (x \downarrow y).$$

Таблица 2.18

| x | y | z | \bar{x} | $\bar{x}x$ | $x \downarrow y$ | $f = \bar{x}x \vee (x \downarrow y)$ |
|-----|-----|-----|-----------|------------|------------------|--------------------------------------|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Далее: $f = (11001010)$ а $f^* = (10101100)$.

б) Используя принцип двойственности и результат выполнения упр. 2.12, из формулы $f(x, y, z) = \bar{z}x \vee (x \downarrow y)$ получим формулу $f(x, y, z) = (\bar{z} \vee x)(x|y)$. ►

Упражнение 2.14. Задать функцию, двойственную к функции $f = (z \leftrightarrow \bar{x})(x|0)$:

а) вектором значений; б) формулой.

2. Задание функции совершенной дизъюнктивной нормальной формой. Мы умеем переходить от реализации функции формулой к заданию ее таблицей истинности. Не меньший интерес представляет обратный переход - от таблицы истинности к формуле. Этот переход неоднозначен - формул, которыми можно задать любую булеву функцию, бесконечно много. Особый интерес имеет задание функций через дизъюнкцию, конъюнкцию и отрицание.

Введем обозначения: $x^0 = \bar{x}$, $x^1 = x$.

Каждую булеву функцию f от n переменных, за исключением тождественно равной нулю, можно задать формулой

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{(\sigma_1, \dots, \sigma_n) \\ f(\sigma_1, \dots, \sigma_n)=1}} (x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n})$$

(здесь дизъюнкция берется по всевозможным наборам значений переменных x_1, x_2, \dots, x_n , на которых функция f равна 1).

Эту формулу называют *совершенной дизъюнктивной нормальной формой* (сокращенно СДНФ).

Доказательство этого утверждения приведено во второй части параграфа.

Чтобы построить СДНФ функции, нужно действовать следующим образом:

- 1) выбрать в таблице истинности функции все наборы значений переменных, на которых функция равна 1;
- 2) для каждого такого набора записать конъюнкцию, в которую войдут все переменные, причем, если значение переменной на данном наборе равно 0, то она войдет в конъюнкцию со знаком отрицания, а если 1, то без знака отрицания;
- 3) записать дизъюнкцию всех выписанных в п. 2 конъюнкций.

Пример 5. Задать функцию в виде СДНФ:

- а) $x_1 \rightarrow x_2$; б) $x_1 | x_2$;
- в) $x_1 \downarrow x_2$; г) $f(x, y, z) = (11001010)$.

◀ а) - в) Зададим функции таблично (табл. 2.19) и выпишем для них СДНФ.

Таблица 2.19

| x_1 | x_2 | $x_1 \rightarrow x_2$ | $x_1 x_2$ | $x_1 \downarrow x_2$ |
|-------|-------|-----------------------|-------------|----------------------|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

$$x_1 \rightarrow x_2 = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 x_2 ;$$

$$x_1 | x_2 = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 \bar{x}_2 ;$$

$$x_1 \downarrow x_2 = \bar{x}_1 \bar{x}_2 .$$

г) Запишем СДНФ, воспользовавшись таблицей истинности функции

$f(x, y, z) = (11001010)$ (см. табл. 2.18):

$$f = \bar{x} \bar{y} \bar{z} \vee \bar{x} \bar{y} z \vee x \bar{y} \bar{z} \vee x y \bar{z} . \blacktriangleright$$

Упражнение 2.15. Задать функцию $f = (01011011)$ в виде СДНФ.

3. Задание функции совершенной конъюнктивной нормальной формой.

Рассмотрим еще один способ задания функции формулой над множеством, состоящим из дизъюнкции, конъюнкции и отрицания.

Каждую булеву функцию f , не равную тождественно единице, можно задать формулой

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{\substack{(\tau_1, \dots, \tau_n) \\ f(\tau_1, \dots, \tau_n)=0}} (x_1^{\tau_1} \vee \dots \vee x_n^{\tau_n})$$

(здесь конъюнкция берется по всевозможным наборам значений переменных x_1, x_2, \dots, x_n , на которых функция f равна 0).

Эту формулу называют *совершенной конъюнктивной нормальной формой* (сокращенно СКНФ).

Доказательство этого утверждения приведено во второй части параграфа.

Чтобы записать СКНФ функции, нужно действовать следующим образом:

- 1) выбрать в таблице истинности функции все наборы значений переменных, на которых функция равна 0;
- 2) для каждого такого набора записать дизъюнкцию, в которую войдут все переменные, причем, если значение переменной на данном наборе равно 1, то она войдет в дизъюнкцию со знаком отрицания, а если 0, то без знака отрицания;
- 3) записать конъюнкцию всех выписанных в п. 2 дизъюнкций.

Пример 6. Задать функцию в виде СКНФ:

- а) $x_1 \rightarrow x_2$; б) $x_1 | x_2$;
 в) $x_1 \downarrow x_2$; г) $f(x, y, z) = (11001010)$.

◀ а) - г) Опираясь на таблицы истинности функций (см. табл. 2.19 и 2.18), запишем:

- а) $x_1 \rightarrow x_2 = \bar{x}_1 \vee x_2$;
 б) $x_1 | x_2 = \bar{x}_1 \vee \bar{x}_2$;
 в) $x_1 \downarrow x_2 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2)$;
 г) $f = (x \vee \bar{y} \vee z)(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee \bar{z})$. ▶

Упражнение 2.16. Задать функцию $f(x, y, z) = (01011011)$ в виде СКНФ.

Каждая булева функция может быть задана формулой над множеством $B = \{\vee, \wedge, \neg\}$.

Действительно, если $f \equiv 0$, то f можно задать в виде СКНФ. Если $f \neq 0$, то f можно задать в виде СДНФ.

Теоретические обоснования

В первой части параграфа были приведены без обоснования несколько утверждений. Вернемся к их обсуждению.

Теорема 2.2 (принцип двойственности). Если формула $\Phi[f_1, f_2, \dots, f_n]$ задает функцию g , то формула $\Phi^*[f_1, f_2, \dots, f_n]$, полученная из нее заменой символов функций f_1, f_2, \dots, f_n на символы двойственных к ним функций $f_1^*, f_2^*, \dots, f_n^*$, задает функцию g^* , двойственную к функции g .

Доказательство. Доказательство проведем индукцией по структуре формул.

Базис индукции. Пусть $dep(\Phi) = 0$. Тогда Φ имеет вид $\Phi = x_i$, где x_i - переменная из X , или $\Phi = c$, где c - константа из B . В первом случае двойственная формула имеет вид $\Phi^* = (x_i)^*$, во втором - $\Phi^* = (c)^*$. В обоих случаях двойственная формула реализует функцию, которая двойственна к функции, заданной формулой Φ .

Индуктивный переход. Пусть утверждение верно для любой формулы глубины, меньшей либо равной k .

Произвольная формула глубины $k + 1$ в соответствии с индуктивным определением формулы может быть записана в виде $\Phi = (f(A_1, A_2, \dots, A_n))$, где f - функция из B , A_1, A_2, \dots, A_n - формулы такие, что $\max_{1 \leq i \leq n} \{dep(A_i)\} = k$. Причем, согласно определению

функции, заданной формулой, всем формулам A_1, A_2, \dots, A_n сопоставлены соответственно некоторые функции $f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)$, а формуле Φ - функция $g(x_1, \dots, x_n) = f(f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$, значение которой на каждом наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ находится как значение функции f на наборе $(f_1(\alpha_1, \alpha_2, \dots, \alpha_n), \dots, f_n(\alpha_1, \alpha_2, \dots, \alpha_n))$.

Используя определение двойственной функции, зададим формулой функцию, двойственную к g :

$$\begin{aligned} g^*(x_1, \dots, x_n) &= \overline{g}(\overline{x_1}, \dots, \overline{x_n}) = \\ &= \overline{f}(f_1(\overline{x_1}, \dots, \overline{x_n}), \dots, f_n(\overline{x_1}, \dots, \overline{x_n})) = \end{aligned}$$

$$\begin{aligned}
&= \overline{f} \left(\overline{\overline{f_1}}(\overline{x_1}, \dots, \overline{x_n}), \dots, \overline{\overline{f_n}}(\overline{x_1}, \dots, \overline{x_n}) \right) = \\
&= \overline{f} \left(\overline{\overline{f_1}(\overline{x_1}, \dots, \overline{x_n})}, \dots, \overline{\overline{f_n}(\overline{x_1}, \dots, \overline{x_n})} \right) = \\
&= \overline{f} \left(\overline{f_1^*}(x_1, \dots, x_n), \dots, \overline{f_n^*}(x_1, \dots, x_n) \right) = f^* \left(f_1^*(x_1, \dots, x_n), \dots, f_n^*(x_1, \dots, x_n) \right).
\end{aligned}$$

Поскольку глубина каждой из формул A_1, A_2, \dots, A_n не превышает k , то для этих формул выполнено предположение индукции, и, значит, функции $f_1^*, f_2^*, \dots, f_n^*$ реализуются формулами, двойственными к формулам A_1, A_2, \dots, A_n , т.е. $f_i^*(x_1, \dots, x_n) = A_i^*(x_1, \dots, x_n)$ ($i = 1, \dots, n$). Но тогда $g^*(x_1, \dots, x_n) = f^*(A_1^*(x_1, \dots, x_n), \dots, A_n^*(x_1, \dots, x_n))$, и, значит, $g^*(x_1, \dots, x_n) = \Phi^*(x_1, \dots, x_n)$. ■

Для доказательства утверждения о представлении функций в виде СДНФ нам понадобится следующая теорема.

Теорема 2.3 (о разложении функции по переменным). Каждую булеву функцию $f(x_1, x_2, \dots, x_n)$ при любом m ($1 \leq m \leq n$) можно задать формулой

$$\bigvee_{(\sigma_1, \sigma_2, \dots, \sigma_m)} \left(x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \dots \wedge x_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, x_{m+1}, \dots, x_n) \right)$$

(здесь дизъюнкция берется по всевозможным наборам значений переменных x_1, x_2, \dots, x_m).

Доказательство. Напомним, что $x^\sigma = \begin{cases} \overline{x} & \text{при } \sigma = 0, \\ x & \text{при } \sigma = 1. \end{cases}$ Следовательно, $0^0 = \overline{0} = 1$, $1^1 = 1$,

$1^0 = \overline{1} = 0$, $0^1 = 0$, т.е. $x^\sigma = 1$ тогда и только тогда, когда $x = \sigma$.

Возьмем произвольный набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$ значений переменных и найдем на этом наборе значение функции, заданной доказываемой формулой:

$$\bigvee_{(\sigma_1, \sigma_2, \dots, \sigma_m)} \left(\alpha_1^{\sigma_1} \wedge \alpha_2^{\sigma_2} \wedge \dots \wedge \alpha_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, \alpha_{m+1}, \dots, \alpha_n) \right) =$$

$$\begin{aligned}
&= \left(\bigwedge_{i=1}^m \alpha_i^{\alpha_i} \wedge f(\alpha_1, \dots, \alpha_m, \alpha_{m+1}, \dots, \alpha_n) \right) \vee \\
&\vee \left(\bigvee_{\substack{(\sigma_1, \dots, \sigma_m) \\ (\sigma_1, \dots, \sigma_m) \neq (\alpha_1, \dots, \alpha_m)}} \left(\alpha_1^{\sigma_1} \wedge \dots \wedge \alpha_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, \alpha_{m+1}, \dots, \alpha_n) \right) \right) = \\
&= f(\alpha_1, \dots, \alpha_m, \alpha_{m+1}, \dots, \alpha_n).
\end{aligned}$$

Поясним последний переход: если $((\sigma_1, \dots, \sigma_m) \neq (\alpha_1, \dots, \alpha_m))$, то хотя бы при одном i $\alpha_i \neq \sigma_i$ и, значит, $\alpha_i^{\sigma_i} = 0$, следовательно, $A = 0$.

Таким образом, имеем:

$$f(x_1, x_2, \dots, x_n) = \bigvee_{(\sigma_1, \sigma_2, \dots, \sigma_m)} \left(x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \dots \wedge x_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, x_{m+1}, \dots, x_n) \right). \blacksquare$$

В качестве примера приведем разложение функции $f(x_1, x_2, \dots, x_n)$:

а) по переменной x_1 :

$$f(x_1, x_2, \dots, x_n) = (x_1 \wedge f(1, x_2, \dots, x_n)) \vee (\overline{x_1} \wedge f(0, x_2, \dots, x_n));$$

б) по переменным x_1, x_2 :

$$\begin{aligned}
f &= (\overline{x_1} \wedge \overline{x_2} \wedge f(0, 0, \dots, x_n)) \vee (\overline{x_1} \wedge x_2 \wedge f(0, 1, \dots, x_n)) \vee \\
&\vee (x_1 \wedge \overline{x_2} \wedge f(1, 0, \dots, x_n)) \vee (x_1 \wedge x_2 \wedge f(1, 1, \dots, x_n)).
\end{aligned}$$

Теорема 2.4 (о задании функции в виде СДНФ). Каждую булеву функцию f от n переменных, за исключением тождественно равной нулю, можно задать формулой

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{(\sigma_1, \dots, \sigma_n) \\ f(\sigma_1, \dots, \sigma_n) = 1}} \left(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} \right).$$

Доказательство. Запишем разложение функции $f(x_1, x_2, \dots, x_n)$ по всем переменным и преобразуем его:

$$\begin{aligned}
f(x_1, x_2, \dots, x_n) &= \bigvee_{(\sigma_1, \dots, \sigma_n)} \left(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} \wedge f(\sigma_1, \dots, \sigma_n) \right) = \\
&= \left(\bigvee_{\substack{(\sigma_1, \dots, \sigma_n) \\ f(\sigma_1, \dots, \sigma_n) = 1}} \left(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} \wedge \underset{=1}{f(\sigma_1, \dots, \sigma_n)} \right) \right) \vee
\end{aligned}$$

$$\begin{aligned} & \bigvee_{\substack{(\sigma_1, \dots, \sigma_n) \\ f(\sigma_1, \dots, \sigma_n)=0}} \left(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} \wedge \underset{=0}{f(\sigma_1, \dots, \sigma_n)} \right) = \\ & = \bigvee_{\substack{(\sigma_1, \dots, \sigma_n) \\ f(\sigma_1, \dots, \sigma_n)=1}} \left(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} \right). \blacksquare \end{aligned}$$

Теорема 2.5 (о задании функции в виде СКНФ). Каждую булеву функцию f , не равную тождественно единице, можно задать формулой

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{\substack{(\tau_1, \dots, \tau_n) \\ f(\tau_1, \dots, \tau_n)=0}} \left(x_1^{\tau_1} \vee \dots \vee x_n^{\tau_n} \right).$$

Доказательство. Представим функцию f^* , двойственную к f , в виде СДНФ:

$$f^*(x_1, x_2, \dots, x_n) = \bigvee_{\substack{(\sigma_1, \dots, \sigma_n) \\ f^*(\sigma_1, \dots, \sigma_n)=1}} \left(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} \right).$$

По принципу двойственности равенство сохранится, если перейти в левой части к двойственной функции, а в правой - к двойственной формуле. Переход к двойственной формуле означает замену всех конъюнкций дизъюнкциями, и наоборот (при этом выражения $x_i^{\sigma_i}$ остаются неизменными, поскольку они представляют собой либо x_i , либо \bar{x}_i):

$$\begin{aligned} (f^*)^* &= \bigwedge_{\substack{(\sigma_1, \dots, \sigma_n) \\ f^*(\sigma_1, \dots, \sigma_n)=1}} \left(x_1^{\sigma_1} \vee \dots \vee x_n^{\sigma_n} \right) = \bigwedge_{\substack{(\sigma_1, \dots, \sigma_n) \\ \bar{f}(\bar{\sigma}_1, \dots, \bar{\sigma}_n)=1}} \left(x_1^{\sigma_1} \vee \dots \vee x_n^{\sigma_n} \right) = \bigwedge_{\substack{(\sigma_1, \dots, \sigma_n) \\ f(\bar{\sigma}_1, \dots, \bar{\sigma}_n)=0}} \left(x_1^{\sigma_1} \vee \dots \vee x_n^{\sigma_n} \right) = \\ &= \bigwedge_{\substack{\bar{\sigma}_1=\tau_1, \\ \bar{\sigma}_n=\tau_n}} \bigwedge_{\substack{(\tau_1, \dots, \tau_n) \\ f(\tau_1, \dots, \tau_n)=0}} \left(x_1^{\tau_1} \vee \dots \vee x_n^{\tau_n} \right). \end{aligned}$$

Так как $(f^*)^* = f$, формула доказана. \blacksquare

Задачи повышенной сложности

2.9. Пусть функция $f(x_1, x_2, \dots, x_n)$ задана вектором значений $(\alpha_1, \alpha_2, \dots, \alpha_{2^n})$. Показать, что $f^*(x_1, x_2, \dots, x_n)$ задается вектором $(\bar{\alpha}_{2^n}, \dots, \bar{\alpha}_2, \bar{\alpha}_1)$.

2.10. Показать, что если функция $f(x_1, x_2, \dots, x_n)$ существенно зависит от переменной x_i ($i=1, \dots, n$), то двойственная к ней функция $f^*(x_1, x_2, \dots, x_n)$ также существенно зависит от переменной x_i .

2.11. Подсчитать число дизъюнктивных слагаемых, образующих СДНФ функции

$$f(x_1, x_2, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n.$$

2.12. Пусть множества $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_m\}$ не пересекаются и пусть

СДНФ функций $f(x_1, x_2, \dots, x_n)$ и $g(y_1, y_2, \dots, y_m)$ имеют соответственно k и p дизъюнктивных слагаемых. Найти число дизъюнктивных слагаемых в СДНФ функций:

а) $f(x_1, x_2, \dots, x_n) \cdot g(y_1, y_2, \dots, y_m);$

б) $f(x_1, x_2, \dots, x_n) \vee g(y_1, y_2, \dots, y_m);$

в) $f(x_1, x_2, \dots, x_n) \oplus g(y_1, y_2, \dots, y_m).$

§ 2.3. Минимизация дизъюнктивных нормальных форм

Элементарная конъюнкция, ранг элементарной конъюнкции. Дизъюнктивная нормальная форма (ДНФ), сложность ДНФ. Минимальная ДНФ. Импликанта, простая импликанта. Сокращенная ДНФ. Тупиковая ДНФ. Представление булевой функции в виде сокращенной, тупиковой и минимальной ДНФ.

Базовые понятия и утверждения

1. Постановка задачи минимизации ДНФ. Пусть задан алфавит переменных

$$X = \{x_1, x_2, \dots, x_n\}.$$

Определение. Формулу вида $K = x_{i_1}^{\sigma_1} \cdot x_{i_2}^{\sigma_2} \cdot \dots \cdot x_{i_r}^{\sigma_r}$, где для любого i σ_i равно 0 или 1 и все переменные разные ($i_v \neq i_\mu$, если $v \neq \mu$), называют *элементарной конъюнкцией ранга r* над множеством X .

Константу 1 считают элементарной конъюнкцией ранга 0.

Элементарные конъюнкции, отличающиеся порядком следования переменных, задают одну функцию, и мы их различать не будем.

Например, пусть задан алфавит переменных $X = \{x_1, x_2, x_3, x_4\}$. Тогда формула

$x_1 \cdot \bar{x}_2 \cdot x_4$ - элементарная конъюнкция ранга 3, а формула $x_1 \cdot \bar{x}_4$ - элементарная

конъюнкция ранга 2 над множеством X . Формула $x_1 \cdot \bar{x}_1 \cdot x_2 \cdot x_3$, напротив, элементарной конъюнкцией не является, так как в этой формуле дважды упоминается переменная x_1 .

Пример 1. Перечислить все элементарные конъюнкции над множеством $\{x_1, x_2\}$: 1, x_1 , \bar{x}_1 , x_2 , \bar{x}_2 , $x_1 x_2$, $\bar{x}_1 x_2$, $x_1 \bar{x}_2$, $\bar{x}_1 \bar{x}_2$.

Вообще, используя алфавит из n переменных $X = \{x_1, x_2, \dots, x_n\}$, можно составить 3^n элементарных конъюнкций. Действительно, произвольную элементарную конъюнкцию можно составить за n этапов. На первом этапе принять решение относительно переменной x_1 (есть три возможности - включить ее в конъюнкцию саму по себе, включить ее отрицание или не включать ее вовсе), на втором - принять решение относительно переменной x_2 (вновь имеем три возможности) и т.д. Действуя описанным образом, мы составим $\underbrace{3 \cdot 3 \cdot \dots \cdot 3}_n = 3^n$ элементарных конъюнкций (в их число входит и константа 1 - сопоставим ее случаю, когда ни одна из переменных в конъюнкцию не включена).

Пример 2. Перечислить все элементарные конъюнкции ранга 3 над множеством $\{x_1, x_2, x_3, x_4\}$, обращающиеся в 1 на наборе $(0,1,0,0)$.

◀ Произвольная элементарная конъюнкция равна 1 в том и только в том случае, когда все образующие ее множители одновременно равны 1. Значит, переменная x_2 может войти в элементарную конъюнкцию, обращающуюся в 1 на наборе $(0,1,0,0)$, только без отрицания, а остальные переменные - обязательно с отрицанием. Таким образом, перебрав все варианты выбора трех переменных из множества $\{x_1, x_2, x_3, x_4\}$, получим четыре конъюнкции: $\bar{x}_1 x_2 \bar{x}_3$, $\bar{x}_1 x_2 \bar{x}_4$, $\bar{x}_1 \bar{x}_3 \bar{x}_4$, $x_2 \bar{x}_3 \bar{x}_4$. ▶

Упражнение 2.17. Перечислить все элементарные конъюнкции над множеством $\{x_1, x_2, x_3\}$, обращающиеся в единицу на наборе $(1,0,0)$.

Определение. Формулу вида $D = K_1 \vee K_2 \vee \dots \vee K_s$ ($K_i \neq K_j$ при $i \neq j$), где через K_i ($i = 1, 2, \dots, s$) обозначены элементарные конъюнкции над множеством X , называют *дизъюнктивной нормальной формой* над множеством X .

Дизъюнкции, отличающиеся только порядком следования конъюнкций, мы различать не будем.

Сумма рангов конъюнкций, входящих в ДНФ, называется *сложностью* ДНФ.

Например, $D_1 = x_1 \cdot x_2 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1$ - ДНФ сложности 6; $D_2 = x_1 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3$ - ДНФ сложности 4; $D_3 = x_1 \vee \bar{x}_2 \cdot x_3$ - ДНФ сложности 3 над множеством $X = \{x_1, x_2, x_3\}$.

Заметим, что формулы D_1, D_2, D_3 равносильны и, следовательно, реализуют одну и ту же функцию $f(x_1, x_2, x_3)$. Таким образом, одна и та же функция может быть задана несколькими ДНФ, причем эти ДНФ могут иметь различную сложность.

Определение. ДНФ, имеющая по сравнению с другими ДНФ, задающими данную функцию, наименьшую сложность, называется *минимальной ДНФ* данной функции.

Так, для функции $f(x_1, x_2, x_3)$, реализуемой рассмотренными выше ДНФ D_1, D_2, D_3 , дизъюнктивная нормальная форма $D_3 = x_1 \vee \bar{x}_2 \cdot x_3$ является минимальной.

Задача минимизации ДНФ формулируется так: для всякой булевой функции $f(x_1, x_2, \dots, x_n)$ найти представление в виде минимальной ДНФ.

Задачу минимизации ДНФ можно решить методом полного перебора, организовав его следующим образом.

1. Построить все ДНФ над множеством $X = \{x_1, x_2, \dots, x_n\}$.

Выше было показано, что число различных элементарных конъюнкций над алфавитом из n переменных равно 2^n . При построении произвольной ДНФ каждую из этих конъюнкций можно в нее либо включить, либо не включить. Значит, всего можно составить $2^{2^n} - 1$ ДНФ (вычтя из 2^{2^n} единицу, мы учли, что если не включить в строящийся объект ни одну из элементарных конъюнкций, то ДНФ не образуется).

2. Отобрать из построенных ДНФ те, которые задают функцию f .

Заметим, что для всякой булевой функции $f(x_1, x_2, \dots, x_n)$, тождественно отличной от нуля, есть, по крайней мере, одна представляющая ее ДНФ - это СДНФ.

3. Для каждой отобранной ДНФ определить сложность; ДНФ наименьшей сложности и есть искомые минимальные ДНФ функции f .

В силу большого объема операций на практике искать минимальную ДНФ функции полным перебором неудобно. Разработано несколько более экономичных методов построения минимальных ДНФ. В этом параграфе мы познакомимся с одним из них.

2. Понятие о сокращенной и тупиковой ДНФ. Выбранный нами для изучения метод построения минимальной ДНФ оперирует с несколькими видами ДНФ, две из которых, совершенная и минимальная ДНФ, нам уже знакомы. Помимо них нам понадобятся так называемые сокращенная и тупиковая ДНФ. Перед тем, как их рассматривать введем ряд понятий.

Определение. Элементарная конъюнкция называется *импликантой* функции $f(x_1, x_2, \dots, x_n)$, если на любом наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$, на котором эта элементарная конъюнкция равна 1, функция f также обращается в 1.

Заметим, что любая элементарная конъюнкция, входящая в СДНФ функции, является импликантой этой функции.

Определение. Будем называть импликанту функции f *простой*, если элементарная конъюнкция, получающаяся из нее удалением любой из букв, уже не является импликантой f .

Пример 3. Перечислить все простые импликанты функции $f(x, y) = x \rightarrow y$.

◀ Имеем девять элементарных конъюнкций над переменными $\{x, y\}$: $1, x, \bar{x}, y, \bar{y}, xy, \bar{x}y, x\bar{y}, \bar{x}\bar{y}$. Относительно каждой из них выясним, является ли она импликантой, и если это так, то простая ли она.

Для удобства дальнейших рассуждений перечислим значения функции f на всех наборах значений переменных: $f(0,0)=1, f(0,1)=1, f(1,0)=0, f(1,1)=1$.

1 импликантой не является, так как обращается в 1 на всех наборах, в то время как $f(1,0)=0$.

x не является импликантой f , так как на наборе $(1,0)$ она обращается в 1 , в то время как $f(1,0)=0$.

\bar{x} обращается в 1 на наборах $(0,0), (0,1)$, и на каждом из этих наборов f также равна 1 , следовательно, \bar{x} - импликанта f . Эта импликанта простая, так как конъюнкция 1 , полученная из нее путем удаления \bar{x} , импликантой не является.

y обращается в 1 на наборах $(0,1), (1,1)$, и на каждом из этих наборов f также равна 1 , следовательно, y - импликанта f . Эта импликанта простая, так как конъюнкция 1 , полученная из нее путем удаления y , импликантой не является.

\bar{y} обращается в 1 на наборах $(0,0)$ и $(1,0)$, а $f(1,0)=0$, следовательно, \bar{y} не является импликантой f .

xy обращается в 1 на одном наборе $(1,1)$, и $f(1,1)=1$, следовательно, xy - импликанта f . Однако простой эта импликанта не является, так как конъюнкция y , полученная из нее путем удаления буквы x , является импликантой.

$\bar{x}y$ обращается в 1 на одном наборе $(0,1)$ и $f(0,1)=1$, следовательно, $\bar{x}y$ - импликанта f . Однако простой эта импликанта не является, поскольку конъюнкция \bar{x} , полученная из нее путем удаления буквы y , является импликантой.

$x\bar{y}$ обращается в 1 на наборе $(1,0)$, а $f(1,0)=0$, следовательно, $x\bar{y}$ не является импликантой f .

$\bar{x}\bar{y}$ обращается в 1 на наборе $(0,0)$, $f(0,0)=1$, следовательно, $\bar{x}\bar{y}$ импликанта f .

Однако простой эта импликанта не является, так как конъюнкция \bar{x} , полученная из нее путем удаления \bar{y} , является импликантой.

Таким образом, функция f имеет две простые импликанты: \bar{x} и y . ►

Упражнение 2.18. Перечислить все простые импликанты функции $f(x, y) = (1011)$.

Определение. Дизъюнкция всех простых импликант функции называется *сокращенной ДНФ* функции.

Например, для функции $f(x, y) = x \rightarrow y$ (см. пример 3) сокращенная ДНФ имеет вид $\bar{x} \vee y$.

Из определения сокращенной ДНФ непосредственно следует, что сокращенная ДНФ у функции единственна.

Справедливо следующее утверждение: *сокращенная ДНФ функции $f(x_1, x_2, \dots, x_n)$ задает функцию $f(x_1, x_2, \dots, x_n)$* (доказательство приведено во второй части параграфа).

Более того, оказывается, что почти всегда можно реализовать функцию дизъюнкцией лишь некоторых (не всех!) простых импликант. В связи с этим уместно ввести понятие тупиковой ДНФ.

Определение. *Тупиковой ДНФ* функции называется такая реализующая ее дизъюнкция простых импликант, из которой нельзя удалить ни одну простую импликанту так, чтобы полученная после удаления ДНФ все еще задавала функцию.

Имеет место следующий теоретический факт: *минимальная ДНФ функции является тупиковой ДНФ* (доказательство приведено во второй части параграфа).

3. Построение минимальных ДНФ. Рассмотрим алгоритм построения минимальных ДНФ функции, исходя из СДНФ. Обоснование данного алгоритма можно найти в [1].

Алгоритм состоит из трех этапов: сначала получают сокращенную ДНФ, далее строят все тупиковые ДНФ и, наконец, из тупиковых ДНФ выделяют минимальные.

1-й этап: получение сокращенной ДНФ функции. Сокращенную ДНФ можно получить из СДНФ, используя алгоритм, называемый методом Квайна.

В алгоритме используются следующие равносильные преобразования:

а) операция неполного склеивания, состоящая в замене выражения $Kx \vee K\bar{x}$ на $Kx \vee K\bar{x} \vee K$;

б) операция поглощения, состоящая в замене выражения $K_1 K_2 \vee K_1$ на K_1 ;

в) операция удаления дублируемых членов, состоящая в замене выражения $K \vee K$ на K .

0-й шаг. Выписываем СДНФ функции. К каждой паре конъюнкций, образующих СДНФ, применяем, если это окажется возможным, операцию неполного склеивания ($Kx \vee K\bar{x}$ заменяем на $Kx \vee K\bar{x} \vee K$). Затем с помощью операции поглощения ($K \vee Kx^\sigma = K$) удаляем те конъюнкции ранга n , которые возможно удалить таким образом. Убираем дублируемые члены ($K \vee K = K$). Полученную в итоге ДНФ обозначаем D_1 .

k -й шаг. К началу этого шага нами построена ДНФ D_k . К каждой паре конъюнкций ранга $(n - k)$ этой ДНФ применяем операции неполного склеивания и поглощения, после чего убираем дублируемые члены. Полученную в результате этой процедуры ДНФ обозначаем D_{k+1} . Если D_{k+1} совпадает с D_k , то D_k - искомая сокращенная ДНФ. В противном случае, увеличив значение k на единицу, повторяем k -й шаг.

В качестве примера построим методом Квайна сокращенную ДНФ функции $g = (10011011)$:

$$\begin{aligned} g(x_1, x_2, x_3) & \stackrel{\text{СДНФ}}{=} \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 = \\ & = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 \vee \bar{x}_2 \bar{x}_3 \vee x_2 x_3 \vee \\ & \vee x_1 \bar{x}_3 \vee x_1 x_2 = \bar{x}_2 \bar{x}_3 \vee x_2 x_3 \vee x_1 \bar{x}_3 \vee x_1 x_2. \end{aligned}$$

2-й этап: построение тупиковых ДНФ.

Напомним, что сокращенная ДНФ представляет собой дизъюнкцию всех простых импликант функции. Значит, если сокращенная ДНФ функции найдена, то найдена и совокупность всех простых импликант функции.

Имея в распоряжении набор всех простых импликант функции, можно построить ее тупиковую ДНФ. Для этого будем использовать следующий алгоритм.

Пусть f - произвольная булева функция. Назовем *носителем функции* множество булевых векторов, на которых функция f принимает значение 1. Пусть булевы векторы $\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_s$ - элементы носителя функции f , K_1, K_2, \dots, K_m - совокупность простых импликант функции. Составим таблицу, строки которой соответствуют простым импликантам функции, столбцы - элементам носителя, а в ячейках на пересечении строки K_i и столбца \mathcal{K}_j проставлено число σ_{ij} , равное 1, если импликанта K_i на наборе \mathcal{K}_j обращается в единицу, и равное 0 в противном случае (табл. 2.20).

Назовем эту таблицу импликантной.

По импликантной таблице составим

Таблица 2.20

формулу вида $\bigwedge_{j=1}^s \left(\bigvee_{i: \sigma_{ij}=1} K_i \right)$, т.е.

каждому булеву вектору σ_j ($1 \leq j \leq n$)

сопоставим дизъюнкцию тех

импликант, значение которых на σ_j

равно 1, и запишем конъюнкцию всех

выписанных дизъюнкций.

Составленную таким образом

формулу преобразуем, используя

законы дистрибутивности, в ДНФ над алфавитом переменных $\{K_1, K_2, \dots, K_m\}$. Затем

полученное выражение упростим, применив, сколько возможно, операции поглощения

и удаления дублируемых членов. В результате всех этих преобразований получим

дизъюнкцию различных конъюнкций вида $K_{i_1} \wedge K_{i_2} \wedge \dots \wedge K_{i_p}$. По каждой такой

конъюнкции выпишем дизъюнкцию $K_{i_1} \vee K_{i_2} \vee \dots \vee K_{i_p}$. Все полученные таким образом

дизъюнкции представляют собой тупиковые ДНФ функции f .

В качестве примера найдем с помощью приведенного выше алгоритма тупиковые

ДНФ функции $g = (10011011)$. Ранее мы нашли сокращенную ДНФ этой функции,

следовательно, можем выписать совокупность всех простых импликант функции:

$K_1 = \bar{x}_2 \bar{x}_3$, $K_2 = x_2 x_3$, $K_3 = x_1 \bar{x}_3$, $K_4 = x_1 x_2$. Составим импликантную таблицу (табл. 2.21).

Таблица 2.21

| | 000 | 011 | 100 | 110 | 111 |
|-----------------------------|-----|-----|-----|-----|-----|
| $K_1 = \bar{x}_2 \bar{x}_3$ | 1 | 0 | 1 | 0 | 0 |
| $K_2 = x_2 x_3$ | 0 | 1 | 0 | 0 | 1 |
| $K_3 = x_1 \bar{x}_3$ | 0 | 0 | 1 | 1 | 0 |
| $K_4 = x_1 x_2$ | 0 | 0 | 0 | 1 | 1 |

Тогда

$$\bigwedge_{j=1}^5 \left(\bigvee_{i: \alpha_{ij}=1} K_i \right) = K_1 K_2 (K_1 \vee K_3) (K_3 \vee K_4) (K_2 \vee K_4) =$$

$$= K_1 K_2 (K_2 K_3 \vee K_4) K_4 = K_1 K_2 K_3 \vee K_1 K_2 K_4.$$

Имеем две тупиковые ДНФ, каждая из которых задает функцию g :

$$K_1 \vee K_2 \vee K_3 = \bar{x}_2 \bar{x}_3 \vee x_2 x_3 \vee x_1 \bar{x}_3$$

и

$$K_1 \vee K_2 \vee K_4 = \bar{x}_2 \bar{x}_3 \vee x_2 x_3 \vee x_1 x_2.$$

3-й этап: построение минимальных ДНФ. Минимальные ДНФ функции являются тупиковыми ДНФ. Поэтому, чтобы их найти, достаточно вычислить сложность всех тупиковых ДНФ функции и отобрать из них ДНФ наименьшей сложности (таких ДНФ может оказаться несколько). Они и будут минимальными ДНФ.

Таблица 2.22 Например, для функции $g = (10011011)$ обе тупиковые

| x_1 | x_2 | x_3 | f |
|-------|-------|-------|-----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

ДНФ имеют сложность, равную 6, и, следовательно, обе являются минимальными.

Пример 4. Построить минимальные ДНФ функции:

а) $f = (11101010)$;

б) $h = (1000100110001011)$.

◀ а) Зададим функции таблицей (табл. 2.22).

1-й этап. Строим сокращенную ДНФ функции:

$$f(x_1, x_2, x_3) \stackrel{\text{сДНФ}}{=} \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 =$$

$$= \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \vee$$

$$\vee x_2 \bar{x}_3 \vee x_1 \bar{x}_3 = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \vee x_2 \bar{x}_3 \vee x_1 \bar{x}_3 =$$

$$= \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \vee x_2 \bar{x}_3 \vee x_1 \bar{x}_3 \vee \bar{x}_3 \vee \bar{x}_3 = \bar{x}_1 \bar{x}_2 \vee \bar{x}_3.$$

2-й и 3-й этапы для данной функции можно опустить. Действительно, нетрудно убедиться, что при опускании любой буквы в сокращенной ДНФ, получается формула, функцию f не представляющая. Следовательно, сокращенная ДНФ $\bar{x}_1 \bar{x}_2 \vee \bar{x}_3$ функции f является также тупиковой и минимальной.

б) Зададим функции таблицей (табл. 2.23).

Таблица 2.23

1-й этап. Построим сокращенную ДНФ

функции:

$$\begin{aligned}
 h &= \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_1 \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4}_2 \vee \underbrace{\bar{x}_1 x_2 x_3 \bar{x}_4}_3 \vee \\
 &\vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_4 \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 x_4}_5 \vee \underbrace{x_1 x_2 x_3 \bar{x}_4}_6 \vee \\
 &\underbrace{x_1 x_2 x_3 x_4}_7 = \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_1 \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4}_2 \vee \\
 &\vee \underbrace{\bar{x}_1 x_2 x_3 \bar{x}_4}_3 \vee \underbrace{\bar{x}_1 x_2 x_3 x_4}_4 \vee \underbrace{x_1 \bar{x}_2 x_3 \bar{x}_4}_5 \vee \\
 &\vee \underbrace{x_1 \bar{x}_2 x_3 x_4}_6 \vee \underbrace{x_1 x_2 \bar{x}_3 \bar{x}_4}_7 \vee \underbrace{x_1 x_2 \bar{x}_3 x_4}_{1,2} \vee \\
 &\vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{1,4} \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4}_{2,5} \vee \underbrace{\bar{x}_1 x_2 x_3 \bar{x}_4}_{3,7} \vee \\
 &\underbrace{x_1 \bar{x}_2 x_3 x_4}_{4,5} \vee \underbrace{x_1 x_2 \bar{x}_3 \bar{x}_4}_{5,6} \vee \underbrace{x_1 x_2 \bar{x}_3 x_4}_{6,7} = \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_8 \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4}_9 \vee \underbrace{\bar{x}_1 x_2 x_3 \bar{x}_4}_{10} \vee \underbrace{x_1 \bar{x}_2 x_3 x_4}_{11} \vee \\
 &\underbrace{x_1 x_2 \bar{x}_3 \bar{x}_4}_{12} \vee \underbrace{x_1 x_2 \bar{x}_3 x_4}_{13} \vee \underbrace{x_1 x_2 x_3 \bar{x}_4}_{14} = \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_8 \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_1 \bar{x}_2 x_3 \bar{x}_4}_{10} \vee \\
 &\vee \underbrace{x_1 \bar{x}_2 x_3 x_4}_{11} \vee \underbrace{x_1 x_2 \bar{x}_3 \bar{x}_4}_{12} \vee \underbrace{x_1 x_2 \bar{x}_3 x_4}_{13} \vee \underbrace{x_1 x_2 x_3 \bar{x}_4}_{14} \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{8,12} \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4}_{9,10} \vee \\
 &\text{IV} \\
 &= x_2 x_3 x_4 \vee x_1 x_2 \bar{x}_4 \vee x_1 x_2 x_3 \vee \bar{x}_3 \bar{x}_4.
 \end{aligned}$$

| x_1 | x_2 | x_3 | x_4 | f |
|-------|-------|-------|-------|-----|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Переход I: операция неполного склеивания применена к парам конъюнкций 1 и 2, 1 и 4, 2 и 5, 3 и 7, 4 и 5, 5 и 6, 6 и 7.

Переход II: применена операция поглощения - каждая из конъюнкций 4-го ранга поглощена какой-то из конъюнкций 3-го ранга.

Переход III: операция неполного склеивания применена к парам конъюнкций 8 и 12, 9 и 10.

Переход IV: применена операция поглощения и убраны дублирующие члены.

Итак, мы нашли сокращенную ДНФ функции: $h = x_2 x_3 x_4 \vee x_1 x_2 \bar{x}_4 \vee x_1 x_2 x_3 \vee \bar{x}_3 \bar{x}_4$.

2-й этап. Введем обозначения для простых импликант функции: $K_1 = x_2x_3x_4$,
 $K_2 = x_1x_2\bar{x}_4$, $K_3 = x_1x_2x_3$, $K_4 = \bar{x}_3\bar{x}_4$. Составим импликантную таблицу (табл. 2.24).

Таблица 2.24

| | 0000 | 0100 | 0111 | 1000 | 1100 | 1110 | 1111 |
|----------------------------|------|------|------|------|------|------|------|
| $K_1 = x_2x_3x_4$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $K_2 = x_1x_2\bar{x}_4$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| $K_3 = x_1x_2x_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| $K_4 = \bar{x}_3\bar{x}_4$ | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

Тогда

$$\bigwedge_{j=1}^7 \left(\bigvee_{i: \sigma_{ij}=1} K_i \right) =$$

$$= K_4 \wedge K_4 \wedge K_1 \wedge K_4 \wedge (K_2 \wedge K_4) \wedge (K_2 \vee K_3) \wedge (K_1 \vee K_3) =$$

$$= K_4 K_1 (K_2 \wedge K_4) (K_2 \vee K_3) (K_1 \vee K_3) = K_4 K_1 (K_2 K_1 \vee K_3) =$$

$$= K_1 K_2 K_4 \vee K_1 K_3 K_4.$$

Таким образом, имеем две тупиковые ДНФ, реализующие функцию h :

$$K_1 \vee K_2 \vee K_4 = x_2x_3x_4 \vee x_1x_2\bar{x}_4 \vee \bar{x}_3\bar{x}_4$$

и

$$K_1 \vee K_3 \vee K_4 = x_2x_3x_4 \vee x_1x_2x_3 \vee \bar{x}_3\bar{x}_4.$$

3-й этап. Тупиковые ДНФ функции h имеют одинаковую сложность (равную 8), и, следовательно, обе являются минимальными. ►

Упражнение 2.19. Построить минимальные ДНФ функции:

а) $f = (11011001)$;

б) $h = (1111000100000011)$.

Теоретические обоснования

Докажем некоторые утверждения, упомянутые в первой части параграфа.

Теорема 2.6. Сокращенная ДНФ функции $f(x_1, x_2, \dots, x_n)$ задает функцию

$$f(x_1, x_2, \dots, x_n).$$

Доказательство. Пусть на некотором наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ функция равна 1. Тогда найдется входящая в СДНФ функции элементарная конъюнкция, которая на этом наборе также равна 1. Значит, эта элементарная конъюнкция - импликанта функции. Удалением части букв из этой импликанты можно получить простую импликанту, которая также будет обращаться в 1 на наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$. Поскольку всякая простая импликанта входит в сокращенную ДНФ, то и сокращенная ДНФ функции f на этом наборе обратится в 1.

С другой стороны, если сокращенная ДНФ функции обращается на некотором наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ в 1, то хотя бы одна из образующих ее простых импликант на этом наборе также равна 1. А если импликанта на наборе равна 1, то и функция на этом же наборе равна 1, т.е. $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$.

Таким образом, на всяком наборе, на котором функция равна 1, ее сокращенная ДНФ равна 1. И, наоборот, на всяком наборе, на котором сокращенная ДНФ функции равна 1, функция также равна 1. Это означает, что значения сокращенной ДНФ функции f и самой функции f совпадают при любых значениях переменных x_1, x_2, \dots, x_n . ■

Теорема 2.7. Минимальная ДНФ функции является дизъюнкцией простых импликант этой функции.

Доказательство. Пусть D - произвольная минимальная ДНФ функции $f(x_1, x_2, \dots, x_n)$. Поскольку D задает f , то все входящие в D элементарные конъюнкции являются импликантами f .

Надо доказать, что они простые. Будем рассуждать от противного. Предположим, что в D входит импликанта K , не являющаяся простой. Опустив часть букв, выделим из K простую импликанту K' . Обозначим D' дизъюнктивную нормальную форму, полученную из D заменой K на K' .

Докажем, что D' задает функцию f . Запишем D в виде $D = K \vee (\vee K_i)$. Тогда $D' = K' \vee (\vee K_i)$. Пусть на некотором наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ f равна 1. Тогда, поскольку $f = D$, то на этом наборе K или $(\vee K_i)$ равна 1. Если K равна 1, то и K' равна 1, т.е. на наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ K' или $(\vee K_i)$ равна 1, откуда D' равно 1. С другой стороны, если на некотором наборе $(\alpha_1, \alpha_2, \dots, \alpha_n)$ D' равна 1, то хотя бы одна из образующих ее импликант на этом наборе также равна 1, а значит, и $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$.

Таким образом, D' содержит меньшее число букв, чем D , и при этом реализует функцию f , что противоречит предположению о минимальности исходной ДНФ. ■

Теорема 2.8. *Минимальная ДНФ функции является тупиковой ДНФ.*

Доказательство. Будем рассуждать от противного. Предположим, что найдется минимальная ДНФ, которая не является тупиковой. Тогда из этой ДНФ можно удалить хотя бы одну простую импликанту так, чтобы образовавшаяся при этом новая ДНФ по-прежнему реализовала функцию. Но эта новая ДНФ будет иметь меньшую сложность, чем исходная минимальная ДНФ. А это противоречит тому, что сложность минимальной ДНФ наименьшая из возможных. ■

Задачи повышенной сложности

2.13. Найти число элементарных конъюнкций ранга r над множеством $\{x_1, x_2, \dots, x_n\}$, обращающихся в единицу на наборе $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$.

2.14. Найти число всех элементарных конъюнкций над множеством $\{x_1, x_2, \dots, x_n\}$, обращающихся в единицу на наборе $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$.

2.15. Подсчитать число функций от n переменных, для которых данная элементарная конъюнкция ранга r является импликантой.

2.16. Пусть множества $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_m\}$ не пересекаются, K - простая импликанта функции $f(x_1, x_2, \dots, x_n)$, а L - простая импликанта функции $g(y_1, y_2, \dots, y_m)$. Показать, что $K \wedge L$ - простая импликанта функции $f \wedge g$.

2.17. Назовем число элементарных конъюнкций, образующих ДНФ, ее *длиной*. Найти длину сокращенной ДНФ функции $x_1 \oplus x_2 \oplus \dots \oplus x_n$.

2.18. Показать, что x_i является существенной переменной тогда и только тогда, когда эта переменная явно входит в сокращенную ДНФ функции.

§ 2.4. Классы Поста и замыкание

Функции, сохраняющие 0. Функции, сохраняющие 1. Самодвойственные функции. Монотонные функции. Полином Жегалкина, представление булевой функции полиномом Жегалкина. Линейные функции. Классы Поста. Замыкание системы булевых функций.

Базовые понятия и утверждения

1. Функции, сохраняющие 0; функции, сохраняющие 1. Говорят, что булева функция *сохраняет* 0, если $f(0,0,\dots,0)=0$.

Говорят, что булева функция *сохраняет* 1, если $f(1,1,\dots,1)=1$.

Обозначим через T_0 (T_1) множество всех булевых функций, сохраняющих 0 (1), а через $T_0(n)$ ($T_1(n)$) множество функций от n переменных, сохраняющих 0 (1).

Пример 1. Определить, принадлежит ли множествам T_0 и T_1 функция

$$f(x, y, z) = (x \rightarrow y) \cdot z.$$

◀ $f(0,0,0) = (0 \rightarrow 0) \cdot 0 = 0$, следовательно,

$$f(x, y, z) = (x \rightarrow y) \cdot z \in T_0;$$

$f(1,1,1) = (1 \rightarrow 1) \cdot 1 = 1$, следовательно,

$$f(x, y, z) = (x \rightarrow y) \cdot z \in T_1. \blacktriangleright$$

Упражнение 2.20. Определить, принадлежит ли множествам T_0 и T_1 функция

$$f(x, y, z) = \bar{z}x \vee (x \downarrow y).$$

Несложно убедиться, что функции $0, x, x \wedge y, x \vee y, x \oplus y$ сохраняют 0, а функции $1, \bar{x}, x|y, x \downarrow y, x \rightarrow y, x \leftrightarrow y$ не сохраняют 0, функции $1, x, x \wedge y, x \vee y, x \rightarrow y, x \leftrightarrow y$ сохраняют 1, а функции $0, \bar{x}, x|y, x \downarrow y, x \oplus y$ не сохраняют 1.

Очевидно, что определения функций, сохраняющих 0 и 1, можно переформулировать следующим образом: функция сохраняет 0, если ее вектор значений начинается нулем, и функция сохраняет 1, если ее вектор значений оканчивается единицей.

Пример 2. Определить число булевых функций от n переменных сохраняющих 0.

◀ Вектор значений функции от n переменных, сохраняющей 0, имеет 2^n координаты, первая из которых равна 0, т.е. $\begin{pmatrix} 0, \alpha_1, \alpha_2, \dots, \alpha_{2^n-1} \end{pmatrix}$. Следовательно, функций от n переменных,

сохраняющих 0, столько же, сколько булевых векторов длины $2^n - 1$, т.е. 2^{2^n-1} . ▶

Упражнение 2.21. Определить число булевых функций от n переменных:

а) сохраняющих 1;

б) сохраняющих как 0, так и 1, т.е. содержащихся в пересечении множеств $T_0(n)$ и $T_1(n)$;

в) содержащихся в объединении множеств $T_0(n)$ и $T_1(n)$.

2. Самодвойственные функции. Булева функция называется *самодвойственной*, если она равна двойственной к ней, т.е. на любом наборе $(\alpha_1, \dots, \alpha_n)$ значений переменных (x_1, \dots, x_n) выполняется равенство $f(\alpha_1, \dots, \alpha_n) = \bar{f}(\bar{\alpha}_1, \dots, \bar{\alpha}_n)$.

Обозначим через S множество всех самодвойственных функций, а через $S(n)$ множество самодвойственных функций от n переменных.

Пример 3. а) $(0)^* = 1$, $(1)^* = 0$, $(x)^* = x$, $(\bar{x})^* = \bar{x}$, значит, среди функций одной переменной самодвойственными являются только тождественная функция и отрицание.

б) Среди элементарных функций двух переменных $x \wedge y, x \vee y, x \oplus y$,

$x|y, x \downarrow y, x \rightarrow y, x \leftrightarrow y$ нет самодвойственных. Чтобы убедиться в этом, достаточно сравнить векторы значений этих функций с векторами значений функций, двойственных к ним:

$$(x \wedge y)^* = (0001)^* = (0111), (x \vee y)^* = (0111)^* = (0001),$$

$$(x \oplus y)^* = (0110)^* = (1001), (x|y)^* = (1110)^* = (1000),$$

$$(x \downarrow y)^* = (1000)^* = (1110), (x \rightarrow y)^* = (1101)^* = (0100),$$

$$(x \leftrightarrow y)^* = (1001)^* = (0110).$$

Упражнение 2.22. Выяснить, является ли самодвойственной функция

$$f(x, y, z) = \bar{z}x \vee (x \downarrow y).$$

3. Монотонные функции. Если для любого i $\alpha_i \leq \beta_i$ ($i = 1, 2, \dots, n$), то говорят, что вектор $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ *предшествует* вектору $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, и пишут $\alpha \preceq \beta$.

Заметим, что если номер вектора α меньше номера вектора β (и, значит, в таблице истинности α стоит выше β), то это еще не значит, что α предшествует β . Чтобы выяснить, предшествует ли один вектор другому, нужно, согласно определению, сравнить их координаты (первую с первой, вторую со второй и т.д.).

Пример 4. а) Вектору $(1, 0)$ предшествует вектор $(0, 0)$ (но не предшествует также меньший его по номеру вектор $(0, 1)$).

б) Вектору $(1, 0, 1)$ предшествуют векторы $(0, 0, 0)$, $(0, 0, 1)$, $(1, 0, 0)$, $(1, 0, 1)$.

Если произвольно взять два вектора одной длины, то может оказаться, что ни один из этих векторов не предшествует другому. Например, вектор $(0, 1, 1)$ не предшествует вектору $(1, 1, 0)$ и вектор $(1, 1, 0)$ не предшествует вектору $(0, 1, 1)$.

Если имеет место хотя бы одно из соотношений $\alpha \preceq \beta$ или $\beta \preceq \alpha$, то α и β называют *сравнимыми*.

Пример 5. Сколько булевых векторов предшествует вектору $(1, 0, 1, 1, 0, 1, 1)$?

◀ Векторы, предшествующие вектору $(1, 0, 1, 1, 0, 1, 1)$, имеют вид $(\alpha_1, 0, \alpha_2, \alpha_3, 0, \alpha_4, \alpha_5)$, где α_i выбираются из множества $\{0, 1\}$. Для подсчета будем использовать правило произведения. Вначале выбираем значение α_1 (это можно сделать двумя способами - взять 0 или 1), затем выбираем значение α_2 (0 или 1) и последним выбираем значение α_5 (0 или 1). Следовательно, общее число векторов, предшествующих вектору $(1, 0, 1, 1, 0, 1, 1)$, равно $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 32$. ▶

Упражнение 2.23. Скольким булевым векторам предшествует вектор $(1, 0, 1, 0, 0, 1, 0, 1)$?

Определение. Булева функция f называется *монотонной*, если для любых наборов α и β значений переменных, таких что $\alpha \preceq \beta$, выполняется неравенство $f(\alpha) \leq f(\beta)$.

Обозначим через \mathbf{M} множество всех монотонных функций, а через $\mathbf{M}(n)$ множество монотонных функций от n переменных.

Чтобы убедиться в немонотонности функции, достаточно найти два булевых вектора α и β , такие что $\alpha \neq \beta$, а $f(\alpha) > f(\beta)$. Для вывода о монотонности нужно сравнить значения функции на всех парах сравнимых векторов.

Пример 6. а) Функции $0, 1, x$ монотонные, а функция \bar{x} немонотонная, что легко проверить, поскольку для функций одной переменной условие монотонности сводится к выполнению неравенства $f(0) \leq f(1)$.

б) Функции $xy, x \vee y$ монотонные, а $x \leftrightarrow y, x|y, x \downarrow y, x \rightarrow y, x \oplus y$ немонотонные.

Действительно, для функций двух переменных условие монотонности сводится к выполнению неравенств $f(0,0) \leq f(0,1), f(0,0) \leq f(1,0), f(0,0) \leq f(1,1), f(0,1) \leq f(1,1), f(1,0) \leq f(1,1)$ (обратите внимание, что значения $f(0,1)$ и $f(1,0)$ не сравниваются, так как в паре векторов $(0,1), (1,0)$ ни один не предшествует другому). Для функций $xy, x \vee y$ все эти неравенства выполняются, а для остальных элементарных функций двух переменных - нет.

Действительно,

для $f(x, y) = x \leftrightarrow y = (1001)$ имеем $f(0,0) > f(0,1)$;

для $f(x, y) = x|y = (1110)$ имеем $f(0,0) > f(1,1)$;

для $f(x, y) = x \downarrow y = (1000)$ имеем $f(0,0) > f(0,1)$;

для $f(x, y) = x \rightarrow y = (1101)$ имеем $f(0,0) > f(1,0)$;

для $f(x, y) = x \oplus y = (0110)$ имеем $f(0,1) > f(1,1)$.

Пример 7. Выяснить, монотонна или нет функция $f = (01010010)$.

◀ Выпишем таблицу истинности функции f (табл. 2.25).

Таблица 2.25

| x_1 | x_2 | x_3 | f |
|-------|-------|-------|-----|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Будем последовательно перебирать пары булевых векторов, отбирать из них сравнимые и сопоставлять значения функции f на векторах этих пар.

Перебор можно вести, придерживаясь такой схемы: сначала перебираем пары с вектором $(0,0,0)$, затем пары с вектором $(0,0,1)$ (вектор $(0,0,0)$ уже не рассматриваем), затем пары с вектором $(0,1,0)$ (векторы $(0,0,0)$ и $(0,0,1)$

уже не рассматриваем) и т.д.

Вектор $(0,0,0)$ предшествует всем прочим булевым векторам длины 3, при этом значение функции на этом векторе, будучи равным нулю, меньше либо равно значениям функции на всех других векторах. Этот факт не позволяет сделать никаких выводов относительно монотонности функции f , поэтому переходим к рассмотрению пар с вектором $(0,0,1)$. Вектор $(0,0,1)$ предшествует векторам $(0,1,1)$, $(1,0,1)$, $(1,1,1)$, при этом $f(0,0,1) \leq f(0,1,1)$, однако уже $f(0,0,1) > f(1,0,1)$. Следовательно, условие монотонности нарушено, и, значит, функция немонотонна. ►

Упражнение 2.24. Выяснить, монотонна или нет функция $h = (01010111)$.

4. Линейные функции. Рассмотрим формулы с двумя бинарными операциями - конъюнкцией (умножением) и сложением по модулю два (вместо $x \oplus y$ часто пишут $x + y$ и говорят просто о сложении x и y , не указывая, что сложение производится по модулю два).

В дальнейшем будем использовать ряд равносильностей, справедливость которых нетрудно проверить с помощью таблиц истинности:

- 1) $x \oplus y = y \oplus x$;
- 2) $(x \oplus y) \oplus z = x \oplus (y \oplus z)$;
- 3) $(x \oplus y)z = xz \oplus yz$;
- 4) $x \oplus 1 = \bar{x}$;
- 5) $x \oplus 0 = x$;
- 6) $x \oplus x = 0$.

Таким образом, для сложения (по модулю два) и умножения (конъюнкции) имеют место все основные арифметические законы: коммутативность, ассоциативность,

дистрибутивность умножения относительно сложения. Это позволяет использовать в записи арифметических выражений все обычные упрощения, в частности, привычным образом перемножать скобки:

$$7) (x \oplus y)(z \oplus t) = xz \oplus yz \oplus xt \oplus yt.$$

Ассоциативность сложения позволяет упростить запись формул, если договориться опускать скобки в случае многократного последовательного применения этой операции. Тогда формулу $x \oplus (y \oplus z)$ можно записывать в виде $x \oplus y \oplus z$. Для краткого обозначения суммы нескольких слагаемых используют знак \sum , например, пишут

$$x_1 \oplus x_2 \oplus \dots \oplus x_n = \sum_{i=1}^n x_i.$$

Полиномом Жегалкина над множеством $X = \{x_1, x_2, \dots, x_n\}$ называется всякая сумма (по модулю два) различных элементарных конъюнкций над X , не содержащих отрицаний переменных.

Заметим, что элементарные конъюнкции, не содержащие отрицаний переменных, называют *монотонными*.

Наибольший из рангов элементарных конъюнкций, входящих в полином, называют *степенью полинома*.

Согласно определению, произвольный полином Жегалкина имеет вид $\sum x_{i_1} x_{i_2} \dots x_{i_k}$ или $1 \oplus \sum x_{i_1} x_{i_2} \dots x_{i_k}$ (здесь суммирование ведется по некоторому множеству наборов (i_1, i_2, \dots, i_k) , в каждом из которых все i_j различны). В число полиномов Жегалкина также входит константа 1 (напомним, что 1 считается элементарной конъюнкцией ранга нуль). Кроме того, в число полиномов включают константу 0.

Например, $1 \oplus x_2 \oplus x_1 x_3$ и $x_1 x_2 \oplus x_1 x_3 \oplus x_1 x_2 x_3$ - полиномы Жегалкина 2-й и 3-й степени соответственно над множеством $X = \{x_1, x_2, x_3\}$.

Зададимся целью реализовать полиномом Жегалкина произвольную булеву функцию. Для начала найдем полином Жегалкина для дизъюнкции:

$$\begin{aligned} f = x_1 \vee x_2 &= \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2} + x_1 \cdot x_2 \stackrel{(4)}{=} x_1 \cdot x_2 \oplus 1 \stackrel{(7)}{=} (x_1 \oplus 1) \cdot (x_2 \oplus 1) \oplus 1 = \\ &= x_1 \cdot x_2 \oplus x_1 \oplus x_2 \oplus 1 \oplus 1 \stackrel{(5),(6)}{=} x_1 \cdot x_2 \oplus x_1 \oplus x_2. \end{aligned}$$

Теперь в нашем распоряжении есть очень полезная формула:

$$8) x_1 \vee x_2 = x_1 \cdot x_2 \oplus x_1 \oplus x_2,$$

Формула (8) в сочетании с тождествами (1) - (7) позволяет перейти от произвольной ДНФ к полиному Жегалкина. Для этого нужно, применив формулы (8) и (4), избавиться от дизъюнкции и отрицания, после чего перемножить скобки, воспользовавшись тождествами (3) и (7), и затем упростить полученное выражение с помощью равенств (1) - (6). Таким образом, если некоторая функция задана в виде ДНФ, то после всех описанных преобразований она окажется заданной полиномом Жегалкина. Поскольку каждую булеву функцию, не равную тождественно нулю, можно представить в виде дизъюнктивной нормальной формы (например, СДНФ), то приведенные выше рассуждения доказывают, что *любую булеву функцию можно задать полиномом Жегалкина.*

Заметим, что если для получения полинома Жегалкина некоторой функции в качестве исходной формулы использовать СДНФ, то на первом шаге преобразований знак \vee дизъюнкции можно просто заменить знаком \oplus суммы. Дело в том, что для любой пары входящих в СДНФ элементарных конъюнкций K_1 и K_2 найдется хотя бы одна переменная, которая в одну из конъюнкций входит без отрицания, а в другую - с отрицанием, и, значит, произведение этих конъюнкций равно 0, а если $K_1 K_2 = 0$, то $K_1 \vee K_2 = K_1 \oplus K_2$.

Чтобы представить функцию в виде полинома Жегалкина, вовсе не обязательно в качестве исходной формулы брать ДНФ. Можно воспользоваться любым представлением функции через дизъюнкцию, конъюнкцию и отрицание, например, использовать СКНФ функции.

Пример 8. Используя метод равносильных преобразований, найти полином Жегалкина, реализующий функцию:

$$а) f(x_1, x_2) = x_1 \downarrow x_2; \quad б) f(x_1, x_2) = x_1 \leftrightarrow x_2;$$

$$в) f(x_1, x_2) = x_1 | x_2; \quad г) f(x_1, x_2) = x_1 \rightarrow x_2.$$

$$\blacktriangleleft а) x_1 \downarrow x_2 \stackrel{\text{СДНФ}}{=} \overline{x_1 \cdot x_2} = (x_1 \oplus 1)(x_2 \oplus 1) = x_1 x_2 \oplus x_1 \oplus x_2 \oplus 1;$$

$$б) x_1 \leftrightarrow x_2 \stackrel{\text{СДНФ}}{=} \overline{x_1} \cdot \overline{x_2} \vee x_1 x_2 = \overline{x_1} \cdot \overline{x_2} \oplus x_1 x_2 = \\ = x_1 x_2 \oplus x_1 \oplus x_2 \oplus 1 \oplus x_1 x_2 = x_1 \oplus x_2 \oplus 1;$$

$$в) x_1 | x_2 \stackrel{\text{СКНФ}}{=} \overline{x_1} \vee \overline{x_2} = \overline{x_1 x_2} = x_1 x_2 \oplus 1;$$

$$\begin{aligned} \text{г) } x_1 \rightarrow x_2 &= \overline{x_1 \vee x_2} = \overline{x_1} \oplus x_2 \oplus \overline{x_1} \cdot x_2 = \\ &= x_1 \oplus 1 \oplus x_2 \oplus (x_1 \oplus 1) x_2 = 1 \oplus x_1 \oplus x_1 x_2. \blacktriangleright \end{aligned}$$

Упражнение 2.25. Используя метод равносильных преобразований, найти полином Жегалкина, реализующий функцию $f = (00010010)$.

Теорема 2.9. *Для любой булевой функции существует задающий ее полином Жегалкина. Он единственен с точностью до перестановок слагаемых.*

Доказательство. Возможность представления полиномом Жегалкина любой булевой функции нами уже доказана. Докажем, что такое представление единственно.

Сначала выясним, сколько всего можно составить полиномов Жегалкина от n переменных. Поставим в соответствие каждой монотонной элементарной конъюнкции $x_{i_1} x_{i_2} \dots x_{i_k}$ булев вектор длины n , в котором координаты с номерами i_1, i_2, \dots, i_k равны 1, а остальные координаты равны 0. Константе 1 поставим в соответствие нулевой набор. Полученное соответствие взаимно-однозначно, значит, число элементарных конъюнкций, которые могут быть включены в произвольный полином Жегалкина, равно числу булевых векторов длины n , т.е. 2^n .

Теперь упорядочим конъюнкции по возрастанию номеров сопоставленных им булевых векторов: $K_0, K_1, \dots, K_{2^n-1}$. Очевидно, что всякий полином Жегалкина единственным образом можно записать в виде суммы: $a_0 K_0 \oplus a_1 K_1 \oplus \dots \oplus a_{2^n-1} K_{2^n-1}$, где каждый коэффициент a_i равен 0 или 1. Суммы такого вида будем называть *каноническими полиномами Жегалкина от n переменных* (примеры записи полиномов Жегалкина в каноническом виде приведены сразу после доказательства теоремы).

Каждый канонический полином Жегалкина от n переменных однозначно определяется набором своих коэффициентов $a_0, a_1, \dots, a_{2^n-1}$, а каждый такой набор можно рассматривать как булев вектор длины 2^n . Следовательно, канонических полиномов Жегалкина от n переменных столько же, сколько булевых векторов длины 2^n , т.е. 2^{2^n} . Таким образом, число полиномов Жегалкина от n переменных равно числу функций от n переменных.

Напомним, что наша цель - доказать, что каждая функция представляется единственным полиномом. Предположим обратное, что найдется функция, заданная двумя разными полиномами. Имеем: каждому полиному соответствует ровно одна

функция, им реализуемая, причем в силу предположения для каких-то двух полиномов эта функция общая. Но тогда число функций, реализуемых полиномами, будет меньше числа всех полиномов, а значит, и числа всех булевых функций. Следовательно, найдется функция, не заданная ни одним полиномом, что невозможно в силу справедливости первого утверждения теоремы. ■

Вернемся к вопросу о записи произвольного полинома Жегалкина от n переменных в каноническом виде. Начнем с того, что запишем в общем виде канонический полином Жегалкина от двух переменных. В него войдут все монотонные элементарные конъюнкции над множеством переменных $\{x_1, x_2\}$, записанные в сумму в порядке возрастания номеров соответствующих им булевых векторов. Напомним, что в каждой такой конъюнкции перемножаются только те переменные, значения которых в соответствующем булевом векторе равны 1. Первой идет элементарная конъюнкция $K_0 = 1$, соответствующая вектору (0,0), за ней $K_1 = x_2$, соответствующая вектору (0,1), далее $K_2 = x_1$, соответствующая вектору (1,0), и, наконец, $K_3 = x_1 x_2$, соответствующая вектору (1,1). Итак, канонический полином Жегалкина от двух переменных имеет вид $a_0 \oplus a_1 x_2 \oplus a_2 x_1 \oplus a_3 x_1 x_2$.

Рассуждая аналогично, выпишем в общем виде канонический полином Жегалкина от трех переменных:

$$a_0 \oplus a_1 x_3 \oplus a_2 x_2 \oplus a_3 x_2 x_3 \oplus a_4 x_1 \oplus a_5 x_1 x_3 \oplus a_6 x_1 x_2 \oplus a_7 x_1 x_2 x_3.$$

Имея перед глазами общие виды канонических полиномов, не составит труда переписать в каноническом виде любой конкретный полином. Например,

$$1 \oplus x_1 \oplus x_1 x_2 = 1 \oplus 0 \cdot x_2 \oplus 1 \cdot x_1 \oplus 1 \cdot x_1 x_2;$$

$$x_1 \oplus x_1 x_3 \oplus x_2 x_3 \oplus x_1 x_2 x_3 =$$

$$= 0 \oplus 0 \cdot x_3 \oplus 0 \cdot x_2 \oplus 1 \cdot x_2 x_3 \oplus 1 \cdot x_1 \oplus 1 \cdot x_1 x_3 \oplus 0 \cdot x_1 x_2 \oplus 1 \cdot x_1 x_2 x_3.$$

Чтобы задать функцию полиномом Жегалкина, применяют **метод равносильных преобразований** и **метод неопределенных коэффициентов**. С первым из них мы уже познакомились, доказывая возможность представления произвольной булевой функции полиномом Жегалкина.

Рассмотрим **метод неопределенных коэффициентов**. Пусть функция f зависит от n переменных. Запишем в общем виде канонический полином Жегалкина P от n

переменных и для каждого набора $(\alpha_1, \alpha_2, \dots, \alpha_n)$ значений переменных составим уравнение

$$f(\alpha_1, \alpha_2, \dots, \alpha_n) = P(\alpha_1, \alpha_2, \dots, \alpha_n).$$

В результате получим систему из 2^n уравнений, которая однозначно определяет коэффициенты полинома.

Пример 9. Используя метод неопределенных коэффициентов, построить полином Жегалкина функции:

а) $f(x_1, x_2) = x_1 | x_2$;

б) $f(x_1, x_2, x_3) = (10110111)$.

◀ а) Выпишем таблицу истинности функции f (табл. 2.26).

Канонический полином Жегалкина от двух переменных имеет вид:

$$P(x_1, x_2) = a_0 \oplus a_1 x_2 \oplus a_2 x_1 \oplus a_3 x_1 x_2.$$

Таблица 2.26 Поэтому должны выполняться равенства:

| x_1 | x_2 | f |
|-------|-------|-----|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$$f(0,0) = 1 = P(0,0) = a_0 \oplus a_1 \cdot 0 \oplus a_2 \cdot 0 \oplus a_3 \cdot 0 \cdot 0;$$

$$f(0,1) = 1 = P(0,1) = a_0 \oplus a_1 \cdot 1 \oplus a_2 \cdot 0 \oplus a_3 \cdot 0 \cdot 1;$$

$$f(1,0) = 1 = P(1,0) = a_0 \oplus a_1 \cdot 0 \oplus a_2 \cdot 1 \oplus a_3 \cdot 1 \cdot 0;$$

$$f(1,1) = 0 = P(1,1) = a_0 \oplus a_1 \cdot 1 \oplus a_2 \cdot 1 \oplus a_3 \cdot 1 \cdot 1.$$

Откуда последовательно находим $a_0 = 1$, $a_1 = 0$, $a_2 = 0$, $a_3 = 1$.

Таким образом, $f = 1 \oplus x_1 x_2$.

Таблица 2.27

| x_1 | x_2 | x_3 | f |
|-------|-------|-------|-----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

б) Выпишем таблицу истинности функции f (табл. 2.27).

Запишем в общем виде канонический полином Жегалкина от трех переменных:

$$P(x_1, x_2, x_3) = a_0 \oplus a_1 x_3 \oplus a_2 x_2 \oplus$$

$$\oplus a_3 x_2 x_3 \oplus a_4 x_1 \oplus a_5 x_1 x_3 \oplus a_6 x_1 x_2 \oplus a_7 x_1 x_2 x_3.$$

Нам нужно

подобрать такие коэффициенты a_i , чтобы на каждом

наборе значений переменных выполнялось равенство

$$f(\alpha_1, \alpha_2, \alpha_3) = P(\alpha_1, \alpha_2, \alpha_3).$$

Имеем:

$$f(0, 0, 0) = 1 = P(0, 0, 0) = a_0 \oplus a_1 \cdot 0 \oplus a_2 \cdot 0 \oplus a_3 \cdot 0 \cdot 0 \oplus$$

$$\oplus a_4 \cdot 0 \oplus a_5 \cdot 0 \cdot 0 \oplus a_6 \cdot 0 \cdot 0 \oplus a_7 \cdot 0 \cdot 0 \cdot 0 \Rightarrow a_0 = 1;$$

$$f(0, 0, 1) = 0 = P(0, 0, 1) = 1 \oplus a_1 \cdot 1 \oplus a_2 \cdot 0 \oplus a_3 \cdot 0 \cdot 1 \oplus$$

$$\oplus a_4 \cdot 0 \oplus a_5 \cdot 0 \cdot 1 \oplus a_6 \cdot 0 \cdot 0 \oplus a_7 \cdot 0 \cdot 0 \cdot 1 \Rightarrow a_1 = 1;$$

$$f(0, 1, 0) = 1 = P(0, 1, 0) = 1 \oplus 1 \cdot 0 \oplus a_2 \cdot 1 \oplus a_3 \cdot 1 \cdot 0 \oplus$$

$$\oplus a_4 \cdot 0 \oplus a_5 \cdot 0 \cdot 0 \oplus a_6 \cdot 0 \cdot 1 \oplus a_7 \cdot 0 \cdot 1 \cdot 0 \Rightarrow a_2 = 0;$$

$$f(0, 1, 1) = 1 = P(0, 1, 1) = 1 \oplus 1 \cdot 1 \oplus 0 \cdot 1 \oplus a_3 \cdot 1 \cdot 1 \oplus$$

$$\oplus a_4 \cdot 0 \oplus a_5 \cdot 0 \cdot 1 \oplus a_6 \cdot 0 \cdot 1 \oplus a_7 \cdot 0 \cdot 1 \cdot 1 \Rightarrow a_3 = 1;$$

$$f(1, 0, 0) = 0 = P(1, 0, 0) = 1 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \cdot 0 \oplus$$

$$\oplus a_4 \cdot 1 \oplus a_5 \cdot 1 \cdot 0 \oplus a_6 \cdot 1 \cdot 0 \oplus a_7 \cdot 1 \cdot 0 \cdot 0 \Rightarrow a_4 = 1;$$

$$f(1, 0, 1) = 1 = P(1, 0, 1) = 1 \oplus 1 \cdot 1 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \cdot 1 \oplus$$

$$\oplus 1 \cdot 1 \oplus a_5 \cdot 1 \cdot 1 \oplus a_6 \cdot 1 \cdot 0 \oplus a_7 \cdot 1 \cdot 0 \cdot 1 \Rightarrow a_5 = 0;$$

$$f(1, 1, 0) = 1 = P(1, 1, 0) = 1 \oplus 1 \cdot 0 \oplus 0 \cdot 1 \oplus 1 \cdot 1 \cdot 0 \oplus 1 \cdot 1 \oplus$$

$$\oplus 0 \cdot 1 \cdot 0 \oplus a_6 \cdot 1 \cdot 1 \oplus a_7 \cdot 1 \cdot 1 \cdot 0 \Rightarrow a_6 = 1;$$

$$f(1, 1, 1) = 1 = P(1, 1, 1) = 1 \oplus 1 \cdot 1 \oplus 0 \cdot 1 \oplus 1 \cdot 1 \cdot 1 \oplus 1 \cdot 1 \oplus$$

$$\oplus 0 \cdot 1 \cdot 1 \oplus 1 \cdot 1 \cdot 1 \oplus a_7 \cdot 1 \cdot 1 \cdot 1 \Rightarrow a_7 = 0.$$

Теперь, когда коэффициенты найдены, запишем полином Жегалкина:

$$f(x_1, x_2, x_3) = 1 \oplus 1 \cdot x_3 \oplus 0 \cdot x_2 \oplus 1 \cdot x_2 x_3 \oplus \\ \oplus 1 \cdot x_1 \oplus 0 \cdot x_1 x_3 \oplus 1 \cdot x_1 x_2 \oplus 0 \cdot x_1 x_2 x_3.$$

Упростив запись, получим

$$f(x_1, x_2, x_3) = 1 \oplus x_3 \oplus x_2 x_3 \oplus x_1 \oplus x_1 x_2. \blacktriangleright$$

Упражнение 2.26. Используя метод неопределенных коэффициентов, построить полином Жегалкина функции:

$$\text{а) } f(x_1, x_2) = (1011); \quad \text{б) } f(x_1, x_2, x_3) = (11100101).$$

Определение. Булева функция называется *линейной*, если ее полином Жегалкина имеет степень 0 или 1.

Иначе говоря, функция линейна, если ее можно представить формулой вида

$$f = b_0 \oplus b_1 x_1 \oplus b_2 x_2 \oplus \dots \oplus b_n x_n.$$

Обозначим через **L** множество всех линейных булевых функций, через **L(n)** - множество линейных функций от n переменных.

Пример 10. а) Все функции одной переменной $0, 1, x, \bar{x} = x \oplus 1$ линейные.

б) Функции $xy, x \vee y = x \oplus y \oplus xy, x|y = 1 \oplus xy, x \downarrow y = 1 \oplus x \oplus y \oplus xy$ и

$x \rightarrow y = \bar{x} \vee y = \bar{x} \oplus y \oplus \bar{x}y = 1 \oplus x \oplus xy$ линейными не являются. Функции $x \oplus y,$

$x \leftrightarrow y = \overline{x \oplus y} = x \oplus y \oplus 1$ линейные.

Упражнение 2.27. Задать полиномами Жегалкина все линейные функции от двух переменных.

Множества функций **T₀**, **T₁**, **S**, **M**, **L** называют *классами Поста*.

Пример 11. Определить, каким из классов Поста принадлежит функция $f(x, y, z) = (x \vee y) \oplus (z \downarrow \bar{x})$.

◀ Выпишем таблицу истинности функции f (табл. 2.28).

Таблица 2.28

| x | y | z | $x \vee y$ | \bar{x} | $z \downarrow \bar{x}$ | $f(x, y, z) = (x \vee y) \oplus (z \downarrow \bar{x})$ |
|-----|-----|-----|------------|-----------|------------------------|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

$f(0,0,0)=0$, следовательно, $f \in \mathbf{T}_0$.

$f(1,1,1)=1$, следовательно, $f \in \mathbf{T}_1$.

$f(0,0,1)=f(1,1,0)$, следовательно, $f \notin \mathbf{S}$.

$(0,1,0) \mathbf{p} (1,1,0)$, а $f(0,1,0) > f(1,1,0)$, следовательно, $f \notin \mathbf{M}$.

Чтобы определить, линейна ли функция, найдем ее полином Жегалкина методом элементарных преобразований. Выпишем СДНФ функции и упростим ее:

$$f = \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}\bar{z} \vee xyz = \bar{x}y(\bar{z} \vee z) \vee xz(\bar{y} \vee y) = \bar{x}y \vee xz =$$

$$= \bar{x}y \oplus xz = (x \oplus 1)y \oplus xz = xy \oplus y \oplus xz.$$

Поскольку в полиноме Жегалкина есть произведения переменных, то $f \notin \mathbf{L}$. ►

Упражнение 2.28. Определить, каким из классов Поста принадлежит функция $f(x, y, z) = (x \downarrow y) \leftrightarrow (z \vee y)$.

5. Замыкание системы булевых функций. Замыкание $[\mathbf{B}]$ системы булевых функций

\mathbf{B} - это множество всех функций, которые можно задать формулами над \mathbf{B} .

Заметим что термин «система» используется в дискретной математике как синоним слова «множество».

Пример 12. а) Пусть $B = \{1, \bar{x}\}$, тогда $[B] = \{0, 1, x, \bar{x}\}$.

б) Замыканием системы $B = \{\wedge, \vee, \neg\}$ является множество всех булевых функций P_2 , поскольку любая булева функция представима в виде СДНФ или СКНФ.

в) Замыканием системы $B = \{0, 1, \wedge, \oplus\}$ является множество всех булевых функций P_2 , поскольку любая булева функция представима в виде полинома Жегалкина.

Определение. Система функций B называется замкнутой, если $[B] = B$.

Классы Поста T_0, T_1, S, M, L являются замкнутыми множествами.

Из этого утверждения (его доказательство приведено во второй части параграфа) вытекает, что если функции f_1, f_2, \dots, f_k принадлежат некоторому классу Поста, то любая функция, записанная формулой с использованием только функций этого класса, также будет принадлежать данному классу Поста.

Пример 13. а) Функция $f = xy \vee xz \vee yz$ монотонная, поскольку она представлена исключительно через дизъюнкцию и конъюнкцию, а эти две функции - монотонные.

б) Функция $f = \overline{x \oplus y} \leftrightarrow \bar{z}$ линейная, поскольку она представлена через линейные функции - отрицание, сложение по модулю два и эквивалентность.

В то же время функцию, не принадлежащую какому-нибудь классу Поста, нельзя представить формулой только через функции, входящие в этот класс.

Пример 14. Доказать, что функцию $f = (1001)$ нельзя задать формулой над множеством $B = \{\wedge\}$.

◀ Конъюнкция \wedge принадлежит к классу T_0 . Класс T_0 является замкнутым множеством, следовательно, любая функция, заданная формулой над его функциями, сохраняет 0. Функция $f = (1001)$ этим свойством не обладает, значит, ее нельзя задать формулой только через конъюнкцию. ▶

Упражнение 2.29. Доказать, что функцию f нельзя задать формулой над множеством B , если:

а) $f = (1100)$, $B = \{\leftrightarrow\}$;

б) $f = (01111101)$, $B = \{\vee\}$;

в) $f = xy \oplus 1$, $B = \{\oplus, \neg\}$.

Теоретические обоснования

Теорема 2.10. *Замыкание обладает следующими свойствами:*

- 1) $B \subseteq [B]$;
- 2) $[[B]] = [B]$;
- 3) $(B_1 \subseteq B_2) \Rightarrow ([B_1] \subseteq [B_2])$;
- 4) $([B_1] \cup [B_2]) \subseteq [B_1 \cup B_2]$.

Доказательство. Справедливость утверждения (1) непосредственно следует из определения замыкания, поскольку всякая функция реализуется формулой в виде символической записи самой функции.

Справедливость утверждения (2) вытекает из индуктивного определения формулы: всякая функция из $[B]$ задается некоторой формулой над B , а тогда всякая функция из $[[B]]$, которая задается формулой над $[B]$, задается также некоторой формулой над B .

Утверждение (3) очевидно, поскольку, если B_1 подмножество B_2 , то любая формула над B_1 является также формулой над B_2 .

Докажем утверждение (4). Возьмем произвольную функцию $f \in [B_1] \cup [B_2]$. Согласно определению объединения множеств, $f \in [B_1]$ или $f \in [B_2]$, т.е. f можно задать формулой над B_1 или формулой над B_2 . Но любая формула над B_1 (B_2) будет формулой и над $B_1 \cup B_2$. Следовательно, $f \in [B_1 \cup B_2]$, что и требовалось доказать. ■

Теорема 2.11. *Классы Поста T_0 , T_1 , S , M , L являются замкнутыми множествами.*

Доказательство. Надо доказать, что классы Поста совпадают со своими замыканиями. Для этого нужно доказать два утверждения: 1) каждый класс Поста принадлежит своему замыканию и 2) замыкание каждого класса Поста принадлежит самому классу.

Справедливость первого утверждения непосредственно вытекает из первого свойства замыкания.

Второе утверждение сводится к тому, что любая функция, заданная формулой через функции класса Поста, также принадлежит этому классу. Справедливость этого утверждения несложно установить индукцией по построению формул.

Базис индукции. Тожественная функция принадлежит всем классам Поста, значит, когда формула есть переменная, утверждение справедливо.

Индуктивный переход. 1. Покажем, что функция $F = f(f_1, f_2, \dots, f_n) \in \mathbf{T}_0$, если $f, f_1, f_2, \dots, f_n \in \mathbf{T}_0$:

$$F(0,0,...,0) = f(f_1(0,0,...,0), f_2(0,0,...,0), ..., f_n(0,0,...,0)) =$$

$$\stackrel{f_i \in \mathbf{T_0}}{=} f(0,0,...,0) \stackrel{f \in \mathbf{T_0}}{=} 0.$$

2. Покажем, что функция $F = f(f_1, f_2, \dots, f_n) \in \mathbf{T}_1$, если $f, f_1, f_2, \dots, f_n \in \mathbf{T}_1$:

$$F(1,1,\dots,1) = f\left(f_1(1,1,\dots,1), f_2(1,1,\dots,1), \dots, f_n(1,1,\dots,1)\right) =$$

$$\stackrel{f_i \in \mathbf{T}_1}{=} f(1,1,\dots,1) \stackrel{f \in \mathbf{T}_1}{=} 1.$$

3. Покажем, что функция $F = f(f_1, f_2, \dots, f_n) \in \mathbf{S}$, если $f, f_1, f_2, \dots, f_n \in \mathbf{S}$:

$$\begin{aligned} F^* &= (f(f_1, f_2, \dots, f_n))^* && \text{принип двойственности} \\ &= f^*(f_1^*, f_2^*, \dots, f_n^*) = f(f_1, f_2, \dots, f_n) = F. \end{aligned}$$

4. Покажем, что функция $F = f(f_1, f_2, \dots, f_n) \in \mathbf{M}$, если $f, f_1, f_2, \dots, f_n \in \mathbf{M}$.

Возьмем два произвольных набора \mathbf{a} и \mathbf{b} значений переменных таких, что $\mathbf{a} \neq \mathbf{b}$.

Тогда в силу монотонности функций f_1, f_2, \dots, f_n имеем:

$$\xi_1 = f_1(\alpha) \leq f_1(\beta) = \eta_1;$$

$$\xi_2 = f_2 \left(\frac{\alpha}{\beta} \right) \leq f_2 \left(\frac{\alpha_0}{\beta_0} \right) = \eta_2 ;$$

.....

$$\xi_n = f_n(\alpha) \leq f_n(\beta) = \eta_n.$$

Следовательно, $(\xi_1, \xi_2, \dots, \xi_n) \mathbf{p}(\eta_1, \eta_2, \dots, \eta_n)$, и в силу монотонности f

$$f(\xi_1, \xi_2, \dots, \xi_n) \leq f(\eta_1, \eta_2, \dots, \eta_n).$$

Или

$$f\left(f_1\left(\frac{a}{b}\right), f_2\left(\frac{a}{b}\right), \dots, f_n\left(\frac{a}{b}\right)\right) \leq f\left(f_1\left(\frac{a}{b_0}\right), f_2\left(\frac{a}{b_0}\right), \dots, f_n\left(\frac{a}{b_0}\right)\right).$$

Таким образом, для любых наборов \mathbf{a} и \mathbf{b} таких что $\mathbf{a} \preceq \mathbf{b}$, выполняется неравенство

$$F\left(\frac{\alpha}{\beta}\right) \leq F\left(\frac{\alpha_0}{\beta_0}\right), \text{ что и требовалось доказать.}$$

2.21. Пусть $f(x_1, x_2, \dots, x_n)$ - произвольная булева функция; $\phi(x)$ получена из $f(x_1, x_2, \dots, x_n)$ путем отождествления всех переменных: $\phi(x) = f(x, x, \dots, x)$. Определить $\phi(x)$, если:

а) $f \in T_1 \setminus T_0$; б) $f \in S$.

2.22. Показать, что если $f \notin T_0$, то f либо немонотонная, либо несамодвойственная.

2.23. Доказать, что функция, двойственная к монотонной функции, монотонна.

2.24. Доказать, что функция, двойственная к линейной функции, линейна.

2.25. Доказать, что пересечение замкнутых классов - замкнутый класс.

2.26. Является ли объединение замкнутых классов замкнутым классом?

2.27. Доказать, что совокупность функций, двойственных к функциям из функционально замкнутого класса, образует функционально замкнутый класс.

2.28. Назовем булеву функцию f монотонно невозрастающей, если для любых наборов α и β значений переменных, таких что $\alpha \leq \beta$, выполняется неравенство $f(\alpha) \geq f(\beta)$. Привести пример булевой функции, заданной формулой над множеством монотонно невозрастающих функций, которая не является монотонно невозрастающей.

Замечание. Задача 2.28 объясняет, почему в теории булевых функций не рассматривают монотонно невозрастающие функции. Дело в том, что интерес представляют свойства функций, которые «наследуются» при задании функций формулами, т.е. такие свойства, которые выделяют обладающие ими функции в замкнутый класс. Множество монотонно невозрастающих функций замкнутым не является.

2.29. Дана произвольная несамодвойственная функция. Отождествить ее переменные так, чтобы получилась несамодвойственная функция от возможно меньшего числа переменных.

2.30. Дана произвольная немонотонная функция. Отождествить ее переменные так, чтобы получилась немонотонная функция от возможно меньшего числа переменных.

2.31. Доказать, что для монотонных функций справедливо разложение $f(x_1, x_2, \dots, x_n) = x_n \cdot f(x_1, \dots, x_{n-1}, 1) \vee f(x_1, \dots, x_{n-1}, 0)$.

2.32. Показать, что x_i является существенной переменной тогда и только тогда, когда x_i явно входит в полином Жегалкина функции.

2.33. Найти число линейных функций от n переменных, существенно зависящих ровно от k переменных.

§ 2.5. Полнота системы булевых функций

Полная система. Теорема о полноте двух систем. Критерий полноты системы булевых функций. Базисы. Предполные классы.

Базовые понятия и утверждения

1. Понятие о полноте системы булевых функций. В предыдущем параграфе мы установили, что функцию, не принадлежащую какому-нибудь классу Поста, нельзя представить формулой через функции, входящие только в этот класс. С другой стороны, нам известно, что любую булеву функцию можно выразить через дизъюнкцию, конъюнкцию и отрицание. Настоящий параграф посвящен обсуждению следующего вопроса: как для произвольной системы булевых функций B определить, всякая ли функция представима формулой над B , или нет?

Определение. Множество булевых функций $B = \{f_1, f_2, \dots, f_m, \dots\}$ называется *полной* системой, если любая булева функция может быть задана формулой над B .

Пример 1. а) $\{\wedge, \vee, \neg\}$ - полная система, поскольку любая булева функция представима в виде СДНФ или СКНФ.

б) $\{0, 1, \wedge, \oplus\}$ - полная система, поскольку любая булева функция представима в виде полинома Жегалкина.

в) $\{\bar{x}, x \wedge y\}$ - полная система, поскольку любую булеву функцию можно сначала выразить формулой над $\{\wedge, \vee, \neg\}$, а затем заменить в этой формуле все дизъюнкции формулой $x \vee y = \overline{\bar{x} \wedge \bar{y}}$. В итоге получится формула над $\{\bar{x}, x \wedge y\}$.

г) $\{\bar{x}, x \vee y\}$ - полная система, поскольку любую булеву функцию можно сначала выразить формулой над $\{\wedge, \vee, \neg\}$, а затем заменить в этой формуле все конъюнкции формулой $x \wedge y = \overline{\bar{x} \vee \bar{y}}$. В итоге получится формула над $\{\bar{x}, x \vee y\}$.

Следующая теорема обобщает подход, использованный нами для доказательства полноты систем $\{\bar{x}, x \wedge y\}$ и $\{\bar{x}, x \vee y\}$.

Теорема 2.12 (о полноте двух систем). Пусть заданы две системы булевых функций: $B_1 = \{f_1, f_2, \dots\}$ и $B_2 = \{g_1, g_2, \dots\}$. Тогда, если система B_1 - полная и каждая ее функция может быть задана формулой над B_2 , то система B_2 тоже полная.

Доказательство. Покажем, как для произвольной булевой функции h построить формулу над B_2 . Возьмем формулу для h над B_1 $h = \Phi[f_1, f_2, \dots]$ (в силу полноты B_1 такая формула обязательно существует) и заменим в ней вхождение символов функций f_1, f_2, \dots формулами $f_i = \Phi_i[g_1, g_2, \dots]$ над B_1 (которые существуют по условию). Получим формулу $\Phi[\Phi_1[g_1, g_2, \dots], \Phi_2[g_1, g_2, \dots], \dots]$ над B_2 , которая задает h . ■

Пример 2. Докажем, что $\{x|y\}$ - полная система. Действительно, пусть $B_1 = \{\bar{x}, x \wedge y\}$ и $B_2 = \{x|y\}$. Имеем $x|y \stackrel{\text{СКНФ}}{=} \bar{x} \vee \bar{y} = \overline{xy}$, откуда $xy = \overline{x|y}$. Следовательно, $x|x = \overline{xx} = \bar{x}$ и $xy = \overline{x|y} = (x|y)|(x|y)$. Таким образом, каждую функцию полной системы B_1 можно задать формулой над B_2 . Следовательно, условия теоремы о полноте двух систем выполнены и, значит, $B_2 = \{x|y\}$ - полная система.

С помощью теоремы о полноте двух систем можно доказать полноту некоторой системы булевых функций. Обосновать неполноту с ее помощью нельзя.

2. Критерий полноты системы булевых функций. Наиболее удобный способ проверки системы функций на полноту связан с использованием **критерия полноты Поста**.

Для того чтобы система функций $B = \{f_1, \dots, f_k\}$ была полна, необходимо и достаточно, чтобы для каждого из классов Поста в B нашлась функция f_i , ему не принадлежащая.

Доказательство этого утверждения приведено во второй части параграфа.

При проверке, выполняются ли для некоторой системы функций $B = \{f_1, \dots, f_k\}$ условия критерия Поста, удобно использовать таблицу, которую называют таблицей Поста (табл. 2.29).

В ячейках таблицы ставится знак «+» или «-», в зависимости от того, входит функция, стоящая в данной строке, в класс, стоящий в данном столбце, или не входит. В силу теоремы Поста для полноты системы необходимо и достаточно, чтобы в каждом столбце стоял хотя бы один минус.

Таблицу Поста исследуемой системы лучше заполнять по столбцам. Вначале рассматриваем класс T_0 . Если B окажется

Таблица 2.29

| Функции | Классы Поста | | | | |
|---------|--------------|-------|-----|-----|-----|
| | T_0 | T_1 | S | M | L |
| f_1 | | | | | |
| ... | ... | ... | ... | ... | ... |
| f_k | | | | | |

подмножеством T_0 (первый столбец таблицы заполнен только плюсами), то система неполна. В противном случае переходим к классу T_1 и т.д.

Пример 3. Используя критерий полноты, выяснить, полна ли система булевых функций B :

а) $B = \{xy \vee z, x \leftrightarrow y, x\bar{y}\}$;

б) $B = \{x \leftrightarrow y, x\bar{x}\}$.

◀ а) Для удобства рассуждений зададим функции системы B таблично (табл. 2.30 и 2.31).

Таблица 2.30

| x | y | z | xy | $f_1 = xy \vee z$ |
|-----|-----|-----|------|-------------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Таблица 2.31

| x | y | $f_2 = x \leftrightarrow y$ | $f_3 = x\bar{y}$ |
|-----|-----|-----------------------------|------------------|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Заполним таблицу Поста исследуемой системы (табл. 2.32).

Таблица 2.32

| Функции | Классы Поста | | | | |
|-----------------------------|--------------|-------|-----|-----|-----|
| | T_0 | T_1 | S | M | L |
| $f_1 = xy \vee z$ | + | + | − | + | − |
| $f_2 = x \leftrightarrow y$ | − | + | | − | |
| $f_3 = x\bar{y}$ | | − | | | |

Класс T_0 : $f_1(0,0,0)=0$, следовательно, $f_1 \in T_0$;

$f_2(0,0,0)=1$, следовательно, $f_2 \notin T_0$. Выяснить,

принадлежит или нет классу T_0 функция f_3 , не надо,

так как в системе уже нашлась функция, не

принадлежащая T_0 .

Класс T_1 : $f_1(1,1,1)=1$, следовательно, $f_1 \in T_1$;

$f_2(1,1,1)=1$, следовательно, $f_2 \in T_1$; $f_3(1,1,1)=0$, следовательно, $f_3 \notin T_1$.

Класс S : $f_1 = (01010111)$, $f_1^* = (00010101)$, следовательно, $f_1 \notin S$.

Класс **М** : перебрав все пары сравнимых векторов, убеждаемся, что условие монотонности для функции f_1 не нарушается, значит, $f_1 \in \mathbf{M}$; $(0,0) \mathbf{p} (0,1)$, а $f_2(0,0) > f_2(0,1)$, следовательно, $f_2 \notin \mathbf{M}$.

Класс **L** : $f_1 = xy \vee z = xy \oplus z \oplus xyz$, следовательно, $f_1 \notin \mathbf{L}$.

Таким образом, для каждого класса Поста можно указать функцию системы **В**, ему не принадлежащую. Следовательно, **В** - полная система.

б) Заполним таблицу Поста системы **В** (табл. 2.33).

Таблица 2.33

| Функции | Классы Поста | | | | |
|-----------------------------|----------------------|----------------------|----------|----------|----------|
| | T₀ | T₁ | S | M | L |
| $f_1 = x \leftrightarrow y$ | — | + | — | — | + |
| $f_2 = \bar{x}$ | | — | | | + |

Класс **T₀**: $f_1(0,0)=1$, следовательно, $f_1 \notin \mathbf{T}_0$.

Класс **T₁**: $f_1(1,1)=1$, следовательно, $f_1 \in \mathbf{T}_1$; $f_2(1,1)=0$, следовательно, $f_2 \notin \mathbf{T}_1$.

Класс **S**: $f_1 = (1001)$, $f_1^* = (0110)$, следовательно, $f_1 \notin \mathbf{S}$.

Класс **M**: $(0,0) \mathbf{p} (0,1)$, $f_1(0,0) > f_1(0,1)$, следовательно,

$f_1 \notin \mathbf{M}$.

Класс **L**: $f_1 = x \leftrightarrow y = \overline{x \oplus y} = 1 \oplus x \oplus y$, следовательно, $f_1 \in \mathbf{L}$; $f_2 = \bar{x} = 1 \oplus x$, следовательно, $f_2 \in \mathbf{L}$.

Поскольку обе функции системы **В** принадлежат классу линейных функций, то система неполна. ►

Упражнение 2.30. Используя критерий полноты, выяснить, полна ли система булевых функций **В**:

а) $\mathbf{B} = \{x \rightarrow y, xy|z\}$;

б) $\mathbf{B} = \{z \vee xy, 1, 0\}$.

Определение. Полная система функций называется *базисом*, если после удаления из нее любой функции получается неполная система.

Пример 4. Укажите подмножества **В**, являющиеся базисами:

а) $\mathbf{B} = \{x \vee y, xy, \bar{x}\}$;

б) $\mathbf{B} = \{xy \vee z, x \leftrightarrow y, xy\}$.

◀ а) Заполним таблицу Поста (табл. 2.34).

Для каждого класса Поста можно указать функцию системы, которая ему не принадлежит (для T_0 , T_1 и M это \bar{x} , для S и L это $x \vee y$ (xy)), значит, система $\{x \vee y, xy, \bar{x}\}$ полная. При этом анализ табл. 2.34 показывает, что эта система базисом не является, поскольку не теряет полноты при удалении $x \vee y$ или xy . Полученные в результате удаления этих функций множества $\{x \vee y, \bar{x}\}$ и $\{xy, \bar{x}\}$ являются базисами, так как полны и при удалении из них любой из функций полноту теряют ($x \vee y \in T_0$, $xy \in T_0$, $\bar{x} \in L$).

Таблица 2.34

| Функции | Классы Поста | | | | |
|------------|--------------|-------|-----|-----|-----|
| | T_0 | T_1 | S | M | L |
| $x \vee y$ | + | + | – | + | – |
| xy | + | + | – | + | – |
| \bar{x} | – | – | + | – | + |

б) Мы рассматривали систему $B = \{xy \vee z, x \leftrightarrow y, xy\}$ в примере 3.

Таблица 2.35

| Функции | Классы Поста | | | | |
|-----------------------------|--------------|-------|-----|-----|-----|
| | T_0 | T_1 | S | M | L |
| $f_1 = xy \vee z$ | + | + | – | + | – |
| $f_2 = x \leftrightarrow y$ | – | + | – | – | + |
| $f_3 = x\bar{y}$ | + | – | – | – | – |

Чтобы выделить из этой полной системы базисы, заполним таблицу Поста системы B полностью (табл. 2.35).

Анализ табл. 2.35 показывает, что сама система B базисом не является, так как при удалении функции $xy \vee z$ полноту не теряет. Система $\{x \leftrightarrow y, xy\}$ является базисом, поскольку удаление из нее любой из функций

приводит к потере полноты. Остальные собственные подмножества B базисами не являются (система $\{xy \vee z, x \leftrightarrow y\}$ неполная, так как лежит в T_1 ; система $\{xy \vee z, x\bar{y}\}$ неполная, так как лежит в T_0 ; неполны и подмножества, содержащие по одной функции). ►

Упражнение 2.31. Указать подмножества B , являющиеся базисами:

- а) $B = \{x \oplus y, 0, 1, xy\}$; б) $B = \{x \downarrow y, x|y\}$.

Теоретические обоснования

Лемма о функции, не сохраняющей 0. Пусть $f \notin T_0$. Тогда формулой над множеством $\{f\}$ можно задать либо константу 1, либо отрицание.

Доказательство. Пусть $f \notin T_0$, т.е. $f(0, 0, \dots, 0) = 1$. отождествим между собой все переменные функции f . Получим функцию от одной переменной $\phi(x) = f(x, x, \dots, x)$.

Возможны два случая: $f(1, 1, \dots, 1) = 0$ или $f(1, 1, \dots, 1) = 1$.

В первом случае $f(0,0,...,0)=1$ и $f(1,1,...,1)=0$, следовательно, $\varphi(0)=1$ и $\varphi(1)=0$, откуда $\varphi(x)=\bar{x}$.

Во втором случае $f(0,0,...,0)=1$ и $f(1,1,...,1)=1$, следовательно, $\varphi(0)=1$ и $\varphi(1)=1$, откуда $\varphi(x)=1$. ■

Лемма о функции, не сохраняющей 1. Пусть $f \notin T_1$. Тогда формулой над множеством $\{f\}$ можно задать либо константу 0, либо отрицание.

Доказательство. Пусть $f \notin T_1$, т.е. $f(1,1,...,1)=0$. отождествим между собой все переменные функции f . Получим функцию от одной переменной $\varphi(x)=f(x,x,...,x)$.

Возможны два случая: $f(0,0,...,0)=0$ или $f(0,0,...,0)=1$.

В первом случае $f(0,0,...,0)=0$ и $f(1,1,...,1)=0$, следовательно,

$\varphi(0)=0$ и $\varphi(1)=0$, откуда $\varphi(x)=0$.

Во втором случае $f(0,0,...,0)=1$ и $f(1,1,...,1)=0$, следовательно, $\varphi(0)=1$ и $\varphi(1)=0$, откуда $\varphi(x)=\bar{x}$. ■

На рис. 2.1 и 2.2 представлены условные обозначения алгоритмов, использованных при доказательствах леммы о функции, не сохраняющей 0, и леммы о функции, не сохраняющей 1.

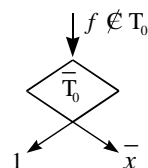


Рис. 2.1.

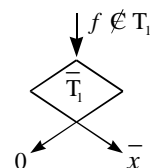


Рис. 2.2.

Лемма о несамодвойственной функции. Пусть $f(x_1, x_2, ..., x_n)$ - несамодвойственная функция. Тогда

формулой над множеством $\{f, \neg\}$ можно задать константы 0 и 1.

Доказательство. Пусть $f \notin S$. Тогда существует набор $(\alpha_1, \alpha_2, ..., \alpha_n)$ значений переменных $(x_1, x_2, ..., x_n)$, такой что $f(\alpha_1, \alpha_2, ..., \alpha_n) \neq \overline{f(\overline{\alpha_1}, \overline{\alpha_2}, ..., \overline{\alpha_n})}$, и, значит,

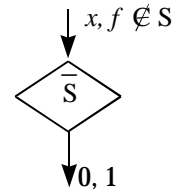
$$f(\alpha_1, \alpha_2, ..., \alpha_n) = f(\overline{\alpha_1}, \overline{\alpha_2}, ..., \overline{\alpha_n}).$$

Рассмотрим функцию $\varphi(x) = f(x^{\alpha_1}, x^{\alpha_2}, ..., x^{\alpha_n})$, где $x^{\alpha_i} = \bar{x}$, если $\alpha_i = 0$, и $x^{\alpha_i} = x$, если $\alpha_i = 1$. Заметим, что $0^{\alpha_i} = \overline{\alpha_i}$, $1^{\alpha_i} = \alpha_i$.

Имеем

$$\varphi(0) = f(0^{\alpha_1}, 0^{\alpha_2}, \dots, 0^{\alpha_n}) = f(\overline{\alpha_1}, \overline{\alpha_2}, \dots, \overline{\alpha_n});$$

$$\varphi(1) = f(1^{\alpha_1}, 1^{\alpha_2}, \dots, 1^{\alpha_n}) = f(\alpha_1, \alpha_2, \dots, \alpha_n).$$



С учетом равенства

$$f(\alpha_1, \alpha_2, \dots, \alpha_n) = f(\overline{\alpha_1}, \overline{\alpha_2}, \dots, \overline{\alpha_n})$$

Рис. 2.3.

получим $\varphi(0) = \varphi(1)$, т.е. $\varphi(x)$ - константа, $\overline{\varphi(x)}$ - другая константа. ■

На рис. 2.3 представлено условное обозначение алгоритма, использованного в доказательстве леммы о несамодвойственной функции.

Лемма о немонотонной функции. Пусть $f(x_1, x_2, \dots, x_n)$ - немонотонная функция.

Тогда формулой над множеством $\{f, 0, 1\}$ можно задать отрицание.

Доказательство. Пусть $f \notin \mathbf{M}$. Тогда существуют такие наборы $(\alpha_1, \alpha_2, \dots, \alpha_n)$ и $(\beta_1, \beta_2, \dots, \beta_n)$ значений переменных, что $(\alpha_1, \alpha_2, \dots, \alpha_n) \mathbf{p} (\beta_1, \beta_2, \dots, \beta_n)$, а $f(\alpha_1, \alpha_2, \dots, \alpha_n) > f(\beta_1, \beta_2, \dots, \beta_n)$, и, значит, $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$, $f(\beta_1, \beta_2, \dots, \beta_n) = 0$.

Поскольку $(\alpha_1, \alpha_2, \dots, \alpha_n) \neq (\beta_1, \beta_2, \dots, \beta_n)$, то можно выделить подпоследовательность индексов $1 \leq i_1 < i_2 < \dots < i_k \leq n$ такую, что $\alpha_{i_1} = \beta_{i_1}, \alpha_{i_2} = \beta_{i_2}, \dots, \alpha_{i_k} = \beta_{i_k}$ и для всякого $i \neq i_1, i_2, \dots, i_k$ $\alpha_i \neq \beta_i$ (и, значит, $\alpha_i = 0, \beta_i = 1$).

Рассмотрим функцию $\varphi(x)$, заданную формулой, полученной в результате подстановки в формулу $f(x_1, x_2, \dots, x_n)$ вместо переменных $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ чисел $\alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_k}$ соответственно, а на остальные места - x . Не ограничивая общности рассуждений, можно считать, что

$$\varphi(x) = f(x, \dots, x, \alpha_{i_1}, x, \dots, x, \alpha_{i_2}, x, \dots, x, \alpha_{i_k}, x, \dots, x).$$

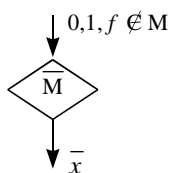
Так как $\alpha_i = 0$ при $i \neq i_1, \dots, i_k$, имеем

$$\varphi(0) = f(0, \dots, 0, \alpha_{i_1}, 0, \dots, 0, \alpha_{i_2}, 0, \dots, 0, \alpha_{i_k}, 0, \dots, 0) = f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1.$$

Так как $\beta_i = 1$ при $i \neq i_1, \dots, i_k$ и $\alpha_{i_1} = \beta_{i_1}, \dots, \alpha_{i_k} = \beta_{i_k}$, имеем

$$\varphi(1) = f(1, \dots, 1, \alpha_{i_1}, 1, \dots, 1, \alpha_{i_2}, 1, \dots, 1, \alpha_{i_k}, 1, \dots, 1) = f(\beta_1, \beta_2, \dots, \beta_n) = 0.$$

Равенства $\varphi(0) = 1$ и $\varphi(1) = 0$ означают, что $\varphi(x) = \overline{x}$. ■



На рис. 2.4 представлено условное обозначение алгоритма, использованного при доказательстве леммы о немонотонной функции.

Рис. 2.4.

Лемма о нелинейной функции. Пусть $f \notin L$. Тогда формулой над множеством $\{f, 0, 1, \bar{x}\}$ можно задать конъюнкцию.

Доказательство. Пусть $f \notin L$. Тогда в каноническом полиноме Жегалкина данной функции найдется член с отличным от нуля коэффициентом, содержащий произведение переменных. Не нарушая общности рассуждений, можно считать, что этот член содержит $x_1 \cdot x_2$. Следовательно, полином Жегалкина функции f можно преобразовать к виду

$$f = x_1 x_2 \cdot f_1(x_3, x_4, \dots, x_n) \oplus x_1 \cdot f_2(x_3, x_4, \dots, x_n) \oplus \\ \oplus x_2 \cdot f_3(x_3, x_4, \dots, x_n) \oplus f_4(x_3, x_4, \dots, x_n),$$

где функция f_1 такова, что найдется набор $(\sigma_3, \sigma_4, \dots, \sigma_n)$ значений переменных такой, что $f_1(\sigma_3, \sigma_4, \dots, \sigma_n) \neq 0$.

Рассмотрим функцию $\phi(x_1, x_2) = f(x_1, x_2, \sigma_3, \sigma_4, \dots, \sigma_n)$, или

$$\phi(x_1, x_2) = x_1 x_2 \cdot f_1(\sigma_3, \sigma_4, \dots, \sigma_n) \oplus x_1 \cdot f_2(\sigma_3, \sigma_4, \dots, \sigma_n) \oplus \\ \oplus x_2 \cdot f_3(\sigma_3, \sigma_4, \dots, \sigma_n) \oplus f_4(\sigma_3, \sigma_4, \dots, \sigma_n) = \\ = x_1 x_2 \oplus \alpha x_1 \oplus \beta x_2 \oplus \gamma,$$

где $\alpha = f_2(\sigma_3, \sigma_4, \dots, \sigma_n)$, $\beta = f_3(\sigma_3, \sigma_4, \dots, \sigma_n)$, $\gamma = f_4(\sigma_3, \sigma_4, \dots, \sigma_n)$.

Теперь рассмотрим функцию $\phi(x_1, x_2)$, получаемую из функции $\phi(x_1, x_2)$ следующим образом:

$$\phi(x_1, x_2) = \phi(x_1 \oplus \beta, x_2 \oplus \alpha) \oplus \alpha \beta \oplus \gamma.$$

Преобразуем формулу, реализующую функцию $\phi(x_1, x_2)$:

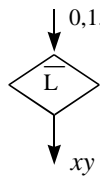
$$\phi(x_1, x_2) = (x_1 \oplus \beta) \cdot (x_2 \oplus \alpha) \oplus \alpha (x_1 \oplus \beta) \oplus \beta (x_2 \oplus \alpha) \oplus \gamma \oplus \alpha \beta \oplus \gamma = x_1 x_2 \oplus \beta x_2 \oplus x_1 \alpha \oplus \beta \alpha \oplus \alpha x_1 \oplus \alpha \beta \oplus \\ \oplus \beta x_2 \oplus \beta \alpha \oplus \gamma \oplus \alpha \beta \oplus \gamma = x_1 x_2.$$

Подытоживая проделанное, можем записать

$$x_1 x_2 = f(x_1 \oplus \beta, x_2 \oplus \alpha, \sigma_3, \sigma_4, \dots, \sigma_n) \oplus \alpha \beta \oplus \gamma.$$

Заметим, что

$$x_1 \oplus \beta = \begin{cases} x_1, & \text{если } \beta = 0; \\ \overline{x_1}, & \text{если } \beta = 1, \end{cases} \quad x_2 \oplus \alpha = \begin{cases} x_2, & \text{если } \alpha = 0; \\ \overline{x_2}, & \text{если } \alpha = 1, \end{cases} \quad f \oplus \alpha \beta \oplus \gamma = \begin{cases} f, & \text{если } \alpha \beta \oplus \gamma = 0; \\ \overline{f}, & \text{если } \alpha \beta \oplus \gamma = 1. \end{cases}$$



Таким образом, можно утверждать, что конъюнкция $x_1 x_2$ реализована формулой над множеством $\{f, 0, 1, \overline{x}\}$. ■

На рис. 2.5 представлено условное обозначение алгоритма, использованного при доказательстве леммы о нелинейной функции.

Рис. 2.5.

Теорема 2.13 (критерий полноты Поста). *Для того чтобы система функций $B = \{f_1, \dots, f_k\}$ была полна, необходимо и достаточно, чтобы для каждого из классов Поста в B нашлась функция f_i , ему не принадлежащая.*

Доказательство. Необходимость. Пусть система B полна. Надо доказать, что для каждого из классов Поста найдется функция f_i , ему не принадлежащая.

Рассуждаем от противного: предположим, что найдется класс Поста (обозначим его D), в котором содержатся все функции из B . Тогда

$$(B \subset D) \Rightarrow ([B] \subset [D]) \xRightarrow[\substack{\text{т.к. } B \text{ полна,} \\ \text{а } D \text{ замкнут}}]{\Rightarrow} (P_2 \subset D).$$

Следовательно, $(P_2 \subset D) \wedge (D \subset P_2)$. т.е. $D = P_2$. Но этого не может быть, поскольку есть булевы функции, не принадлежащие ни одному из классов Поста (например, штрих Шеффера). Получили противоречие, следовательно, наше предположение было неверным.

Достаточность. Пусть найдутся функции f_0, f_1, f_S, f_M, f_L из B такие, что $f_0 \notin T_0$, $f_1 \notin T_1$, $f_S \notin S$, $f_M \notin M$, $f_L \notin L$. Покажем, что отрицание и конъюнкцию можно реализовать формулами над множеством $\{f_0, f_1, f_S, f_M, f_L\}$. Схема построения соответствующих формул представлена на рис. 2.6.

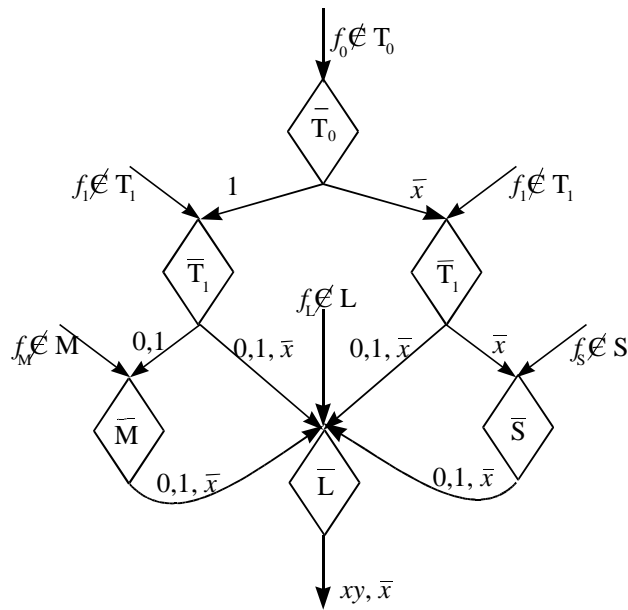


Рис. 2.6.

Таким образом, конъюнкцию и отрицание можно задать формулой над \mathbf{B} . Имеем: система $\{\bar{x}, xy\}$ полная, и каждая ее функция может быть выражена формулой над \mathbf{B} . Следовательно, условия теоремы о полноте двух систем выполнены и, значит, \mathbf{B} - полная система. ■

Задачи повышенной сложности

- 2.34.** Используя теорему о полноте двух систем, доказать, что $\{x \downarrow y\}$ полная система.
- 2.35.** Опираясь на доказательство леммы о функции, не сохраняющей 0, реализовать формулой над множеством $f = (10010000)$ либо 1, либо отрицание.
- 2.36.** Опираясь на доказательство леммы о функции, не сохраняющей 1, реализовать формулой над множеством $f = (00010000)$ либо 0, либо отрицание.
- 2.37.** Опираясь на доказательство леммы о несамодвойственной функции, реализовать формулой над множеством $\{(01101011), \bar{x}\}$ константы 0 и 1.
- 2.38.** Опираясь на доказательство леммы о немонотонной функции, реализовать формулой над множеством $\{(01101011), 0, 1\}$ отрицание.
- 2.39.** Опираясь на доказательство леммы о нелинейной функции, реализовать формулой над множеством $\{(11100100), 0, 1\}$ конъюнкцию.

2.40. Доказать, что из любой полной системы можно выделить полную подсистему, содержащую не более пяти функций.

2.41. Доказать, что всякий замкнутый класс функций, не совпадающий с P_2 , содержится, по крайней мере, в одном из классов Поста.

2.42. Назовем систему булевых функций предполным классом, если эта система неполна и добавление к ней любой функции, ей не принадлежащей, приводит к образованию полной системы. Доказать, что:

а) классы Поста являются предполными;

б) других предполных классов, кроме классов Поста, нет.

2.43. Доказать, что во всяком базисе содержится не более четырех функций.

Глава 3. Теория графов

§ 3.1. Основные определения

Граф, его вершины и ребра. Смежные вершины. Кратные ребра, петли. Инцидентные вершины и ребра. Степени вершин. Висячие и изолированные вершины. Лемма о рукопожатиях. Диаграмма графа. Изоморфные графы. Специальные виды графов: обыкновенные, полные, двудольные, полные двудольные графы. Матрица смежности и матрица инцидентности. Подграф. Операции над графами: объединение, пересечение, декартово произведение.

Базовые понятия и утверждения

1. Общие понятия. Различают два вида графов - неориентированные и ориентированные. Знакомство с теорией графов мы начнем с изучения графов первого вида.

Определение. Пусть V - конечное непустое множество, E - конечное множество, состоящее из поименованных неупорядоченных пар элементов множества V , причем это могут быть пары из одинаковых элементов и одинаковые пары с разными именами. Совокупность множеств V и E называют *графом* (или *неориентированным графом*) и обозначают $G = (V, E)$.

Элементы множества V называют *вершинами*, а элементы множества E - *ребрами*.

Граф может вовсе не иметь ребер. Такой граф называют *нулевым*.

Множества вершин V и ребер E графа G также обозначают $V(G)$ и $E(G)$ соответственно. Если ребро e - это пара вершин v_i и v_j , то пишут $e = v_i v_j$.

Обычно графы представляют диаграммами. Каждой вершине графа v_i ставят в соответствие свою точку плоскости или пространства, которую помечают тем же символом, что и вершину v_i , а каждому ребру $e = v_i v_j$ ставят в соответствие непрерывную кривую, соединяющую точки v_i , v_j и не проходящую через точки, изображающие другие вершины графа.

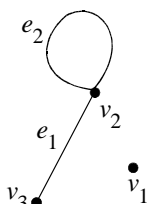


Рис. 3.1.

На рис. 3.1 приведен пример диаграммы графа с тремя вершинами v_1 , v_2 , v_3 и двумя ребрами $e_1 = v_2v_3$, $e_2 = v_2v_2$. Конечно, точки и кривые, изображающие вершины и ребра данного графа, можно было расположить на плоскости иначе. При задании графа важно лишь то, что некоторые из точек плоскости соединены кривой, а сам вид этой кривой и места расположения точек не имеют значения.

Если $e = v_i v_j$, то говорят, что ребро e соединяет вершины v_i и v_j , и вершины v_i и v_j называют *концами ребра e* . Если две вершины соединены ребром, их называют *смежными вершинами*. Если ребра имеют общую концевую вершину, то их называют *смежными ребрами*. Если v_i - конец ребра e , то ребро e и вершину v_i называют *инцидентными*. Если $e_1 = v_i v_j$ и $e_2 = v_i v_j$, то ребра e_1 и e_2 называют *кратными*. Ребро вида $e = v_i v_i$ называют *петлей*.

Граф без петель и кратных ребер называется *обыкновенным*.

Число ребер, инцидентных вершине v_i (петля учитывается дважды), называют *степенью* вершины и обозначают $\deg v_i$. Если $\deg v_i = 0$, то вершину v_i называют *изолированной*, а если $\deg v_i = 1$, то - *висячей*. Ребро, инцидентное висячей вершине, также называют *висячим*.

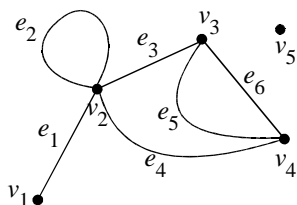


Рис. 3.2.

Пример 1. Рассмотрим граф с вершинами v_1, v_2, v_3, v_4, v_5 и ребрами $e_1 = v_1 v_2$, $e_2 = v_2 v_2$, $e_3 = v_2 v_3$, $e_4 = v_2 v_4$, $e_5 = v_3 v_4$, $e_6 = v_3 v_4$ (на рис. 3.2 изображена его диаграмма).

Вершины v_1 и v_2 , v_2 и v_3 , v_3 и v_4 , v_2 и v_4 - смежные. Ребра e_5 , e_6 - кратные, ребро e_2 - петля. Вершина v_1 - инцидентна

ребру e_1 ,

v_2 - ребрам e_1 , e_2 , e_3 и e_4 , v_3 - ребрам e_3 , e_5 и e_6 , v_4 - ребрам e_4 , e_5 и e_6 . Вершины графа имеют степени: $\deg v_1 = 1$, $\deg v_2 = 5$, $\deg v_3 = 3$, $\deg v_4 = 3$, $\deg v_5 = 0$. Вершина v_1 - висячая, вершина v_5 - изолированная. Ребро e_1 - висячее.

Упражнение 3.1. Изобразить диаграмму графа с вершинами a_1, a_2, a_3, a_4 и ребрами $e_1 = a_1 a_2$, $e_2 = a_1 a_3$, $e_3 = a_2 a_3$, $e_4 = a_2 a_4$. Указать смежные вершины, для каждой вершины перечислить инцидентные ей ребра, определить степени вершин, перечислить изолированные и висячие вершины (если они есть). Перечислить висячие ребра, кратные ребра и петли (если они есть).

Лемма (о рукопожатиях). Для любого графа $G = (V, E)$ сумма степеней вершин равна удвоенному числу ребер:

$$\sum_{v \in V} \deg v = 2|E|.$$

Доказательство. При подсчете суммы степеней вершин произвольное ребро $e = v_1 v_2$ внесет вклад, равный 1, как в степень вершины v_1 , так и в степень вершины v_2 , т.е. будет учтено в сумме дважды. ■

Пример 2. Выяснить, существует ли граф со следующим набором степеней вершин:

а) 1, 2, 2, 2, 3, 4, 5;

б) 1, 1, 1, 2, 3, 3, 3.

◀ а) Предположим, что такой граф G существует. Тогда, согласно утверждению леммы о рукопожатиях, сумма степеней его вершин равна удвоенному числу ребер, т.е. $1 + 2 + 2 + 2 + 3 + 4 + 5 = 2|E|$, откуда $|E| = 9,5$. Но это невозможно, так как число ребер не может быть дробным. Полученное противоречие показывает, что предположение о существовании графа с указанным набором степеней вершин было неверным.

б) Если граф с набором степеней вершин 1, 1, 1, 2, 3, 3, 3 существует, то он, согласно лемме о рукопожатиях, имеет 7 ребер ($1 + 1 + 1 + 2 + 3 + 3 + 3 = 2|E|$, откуда $|E| = 7$).

Действуя методом проб, несложно подобрать примеры таких графов (диаграммы двух из них изображены на рис. 3.3 и 3.4). ►

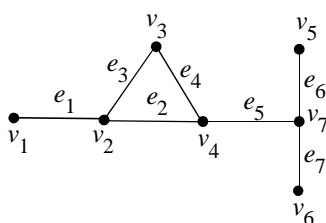


Рис. 3.3.

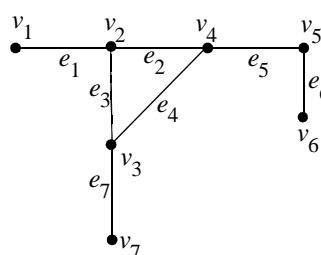


Рис. 3.4.

Упражнение 3.2. Выяснить, существует ли граф со следующим набором степеней вершин: 0, 1, 3, 3, 5.

2. Изоморфные графы. Графы G_1 и G_2 называются *изоморфными*, если существуют такие два взаимно-однозначных отображения $\phi: V(G_1) \rightarrow V(G_2)$ и $\phi: E(G_1) \rightarrow E(G_2)$, что для всякого ребра $e = vu$ из $E(G_1)$ справедливо $\phi(e) = \phi(v)\phi(u)$.

Иными словами, два графа изоморфны, если с помощью переименования вершин и ребер их можно сделать одинаковыми. У изоморфных графов одни и те же свойства,

выражаемые в терминах теории графов - например, одинаковое число вершин, ребер, висячих ребер, петель, вершин определенной степени.

Пример 3. Рассмотрим графы, представленные диаграммами на рис. 3.5, и выясним, какие из них изоморфны друг другу.

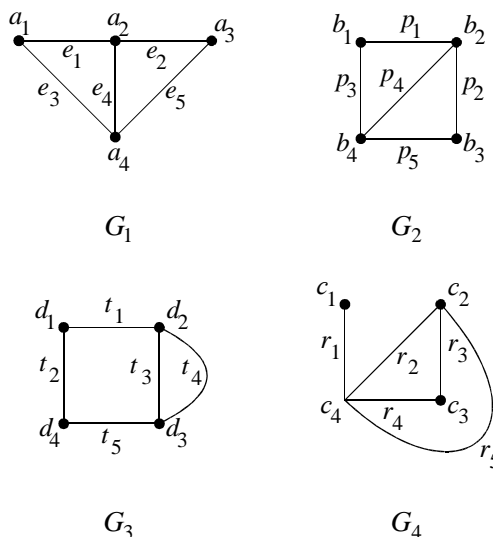


Рис. 3.5.

Графы G_1 и G_2 изоморфны. Действительно, отображения $\phi(a_i) = b_i$ и $\phi(e_i) = p_i$ ($1 \leq i \leq 4$) взаимно-однозначны и удовлетворяют условию, описанному в определении изоморфных графов.

Графы G_3 и G_4 неизоморфны графам G_1 и G_2 , поскольку в каждом из них есть кратные ребра, а в графах G_1 и G_2 кратных ребер нет. Друг другу графы G_3 и G_4 также неизоморфны, так как в графе G_3 каждые две вершины смежные, а в графе G_4 есть несмежные вершины.

Еще одним примером неизоморфных графов служат графы, изображенные на рис. 3.3 и 3.4. Действительно, в графе на рис. 3.3 все вершины степени 1 инцидентны вершинам степени 3, а в графе на рис. 3.4 это не так. Этот пример показывает, что набор степеней вершин не определяет граф однозначно даже с точностью до изоморфизма.

Для доказательства того, что графы изоморфны, достаточно указать отображения ϕ и ψ , удовлетворяющие условию, описанному в определении. Чтобы доказать, что графы неизоморфны, достаточно найти какое-нибудь свойство, которым обладает один граф и не обладает другой, и которое у изоморфных графов должно быть общим.

Упражнение 3.3. Выяснить, какие из графов, представленных на рис. 3.6 своими диаграммами, изоморфны между собой.

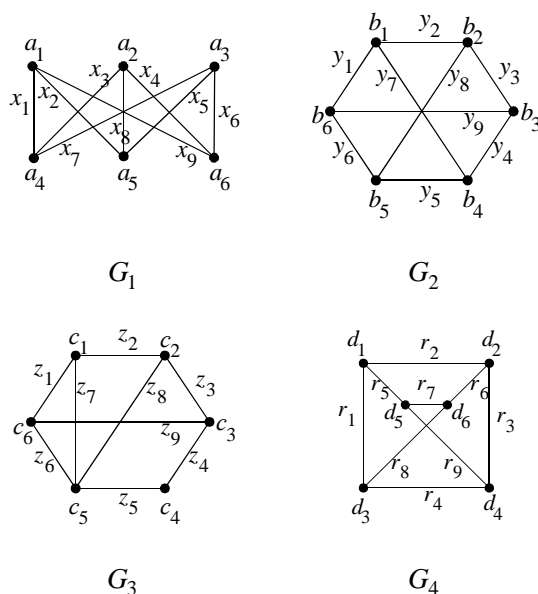


Рис. 3.6.

Рассмотрим на множестве графов бинарное отношение, состоящее из всех пар изоморфных графов. Назовем это бинарное отношение *отношением изоморфизма* и обозначим \cong (таким образом, запись $G_1 \cong G_2$ будет означать, что графы G_1 и G_2 изоморфны).

Отношение изоморфизма рефлексивно, симметрично и транзитивно и, значит, является отношением эквивалентности. Следовательно, оно разбивает множество всех графов на классы эквивалентности так, что графы одного класса попарно изоморфны, а графы разных классов не изоморфны. В дальнейшем, если не оговорено иное, графы рассматриваются с точностью до изоморфизма, т.е. объектом изучения являются не отдельные графы, а классы эквивалентности по отношению изоморфизма (эти классы называют абстрактными графами).

Каждый класс эквивалентности по отношению изоморфизма можно задать диаграммой, не указывая на ней имена вершин и ребер.

Упражнение 3.4. Найти все (с точностью до изоморфизма) обыкновенные графы с тремя вершинами.

3. Виды графов. Обыкновенный граф называется *полным*, если любые две его вершины смежные. Все полные графы с n вершинами изоморфны друг другу и образуют класс эквивалентности по отношению изоморфизма. Для этого класса, т.е. для полного абстрактного графа с n вершинами, применяется обозначение K_n .

На рис. 3.7 в качестве примера приведены диаграммы графов K_3 , K_4 , K_5 .

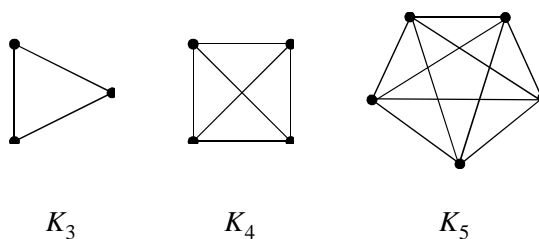


Рис. 3.7.

Заметим, что степень каждой вершины полного графа K_n равна $n - 1$, так что

$$2|E| = \sum_{v \in V} \deg v = \sum_{i=1}^n (n-1) = n(n-1).$$

Значит, число ребер полного графа с n вершинами равно $\frac{n(n-1)}{2}$.

Двудольным графом называется обыкновенный граф, множество вершин которого может быть разбито на два непустых непересекающихся подмножества (доли), так что концы каждого ребра графа принадлежат разным долям.

Двудольный граф называется *полным двудольным графом*, если любые две его вершины, принадлежащие разным долям, смежные.

Все полные двудольные графы с n вершинами в одной доле и m вершинами в другой ($n \leq m$) изоморфны друг другу и образуют класс эквивалентности по отношению изоморфизма. Для этого класса, т.е. для полного двудольного абстрактного графа с n вершинами в одной доле и m вершинами в другой ($n \leq m$), применяется обозначение $K_{n,m}$.

На рис. 3.8 в качестве примера приведены диаграммы графов $K_{2,3}$ и $K_{3,3}$.

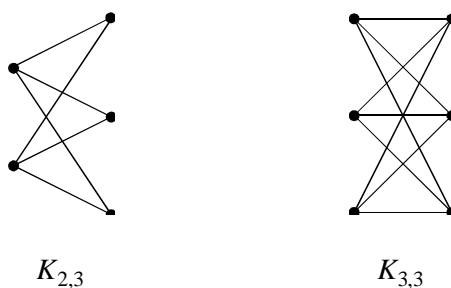


Рис. 3.8.

4. Матрица смежности и матрица инцидентности. Существует несколько способов задания графов. Во-первых, граф можно задать указанием множества вершин и полного

списка ребер (это задание «по определению»). Во-вторых, граф можно задать диаграммой. Кроме того, графы можно описать матрицами.

Пусть G - произвольный граф с m вершинами и n ребрами. Занумеруем вершины графа числами от 1 до m .

Определение. Матрицей смежности графа G называют квадратную матрицу $A(G)$ размера $m \times m$, в которой элемент a_{ij} , стоящий на пересечении i -й строки и j -го столбца, равен числу ребер, соединяющих вершины с номерами i, j , причем при $i = j$ каждая петля учитывается дважды.

Очевидно, что вид матрицы смежности графа зависит от того, как занумерованы вершины графа. Вообще говоря, для каждого графа имеется несколько матриц смежности. Матрицы смежности графа, соответствующие различным нумерациям вершин, получаются одна из другой с помощью некоторой перестановки строк и аналогичной перестановки столбцов (если строки i и j переставлены, то и столбцы i и j тоже должны быть переставлены).

Матрица смежности определяет граф с точностью до изоморфизма: графы, имеющие одинаковые матрицы смежности, изоморфны друг другу.

Пример 4. Составим матрицу смежности $A(G)$ графа G , представленного диаграммой на рис. 3.1. Вершинам v_1, v_2, v_3 дадим номера 1, 2, 3 соответственно. Поскольку в графе три вершины, то матрица $A(G)$ имеет три строки и три столбца. Будем заполнять матрицу $A(G)$ по строкам. Вершина v_1 не соединена ребром ни с одной из вершин, поэтому первая строка матрицы состоит из нулей. Вершина v_2 с вершиной v_1 не соединена, поэтому первый элемент второй строки равен 0. Далее, v_2 соединена сама с собой петлей, поэтому второй элемент второй строки равен 2. Кроме того, v_2 соединена ребром с v_3 , поэтому третий элемент второй строки равен 1. Вершина v_3 соединена ребром только с вершиной v_2 , поэтому в третьей строке второй элемент равен 1, а остальные элементы равны 0. Таким образом, у нас получилась матрица

$$A(G) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Матрица смежности любого графа симметрична относительно главной диагонали. Сумма элементов ее i -й строки равна степени i -й вершины. Матрица смежности обыкновенного графа состоит из нулей и единиц, причем ее главная диагональ состоит только из нулей.

Помимо вершин занумеруем ребра графа числами от 1 до n .

Определение. Матрицей инцидентности графа G называют матрицу $B(G)$ размера $m \times n$, элементы которой b_{ij} определяются следующими условиями:

- 1) $b_{ij} = 1$, если вершина с номером i инцидентна ребру с номером j и j -е ребро - не петля;
- 2) $b_{ij} = 0$ во всех остальных случаях.

Пример 5. Составим матрицу инцидентности $B(G)$ графа G , представленного диаграммой на рис. 3.1. Вершинам v_1, v_2, v_3 дадим номера 1, 2, 3 соответственно, а ребрам e_1, e_2 - номера 1 и 2. Поскольку в графе три вершины и два ребра, то матрица $B(G)$ имеет три строки и два столбца. Будем заполнять матрицу $B(G)$ по столбцам. Ребро e_1 соединяет вершины v_2 и v_3 , поэтому второй и третий элементы первого столбца равны 1, а первый элемент равен 0. Ребро e_2 - петля, поэтому второй столбец состоит из нулей. Таким образом, у нас получилась матрица

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

Вид матрицы инцидентности графа зависит от того, как занумерованы его вершины и ребра. Матрицы инцидентности, соответствующие различным нумерациям вершин и ребер графа, получаются одна из другой с помощью произвольной перестановки строк и произвольной перестановки столбцов.

Если в графе нет петель, то матрица инцидентности, так же как и матрица смежности, определяет граф с точностью до изоморфизма: графы, имеющие одинаковые матрицы инцидентности, изоморфны друг другу. Если в графе есть петли, то восстановить граф по матрице инцидентности однозначно нельзя, поскольку отсутствует указание на то, какие вершины являются концами петли.

Упражнение 3.5. Выписать матрицу смежности и матрицу инцидентности графа из упр. 3.1.

Упражнение 3.6. Изобразить диаграмму графа, матрица смежности которого имеет вид:

$$\text{а) } A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}; \quad \text{б) } A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Упражнение 3.7. Изобразить диаграмму графа, матрица инцидентности которого имеет вид:

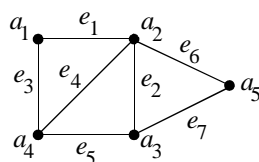
$$\text{а) } B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}; \quad \text{б) } B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

5. Подграфы и операции над ними. Пусть $G = (V, E)$ - граф и $V' \subset V$, $E' \subset E$. Если подмножества V' и E' таковы, что концы любого ребра из E' принадлежат множеству V' , то граф $G' = (V', E')$ называется *подграфом* графа G .

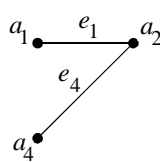
В число подграфов графа G будем включать *пустой подграф* и обозначать его \emptyset .

Пусть V' - подмножество множества вершин графа $G = (V, E)$. Включим в множество E' все ребра графа G , концы которых принадлежат V' . Подграф $G' = (V', E')$ графа $G = (V, E)$ назовем подграфом, *порожденным множеством* V' .

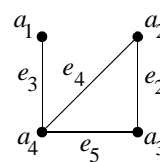
Пример 6. Рассмотрим граф G , представленный диаграммой на рис. 3.9. Графы H_1 , H_2 , H_3 , H_4 , H_5 - подграфы G . Подграф H_3 порожден подмножеством $V' = \{a_1, a_2, a_3, a_4\}$ вершин графа G , в то время как подграф H_2 , множество вершин которого совпадает с множеством вершин графа H_3 , не является подграфом, порожденным множеством своих вершин V' .



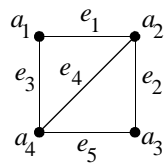
G



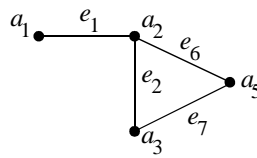
H_1



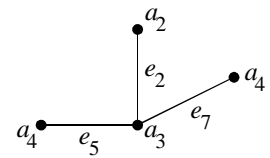
H_2



H_3



H_4



H_5

Рис. 3.9.

Пусть $H_1 = (V_1, E_1)$ и $H_2 = (V_2, E_2)$ - подграфы графа $G = (V, E)$.

Пересечением графов H_1 и H_2 называется граф $H_1 \cap H_2 = (V_1 \cap V_2, E_1 \cap E_2)$, т.е. граф, множества вершин и ребер которого являются пересечением соответствующих множеств графов H_1 и H_2 .

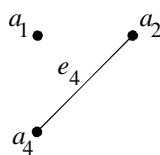
Объединением графов H_1 и H_2 называется граф $H_1 \cup H_2 = (V_1 \cup V_2, E_1 \cup E_2)$, т.е. граф, множества вершин и ребер которого являются объединением соответствующих множеств графов H_1 и H_2 .

Аналогично определяется пересечение и объединение любого конечного числа подграфов.

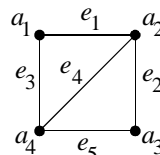
Пример 7. На рис. 3.10 показаны пересечения и объединения подграфов из примера 6.

Совокупность непустых, попарно-непересекающихся подграфов H_1, H_2, \dots, H_k графа G называют *дизъюнктным разбиением* графа G , если их объединение совпадает с

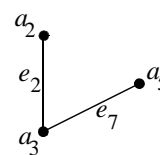
графом G , т.е. $G = \bigcup_{i=1}^k H_i$.



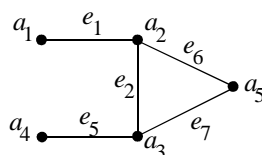
$H_1 \cap H_2$



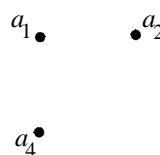
$H_1 \cup H_2$



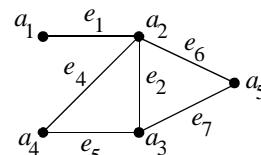
$H_4 \cap H_5$



$H_4 \cup H_5$



$H_1 \cap H_2 \cap H_5$



$H_1 \cup H_4 \cup H_5$

Рис. 3.10.

Пример 8. На рис. 3.11 показаны диаграммы графа G и трех его подграфов H_1 , H_2 , H_3 , образующих в совокупности дизъюнктное разбиение G .

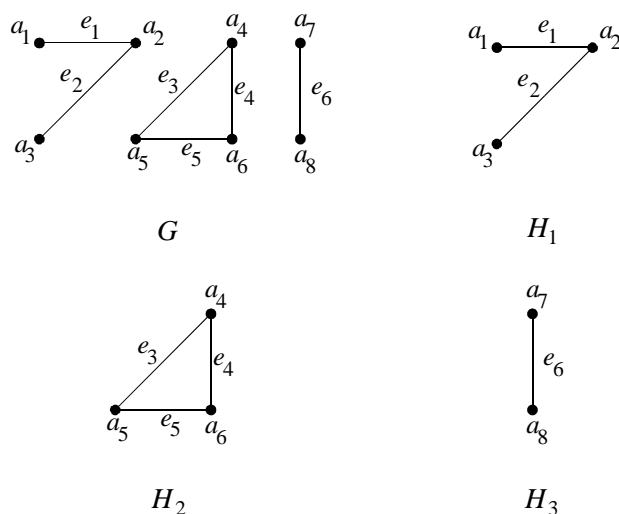


Рис. 3.11.

Назовем *декартовым произведением графов* $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ граф $G_1 \times G_2$, вершинами которого являются упорядоченные пары вида (v, u) , где $v \in V_1$, $u \in V_2$, и в котором вершины (v_1, u_1) и (v_2, u_2) смежные в точности в одном из двух случаев: 1) v_1 и v_2 - смежные вершины в графе G_1 , а $u_1 = u_2$; 2) u_1 и u_2 - смежные вершины в графе G_2 , а $v_1 = v_2$.

Пример 9. Пусть G_1 - граф с вершинами a и b и ребром $e = ab$, а G_2 - граф с вершинами p, q, r и ребрами $e_1 = pq, e_2 = qr$. На рис. 3.12 показаны диаграммы графов G_1, G_2 и $G_1 \times G_2$.

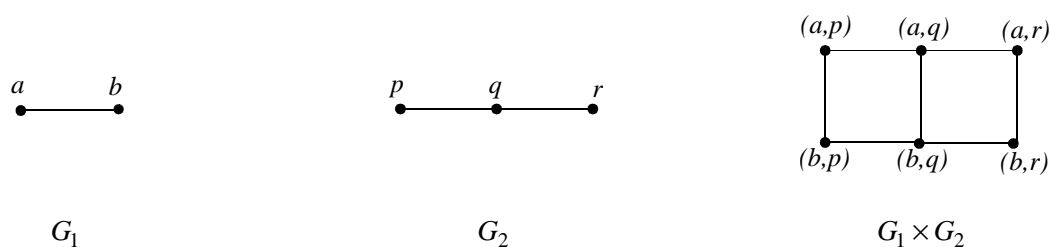


Рис. 3.12.

Упражнение 3.8. а) Пусть B_1 - граф с вершинами 0 и 1 и ребром 01. Построить диаграммы графов $B_2 = B_1 \times B_1$, $B_3 = B_2 \times B_1$.

б) Построить диаграмму графа $K_2 \times K_3$.

Удалением вершины v_i из графа $G = (V, E)$ называется операция, которая удаляет из графа G вершину v_i и все инцидентные ей ребра. Получающийся в результате подграф обозначают $G - v_i$.

Удалением ребра e_i из графа $G = (V, E)$ называется операция, которая удаляет из графа G ребро e_i (концы ребра не удаляются). Получающийся в результате подграф обозначают $G - e_i$.

Пример 10. На рис. 3.13 изображены диаграммы графа G , графа $G - a_3$, полученного из G удалением вершины a_3 , и графа $G - e_4$, полученного из G удалением ребра e_4 .

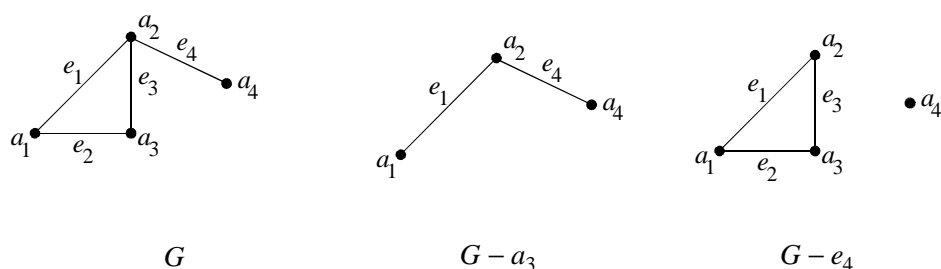


Рис. 3.13.

Задачи повышенной сложности

3.1. Доказать, что в любом графе G число вершин нечетной степени может быть только четным.

3.2. Рассмотрим на множестве вершин произвольного графа бинарное отношение смежности: будем считать вершины связанными этим отношением, если они соединены ребром. Нарисуйте диаграмму графа, для которого отношение смежности будет:

- а) рефлексивным, но не транзитивным;
- б) транзитивным, но не рефлексивным;
- в) отношением эквивалентности.

3.3. Пусть G - полный граф с множеством вершин $V = \{v_1, v_2, \dots, v_n\}$.

- а) Сколько у графа G подграфов с тем же множеством вершин, что и у графа G ?
- б) Сколько у графа G всего подграфов?

3.4. Доказать, что если графы G_1 и G_2 изоморфны и одно из ребер графа G_1 есть петля, то одно из ребер графа G_2 также является петлей.

3.5. Доказать, что если графы G_1 и G_2 изоморфны и $\varphi: VG_1 \rightarrow VG_2$ - взаимно-однозначное отображение в этом изоморфизме, то для любой вершины a графа G_1 выполняется равенство $\deg a = \deg \varphi(a)$.

3.6. Доказать, что если графы G_1 и G_2 изоморфны и на графе G_1 есть цикл длины k , то и на графе G_2 также есть цикл длины k .

3.7. Определим по индукции граф B_n : пусть B_1 - граф с вершинами 0 и 1 и ребром 01; тогда для любого натурального $n \geq 2$ $B_n = B_{n-1} \times B_1$. Найти число вершин и ребер графа B_n .

3.8. Пусть $G = (V, E)$ - обыкновенный граф, $A(G) = \|a_{i,j}\|$ - матрица смежности этого графа, отвечающая некоторой нумерации вершин v_1, v_2, \dots, v_n . Выразить через число вершин, ребер или степени вершин следующие суммы:

а) $\sum_{i=1}^n a_{i,j}$; б) $\sum_{i=1}^n \sum_{j=1}^n a_{i,j}$.

3.9. Сколько существует обыкновенных графов, у которых n вершин (вершины помечены)? Сколько из них имеет ровно m ребер?

3.10. Доказать, что в любом обыкновенном графе, содержащем более одной вершины, найдутся две различные вершины одинаковой степени.

§ 3.2. Достижимость и компоненты связности, циклы и мосты, цикломатическое число

Пути, цепи, циклы на графе. Отношение достижимости (связности).
Компоненты связности графа. Число связности. Связный граф. Мост.
Цикломатическое число.

Базовые понятия и утверждения

1. Пути, цепи, циклы на графе. *Путем длины k* на графе G из вершины v_0 в вершину v_k называется такая последовательность

$$v_0 e_1 v_1 \dots v_{k-1} e_k v_k$$

вершин и ребер графа, в которой $e_i = v_{i-1} v_i$ ($1 \leq i \leq k$).

Путь из v_0 в v_k обозначают $p_{v_0 v_k}$ и говорят, что он соединяет вершину v_0 с вершиной v_k . Вершины v_0 и v_k называют соответственно началом и концом пути.

Кроме того, каждую вершину считают путем длины нуль.

Если $v_0 = v_k$, то путь называется *замкнутым*.

Заметим, что в обыкновенном графе путь полностью определяется последовательностью v_0, v_1, \dots, v_k своих вершин.

В произвольном пути любое ребро и любая вершина могут повторяться. Накладывая ограничения на число повторений вершин и ребер, приходим к следующим частным видам путей.

Цепь - это путь без повторяющихся ребер. Цепь, соединяющую вершину v_0 с вершиной v_k , обозначают $z_{v_0 v_k}$.

Цепь называется *простой*, если в ней нет повторяющихся вершин, за исключением, быть может, совпадающих конечных.

Замкнутая цепь ненулевой длины называется *циклом*.

Замкнутая простая цепь называется *простым циклом*.

Пример 1. Рассмотрим граф на рис. 3.14.

$a e_2 b e_3 c e_5 b e_6 f$ - путь из a в f ; его длина равна 4; данный путь является цепью; эта цепь незамкнутая и не простая.

$f e_7 t e_1 d e_4 b e_6 f e_8 f$ - цикл длины 5, этот цикл не является простым;

$b e_3 c e_5 b$ - простой цикл.

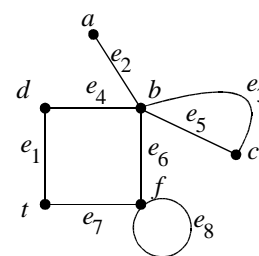


Рис. 3.14.

Лемма (о простой цепи). Если на графе существует путь из a в b , то существует и простая цепь, соединяющая вершины a и b .

Доказательство. Рассмотрим на графе путь наименьшей длины из a в b . Покажем, что этот путь является простой цепью. Будем рассуждать от противного. Пусть в нашем пути имеется повторяющаяся вершина c . Тогда, заменив часть пути от первого вхождения вершины c до ее второго вхождения на одну вершину c , мы получим более короткий путь из a в b . Получили противоречие. ■

Упражнение 3.9. На графе G_1 из упр. 3.3 найти:

- а) путь длины 7, не являющийся цепью;
- б) цепь длины 5, не являющуюся простой;
- в) цикл, не являющийся простым;
- г) простой цикл четной длины;
- д) простой цикл нечетной длины.

2. Достижимость и компоненты связности графа. На множестве вершин графа G введем бинарное отношение - *отношение достижимости (связности)* (\sim), образованное всеми теми парами вершин (a, b) , для которых на графе есть путь из a в b . Таким образом, запись $a : b$ будет означать, что на графе G есть путь из a в b .

Поскольку мы договорились считать каждую вершину графа a путем длины 0 из a в a , то для любой вершины a можно утверждать, что $a : a$. Это означает, что отношение достижимости рефлексивно.

Очевидно, что если на графе G есть путь из вершины a в вершину b , то есть и путь из b в a , т.е. если $a : b$, то и $b : a$. Значит, отношение достижимости симметрично.

Несложно показать, что отношение достижимости является также и транзитивным, т.е. если $a : b$ и $b : c$, то и $a : c$. Иными словами, из существования на графе G путей из a в b и из b в c следует существование пути из a в c . Это действительно так, поскольку из пути из a в $b : a e_1 v_1 \dots v_{k-1} e_k, b$ и пути из b в $c : b e'_1 v'_1 \dots v'_{k'-1} e'_{k'} c$ можно «склеить» путь из a в $c : a e_1 v_1 \dots v_{k-1} e_k b e'_1 v'_1 \dots v'_{k'-1} e'_{k'} c$.

Таким образом, отношение достижимости рефлексивно, симметрично и транзитивно, и, значит, является отношением эквивалентности.

Пусть a - произвольная вершина графа $G = (V, E)$. Обозначим через $[a]_{\sim}$ класс эквивалентности вершины a по отношению достижимости, т.е. $[a]_{\sim} = \{b \in V \mid a \sim b\}$.

Классы эквивалентности по отношению достижимости обладают рядом свойств, присущих классам эквивалентности любого бинарного отношения (см. теорему 1.1), а именно:

- 1) класс эквивалентности любой вершины по отношению достижимости - непустое множество;
- 2) классы эквивалентности любых двух вершин по отношению достижимости либо не пересекаются, либо совпадают;
- 3) объединение классов эквивалентности всех вершин графа по отношению достижимости совпадает с самим множеством вершин графа.

Вследствие перечисленных свойств отношение достижимости, заданное на множестве V вершин графа G , порождает разбиение множества V на классы эквивалентности по этому отношению.

Определение. Подграф G_a , порождаемый классом эквивалентности $[a]_{\sim}$ вершины a по отношению достижимости, называют *компонентой связности* вершины a .

Другими словами, компонента связности вершины a представляет собой подграф графа G с множеством вершин $[a]_{\sim}$ и множеством ребер, элементами которого являются все те ребра графа G , концы которых лежат в $[a]_{\sim}$.

Компоненты связности графа обладают следующими свойствами:

- 1) каждая компонента связности - непустой подграф;
- 2) компоненты связности любых двух вершин либо не пересекаются, либо совпадают;
- 3) объединение компонент связности всех вершин графа совпадает с самим графом.

Вследствие перечисленных свойств совокупность всех различных компонент связности графа образует его дизъюнктное разбиение.

Определение. Число различных компонент связности графа G называется *числом связности* и обозначается $k(G)$.

Если $k(G) = 1$, то граф называется *связным*. Иными словами, граф связный, если любая пара его вершин соединена путем.

Пример 2. Рассмотрим граф G с вершинами a, b, c, d, t, g и ребрами $e_1 = ab$, $e_2 = ac$, $e_3 = bc$, $e_4 = tg$ (рис. 3.15).

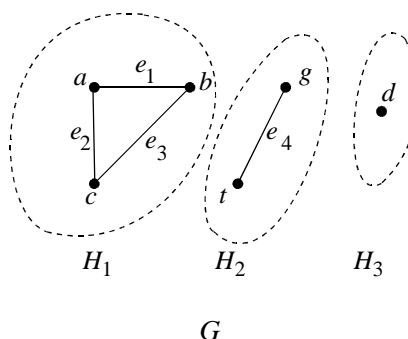


Рис. 3.15.

Перечислим классы эквивалентности вершин графа G по отношению достижимости:

$$[a]_{\sim} = [b]_{\sim} = [c]_{\sim} = \{a, b, c\}, [t]_{\sim} = [g]_{\sim} = \{t, g\}, [d]_{\sim} = \{d\}.$$

Таким образом, имеем три различных класса эквивалентности и, соответственно, три компоненты связности:

$$H_1 = (V_1, E_1), \text{ где } V_1 = \{a, b, c\}, E_1 = \{e_1 = ab, e_2 = ac, e_3 = bc\};$$

$$H_2 = (V_2, E_2), \text{ где } V_2 = \{t, g\}, E_2 = \{e_4 = tg\};$$

$$H_3 = (V_3, E_3), \text{ где } V_3 = \{d\}, E_3 = \emptyset.$$

Упражнение 3.10. Найти все (с точностью до изоморфизма) связные обыкновенные графы с четырьмя вершинами.

Упражнение 3.11. Определить, сколько компонент связности имеет граф G , заданный:

а) матрицей смежности

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

б) матрицей инцидентностей

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

3. Мосты и циклы графа. Ребро e графа G называется *мостом*, если $k(G) < k(G - e)$.

Напомним, что $G - e$ - это граф, получающийся из графа G удалением ребра e (концы ребра e при этом не удаляются).

Пример 3. У связного графа G , представленного диаграммой на рис. 3.16, мостами являются ребра $e_1 = v_1v_2$, $e_5 = v_4v_5$, $e_6 = v_5v_6$. Действительно, при удалении каждого из этих ребер получается граф, число связности которого равно 2, что на единицу больше числа связности графа G .

Если же удалить из графа G ребро $e_2 = v_2v_4$ (равно как и $e_3 = v_2v_3$ или $e_4 = v_3v_4$), то получится граф, у которого, так же как у графа G , одна компонента связности.

На рис. 3.17 приведена диаграмма графа $G - e_5$, полученного из графа G удалением ребра $e_5 = v_4v_5$.

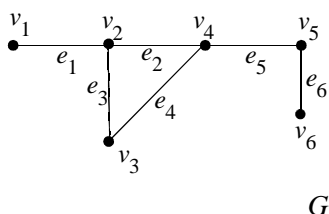


Рис. 3.16.

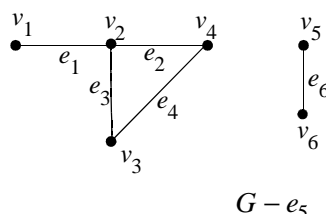


Рис. 3.17.

Далее, если из графа $G - e_5$ удалить ребро e_6 , то получится граф $(G - e_5) - e_6$ (рис. 3.18), число связности которого равно 3, т.е. и в этом случае число компонент связности увеличивается на единицу.

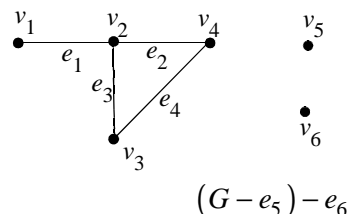


Рис. 3.18.

Перечислим свойства мостов:

1. Если ребро e - мост графа G , то $k(G - e) = k(G) + 1$, т.е. при удалении моста число связности графа увеличивается ровно на единицу (**теорема о мостах**).

2. Ребро графа является мостом тогда и только тогда, когда оно не содержится ни в одном цикле (**теорема о мостах и циклах**).

Доказательства этих теорем приведены во второй части параграфа.

Упражнение 3.12. Сколько ребер графа, диаграмма которого изображена на нижеуказанном рисунке, являются мостами:

а) на рис. 3.1;

б) на рис. 3.2;

в) на рис. 3.3?

Упражнение 3.13. Привести пример связного обыкновенного графа с тремя ребрами:

а) ни одно ребро которого не является мостом;

б) все ребра которого являются мостами.

4. Цикломатическое число графа. Число $\nu(G) = |E| - |V| + k(G)$ называется *цикломатическим числом* графа $G = (V, E)$.

Например, цикломатическое число графа G из примера 2 равно $\nu(G) = 4 - 6 + 3 = 1$, а цикломатические числа его подграфов H_1, H_2, H_3 равны соответственно $\nu(H_1) = 3 - 3 + 1 = 1$, $\nu(H_2) = 1 - 2 + 1 = 0$, $\nu(H_3) = 0 - 1 + 1 = 0$.

Цикломатическое число обладает следующими свойствами:

1) цикломатическое число любого графа неотрицательно, т.е. $\nu(G) \geq 0$ (**теорема о знаке цикломатического числа**);

2) цикломатическое число любого графа равно сумме цикломатических чисел его компонент связности, т.е. если G_1, G_2, \dots, G_k - все компоненты связности графа G , то

$$\nu(G) = \sum_{i=1}^k \nu(G_i).$$

Доказательство этих утверждений приведено во второй части параграфа.

Упражнение 3.14. Найти цикломатическое число графа:

а) K_4 ;

б) K_5 ;

в) $K_{2,3}$;

г) представленного на рис. 3.2.

Теоретические обоснования

Теорема 3.1 (о мостах). Если ребро e - мост графа G , то $k(G - e) = k(G) + 1$.

Доказательство. Доказательство проведем в два этапа: на первом этапе докажем это утверждение для связных графов, на втором - рассмотрим общий случай.

1. Докажем, что если граф G - связный и его ребро e - мост, то $k(G - e) = 2$.

Предположим противное, т.е. что существуют граф G и мост e в нем такие, что $k(G) = 1$ и $k(G - e) \geq 3$. Последнее означает, что в графе $G - e$ найдутся вершины a, b, c , лежащие в разных компонентах связности. Поскольку исходный граф G связан, то на нем существуют пути из a в b и из a в c , а, значит, согласно лемме о простой цепи, существуют простые цепи z_{ab} и z_{ac} .

В графе $G - e$ вершины a, b и c лежат в разных компонентах связности, следовательно, в нем нет путей, соединяющих a с b и a с c , а это означает, что цепи z_{ab} и z_{ac} разорвались при удалении ребра e . Последнее указывает на то, что эти цепи ребро e содержали.

Возможные цепи z_{ab} и z_{ac} таковы:

$$\begin{aligned} 1) \quad z_{ab} &: a \overset{z'_{au}}{\mathbf{K}} u e v \mathbf{K} b; & z_{ac} &: a \overset{z''_{uc}}{\mathbf{K}} v e u \mathbf{K} c; \\ 2) \quad z_{ab} &: a \overset{z'_{vb}}{\mathbf{K}} u e v \mathbf{K} b; & z_{ac} &: a \overset{z''_{vc}}{\mathbf{K}} u e v \mathbf{K} c. \end{aligned}$$

Отметим, что фрагменты цепей, спрятанные за \mathbf{K} , не содержат ребра e и, значит, не пострадают при его удалении.

В первом случае, склеив фрагменты z'_{au} и z''_{uc} цепей z_{ab} и z_{ac} , получим p_{ac} - путь на графе $G - e$. Это противоречит тому, что вершины a и c лежат в разных компонентах связности этого графа.

Во втором случае, склеив инвертированный фрагмент z'_{vb} цепи z_{ab} и фрагмент z''_{vc} цепи z_{ac} , получим p_{bc} - путь на графе $G - e$. А это противоречит тому, что вершины b и c лежат в разных компонентах связности этого графа.

Таким образом, наше предположение, что $k(G - e) \geq 3$ было неверным и $k(G - e) = 2$.

2. Пусть теперь граф G имеет несколько компонент связности. Так как каждое ребро содержится ровно в одной компоненте связности графа G , то при удалении моста e

точно одна компонента связности графа, согласно лемме, распадется на две компоненты связности графа $G - e$. Следовательно, при удалении моста e число компонент связности графа G увеличится ровно на единицу. ■

Теорема 3.2 (о мостах и циклах). *Ребро графа является мостом тогда и только тогда, когда оно не содержится ни в одном цикле.*

Доказательство. Необходимость. Пусть на графе G ребро $e = uv$ - мост. Покажем, что оно не содержится ни в одном цикле. Будем рассуждать от противного.

Предположим, что найдется цикл, в котором содержится мост $e : u \overset{z_{vu}}{\curvearrowright} v$. Пусть a и b - произвольная пара связных вершин графа G . Тогда на этом графе найдется простая цепь из a в b . Есть две возможности: 1) эта цепь не содержит ребра e ; 2) эта цепь ребро e содержит. В первом случае при удалении e цепь не нарушается, так что вершины a и b остаются связными. Во втором случае при удалении ребра e цепь нарушается, но есть возможность заменить участок цепи veu (или uev) на фрагмент (или инвертированный фрагмент) z_{vu} рассматриваемого цикла, в результате чего получится новый путь из a в b , т.е. и в этом случае при удалении ребра e вершины a и b останутся связными. Следовательно, при удалении ребра e число компонент связности графа G не изменится, а это противоречит тому, что e - мост.

Достаточность. Пусть ребро $e = uv$ не содержится ни в одном цикле. Надо показать, что оно является мостом. Будем рассуждать от противного. Предположим, что e - не мост. Тогда компонента связности графа, содержащая ребро e , после удаления этого ребра останется связным подграфом, и, значит, в графе $G - e$ найдется цепь, соединяющая u и v . Склеив эту цепь с ребром e , получим в графе G цикл, содержащий e . Таким образом, пришли к противоречию с условием; следовательно, предположение было неверным и ребро e - мост. ■

Теорема 3.3 (о знаке цикломатического числа). *Цикломатическое число любого графа неотрицательно:*

$$v(G) = |E| - |V| + k(G) \geq 0.$$

Доказательство. Доказательство проведем по принципу математической индукции, взяв в качестве параметра число ребер графа.

Базис индукции. Пусть $|E| = 0$. Тогда $k(G) = |V|$ и $v(G) = 0 - |V| + |V| = 0$.

Индуктивный переход. Предположим, что неравенство $\nu(G) \geq 0$ выполнено для любого графа, у которого число ребер меньше или равно n_0 . Докажем, что тогда оно справедливо и для графа $G = (V, E)$, у которого $|E| = n_0 + 1$.

Удалим произвольное ребро графа G , т.е. перейдем к графу $G - e = (V, E \setminus \{e\})$. Число ребер графа $G - e$ равно $|E \setminus \{e\}| = |E| - 1 = n_0$, следовательно, по предположению индукции, для этого графа выполнено условие $\nu(G - e) \geq 0$, т.е. $|E \setminus \{e\}| - |V| + k(G - e) \geq 0$.

Возможны два случая:

- 1) удаленное ребро - не мост;
- 2) удаленное ребро - мост.

В первом случае $k(G - e) = k(G)$. Следовательно, $|E| - 1 - |V| + k(G) \geq 0$ или $|E| - |V| + k(G) \geq 1$.

Во втором случае по теореме о мостах $k(G - e) = k(G) + 1$. Следовательно, $|E| - 1 - |V| + k(G) + 1 \geq 0$ или $|E| - |V| + k(G) \geq 0$.

В обоих случаях показано, что цикломатическое число графа G неотрицательно. Таким образом, справедливость индуктивного перехода обоснована. ■

Задачи повышенной сложности

3.11. Пусть $G = (V, E)$ - обыкновенный граф, $A(G)$ - матрица смежности этого графа, отвечающая некоторой нумерации вершин v_1, v_2, \dots, v_n .

- а) Доказать, что элемент, стоящий на пересечении i -й строки и j -го столбца матрицы A^k , равен числу путей длины k из v_i в v_j .
- б) Как в случае связного обыкновенного графа по матрице $A(G)$ определить длину кратчайшего пути между вершинами v_i и v_j ($i \neq j$)?

3.12. Доказать, что замкнутый путь нечетной длины содержит простой цикл. Верно ли аналогичное утверждение для замкнутых путей произвольной длины?

3.13. Доказать или опровергнуть следующие утверждения:

- а) объединение любых двух различных цепей, соединяющих две вершины, содержит простой цикл;
- б) объединение двух различных простых цепей, соединяющих две вершины, содержит простой цикл.

3.14. Граф $G = (V, E)$ связан тогда и только тогда, когда для любого разбиения множества V на два подмножества V_1 и V_2 существует ребро графа G , соединяющее некоторую вершину из V_1 с некоторой вершиной из V_2 .

3.15. Показать, что связный граф с n вершинами содержит не менее $n - 1$ ребер.

3.16. Доказать, что в связном графе любые две простые цепи максимальной длины имеют по крайней мере одну общую вершину.

3.17. Показать, что граф с n вершинами и двумя компонентами связности имеет не более $\frac{(n-1)(n-2)}{2}$ ребер.

3.18. Доказать, что обыкновенный граф, имеющий более $\frac{(n-1)(n-2)}{2}$ ребер ($n \geq 2$), является связным.

3.19. Сколько существует попарно-неизоморфных обыкновенных графов с 16 вершинами и 118 ребрами?

3.20. Пусть G_1, G_2, \dots, G_k - все компоненты связности графа G . Доказать, что

$$v(G) = \sum_{i=1}^k v(G_i).$$

§ 3.3. Деревья

Дерево. Лес (ациклический граф). Остовный подграф. Остов. Взвешенный граф. Минимальный остов. Кодирование деревьев.

Базовые понятия и утверждения

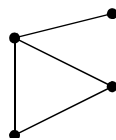
1. Определение и основные свойства деревьев.

Определение. Граф называется *деревом*, если он связный и в нем нет циклов.

Одноэлементный граф, т.е. граф, имеющий одну вершину и не имеющий ребер, также считается деревом.

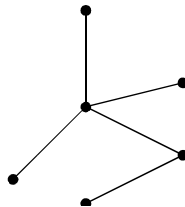
Граф называется *лесом* (или *ациклическим графом*), если в нем нет циклов. Очевидно, что каждая компонента связности леса - дерево.

Пример 1. Граф G_1 (рис. 3.19) не является ни деревом, ни лесом. Граф G_2 (рис. 3.20) - дерево. Граф G_3 (рис. 3.21) - лес, состоящий из четырех деревьев.



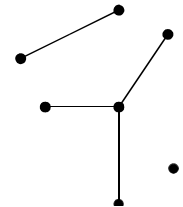
G_1

Рис. 3.19.



G_2

Рис. 3.20.

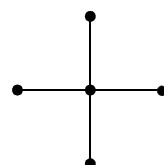
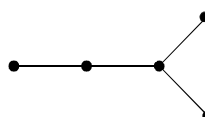
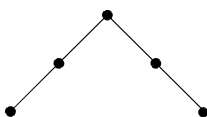


G_3

Рис. 3.21.

Пример 2. Представьте диаграммами все (с точностью до изоморфизма) деревья с пятью вершинами.

◀ Имеется три различных (с точностью изоморфизма) дерева с пятью вершинами (рис. 3.22 - 3.24). ▶



Упражнение 3.15. Представить диаграммами все (с точностью до изоморфизма) деревья с шестью вершинами.

Деревья обладают рядом **характеристических свойств**, по наличию или отсутствию каждого из которых в рассматриваемом графе $G = (V, E)$ можно определить, является граф деревом или нет. Перечислим эти свойства:

- 1) граф G - дерево в том и только в том случае, когда в нем нет циклов и $|E| = |V| - 1$;
- 2) граф G - дерево в том и только в том случае, когда он связный и $|E| = |V| - 1$;
- 3) граф G - дерево в том и только в том случае, когда он связный, и каждое его ребро является мостом;
- 4) граф G - дерево в том и только в том случае, когда любые две вершины графа G можно соединить простой цепью, притом единственной;
- 5) граф G - дерево в том и только в том случае, когда в нем нет циклов и добавление к нему нового ребра приводит к образованию единственного простого цикла.

Также приведем одно из характеристических свойств леса: граф $G = (V, E)$, имеющий k компонент связности, является лесом в том и только в том случае, когда $|E| = |V| - k$.

Упражнение 3.16. Представить диаграммами все леса с тремя ребрами и шестью вершинами.

2. Остовы графа. Подграф \mathcal{G} графа G называется *остовным подграфом*, если множество его вершин совпадает с множеством вершин графа G .

Остовом обыкновенного графа называется его остовный подграф, являющийся деревом.

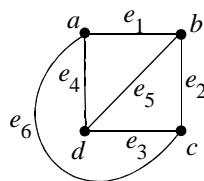
Пусть $G = (V, E)$ - связный граф. Если G содержит хотя бы один цикл, то удалив из графа G некоторое ребро этого цикла, мы уменьшим число циклов графа по крайней мере на единицу, сохранив при этом его связность. Ясно, что, последовательно разрушая циклы данного графа, можно прийти к остову графа. Поскольку дерево с $|V|$ вершинами имеет ровно $|V| - 1$ ребро, то для получения остова нужно удалить из графа $|E| - (|V| - 1)$ ребро, т.е. число ребер, равное цикломатическому числу $\nu(G)$ связного графа G .

Пусть теперь $G = (V, E)$ - произвольный граф с k компонентами связности. Из каждой компоненты связности $G_i = (V_i, E_i)$ этого графа удалим $|E_i| - |V_i| + 1$ ребро так, чтобы получился остов этой компоненты. В результате получим некоторый остовный подграф графа G . Подсчитаем общее число ребер, которое нам пришлось для этого удалить. Сложив равенства $|E_i| - |V_i| + 1, 1 \leq i \leq k$, получим:

$$\sum_{i=1}^k (|E_i| - |V_i| + 1) = \sum_{i=1}^k |E_i| - \sum_{i=1}^k |V_i| + k = |E| - |V| + k = v(G).$$

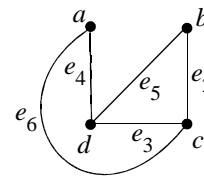
Таким образом, чтобы получить остовный подграф, нужно, последовательно разрушая циклы графа, удалить из него число ребер, равное его цикломатическому числу.

Пример 3. Построим остов графа G , диаграмма которого изображена на рис. 3.25. Удалим из графа G ребро e_1 ; получим граф G_1 (рис. 3.26). Из графа G_1 удалим ребро e_5 ; получим граф G_2 (рис. 3.27). Из графа G_2 удалим ребро e_6 ; получим граф G_3 (рис. 3.28), который является одним из остовов графа G .



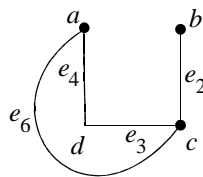
G

Рис. 3.25.



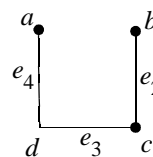
G_1

Рис. 3.26.



G_2

Рис. 3.27.



G_3

Рис. 3.28.

Упражнение 3.17. Построить остовы графов G_1, G_2, G_3, G_4 , изображенных на рис. 3.6.

Пусть G - обыкновенный связный граф. Упорядочим множество его вершин $V = \{v_1, v_2, \dots, v_n\}$. Определим матрицу Кирхгофа $K(G)$ графа G , положив:

$$K(G) = \begin{pmatrix} \deg v_1 & 0 & \mathbf{K} & 0 \\ 0 & \deg v_2 & \mathbf{K} & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ 0 & 0 & \mathbf{K} & \deg v_n \end{pmatrix} - A(G),$$

где $A(G)$ - матрица смежности графа, соответствующая данному упорядочению вершин.

Справедливы следующие утверждения:

1) алгебраические дополнения всех элементов матрицы Кирхгофа графа равны между собой;

2) число остовов в связном неоднородном обыкновенном графе равно алгебраическому дополнению любого элемента его матрицы Кирхгофа.

Доказательство этих утверждений опустим.

Пример 4. Найдём число остовов в полном графе K_3 с помеченными вершинами и представим их диаграммами.

◀ Пометим вершины графа K_3 (рис. 3.29) и выпишем для полученного графа матрицу Кирхгофа:

$$K(K_3) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

Найдём алгебраическое дополнение к элементу матрицы K , стоящему на пересечении первой строки и первого столбца (элемент взят нами произвольно):

$$A^{(1,1)} = (-1)^{1+1} \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} = 3.$$

Таким образом, число помеченных остовов в полном графе с тремя вершинами равно 3.

Их диаграммы изображены на рис. 3.30 - 3.32. ▶

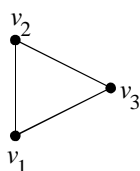


Рис. 3.29.

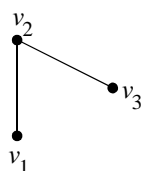


Рис. 3.30.

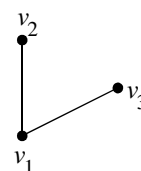


Рис. 3.31.

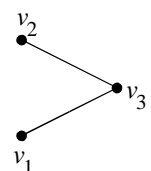


Рис. 3.32.

Упражнение 3.18. Найти число остовов, а также сами остовы в помеченном графе $K_{2,2}$.

3. Построение минимального остова. Взвешенным графом называется граф, на множестве ребер которого задано отображение $\rho: E \rightarrow [0; +\infty)$, приписывающее каждому ребру e неотрицательное число $\rho(e)$.

Число $\rho(e)$ называется *весом* ребра e , число $\rho(G) = \sum_{e \in E} \rho(e)$ - *весом* графа G .

Остов T связного взвешенного графа G называют *минимальным остовом*, если для любого остова T' выполнено неравенство $\rho(T) \leq \rho(T')$.

Опишем **алгоритм построения минимального остова** в связном взвешенном графе, предложенный Дж. Краскалом в 1956 г.

Пусть G - связный взвешенный граф.

0-й шаг. Строим остовный подграф T_0 графа G , множество E_0 ребер которого пусто.

k-й шаг. Пусть T_{k-1} - остовный подграф с множеством ребер $E_{k-1} = \{e_1, e_2, \dots, e_{k-1}\}$, построенный к началу этого шага. Из множества ребер $E \setminus E_{k-1}$ выбираем ребро e_k так, чтобы выполнялись два условия:

- а) добавление ребра e_k не приводит к образованию циклов;
- б) из ребер, удовлетворяющих условию а), ребро e_k обладает наименьшим весом.

Если такого ребра нет, то T_{k-1} - остов минимального веса; если есть, то, добавляя выбранное ребро e_k к остовному подграфу T_{k-1} , получаем остовный подграф T_k с множеством ребер $E_k = E_{k-1} \cup \{e_k\}$.

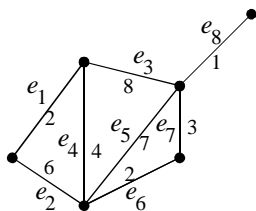
После чего повторяем k -й шаг.

Замечания. 1. В случае несвязного графа, следуя алгоритму Краскала, можно построить остовный лес минимального веса.

2. Если граф невзвешенный, то, присвоив всем ребрам одинаковые веса, по алгоритму Краскала можно построить остов (остовный лес).

Пример 5. Построить остов минимального веса графа G , представленного диаграммой на рис. 3.33 (на диаграмме около каждого ребра проставлен его вес).

◀ Согласно алгоритму Краскала, строим последовательность остовных подграфов:



T_0 с пустым множеством ребер (рис. 3.34);

T_1 с множеством ребер $\{e_8\}$ (рис. 3.35);

T_2 с множеством ребер $\{e_8, e_1\}$ (рис. 3.36);

Рис. 3.33. T_3 с множеством ребер $\{e_8, e_1, e_6\}$ (рис. 3.37);

T_4 с множеством ребер $\{e_8, e_1, e_6, e_7\}$ (рис. 3.38);

T_5 с множеством ребер $\{e_8, e_1, e_6, e_7, e_4\}$

(рис. 3.39).

Добавление к графу T_5 любого из оставшихся ребер графа G ведет к образованию цикла. Таким образом, остовный подграф T_5 с множеством ребер $\{e_8, e_1, e_6, e_7, e_4\}$ - минимальный остов; его вес равен $\rho(T_5) = 1 + 3 + 2 + 4 + 6 = 16$. ►

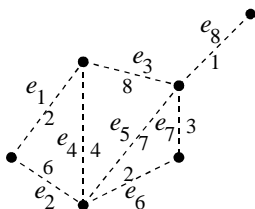


Рис. 3.34.

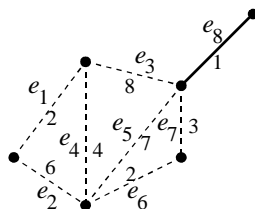


Рис. 3.35.

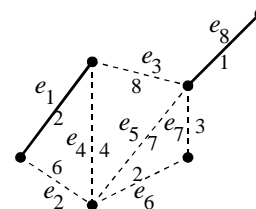


Рис. 3.36.

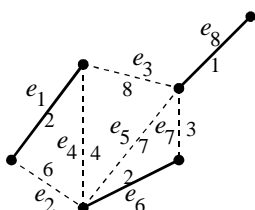


Рис. 3.37.

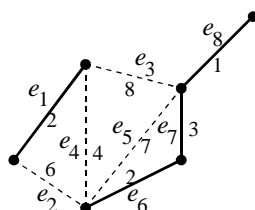


Рис. 3.38.

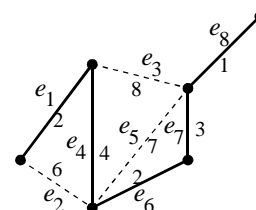


Рис. 3.39.

Упражнение 3.19. Дан граф G с вершинами v_1, v_2, \dots, v_6 , ребрами $e_1 = v_1 v_6$, $e_2 = v_1 v_3$, $e_3 = v_2 v_3$, $e_4 = v_1 v_2$, $e_5 = v_2 v_4$, $e_6 = v_2 v_6$, $e_7 = v_4 v_6$, $e_8 = v_4 v_5$, $e_9 = v_4 v_5$, $e_{10} = v_3 v_4$, $e_{11} = v_3 v_5$ и весовым отображением ρ : $\rho(e_1) = 1$, $\rho(e_2) = 1$, $\rho(e_3) = 1$, $\rho(e_4) = 3$, $\rho(e_5) = 2$, $\rho(e_6) = 2$, $\rho(e_7) = 2$, $\rho(e_8) = 1$, $\rho(e_9) = 3$, $\rho(e_{10}) = 2$, $\rho(e_{11}) = 1$. Построить минимальный остов графа G и вычислить его вес.

4. Кодирование деревьев. Выделим в дереве какую-нибудь одну вершину, которую назовем *корнем*. Полученное дерево с выделенной вершиной называется *корневым*.

Для задания (с точностью до изоморфизма) корневых деревьев используют бинарный код (из 0 и 1), который мы определим индуктивно.

Определение. Бинарным кодом корневого дерева с одним ребром является последовательность (01) . Пусть деревья T_1 и T_2 с корнями a и b соответственно (рис. 3.40) имеют коды α и β . Тогда кодом дерева T_3 с корнем c является код $(0\alpha\beta)$, а кодом дерева T_4 с корнем $c = a = b$ - код $(\alpha\beta)$.

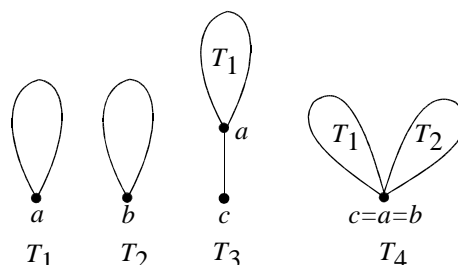


Рис. 3.40.

Пример 6. Построить бинарный код дерева, изображенного на рис. 3.41.

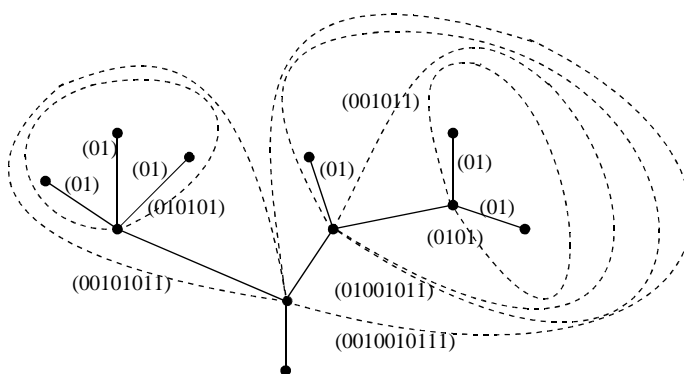


Рис. 3.41.

◀ Этапы построения кода отражены на рис. 3.41. Код дерева - (00010101100100101111) . ▶

Упражнение 3.20. Построить бинарный код дерева, изображенного на рис. 3.42.

Справедливо следующее утверждение: для того чтобы последовательность нулей и единиц являлась кодом некоторого дерева, необходимо и достаточно, чтобы число нулей и единиц в последовательности было одинаковым, причем в любом начальном отрезке последовательности количество нулей было не меньше количества единиц.

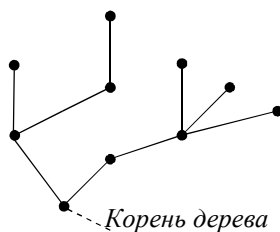


Рис. 3.42.

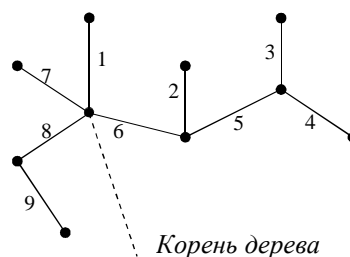


Рис. 3.43.

Например, последовательность (0011101001) не может быть бинарным кодом дерева, поскольку в отрезке 00111 из пяти первых цифр этой последовательности единиц больше, чем нулей.

Чтобы **построить корневое дерево по коду из нулей и единиц**, нужно разбить последовательность на пары 0 и 1, следуя правилу: первая попавшаяся в коде единица образует пару с предшествующим нулем; каждая следующая единица образует пару с ближайшим слева неиспользованным нулем. Если образованные таким образом пары пометить снизу кода фигурными скобками, то каждая такая скобка будет соответствовать ребру графа.

Пример 7. Построить дерево по коду (010010010111010011).

◀Разобьем элементы последовательности на пары: 010010010111010011 и построим

~~0~~1~~0~~01~~0~~01~~0~~111~~0~~10011
 1 2 3 4 5 6 7 8 9
 1 4 2 5 4 3
 6

дерево (рис. 3.43).▶

Упражнение 3.21. Построить дерево по коду (001000111011).

Для задания деревьев с занумерованными вершинами используют *код из натуральных чисел (код Прюфера)*.

Код Прюфера для дерева с занумерованными вершинами строят следующим образом. Находят висячую вершину с наименьшим номером. Записывают номер смежной с ней вершины (это начало кода), после чего удаляют эту висячую вершину вместе с инцидентным ей ребром. Для полученного в результате данной операции дерева находят висячую вершину с наименьшим номером, записывают номер смежной с ней вершины (это продолжение кода), после чего удаляют эту висячую вершину вместе с инцидентным ей ребром. Так поступают до тех пор, пока не останется последнее ребро (его вершины в код не включают). Полученная в результате описанных действий

последовательность и есть код Прюфера. Заметим, что длина этого кода на единицу меньше числа ребер и на две единицы меньше числа вершин дерева.

Пример 8. Построить код Прюфера для дерева, изображенного на рис. 3.44.

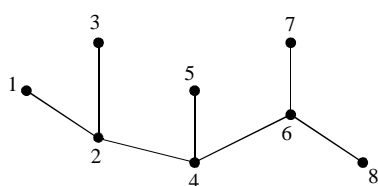


Рис. 3.44.

◀ Применим описанный выше алгоритм:

1-й шаг. У исходного дерева (см. рис. 3.44) висячая вершина с наименьшим номером - это 1. Записываем смежную с ней вершину 2 в начало кода - 2. Удаляем ребро с концами 1, 2 и получаем дерево, изображенное на рис. 3.45,а.

2-й шаг. У дерева на рис. 3.45,а висячая вершина с наименьшим номером - это 3. Записываем смежную с ней вершину 2 в конец строящегося кода - 22. Удаляем ребро с концами 3, 2 и получаем дерево, изображенное на рис. 3.45,б.

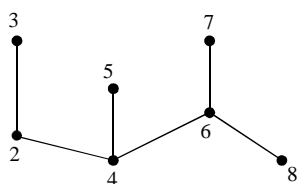
3-й шаг. У дерева на рис. 3.45,б висячая вершина с наименьшим номером - это 2. Записываем смежную с ней вершину 4 в конец строящегося кода - 224. Удаляем ребро с концами 2, 4 и получаем дерево, изображенное на рис. 3.45,в.

4-й шаг. У дерева на рис. 3.45,в висячая вершина с наименьшим номером - это 5. Записываем смежную с ней вершину 4 в конец строящегося кода - 2244. Удаляем ребро с концами 5, 4 и получаем дерево, изображенное на рис. 3.45,г.

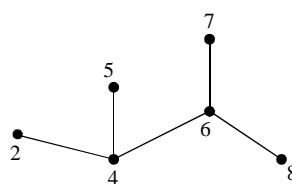
5-й шаг. У дерева на рис. 3.45,г висячая вершина с наименьшим номером - это 4. Записываем смежную с ней вершину 6 в конец строящегося кода - 22446. Удаляем ребро с концами 4, 6 и получаем дерево, изображенное на рис. 3.45,д.

6-й шаг. У дерева на рис. 3.45,д висячая вершина с наименьшим номером - это 7. Записываем смежную с ней вершину 6 в конец строящегося кода - 224466. Удаляем ребро с концами 6, 7 и получаем дерево, изображенное на рис. 3.45,е.

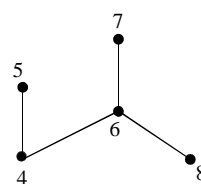
Таким образом, код дерева, изображенного на рис. 3.44 - [2 2 4 4 6 6]. ▶



а



б



в

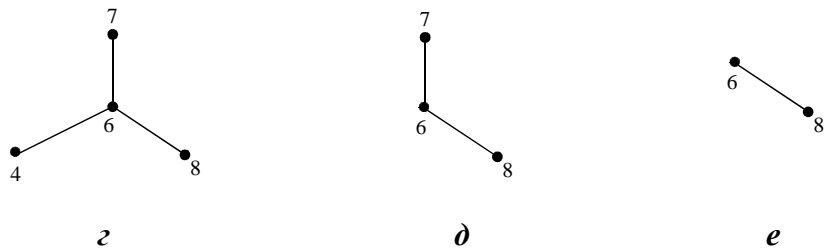


Рис. 3.45.

Упражнение 3.22. Построить код Прюфера для дерева, изображенного на рис. 3.46.

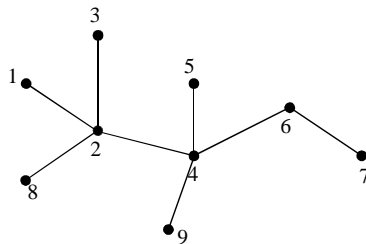


Рис. 3.46.

Построение дерева по коду из натуральных чисел рассмотрим на примере кода $[2\ 2\ 4\ 4\ 6\ 6]$. Прежде всего заметим, что дерево, которое нам предстоит построить, имеет 8 вершин.

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 2 | 4 | 4 | 6 | 6 | Будем преобразовывать последовательность 224466, |
| 1 | 2 | 4 | 4 | 6 | 6 | действуя по следующей схеме. Вместо первого числа |
| 1 | 3 | 4 | 4 | 6 | 6 | запишем наименьшее натуральное число, которое в |
| 1 | 3 | 2 | 4 | 6 | 6 | этой последовательности не встречается, т.е. 1; |
| 1 | 3 | 2 | 5 | 6 | 6 | получим последовательность 124466. Вместо второго |
| 1 | 3 | 2 | 5 | 4 | 6 | числа в новой последовательности запишем |
| 1 | 3 | 2 | 5 | 4 | 6 | наименьшее, не встречающееся в ней, т.е. 3; получим |
| 1 | 3 | 2 | 5 | 4 | 7 | последовательность 134466, и т.д. Действуем так до |
| 2 | 2 | 4 | 4 | 6 | 6 | тех пор, пока все числа в исходной |
| | | | | | | последовательности не будут заменены. Расположим |
| | | | | | | все последовательности друг под другом; под |
| | | | | | | последней из них запишем код дерева (рис. 3.47). |

Рис. 3.47.

Выпишем пары вершин, записанные друг под другом в последних двух строчках: (12), (32), (24), (54), (46), (76). Каждая такая пара - это пара концов одного из ребер дерева. Этот список дополним парой вершин, отсутствующих в предпоследней строчке, т.е. парой (6,8). Теперь построим дерево: отметим на плоскости точки - вершины дерева и соединим их ребрами согласно списку (см. рис. 3.47).

Упражнение 3.23. Построить дерево по коду [2 2 5 5 5 5].

Теоретические обоснования

Теорема 3.4 (о характеристических свойствах деревьев). Для графа $G=(V, E)$ следующие утверждения равносильны:

- 1) граф G - дерево;
- 2) G ациклический и $|E|=|V|-1$;
- 3) G связный и $|E|=|V|-1$;
- 4) G связный, и каждое его ребро является мостом;
- 5) любые две вершины графа G можно соединить простой цепью, притом единственной;
- 6) G ациклический, и добавление к нему нового ребра приводит к образованию единственного простого цикла.

Доказательство. Доказательство проведем по следующей схеме:

$$1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 6 \Rightarrow 1.$$

$1 \Rightarrow 2$. Индукцией по числу ребер проверим, что для любого дерева выполняется равенство $|E|=|V|-1$.

Базис индукции. Пусть $|E|=0$, тогда $|V|=1$, и равенство $|E|=|V|-1$ выполнено.

Индуктивный переход. Предположим, что требуемое равенство выполняется для любого дерева с числом ребер, меньшим либо равным n_0 . Докажем, что оно справедливо и для дерева с числом ребер $n_0 + 1$. Удалим из графа G произвольное ребро e . Согласно свойству мостов (см. теорему о мостах и циклах из параграфа 3.2), ребро e - мост. По теореме о мостах, $k(G - e) = k(G) + 1 = 2$. Следовательно, граф $G - e$ состоит из двух компонент связности $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$, каждая из которых - дерево с числом ребер, меньшим либо равным n_0 . Для каждой компоненты связности справедливо предположение индукции, т.е. выполнены равенства $|E_1|=|V_1|-1$ и $|E_2|=|V_2|-1$. Сложив эти равенства почленно, получим: $|E_1| + |E_2| = |V_1| + |V_2| - 2$. Или $|E| = |V| - 1$.

$2 \Rightarrow 3$. Докажем, что если граф $G=(V, E)$ ациклический и $|E|=|V|-1$, то граф связный. Будем рассуждать от противного, т.е. предположим, что найдется ациклический граф

G , число ребер которого на единицу меньше числа вершин, который связным не является. Пусть $k, k \geq 2$, - число компонент связности графа G . Каждая компонента связности - дерево. Переход $1 \Rightarrow 2$ уже доказан, следовательно, для каждой компоненты связности $G_i = (V_i, E_i) \ (i=1, \dots, k)$ можем записать: $|E_i| = |V_i| - 1$.

Просуммировав по i , получим:

$$\sum_{i=1}^k |E_i| = \sum_{i=1}^k (|V_i| - 1) = \sum_{i=1}^k |V_i| - \sum_{i=1}^k 1,$$

или

$$|E| = |V| - k.$$

Так как $k \geq 2$, то пришли к противоречию с условием $|E| = |V| - 1$. Следовательно, наше предположение было неверным.

$3 \Rightarrow 4$. Докажем, что если граф связный и $|E| = |V| - 1$, то каждое его ребро является мостом. Будем рассуждать от противного. Предположим, что найдется связный граф $G = (V, E)$, у которого $|E| = |V| - 1$ и при этом есть ребро e , не являющееся мостом.

Тогда граф $G - e$ связный и

$$\begin{aligned} v(G - e) &= |E(G - e)| - |V(G - e)| + 1 = \\ &= (|E| - 1) - |V| + 1 = |E| - |V| = -1 < 0. \end{aligned}$$

Получили противоречие с утверждением теоремы о знаке цикломатического числа, значит, наше предположение было неверным.

$4 \Rightarrow 5$. Из связности графа вытекает, что любые две его вершины можно соединить путем, а следовательно, и простой цепью. Докажем, что эта простая цепь единственна. Доказательство проведем от противного. Предположим, что найдется связный граф, все ребра которого - мосты, такой, что в нем есть две вершины a и b , соединенные двумя различными простыми цепями z'_{ab} и z''_{ab} . Поскольку цепи z'_{ab} и z''_{ab} различны, то имеется ребро $e = uv$, входящее в цепь z'_{ab} и не входящее в цепь z''_{ab} .

Пусть z'_{au} и z'_{vb} - фрагменты цепи z'_{ab} : $\overset{z'_{au}}{a \dots u} e \overset{z'_{vb}}{v \dots b}$. Склеим инвертированный фрагмент z'_{au} , цепь z''_{ab} и инвертированный фрагмент z'_{vb} . Получим на графе G путь из u в v , не содержащий ребра e . Из этого маршрута выделим простую цепь из u в

v и склеим ее с цепью veu . В результате получим цикл, содержащий e , а это противоречит тому, что ребро e - мост.

$5 \Rightarrow 6$. Пусть для графа G выполнено условие 5.

Проверим сначала, что граф G не содержит циклов. Будем рассуждать от противного. Предположим, что на графе G имеется цикл. Пусть $e = uv$ - одно из ребер этого цикла. Тогда uev - одна простая цепь, соединяющая вершины u и v , а часть цикла за вычетом ребра e - другая (если $e = uu$ - петля, то имеется также две простые цепи из u в u : ими является цепь из одной вершины u и цепь ueu). Получили противоречие с условием 5.

Покажем, что добавление к графу G нового ребра приводит к образованию цикла, причем только одного. Возьмем на графе G две произвольные вершины a и b и соединим их новым ребром e ; получим граф $G + e$. По условию на графе G имеется единственная простая цепь из a в b . Склеив ее с цепью bea , получим на графе $G + e$ простой цикл. Докажем, что этот цикл единственный. Предположим, что при добавлении ребра e образовалось два простых цикла. Тогда, удалив из каждого из них ребро e , получим на графе G две простые цепи из a в b , а наличие двух таких цепей противоречит условию 5.

$6 \Rightarrow 1$. Будем рассуждать от противного, т.е. предположим, что существует несвязный граф G , для которого выполнено условие 6. Возьмем на этом графе две вершины, лежащие в разных компонентах связности, и соединим их ребром e . В результате образуется граф $G + e$, для которого ребро e является мостом и, следовательно, не содержится ни в одном цикле. Таким образом, добавление ребра e не привело к образованию цикла, что противоречит условию 6. ■

Следствие. Неодноэлементное дерево имеет не менее двух висячих вершин.

Доказательство. Рассмотрим произвольное дерево, имеющее не менее двух вершин. Представим множество его вершин V в виде $V = (V \setminus V_{\text{вис}}) \cup V_{\text{вис}}$, где $V_{\text{вис}}$ - множество висячих вершин этого дерева. Тогда

$$\begin{aligned} 2|E| & \stackrel{\text{Лемма о рукопожатиях}}{=} \sum_{v \in V} \deg v = \sum_{v \in V \setminus V_{\text{вис}}} \deg v + \sum_{v \in V_{\text{вис}}} \deg v = \\ &= \sum_{v \in V \setminus V_{\text{вис}}} \deg v + \sum_{v \in V_{\text{вис}}} 1 = \sum_{v \in V \setminus V_{\text{вис}}} \deg v + |V_{\text{вис}}| \geq \sum_{v \in V \setminus V_{\text{вис}}} 2 + |V_{\text{вис}}| = \end{aligned}$$

$$= 2(|V| - |V_{\text{вис}}|) + |V_{\text{вис}}| = 2|V| - |V_{\text{вис}}|.$$

Но $|E| = |V| - 1$, поэтому $2|V| - 2 \geq 2|V| - |V_{\text{вис}}|$, откуда $|V_{\text{вис}}| \geq 2$. ■

Задачи повышенной сложности

3.21. Опираясь на теорему о характеристических свойствах деревьев, доказать, что граф $G = (V, E)$, имеющий k компонент связности, является лесом в том и только в том случае, когда $|E| = |V| - k$.

3.22. Опираясь на теорему о характеристических свойствах деревьев, доказать, что если степень каждой вершины неориентированного графа не меньше двух, то он содержит цикл.

3.23. Используя теорему Кирхгофа, показать, что число остовов в полном графе K_n с помеченными вершинами равно n^{n-2} .

Замечание. Любой остов в полном графе K_n с помеченными вершинами - это дерево с вершинами $1, 2, 3, \dots, n$, а совокупность всех таких остовов - это совокупность всех деревьев с помеченными вершинами. Отсюда заключаем: *число деревьев с n помеченными вершинами равно n^{n-2} (теорема Кэли).*

§ 3.4. Планарность

Укладки графа в пространстве и на плоскости. Планарные графы. Плоские графы. Грани плоского графа. Связь между числом вершин, ребер и граней плоского графа, формула Эйлера. Критерии планарности.

Базовые понятия и утверждения

1. Укладка графа в трехмерном пространстве. Говорят, что *граф обладает укладкой* в пространстве, если в трехмерном евклидовом пространстве можно построить такую диаграмму графа, в которой ребра не имеют общих точек, отличных от концевых. Сама диаграмма, обладающая указанным свойством, называется при этом *укладкой графа* в пространстве.

Пример 1. Рассмотрим граф G с вершинами a, b, c, d и ребрами $e_1 = ab, e_2 = bc, e_3 = ca, e_4 = ad, e_5 = bd, e_6 = cd$.

Укладка данного графа в пространстве изображена на рис. 3.48.

Утверждение. Любой граф обладает укладкой в пространстве.

Доказательство. Покажем, как в трехмерном евклидовом пространстве построить укладку для произвольного графа.

Возьмем в пространстве любую прямую l (рис. 3.49). Вершинам графа поставим в соответствие точки прямой l так, чтобы каждой вершине соответствовала своя точка; каждую точку пометим именем сопоставленной ей вершины.

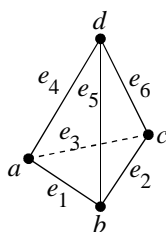


Рис.
3.48.

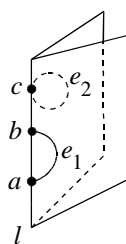


Рис. 3.49.

Через прямую l проведем столько плоскостей, сколько ребер у графа (каждому ребру будет соответствовать своя плоскость). Если ребро $e_1 = ab$ не является петлей, то изобразим его в соответствующей ему плоскости в виде полуокружности с диаметром

$[a, b]$. Если ребро $e_2 = cc$ - петля, то изобразим его в соответствующей ему плоскости в виде окружности, которая касается прямой l в точке c .

Очевидно, построенная таким образом диаграмма будет укладкой графа в трехмерном пространстве. ■

2. Планарные графы. Говорят, что *граф обладает укладкой* на плоскости, если в двумерном евклидовом пространстве существует такая диаграмма графа, в которой ребра не имеют общих точек, отличных от концевых.

На рис. 3.50 изображена укладка на плоскости графа из примера 1.

Далее мы убедимся в том, что не все графы имеют плоскую укладку. Граф, обладающий укладкой на плоскости, называют *планарным*. Любую укладку планарного графа на плоскости называют *плоским графом*.

Например, граф из примера 1 - планарный, а его диаграмма на рис. 3.50 - плоский граф.

Совокупность ребер каждого простого цикла плоского графа может рассматриваться как граница некоторой области на плоскости. Те из этих областей, которые не содержат внутри себя ребер других простых циклов данного плоского графа, называют *гранями* плоского графа.

Отметим, что одна из граней любого плоского графа не ограничена. Ее называют *внешней гранью*, остальные грани называют *внутренними* гранями.

Например, плоский граф, изображенный на рис. 3.51, имеет четыре грани: три внутренние (S_1, S_2, S_3) и одну внешнюю (S_4). Любое дерево имеет только одну грань - внешнюю.

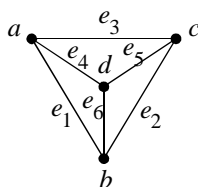


Рис. 3.50.

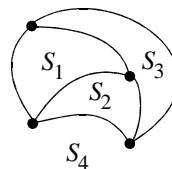


Рис. 3.51.

Справедливо следующее утверждение.

Пусть $G = (V, E)$ - связный плоский граф, S - множество его граней. Тогда

$|S| = |E| - |V| + 2$ (формула Эйлера).

Доказательство утверждения приведено во второй части параграфа.

Чтобы доказать планарность графа, достаточно нарисовать его плоскую укладку. Непланарность графов обосновывать сложнее (ссылки на тщетность попыток нарисовать плоскую укладку графа доказательством не являются). В некоторых случаях непланарность удается доказать с помощью формулы Эйлера.

Пример 2. Доказать, что полный граф K_5 непланарен.

◀ Будем рассуждать от противного. Предположим, что граф K_5 (см. рис. 3.7) планарен. Тогда у него есть укладка на плоскости. Пусть S - множество граней этой укладки и граница i -й грани состоит из l_i ребер, где $i=1, \dots, |S|$. Поскольку каждое ребро графа K_5 содержится в некотором цикле и, следовательно, лежит на границе двух граней, то

$$l_1 + l_2 + \dots + l_{|S|} = 2|E|.$$

Граф K_5 не имеет петель и кратных ребер, значит, каждая грань его укладки граничит не менее чем с тремя ребрами и $l_i \geq 3$. Следовательно, $3|S| \leq 2|E|$, т.е. $3|S| \leq 2|E|$.

Граф K_5 имеет 5 вершин и 10 ребер, следовательно, по формуле Эйлера

$|S| = |E| - |V| + 2 = 10 - 5 + 2 = 7$. Значит, должно выполняться неравенство $21 \leq 20$, что неверно. Пришли к противоречию. Следовательно, предположение о планарности графа K_5 было неверным. ►

Пример 3. Доказать, что полный двудольный граф $K_{3,3}$ непланарен.

◀ Будем рассуждать от противного. Предположим, что граф $K_{3,3}$ (см. рис. 3.8) планарен. Пусть S - множество граней этой укладки и граница i -й грани состоит из l_i ребер, где $i=1, \dots, |S|$. Поскольку каждое ребро графа $K_{3,3}$ содержится в некотором цикле и, следовательно, лежит на границе двух граней, то

$$l_1 + l_2 + \dots + l_{|S|} = 2|E|.$$

Несложно убедиться, что в графе $K_{3,3}$ все простые циклы имеют длину, не меньшую 4, т.е. $l_i \geq 4$ ($i=1, \dots, |S|$), и, значит, $4|S| \leq 2|E|$, откуда $2|S| \leq |E|$. С учетом того, что граф $K_{3,3}$ имеет 6 вершин и 9 ребер по формуле Эйлера получим $|S| = |E| - |V| + 2 = 9 - 6 + 2 = 5$, значит, должно выполняться неравенство $2 \cdot 5 \leq 9$. Пришли к противоречию. Следовательно, предположение о планарности графа $K_{3,3}$ было неверным. ►

Упражнение 3.24. а) Одной из диаграмм графа является трехмерный куб (рис. 3.52,а). Планарен ли данный граф?

б) Одной из диаграмм графа является трехмерный куб, в котором проведена диагональ (рис. 3.52,б). Планарен ли данный граф?

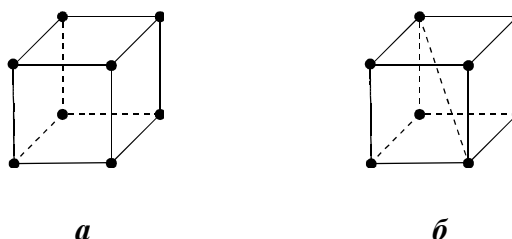


Рис. 3.52.

Упражнение 3.25. а) Одной из диаграмм графа является октаэдр (рис. 3.53,а). Планарен ли данный граф?

б) Одной из диаграмм графа является октаэдр, в котором проведена диагональ (рис. 3.53,б). Планарен ли данный граф?

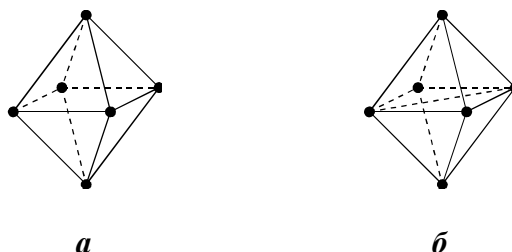


Рис. 3.53.

Теоретические обоснования

Теорема 3.5 (о плоских графах). Пусть $G=(V, E)$ - плоский граф, S - множество его граней. Тогда

$$|E| - |V| + k(G) = |S| - 1.$$

Доказательство. Доказательство проведем по индукции, взяв в качестве параметра число ребер графа.

Базис индукции. Пусть $|E|=0$, т.е. граф не имеет ребер. Тогда $|V|=k(G)$, $|S|=1$, поэтому доказываемая формула справедлива.

Индуктивный переход. Предположим, что формула верна для любого графа с числом ребер, меньшим либо равным n_0 . Покажем, что она остается в силе для графа G , число ребер которого равно $n_0 + 1$. Возьмем какое-нибудь ребро e графа G и удалим его. Получим граф $G - e$, число ребер которого равно n_0 . Тогда для плоской укладки графа G справедливо предположение индукции и, значит, выполняется равенство

$$|E(G - e)| - |V(G - e)| + |k(G - e)| = |S(G - e)| - 1.$$

Рассмотрим два случая: 1) удаленное ребро - не мост и 2) удаленное ребро - мост.

1. Пусть удаленное ребро e - не мост. Тогда e содержится в некотором цикле графа G и поэтому лежит на границе двух граней его плоской укладки. Если удалить из графа ребро e , то эти две грани сольются в одну и число граней в плоской укладке графа $G - e$ окажется на единицу меньшим числа граней в плоской укладке графа G , т.е.
 $|S(G - e)| = |S(G)| - 1.$

Из того, что e - не мост, следует также, что $|k(G - e)| = |k(G)|$. Кроме того,
 $|V(G - e)| = |V(G)|, |E(G - e)| = |E(G)| - 1.$

Следовательно, равенство

$$|E(G - e)| - |V(G - e)| + |k(G - e)| = |S(G - e)| - 1$$

можно переписать в виде

$$(|E(G)| - 1) - |V(G)| + |k(G)| = (|S(G)| - 1) - 1$$

или

$$|E(G)| - |V(G)| + |k(G)| = |S(G)| - 1.$$

2. Пусть теперь удаленное ребро e - мост. Тогда e не содержится ни в одном цикле графа G и поэтому не входит в границы граней плоской укладки этого графа. Значит, ни одна из граней графа G не пострадает при его удалении, так что $|S(G - e)| = |S(G)|$. Из того, что e - мост следует также, что $|k(G - e)| = |k(G)| + 1$. Кроме того,
 $|E(G - e)| = |E(G)| - 1, |V(G - e)| = |V(G)|.$

Следовательно, равенство

$$|E(G - e)| - |V(G - e)| + |k(G - e)| = |S(G - e)| - 1$$

можно переписать в виде

$$(|E(G)| - 1) - |V(G)| + (|k(G)| + 1) = |S(G)| - 1,$$

или

$$|E(G)| - |V(G)| + |k(G)| = |S(G)| - 1.$$

Таким образом, в обоих случаях индуктивный переход доказан. ■

Заметим, что поскольку $|E| - |V| + k(G) = v(G)$, то из теоремы следует, что цикломатическое число плоского графа равно числу его внутренних граней.

Следствие. В частном случае связного плоского графа из равенства

$$|E| - |V| + k(G) = |S| - 1 \text{ получаем формулу Эйлера } |S| = |E| - |V| + 2.$$

Будем говорить, что граф G' получен из графа G *включением вершины степени 2*, если в графе G одно из ребер $e = ab$ (равенство $a = b$ не исключается) заменено на два новых ребра $e_1 = ac$ и $e_2 = cb$, где c - новая вершина степени 2, а остальные вершины и ребра остались без изменения.

Операция, обратная операции включения вершины степени 2, называется *исключением вершины степени 2*.

Будем говорить, что граф G_1 *гомеоморфен* графу G_2 , если G_1 изоморфен некоторому графу, который можно получить из G_2 путем применения конечное число раз операций включения и исключения вершин степени 2.

Пример 4. Графы G_1 , G_2 и G_3 (рис. 3.54) - попарно гомеоморфны. Действительно, граф G_1 можно получить из графа G_2 путем исключения вершины b (и наоборот, граф G_2 можно получить из графа G_1 путем включения вершины b).

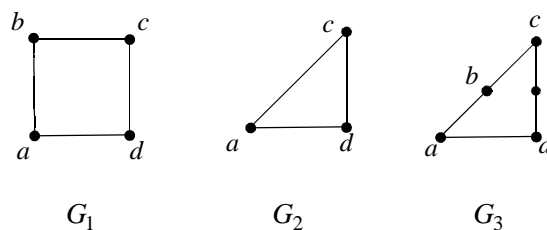


Рис. 3.54.

Граф G_1 можно получить из графа G_3 путем включения вершины t (и наоборот, граф G_3 можно получить из графа G_1 путем исключения вершины t). Граф G_2 можно

получить из графа G_3 путем включения вершин b и t (и наоборот, граф G_3 можно получить из графа G_2 путем исключения вершин b и t).

Приведем без доказательства один из критериев планарности.

Теорема 3.6 (Понтрягина и Куратовского). *Граф G планарен тогда и только тогда, когда у него нет подграфов, гомеоморфных K_5 , и нет подграфов, гомеоморфных $K_{3,3}$.*

Пример 5. На рис. 3.55 изображены граф Петерсена G и его подграф H .

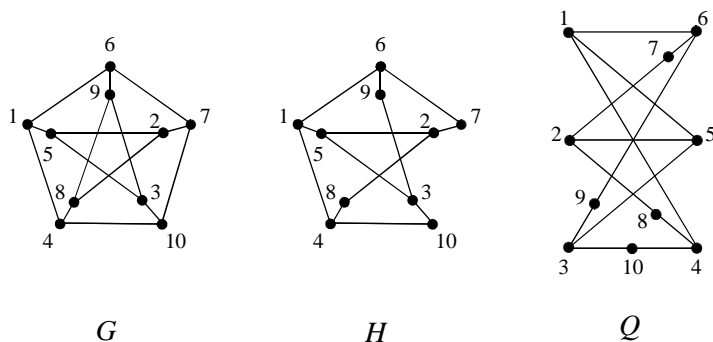


Рис. 3.55.

Граф H гомеоморфен графу $K_{3,3}$. Это следует из того, что граф H изоморфен графу Q (на рис. 3.55 соответствующие вершины помечены одинаковыми числами), а граф Q гомеоморфен графу $K_{3,3}$. Таким образом, граф Петерсена непланарен. Заметим, что при этом граф Петерсена не содержит подграфов, гомеоморфных K_5 .

Теперь определим на графе операцию стягивания ребра. Пусть $e = ab$ - ребро графа G , не являющееся петлей. Последовательно выполним следующие действия: отождествим вершины a и b , отождествим возникшие при этом кратные ребра и отбросим все возникшие при этом петли. Обозначим полученный граф G' . Будем говорить, что граф G' получен из графа G *стягиванием ребра e* .

Пример 6. На рис. 3.56 показан пример применения операции стягивания ребра: граф G' получен из графа G стягиванием ребра e .

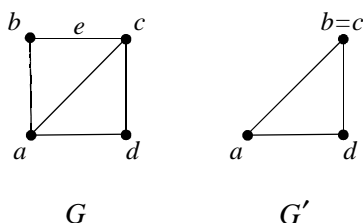


Рис. 3.56.

Будем говорить, что граф G_1 *стягиваем* к графу G_2 , если G_2 можно получить из G_1 с помощью конечного числа применений операции стягивания ребер.

Приведем без доказательства еще один критерий планарности.

Теорема 3.7 (Вагнера, Харари и Татта). *Граф G планарен тогда и только тогда, когда у него нет подграфов, стягиваемых к K_5 , и нет подграфов, стягиваемых к $K_{3,3}$.*

Пример 7. Граф Петерсена непланарен, поскольку он стягиваем к графу K_5 .

Действительно, K_5 получается из графа Петерсена путем последовательного стягивания ребер с концами 1 и 5, 6 и 9, 2 и 7, 4 и 8, 3 и 10 (см. рис. 3.55).

Задачи повышенной сложности

3.24. Рассмотрим на множестве графов бинарное отношение гомеоморфизма (пара графов G_1 и G_2 связана этим отношением, если G_1 гомеоморфен G_2). Какими свойствами это бинарное отношение обладает?

3.25. Какие из графов, изображенных на рис. 3.57 - 3.59, являются планарными?

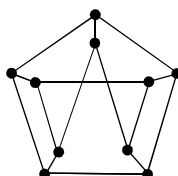


Рис. 3.57.

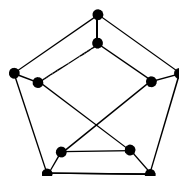


Рис. 3.58.

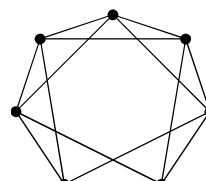


Рис. 3.59.

3.26. При каких n граф, изображенный на рис. 3.60, является планарным?

3.27. При каких n граф, изображенный на рис. 3.61, является планарным?

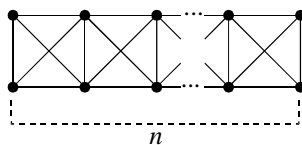


Рис. 3.60.

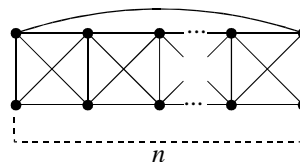


Рис. 3.61.

3.28. При каких m и n граф $K_{m,n}$ является планарным?

3.29. Доказать, что граф B_4 непланарен.

§ 3.5. Обходы графов

Эйлеров цикл. Эйлеров граф. Эйлерова цепь. Гамильтонов цикл. Гамильтонов граф. Гамильтонова цепь.

Базовые понятия и утверждения

1. Эйлеров цикл и эйлерова цепь. В 1736 г. Леонард Эйлер опубликовал решение широко известной в то время задачи о кенигсбергских мостах. Задача эта состояла в следующем. На реке Прегель в Кенигсберге было два острова, соединенных между собой и с берегами семью мостами, как показано на рис. 3.62. Спрашивалось, можно ли, начиная с некоторого места суши, обойти все мосты по одному разу и вернуться назад?

Эмпирические попытки найти такой обход оканчиваются неудачей. Вклад Эйлера в решение задачи состоит в теоретическом обосновании невозможности такого обхода. Для доказательства того, что задача не имеет решения, Эйлер обозначил каждую часть суши точкой, а каждый мост - линией, соединяющей соответствующие точки. Получился рисунок, который мы называем диаграммой графа (рис. 3.63).

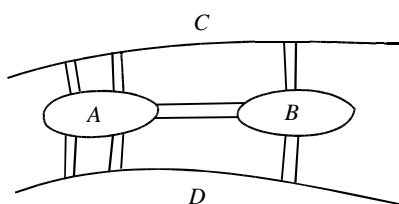


Рис. 3.62.

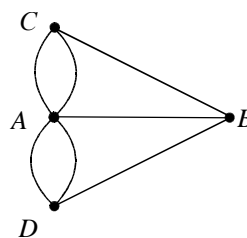


Рис. 3.63.

Используя современную терминологию, можно сказать, что Эйлер свел задачу об обходе мостов к поиску на данном графе цикла, содержащего все вершины и все ребра графа. Отправляясь от этого частного случая, Эйлер обобщил постановку задачи и доказал, что критерием существования такого рода цикла на произвольном связном графе является четность степеней всех его вершин.

Цикл на графе, содержащий все вершины и все ребра графа, стали называть *эйлеровым циклом*, а граф, на котором имеется эйлеровый цикл, - *эйлеровым графом*.

Цепь на графе (как замкнутая, так и незамкнутая), содержащая все вершины и все ребра графа, называют *эйлеровой цепью* (замкнутая эйлерова цепь есть не что иное, как эйлеров цикл).

Пример 1. Граф на рис. 3.64 - эйлеров, поскольку содержит эйлеров цикл $be_4de_1te_7ae_2ae_6be_3ce_5b$.

Пример 2. Граф на рис. 3.65 содержит эйлерову цепь $ae_2be_4de_1te_3fe_6be_5c$.

Пример 3. Граф на рис. 3.66, очевидно не содержит ни одну эйлерову цепь и, в частности, не содержит ни одного эйлерова цикла.

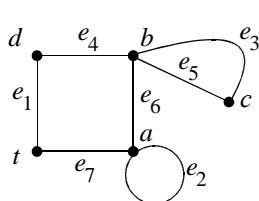


Рис. 3.64.

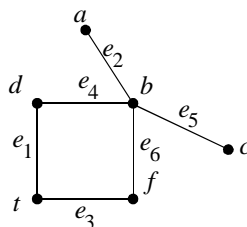


Рис. 3.65.

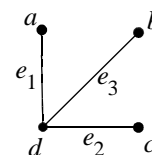


Рис. 3.66.

Следующие два утверждения являются **критериями существования эйлерова цикла и эйлеровой цепи** на связном графе:

- 1) *связный граф содержит эйлеров цикл тогда и только тогда, когда каждая его вершина имеет четную степень;*
- 2) *связный граф содержит эйлерову цепь тогда и только тогда, когда он имеет не более двух вершин нечетной степени.*

Доказательство этих утверждений приведены во второй части параграфа.

Опишем **алгоритм построения эйлерова цикла** на графе.

Пусть G - связный граф, все вершины которого имеют четную степень.

1-й шаг. Произвольно выбираем некоторую вершину v_1 и инцидентное ей ребро.

Этому ребру присваиваем номер 1. Переходим по этому ребру в вершину v_2 , после чего ребро удаляем.

k-й шаг. Пусть к началу этого шага мы находимся в вершине v_k . Выбираем инцидентное этой вершине ребро, соблюдая следующие правила:

- 1) находясь в вершине v_k , не следует выбирать ребро, соединяющее v_k с v_1 , если имеется возможность иного выбора;
- 2) находясь в вершине v_k , не следует выбирать ребро, которое является перешейком, т.е. таким ребром, при удалении которого граф, образованный оставшимися ребрами, распадается на ненулевые компоненты связности.

Выбранному ребру присваиваем номер k , проходим по нему из вершины v_k в смежную с ней вершину v_{k+1} , после чего ребро с номером k удаляем. Если после этого все ребра в графе оказались удаленными, то эйлеров цикл построен (порядок нумерации ребер соответствует последовательности их обхода в графе G).

В противном случае удаляем ребро с номером k , увеличиваем k на единицу и повторяем k -й шаг.

Пример 4. На рис. 3.67 изображен связный граф, степени всех вершин которого четны. Ниже приводится эйлеров цикл, построенный изложенным выше алгоритмом:

$$v_1 1 v_2 2 v_3 3 v_6 4 v_7 5 v_{10} 6 v_8 7 v_6 8 v_5 9 v_2 10 v_4 11 v_5 12 v_8 13 v_9 14 v_{10} 15 v_{11} 16 v_7 17 v_3 18 v_1.$$

Пример 5. На рис. 3.68 изображен связный граф, у которого ровно две вершины v_4 и v_8 имеют нечетную степень (ребро e графу не принадлежит). Соединим вершины v_4 и v_8 , имеющие нечетную степень, ребром e и построим на полученном графе эйлеров цикл, начав его с этого ребра:

$$v_4 e v_8 1 v_9 2 v_7 3 v_3 4 v_1 5 v_2 6 v_5 7 v_4 8 v_3 9 v_2 10 v_4 11 v_6 12 v_8 13 v_7 14 v_4.$$

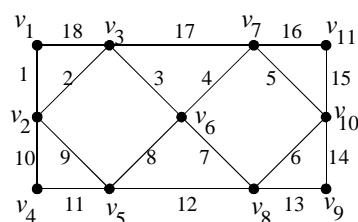


Рис. 3.67.

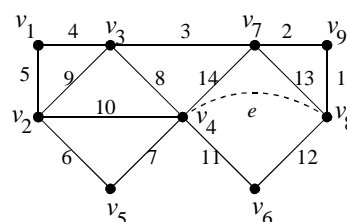


Рис. 3.68.

Удалив из построенного эйлерова цикла первую вершину и ребро ($v_4 e$), получим на исходном графе эйлерову цепь:

$$v_8 1 v_9 2 v_7 3 v_3 4 v_1 5 v_2 6 v_5 7 v_4 8 v_3 9 v_2 10 v_4 11 v_6 12 v_8 13 v_7 14 v_4.$$

Упражнение 3.26. Построить эйлеров цикл на графе, изображенном на рис. 3.69.

Упражнение 3.27. Построить эйлерову цепь на графе, изображенном на рис. 3.70.

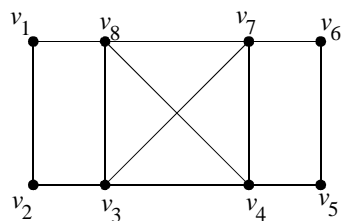


Рис. 3.69.

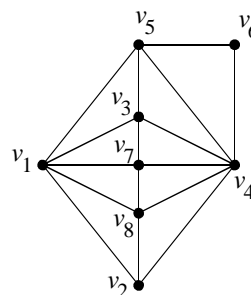


Рис. 3.70.

2. Гамильтонов цикл и гамильтонова цепь. Гамильтоновой цепью графа называется такая его простая цепь, которая проходит через каждую вершину ровно один раз. Замкнутая гамильтонова цепь называется *гамильтоновым циклом*. Граф называется *гамильтоновым*, если он обладает гамильтоновым циклом.

Гамильтонов цикл не обязан проходить по всем ребрам графа.

Простого и эффективного критерия существования гамильтонова цикла на графе не найдено. Однако известно несколько необходимых и достаточных условий существования гамильтоновых циклов [4].

Упражнение 3.28. Показать, что на следующих графах есть гамильтоновы циклы:

- а) K_5 ; б) $K_{3,3}$.

Упражнение 3.29. а) Привести пример графа, который является эйлеровым, но не гамильтоновым.

б) Привести пример графа, который является гамильтоновым, но не эйлеровым.

в) Привести пример графа, который является как эйлеровым, так и гамильтоновым.

г) Привести пример графа, который не является ни гамильтоновым, ни эйлеровым.

Теоретические обоснования

Теорема 3.8 (об эйлеровых циклах). Для неоднородного связного графа G следующие условия эквивалентны:

- 1) G - эйлеров граф;
- 2) каждая вершина графа G имеет четную степень;
- 3) множество всех ребер графа можно разбить на циклы.

Доказательство. Доказательство проведем по схеме: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$.

$1 \Rightarrow 2$. Пусть $C: v_0 e_1 v_1 \dots v_{k-1} e_k v_k$ - эйлеров цикл на графе и a - произвольная вершина графа. Поскольку цикл C - эйлеров, то вершина a встречается в цикле C по меньшей мере один раз. Двигаясь по циклу C , будем подсчитывать число ребер, идущих в цикле перед ней и после нее. Это число равно степени вершины a , поскольку цикл C содержит каждое ребро графа, причем в единственном числе. В общем случае вершина a может совпадать как с внутренними, так и с концевыми вершинами цикла C . Прохождение каждой совпадающей с a внутренней вершины цикла C вносит вклад, равный 2, в степень этой вершины. Если вершина a совпадает с началом и концом цикла, то первое и последнее ребра цикла C совместно вносят в ее степень вклад 2. Следовательно, степени всех вершин графа - четные.

$2 \Rightarrow 3$. Докажем, что если каждая вершина связного графа имеет четную степень, то множество всех ребер графа можно разбить на циклы. Доказательство проведем индукцией по числу ребер графа.

Базис индукции. Пусть $|E|=1$, т.е. граф имеет одно ребро (обозначим его e). Концы этого ребра имеют четную степень, только если это ребро - петля и совпадающие концы этого ребра - единственная вершина графа (обозначим ее a). Следовательно, цикл aea - искомое разбиение.

Индуктивный переход. Предположим, что утверждение верно для любого связного графа с числом ребер, меньшим либо равным n_0 . Покажем, что оно остается в силе для графа G , число ребер которого равно $n_0 + 1$.

Граф G - не дерево (в противном случае у него было бы не менее двух висячих вершин, а это противоречит условию о четности степеней всех вершин графа). Следовательно, на графе G имеется цикл; обозначим его C . Если цикл C проходит через все ребра графа G , то множество его ребер есть искомое разбиение, и индуктивный переход доказан. В противном случае рассмотрим граф G_c , полученный из G удалением ребер цикла C . Каждая компонента связности графа G_c - либо изолированная вершина, либо связный ненулевой граф с вершинами четных степеней и числом ребер, меньшим либо равным n_0 . Значит, по предположению индукции на каждой ненулевой компоненте связности множество всех ребер можно разбить на циклы. Множества ребер этих циклов в совокупности с множеством ребер цикла C дадут разбиение на циклы множества ребер исходного графа G .

$3 \Rightarrow 1$. Разобьем множество всех ребер графа на наименьшее число циклов C_1, C_2, \dots, C_s (такое разбиение существует в силу условия 3 и конечности множества ребер графа G) и докажем, что $s = 1$. Будем рассуждать от противного: предположим, что $s > 1$. В силу связности графа G найдется такое $i \geq 2$, что циклы C_1 и C_i имеют общую вершину. Поскольку C_1 и C_i не имеют общих ребер, их можно объединить в один цикл, уменьшив таким образом общее число s циклов. Получили противоречие с тем, что исходное разбиение наименьшее по числу циклов. Следовательно, все ребра графа G принадлежат некоторому циклу, т.е. граф является эйлеровым. ■

Теорема 3.9 (об эйлеровых цепях). *Связный граф содержит эйлерову цепь тогда и только тогда, когда он имеет не более двух вершин нечетной степени.*

Доказательство. Необходимость. Пусть на связном графе G есть эйлерова цепь Z . Возможны два случая: а) цепь Z замкнута и б) цепь Z не замкнута. Рассмотрим их.

а) Пусть цепь Z замкнута, тогда по теореме об эйлеровых циклах степени всех вершин графа четны и, следовательно, верно, что граф содержит не более двух вершин нечетной степени.

б) Пусть цепь Z не замкнута и вершина a - ее начало, вершина b - конец. Добавим к графу G новое ребро $e = ab$. Склеив Z с ребром e , получим на связном неоднородном графе $G + e$ эйлеров цикл. Следовательно, граф $G + e$ - эйлеров и степени всех его вершин четны. Удаление из графа $G + e$ ребра e уменьшает на единицу степени вершин a и b (они становятся нечетными) и не меняет степени остальных вершин. Значит, на графе G две вершины (a и b) имеют нечетные степени, а остальные - четные.

Достаточность. Пусть связный граф G содержит не более двух вершин нечетной степени. Возможны два случая: а) число вершин нечетной степени графа G равно нулю; б) число вершин нечетной степени графа G равно двум. Рассмотрим их.

а) Если на графе G нет вершин нечетной степени, то граф G - эйлеров, а значит, на нем есть эйлеров цикл, который и является эйлеровой цепью.

б) Пусть граф G имеет ровно две вершины нечетной степени a и b . Рассмотрим граф G_1 , полученный из G добавлением нового ребра $e = ab$. Граф G_1 связан, неоднороден и степень каждой его вершины - четное число. Поэтому в G_1 существует эйлеров цикл C . Удалив из C ребро e , получим на графе G эйлерову цепь z_{ab} . ■

Задачи повышенной сложности

- 3.30.** Можно ли прогуляться по парку и его окрестностям (план парка изображен на рис. 3.67) так, чтобы при этом перелезть через каждый его забор ровно один раз?
- 3.31.** Есть ли гамильтонов цикл на графе B_n ? Ответ обосновать.

§ 3.6. Раскраска графов

Раскраска графа. k -раскрашиваемый граф. k -хроматический граф.
Хроматическое число графа.

Базовые понятия и утверждения

Пусть $G=(V, E)$ - произвольный граф и $\{c_1, c_2, \dots, c_k\}$ - некоторое множество, элементы которого будем называть красками.

Определение. Раскраской графа G в k цветов называется отображение

$p: V \rightarrow \{c_1, c_2, \dots, c_k\}$ такое, что для любых двух смежных вершин a и b выполняется условие $p(a) \neq p(b)$.

Если раскраска графа задана, то говорят, что вершина a имеет цвет $p(a)$.

Отметим, что здесь не предполагается, что P отображает V на все множество красок $\{c_1, c_2, \dots, c_k\}$ (т.е. какие-то краски могут оказаться неиспользованными).

Граф называется k -раскрашиваемым, если он может быть раскрашен в k цветов. Если при этом его нельзя раскрасить в $k-1$ цвет, то он называется k -хроматическим. Число k в таком случае называют хроматическим числом графа и обозначают $\chi(G)$.

Пример 1. Рассмотрим граф, изображенный на рис. 3.71. Поскольку в графе есть смежные вершины, то раскрасить его в один цвет нельзя. На рис. 3.71 дан пример раскраски графа в два цвета: вершины a и c красим в красный цвет, а вершины b и d - в оранжевый. Таким образом, для данного графа хроматическое число $\chi(G)=2$.

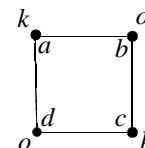


Рис. 3.71.

Приведем ряд утверждений о хроматических числах графов:

- 1) 1-хроматические графы - это нулевые графы и только они;
- 2) 2-хроматические графы - это ненулевые двудольные графы и только они (заметьте, что 2-хроматические графы также называют бихроматическими);
- 3) ненулевой обыкновенный граф является бихроматическим тогда и только тогда, когда не содержит циклов нечетной длины;
- 4) любой планарный граф можно раскрасить не более чем в четыре цвета.

Заметим, что из утверждения 2 следует, что любое неоднородное дерево является бихроматическим графом.

Упражнение 3.30. Построить обыкновенный плоский граф с минимальным числом вершин, хроматическое число которого равно 4.

Упражнение 3.31. Для следующих графов найти хроматические числа и привести пример раскраски графов в соответствующее число цветов:

- а) K_4 ; б) K_5 ; в) $K_{2,3}$;
г) $K_{3,3}$; д) граф Петерсена (см. рис. 3.55).

Теоретические обоснования

Теорема 3.10 (критерий бихроматичности). Пусть G - ненулевой обыкновенный граф. Тогда следующие условия эквивалентны:

- 1) граф G бихроматический;
- 2) граф G не имеет циклов нечетной длины;
- 3) граф G - двудольный.

Доказательство. Доказательство проведем по следующей схеме: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$.

$1 \Rightarrow 2$. Будем рассуждать от противного. Предположим, что граф G бихроматический и на нем имеется цикл нечетной длины: $C: v_0 e_1 v_1 \dots v_{2m-2} e_{2m-1} v_{2m-1} (v_0 = v_{2m-1})$.

Зафиксируем некоторую раскраску графа в два цвета. Так как концы каждого ребра графа имеют разный цвет, то вершины цикла C , совпадающие с членами

последовательности $\{v_i\}_{i=1}^{2m-1}$ с нечетными номерами, окрашены в один цвет.

Следовательно, вершины v_1 и v_{2m-1} , а значит v_1 и v_0 окрашены одинаково, что невозможно, поскольку эти вершины смежные. Пришли к противоречию, следовательно, наше предположение было неверным.

$2 \Rightarrow 3$. Вначале докажем справедливость этого утверждения для связного графа.

Пусть G - ненулевой обыкновенный связный граф, не имеющий циклов нечетной длины. Зафиксируем некоторую вершину a графа и разобьем множество V вершин графа G на два непересекающихся подмножества V_1 и V_2 : в множество V_1 соберем все вершины b , для которых длина любой кратчайшей цепи, идущей из a в b четна, а в

V_2 - вершины, для которых эта длина нечетна. Вершину a поместим в V_1 . Для удобства подмножества V_1 и V_2 будем далее называть долями.

Покажем, что концы любого ребра графа G лежат в разных долях. Будем рассуждать от противного: предположим, найдется ребро e , концы c и d которого лежат одновременно в V_1 или в V_2 . Тогда $a \neq c$, $a \neq d$ (поскольку вершины, смежные с a , лежат с ней в разных долях). Пусть z'_{ac} - кратчайшая цепь, идущая из a в c , z''_{ad} - кратчайшая цепь, идущая из a в d , и w - последняя вершина цепи z'_{ac} , принадлежащая цепи z''_{ad} : $z'_{ac} : a \dots w \dots c$, $z''_{ad} : a \dots w \dots d$. Вершины c и d лежат в одной доле, следовательно, длины цепей z'_{ac} и z''_{ad} имеют одинаковую четность. А поскольку цепи z'_{ac} и z''_{ad} - кратчайшие, то их фрагменты z'_{aw} и z''_{aw} имеют одинаковую длину. Делаем вывод: длины фрагментов z'_{wc} и z''_{wd} имеют одинаковую четность.

Заметим, что по фрагменты z'_{wc} и z''_{wd} цепей z'_{ac} и z''_{ad} не имеют одинаковых ребер. Кроме того, они не содержат ребра e (ребро e не может быть внутренним ребром ни одной из этих цепей, так как в этом случае цепи z'_{wc} и z''_{wd} не могут быть кратчайшими; ребро e не может быть последним ребром ни одной из цепей, так как в этом случае длины цепей отличаются на единицу и, значит, не могут иметь одну четность).

Поэтому, склеивая инвертированный фрагмент z'_{wc} , фрагмент z''_{wd} и цепь dec , получаем цикл нечетной длины, что противоречит условию.

Таким образом, мы разбили множество вершин графа G на два непустых, непересекающихся подмножества (доли) так, что концы любого ребра принадлежат разным долям. Следовательно, граф G - двудольный.

Пусть теперь граф G - ненулевой обыкновенный несвязный граф, не имеющий циклов нечетной длины. Так как граф ненулевой, то среди его компонент связности есть неоднородные, причем, как показано выше, каждая неоднородная компонента - двудольный граф. Разобьем множество вершин графа G на два подмножества. В одно подмножество включим вершины одной из долей каждой двудольной компоненты связности, в другое - все остальные вершины графа (включая вершины однородных компонент связности). Тем самым множество вершин графа G окажется разбитым на два непустых, непересекающихся подмножества (доли) так, что

концы любого ребра принадлежат разным долям. Следовательно, граф G - двудольный.

$3 \Rightarrow 1$. Пусть G - ненулевой двудольный граф. Покрасим вершины одной доли одним цветом, а вершины другой - другим. Получим раскраску графа в два цвета. Поскольку граф ненулевой, т.е. имеет хотя бы одно ребро, то он не может быть раскрашен в один цвет. Таким образом, граф G бихроматический. ■

Задачи повышенной сложности

3.32. Доказать, что изоморфные графы имеют одинаковые хроматические числа.

3.33. Найти хроматическое число графа:

- а) $K_{n,m}$; б) K_n ; в) B_n .

3.34. Граф называется *критическим*, если удаление любой из его вершин (вместе с инцидентными ей ребрами) приводит к графу с меньшим хроматическим числом. Показать, что:

- а) полный граф K_n является критическим для всех $n > 1$;
б) граф C_n (цикл с n вершинами) является критическим тогда и только тогда, когда n нечетно.

3.35. Доказать, что для любого натурального P найдется обыкновенный бихроматический граф с 2^P вершинами и P^2 ребрами.

3.36. Описать обыкновенный граф с P вершинами, имеющий среди всех таких графов наибольшее число ребер.

§ 3.7. Фундаментальная система циклов графа

Абстрактный цикл. Обобщенный цикл. Пространство циклов.
Фундаментальная система циклов (базис пространства циклов).

Базовые понятия и утверждения

Пусть G - произвольный граф. На множестве циклов графа G введем бинарное отношение, которое назовем *отношением равенства* и определим следующим условием: будем считать, что два цикла равны, если множества их ребер совпадают.

Это бинарное отношение, будучи отношением эквивалентности, разбивает множество циклов данного графа на классы эквивалентности. Далее в этом параграфе, говоря о циклах графа, будем иметь в виду, если не оговорено противное, классы эквивалентности, а не отдельных их представителей. Эти классы эквивалентности назовем *абстрактными циклами*. Любой абстрактный цикл C задается множеством своих ребер.

Пример 1. Граф G , представленный диаграммой на рис. 3.72, имеет следующие абстрактные циклы:

$$C_1 = \{e_1, e_3, e_4\}, \quad C_2 = \{e_2, e_4, e_5\},$$

$$C_3 = \{e_2, e_6, e_7\}, \quad C_4 = \{e_1, e_2, e_3, e_5\},$$

$$C_5 = \{e_4, e_5, e_6, e_7\},$$

$$C_6 = \{e_1, e_2, e_3, e_4, e_6, e_7\}, \quad C_7 = \{e_1, e_3, e_5, e_6, e_7\}.$$

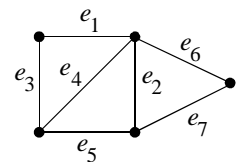


Рис. 3.72.

Обобщенным циклом будем называть абстрактный цикл или объединение непересекающихся абстрактных циклов. В число обобщенных циклов включим также цикл без ребер; назовем его *пустым циклом* и обозначим символом \emptyset .

На множестве всех обобщенных циклов графа G введем две операции:

а) операцию \oplus сложения по модулю 2: суммой обобщенных циклов C_1 и C_2 назовем обобщенный цикл $C_1 \oplus C_2$ с множеством ребер

$$\{e \mid ((e \in E_1) \wedge (e \notin E_2)) \vee ((e \notin E_1) \wedge (e \in E_2))\}$$

(здесь E_1 и E_2 - множества ребер обобщенных циклов C_1 и C_2 соответственно);

б) операцию умножения на 0 и 1: $0 \cdot C = \emptyset$; $1 \cdot C = C$.

Например, для обобщенных циклов графа G из примера 1 имеем:

$$C_1 \oplus C_2 = C_4, \quad C_1 \oplus C_5 = C_6, \quad C_3 \oplus C_4 = C_5.$$

Множество всех обобщенных циклов графа G с операциями сложения по модулю 2 и умножения на 0 и 1 образуют линейное пространство (убедиться в выполнении восьми аксиом линейного пространства несложно).

Операция \oplus естественным образом распространяется на любое конечное число обобщенных циклов.

Линейной комбинацией обобщенных циклов C_1, C_2, \dots, C_n назовем выражение

$$\alpha_1 \cdot C_1 \oplus \alpha_2 \cdot C_2 \oplus \dots \oplus \alpha_n \cdot C_n, \text{ где } \alpha_i \in \{0, 1\}.$$

Говорят, что система обобщенных циклов C_1, C_2, \dots, C_n *зависима*, если найдется набор чисел $\alpha_1, \alpha_2, \dots, \alpha_n$, не все из которых равны 0, такой, что

$$\alpha_1 \cdot C_1 \oplus \alpha_2 \cdot C_2 \oplus \dots \oplus \alpha_n \cdot C_n = \emptyset. \text{ В противном случае систему обобщенных циклов } C_1, C_2, \dots, C_n \text{ называют независимой.}$$

Система обобщенных циклов C_1, C_2, \dots, C_k графа G образует *базис линейного пространства циклов*, если она удовлетворяет двум условиям:

- 1) C_1, C_2, \dots, C_k линейно независима;
- 2) любой обобщенный цикл C графа G может быть представлен в виде линейной комбинации обобщенных циклов C_1, C_2, \dots, C_k .

Базис линейного пространства циклов называют *фундаментальной системой циклов*.

Из курса линейной алгебры нам известно, что в линейном пространстве может быть несколько базисов, но все они состоят из одного числа векторов линейного пространства (напомним, что это число называют размерностью линейного пространства).

Представляют интерес два вопроса: 1) из скольких циклов состоит фундаментальная система циклов произвольного графа и 2) как найти фундаментальную систему циклов графа?

Ответ на первый вопрос дает следующее утверждение: *число циклов в любой фундаментальной системе циклов графа равно его цикломатическому числу*.

Приведем **алгоритм построения фундаментальной системы циклов** произвольного графа G .

1-й шаг. Находим в графе G какой-либо обобщенный цикл C_1 и удаляем из него ребро e_1 («разрываем» цикл). Получаем граф G_1 .

k-й шаг. В графе G_{k-1} , построенном на $(k-1)$ -м шаге, находим какой-либо обобщенный цикл C_k и удаляем из него произвольное ребро e_k . Получаем граф G_k .

Если в графе G_k циклов нет, то C_1, C_2, \dots, C_k - искомая фундаментальная система циклов. Если в графе G_k обобщенные циклы остались, то повторяем k -й шаг.

Пример 2. Построим фундаментальную систему циклов графа G из примера 1.

1-й шаг. Возьмем цикл $C_1 = \{e_1, e_3, e_4\}$ и удалим из него одно ребро, пусть это будет ребро e_1 . Получим граф G_1 (рис. 3.73).

2-й шаг. В графе G_1 возьмем цикл $C_2 = \{e_2, e_4, e_5\}$ и удалим из него одно ребро, пусть это будет ребро e_5 . Получим граф G_2 (см. рис. 3.73).

3-й шаг. В графе G_2 возьмем цикл $C_3 = \{e_2, e_6, e_7\}$ и удалим из него одно ребро, пусть это будет ребро e_6 . Получим граф G_3 (рис. 3.73), в котором нет циклов.

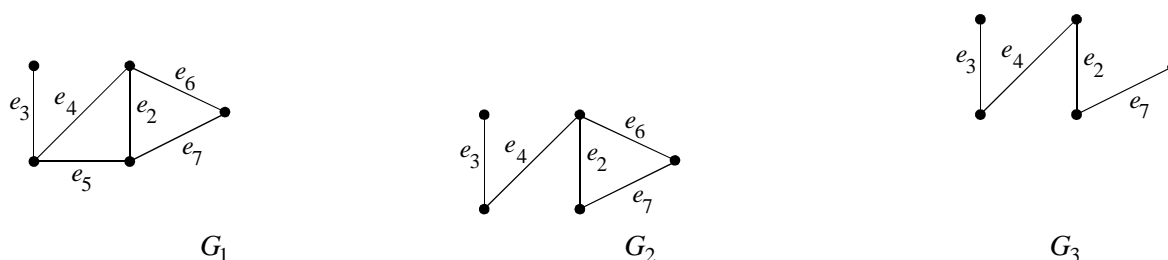


Рис. 3.73.

Следовательно, C_1, C_2, C_3 - фундаментальная система циклов (или, что то же самое, базис пространства циклов) графа G . Все остальные обобщенные циклы графа - линейные комбинации базисных циклов. Например, $C_5 = 0 \cdot C_1 \oplus 1 \cdot C_2 \oplus 1 \cdot C_3$.

Заметим, что действия по алгоритму не строго детерминированы: на каждом шаге мы имели несколько вариантов выбираемого цикла и удаляемого из него ребра. Если бы наш выбор был иным, то мы нашли бы другую фундаментальную систему циклов. Однако число циклов в этой другой фундаментальной системе тоже было бы равно трем. Например, C_1, C_2, C_7 также является фундаментальной системой циклов графа G .

Задачи повышенной сложности

3.37. Найти фундаментальную систему циклов следующих графов:

- а) графа, представленного диаграммой на рис. 3.2;
- б) графа G_4 , представленного диаграммой на рис. 3.6.

3.38. Подсчитать, сколько циклов входит в фундаментальную систему циклов следующих графов:

- а) графа, все ребра которого мосты;
- б) графа, имеющего 11 вершин, 10 ребер и состоящего из четырех компонент связности.

3.39. Подсчитать, сколько циклов входит в фундаментальную систему циклов следующих графов:

- а) K_5 ; б) $K_{3,3}$; в) K_n ;
- г) $K_{n,m}$; д) B_n .

3.40. Сколько всего обобщенных циклов имеет граф G , если $v(G) = 6$?

§ 3.8. Ориентированные графы

Ориентированный граф (орграф). Диаграмма орграфа. Полустепень исхода и полустепень захода вершины. Изоморфные орграфы. Матрицы смежности и инцидентности орграфа. Ориентированные пути, цепи, циклы. Слабая и сильная связность.

Базовые понятия и утверждения

1. Общие понятия. Перейдем к рассмотрению ориентированных графов.

Определение. Пусть V - конечное непустое множество, U - конечное множество, состоящее из поименованных упорядоченных пар элементов множества V , причем это могут быть пары из одинаковых элементов и одинаковые пары с разными именами. Совокупность множеств V и U называют *ориентированным графом* или, короче, *орграфом* и обозначают $G = (V, U)$.

Элементы множества V называют *вершинами*, а элементы множества U - *дугами*.

Если дуга u есть упорядоченная пара вершин v_i и v_j , то пишут $u = (v_i, v_j)$. Вершину v_i называют *началом дуги u* , а вершину v_j - ее *концом*. При этом говорят, что дуга u исходит (выходит) из вершины v_i и заходит в вершину v_j . Дугу u и вершину v_i (v_j) называют при этом *инцидентными*.

Орграфы принято изображать при помощи диаграмм, аналогичных диаграммам графов, отличие состоит лишь в том, что линиям, изображающим дуги, придают направление.

Пример 1. На рис. 3.74 изображена диаграмма ориентированного графа G с вершинами v_1, v_2, v_3, v_4, v_5 и дугами $u_1 = (v_3, v_2)$, $u_2 = (v_2, v_2)$, $u_3 = (v_5, v_4)$, $u_4 = (v_5, v_4)$.

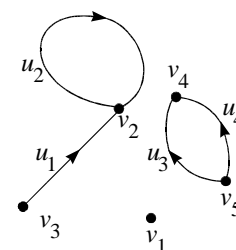


Рис. 3.74.

Число дуг, исходящих из вершины v_i орграфа G , называют *полустепенью исхода* вершины v_i и обозначают $\deg_+ v_i$. Число дуг, заходящих в вершину v_i , называют *полустепенью захода* вершины v_i и обозначают $\deg_- v_i$. Число $\deg v_i = \deg_- v_i + \deg_+ v_i$ называют *степенью вершины v_i* .

Из определений следует, что для любого орграфа $G = (V, U)$ справедливы равенства:

$$\sum_{a \in V} \deg_- a = \sum_{a \in V} \deg_+ a = |U|.$$

Понятия изолированных и висячих вершин, а также висячих, кратных дуг и петель вводятся для ориентированного графа по аналогии с подобными понятиями для неориентированного графа.

Дуги вида $u_1 = (a, b)$ и $u_2 = (b, a)$ называются симметричными.

Например, у орграфа на рис. 3.74 вершина v_1 - изолированная, вершина v_3 - висячая, дуга u_1 - висячая, дуга u_2 - петля, дуги u_3 и u_4 - кратные. Симметричных дуг у этого орграфа нет.

Ориентированные графы без петель и кратных дуг называют *обыкновенными ориентированными графами*.

Направленным графом называется такой обыкновенный орграф, который не имеет симметричных пар ориентированных дуг. Граф на рис. 3.74 - направленный.

2. Изоморфные орграфы. Орграфы G_1 и G_2 называются *изоморфными*, если существуют такие два взаимно-однозначных отображения $\phi: V(G_1) \rightarrow V(G_2)$ и $\phi: U(G_1) \rightarrow U(G_2)$, что для всякой дуги $u = (v_1, v_2)$ из $U(G_1)$ справедливо равенство $\phi(u) = (\phi(v_1), \phi(v_2))$.

Введем на множестве орграфов бинарное отношение изоморфизма, состоящее из всех пар изоморфных орграфов. Это отношение рефлексивно, симметрично и транзитивно, и значит, является отношением эквивалентности. Оно разбивает множество всех орграфов на классы эквивалентности так, что орграфы одного класса попарно изоморфны, а орграфы разных классов не изоморфны. Если не оговорено иное, орграфы рассматриваются с точностью до изоморфизма, т.е. объектом изучения являются не отдельные орграфы, а классы эквивалентности по отношению изоморфизма. Каждый класс эквивалентности по отношению изоморфизма можно задать диаграммой, не указывая на ней имена вершин и дуг.

Пример 2. На рис. 3.75 изображены диаграммы всех (с точностью до изоморфизма) обыкновенных орграфов с двумя вершинами.

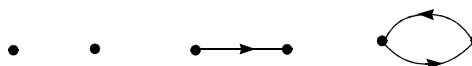


Рис. 3.75.

Упражнение 3.32. Найти все (с точностью до изоморфизма) обыкновенные орграфы с тремя вершинами и не более чем тремя дугами.

3. Матрица смежности и матрица инцидентности орграфа. Пусть G - произвольный орграф с m вершинами и n дугами. Пометим его дважды, т.е. упорядочим множества его вершин и дуг.

Определение. Матрицей смежности орграфа G называется матрица $A(G)$ размера $m \times m$, элементы которой $a_{ij} = s$, где s - число дуг, исходящих из вершины с номером i и заходящих в вершину с номером j .

Определение. Матрицей инцидентности орграфа G называется матрица $B(G)$ размера $m \times n$, элементы которой b_{ij} определены следующим образом:

- 1) $b_{ij} = 1$, если вершина с номером i - начало дуги с номером j и j -я дуга - не петля;
- 2) $b_{ij} = -1$, если вершина с номером i - конец дуги с номером j и j -я дуга - не петля;
- 3) $b_{ij} = 0$ во всех остальных случаях.

Пример 3. Выпишем матрицы смежности и инцидентности для графа из примера 1. Упорядочим естественным образом вершины и дуги графа, т.е. дадим вершинам и дугам номера в соответствии с их индексами (вершине v_1 - номер 1, вершине v_2 - номер 2 и т.д.). При таком упорядочивании вершин и дуг матрицы смежности $A(G)$ и инцидентности $B(G)$ выглядят следующим образом:

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix}; \quad B(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Упражнение 3.33. Выписать матрицы смежности и инцидентности для орграфа G с вершинами v_1, v_2, v_3, v_4 и дугами $u_1 = (v_2, v_1)$, $u_2 = (v_2, v_3)$, $u_3 = (v_1, v_3)$, $u_4 = (v_4, v_3)$.

4. Ориентированные пути, цепи, циклы на орграфе. Ориентированные путь, замкнутый путь, цепь, цикл, простая цепь и простой цикл определяются в ориентированном графе аналогично соответствующим неориентированным объектам в графах, однако с тем отличием, что в последовательности вершин и дуг

$v_0 u_1 v_1 \dots v_{k-1} u_k v_k$ для любой дуги u_i ($i=1, \dots, k$) вершина v_{i-1} является началом, а вершина v_i - концом.

Простой орцикл часто называют *контуром*.

Очевидно, что если на орграфе есть ориентированный путь из вершины v_1 в вершину v_2 , то есть и ориентированная простая цепь из v_1 в v_2 .

На множестве вершин ориентированного графа G введем бинарное отношение *достижимости (связности)* (\sim), включив в него все те пары вершин (a, b) , для которых на орграфе есть путь из a в b . Если $a \sim b$, то говорят, что b достижима из a .

Ориентированный граф называется *сильно связным*, если любая вершина в нем достижима из любой другой вершины.

Заменяя каждую дугу $u = (a, b)$ орграфа G на ребро $u = ab$, получаем неориентированный граф G_0 , называемый основанием данного орграфа G .

Орграф называется *связным (слабо связным)*, если связно его основание.

Очевидно, что сильно связный граф является связным; обратное утверждение в общем случае неверно.

Например, орграф G_1 (рис. 3.76) слабо связным является, а сильно связным - нет, орграф G_2 является как слабо связным, так и сильно связным, орграф G_3 не является ни слабо, ни сильно связным.

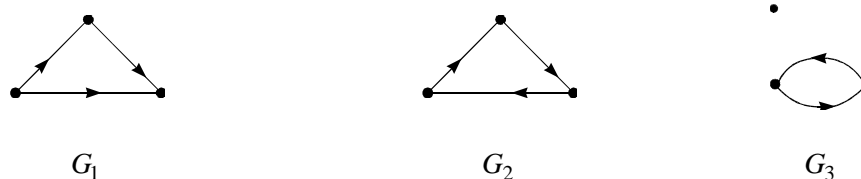


Рис. 3.76.

Орграф, основание которого есть полный граф, называется *турниром*.

Упражнение 3.34. Изобразить все (с точностью до изоморфизма) турниры с четырьмя вершинами. Выяснить, сколько среди них сильно связных.

5. Ориентированные деревья. Определим по индукции понятие *ориентированного дерева*.

Определение. Базис индукции. Орграф $T_0 = (V_0, U_0)$ с единственной вершиной a и пустым множеством дуг является ориентированным деревом. Вершина a - называется корнем этого дерева.

Индуктивный переход. Пусть орграфы $T_1 = (V_1, U_1)$ и $T_2 = (V_2, U_2)$ - ориентированные деревья с корнями a и b соответственно. Тогда орграф $T_3 = (V_3, U_3)$, где $V_3 = V_1 \cup \{c\}$, $c \notin V_1$, и $U_3 = U_1 \cup \{(c, a)\}$, является ориентированным деревом с корнем c и орграф $T_4 = (V_4, U_4)$, где $V_4 = V_1 \cup V_2$, $U_4 = U_1 \cup U_2$, является деревом с корнем $c = a = b$ (рис. 3.77).

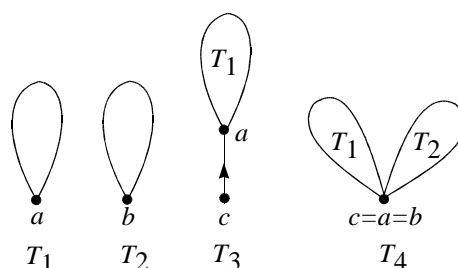


Рис. 3.77.

Из определения непосредственно следует, что в каждом ориентированном дереве есть ровно одна вершина (корень), в которую не входят дуги, в каждую из остальных вершин входит ровно по одной дуге и все вершины достижимы из корня.

Вершины ориентированного дерева, из которых не выходят дуги, называются *листьями*. Если из вершины a ведет дуга в вершину b , то a называется *отцом*, а b - *сыном*. Из определения дерева следует, что у каждой вершины (кроме корня) имеется единственный отец. Если из вершины a ведет путь в вершину b , то a называется *предком*, а b - *потомком*.

Путь из корня в лист называется *ветвью* дерева. Максимальная из длин ветвей дерева называется *высотой* дерева. *Глубина* вершины - это длина пути из корня в эту вершину. Для вершины v подграф дерева T , включающий все достижимые из v вершины и инцидентные им дуги, образует *поддерево* T_v с корнем v . *Высота вершины* v - это высота дерева T_v .

Пример 4. На рис. 3.78 изображена диаграмма ориентированного дерева, корнем которого является вершина a . Вершины d, t, s и p - листья. Вершина s - потомок вершин a, c и q ; вершина d - потомок вершин a и b ; вершина c - предок вершин t, p, q и s ; вершина a - отец вершин b и c ; вершина b - сын вершины a , отец

вершины d ; вершина q - сын вершины c , отец вершины s . Высота дерева равна трем. Глубина вершины q - 2, высота - 1; глубина вершины c - 1, высота - 2.

Ориентированное дерево называется *бинарным*, если у каждой его вершины имеется не более двух сыновей, причем дуги, ведущие к ним, помечены двумя разными метками (например, 0 и 1).

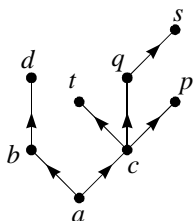


Рис. 3.78.

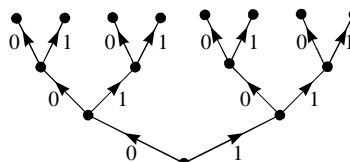


Рис. 3.79.

Бинарное дерево называется *полным*, если у всех его вершин (за исключением листьев) имеется ровно два сына и все его ветви имеют одинаковую длину.

На рис. 3.79 приведен пример диаграммы полного бинарного дерева.

Задачи повышенной сложности

3.41. Пусть $G = (V, E)$ - обыкновенный ориентированный граф, $A(G) = \|a_{i,j}\|$ - матрица смежности этого графа, отвечающая некоторой нумерации вершин v_1, v_2, \dots, v_n . Выразить через число вершин, дуг или степени вершин следующие суммы:

а) $\sum_{i=1}^n a_{i,j}$; б) $\sum_{j=1}^n a_{i,j}$; в) $\sum_{i=1}^n \sum_{j=1}^n a_{i,j}$.

3.42. Пусть $G = (V, E)$ - обыкновенный ориентированный граф, $A(G)$ - матрица смежности этого графа, отвечающая некоторой нумерации вершин v_1, v_2, \dots, v_n .

а) Доказать, что элемент, стоящий на пересечении i -й строки и j -го столбца матрицы A^k , равен числу путей длины k , ведущих из вершины v_i в вершину v_j .

б) Как в случае связного сильно ориентированного обыкновенного графа по матрице $A(G)$ определить длину кратчайшего по числу дуг пути между вершинами v_i и v_j ($i \neq j$)?

§ 3.9. Отыскание кратчайших путей. Алгоритм Дейкстры

Сеть. Длина пути. Расстояние между вершинами. Кратчайший путь. Задача отыскания кратчайших путей.

Базовые понятия и утверждения

Ориентированный граф $G = (V, U)$ называют *взвешенным*, если на множестве его дуг задано отображение $\rho: U \rightarrow R$, т.е. каждой дуге $u = (a, b)$ приписано действительное число $\rho(u)$, называемое *весом*, или *длиной дуги*. Взвешенный связный орграф называется *сетью*.

В этом параграфе речь пойдет исключительно об ориентированных графах, поэтому далее, чтобы избежать излишней громоздкости, прилагательное «ориентированный» будем опускать.

Ограничимся случаем неотрицательных весов. Пусть на графе G имеется путь P из вершины a в вершину b и u_1, u_2, \dots, u_k - последовательность дуг этого пути. Число

$$\rho(P) = \sum_{i=1}^k \rho(u_i) \text{ называется } \textit{весом}, \text{ или } \textit{длиной пути } P.$$

Наименьшую из длин путей из вершины a в вершину b назовем *расстоянием от a до b* , а тот путь из a в b , длина которого равна расстоянию от a до b , будем называть *кратчайшим путем из a в b* .

Рассмотрим *задачу о кратчайших путях*: в заданной сети G найти расстояние и кратчайший путь от фиксированной вершины s до остальных вершин.

Одним из алгоритмов ее решения является алгоритм Дейкстры. Прежде чем перейти к его изложению, введем ряд условных обозначений.

Через $d(v)$ обозначим расстояние от s до v , т.е. длину кратчайшего пути из s в v (если, конечно, хотя бы один путь из s в v существует). Каждой паре вершин v и w поставим в соответствие число $c(v, w)$, которое определим следующим образом: если в графе есть дуги, исходящие из v и заходящие в w , то $c(v, w)$ положим равным весу самой «легкой» из них; если ни одной такой дуги в графе нет, то будем считать $c(v, w) = \infty$.

Алгоритм Дейкстры последовательно находит ближайшую к s вершину (т.е. вершину, расстояние от s до которой является наименьшим из расстояний от вершины s до других вершин сети) и расстояние от s до нее, затем следующую ближайшую к s вершину и т.д.

На каждом шаге алгоритма всем вершинам графа даются «метки», временные либо постоянные. Постоянные метки имеют те вершины, расстояние до которых от вершины s уже определено (собственно, это расстояние и является постоянной меткой вершины). Любая другая вершина w получает временную метку, равную длине того пути, который является самым коротким (т.е. наименьшим по длине) из всех путей из s в w , идущих, вплоть до w , исключительно через вершины с постоянными метками.

Перейдем к точной формулировке алгоритма Дейкстры.

Алгоритм Дейкстры. 0-й шаг. Определим первую ближайшую к s вершину: это сама вершина s , для нее $d(s) = 0$.

k -й шаг. Пусть к началу шага определены k вершин, ближайших к вершине s , т.е. определено множество $S_k = \{v_1 = s, v_2, \dots, v_k\}$, и для всех вершин этого множества вычислены расстояния $d(v_i)$, $i = 1, 2, \dots, k$ (эти расстояния удобно рассматривать как постоянные метки вершин из S_k). Обозначим через F_k множество $V \setminus S_k$.

Если множество F_k пусто, то все возможные расстояния от вершины s до остальных вершин определены до k -го шага.

В противном случае для каждой вершины $w \in F_k$ определим временную метку

$$D_k(w) = \min_{v \in S_k} \{d(v) + c(v, w)\}.$$

Если ни для одной вершины $w \in F_k$ этот минимум определить нельзя, то все возможные расстояния от вершины s до остальных вершин определены до k -го шага.

В противном случае из равенства

$$D_k(v_{k+1}) = \min_{w \in F_k} \{D_k(w)\}$$

определится вершина v_{k+1} , ближайшая к s из вершин подмножества F_k , причем

$$d(v_{k+1}) = D_k(v_{k+1})$$

(вершина v_{k+1} получает при этом постоянную метку - расстояние $d(v_{k+1})$).

Таким образом, по окончании k -го шага множество вершин, до которых от вершины s вычислено расстояние, расширяется на один элемент.

Восстановление кратчайшего пути. Кратчайший путь от вершины s до вершины w восстанавливается пошагово, начиная с вершины w до возврата к вершине s . Опишем один шаг возвращения. Для вершины w среди всех вершин v , для которых существует дуга, ведущая из v в w , находится такая дуга, для которой выполнено условие $d(w) = d(v) + c(v, w)$. Тем самым восстанавливается последняя дуга кратчайшего пути от s до w .

Пример 1. В графе, изображенном на рис. 3.80, найти расстояния от вершины s до остальных вершин и восстановить кратчайший путь от вершины s до вершины d .

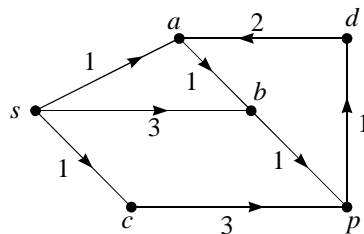


Рис. 3.80.

◀ В табл. 3.1 отражены шаги алгоритма Дейкстры. В первом столбце каждой строки указан номер шага, во втором столбце - множество вершин S_k , для которых к началу k -го шага найдено расстояние от вершины s . В остальных столбцах указаны метки вершин на шаге, номер которого обозначен в заголовке строки.

Таблица 3.1

| k | S_k | $D_k(s)$ | $D_k(a)$ | $D_k(b)$ | $D_k(c)$ | $D_k(d)$ | $D_k(p)$ |
|-----|---------------------|----------|----------|----------|----------|----------|----------|
| 0 | | 0 | | | | | |
| 1 | $\{s\}$ | | 1 | 3 | 1 | ∞ | ∞ |
| 2 | $\{s, a\}$ | | | 2 | 1 | ∞ | ∞ |
| 3 | $\{s, a, c\}$ | | | 2 | | ∞ | 4 |
| 4 | $\{s, a, c, b\}$ | | | | | ∞ | 3 |
| 5 | $\{s, a, c, b, p\}$ | | | | | 4 | |

Расстояние от вершины s до a равно 1, от s до b равно 2, от s до c равно 1, от s до d равно 4, от s до p равно 3.

Кратчайший путь от вершины s до вершины d проходит последовательно через вершины s, a, b, p, d . ►

Упражнение 3.35. В сети, изображенной на рис. 3.81, найти расстояние от вершины s до остальных вершин и указать кратчайший путь от вершины s до вершины q .

Упражнение 3.36. В сети, изображенной на рис. 3.82, найти расстояние от вершины s до остальных вершин и указать кратчайший путь от вершины s до вершины t .

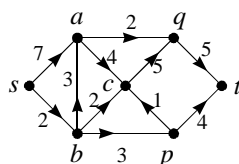


Рис. 3.81.

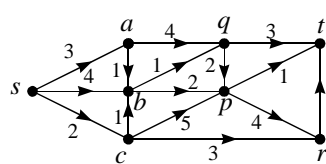


Рис. 3.82.

Теоретические обоснования

Обоснование алгоритма Дейкстры. Обоснуем вначале равенство $d(v_{k+1}) = D_k(v_{k+1})$.

Пусть $P: s = w_0 u_1 w_1 \dots u_r w_r = v_{k+1}$ - произвольный путь из s в v_{k+1} . Поскольку $D_k(v_{k+1})$ - длина одного из путей из вершины s в вершину v_{k+1} , то достаточно доказать неравенство $D_k(v_{k+1}) \leq \rho(P)$. Среди всех вершин пути P выберем вершину w_j с наименьшим номером среди тех, которые не входят в множество S_k . Так как начальная вершина пути P входит в S_k , а конечная не входит в S_k , то такой номер j найдется. Итак, $w_{j-1} \in S_k$, $w_j \in F_k$. Тогда из определения $D_k(w)$ и длины пути вытекают соотношения

$$\begin{aligned} D_k(w_j) &\leq d(w_{j-1}) + c(w_{j-1}, w_j) \leq \\ &\leq c(w_0, w_1) + \dots + c(w_{j-1}, w_j) \leq \rho(P). \end{aligned}$$

Кроме того, в силу выбора вершины v_{k+1} имеем, что $D_k(v_{k+1}) \leq D_k(w_j)$. С учетом этих неравенств получаем, что $D_k(v_{k+1}) \leq \rho(P)$. Следовательно, $d(v_{k+1}) = D_k(v_{k+1})$.

Осталось убедиться, что вершина v_{k+1} является $(k+1)$ -й ближайшей к s вершиной. Для этого достаточно доказать неравенство $d(v_{k+1}) \leq d(w)$ для всех $w \in F_k$. Зафиксируем одну из вершин $w \in F_k$. Обозначим через P_1 кратчайший путь из вершины s в вершину v_{k+1} , а через P_2 - кратчайший путь из вершины s в вершину w . Тогда $\rho(P_1) = d(v_{k+1})$ и $\rho(P_2) = d(w)$. Выше доказано равенство $d(v_{k+1}) = D_k(v_{k+1})$. Повторим для пути P_2

рассуждения, приведенные в первой части обоснования для пути P . Среди всех вершин пути P_2 выберем вершину w_j с наименьшим номером среди тех, которые не входят в множество S_k . Имеем: $w_{j-1} \in S_k$, $w_j \in F_k$. Тогда

$$\begin{aligned} D_k(w_j) &\leq d(w_{j-1}) + c(w_{j-1}, w_j) \leq \\ &\leq c(w_0, w_1) + \dots + c(w_{j-1}, w_j) \leq \rho(P_2). \end{aligned}$$

Кроме того, в силу выбора вершины v_{k+1} имеем, что $D_k(v_{k+1}) \leq D_k(w_j)$. Следовательно,

$$\rho(P_1) = d(v_{k+1}) = D_k(v_{k+1}) \leq D_k(w_j) \leq d(w) = \rho(P_2),$$

что и требовалось доказать. ■

§ 3.10. Задача о максимальном потоке в сети

Сеть. Поток в сети. Величина потока. Максимальный поток. Дополняющая цепь. Разрез в сети. Минимальный разрез. Алгоритм Форда - Фалкерсона.

Базовые понятия и утверждения

В этом параграфе будут рассматриваться сети $G=(V,U,\rho)$, имеющие единственную вершину s с нулевой полустепенью захода и единственную вершину t с нулевой полустепенью исхода. Вершину s будем называть *источником*, а вершину t - *стоком*, вес $\rho(u)$ дуги u - *пропускной способностью* дуги u .

Для удобства изложения введем следующие обозначения. Через $v+$ обозначим множество дуг, для которых вершина v является началом, а через $v-$ обозначим множество дуг, для которых вершина v является концом.

Определение. *Потоком* в сети $G=(V,U,\rho)$ называется функция $f:U \rightarrow [0,\infty)$, удовлетворяющая условиям:

$$1) \forall u \in U \quad f(u) \leq \rho(u);$$

$$2) f(v-) = f(v+) \text{ для всех вершин } v, v \neq s, v \neq t, \text{ где } f(v-) = \sum_{u \in v-} f(u) \text{ и}$$

$$f(v+) = \sum_{u \in v+} f(u).$$

Значение $f(v-)$ можно интерпретировать как поток, втекающий в вершину v , а значение $f(v+)$ - как поток, вытекающий из вершины v . Тогда второе условие можно переформулировать так: поток, втекающий в любую вершину, за исключением источника и стока, должен быть равен потоку, вытекающему из этой вершины.

Условие 1) называется условием ограничения по пропускной способности, а условие 2) - условием сохранения потока в вершинах.

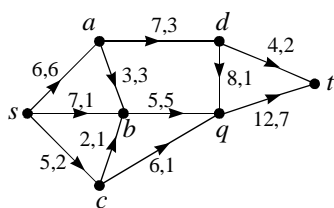


Рис. 3.83.

Пример 1. На рис. 3.83 дан пример сети и потока f в ней.

Около каждой дуги u указаны через запятую ее пропускная способность $\rho(u)$ и текущий по ней поток $f(u)$.

Определение. Положим $\|f\| = f(s+)$. Число $\|f\|$ называется *величиной потока*.

Определение. Поток f^* называется *максимальным*, если для любого потока f справедливо неравенство $\|f\| \leq \|f^*\|$.

Задача о максимальном потоке формулируется следующим образом: *в заданной сети G найти поток максимальной величины.*

Задача о максимальном потоке имеет особенность, отличающую ее от рассмотренных нами ранее задач дискретной оптимизации. В предшествующих задачах искомый объект существовал очевидным образом и в принципе мог быть найден полным перебором. Например, можно было перебрать все остовы и выбрать среди них минимальный или перебрать все пути между заданными вершинами и выбрать среди них кратчайший. В задаче о максимальном потоке полный перебор принципиально невозможен и существование максимального потока не является очевидным. Однако можно доказать, что *в каждой сети существует максимальный поток* (доказательство этого утверждения опустим).

При решении задачи об отыскании максимального потока в сети приходится рассматривать неориентированные простые цепи, т.е. такие последовательности попарно различных вершин и дуг $v_0 u_1 v_1 \dots v_{k-1} u_k v_k$, в которой любые два соседних элемента инцидентны. Поскольку цепи других видов нас при этом интересовать не будут, то далее в этом параграфе неориентированные простые цепи будем называть просто цепями.

Если в цепи $v_0 u_1 v_1 \dots v_{k-1} u_k v_k$ дуга u_i выходит из вершины v_{i-1} и заходит в вершину v_i , то она называется *прямой* дугой цепи. Если же дуга u_i выходит из вершины v_i и заходит в вершину v_{i-1} , то она называется *обратной* дугой цепи.

Пусть f - поток в сети G и P - цепь из v в w . Для каждой дуги u цепи P будем вычислять остаточную пропускную способность $h(u)$ при данном потоке f по следующему правилу: если дуга u прямая, то $h(u) = r(u) - f(u)$, а если обратная, то $h(u) = f(u)$. В свою очередь остаточную пропускную способность $h(P)$ цепи P при данном потоке f будем определять по формуле $h(P) = \min_{u \in P} \{h(u)\}$.

Определение. Цепь P из v в w называется *дополняющей* для f , если $h(P) > 0$.

Пример 2. В сети, изображенной на рис. 3.83, цепь, включающая последовательно вершины s, b, c, q, t , является дополняющей для потока, рассмотренного в примере 1.

Одним из алгоритмов, позволяющих построить максимальный поток, является алгоритм Форда - Фалкерсона.

Алгоритм Форда - Фалкерсона. *0-й шаг.* Пускаем по сети нулевой поток f_0 , т.е. полагаем $f_0(u) = 0$ для всех дуг $u \in U$.

k-й шаг. Пусть к началу шага в сети течет поток f_{k-1} . Для этого потока f_{k-1} ищем дополняющую цепь из s в t .

Если такой цепи нет, то f_{k-1} - искомый максимальный поток.

В противном случае, если для потока f_{k-1} дополняющая цепь из s в t имеется, даем ей имя P_k и определяем на множестве U функцию f_k по следующему правилу:

$$f_k(u) = \begin{cases} f_{k-1}(u) + h(P_k), & \text{если } u \in P_k, u - \text{прямая дуга,} \\ f_{k-1}(u) - h(P_k), & \text{если } u \in P_k, u - \text{обратная дуга,} \\ f_{k-1}(u), & \text{если } u \notin P_k. \end{cases}$$

Построенная таким образом функция f_k является потоком, и величина этого потока определяется равенством

$$\|f_k\| = \|f_{k-1}\| + h(P_k).$$

Замечание. Возникает существенный вопрос: закончится ли работа алгоритма за конечное число шагов? Оказывается, хотя в подавляющем большинстве случаев алгоритм заканчивает свою работу за конечное число шагов, заикливание возможно. Гарантировать построение максимального потока можно в том случае, если на каждом шаге производить увеличение потока вдоль кратчайших по числу дуг дополняющих цепей.

Доказательство алгоритма Форда - Фалкерсона приведено во второй части параграфа.

Пример 3. Построим максимальный поток для сети, изображенной на рис. 3.84,а.

◀ *Шаг 0.* Для любой дуги u положим $f_0(u) = 0$ (на рисунке этот поток отмечать не будем, более того, договоримся и далее нулевой поток на дугах не отмечать), $\|f_0\| = 0$.

Шаг 1. $P_1: s \xrightarrow{6} a \xrightarrow{7} d \xrightarrow{4} t$ - дополняющая цепь для потока f_0 (стрелки заменяют имена дуг, под каждой стрелкой записана остаточная пропускная способность $h(u)$ соответствующей дуги u), $h(P_1) = \min_{u \in P_1} \{h(u)\} = 4$, поток f_1 указан на рис. 3.84,б,

$$\|f_1\| = \|f_0\| + h(P_1) = 4.$$

Шаг 2. $P_2: s \xrightarrow{2} a \xrightarrow{3} b \xrightarrow{5} q \xrightarrow{12} t$ - дополняющая цепь для потока f_1 , $h(P_2)=2$, поток f_2

указан на рис. 3.84,в, $\|f_2\|=\|f_1\|+h(P_2)=6$.

Шаг 3. $P_3: s \xrightarrow{7} b \xrightarrow{3} q \xrightarrow{10} t$ - дополняющая цепь для потока f_2 , $h(P_3)=3$, поток f_3

указан на рис. 3.84,г, $\|f_3\|=\|f_2\|+h(P_3)=9$.

Шаг 4. $P_4: s \xrightarrow{5} c \xrightarrow{6} q \xrightarrow{7} t$ - дополняющая цепь для потока f_3 , $h(P_4)=5$, поток f_4

указан на рис. 3.84,д, $\|f_4\|=\|f_3\|+h(P_4)=14$.

Шаг 5. $P_5: s \xrightarrow{4} b \xrightarrow{2} a \xrightarrow{3} d \xrightarrow{8} q \xrightarrow{2} t$ - дополняющая цепь для потока f_3 , $h(P_5)=2$, поток

f_5 указан на рис. 3.84,е, $\|f_5\|=\|f_4\|+h(P_5)=16$.

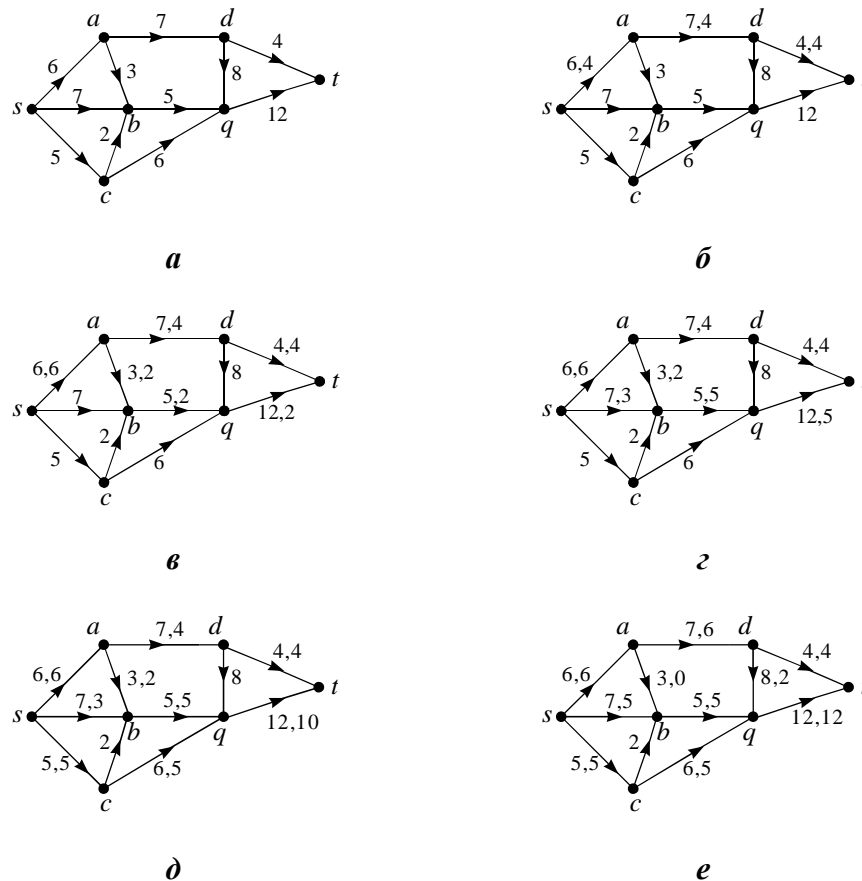


Рис. 3.84.

Для потока f_5 нет ни одной дополняющей цепи из s в t . Следовательно, поток f_5 является максимальным. ►

Определение. Разрезом (V_s, V_t) в сети G называется пара множеств V_s и V_t , удовлетворяющих условиям:

$$1) s \in V_s, t \in V_t;$$

$$2) V_s \cup V_t = V;$$

$$3) V_s \cap V_t = \emptyset.$$

Через $V_s \rightarrow V_t$ обозначим множество всех дуг, начала которых лежат в V_s , а концы - в V_t , а через $V_t \rightarrow V_s$ - множество всех дуг, начала которых лежат в V_t , а концы - в V_s .

Определение. Положим $\rho(V_s \rightarrow V_t) = \sum_{u \in V_s \rightarrow V_t} \rho(u)$. Число $\rho(V_s \rightarrow V_t)$ называется

пропускной способностью разреза (V_s, V_t) .

Определение. Разрез (V_s^*, V_t^*) называется *минимальным*, если для любого разреза (V_s, V_t) справедливо неравенство $\rho(V_s^* \rightarrow V_t^*) \leq \rho(V_s \rightarrow V_t)$.

Во второй части параграфа будет доказано, что *величина максимального потока в произвольной сети равна пропускной способности минимального разреза.*

Если для сети найден максимальный поток, то минимальный разрез можно найти следующим образом: включить в множество V_s вершину s и все вершины $v \in V$, для каждой из которых существует цепь P из s в v с остаточной пропускной способностью $h(P) > 0$; положить $V_t = V \setminus V_s$. Полученный таким образом разрез (V_s, V_t) будет минимальным.

Пример 4. Вернемся к примеру 3. Используя найденный максимальный поток, несложно определить, что единственной вершиной, в которую можно добраться по цепи с положительной остаточной пропускной способностью, является вершина b . Следовательно, пара множеств $V_s = \{s, b\}$ и $V_t = \{a, c, d, q, t\}$ образует минимальный разрез.

Упражнение 3.37. Найти максимальный поток и минимальный разрез для сети, изображенной:

а) на рис. 3.85; б) на рис. 3.86.

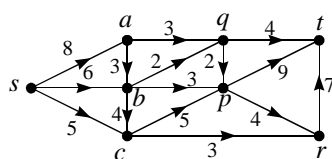


Рис. 3.85.

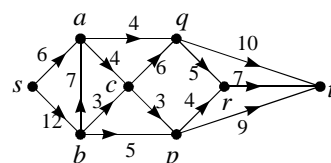


Рис. 3.86.

Теоретические обоснования

Перейдем к обоснованию алгоритма Форда - Фалкерсона. Для произвольных разреза (V_s, V_t) и потока f введем две числовые характеристики:

$$f(V_s \rightarrow V_t) = \sum_{u \in V_s \rightarrow V_t} f(u) - \text{величина потока из } V_s \text{ в } V_t,$$

$$f(V_t \rightarrow V_s) = \sum_{u \in V_t \rightarrow V_s} f(u) - \text{величина потока из } V_t \text{ в } V_s.$$

Пример 5. Для потока f , текущего по сети в примере 1, и разреза (V_s, V_t) , где $V_s = \{s, a, c\}$ и $V_t = \{b, d, q, t\}$, имеем $f(V_s \rightarrow V_t) = 9$, $f(V_t \rightarrow V_s) = 0$.

Лемма 1. Для любого потока f и любого разреза (V_s, V_t) справедливо равенство $\|f\| = f(V_s \rightarrow V_t) - f(V_t \rightarrow V_s)$.

Доказательство. Согласно условию сохранения потока в вершинах, для произвольной вершины $v \in V_s \setminus \{s\}$ имеем:

$$f(v-) = f(v+) \text{ или } f(v+) - f(v-) = 0.$$

Следовательно, $\sum_{v \in V_s \setminus \{s\}} (f(v+) - f(v-)) = 0$. Сложив это равенство почленно с равенством $f(s+) = \|f\|$, получим:

$$\sum_{v \in V_s} (f(v+) - f(v-)) = \|f\|,$$

$$\sum_{v \in V_s} \left(\sum_{u \in v+} f(u) \right) - \sum_{v \in V_s} \left(\sum_{u \in v-} f(u) \right) = \|f\|,$$

$$\sum_{\substack{u: u=(a,b), \\ a \in V_s}} f(u) - \sum_{\substack{u: u=(a,b), \\ b \in V_s}} f(u) = \|f\|,$$

$$\sum_{\substack{u: u=(a,b), \\ a \in V_s, b \notin V_s}} f(u) + \sum_{\substack{u: u=(a,b), \\ a \in V_s, b \in V_s}} f(u) - \sum_{\substack{u: u=(a,b), \\ a \notin V_s, b \in V_s}} f(u) - \sum_{\substack{u: u=(a,b), \\ a \notin V_s, b \notin V_s}} f(u) = \|f\|.$$

Первое и третье слагаемые последнего равенства взаимно уничтожаются, следовательно,

$$\sum_{\substack{u: u=(a,b), \\ a \in V_s, b \notin V_s}} f(u) - \sum_{\substack{u: u=(a,b), \\ a \notin V_s, b \in V_s}} f(u) = \|f\|,$$

$$f(V_s \rightarrow V_t) - f(V_t \rightarrow V_s) = \|f\|. \blacksquare$$

Следствие 1. $\|f\| = f(t-)$.

Доказательство. Пусть $V_t = \{t\}$, $V_s = V \setminus \{t\}$. Тогда $f(V_s \rightarrow V_t) = f(t-)$, $f(V_t \rightarrow V_s) = 0$ и равенство из леммы 1 запишется следующим образом: $\|f\| = f(t-)$. \blacksquare

Следствие 2. Для любого потока f и любого разреза (V_s, V_t) справедливо неравенство $\|f\| \leq \rho(V_s \rightarrow V_t)$.

Доказательство. Из условия ограничения по пропускной способности

$$\sum_{u \in V_s \rightarrow V_t} f(u) \leq \sum_{u \in V_s \rightarrow V_t} \rho(u), \text{ т.е. } f(V_s \rightarrow V_t) \leq \rho(V_s \rightarrow V_t).$$

Тогда с учетом равенства из леммы 1 имеем:

$$\|f\| = f(V_s \rightarrow V_t) - f(V_t \rightarrow V_s) \leq f(V_s \rightarrow V_t) \leq \rho(V_s \rightarrow V_t). \blacksquare$$

Лемма 2. Если для некоторого потока f и некоторого разреза (V_s, V_t) выполняется равенство $\|f\| = \rho(V_s \rightarrow V_t)$, то поток f максимален, а разрез (V_s, V_t) минимален.

Доказательство. Пусть f^* - максимальный поток, а (V_s^*, V_t^*) - минимальный разрез.

Тогда справедлива следующая цепочка неравенств:

$$\|f\| \leq \|f^*\| \stackrel{\text{сл. 2 леммы 1}}{\leq} \rho(V_s^* \rightarrow V_t^*) \leq \rho(V_s \rightarrow V_t).$$

Поскольку крайние члены в этой цепочке неравенств совпадают, то она превращается в цепочку равенств, а это означает, что поток f максимален, а разрез (V_s, V_t) минимален.

\blacksquare

Лемма 3. Пусть f - поток в сети G и P является для f дополняющей цепью из s в t .

Тогда в сети G существует поток $f \%$ такой, что $\|f \%\| = \|f\| + h(P)$.

Доказательство. Определим в сети G функцию $f \%: U \rightarrow R$ по формуле

$$f \%(u) = \begin{cases} f(u) + h(P), & \text{если } u \in P, \text{ } u \text{ - прямая дуга,} \\ f(u) - h(P), & \text{если } u \in P, \text{ } u \text{ - обратная дуга,} \\ f(u), & \text{если } u \notin P. \end{cases}$$

Докажем, что функция f^P является потоком.

Вначале покажем, что функция f^P принимает только неотрицательные значения и удовлетворяет условию 1) определения потока.

Пусть u - прямая дуга цепи P . Тогда

$$0 \leq f^P(u) = f(u) + h(P) \leq f(u) + h(u) = f(u) + (p(u) - f(u)) = p(u).$$

Пусть u - обратная дуга цепи P . Тогда

$$f^P(u) = f(u) - h(P) \geq f(u) - h(u) = f(u) - f(u) = 0 \text{ и}$$

$$f^P(u) = f(u) - h(P) \leq f(u) \leq p(u).$$

Таким образом, для любой дуги u цепи P выполнено $0 \leq f^P(u) \leq p(u)$. Если $u \notin P$, то $0 \leq f^P(u) = f(u) \leq p(u)$.

Теперь убедимся в выполнении для функции f^P условия 2) определения потока.

Понятно, что это условие следует проверить лишь для вершин v , входящих в цепь P .

Пусть u_1 - дуга, по которой пришли в вершину v , а u_2 - дуга, по которой ушли из v .

Каждая из этих дуг может быть как прямой, так и обратной в цепи P . Следовательно, возможны четыре различных случая.

1. Пусть обе дуги u_1 и u_2 - прямые. В этом случае

$$f^P(u_1) = f(u_1) + h(P) \text{ и } f^P(u_2) = f(u_2) + h(P),$$

так что

$$f^P(v+) = f(v+) + h(P) \text{ и } f^P(v-) = f(v-) + h(P).$$

Но для потока f выполнено условие $f(v+) = f(v-)$, следовательно, и для потока f^P имеем $f^P(v+) = f^P(v-)$.

2. Пусть дуга u_1 - прямая и u_2 - обратная, т.е. обе дуги входят в вершину v . В этом случае

$$f^P(u_1) = f(u_1) + h(P) \text{ и } f^P(u_2) = f(u_2) - h(P),$$

так что

$$f^P(v-) = f(v-) + h(P) - h(P) = f(v-) \text{ и } f^P(v+) = f(v+).$$

Так как для потока f выполнено условие $f(v+) = f(v-)$, то и для потока $f\%$ имеем $f\%(v+) = f\%(v-)$.

Аналогично рассматриваются случай 3 (обе дуги u_1 и u_2 - обратные) и случай 4 (дуга u_1 - обратная и u_2 - прямая).

Таким образом, мы показали, что функция $f\%$ - поток в цепи P .

Определим теперь величину потока $f\%$. Напомним, что $\|f\% \| = f\%(s+)$. Из вершины s дуги только выходят, поэтому та единственная дуга u с началом в вершине s , которая входит в цепь P , является прямой дугой цепи P . Следовательно, для нее выполнено равенство $f\%(u) = f(u) + h(P)$. На остальных дугах, выходящих из s , поток не менялся, так что имеем:

$$\|f\% \| = f\%(s+) = \|f\| + h(P) . \blacksquare$$

Объединяет приведенные выше результаты следующая теорема.

Теорема 3.11 (Форда - Фалкерсона). Для потока f в сети G следующие условия эквивалентны:

- 1) поток f максимален;
- 2) для f не существует дополняющей цепи из s в t ;
- 3) существует разрез (V_s, V_t) , для которого $\|f\| = \rho(V_s \rightarrow V_t)$.

Доказательство. Доказательство проведем по следующей схеме: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$.

$1 \Rightarrow 2$. Будем рассуждать от противного. Предположим, что для максимального потока f найдется дополняющая цепь P из s в t . Тогда по лемме 3 существует поток $f\%$ такой, что $\|f\% \| = \|f\| + h(P)$, т.е. $\|f\% \| > \|f\|$. Полученное неравенство противоречит тому, что поток f максимален.

$2 \Rightarrow 3$. Определим V_s как множество всех вершин $v \in V$, для каждой из которых существует цепь P из s в v такая, что $h(P) > 0$. Добавим в V_s вершину s и положим $V_t = V \setminus V_s$. Докажем, что для полученного разреза (V_s, V_t) выполняется равенство

$$\|f\| = \rho(V_s \rightarrow V_t) .$$

Вначале покажем, что для любой дуги $u \in V_s \rightarrow V_t$ выполнено равенство $f(u) = \rho(u)$.

Будем рассуждать от противного. Предположим, найдется дуга $u \in V_s \rightarrow V_t$, для которой $f(u) < \rho(u)$. Пусть v - начало, w - конец этой дуги, $v \in V_s$, $w \in V_t$. Множество V_s определено нами так, что можно говорить о существовании цепи P из s в v , для которой $h(P) > 0$. Дополним эту цепь дугой u и вершиной w до цепи Q из s в w . Тогда

$$h(Q) = \min(h(P), h(u)) = \min(h(P), \rho(u) - f(u)) > 0$$

(здесь учтено, что u - прямая дуга в цепи Q и, значит, $h(u) = \rho(u) - f(u)$). Но выполнение неравенства $h(Q) > 0$ означает, что вершина $w \in V_s$. Пришли к противоречию. Таким образом, предположение было неверным и для любой дуги $u \in V_s \rightarrow V_t$ выполнено равенство $f(u) = \rho(u)$. Следовательно, $f(V_s \rightarrow V_t) = \rho(V_s \rightarrow V_t)$.

Аналогично показывается, что для всех дуг $u \in V_t \rightarrow V_s$ имеет место равенство $f(u) = 0$ и, следовательно, $f(V_t \rightarrow V_s) = 0$. Поскольку (в силу леммы 1) для любого разреза справедливо равенство $\|f\| = f(V_s \rightarrow V_t) - f(V_t \rightarrow V_s)$, для построенного разреза получаем равенство

$$\|f\| = \rho(V_s, V_t).$$

$3 \Rightarrow 1$. Это утверждение полностью совпадает с утверждением доказанной ранее леммы 2. ■

Следствие. *Величина максимального потока в произвольной сети равна пропускной способности минимального разреза.*

Замечание. В ходе доказательства теоремы при обосновании справедливости перехода $2 \Rightarrow 3$ был, по существу, предложен алгоритм нахождения минимального разреза.

§ 3.11. Реализация булевых функций с помощью схем из функциональных элементов

Схема из функциональных элементов. Входы, функциональные элементы, выходы схемы. Сложность схемы. Глубина вершины. Функции, реализуемые в вершинах схемы. Функции, реализуемые схемой. Проблема анализа и проблема синтеза схем.

Схемой из функциональных элементов в базисе $\{\wedge, \vee, \neg\}$ называют ориентированный граф $S = (V, U)$ без циклов, удовлетворяющий следующим условиям.

1. Все вершины этого графа разбиваются на три подмножества V_1, V_2, V_3 ($V = V_1 \cup V_2 \cup V_3$, $V_i \cap V_j = \emptyset$ при $i \neq j$) так, что:

а) в каждую вершину из V_1 не входит ни одной дуги;

б) в каждую вершину из V_2 входит по одной дуге;

в) в каждую вершину из V_3 входит по две дуги.

2. Каждая вершина из V_1 помечена некоторой переменной, причем разные вершины помечены разными переменными.

3. Каждая вершина из V_2 помечена связкой \neg .

4. Каждая вершина из V_3 помечена одной из связок \wedge и \vee .

5. Выделено некоторое подмножество V^* ($V^* \subseteq V$) вершин.

Вершины из V_1 называются *входами* схемы. Если вершина v из V_1 помечена переменной x_i , то говорят, что на вход v подается переменная x_i .

Вершины из V_2 и V_3 называются *функциональными элементами* схемы.

Вершины из V^* называют *выходами* схемы.

Сложностью схемы называют число входящих в нее функциональных элементов.

Пример 1. На рис. 3.87 приведена схема, имеющая два входа, один выход и четыре функциональных элемента. Схема имеет сложность 4.

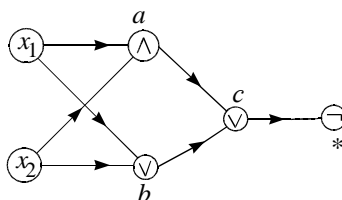


Рис. 3.87.

Глубиной вершины $v \in V$ в схеме $S = (V, U)$ назовем наибольшую из длин путей из входов S в v .

Например, в схеме из примера 1 вершины a и b имеют глубину 1, вершина c - глубину 2, вершина $*$ - глубину 3.

Пусть входы схемы помечены переменными x_1, x_2, \dots, x_n . Каждой вершине $v \in V$ сопоставим булеву функцию $f_v(x_1, x_2, \dots, x_n)$ по следующему индуктивному правилу.

Базис индукции. Пусть v имеет глубину 0. Тогда это вход схемы, который помечен некоторой переменной x_i . Положим $f_v(x_1, x_2, \dots, x_n) = x_i$.

Индуктивный переход. Пусть всем вершинам w глубины, меньшей или равной k , уже сопоставлены функции f_w и пусть v - произвольная вершина глубины $k+1$. Эта вершина принадлежит одному из множеств V_2 или V_3 .

а) Если вершина v принадлежит V_2 , то в нее входит одна дуга, выходящая из некоторой вершины w . Глубина вершины w не превосходит k , и, значит, в силу предположения индукции вершине w уже сопоставлена некоторая булева функция f_w . Вершине v сопоставим отрицание этой функции, т.е. $\overline{f_w}$.

б) Если вершина v принадлежит V_3 , то в нее входят две дуги, выходящие из некоторых вершин w' и w'' , которым в силу предположения индукции уже сопоставлены булевы функции $f_{w'}$ и $f_{w''}$ соответственно. Если вершина v помечена связкой \wedge , то сопоставим ей функцию $f_{w'} \wedge f_{w''}$. Если вершина v помечена связкой \vee , то сопоставим ей функцию $f_{w'} \vee f_{w''}$.

Если вершине v сопоставлена функция $f_v(x_1, x_2, \dots, x_n)$, то говорят, что в v реализуется булева функция f_v .

Схема по определению реализует систему булевых функций, сопоставленных выходам этой схемы.

Пример 2. В соответствии с данным выше определением в вершинах схемы, изображенной на рис. 3.87, реализуются следующие функции:

$$f_a = x_1 \wedge x_2, f_b = x_1 \vee x_2, f_c = f_a \vee f_b = (x_1 \wedge x_2) \vee (x_1 \vee x_2), f_* = \overline{f_c} = \overline{(x_1 \vee x_2) \vee (x_1 \wedge x_2)}.$$

Схема реализует последнюю из перечисленных функций.

Со схемами из функциональных элементов связаны две основные проблемы - проблема анализа и проблема синтеза схемы.

Проблема анализа состоит в том, чтобы по заданной схеме из функциональных элементов и выделенному подмножеству ее выходных вершин определить булевы функции, реализуемые в этих вершинах.

Проблема синтеза заключается в построении схемы из функциональных элементов, реализующей заданную функцию. При этом желательно, чтобы схема имела наименьшую сложность.

Пример 3. Построим схему в базисе $\{\neg, \vee, \wedge\}$, реализующую функцию

$$g(x_1, x_2) = x_1 \downarrow x_2.$$

Вспользуемся тем, что g можно задать формулой

$$g(x_1, x_2) = \overline{x_1 \vee x_2}, \text{ и реализуем ее схемой, изображенной на рис. 3.88.}$$

3.88. Эта схема имеет сложность два. Заметим, что функция f из примера 2 равна функции g из примера 3. Очевидно, что схема на рис. 3.88 имеет наименьшую сложность из всех схем в базисе $\{\neg, \vee, \wedge\}$, реализующих эту функцию.

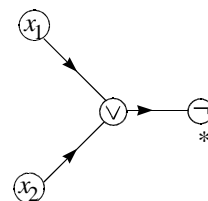


Рис. 3.88.

Упражнение 3.38. Построить схемы в базисе $\{\neg, \vee, \wedge\}$, реализующие основные элементарные функции двух переменных.

Упражнение 3.39. Построить схемы в базисе $\{\neg, \vee, \wedge\}$, реализующие функции:

а) $f = (001001101);$

б) $f = (1110010100101111).$

§ 3.12. Реализация булевых функций с помощью упорядоченных бинарных диаграмм решений

Упорядоченная бинарная диаграмма решений (УБДР). Сток, внутренняя вершина. Сложность УБДР. Минимальная УБДР. Сокращенная УБДР. Правило сокращения. Правило слияния.

Базовые понятия и утверждения

1. Основные понятия. Рассмотрим на примере функции от трех переменных еще один способ задания булевой функции с помощью ориентированных графов.

Построим полное бинарное дерево высотой 3, т.е. ориентированное дерево, все ветви которого имеют длину 3, и из всех вершин которого (за исключением листьев) выходят две дуги. Далее любую вершину дерева, не являющуюся листом, будем называть внутренней вершиной. Пометим внутренние вершины дерева именами переменных: корню дадим метку x_1 , вершинам глубины 1 - метку x_2 , вершинам глубины 3 - метку x_3 . Одну из дуг, выходящих из внутренней вершины v , пометим 0, другую - 1.

Вершину w_0 , в которую ведет дуга, помеченная 0, будем называть 0-сыном вершины v , а вершину w_1 , в которую ведет дуга, помеченная 1, будем называть 1-сыном вершины v .

Каждому набору $(\alpha_1, \alpha_2, \alpha_3)$ значений переменных (x_1, x_2, x_3) сопоставим ветвь дерева, идущую из корня x_1 по дуге с меткой α_1 в вершину x_2 , от вершины x_2 по дуге с меткой α_2 в вершину x_3 , из вершины x_3 по дуге α_3 к листу. Этот лист пометим значением функции $f(x_1, x_2, x_3)$ на наборе $(\alpha_1, \alpha_2, \alpha_3)$, т.е. $f(\alpha_1, \alpha_2, \alpha_3)$. В результате получим так называемое бинарное дерево решений функции $f(x_1, x_2, x_3)$.

Пример 1. Рассмотрим функцию $f(x_1, x_2, x_3) = (00101111)$ (табл. 3.2). На рис. 3.89 дано ее представление в виде дерева решений.

Таблица 3.2

| x | y | z | f |
|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

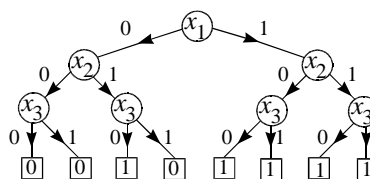


Рис. 3.89.

Преобразуем построенное бинарное дерево решений функции f следующим образом: объединим в одной вершине все листья с меткой 0, а в другой - все листья с меткой 1 (все дуги, которые вели в листья с меткой 0, теперь будут вести в объединенную вершину, помеченную 0, а все дуги, которые вели в листья с меткой 1, будут вести в объединенную вершину, помеченную 1).

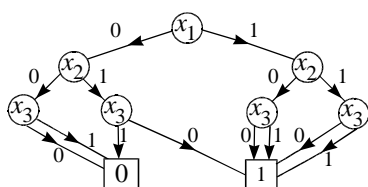


Рис. 3.90.

На рис. 3.90 показан граф, который получился в результате преобразования описанным выше способом бинарного дерева решений функции $f(x_1, x_2, x_3) = (00101111)$. Граф на рис. 3.90 относится к классу ориентированных графов, представителей которого называют

упорядоченными бинарными диаграммами решений.

Определение. Пусть для n переменных x_1, x_2, \dots, x_n зафиксирован порядок π :

$x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}$. Упорядоченной бинарной диаграммой решений относительно порядка

π называется ориентированный граф без циклов с одним корнем, в котором:

- 1) существуют лишь две вершины, в которые не входят дуги; они помечены константами 0 и 1 и называются стоками;
- 2) остальные (внутренние) вершины помечены переменными, и из каждой из них выходят две дуги, одна из этих дуг помечена 0, а другая - 1;

3) порядок, в котором переменные встречаются на любом пути из корня в сток, совместим с π , т.е. если из вершины, помеченной $x_{\pi(i)}$, есть путь в вершину, помеченную $x_{\pi(j)}$, то $i < j$.

Говорят, что УБДР реализует функцию $f(x_1, x_2, \dots, x_n)$, если для каждого набора переменных $(\alpha_1, \alpha_2, \dots, \alpha_n)$ ветвь в дереве, соответствующая этому набору (из вершины $x_{\pi(i)}$ идем по дуге, помеченной $\alpha_{\pi(i)}$), завершается стоком с меткой $f(\alpha_1, \alpha_2, \dots, \alpha_n)$.

Заметим, что из определения не следует, что последовательность вершин произвольного пути от корня к стоку содержит все n переменных $x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}$.

Если путь от корня к стоку не содержит вершины с меткой $x_{\pi(i)}$, то этот путь соответствует и набору, в котором значение $\alpha_{\pi(i)}$ переменной $x_{\pi(i)}$ равно 0, и набору, в котором значение $\alpha_{\pi(i)}$ переменной $x_{\pi(i)}$ равно 1.

Из определения УБДР непосредственно следует, что каждая внутренняя вершина v УБДР является корнем поддиаграммы, которая включает все достижимые из v вершины исходной диаграммы. Пусть вершина v помечена переменной $x_{\pi(k)}$. Тогда внутренние вершины поддиаграммы с корнем v помечены метками $x_{\pi(i)}$, где $i \geq k$, и, значит, поддиаграмма реализует некоторую функцию от переменных

$x_{\pi(k)}, x_{\pi(k+1)}, \dots, x_{\pi(n)}$. Обозначим эту функцию $f_v(x_{\pi(k)}, x_{\pi(k+1)}, \dots, x_{\pi(n)})$. В свою очередь 0-сын w_0 вершины v является корнем поддиаграммы, реализующей функцию

$f_{w_0}(x_{\pi(k+1)}, x_{\pi(k+2)}, \dots, x_{\pi(n)}) = f_v(0, x_{\pi(k+1)}, \dots, x_{\pi(n)})$, а 1-сын w_1 вершины v является корнем поддиаграммы, реализующей функцию $f_{w_1}(x_{\pi(k+1)}, x_{\pi(k+2)}, \dots, x_{\pi(n)}) = f_v(1, x_{\pi(k+1)}, \dots, x_{\pi(n)})$.

Пусть диаграмма реализует функцию $f(x_1, x_2, \dots, x_n) = f'(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)})$ и

$\alpha_{\pi(1)}, \dots, \alpha_{\pi(k-1)}$ - это набор значений переменных $x_{\pi(1)}, \dots, x_{\pi(k-1)}$, который соответствует пути из корня в вершину v (таких наборов может быть несколько). Тогда

$$f_v(x_{\pi(k)}, x_{\pi(k+1)}, \dots, x_{\pi(n)}) = f'(\sigma_{\pi(1)}, \dots, \sigma_{\pi(k-1)}, x_{\pi(k)}, \dots, x_{\pi(n)}).$$

Очевидно, что, зафиксировав порядок переменных, любую булеву функцию можно реализовать в виде УБДР относительно этого порядка. Для этого достаточно действовать так, как описано в примере 1. Однако полученная таким образом УБДР может оказаться не единственной УБДР, реализующей данную функцию.

Пример 2. УБДР, изображенные на рис. 3.91 и 3.92, так же как и УБДР на рис. 3.90, реализуют функцию $f(x_1, x_2, x_3) = (00101111)$.

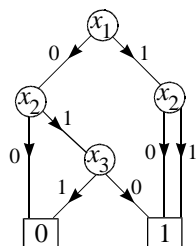


Рис. 3.91.

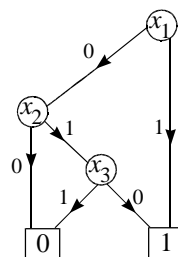


Рис. 3.92.

Под *сложностью* УБДР будем понимать число ее внутренних вершин.

Например, УБДР, изображенные на рис. 3.90, 3.91 и 3.92, имеют соответственно сложность 7, 4 и 3.

Таким образом, одна и та же функция может быть реализована несколькими УБДР относительно одного порядка переменных, причем эти УБДР могут иметь различную сложность. Более того, как мы увидим ниже, изменение порядка следования переменных может существенно влиять на сложность УБДР.

УБДР, имеющая по сравнению с другими УБДР, реализующими данную функцию, наименьшую сложность, называется *минимальной УБДР* данной функции.

2. Построение минимальных УБДР функции относительно заданного порядка переменных. Рассмотрим два правила преобразования УБДР.

Правило удаления вершин. Если 0-сын или 1-сын вершины v совпадают и равны w , то удаляем v , перенаправив все входящие в нее дуги в вершину w (рис. 3.93).

Правило слияния вершин. Если вершины v и w помечены одной переменной и имеют одинаковых 0-сыновей и 1-сыновей, то удаляем вершину v , перенаправив все входящие в нее дуги в вершину w (рис. 3.94).

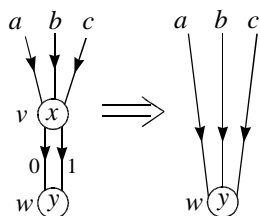


Рис. 3.93.

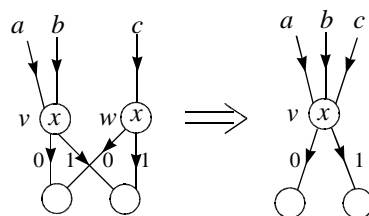


Рис. 3.94.

Определение. УБДР называется *сокращенной*, если выполнены условия:

- 1) у любой внутренней вершины v ее 0-сын и 1-сын не совпадают;

2) нет такой пары внутренних вершин u и v , для которых поддиаграммы с корнями u и v являются изоморфными (т.е. взаимно-однозначно отображаются друг на друга с сохранением всех меток).

Справедливо утверждение, которое мы примем без доказательства.

Теорема 3.12. *УБДР является сокращенной тогда и только тогда, когда к ней не применимо ни правило удаления вершин, ни правило слияния.*

Из этого утверждения непосредственно следует, что, применяя к произвольной УБДР правило сокращения и удаления вершин столько раз, сколько это возможно, мы в итоге получаем сокращенную УБДР.

Опишем эту процедуру в виде **алгоритма построения сокращенной УБДР**.

Пусть для некоторого порядка переменных $x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}$ построена УБДР, реализующая функцию f .

0-й шаг. Применим последовательно ко всем вершинам с метками $x_{\pi(n)}$ данной УБДР вначале правило удаления (перебор вершин с метками $x_{\pi(n-k)}$ можно вести в любом порядке), а затем правило слияния (если это возможно).

k-й шаг. Пусть к началу этого шага построена некоторая УБДР. Применим последовательно ко всем вершинам с метками $x_{\pi(n-k)}$ данной УБДР вначале правило удаления, а затем правило слияния (если это возможно).

По завершении $(n-1)$ -го шага алгоритма получим сокращенную УБДР.

Справедливо утверждение (примем его без доказательства).

Теорема 3.13. *1) Описанный выше алгоритм строит сокращенную УБДР, которая реализует ту же функцию, что и исходная УБДР.*

2) Эта сокращенная УБДР является при данном порядке переменных единственной (с точностью до изоморфизма) и минимальной (т.е. она имеет наименьшую сложность среди всех УБДР данного порядка).

Таким образом, если алгоритм построения сокращенной УБДР применить к УБДР, реализующей некоторую функцию f , то получим минимальную для данного порядка УБДР, реализующую f .

Пример 3. На рис. 3.95 - 3.97 показаны этапы построения минимальной УБДР функции $f(x_1, x_2, x_3) = (10100011)$ относительно порядка x_1, x_2, x_3 с помощью описанного

выше алгоритма. На рис. 3.95 изображена УБДР, полученная из бинарного дерева решений функции. На рис. 3.96 показана УБДР, полученная из УБДР на рис. 3.95 после применения правил удаления и слияния вершин с метками x_3 . На рис. 3.97 изображена УБДР, полученная из УБДР на рис. 3.96 в результате применения правила удаления к вершине с меткой x_2 . УБДР на рис. 3.97 - минимальная УБДР относительно порядка x_1, x_2, x_3 . Ее сложность равна 3.

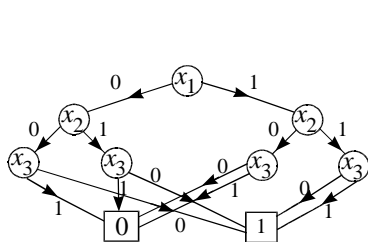


Рис. 3.95.

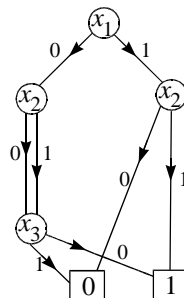


Рис. 3.96.

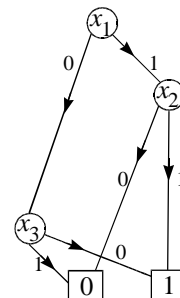


Рис. 3.97.

Пример 4. На рис. 3.89 - 3.92 показаны этапы построения минимальной УБДР функции $f(x_1, x_2, x_3) = (00101111)$ относительно порядка x_1, x_2, x_3 . На рис. 3.90 изображена УБДР функции, полученная из бинарного дерева решений функции. На рис. 3.91 и 3.92 показаны УБДР, полученные по завершении 0-го и 1-го этапов алгоритма. УБДР на рис. 3.92 - минимальная при данном порядке переменных. Ее сложность равна 3.

Пример 5. На рис. 3.98 и 3.99 показаны этапы построения минимальной УБДР функции $f(x_1, x_2, x_3) = (10100011)$ относительно порядка x_2, x_1, x_3 . Минимальная УБДР относительно этого порядка имеет сложность 4 (рис. 3.99).

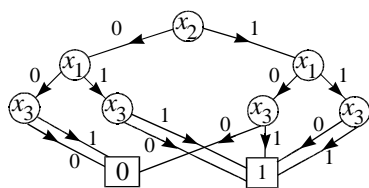


Рис. 3.98.

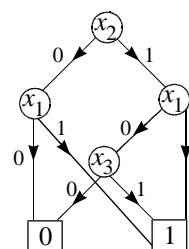


Рис. 3.99.

3. Построение сокращенных УБДР по формулам. Рассмотрим еще один подход к построению сокращенной УБДР - построение по формуле. Опишем его для естественного порядка переменных x_1, x_2, \dots, x_n и произвольной функции $f(x_1, x_2, \dots, x_n)$.

Пусть $(\alpha_1, \dots, \alpha_i)$ - некоторый набор значений переменных (x_1, \dots, x_i) . Функцию $f_{\alpha_1 \dots \alpha_i}(x_{i+1}, \dots, x_n) = f(\alpha_1, \dots, \alpha_i, x_{i+1}, \dots, x_n)$ назовем остаточной функцией.

Построение УБДР начнем с изображения корня, помеченного x_1 . Рассмотрим две остаточные функции: $f_0(x_2, \dots, x_n) = f(0, x_2, \dots, x_n)$ и $f_1(x_2, \dots, x_n) = f(1, x_2, \dots, x_n)$. Если они одинаковы, то f не зависит существенно от переменной x_1 , и тогда изменим метку у корня на x_2 . Если же f существенно зависит от переменной x_1 , то метку x_1 корня сохраним. Если обе функции f_0 и f_1 существенно зависят от x_2 , то из вершины x_1 выпустим две дуги, концам которых дадим метку x_2 и далее реализуем по индукции. Если f_k ($k \in \{0, 1\}$) не зависит существенным образом от переменных x_2, \dots, x_j , но зависит существенно от x_{j+1} , то для этой f_k добавим вершину, помеченную x_{j+1} и проведем в нее дугу с меткой k из корня. Пусть для некоторого i уже построены вершины для всех остаточных функций $f_{\alpha_1 \dots \alpha_i}(x_{i+1}, \dots, x_n) = f(\alpha_1, \dots, \alpha_i, x_{i+1}, \dots, x_n)$, существенно зависящих от x_i . Для каждой из них получим две остаточные функции

$$f_{\alpha_1 \dots \alpha_i 0}(0, x_{i+2}, \dots, x_n) = f(\alpha_1, \dots, \alpha_i, 0, x_{i+2}, \dots, x_n) \text{ и}$$

$$f_{\alpha_1 \dots \alpha_i 1}(1, x_{i+2}, \dots, x_n) = f(\alpha_1, \dots, \alpha_i, 1, x_{i+2}, \dots, x_n).$$

Затем выберем из множества этих функций разные, для каждой из них добавим в диаграмму вершину, помеченную x_{i+1} , и проведем в вершины, помеченные x_{i+1} , соответствующие дуги из вершин, помеченных x_i . Продолжив построение, дойдем до функций от первой переменной x_n и до констант, для которых минимальные реализации очевидны.

Пример 6. Используя описанную процедуру, построить УБДР относительно порядка переменных x_1, x_2, x_3, x_4 для функции f , заданной формулой

$$f(x_1, x_2, x_3, x_4) = \overline{x_1}x_2x_3 \vee x_1\overline{x_2}\overline{x_3} \vee \overline{x_2}x_3 \vee \overline{x_2}x_4.$$

◀ Вначале изобразим корень, пометив его x_1 , и рассмотрим остаточные функции, получающиеся при $x_1 = 0$ и $x_1 = 1$. Имеем:

$$f_0(x_2, x_3, x_4) = f(0, x_2, x_3, x_4) = x_2x_3 \vee \overline{x_2}x_3 \vee \overline{x_2}x_4,$$

$$f_1(x_2, x_3, x_4) = f(1, x_2, x_3, x_4) = \overline{x_2}x_3 \vee \overline{x_2}x_3 \vee \overline{x_2}x_4.$$

Они разные и существенно зависят от x_1 . Поэтому метку корня сохраним и добавим для каждой из функций f_0, f_1 вершину, помеченную x_2 (одна из этих вершин - корень поддиаграммы, реализующей f_0 , другая - корень поддиаграммы, реализующей f_1). Затем для каждой из функций f_0, f_1 определим остаточные функции, получающиеся при $x_2 = 0$ и $x_2 = 1$. Тогда

$$f_{00}(x_3, x_4) = f_0(0, x_3, x_4) = x_3 \vee x_4,$$

$$f_{01}(x_3, x_4) = f_0(1, x_3, x_4) = x_3,$$

$$f_{10}(x_3, x_4) = f_1(0, x_3, x_4) = x_3 \vee x_4,$$

$$f_{11}(x_3, x_4) = f_1(1, x_3, x_4) = \overline{x_3}.$$

Так как $f_{00} = f_{10}$, а f_{01} и f_{11} различны и существенно зависят от x_3 , то нам понадобятся три вершины, помеченные x_3 (первая из этих вершин - корень поддиаграммы реализующей $f_{00} (f_{10})$, вторая - корень поддиаграммы, реализующей f_{01} , третья - корень поддиаграммы, реализующей f_{11}). Для каждой из функций $f_{00} (f_{10})$, f_{01} , f_{11} определим остаточные функции, получающиеся при $x_3 = 0$ и $x_3 = 1$:

$$f_{000}(x_4) = f_{00}(0, x_4) = x_4,$$

$$f_{001}(x_4) = f_{00}(1, x_4) = 1,$$

$$f_{010}(x_4) = f_{01}(0, x_4) = 0,$$

$$f_{011}(x_4) = f_{01}(1, x_4) = 1,$$

$$f_{110}(x_4) = f_{11}(0, x_4) = 1,$$

$$f_{111}(x_4) = f_{11}(1, x_4) = 0.$$

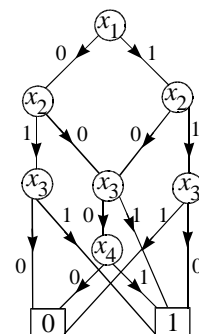


Рис. 3.100.

Так как только одна из этих функций существенно зависит от x_4 , то нам понадобится лишь одна вершина, помеченная x_4 (она будет корнем поддиаграммы, реализующей функцию f_{000}). В результате получается УБДР, изображенная на рис. 3.100. ►

Задачи повышенной сложности

3.43. Построить минимальные УБДР функции $f(x, y, z) = (00101111)$ относительно порядка:

- а) x, y, z ; б) y, x, z .

3.44. Построить минимальную УБДР относительно порядка x_1, x_2, x_3, x_4 для функции

$f(x_1, x_2, x_3, x_4)$, заданной формулой:

а) $f = x_1 x_2 \oplus x_2 x_3 \vee x_3 x_4$;

б) $f = \overline{x_1} x_2 x_4 \vee \overline{x_1} x_2 \overline{x_4} \vee x_1 x_3 \vee x_1 x_4$.

Глава 4. Элементы теории автоматов

§ 4.1. Ограниченно-детерминированные функции

Сверхслово. Детерминированная функция. Информационное дерево.
Ограниченно-детерминированные функции. Диаграмма Мура ограниченно-детерминированной функции. Таблица ограниченно-детерминированной функции. Канонические уравнения ограниченно-детерминированной функции.

Базовые понятия и утверждения

1. Детерминированные функции. Пусть A - конечное множество. Назовем это множество *алфавитом*, а его элементы - *буквами*.

Определение. Любая бесконечная последовательность элементов множества A называется *сверхсловом* над алфавитом A .

Для обозначения сверхслов используют запись $\alpha = \alpha(1)\alpha(2)\alpha(3)\dots$, где $\forall i \quad \alpha(i) \in A$.

Пусть A^∞ и B^∞ - множества сверхслов над алфавитами A и B соответственно.

Рассмотрим функции, определенные на множестве A^∞ и принимающие значения из B^∞ .

Определение. Функция $f: A^\infty \rightarrow B^\infty$ называется *детерминированной*, если для любого натурального числа t и для любых сверхслов α и α' из A^∞ , у которых первые t букв совпадают, верно, что в сверхсловах $\beta = f(\alpha)$ и $\beta' = f(\alpha')$ первые t букв также совпадают (т.е. если $\alpha(1) = \alpha'(1)$, $\alpha(2) = \alpha'(2)$, ..., $\alpha(t) = \alpha'(t)$, то и $\beta(1) = \beta'(1)$, $\beta(2) = \beta'(2)$, ..., $\beta(t) = \beta'(t)$).

Пример 1. Функция $f(\alpha(1)\alpha(2)\alpha(3)\alpha(4)\dots) = \alpha(2)\alpha(3)\alpha(4)\dots$ детерминированной не является.

Пример 2. Функция $f: \{0,1\}^\infty \rightarrow \{0,1\}^\infty$ такая, что

$$f(\alpha(1)\alpha(2)\alpha(3)\alpha(4)\alpha(5)\dots) = \overline{\alpha(1)}\alpha(2)\overline{\alpha(3)}\alpha(4)\overline{\alpha(5)}\dots$$

($\beta(i) = \overline{\alpha(i)}$, если $i = 2n-1$, $n \in \mathbf{N}$ и $\beta(i) = \alpha(i)$, если $i = 2n$, $n \in \mathbf{N}$), является детерминированной.

Пример 3. Функция $f: \{0,1\}^\infty \rightarrow \{0,1\}^\infty$, такая что

$$f(\alpha(1)\alpha(2)\dots) = \beta(1)\beta(2)\dots \text{ и } \forall i \quad \beta(i) = \alpha(1) \oplus \alpha(2) \oplus \dots \oplus \alpha(i),$$

является детерминированной.

Упражнение 4.1. Выяснить, является ли функция $f: \{0,1\}^\infty \rightarrow \{0,1\}^\infty$ детерминированной:

- а) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1)\alpha(3)\alpha(5)\dots$;
- б) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1)\alpha(1)\alpha(2)\alpha(3)\dots$, т.е. $\beta(1) = \alpha(1)$ и при любом $i \in \mathbf{N}$ $\beta(i+1) = \alpha(i)$;
- в) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1), \alpha(1) \vee \alpha(2), \alpha(1) \vee \alpha(2) \vee \alpha(3), \dots$.

Пусть дан алфавит A , $|A| = n$. Рассмотрим бесконечное корневое ориентированное дерево, изображенное на рис. 4.1.

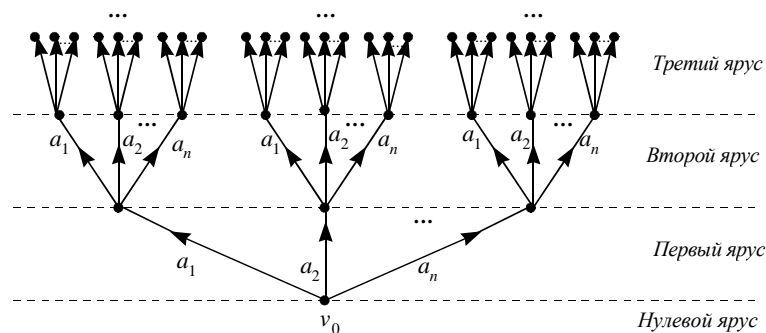


Рис. 4.1.

Из корня v_0 этого дерева исходит пучок из n дуг, образующих первый ярус. Каждая из дуг первого яруса ведет в вершину, из которой в свою очередь исходит пучок из n дуг, образующих второй ярус, и т.д. Вершины, являющиеся концами дуг яруса n , причисляются также к ярусу n . Вершина v_0 считается вершиной нулевого яруса.

Упорядочим множество A , после чего дуги каждого пучка пометим буквами алфавита A слева направо, согласно введенному порядку. Получим помеченное дерево T .

Бесконечный путь с началом в корне v_0 назовем *ветвью* рассматриваемого дерева.

Каждой ветви дерева можно сопоставить сверхслово

$\alpha = \{\alpha(1), \alpha(2), \alpha(3), \dots\} = \alpha(1)\alpha(2)\alpha(3)\dots$ - последовательность меток дуг, входящих в эту

ветвь, если идти по ней, начиная от корня. И наоборот, каждому сверхслову из A^∞ однозначно соответствует некоторая ветвь дерева. Таким образом, существует взаимно-однозначное соответствие между множеством всех ветвей дерева и сверхсловами из A^∞ .

Пусть $f: A^\infty \rightarrow B^\infty$ - произвольная детерминированная функция. Возьмем любую дугу яруса t ($t = 1, 2, 3, \dots$) и рассмотрим путь, ведущий из корня дерева к этой дуге.

Очевидно, что этот путь определен однозначным образом и характеризуется набором $(\alpha(1), \alpha(2), \alpha(3), \dots, \alpha(t))$ букв, которыми помечены его дуги. Возьмем любое сверхслово α , первые t членов которого совпадают с набором $(\alpha(1), \alpha(2), \alpha(3), \dots, \alpha(t))$ и припишем исходной дуге t -ю букву $\beta(t)$ сверхслова $\beta = f(\alpha)$. В силу того, что функция f - детерминированная, буква $\beta(t)$ определена однозначно. В итоге каждая дуга дерева будет помечена двумя метками: буквой сверхслова α и, через запятую, буквой сверхслова β . Дерево, дуги которого помечены вышеописанным способом, назовем *информационным деревом*, соответствующим детерминированной функции f , и обозначим его T_f .

Таким образом, имея произвольную детерминированную функцию, можно построить соответствующее ей информационное дерево.

С другой стороны, возьмем помеченное дерево T . Каждой дуге дерева дадим вторую метку, взяв в качестве таковой произвольную букву алфавита B . Ясно, что полученное дерево однозначно определит некоторую детерминированную функцию $f: A^\infty \rightarrow B^\infty$.

Следовательно, можно говорить о существовании взаимно-однозначного соответствия между детерминированными функциями и информационными деревьями.

Пример 4. На рис. 4.2 изображено информационное дерево функции $f: \{0,1\}^\infty \rightarrow \{0,1\}^\infty$, определенной правилом

$$f(\alpha(1)\alpha(2)\alpha(3)\alpha(4)\alpha(5)\dots) = \overline{\alpha(1)}\alpha(2)\overline{\alpha(3)}\alpha(4)\overline{\alpha(5)}\dots$$

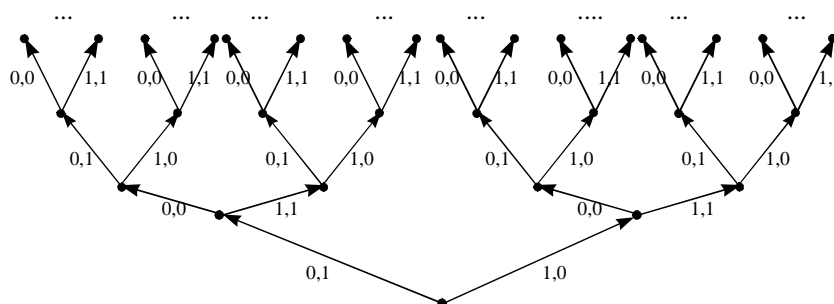


Рис. 4.2.

Пример 5. На рис. 4.3 изображено информационное дерево функции $f: \{0,1\}^\infty \rightarrow \{0,1\}^\infty$, определенной правилом

$$f(\alpha(1)\alpha(2)\dots) = \beta(1)\beta(2)\dots \text{ и } \forall i \quad \beta(i) = \alpha(1) \oplus \alpha(2) \oplus \dots \oplus \alpha(i).$$

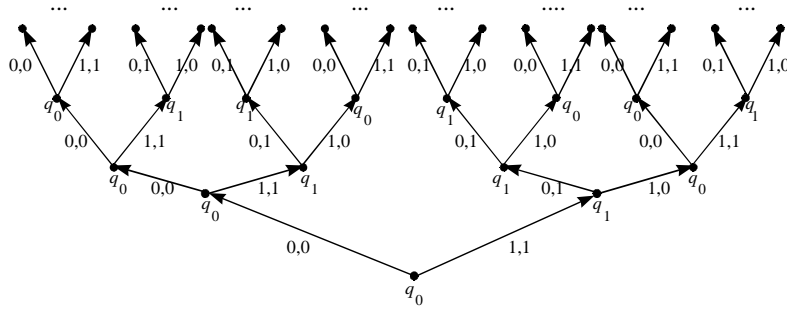


Рис. 4.3.

Упражнение 4.2. Выяснить, является ли функция $f : \{0,1\}^\infty \rightarrow \{0,1\}^\infty$ ограниченно-детерминированной, и найти ее вес:

- а) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1)\alpha(1)\alpha(2)\alpha(3)\dots$;
- б) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1), \alpha(1) \vee \alpha(2), \alpha(1) \vee \alpha(2) \vee \alpha(3), \dots$.

Пусть f - детерминированная функция, T_f - информационное дерево функции f , v - произвольная вершина яруса t дерева T_f и $\alpha(1), \alpha(2), \alpha(3), \dots, \alpha(t)$ - путь, который ведет из корня v_0 в вершину v . Совокупность всех ветвей, исходящих из v , порождает некоторое дерево с корнем v (являющееся поддеревом исходного дерева). Так как исходное дерево дважды помечено, то поддерево с корнем v также является дважды помеченным, и если в этом поддереве вести нумерацию ярусов, начиная с нулевого, то о нем можно говорить как об информационном дереве, которому соответствует некоторая детерминированная функция f^v .

На множестве вершин дерева T_f введем бинарное отношение, которое определим следующим образом: $v \sim v' \Leftrightarrow f^v = f^{v'}$. Это отношение является отношением эквивалентности и порождает разбиение множества вершин дерева на классы эквивалентности. Число s различных классов эквивалентности по данному бинарному отношению называется *весом дерева*, и соответственно, *весом детерминированной функции*. Классы эквивалентности назовем *состояниями функции*.

Пометим вершины информационного дерева T_f . Сначала пронумеруем классы эквивалентности так, чтобы класс, в который попадет исходное дерево, имел номер 0. Далее, взяв произвольную вершину v , определим номер класса, в который попадет дерево с корнем v . Пусть i - номер этого класса, тогда вершину v пометим q_i .

Рассмотрим дерево T_f с помеченными вершинами. Возьмем произвольную ветвь; пусть она проходит через вершины $v_0, v_1, \dots, v_i, \dots, v_j, \dots$ и эти вершины имеют соответственно

метки $q_0, q_1, \dots, q_i, \dots, q_j, \dots$. Допустим метки q_i и q_j ($i \neq j$) совпадают, причем в рассматриваемой последовательности меток это первая пара совпадающих меток. Произведем усечение данной ветви, сохранив ее начальный отрезок до вершины v_j . Произведя эту операцию для каждой ветви, получим *усеченное дерево*. Легко видеть, что по усеченному дереву однозначно восстанавливается исходное дерево.

2. Ограниченно-детерминированные функции. Рассмотрим случай, когда детерминированная функция имеет конечный вес.

Определение. Детерминированные функции, имеющие конечный вес, называются *ограниченно-детерминированными*.

Для случая функции конечного веса s на каждой ветке происходит повторение меток вершин, и индекс j , определяющий усечение, удовлетворяет неравенству $j \leq s$. Поэтому для этих функций усеченное дерево будет конечным. Таким образом, в случае ограниченно-детерминированной функции полное (бесконечное) информационное дерево можно всегда свести к конечному дереву с дважды помеченными дугами и помеченными вершинами. Если в этом усеченном дереве произвести отождествление вершин с одинаковыми именами, а также отождествление кратных дуг с одинаковыми двойными метками, то получим так называемую *диаграмму переходов* ограниченно-детерминированной функции или, как ее еще называют, *диаграмму Мура*.

Пример 6. На рис. 4.4 и 4.5 показаны усеченное дерево и диаграмма переходов функции, рассмотренной в примерах 2, 4.

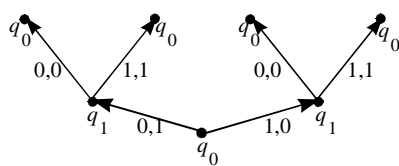


Рис. 4.4.

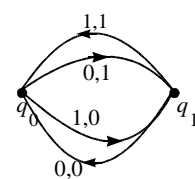


Рис. 4.5.

Пример 7. На рис. 4.6 и 4.7 показаны усеченное дерево и диаграмма переходов функции, рассмотренной в примерах 3, 5.

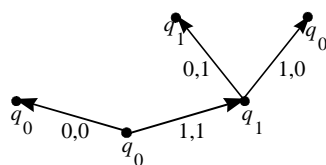


Рис. 4.6.

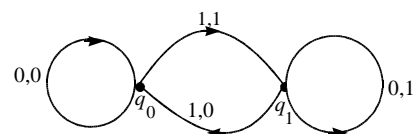


Рис. 4.7.

По диаграмме Мура ограниченно-детерминированной функции построим *таблицу ограниченно-детерминированной функции*. Пусть $f : A^\infty \rightarrow B^\infty$ - произвольная ограниченно-детерминированная функция веса s . Ее диаграмма Мура имеет s вершин; из каждой вершины q_i ($i = 1, 2, \dots, s$) исходит $n = |A|$ дуг, дугам приписаны пары $(a_1, b_{p_{i1}})$, $(a_2, b_{p_{i2}})$, $(a_n, b_{p_{in}})$, где a_j , $b_{p_{ij}}$ - буквы алфавитов A и B соответственно; выделена начальная вершина. Построим таблицу, строки которой соответствуют различным вершинам диаграммы Мура, а столбцы - различным буквам алфавита A . Ячейки таблицы заполним по следующему правилу. Пусть ячейка, которую мы заполняем, соответствует вершине q_i и букве a_j . На диаграмме Мура отыщем дугу с началом в вершине q_i и первой меткой a_j , после чего запишем в ячейку имя конца этой дуги $q_{r_{ij}}$ и вторую метку $b_{p_{ij}}$ самой дуги. Метку начальной вершины в заголовке строки выделим.

По сути, диаграммой Мура и таблицей задаются две функции: функция ϕ , которая каждой упорядоченной паре q_i, a_j ставит в соответствие состояние $q_{r_{ij}}$, $q_{r_{ij}} = \phi(q_i, a_j)$, и функция ϕ , которая каждой упорядоченной паре q_i, a_j ставит в соответствие букву $b_{p_{ij}}$, $b_{p_{ij}} = \phi(q_i, a_j)$.

Пример 8. Табл. 4.1 - это таблица ограниченно-детерминированной функции, рассмотренной в примерах 2, 4, 6.

Пример 9. Табл. 4.2 - это таблица ограниченно-детерминированной функции, рассмотренной в примерах 3, 5, 7.

Таблица 4.1

| $\begin{array}{c} A \\ \diagdown \\ Q \end{array}$ | 0 | 1 |
|--|--|--|
| q_0 | $\begin{array}{c} q_1 \\ \diagup \\ 1 \end{array}$ | $\begin{array}{c} q_1 \\ \diagup \\ 0 \end{array}$ |
| q_1 | $\begin{array}{c} q_0 \\ \diagup \\ 0 \end{array}$ | $\begin{array}{c} q_0 \\ \diagup \\ 1 \end{array}$ |

Таблица 4.2

| $\begin{array}{c} A \\ \diagdown \\ Q \end{array}$ | 0 | 1 |
|--|--|--|
| q_0 | $\begin{array}{c} q_0 \\ \diagup \\ 0 \end{array}$ | $\begin{array}{c} q_1 \\ \diagup \\ 1 \end{array}$ |
| q_1 | $\begin{array}{c} q_1 \\ \diagup \\ 1 \end{array}$ | $\begin{array}{c} q_0 \\ \diagup \\ 0 \end{array}$ |

Упражнение 4.3. Построить диаграмму Мура и таблицу ограниченно-детерминированной функции $f: \{0,1\}^\infty \rightarrow \{0,1\}^\infty$:

а) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1)\alpha(1)\alpha(2)\alpha(3)\dots$;

б) $f(\alpha(1)\alpha(2)\alpha(3)\dots) = \alpha(1), \alpha(1) \vee \alpha(2), \alpha(1) \vee \alpha(2) \vee \alpha(3), \dots$.

И по диаграмме переходов, и по таблице информационное дерево, а значит, и сама ограниченно-детерминированная функция $f: A^\infty \rightarrow B^\infty$ восстанавливается однозначно. Опишем процедуру восстановления.

Возьмем помеченное дерево T алфавита A . Мы восстановим информационное дерево функции f , если каждую дугу дерева T пометим второй меткой. Вторые метки дуг будем расставлять пошагово: на первом шаге пометим дуги первого яруса, на втором - второго и т.д. Нам будет проще, если мы на произвольном шаге t будем помечать не только дуги, но и вершины яруса t . Для удобства рассуждений для каждой ветви восстанавливаемого дерева введем в рассмотрение последовательность $q(0), q(1), \dots, q(t), \dots$ меток ее вершин. При расстановке меток дуг и вершин будем придерживаться следующего алгоритма.

0-й шаг. Пометим вершину нулевого яруса, присвоив ей имя выделенной вершины диаграммы Мура q_0 : $q(0) = q_0$.

t-й шаг. К началу этого шага мы поместили дуги и вершины всех ярусов, начиная с нулевого и кончая $(t-1)$ -м. Возьмем произвольную дугу яруса t ; пусть a' - ее первая метка, т.е. $\alpha(t) = a'$. Вершина, из которой эта дуга выходит, принадлежит $(t-1)$ -му ярусу и, значит, ей уже дано какое-то имя q' , т.е. $q(t-1) = q'$. На диаграмме Мура отыщем вершину с именем q' и выходящую из нее дугу с первой меткой a' . Пусть вершина q'' - конец этой дуги, а b' - ее вторая метка. Тогда рассматриваемой дуге дерева дадим вторую метку b' , т.е. $\beta(t) = b'$, а ее концу дадим имя q'' , т.е. $q(t) = q''$.

Таким образом, шаг за шагом информационное дерево функции f будет восстановлено.

Чтобы задать ограниченно-детерминированную функцию, нужно указать правило, по которому каждой последовательности $\alpha = \alpha(1)\alpha(2)\alpha(3)\dots$ ставится в соответствие последовательность $\beta = \beta(1)\beta(2)\beta(3)\dots$. Процедура восстановления информационного дерева функции f по диаграмме Мура наводит на мысль, что, поскольку на каждом шаге восстановления $\beta(t)$ однозначно определялось по $q(t-1)$ и $\alpha(t)$, то

последовательность $\beta = \beta(1)\beta(2)\beta(3)\dots$ можно задавать рекуррентно, используя при этом рекуррентное же задание последовательности $q(0), q(1), \dots, q(t), \dots$. Таким образом, приходим к следующим уравнениям:

$$\begin{cases} \beta(t) = \phi(q(t-1), \alpha(t)), \\ q(t) = \phi(q(t-1), \alpha(t)), \\ q(0) = q_0. \end{cases}$$

Данные уравнения называются *каноническими уравнениями ограниченной детерминированной функции* и могут рассматриваться как один из способов ее задания.

Пример 10. Найдем канонические уравнения для функции, рассмотренной в примерах 2, 4, 6, 8.

Положим $q_0 = 0$, $q_1 = 1$. Значения $q(t)$ и $b(t)$ будут определяться по значениям $a(t)$ и $q(t-1)$ согласно табл. 4.3.

Таблица 4.3 Значит, канонические уравнения имеют вид:

| $a(t)$ | $q(t-1)$ | $q(t)$ | $b(t)$ |
|--------|----------|--------|--------|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

$$\begin{cases} \beta(t) = \overline{q(t-1)}, \\ q(t) = q(t-1) \leftrightarrow a(t), \\ q(0) = 0. \end{cases}$$

Пример 11. Определим канонические уравнения для функции, рассмотренной в примерах 3, 5, 7, 9.

Положим $q_0 = 0$, $q_1 = 1$. Значения $q(t)$ и $b(t)$ будут определяться по значениям $a(t)$ и $q(t-1)$ согласно табл. 4.4.

Таблица 4.4 Значит, канонические уравнения имеют вид:

| $a(t)$ | $q(t-1)$ | $q(t)$ | $b(t)$ |
|--------|----------|--------|--------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

$$\begin{cases} \beta(t) = q(t-1) \oplus a(t), \\ q(t) = q(t-1) \oplus a(t), \\ q(0) = 0. \end{cases}$$

§ 4.2. Реализация ограниченно-детерминированных функций конечными автоматами

Конечный автомат Мили. Задание автомата таблицей, диаграммой Мура, каноническими уравнениями. Продолжение функций переходов и выходов на слова. Отличимость состояний автомата. Приведенный автомат.

Базовые понятия и утверждения

1. Конечный автомат Мили, способы его задания. Рассмотрим произвольную ограниченно-детерминированную функцию $f : A^{\infty} \rightarrow B^{\infty}$. Функцию f можно интерпретировать как функцию, описывающую работу дискретного преобразователя информации. На входы этого преобразователя в моменты времени $t = 1, 2, 3, \dots$ подаются члены последовательности $\alpha = \alpha(1)\alpha(2)\alpha(3)\dots$. В эти же моменты времени на выходе возникают члены последовательности $\beta = \beta(1)\beta(2)\beta(3)\dots$, где $\beta(1)\beta(2)\beta(3)\dots = f(\alpha(1)\alpha(2)\alpha(3)\dots)$. Такой дискретный преобразователь называют конечным автоматом, реализующим функцию f . При этом $\alpha = \alpha(1)\alpha(2)\alpha(3)\dots$ называют входной последовательностью, $\beta = \beta(1)\beta(2)\beta(3)\dots$ - выходной последовательностью, буквы $\alpha(t)$ и $\beta(t)$ - сигналами, поступившими в момент времени t соответственно на вход и выход автомата.

При формальном определении автомата будем исходить из задания ограниченно-детерминированной функции функциями ϕ и φ .

Определение. Конечным автоматом Мили называется шестерка объектов

$$K = (A, B, Q, \phi, \varphi, q_0), \text{ где}$$

A - конечное непустое множество (входной алфавит);

B - конечное непустое множество (выходной алфавит);

Q - конечное непустое множество (множество состояний);

$\phi : Q \times A \rightarrow Q$ - отображение, называемое функцией переходов;

$\varphi : Q \times A \rightarrow B$ - отображение, называемое функцией выходов;

$q_0 \in Q$ - начальное состояние.

Можно рассматривать автоматы, у которых начальное состояние не задано. Такие автоматы называют неинициальными.

Способы задания конечных автоматов те же, что и способы задания ограниченно-детерминированных функций.

1. Конечный автомат можно задать таблицей. Табл. 4.5 задает функции переходов ϕ и выходов φ автомата с алфавитами A , B и множеством состояний Q . Дополнительно указывается начальное состояние автомата.

Таблица 4.5

| $Q \backslash A$ | a_1 | a_2 | \dots | a_n |
|------------------|---|---|---------|---|
| q_1 | $\phi(q_1, a_1)$ $\varphi(q_1, a_1)$ | $\phi(q_1, a_2)$ $\varphi(q_1, a_2)$ | \dots | $\phi(q_1, a_n)$ $\varphi(q_1, a_n)$ |
| q_2 | $\phi(q_2, a_1)$ $\varphi(q_2, a_1)$ | $\phi(q_2, a_2)$ $\varphi(q_2, a_2)$ | \dots | $\phi(q_2, a_n)$ $\varphi(q_2, a_n)$ |
| \dots | \dots | \dots | \dots | \dots |
| q_s | $\phi(q_s, a_1)$ $\varphi(q_s, a_1)$ | $\phi(q_s, a_2)$ $\varphi(q_s, a_2)$ | \dots | $\phi(q_s, a_n)$ $\varphi(q_s, a_n)$ |

2. Конечный автомат можно задать с помощью диаграммы Мура, т.е. геометрического графа, определенного следующим образом: каждому состоянию автомата сопоставляется вершина графа; если для двух состояний q_i , q_j существует входной сигнал a такой, что $q_j = \phi(q_i, a)$, то из вершины q_i в вершину q_j проводится дуга, помечаемая двойной меткой a, b , где $b = \varphi(q_i, a)$.

Чтобы полностью определить автомат, нужно также указать на графе вершину, соответствующую его начальному состоянию.

3. Автомат можно задать системой канонических уравнений:

$$\begin{cases} \beta(t) = \varphi(q(t-1), \alpha(t)), \\ q(t) = \phi(q(t-1), \alpha(t)), \\ q(0) = q_0. \end{cases}$$

Рассмотрим некоторые частные виды конечных автоматов.

Автомат без памяти - автомат, множество состояний которого включает в себя ровно один элемент, т.е. $Q = \{q\}$. Тогда функция переходов ϕ всегда принимает одно и то же значение q , и ее можно не рассматривать. Функция выходов ϕ зависит фактически только от входного сигнала. Работу автомата можно описать только одним каноническим уравнением: $\beta(t) = \phi(q, \alpha(t))$. Так, в частности, работает автомат, который осуществляет перекодировку символов из одного алфавита в другой.

Элемент задержки - автомат, у которого входной и выходной алфавиты совпадают, причем $\beta(t+1) = \alpha(t)$ для любого момента времени t , $t = 1, 2, 3, \dots$. Несложно построить канонические уравнения такого автомата, задав их, например, так:

$$\begin{cases} \beta(t) = q(t-1), \\ q(t) = \alpha(t), \\ q(0) = q_0. \end{cases}$$

Если же мы хотим описать такой автомат, исходя из определения, то будем иметь:

$$A = B = Q; \quad \phi(q, a) = a, \quad \phi(q, a) = q.$$

Пример 1. Рассмотрим частный случай элемента задержки $A = B = \{0, 1\}$.

Табл. 4.6 - таблица этого автомата. Диаграмма Мура автомата изображена на рис. 4.8.

Таблица 4.6

| $\begin{matrix} A \\ Q \end{matrix}$ | 0 | 1 |
|--------------------------------------|--------|--------|
| 0 | 0 0 | 1 0 |
| 1 | 0 1 | 1 1 |

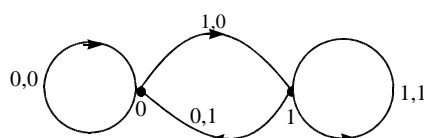


Рис. 4.8.

В качестве начального состояния элемента задержки можно взять любое из его состояний. Пусть начальное состояние автомата 0 и на его вход подается последовательность 10101010..., тогда на выходе автомата наблюдается 01010101...

Конечный автомат Мили определен таким образом, что каждый автомат реализует конкретную ограниченно-детерминированную функцию, однако одна и та же

ограниченно-детерминированная функция может быть реализована различными автоматами. Остановимся на этом вопросе подробнее.

2. Продолжение функций переходов и выходов на слова. *Словом* длины k , $k = 0, 1, 2, \dots$, над алфавитом A называется любая конечная последовательность длины k элементов множества A .

Для обозначения слов используют запись $w = a_{i_1} a_{i_2} \dots a_{i_k}$.

Слово, в котором нет ни одной буквы, называют *пустым* словом и обозначают символом λ .

Введем ряд обозначений:

A^* - множество всех слов над алфавитом A ;

A^m - множество всех слов длины m ;

A^+ - множество всех непустых слов.

Очевидно, выполнены равенства: $A^* = \bigcup_{m=0}^{+\infty} A^m$, $A^+ = \bigcup_{m=1}^{+\infty} A^m$.

Произведением двух слов $w_1 = a_{i_1} a_{i_2} \dots a_{i_k}$ и $w_2 = a_{j_1} a_{j_2} \dots a_{j_p}$ называется слово

$w = a_{i_1} a_{i_2} \dots a_{i_k} a_{j_1} a_{j_2} \dots a_{j_p}$.

Произведение двух слов - ассоциативная операция, т.е. для любых трех слов w_1, w_2, w_3 выполняется равенство $(w_1 w_2) w_3 = w_1 (w_2 w_3)$.

Поскольку расстановка скобок в произведении не влияет на результат, то, записывая произведение нескольких слов, скобки опускают. Например, вместо $(w_1 w_2) w_3$ пишут $w_1 w_2 w_3$.

Заметим, что произведение слов зависит от порядка сомножителей, т.е. в общем случае $w_1 w_2 \neq w_2 w_1$.

Определение. *Продолжением функций ϕ и $\bar{\phi}$ на слова* называются соответственно функции $\bar{\phi}: Q \times A^* \rightarrow Q$ и $\bar{\phi}: Q \times A^* \rightarrow B^*$, определенные индуктивно следующим образом:

$$\forall a \in A \quad \bar{\phi}(q, a) = \phi(q, a);$$

$$\bar{\phi}(q, \lambda) = q;$$

$$\forall a \in A, w \in A^+ \quad \bar{\phi}(q, aw) = \bar{\phi}(\phi(q, a), w);$$

$$\forall a \in A \quad \bar{\phi}(q, a) = \phi(q, a);$$

$$\forall a \in A, w \in A^+ \quad \bar{\phi}(q, aw) = \phi(q, a) \bar{\phi}(\phi(q, a), w);$$

$$\bar{\phi}(q, \lambda) = \lambda.$$

Пример 2. Рассмотрим автомат

$K = (A, B, Q, \phi, \bar{\phi}, q_0)$, с алфавитами $A = \{a, b, c\}$ и $B = \{0, 1\}$, множеством состояний $Q = \{q_1, q_2\}$, начальным состоянием $q_0 = q_2$ и функциями ϕ и $\bar{\phi}$, представленными в табл. 4.7.

Тогда

$$\begin{aligned} \bar{\phi}(q_2, cba) &= \bar{\phi}(\phi(q_2, c), ba) = \bar{\phi}(q_2, ba) = \bar{\phi}(\phi(q_2, b), a) \\ &= \bar{\phi}(q_1, a) = q_1; \end{aligned}$$

$$\bar{\phi}(q_2, cba) = \phi(q_2, c) \bar{\phi}(\phi(q_2, c), ba) =$$

$$\begin{aligned} &= \phi(q_2, c) \bar{\phi}(q_2, ba) = \phi(q_2, c) \phi(q_2, b) \bar{\phi}(\phi(q_2, b), a) = \\ &= \phi(q_2, c) \phi(q_2, b) \bar{\phi}(q_1, a) = \phi(q_2, c) \phi(q_2, b) \phi(q_1, a) = 010. \end{aligned}$$

Таблица 4.7

| $\begin{matrix} A \\ Q \end{matrix}$ | a | b | c |
|--------------------------------------|--|--|--|
| q_1 | $\begin{matrix} q_1 \\ 0 \end{matrix}$ | $\begin{matrix} q_2 \\ 0 \end{matrix}$ | $\begin{matrix} q_2 \\ 1 \end{matrix}$ |
| q_2 | $\begin{matrix} q_2 \\ 1 \end{matrix}$ | $\begin{matrix} q_1 \\ 1 \end{matrix}$ | $\begin{matrix} q_2 \\ 0 \end{matrix}$ |

Упражнение 4.4. Работа конечного автомата задана диаграммой Мура (рис. 4.9).

Найти:

а) $\bar{\phi}(q_1, aabab)$; б) $\bar{\phi}(q_3, abba)$;

в) $\bar{\phi}(q_2, bba)$; г) $\bar{\phi}(q_1, bbba)$.

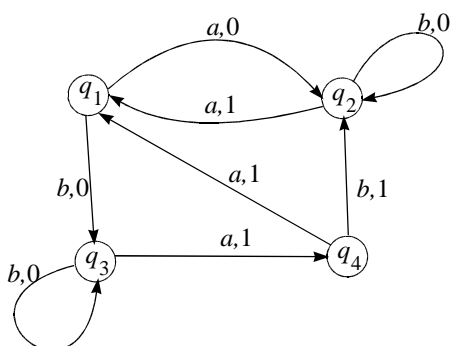


Рис. 4.9.

3. Приведенный автомат. Состояния q_1 и q_2 автомата называются *неотличимыми*, если для $\forall w \in A^+ \quad \bar{\phi}(q_1, w) = \bar{\phi}(q_2, w)$. В противном случае состояния q_1 и q_2 называются *отличимыми*.

Введем на множестве Q состояний автомата бинарное отношение неотличимости \sim , включив в него все пары неотличимых состояний.

Это отношение является отношением эквивалентности. Следовательно, оно порождает разбиение множества Q на классы эквивалентности. Следуя определению классов эквивалентности, класс эквивалентности произвольного элемента q по отношению неотличимости \sim определим как множество $[q]_{\sim} = \{q' \in Q \mid q \sim q'\}$. Множество классов эквивалентности по отношению неотличимости обозначим \mathcal{Q} , для его элементов будем использовать обозначение \mathcal{Q} .

Определение. Приведенным автоматом, соответствующим конечному автомату

$K = (A, B, Q, \phi, \varphi, q_0)$, называется автомат $\tilde{K} = (A, B, \mathcal{Q}, \mathcal{\phi}, \mathcal{\varphi}, \tilde{q}_0)$, функция переходов

$\mathcal{\phi}: \mathcal{Q} \times A \rightarrow \mathcal{Q}$ и функция выходов $\mathcal{\varphi}: \mathcal{Q} \times A \rightarrow B$ которого определены следующим образом: $\mathcal{\phi}(\mathcal{Q}, a) = [\phi(q, a)]_{\sim}$; $\mathcal{\varphi}(\mathcal{Q}, a) = \varphi(q, a)$, где $q \in \mathcal{Q}$.

Докажем корректность этого определения, т.е. его независимость от выбора представителя q в классе эквивалентности \mathcal{Q} .

1. Чтобы доказать корректность определения функции $\mathcal{\phi}$, будем рассуждать от противного. Предположим, что найдутся такие $q_1 \in \mathcal{Q}$ и $q_2 \in \mathcal{Q}$, что $[\phi(q_1, a)]_{\sim} \neq [\phi(q_2, a)]_{\sim}$. Это означает, что состояния $\phi(q_1, a)$ и $\phi(q_2, a)$ отличимы, и, значит, $\exists w \in A^+$, для которого $\bar{\phi}(\phi(q_1, a), w) \neq \bar{\phi}(\phi(q_2, a), w)$. Но тогда $\bar{\phi}(q_1, aw) \neq \bar{\phi}(q_2, aw)$. Следовательно, q_1 и q_2 отличимы, а это противоречит тому, что q_1 и q_2 принадлежат одному и тому же классу эквивалентности \mathcal{Q} .

2. Докажем корректность определения функции $\mathcal{\varphi}$. Возьмем $\forall q_1 \in \mathcal{Q}$ и $\forall q_2 \in \tilde{q}$; q_1 и q_2 неотличимы, следовательно, $\varphi(q_1, a) = \varphi(q_2, a)$. Таким образом, функция $\mathcal{\varphi}$ определена корректно.

Утверждение. Все состояния приведенного автомата попарно отличимы.

Доказательство. Будем рассуждать от противного. Пусть найдутся различные $\mathcal{Q} \in \mathcal{Q}$ и $\mathcal{Q}' \in \mathcal{Q}$ такие, что $\mathcal{Q} \sim \mathcal{Q}'$. Тогда $\forall w \in A^+ \quad \bar{\mathcal{\phi}}(\mathcal{Q}, w) = \bar{\mathcal{\phi}}(\mathcal{Q}', w)$. Согласно определению функции $\mathcal{\phi}$, будем иметь $\bar{\phi}(q_1, w) = \bar{\phi}(q_2, w)$, где $q_1 \in \mathcal{Q}$, $q_2 \in \mathcal{Q}'$. Отсюда следует, что $q_1 \sim q_2$ и, значит, по свойству классов эквивалентности (теор. 1.1) $\mathcal{Q} = \mathcal{Q}'$. Получили

противоречие. Следовательно, наше предположение о существовании различных неотличимых состояний приведенного автомата было неверным. ■

Замечания. 1. Автоматы $K = (A, B, Q, \phi, \varphi, q_0)$ и $\bar{K} = (A, B, \bar{Q}, \bar{\phi}, \bar{\varphi}, q_0)$ работают одинаково, т.е. если на вход этих автоматов подавать одну и ту же входную последовательность, то выходные последовательности автоматов будут одинаковыми.

2. Если строить автомат, реализующий данную ограниченно-детерминированную функцию, через информационное дерево, то получится автомат, все состояния которого отличимы.

Логично задаться вопросом: сколько и каких последовательностей достаточно подавать на вход автомата, чтобы установить отличимость двух его состояний? Ответ на этот вопрос дает теорема Мура.

Пусть $M \subseteq A^*$. Говорят, что состояния q_1 и q_2 автомата $K = (A, B, Q, \phi, \varphi, q_0)$ *отличимы* множеством M , если $\exists w \in M \setminus \{\lambda\}$ такое, что $\bar{\varphi}(q_1, w) \neq \bar{\varphi}(q_2, w)$. В противном случае говорят, что состояния q_1 и q_2 множеством M неотличимы.

Заметим, что обычная отличимость - это отличимость с помощью множества A^* .

Теорема 4.1 (Мура). Если состояния q_1 и q_2 автомата K отличимы, то они отличимы и множеством A^{n-1} , где $n = |Q|$.

Доказательство теоремы опустим.

Пример 3. Табл. 4.8 и 4.9 заданы автомат и соответствующий ему приведенный автомат.

Таблица 4.8

| $A \backslash Q$ | 0 | 1 |
|------------------|------------|------------|
| q_1 | q_3 1 | q_4 1 |
| q_2 | q_4 1 | q_3 0 |
| q_3 | q_5 0 | q_4 0 |
| q_4 | q_3 1 | q_4 1 |
| q_5 | q_4 1 | q_3 0 |

Таблица 4.9

| $A \backslash Q$ | 0 | 1 |
|------------------|------------|------------|
| q_1 | q_1 0 | q_2 0 |
| q_2 | q_1 1 | q_2 1 |
| q_3 | q_1 1 | q_2 0 |

Задачи повышенной сложности

4.1. Автомат задан каноническими уравнениями:

$$\begin{cases} q(0) = 0, \\ b(t) = \overline{a(t)} \cdot \overline{q(t)}, \\ q(t+1) = \overline{a(t)} \vee \overline{q(t)}. \end{cases}$$

а) Задать автомат таблицей и диаграммой Мура.

б) Найти $\bar{\phi}(q_1, 001)$, $\bar{\phi}(q_1, 1101)$, $\bar{\phi}(q_2, 110)$, $\bar{\phi}(q_1, 1101)$.

4.2. а) Автомат задан табл. 4.10. Построить приведенный автомат, соответствующий данному автомату. Задать приведенный автомат таблицей и диаграммой Мура.

Таблица 4.10

| $Q \backslash A$ | q_1 | q_2 | q_3 | q_4 |
|------------------|-----------|-----------|-----------|-----------|
| 0 | $q_2 / 0$ | $q_1 / 0$ | $q_2 / 1$ | $q_2 / 0$ |
| 1 | $q_1 / 1$ | $q_3 / 0$ | $q_4 / 1$ | $q_1 / 1$ |

б) Автомат задан табл. 4.11. Построить приведенный автомат, соответствующий данному автомату. Задать приведенный автомат таблицей и диаграммой Мура.

Таблица 4.11

| $Q \backslash A$ | q_1 | q_2 | q_3 | q_4 | q_5 |
|------------------|-----------|-----------|-----------|-----------|-----------|
| a | q_2 / a | q_1 / a | q_5 / a | q_1 / a | q_3 / a |
| b | q_5 / b | q_4 / a | q_2 / b | q_3 / b | q_4 / a |

4.3. Построить автомат (генератор единиц), заданный условиями: входной и выходной алфавиты автомата совпадают и состоят из двух символов: 0 и 1; входные и выходные сигналы автомата до тех пор пока на его вход первый раз не поступит сигнал 1, совпадают, а после того, как это произойдет, на каждом такте, независимо от сигнала на входе, автомат выдает сигнал 1. Описать работу такого автомата:

- а) диаграммой Мура;
- б) таблицей;
- в) каноническими уравнениями.

4.4. Построить автомат с входным и выходным алфавитами $A = B = \{a, b, c\}$, который отыскивает во входной последовательности сочетание aab , причем до первой встречи с данным сочетанием автомат на выходе выдает тот символ, который поступил на входе, а после встречи с этим сочетанием независимо от входного символа выдает на выходе символ c . Задать автомат:

- а) диаграммой Мура;
- б) таблицей.

Все ли состояния построенного вами автомата отличимы? Если нет, построить по данному автомату приведенный автомат.

4.5. Построить автомат, входной и выходной алфавиты которого совпадают, состоят из трех элементов, а выходной символ представляет собой задержанный на один такт входной символ. Автомат задать:

- а) диаграммой Мура;
- б) таблицей.

4.6. Построить автомат с входным и выходным алфавитами $A = B = \{0,1\}$, который во входной последовательности заменяет символы, стоящие на четных местах, на противоположные, например: $01100101001001 \rightarrow 00110000011100$. Автомат задать:

- а) диаграммой Мура;
- б) таблицей;
- в) системой канонических уравнений.

4.7. Построить автомат с наименьшим числом состояний, удовлетворяющий следующим условиям: $\bar{\varphi}(q_0, 011) = 111$; $\bar{\varphi}(q_0, 1011) = 1101$; $\bar{\varphi}(q_0, 11) = 10$. Автомат задать:

- а) диаграммой Мура;
- б) таблицей;
- в) системой канонических уравнений.

Литература

1. **Гиндикин С.Г.** Алгебра логики в задачах. - М.: Наука, 1972.
2. **Гаврилов Г.П., Сапоженко А.А.** Сборник задач по дискретной математике. - М.: Наука, 1977.
3. **Кузнецов О.П.** Дискретная математика для инженеров. - 3-е изд., перераб. и доп. - СПб.: Изд-во «Лань», 2004.
4. **Дехтярь М.И.** Лекции по дискретной математике: уч. пос. - М.: Интернет-университет информационных технологий; БИНОМ, Лаборатория знаний, 2007.
5. **Асанов М.О., Баранский В.А., Расин В.В.** Дискретная математика: графы, матроиды, алгоритмы. - Ижевск; НИЦ «Регулярная и хаотическая динамика», 2001.
6. **Яблонский С.В.** Введение в дискретную математику: уч. пос. для вузов / *Под. ред. В.А. Садовниченко*. - 3-е изд., стер. - М.: Высшая школа, 2001.
7. **Новиков Ф.Н.** Дискретная математика для программистов. - СПб.: Питер, 2002.
8. **Еруслимский Я.М.** Дискретная математика: теория, задачи, приложения. - М.: Вузовская книга, 2000.
9. **Оре О.** Графы и их применения. - Н.: ИО НФМИ, 2000.
10. **Харари Ф.** Теория графов. - М.: Едиториал УРСС, 2003.
11. **Кожухов И.Б., Прокофьев А.А., Соколова Т.В.** Курс дискретной математики. - М.: МИЭТ, 2000.
12. **Шаноров С.Д.** Дискретная математика: курс лекций и практических занятий. - СПб.: БХВ-Петербург, 2006.