Министерство образования и науки Республики Казахстан

«Алматинский университет энергетики и связи имени Гумарбека Даукеева»
Некоммерческое акционерное общество

Л.Н. Астраханцева, М.Ж. Байсалова

ДИСКРЕТНАЯ МАТЕМАТИКА

Учебное пособие

Алматы
АУЭС
2021

УДК 519.6(075.8)
ББК 22.1 я 73
А 91

Рецензенты:
доктор физико-математических наук,
профессор кафедры «Математика» КазНУ имени аль-Фараби
М.К.Дауылбаев
кандидат физико-математических наук,
доцент кафедры «Математика» КазНУ имени аль-Фараби
У.К.Койлышов
кандидат физико-математических наук,
доцент кафедры «Математика и математическое моделирование»
Алматинский университет энергетики и связи имени Гумарбека Даукеева
Р.Е.Ким

Астраханцева Л.Н., Байсалова М.Ж.
А 91 Дискретная математика: Учебное пособие. – Алматы: Алматинский университет энергетики и связи имени Гумарбека Даукеева, 2021. – 100 с.: табл – 18, иллюстраций – 69, библиогр. – 19.

ISBN 978-601-7939-79-3

Учебное пособие предназначено для студентов всех форм обучения и специальностей, в программу которых входит предмет «Дискретная математика».

УДК 519.6(075.8)
ББК 22.1я73
А 91

1 Элементы теории множеств.

1.1 Множества

Понятия “множество”, “отношение”, “функция” и некоторые другие являются базовыми понятиями дискретной математики, составляют её основной словарь. Множество и элемент множества первичные понятия, т.е. не определяются с помощью других, более простых понятий, такие как, например, точка и прямая. Под множеством понимается совокупность некоторых объектов (предметов), которые называются элементами множества. Элементы множеств различны и различимы. Приняты следующие обозначения: $A, B, X,\ldots$ - множества; $a, b, x, x_1, x_2,\ldots$- элементы множеств; $a \in$ A - элемент a принадлежит A, $b \notin$ A - элемент b не принадлежит A; N (или ϖ) – множество натуральных чисел; Z – множество целых чисел; Q – множество рациональных чисел; I - множество иррациональных чисел; R – множество действительных чисел; C – множество комплексных чисел; $\emptyset$ – пустое множество (не содержит ни одного элемента).

Конечные и бесконечные множества состоят соответственно из конечного и бесконечного числа элементов.

Способы задания множеств:

а) перечислением элементов, например,

$$X = \{x_1, x_2, \ldots, x_n\},\ \ A = \{2, 4, 6, 8, \ldots\}$$

б) с помощью характеристического свойства: $A = \{x|P(x)\}$, где $P(x)$ – свойство P, которым обладает элемент x, например $A = \{x|x = \frac{\pi}{2} + \pi k;\ k \in \mathrm{Z}\}$;

в) порождающей процедурой, которая описывает способ получения элементов из уже имеющихся элементов, например, числовое множество $M = \{1, 2, 4, 8, 16, \ldots\}$ можно задать так: 1) $1 \in M$; 2) $m \in\ M \to 2m \in M$.

Определения:

а) множество B называется подмножеством множества A (обозначается $B \subseteq A$), если каждый элемент множества B является элементом множества A: $B \subseteq A \Leftrightarrow \forall x(x \in B \Rightarrow x \in A)$, $\subseteq$ - знак включения;

б) множества A и B называют равными, если они состоят из одних и тех же элементов: $A = B \Leftrightarrow\ B \subseteq A$ и $A \subseteq B$;

в) если $B \subseteq A$ и $A \neq B$, то B является собственным подмножеством множества A: $B \subset A$ - строгое включение.

Заметим, что для обозначения отношения включения применяют как знак строгого, так и не строгого включения, как для собственных, так и для несобственных подмножеств. И только если требуется различить эти подмножества, различают и эти знаки. Не следует путать знаки $\in$ и $\subseteq$:

$$\left.\begin{array}{c} 1 \in \{1,2,3\} \\ \{a\} \in \{\{a\}\} \\ \{1\} \subset \{1,2,3\} \end{array}\right\} \text{- верно,} \qquad \left.\begin{array}{c} 1 \subseteq \{1,2,3\} \\ 1 \notin \{1,2,3\} \\ \{a\} \in \{a,b\} \end{array}\right\} \text{- не верно.}$$

Множества могут быть элементами других множеств. Множество, элементами которого являются множества, иногда называют семейством и обычно обозначают прописными (готическимими) буквами латинского алфавита. Совокупность всех подмножеств множества A называется его булеаном или множеством - степенью. Обозначается $\mathcal{P}(A)$ или 2^A.

Итак, $\mathcal{P}(A) = \{B \mid B \subseteq A\}$. Булеан множества из n элементов, содержит 2^n элементов.

Пример 1.1.1 $A = \{1,2,3\}$, $\mathcal{P}(A) = \{\emptyset,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\}, A\}$. $\mathcal{P}(A)$ содержит 8 элементов, $8 = 2^3$.

Если в конкретных рассуждениях все элементы рассматриваемых множеств принадлежат какому-то одному большому множеству, то такое множество называется универсальным или универсумом и обозначается U. Для наглядного изображения множеств используют диаграммы Эйлера-Венна, на которых множества обозначаются точками кругов внутри прямоугольника, точки которого – множество U- универсум.

Операции над множествами.

$\forall A, B \in \mathcal{P}(U)$ следующие операции определяются так:

а) объединение (сумма) (обозначение $\cup, +$): $A \cup B = \{x \mid x \in A$ или $x \in B\}$;

б) пересечение (произведение) ($\cap$, $\bullet$): $A \cap B = \{x \mid x \in A$ и $x \in B\}$;

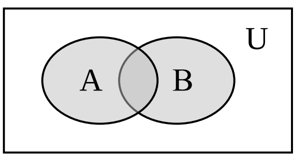
в) разность ($A \setminus B$; $A - B$): $A \setminus B = \{x \mid x \in A$ и $x \notin B\}$;

г) симметрическая разность или кольцевая сумма $\oplus$, Δ, $+$):

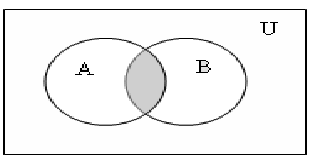
$A \oplus B = (A \setminus B) \cup (B \setminus A) = \{x \mid (x \in A$ и $x \notin B)$ или $(x \in B$ и $x \notin A)\}$;

д) дополнение множества A есть ($\overline{A}$): $\overline{A} = \{x \mid x \in U$ и $x \notin A\} = U \setminus A$.

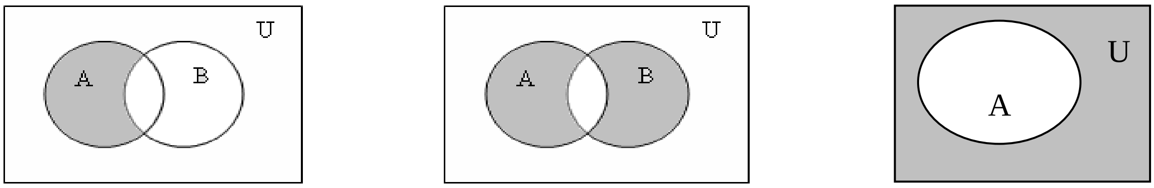
Иллюстрация операций над множествами диаграммами Эйлера-Венна на рисунке 1.1.1.



A ∪ B



A ∩ B



$A \setminus B$ $\qquad$ $A \oplus B$ $\qquad$ $\overline{A}$

Рисунок 1.1.1

Операции объединения и пересечения допускают обобщения:

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

Свойства операций над множествами

Для преобразования теоретико-множественных выражений, упрощения записей, доказательств теорем и свойств необходимо знать свойства операций над множествами. Рассмотрим важнейшие из этих свойств. Пусть задан универсум U и множества A, B, C $\subset$ U.

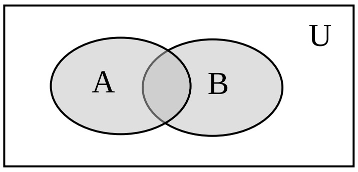
Таблица 1.1.1 Свойства операций над множествами

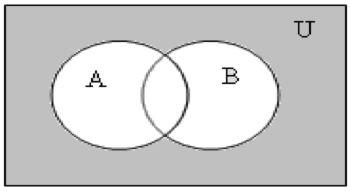
Идемпотентность	$A \cup A = A$	$A \cap A = A$
Коммутативность	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Дистрибутивность	$A \cup (B \cap C) =$ $(A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) =$ $(A \cap B) \cup (A \cap C)$
Ассоциативность	$A \cup (B \cup C) = (A \cup B) \cup C$	$A \cap (B \cap C) = (A \cap B) \cap C$
Свойства поглощения	$A \cup (B \cap A) = A$	$A \cap (B \cup A) = A$
Свойства нуля и единицы	$A \cup \emptyset = A$ $A \cup U = U$	$A \cap \emptyset = \emptyset$ $A \cap U = A$
Законы де Моргана	$\overline{A \cup B} = \overline{A} \cap \overline{B}$	$\overline{A \cap B} = \overline{A} \cup \overline{B}$
Закон двойного отрицания или инволютивности	$\overline{\overline{A}} = A$	
Свойства дополнения	$A \cup \overline{A} = U$	$A \cap \overline{A} = \emptyset$

Доказать эти свойства можно либо с помощью диаграмм Эйлера-Венна, либо формальными рассуждениями, опирающимися на определение операций, например, докажем $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

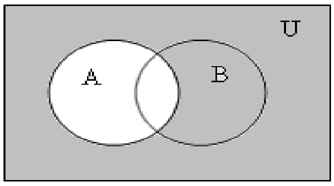
1. Доказательство с помощью диаграмм:

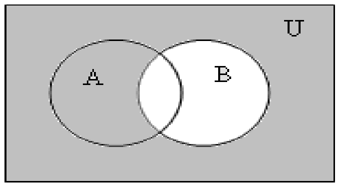
а) A∪B $\longrightarrow$ $\overline{A \cup B}$





б) $\overline{A}$ $\overline{B}$





$\overline{A} \cap \overline{B}$

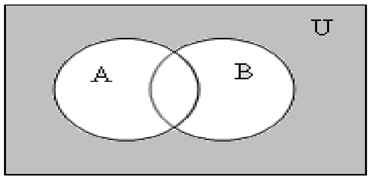


Рисунок 1.1.2

На последних рисунках в пунктах а) и б) отмечена одна и та же область, что доказывает тождество.

2. Докажем $\overline{A \cup B} = \overline{A} \cap \overline{B}$ формальными рассуждениями.

В формальных рассуждениях исходят из того, что $A = B \Leftrightarrow A \subseteq B$ и $B \subseteq A$, а последнее имеет место по определению отношения включения: $A \subseteq B \Leftrightarrow (x \in A \rightarrow x \in B)$ и $B \subseteq A \Leftrightarrow (x \in B \rightarrow x \in A)$, поэтому:

а) $x \in \overline{A \cup B} \Rightarrow x \notin A \cup B \Rightarrow x \notin A$ и $x \notin B \Rightarrow x \in \overline{A}$ и $x \in \overline{B} \Rightarrow x \in \overline{A} \cap \overline{B}$;

б) $x \in \overline{A} \cap \overline{B} \Rightarrow x \in \overline{A}$ и $x \in \overline{B} \Rightarrow x \notin A$ и $x \notin B \Rightarrow x \notin A \cup B \Rightarrow x \in \overline{A \cup B}$.

Теорема. Для любых множеств A и B следующие условия эквивалентны:

а) $A \subseteq B$; б) $A \cap B = A$; в) $A \cup B = B$;

г) $A \setminus B = \emptyset$; д) $\overline{A} \cup B = U$.

В примере 1.1.2 свойства операций использованы для упрощения выражения.

П р и м е р 1.1.2 $\overline{A \cap \overline{B}} \cup B = \overline{A} \cup \overline{\overline{B}} \cup B = \overline{A} \cup B \cup B = \overline{A} \cup B$.

Разбиения и покрытия множеств

Пусть дано множество A. $\mathcal{A} = \{A_1, A_2, \dots A_n\}$– множество подмножеств A (семейство подмножеств).

Определение. $\mathcal{A}$ называется покрытием множества A, если

1. $\forall A_i \in \mathcal{A} \ (A_i \subset A, A_i \neq \emptyset)$; 2. $A = \bigcup_{i=1}^{n} A_i$.

Определение. $\mathcal{A}$ называется разбиением множества А, если

1. $\forall A_i \in \mathcal{A} \ (A_i \subset A, A_i \neq \emptyset)$; 2. $A = \bigcup_{i=1}^{n} A_i$;

3. $\forall A_i, A_j \in \mathcal{A} \ [A_i \neq A_j \Rightarrow A_i \cap A_j = \emptyset]$.

Элементы разбиения, т.е. подмножества множества A также называются блоками разбиения.

П р и м е р 1.1.3 $A = \{1, 2, 3\}$. $\mathcal{A}_1 = \{\{1,2\}, \{2,3\}, \{1,3\}\}$ – покрытие; $\mathcal{A}_2 = \{\{1\}, \{2\}, \{1\}\}$– разбиение; $\mathcal{A}_3 = \{\{1\}, \{2,3\}\}$– разбиение; $\mathcal{A}_4 = \{\{1\}, \{3\}\}$– множество подмножеств множества A (ни булеан, ни покрытие, ни разбиение).

П р и м е р 1.1.4. N – множество натуральных чисел. N_0, N_1– множества чётных и нечётных чисел. $\mathcal{N} = \{N_0, N_1\}$ – разбиение N.

Прямое произведение множеств

Упорядоченную последовательность из элементов $x_1, x_2, \ldots, x_n$ будем обозначать $(x_1, x_2, \ldots, x_n)$ или $< x_1, x_2, \ldots, x_n >$ и называть кортеж длины n, упорядоченный набор или последовательность (конечная) из n элементов, вектор длины n, n-ка (энка). x_i – i-ая координата или компонента. Если $n = 2$, то (x_1, x_2) – пара, упорядоченная двойка; $n = 3$ – (x_1, x_2, x_3) – тройка, упорядоченная тройка; $n = 0$ – $<>= \emptyset$ – кортеж, не содержащий элементов.

Если $\vec{x} = (x_1, \ldots, x_n)$, $\vec{y} = (y_1, \ldots, y_n)$, то $\vec{x} = \vec{y} \Leftrightarrow (x_1 = y_1, x_2 = y_2, \ldots, x_n = y_n)$. Ясно, что $(1,2) \neq (2,1)$, $\{1,2\} = \{2,1\}$..

Определение. Прямым (декартовым) произведением множеств A и B (обозначается $A \times B$) называется множество таких пар (a, b), что $a \in A$ и $b \in B$: $A \times B = \{(a, b) \mid a \in A \text{ и } b \in B\}$.

Обобщение прямого произведения:

$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n) \mid a_1 \in A_1, \ a_2 \in A_2, \ldots, a_n \in A_n\}$.

Если $A = B$, то $A \times A = A^2$; $\underbrace{A \times A \ldots \times A}_{n} = A^n$; $A^1 = A$; $A^0 = \{\emptyset\}$.

П р и м е р 1.1.5 $A = \{1, 2\}$, $B = \{1, 2, 3\}$.

$A \times \mathrm{B} = \{(1; 1), (1; 2), (1; 3), (2; 1), (2; 2), (2; 3)\}$;

$B \times A = \{(1; 1), (1; 2), (2; 1), (2; 2), (3; 1), (3; 2)\}$;

$A \times B \neq B \times A$; $A^2 = \{(1,1), (1,2), (2,1), (2,2)\}$;

П р и м е р 1.1.6 $R \times R = R^2 = \{(a, b) | a, b \in R, (a, b) \text{ – точки плоскости}\}$.

1.2 Отношения

При решении различных задач практики часто требуется учитывать связи или отношения между элементами одного и того же или разных множеств. Например, если имеем множество стран мира, то можно рассматривать между странами такие отношения: «в стране x населения больше, чем в стране y» или «страны x и y имеют общую границу»; если имеем множества мужчин, женщин и детей, то можно рассматривать отношение «x и y родители z» и т.д.

Определение. n-местным отношением P (n-местным предикатом) на множествах $A_1, A_2, \dots, A_n$ называется любое подмножество прямого произведения этих множеств: $P \subseteq A_1 \times A_2 \times \dots \times A_n$ и $P = \{(x_1, x_2, \dots, x_n) \mid x_1 \in A_1, \dots, x_n \in A_n\}$. То, что элементы $x_1, x_2, \dots, x_n$ связаны соотношением P записывается в виде: $(x_1, \dots, x_n) \in P$ или $P(x_1, \dots, x_n)$. Если $P \subseteq A^n$, то P – n-местное отношение на множестве A.

$n = 1$, то $P \subseteq A_1$ - одноместное (унарное) отношение или свойство;

$n = 3$, то $P \subseteq A_1 \times A_2 \times A_3$ – трёхместное (тернарное) отношение.

Наиболее часто встречаются и хорошо изучены бинарные отношения ($n = 2$) или соответствия $P \subseteq A_1 \times A_2$ или $P = \{(x, y) \mid x \in A_1,\ y \in A_2\}$. Записывают $P(x, y)$ или xPy. Например, вместо “$< (x; y)$” или “$(x; y) \in <$” записывают $x < y$. Далее будем рассматривать бинарные отношения, называя их просто отношениями.

Определение. Областью определения отношения P (обозначается D_P) называется $D_P = \{x \mid (x, y) \in P$ для некоторого $y\}$; областью значений (обозначается E_P) называется $E_P = \{y \mid (x, y) \in P$ для некоторого $x\}$ (т.е. D_P – это множество первых координат пар P, E_P – вторых).

Отношение можно задать перечислением элементов, характеристическим свойством, графически, с помощью матриц.

Бинарные отношения на конечных множествах обычно задаются либо списком пар, либо матрицей, либо графически.

Определение. Пусть $A = \{a_1, a_2, \dots, a_n\}$, $B = \{b_1, b_2, \dots, b_n\}$ и $P \subseteq A \times B$. Матрица $[P] = (p_{ij})$, размера $m \times n$, называется матрицей отношения P, если

$$p_{ij} = \begin{cases} 1, \text{если } (a_i, b_j) \in P \\ 0, \text{если } (a_i, b_j) \notin P \end{cases}, \quad i = 1,2, \dots, m;\ j = 1,2, \dots, n.$$

Графически отношение P можно задать по-разному:

а) на координатной плоскости по осям координат отмечают элементы множеств A и B, тогда точки с координатами (x, y) представляют элементы отношения $P \subseteq A \times B$, $x \in A$, $y \in B$;

б) на плоскости изображают две ограниченные области, в которых точками отмечают элементы множеств A и B, затем, если $(x, y) \in P \subseteq A \times B$, то точки x и y соединяют стрелкой;

в) отношение $P \subseteq A^2$ можно задать ориентированным графом, т.е. элементы $x, y \in A$, изображённые точками, соединяют стрелками.

П р и м е р 1.2.1 $A = \{\mathrm{a, b, c}\}$, $B = \{1,2,3,4\}$, $P_1 = \{(\mathrm{a}, 1), (\mathrm{b}, 1), (\mathrm{b}, 3), (\mathrm{c}, 2)\} \subseteq A \times B$, $P_2 = \{(1,1), (1,2), (2,3), (3,1), (4,3)\} \subseteq B^2$.

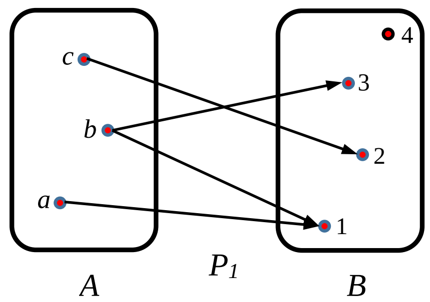
Матричное задание отношений:

$$[P_1] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad [P_2] = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix};$$

Графическое задание отношений:

а)

б)



в)

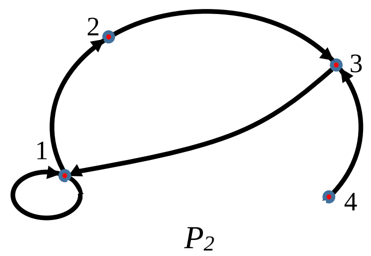


Рисунок 1.2.1

Определения. Пусть $P \subseteq A \times B$, $P = \{(a, b) \mid a \in A, b \in B\}$.

а) P^{-1} – обратное $P \Leftrightarrow P^{-1} = \{(b, a) \mid (a, b) \in P\}$, $P^{-1} \subseteq B \times A$;

б) $\overline{P}$ - дополнение $P \Leftrightarrow \overline{P} = \{(a, b) \mid (a, b) \notin P\}$, $\overline{P} \subseteq A \times B$;

в) I – тождественное отношение на множестве A (иногда обозначается id$_A$). $I = \{(a, a) \mid a \in A\}, I \subseteq A^2$ (называют также диагональю в A^2, т.к. его матрицей является единичная матрица);

г) U- универсальное отношение - $U = \{(a, b) | a \in A$ и $b \in A\}$, т.е. $U = A^2$.

Определение. Композицией (произведением) бинарных отношений $P_1 \subseteq A \times B$ и $P_2 \subseteq B \times C$ (обозначается $P_1 \circ P_2$) называется отношение
$P = P_1 \circ P_2 = \{(a, c) | a \in A, c \in C$ и $\exists b \in B$, что $(a, b) \in P_1$ и $(b, c) \in P_2\}$.

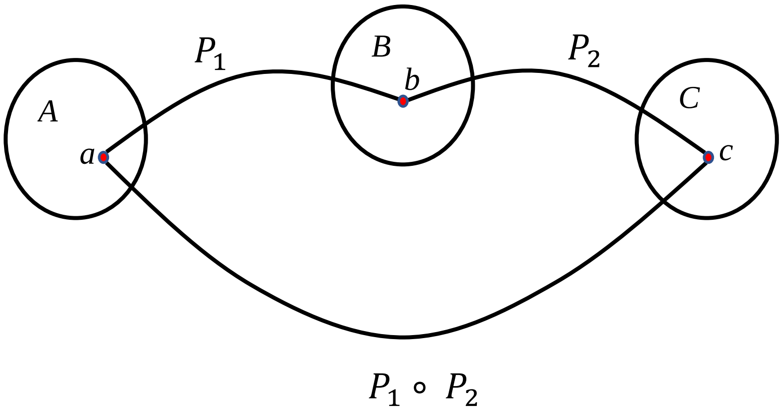


Рисунок 1.2.2

П р и м е р 1.2.2 $A = \{1, 2, 3, 4\}$; $B = \{6, 7, 8\}$; $C = \{ 10, 11, 12\}$.

Пусть $P_1 = \{(1,7), (4,6), (2,8)\} \subseteq A \times B$,
$P_2 = \{(6,10), (6,11), (7,10), (8,12)\} \subseteq B \times C$, тогда $P_1^{-1} = \{(7,1), (6,4), (8,2)\}$; т.к. $A \times B = \{(1,6), (1,7), (1,8), (2,6), (2,7), (2,8), (3,6), (3,7), (3,8), (4,6), (4,7), (4,8)\}$, то $\overline{P}_1 = (A \times \mathrm{B}) \backslash \mathrm{P}_1 = \{(1,6), (1,8), (2,6), (2,7), (3,6), (3,7), (3,8), (4,6), (4,7), (4,8)\}$;

$$P_1 \circ P_2 = \{(1,10), (4,10), (4,11), (2,12)\}.$$

Эта композиция получается так: для первого элемента (1,7) отношения P_1 находим элементы отношения P_2 , у которых число 7 стоит на первом месте – это (7,10), делаем вывод, что элемент (1,10) входит в композицию; для второго элемента (4,6) находим элементы P_2 , у которых 6 стоит на первом месте – это (6,10) и (6,11), тогда (4,10), (4,11) входят в композицию и т.д. $P_1 \circ P_2 = \emptyset$.

Теорема. Для любых бинарных отношений P, Q, R выполняются следующие свойства:

а) $(P^{-1})^{-1} = P$;

б) $(P \circ Q)^{-1} = Q^{-1} \circ P^{-1}$;

в) $(P \circ Q) \circ R = P \circ (Q \circ R)$ (ассоциативность);

г) $P \circ Q \neq Q \circ P$ (коммутативность в общем случае не выполняется).

Основные свойства матриц бинарных отношений

1 Если $P, Q \subseteq A \times B$, $[P] = (p_{ij}), [Q] = (q_{ij})$, то $[P \cup Q] = (p_{ij} + q_{ij}) = [P] + [Q]$, где элементы матриц складываются по правилам: $0 + 0 = 0$, $1 + 0 = 0 + 1 = 1 + 1 = 1$;$[P \cap Q] = (p_{ij} * q_{ij}) = [P] * [Q]$, где элементы матриц перемножаются по обычным правилам: $0 * 0 = 1 * 0 = 0 * 1 = 0$, $1 * 1 = 1$.

2 Если $P \subseteq A \times B, Q \subseteq B \times C$, то $[P \circ Q] = [P] \cdot [Q]$ - обычное умножение матриц, но элементы матриц $[P]$ и $[Q]$ складываются и умножаются по правилам из свойства 1.

3 Если P^{-1} отношение, обратное к P, то $[P^{-1}] = [P]^T$.

4 Если $P \subseteq Q$ и $[P] = (p_{ij}), [Q] = (q_{ij})$, тогда $p_{ij} \leq q_{ij}$.

5 Если I –тождественное отношение, то

$$[I] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{- единичная матрица.}$$

6 Если $\overline{P}$ - дополнение P, то $[\overline{P}]$ равна матрице отношения P, в которой нули заменены единицами и единицы нулями.

Пример 1.2.3

$[P] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $[Q] = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$, тогда $[P \cup Q] = [P] + [Q] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$;

$[P \cap Q] = [P] * [Q] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$;

$[P \circ Q] = [P] \cdot [Q] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$; $[Q^{-1}] = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$, $[\overline{Q}] = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$.

Свойства отношений

Таблица 1.2.1 Свойства отношения $P \subseteq A^2$

P	определение		$[P]$
рефлексивное	$\forall a \in A;\ (a,a) \in P$	$I \subseteq P$	$\begin{pmatrix} 1 & * & * & * \\ * & 1 & * & * \\ \dots & \dots & \dots & \dots \\ * & * & * & 1 \end{pmatrix}$
антирефлексивное	$\forall a \in A;\ (a,a) \notin P$	$P \cap I = \emptyset$	$\begin{pmatrix} 0 & * & * & * \\ * & 0 & * & * \\ \dots & \dots & \dots & \dots \\ * & * & * & 0 \end{pmatrix}$
симметричное	$\forall a,b \in A; (a,b) \in P \Rightarrow (b,a) \in P$	$P = P^{-1}$	$[P] = [P^{T}]$
антисимметричное	$\forall\, a,b \in A\ [(a,b) \in P$ и $(b,a) \in P \Rightarrow a = b]$	$P \cap P^{-1} \subseteq I$	$[P \cap P^{-1}] = [P]*[P^{-1}] = \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & * \end{pmatrix}$
транзитивное	$\forall\, a,b,c \in A\ [(a,b) \in P$ и $(b,c) \in P \Rightarrow (a,c) \in P]$	$P \circ P \subseteq P$	если $[P \circ P] = (a_{ij})$, $[P]=(p_{ij})$, то $a_{ij} \leq p_{ij}$

Проверить, какими свойствами обладает отношение, на практике лучше всего по его матрице.

Пример 1.2.4 Пусть дано отношение:

$P = \{(1,2), (2,3), (3,2)\} \subseteq A^2,\ A = \{1,2,3\}$.

Проверим, какими свойствами оно обладает.

Составим матрицу этого отношения $[P] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

- P не рефлексивное, т.к. на главной диагонали матрицы нет единиц;
- P антирефлексивное, т.к. на главной диагонали матрицы все нули;
- P не симметричное, т.к. $(1;2) \in P$, но $(2;1) \notin P$ или

  $[P] \neq [P^T] = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$;

- P не антисимметричное, т.к. (2;3) ∈ P, (3;2) ∈ P, но 2 ≠ 3, или в матрице

$$[P] * [P]^T = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

не все элементы вне главной диагонали нули;

- Р не транзитивное, т.к., (1,2) ∈ Р, (2,3) ∈ Р, но (1,3) ∉ Р или для

$$[P] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = (p_{ij}),$$

$[P \circ P] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (a_{ij})$ условие $a_{ij} \leq p_{ij}$ не выполняется для всех i, j, например, $a_{13} = 1,\ p_{13} = 0$, т.е. $a_{13} > p_{13}$.

Многие отношения обладают определённым набором указанных выше свойств, и этот набор встречается так часто, что обладающие им отношения заслуживают специального названия и отдельного изучения.

Отношение эквивалентности

Определение. Отношение P называется отношением эквивалентности, если оно рефлексивно, симметрично и транзитивно (обозначается $\sim,\ E,\ \equiv$).

Примеры отношений эквивалентности:

а) отношение равенства на любом множестве:

1) $x = x$;

2) $x = y \to y = x$;

3) $x = y;\ y = z\ \to\ x = z$;

б) отношение параллельности на множестве прямых на плоскости.

Определение. Пусть E – отношение эквивалентности на множестве A и $x \in A$. Подмножество элементов множества A, эквивалентных x, называется классом эквивалентности элемента x. Обозначается: $[x]_E, E(x)$. Таким образом, $E(x) = \{y \mid yEx\}$.

Определение. Множество классов эквивалентности называется фактор - множеством множества A относительно эквивалентности E, обозначается $A/E : A/E = \{E(x) \mid x \in A\}$. Фактор–множество является подмножеством булеана.

Пример 1.2.5 A – множество студентов в университете. E – отношение принадлежности к одной группе. $[x]_E$ -студенты одной группы. A/E – множество студенческих групп университета.

Из определений ясно, что

1) любой элемент класса эквивалентности порождает класс эквивалентности, т.е. $b \in [a] \to [a] = b$;

2) каждый класс эквивалентности содержит хотя бы один элемент, т.е. $\forall a \in A,\ [a] \neq \emptyset$;

3) никакой элемент множества А не может принадлежать двум различным классам: $aEb \to [a] = [b]$.

*Теорема***.** Фактор – множество A/E является разбиением множества A. Обратно, если $\mathcal{A} = \{A_i\}$ какое-то разбиение множества A, то ему соответствует некоторое отношение эквивалентности E: $xEy \Leftrightarrow x, y \in A_i$ для некоторого i или $E = \{(x; y) \mid x, y \in A_i\}$ для некоторого i }.

Таким образом, существует взаимно однозначное соответствие между множеством всех разбиений A и множеством всех отношений эквивалентности на множестве A.

Пример 1.2.6 $A = \{1,2,3,4\}$. $\mathcal{A} = \big\{\{1\}, \{2,3,4\}\big\} = \{A_1, A_2\}$- разбиение A.

$E = \{(x; y) \mid x, y \in A_i, i = 1,2 \} =$
$= \{(1,1); (2,2); (2,3); (2,4); (3,2); (3,3); (3,4); (4,2); (4,3); (4,4)\}$ - отношение эквивалентности, соответствующее данному разбиению.

Пример 1.2.7 $A = \{1,2,3,4,5,6\}$, $P \subseteq A^2$, $P = \{(1,1); (2,2); (3,3);$ $(4,4); (5,5); (6,6); (1,2); (1,4); (2,1); (2,4); (3,5); (5,3); (4,1); (4,2)\}$.

Покажем, что данное отношение является отношением эквивалентности.

По его матрице

$$[P] = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

определяем, что P рефлексивно, симметрично, транзитивно (см. выше), и следовательно, P есть отношение эквивалентности на множестве A. Построим классы эквивалентности и фактор – множество:

$[1]_P = \{x \mid (x;1) \in P\} = \{1,2,4\}$; $[2]_P = \{x \mid (x;2) \in P\} = \{1,2,4\}$;

$[3]_P = \{x \mid (x;3) \in P\} = \{3,5\}$; $[4]_P = \{x \mid (x;4) \in P\} = \{1,2,4\}$;

$[5]_P = \{x \mid (x;5) \in P\} = \{3,5\}$; $[6]_P = \{x \mid (x;6) \in P\} = \{6\}$.

Таким образом, имеется только три различных класса эквивалентности:

$$[1] = [2] = [4] = \{1,2,4\},\ \ [3] = [5] = \{3,5\},\ \ [6] = \{6\}.$$

Фактор – множество $A/P = \{[1],[3],[6]\} = \big\{\{1,2,4\},\{3,5\},\{6\}\big\}$ является разбиением A, которое соответствует данному отношению эквивалентности.

Отношение порядка

Определение. Отношение P на множестве A называется отношением порядка если оно антисимметрично и транзитивно. Часто обозначается символом $<$.

Если к тому же оно:

1) рефлексивно, то называется частичным или нестрогим порядком ($\leq$);

2) антирефлексивно, то называется отношением строгого порядка ($<$).

Определение. Пусть на множестве A задано отношение порядка $<$, если для любых двух элементов a и b этого множества имеет место $a < b$ или $b < a$, то элементы называются сравнимыми, в противном случае несравнимыми.

Определение. Частичный порядок на множестве A называется линейным или цепью, если любые два элемента этого множества сравнимы.

Множество A, на котором определен частичный (линейный) порядок, называется частично упорядоченным множеством (ч.у.м) (линейно упорядоченным множеством (л.у.м)). Обозначается $(A,<)$.

П р и м е р 1.2.8 $A = \{a,b,c\}$. $P = \{(a,a),(a,b),(b,b),(c,c)\}$ – отношение частичного порядка (т.е. рефлексивно, антисимметрично, транзитивно), что легко проверить по его матрице

$$[P] = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

P не является линейным порядком, т.к. a и c, b и c не сравнимы.

Примерами линейно упорядоченных множеств являются множества N, Z, Q, R, где определён естественный порядок.

Определение. Элемент a упорядоченного множества A называется наибольшим (наименьшим), если $x \leq a$, $(a \leq x)$ $\forall x \in A$. Л.у.м. называется вполне упорядоченным множеством (в.у.м), если любое его непустое подмножество имеет наименьший элемент.

П р и м е р 1.2.9 $(N; \leq)$ – в.у.м. ([0;1]; $\leq$) – не является в.у.м., т.к., например, (0;1] ⊆ [0;1], но (0;1] не содержит наименьшего элемента.

Определение. Рассмотрим ч.у.м. $(X; \leq)$. Элемент y покрывает элемент x, если $x \leq y$ и не существует такого элемента z, что $x < z < y$.

В случае конечного множества X, ч.у.м. $(X; \leq)$ можно представить в виде схемы, в которой каждый элемент изображается точкой на плоскости. Если y покрывает x, то точки x и y соединяют отрезком, причём точку, соответствующую x, располагают ниже точки y. Если отношение порядка нестрогое, то в каждой точке изображается петля. Кроме того, т.к. отношение порядка транзитивное, например, $(a, b) \in P, (b, c) \in P$ и $(a, c) \in P$, то точки a и c не надо соединять отрезком, т.к. они соединены через точку b. Такие схемы называют диаграммами Хассе.

П р и м е р 1.2.10 Рассмотрим ч.у.м. $(\mathcal{P}(A), \subseteq) = \{(A_i, A_j) | A_i \subseteq A_j\}$, где $A = \{a, b, c\}$, $\mathcal{P}(A)$ булеан A. Диаграмма Хассе для $(\mathcal{P}(A), \subseteq)$ на рисунке 1.2.3 а); для л.у.м. $(\{1, 2, 3, 4\}, \leq)$ с обычным отношением порядка на множестве натуральных чисел, не превосходящих четырёх, диаграмма Хассе изображена на рисунке 1.2.3 б); диаграмма Хассе для отношения из примера 1.2.8 на рисунке 1.2.3 в).

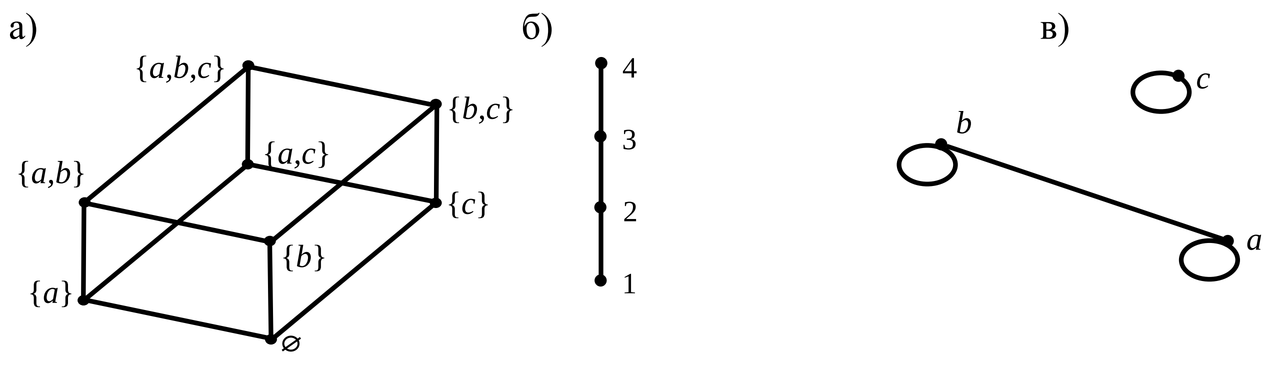


Рисунок 1.2.3

Лексикографический порядок

Лексикографический порядок лежит в основе упорядочения слов в различных словарях. Рассмотрим непустое множество символов $X = \{x, y, \dots\}$, называемое алфавитом. Словами будем называть конечные наборы написанных друг за другом элементов X. Элемент x_i слова $x_1, x_2, \dots, x_n$ назовём его i -ой координатой, а число n - его длиной.

Определение. Пусть $W(X)$ – множество слов алфавита X, пусть $\leq$ - отношение порядка на множестве X, т.е. $(X, \leq)$ – упорядоченное множество. Отношение лексикографического порядка (обозначается $<$ или $\mathcal{L}$) на $W(X)$ задаётся по правилу: $x_1, x_2, \dots, x_m \; \mathcal{L} \; y_1, y_2, \dots, y_n$, если выполнено одно из условий:

а) $x_1 < y_1$;

б) $x_i = y_i \; \forall i: 1 \leq i \leq m, m < n$;

в) $x_i = y_i \; \forall i: 1 \leq i \leq k, x_{k+1} < y_{k+1}$.

Функциональные отношения (функции)

Определение. Бинарное отношение $f \subseteq A \times B$ называется функциональным или функцией из множества A в множество B, если:

а) $(x, y_1) \in f, (x, y_2) \in f \to y_1 = y_2$ или

$\forall x \in A, \; \exists!\, y \in B: (x, y) \in f$;

б) $D_f = A, \; E_f \subseteq B$, где D_f - область определения функции, E_f - область значений.

В этом случае функцию иногда называют тотальной или всюду определённой; если $D_f \subseteq A$, то f называют частичной функцией или частично определённой. В математике, как правило, рассматривают тотальные функции и называют их просто функциями.

Пример 1.2.11 $A = \{1,2,3\}, f = \{(1,2), (2,3), (3,2)\} \subseteq A^2$ - функция, т.к. $D_f = A, \; E_f \subseteq A$.

$P = \{(1,1), (1,2), (2,3)\} \subseteq \mathrm{A}^2$– не функция, т.к. содержит пары (1,1) и (1,2) с одинаковыми первыми и разными вторыми элементами.

Отношение $P = \{(x; x^2 + x + 1) \mid x \in R\}$ - функция, т.к. из того, что $x = y$ следует $x^2 + x + 1 = y^2 + y + 1$.

$P = \{(x^2; x) \mid x \in R\}$ – не функция, т.к. содержит пары с одинаковыми первыми и разными вторыми элементами, например, $(1, -1) \in P$, $(1,1) \in P$.

$P = \left\{\left(x; \sqrt{x}\right) \mid x \in R\right\}$ - частичная функция, т.к. $D_P = [0; +\infty)$, $D_P \subseteq R$.

Если задана функция $f = \{(x, y) \mid x \in A, y \in B\}$, то x - аргумент, y - значение функции. Различные обозначения функции:

$$y = f(x);\ \ f: A \to B;\ f: x \to y;\ A \xrightarrow{f} B;\ \ x \xrightarrow{f} y$$

Говорят также, что f ставит в соответствие элементу x элемент y.

Пусть $f = \{(x, y) \mid x \in A, y \in B\}$ – функция. Она называется:

а) инъективной (инъекцией, разнозначной), если:

$(x_1, y) \in f$ и $(x_2, y) \in f \to x_1 = x_2$ (или $x_1 \neq x_2 \to f(x_1) \neq f(x_1)$), при этом f^{-1}- частичная функция;

б) сюръективной (сюръекцией, отображением A на B), если $\forall y \in B$, $\exists x \in A$, что $(x, y) \in f$, т.е. $E_f \in B$;

в) биективной (биекцией, взаимно-однозначным соответствием), если она является и инъективной и сюръективной (для тотальной функции). Обозначается: $A \leftrightarrow B$.

Заметим, что если функция частичная, то, в случае её инъективности и сюръективности, она не всегда биективна, например, $P = \{(x; y) | y = ln\, x$, $x, y \in R\}$ ($P \subseteq R \times R$) - частичная функция, т.к. $D_P = (0, +\infty)$, $D_P \subseteq R$. Она инъективна, т.к. для любых $x_1 \neq x_2$ из области определения выполняется $ln\, x_1 \neq ln\, x_2$; она сюръективна, т.к. $E_P = R$, но биекции нет, т.к. существуют $x \in R$ (например, $x = -1$), которым не соответствует ни один $y \in R$.

Если биекция f: $A \leftrightarrow A$, то она называется подстановкой множества A. Простейший пример подстановки есть тождественное отношение I.

Пример 1.2.12 Рассмотрим три функции f_i: $R \to R (i = 1,2,3)$.

1) $f_1(x) = e^x$ - инъективна, т.к. $\forall\, x_1 \neq x_2$ выполняется $e^{x_1} \neq e^{x_2}$; но не сюръективна, т.к. $D_{f_1} = R, E_{f_1} \subseteq R$ (см. рисунок 1.2.4);

2) $f_2(x) = x^3 - 4x$ - сюръективна, т.к. $E_{f_2} = R$, но не инъективна, поскольку $\exists\ x_1 \neq x_2$, но $f_2(x_1) = f_2(x_2)$ (см. рисунок 1.2.5);

3) $f_3(x) = x + 3$ - биективна, каждому $x \in R$ соответствует единственный $y = f_3(x) \in R$ и, наоборот, (график – прямая линия).

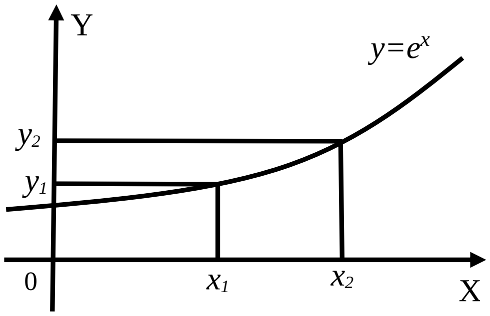


Рисунок 1.2.4

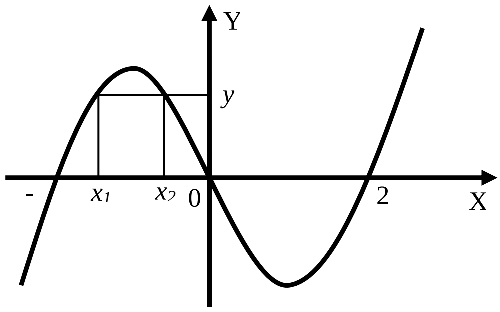


Рисунок 1.2.5

Заметим, что свойства этих, а также других функций проще всего определять по их графикам.

П р и м е р 1.2.13 Рассмотрим функции f_i: $[0,1] \to [0,1], (i = 1,2,3,4)$, графики которых изображены на рисунке 1.2.6:

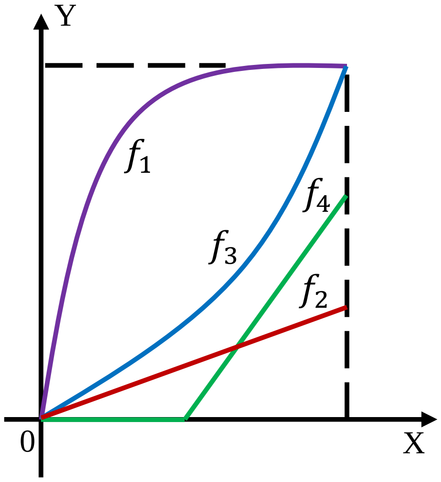


Рисунок 1.2.6

По графикам можно установить, что

а) $f_1(x)$ - сюръекция (не инъекция);

б) $f_2(x)$ - инъекция (не сюръекция);

в) $f_3(x)$ - биекция;

г) $f_4(x)$ - не инъективная и не сюръективная функция.

1.3 Понятие о мощности множеств

Часто возникает необходимость сравнивать множества по числу элементов. В этом случае возникает понятие мощности множества.

Определение. Множества A и B называются эквивалентными (обозначается $A \sim B$) если существует биекция f: $A \leftrightarrow B$ (т.е. между ними можно установить взаимно однозначное соответствие (в.о.с.)).

Очевидно, что два конечных множества эквивалентны тогда и только тогда, когда они имеют одинаковое число элементов. Если два множества бесконечны и между ними можно установить в.о.с., то они обладают также чем-то общим, что мы будем называть мощностью. Итак, любые два эквивалентных множества (конечные или бесконечные) имеют одинаковую мощность или являются равномощными. Или более точное определение.

Определение. Мощностью множества A называется класс всех множеств, эквивалентных множеству A (мощность обозначается $|A|$).

Имеется три возможности:

а) если A - конечное множество, имеет n элементов, то $|A|= n$;

б) если A - бесконечное множество и эквивалентно множеству натуральных чисел N, то A называют счётным множеством, записывают $|A| = \omega$. Таким образом, у счётного множества все элементы можно пронумеровать;

в) существуют бесконечные множества, которые нельзя привести во в.о.с. с множеством натуральных чисел. Например, установлено, что множество всех действительных чисел отрезка [0,1] не является счётным (теорема Кантора). Принято мощность этого множества называть континуум (часто обозначается с), а множества такой мощности континуальными.

Доказано, что если A континуальное множество, то $|A| = c = 2^{\omega}$, т.е. мощность континуума равна мощности множества всех подмножеств счётного множества. Вообще, для любого множества A: $\mathcal{P}(A)| = 2^{|A|}$, где $\mathcal{P}(A)$ – булеан.

Примеры счётных множеств:

а) множество целых чисел Z, а также Z^+, Z^-;

б) множество рациональных чисел Q;

в) любое бесконечное подмножество множества N, например, $\{2,4,6,8,\dots\}$;

г) N^2 (вообще $N^n \sim N$).

Примеры континуальных множеств:

а) множество всех действительных чисел R или множество точек числовой оси;

б) множество всех точек плоскости (пространства) $R \times R (R \times R \times R)$;

в) множество всех подмножеств счётного множества (т.е. булеан счётного множества).

Как показано в теории множеств, для множества любой мощности множество его подмножеств имеет более высокую мощность. Поэтому не существует множества максимальной мощности. На мощность множеств можно смотреть как на новый объект, называемый кардинальным числом или кардиналом. Примерами кардиналов могут быть:

а) любое натуральное число (как мощность конечного множества);

б) $\omega, 2^{\omega} = c, 2^{2^{\omega}}$ и т.д.

Отметим, что существование биекции между двумя множествами позволяет перенести изучение свойств с одного множества на другое, что многое упрощает, например, некоторые свойства конечного множества A, $|A| = n$ можно изучать по множеству $\{0, 1, 2, 3, \dots, n\text{-}1\}$.

2 Элементы математической логики

2.1 Высказывания и логические операции

Логику чаще всего определяют как анализ методов рассуждений. Эти методы позволяют определить истинность или ложность того или иного рассуждения, причём логику интересует прежде всего форма, а не содержание доводов в этом рассуждении. Если при этом применяют математический аппарат и изучают математические рассуждения, то логику называют математической логикой. Математическая логика играет большую роль в основах математики, она – фундамент, на котором построено здание математики. Логика применяется в информатике для построения компьютерных программ и доказательства их корректности. Понятия, методы и средства математической логики лежат в основе современных информационных технологий.

Определение. Высказыванием называется повествовательное предложение, о котором в данной ситуации можно говорить, что оно истинно или ложно.

П р и м е р 2.1.1 Высказывание: «дважды два – четыре» - истинно; высказывание: «в году – 360 дней» - ложно.

Определение. Простое (элементарное) высказывание рассматривается как некое неделимое целое. Обозначается $A, B, C, \dots, P \dots$; сложным (составным) называется высказывание, составленное из простых с помощью логических связок (операций).

Основными логическими операциями (связками) являются:

а) конъюкция (операция «и», логическое произведение).

Конъюнкцией двух высказываний P и Q (обозначается $P \wedge Q, P \cdot Q, PQ, P\&Q$, читается " P и Q ") называется высказывание, истинное тогда и только тогда, когда истинны оба высказывания и ложное во всех остальных случаях;

б) дизъюнкция (операция «или», логическая сумма). Дизъюнкцией двух высказываний P и Q (обозначается $P \vee Q, P + Q$, читается "P или Q") называется высказывание, ложное, если оба высказывания ложные и истинное во всех остальных случаях;

в) отрицание (инверсия). Отрицанием высказывания P (обозначается $\overline{P}$, $\neg P$, читается "не P ") называется высказывание, истинное, когда P ложное и ложное, когда P истинное;

г) импликация (логическое следование). Импликацией двух высказываний P и Q (обозначается $P \rightarrow Q$, $P \supset Q$, читается, "если P, то Q", " P влечёт Q ") называется высказывание, ложное, когда P истинное, а Q ложное и истинное во всех остальных случаях;

д) эквиваленция (эквивалентность). Эквиваленцией двух высказываний P и Q (обозначается $P \sim Q, P \leftrightarrow Q, P \equiv Q$, читается “ P эквиваентно Q ”, “ P тогда и только тогда, когда Q ”) называется высказывание, истинное, когда P и Q - оба истинны или оба ложны и ложное во всех остальных случаях.

Формулы алгебры логики

Рассмотрим содержание логики высказываний, используя язык алгебры логики, которая изучает строение логических высказываний (т.е. логических формул) и способы установления их истинности с помощью алгебраических методов.

Поставим в соответствие высказыванию P логическую переменную x, которая принимает значение 1, если P истинно, 0, если P ложно. Из логических переменных можно с помощью логических операций составлять различные конструкции, которые являются формулами алгебры логики.

Определение формулы:

а) любая логическая переменная является формулой**;**

б) если φ и ψ формулы, то выражения $\overline{\varphi}$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \sim \psi$ являются формулами;

в) никаких других формул, кроме построенных в а) и б) нет.

Если формула φ построена из логических переменных, принадлежащих множеству $\{x_1, x_2, \ldots, x_n\}$, то будем писать $\varphi(x_1, x_2, \ldots x_n)$.

Действия логических операций задаются таблицами истинности, в каждой строке которых отмечены различные наборы значений переменных и соответствующее им значение формулы. Составим таблицы истинности логических операций в соответствии с их определением:

Таблица 2.1.1

x	y	$x \wedge y$	$x \vee y$	$x \to y$	$x \leftrightarrow y$	$\overline{x}$
0	0	0	0	1	1	1
0	1	0	1	1	0	
1	0	0	1	0	0	0
1	1	1	1	1	1	

Таблицы истинности называются также интерпретациями логических операций, они придают формулам смысл в отличие от формальных законов их построения, данных в определении формулы.

Исходя из таблиц истинности для логических операций, можно строить таблицы истинности для произвольных формул.

П р и м е р 2.1.2 Для формулы $\varphi(x, y, z) = z \to (\overline{x} \vee \overline{y})$ таблица истинности имеет вид:

Таблица 2.1.2

x	y	z	$\overline{x}$	$\overline{y}$	$\overline{x} \vee \overline{y}$	$z \to (\overline{x} \vee \overline{y})$
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	1	1	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	0	0	0	1
1	1	1	0	0	0	0

При рассмотрении вопроса о минимальном количестве логических связок (операций), необходимых для выражения любого высказывания, возникла необходимость ввести новые логические операции, добавляя к пяти основным ещё три операции:

е) штрих Шеффера (антиконъюнкция). Обозначается x | y.

По определению $(x \mid y) = \neg\,(x \wedge y)$, или $(x \mid y) = \overline{x \wedge y}$;

ж) стрелка Пирса (антидизъюнкция). Обозначается x ↓ y.

По определению $(x \downarrow y) = \neg\,(x \vee y)$ или $(x \downarrow y) = \overline{x \vee y}$;

и) кольцевая сумма (сложение по модулю два, антиэквиваленция). Обозначается $x \oplus y$. Определяется $x \oplus y = \neg\,(x \leftrightarrow y)$ или $x \oplus y = \overline{x \sim y}$.

Составим таблицы истинности этих операций, исходя из определений.

Таблица 2.1.3

x	y	$x \mid y$	$x \downarrow y$	$x \oplus y$
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0

При тождественных преобразованиях, при составлении формул важно знать приоритеты операций (какая сильнее, какая слабее). Схему приоритетов можно представить следующими способами:

а) на множестве $\{\neg, (\wedge, \mid, \downarrow), \vee, \to, (\leftrightarrow, \oplus)\}$ знаки операций расположены в порядке убывания старшинства, в круглых скобках указаны равносильные операции;

б) на множестве $\{\neg, \wedge, \vee, \to, \leftrightarrow, |, \downarrow, \oplus\}$ вводится транзитивное отношение « > » - быть более сильным и отношение « ~ » - быть равносильным, по правилам, указанным на схеме:

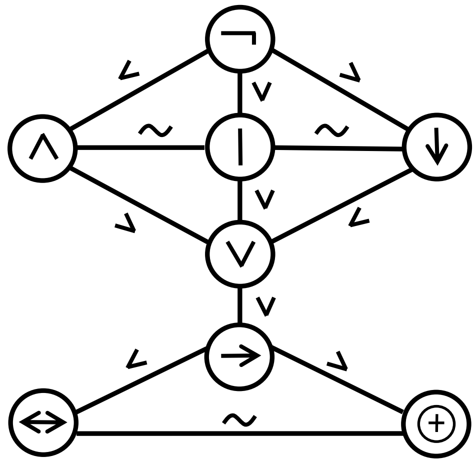


Рисунок 2.1.1

Кроме того, приняты следующие соглашения:

а) внешние скобки не пишутся, например, вместо $((x \wedge y) \to z)$, будем писать $(x \wedge y) \to z$;

б) недостающие скобки в формуле расставляют последовательно, начиная с наиболее сильных связок и кончая наиболее слабыми;

в) для равносильных связок расстановка скобок выполняется слева направо.

П р и м е р 2.1.3

а) в формуле $x \wedge y \vee z$ скобки расставляются так: $(x \wedge y) \vee z$;

б) в формуле $x \vee y \leftrightarrow z \to u$ скобки расставляются так: $(x \vee y) \leftrightarrow (z \to u)$;

в) в формуле $x \to (\overline{y} \to z)$ скобки опустить нельзя, т.к. в силу наших соглашений выражению $x \to \overline{y} \to z$ соответствует формула $(x \to \overline{y}) \to z$.

2.2 Функции алгебры логики

Каждая формула представляет логическую функцию от логических переменных, которые могут принимать только два значения 0 и 1.

Определение. Функцией алгебры логики (логической функцией) от n переменных $x_1, x_2, \ldots, x_n$ (обозначается $f(x_1, \ldots, x_n)$) называется любая функция, которая произвольному набору $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ нулей и единиц ставит в соответствие значение $f(\sigma_1, \sigma_2, \ldots, \sigma_n) \in \{0,1\}$, т.е. $f: \{0,1\}^n \to \{0,1\}$.

Функции алгебры логики называются также булевыми, двоичными или переключательными. Множество таких функций обозначается P_2, $P_2(n), P_n$.

Наиболее распространёнными способами задания логических функций являются следующие:

а) задание таблицей истинности, в левой части которой выписаны все возможные наборы значений аргументов $x_1, x_2, \ldots, x_n$, а в правой – столбец значений f, соответствующих этим наборам. Число всех наборов 0 и 1 функции n переменных будет равно 2^n, т.е. для функции одной переменной - $2^1 = 2$, двух - $2^2 = 4$, трёх - $2^3 = 8$ и т.д. Таким образом, таблица истинности функции одной переменной содержит 2 строки, двух – 4, трёх – 8 и т.д. При этом множество значений аргументов принято упорядочивать по лексикографическому порядку, или в порядке возрастания записываются числа от 0 до 2^n в двоичной системе счисления:

Таблица 2.2.1

x_1		x_{n-1}	x_n	$f(x_1, \ldots, x_n)$
0	…	0	0	$f(0, \ldots ,0,0)$
0	…	0	1	$f(0, \ldots ,0,1)$
0	…	1	0	$f(0, \ldots ,1,0)$
…	…	…	…	…
1	…	1	1	$f(1, \ldots ,1,1)$

б) задание функций перечислением всех наборов, на которых f принимает значение 0 (нулевые наборы) и всех наборов, на которых она принимает значение 1 (единичные наборы);

в) задание функции формулой;

г) задание с помощью вектора значений. Вектором значений функции $f(x_1, \ldots, x_n)$ называется упорядоченный набор всех значений f, при котором все значения упорядочены по лексикографическому порядку множества аргументов $\{0,1\}^n$.

Рассмотрим на примере эти способы задания функции.

П р и м е р 2.2.1 Имеется устройство, фиксирующее принятие некоторой резолюции тремя персонами (комитетом трёх). Каждый член комитета при одобрении резолюции нажимает свою кнопку. Если большинство согласно, то резолюция принята, что фиксирует устройство. Таким образом, устройство реализует функцию $f(x, y, z)$, таблица истинности которой имеет вид:

Таблица 2.2.2

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Зададим эту функцию нулевыми и единичными наборами: $f(0{,}0{,}0) = = f(0{,}0{,}1) = f(0{,}1{,}0) = f(1{,}0{,}0) = 0$ - нулевые наборы; $f(0{,}1{,}1) = f(1{,}0{,}1) = = f(1{,}1{,}0) = f(1{,}1{,}1) = 1$ - единичные наборы. Если задавать эту функцию с помощью вектора значений, то это будет набор (00010111).

Иногда применяют и другие способы задания функций (см., например, [3], стр.124 или [9], стр.124).

Итак, число всех наборов 0 и 1 для функции n переменных равно 2^n. Число же всех возможных различных функций n переменных равно числу всех возможных расстановок 0 и 1 в столбце с 2^n строками, т.е. равно 2^{2^n}. Значит, если $B^n = \{0,1\}^n$- множество всех значений аргументов функции $f(x_1, \dots, x_n)$, а $P_2(n)$ - множество всех функций от $x_1, x_2, \dots, x_n$, то мощности этих множеств будут $|B^n| = 2^n$, $|P_2(n)| = 2^{2^n}$. Таким образом, $|P_2(n)|$ растёт очень быстро: $|P_2(1)| = 4$, $|P_2(2)| = 16$, $|P_2(3)| = 256$ и т.д.

Особую роль в алгебре логики играют функции одной и двух переменных. Например, множество всех логических функций одной переменной $P_2(1)$ состоит из четырёх функций, которые можно представить их таблицей истинности:

Таблица 2.2.3

x	f_1	f_2	f_3	f_4
0	0	0	1	1
1	0	1	0	1
обозначение	0	x	$\overline{x}$	1
название	нуль	тождественная	отрицание	единица

Функции f_1 и f_4 константы 0 и 1, $f_2 = x$ – повторяет переменную x, (т.е. значения этих функций не зависят от переменной x, поэтому говорят, что переменная x не существенна или фиктивна для этих функций). Наибольший интерес представляет функция $f_3 = \overline{x}$ - унарная операция отрицания. Аналогичным образом можно построить таблицу истинности для всех $2^{2^2} = 16$ функций двух переменных. Среди них 7 логических операций (конъюнкция, дизъюнкция, и т.д.), остальные не представляют интереса. Заметим, что функция может быть представлена как операция, если её значения лежат в области определения этой функции. В этом смысле все функции математической логики могут быть представлены операциями.

Логические функции трёх и более переменных обычно задаются либо таблицей истинности, либо формулой, состоящей из знаков переменных и знаков унарных и бинарных операций. В общем случае формула описывает логическую функцию как суперпозицию других, более простых функций.

Определение. Функцию $f(x_1,\dots,x_n)$ называют суперпозицией функций $g(y_1,\dots,y_m)$ и $y_1 = h_1(x_1,\dots,x_n),\dots, y_m = h_m(x_1,\dots,x_n)$, если

$$f(x_1,\dots,x_n) = g\big(h_1(x_1,\dots x_n),\dots,h_m(x_1,\dots,x_n)\big).$$

П р и м е р 2.2.2 Функция $f(x_1,x_2,x_3,x_4) = x_1x_2 \downarrow x_2x_3 \downarrow x_4x_1$ есть суперпозиция $g(y_1,y_2,y_3) = y_1 \downarrow y_2 \downarrow y_3$ и $y_1 = x_1x_2$, $y_2 = x_2x_3$, $y_3 = x_4x_1$.

Эквивалентность формул. Основные эквивалентные соотношения алгебры логики

Одна и та же функция может быть представлена различными формулами. В этом случае возникает понятие эквивалентности формул.

Определение. Формулы $\varphi(x_1,\dots,x_n)$ и $\psi(x_1,\dots,x_n)$ называются эквивалентными или равносильными (обозначается $\varphi \sim \psi$, $\varphi \equiv \psi$, $\varphi = \psi$), если они представляют одну и ту же функцию.

П р и м е р 2.2.3 Построим таблицы истинности для формул $\varphi = x \to y$ и $\psi = \overline{y} \to \overline{x}$

Таблица 2.2.4

x	y	$\overline{y}$	$\overline{x}$	φ	ψ
0	0	1	1	1	1
0	1	0	1	1	1
1	0	1	0	0	0
1	1	0	0	1	1

Так как столбцы значений этих формул совпадают, то они представляют одну и ту же функцию и поэтому эквивалентны: $(x \to y) \sim (\overline{y} \to \overline{x})$.

В этом примере для установления эквивалентности формул применён метод построения их таблиц истинности и сравнения полученных таблиц по каждому набору значений переменных.

Другим методом являются эквивалентные преобразования формул, которые используют эквивалентные соотношения (законы) и правила подстановки и замены.

Корректность преобразований обеспечивается выполнением следующих двух правил:

а) (правило подстановки): если в исходном эквивалентном соотношении все вхождения переменной x одновременно заменены формулой φ, то получим новое эквивалентное соотношение;

б) (правило замены): если какая - либо формула φ, описывающая функцию f, содержит подформулу ψ, то замена ψ на ψ_1 ($\psi = \psi_1$) не изменит функции f.

Перечислим теперь основные эквивалентные соотношения (законы), которые не выводятся друг из друга, доказать их справедливость можно с помощью таблиц истинности.

Таблица 2.2.5

1	Коммутативность	$x \wedge y = y \wedge x$	$x \vee y = y \vee x$
2	Ассоциативность	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$	$(x \vee y) \vee z = x \vee (y \vee z)$
3	Дистрибутивность	$x \wedge (y \vee z) =$ $= (x \wedge y) \vee (x \wedge z)$	$x \vee (y \wedge z) =$ $= (x \vee y) \wedge (x \vee z)$
4	Идемпотентность	$x \wedge x = x$	$x \vee x = x$
5	Законы поглощения	$x \wedge (x \vee y) = x$	$x \vee (x \wedge y) = x$
6	Законы де Моргана	$\overline{x \wedge y} = \bar{x} \vee \bar{y}$	$\overline{x \vee y} = \bar{x} \wedge \bar{y}$
7	Закон двойного отрицания $\bar{\bar{x}} = x$		
8	Свойства констант	$x \wedge 1 = x \quad x \wedge 0 = 0$	$x \vee 1 = 1 \quad x \vee 0 = x$
9	$x \wedge \bar{x} = 0$ - закон противоречия	$x \vee \bar{x} = 1$ - закон исключённого третьего	

Наряду с основными эквивалентностями, часто используют ещё некоторые, которые можно вывести из основных или доказать с помощью таблиц истинности, сведём их также в таблицу.

Таблица 2.2.6

10	Законы склеивания	$(x \wedge y) \vee (x \wedge \bar{y}) = x$	$(x \vee y) \wedge (x \vee \bar{y}) = x$
10 а	Законы расщепления	$x = (x \wedge y) \vee (x \wedge \bar{y})$	$x = (x \vee y) \wedge (x \vee \bar{y})$
11	Обобщённое склеивание	$(x \wedge z) \vee (y \wedge \bar{z}) \vee (x \wedge y) = (x \wedge z) \vee (y \wedge \bar{z})$	
12	$x \vee (\bar{x} \wedge y) = x \vee y$	$\bar{x} \vee (x \wedge y) = \bar{x} \vee y$	
13	$x \wedge (\bar{x} \vee y) = x \wedge y$	$\bar{x} \wedge (x \vee y) = \bar{x} \wedge y$	
14	$x \leftrightarrow y = (x \to y) \wedge (y \to x) = (\bar{x} \vee y) \wedge (x \vee \bar{y}) = xy \vee \bar{x}\,\bar{y}$		
15	$(x \to y) = \bar{x} \vee y$	$x \mid y = \overline{x \wedge y}$	$x \downarrow y = \overline{x \vee y}$
16	$x \oplus y = \overline{x \leftrightarrow y} = \overline{(\bar{x} \vee y) \wedge (\bar{y} \vee x)} = \overline{xy \vee \bar{x}\,\bar{y}} = x\bar{y} \vee \bar{x}y$		

Определение. Формула $\varphi(x_1, \ldots, x_n)$ называется выполнимой (опровержимой), если существует такой набор значений переменных, при котором формула принимает значение 1 (0).

П р и м е р 2.2.4 Формула $\varphi = x \wedge y$ является одновременно и выполнимой, и опровержимой, т.к. $1 \wedge 1 = 1$ и $0 \wedge 0 = 0$.

Определение. Формула $\varphi(x_1, \ldots, x_n)$ называется тождественно истинной, общезначимой или тавтологией (тождественно ложной или противоречием), если эта формула принимает значение 1 (соответственно 0) при всех наборах значений переменных (т.е. функция является константой 1 (0)).

П р и м е р 2.2.5 Формула $x \vee \overline{x}$ тождественно истинна, т.к. $x \vee \overline{x} = 1$ для всех значений x; формула $x \wedge \overline{x}$ тождественно ложна, т.к. $x \wedge \overline{x} = 0$ для всех x.

Таблица 2.2.7

x	$\overline{x}$	$x \vee \overline{x}$	$x \wedge \overline{x}$
0	1	1	0
1	0	1	0

Заметим, что, так как формула может представлять какое-то логическое высказывание или рассуждение, то это рассуждение будет логически правильным, если представляющая его формула тождественно истинна.

Двойственность

Определение. Функция $f^*(x_1, \ldots, x_n)$ называется двойственной к функции $f(x_1, \ldots, x_n)$, если $f^*(x_1, \ldots, x_n) = \overline{f(\overline{x_1}, \ldots, \overline{x_n})}$; функция, двойственная к самой себе, называется самодвойственной, т.е. $f^* = f$.

Отношение двойственности симметрично (если $f^* = g$, то $g^* = f$) и инволютивно ($f^{**} = f$).

П р и м е р 2.2.6 Дизъюнкция двойственна конъюнкции, конъюнкция – дизъюнкции, константа 1 – константе 0 и константа 0 – константе 1, отрицание самодвойственно. Действительно, например, для $f = x \wedge y$ имеем:

$$f^* = \overline{\overline{x} \wedge \overline{y}} = \overline{\overline{x}} \vee \overline{\overline{y}} = x \vee y; \text{ для } f = \overline{x} \Rightarrow f^* = \overline{\overline{\overline{x}}} = \overline{x}, \text{ т.е. } f^* = f.$$

Двойственную функцию можно получить также с помощью таблиц истинности, заменяя в таблице истинности функции f все значения на противоположные.

Принцип двойственности: функция, двойственная суперпозиции функций, есть соответствующая суперпозиция двойственных функций, т.е. если $f(x_1, \ldots, x_2) = g(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n))$, то

$$f^*(x_1, \ldots, x_2) = g^*(h_1^*(x_1, \ldots, x_n), \ldots, h_m^*(x_1, \ldots, x_n)).$$

В булевой алгебре принцип двойственности имеет более конкретный вид: если в формуле F , представляющей функцию f, все конъюнкции заменить на дизъюнкции, дизъюнкции на конъюнкции, 1 на 0, 0 на 1, то получим формулу F^*, представляющую двойственную функцию f^*.

П р и м е р 2.2.7 Для функции $f(x, y, z) = xy \vee \overline{x}\,\overline{y} \vee \overline{x}y\overline{z}$ получить двойственную функцию.

1 способ: по определению $f^* = \overline{\overline{x}\,\overline{y} \vee \overline{\overline{x}}\,\overline{\overline{y}} \vee \overline{\overline{x}}\,\overline{y}\,\overline{\overline{z}}} = (x \vee y) \wedge (\overline{x} \vee \overline{y}) \wedge (\overline{x} \vee y \vee \overline{z})$;

2 способ: построим таблицу истинности для f (см. таблицу 2.2.8), затем заменим в ней все значения на противоположные, получим таблицу для f^* (см.

таблицу 2.2.9). Чтобы получить таблицу функции f^* в обычном виде, где значения переменных расположены в лексикографическом порядке, нужно все столбцы последней таблицы «перевернуть» (см. таблицу 2.2.10).

Т а б л и ц а 2.2.8

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Т а б л и ц а 2.2.9

x	y	z	f^*
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	0
0	0	0	0

Т а б л и ц а 2.2.10

x	y	z	f^*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

3 способ: т.к. формула f записана в булевой форме, т.е. только с операциями конъюнкции, дизъюнкции и отрицания, то по принципу двойственности в булевой алгебре для получения двойственной функции надо в данной формуле заменить все конъюнкции на дизъюнкции, дизъюнкции на конъюнкции: $f(x,y,z) = xy \vee \overline{x}\,\overline{y} \vee \overline{x}y\overline{z} \ \ \rightarrow\ f^* = (x \vee y) \wedge (\overline{x} \vee \overline{y}) \wedge (\overline{x} \vee y \vee \overline{z})$.

Классы Поста. Полные системы логических функций

Пусть дана булева функция $f(x_1, \ldots, x_n)$. Введём определение классов булевых функций или классов Поста:

а) говорят, что функция f сохраняет константу 0, если $f(0,0,\ldots,0) = 0$. Множество всех функций, сохраняющих 0, образует класс P_0 ($P_0 \subset P_2$);

б) говорят, что функция f сохраняет константу 1, если $f(1,1,\ldots,1) = 1$. Множество всех функций, сохраняющих 1, образует класс P_1 ($P_1 \subset P_2$);

в) обозначим через S множество или класс самодвойственных функций;

г) функция f называется линейной, если она может быть записана в виде $f(x_1, \ldots, x_n) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_nx_n$, где $c_k \in \{0; 1\}, (k = 0,1,2, \ldots, n)$. Класс линейных функций обозначается L;

д) функция f называется монотонной, если для любых наборов нулей и единиц $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ и $(\beta_1, \beta_2, \ldots, \beta_n)$ из условий $\alpha_1 \leq \beta_1, \ldots, \alpha_n \leq \beta_n$ следует $f(\alpha_1, \alpha_2, \ldots, \alpha_n) \leq f(\beta_1, \beta_2, \ldots, \beta_n)$. Обозначим через M класс монотонных функций. Каждый класс Поста замкнут относительно операций замены переменных и суперпозиции, т.е. с помощью этих операций из функций данного класса можно получить только функции этого же класса.

В таблице ниже рассмотрены примеры принадлежности некоторых булевых функций к классам Поста (принадлежит – (+); не принадлежит – (-)).

Т а б л и ц а 2.2.11

Функция	P_0	P_1	L	M	S
Отрицание	-	-	+	-	+
Конъюнкция	+	+	-	+	-
Дизъюнкция	+	+	-	+	-
Импликация	-	+	-	-	-

Одна и та же логическая функция может быть задана формулами, включающими различные наборы логических операций. Например: $x_1|x_2 = \overline{x_1 \wedge x_2} = \overline{x_1} \vee \overline{x_2}$. Существуют наборы логических операций (функций), через которые можно выразить любые другие логические функции.

Определение. Система логических функций Σ называется функционально полной системой или базисом, если любая логическая функция является суперпозицией функций из Σ.

Ответ на вопрос о полноте произвольной системы функций даёт следующая теорема.

Теорема Поста. Для того чтобы система булевых функций была полной, необходимо и достаточно, чтобы она содержала хотя бы одну функцию, не сохраняющую 0, хотя бы одну функцию, не сохраняющую 1, хотя бы одну не самодвойственную функцию, хотя бы одну не линейную и хотя бы одну не монотонную функции.

При определении полноты булевых функций можно пользоваться таблицами принадлежности этих функций классам Поста. Например, так как в системе $\Sigma_1 = \{\wedge, \neg\}$ отрицание не сохраняет констант и не монотонно, а конъюнкция не линейна и не самодвойственна (см. таблицу 2.2.9), то эта система полна. Примерами функционально полных систем являются $\{\wedge,\ \vee,\ \neg\}$, $\{\wedge,\ \neg\}, \{\vee,\ \neg\}, \{\,|\,\}, \{\downarrow\}, \{\wedge,\ \oplus,\ 1\}, \{\rightarrow,\ \neg\}, \{\vee,\ \leftrightarrow,\ \oplus\}, \{\wedge,\ \leftrightarrow,\ \oplus\}$ и другие. Формулы, содержащие, кроме переменных и скобок, только операции конъюнкции, дизъюнкции и отрицания, называются булевыми.

Теорема. Всякая логическая функция может быть представлена булевой формулой (т.е. как суперпозиция конъюнкции, дизъюнкции и отрицания).

Отсюда следует, что система $\{\wedge,\ \vee, \neg\}$ функционально полна. Для доказательства функциональной полноты какого-либо другого набора операций, кроме указанного выше способа, достаточно показать, что через операции набора можно выразить конъюнкцию, дизъюнкцию и отрицание.

Справедливо и более общее утверждение, позволяющее доказать полноту системы другим способом.

Теорема. Если все функции функционально полной системы Σ^* представимы формулами над Σ, то Σ также функционально полна.

Пример 2.2.8 - Рассмотрим системы $\Sigma_1 = \{\wedge, \neg\}$ и $\Sigma_2 = \{\vee, \neg\}$ и базис $\Sigma_0 = \{\wedge, \vee, \neg\}$. Докажем, что Σ_1, Σ_2 также базисы. Действительно, в каждой из этих систем недостающая до Σ_0 операция может быть выражена через остальные две:

$$\left.\begin{array}{l}\text{для } \Sigma_1 \text{ не достаёт «}\vee\text{»: } x_1 \vee x_2 = \overline{\overline{x_1} \wedge \overline{x_2}} \\ \text{для } \Sigma_2 \text{ не достаёт «}\wedge\text{»: } x_1 \wedge x_2 = \overline{\overline{x_1} \vee \overline{x_2}}\end{array}\right\} \quad (*) .$$

Формула $f = x_1 x_2 \vee \overline{x_2}(x_3 \vee x_4)$ в системах Σ_1, Σ_2 перейдёт соответственно в формулы: $f = [\Sigma_1] = \overline{\overline{x_1 \cdot x_2} \cdot \overline{\overline{x_2} \cdot \overline{\overline{x_3} \cdot \overline{x_4}}}}$, $f = [\Sigma_2] = \overline{\overline{x_1} \vee \overline{x_2}} \vee \overline{x_2 \vee \overline{x_3 \vee x_4}}$.

Таким образом, с точки зрения функциональной полноты систему Σ_0 можно считать избыточной, т.к. она сохраняет свойство полноты и при удалении из неё дизъюнкции или конъюнкции. Но, как мы видим, за не избыточность систем Σ_1, Σ_2 приходится платить избыточностью формул, т.к. каждая замена одной операции на другую по формулам $(*)$ вносит в формулу лишнее отрицание.

2.3 Дизъюнктивные и конъюнктивные нормальные формы

Базис $\Sigma_0 = \{\wedge, \vee, \neg\}$ наиболее изучен и имеет самое широкое применение на практике.

Определение. Элементарной конъюнкцией (дизъюнкцией) называется конъюнкция (дизъюнкция) переменных или их отрицаний.

Пример 2.3.1

а) $x \vee \overline{y} \vee \overline{z}$ и $x \vee y \vee \overline{x}$ –элементарные дизъюнкции;

б) $\overline{x_1}\, x_2\, x_3$ и $x_1 \wedge x_2 \wedge \overline{x_3}$ –элементарные конъюнкции;

в) $\overline{x}$ – одновременно является и элементарной дизъюнкцией и элементарной конъюнкцией.

Определение. Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция элементарных конъюнкций. Конъюнктивной нормальной формой (КНФ) называется конъюнкция элементарных дизъюнкций.

Пример 2.3.2

а) $x\overline{y} \vee yz$ –ДНФ; б) $(x \vee \overline{y})(\overline{x} \vee z)(\overline{x} \vee y \vee \overline{z})$ - КНФ.

Теорема. Любая формула может быть приведена к ДНФ (КНФ) (т.е. любая формула эквивалентна некоторой ДНФ (КНФ)).

Правило приведения формулы к ДНФ:

а) все логические операции, присутствующие в формуле, выразить через $\{\wedge,\ \vee, \neg\}$, используя эквивалентности:

1) $(x \to y) = \overline{x} \vee y$;

2) $x \leftrightarrow y = (\,\overline{x} \vee y) \wedge (\overline{y} \vee x) = xy \vee \overline{x}\,\overline{y}$;

3) $x \mid y = \overline{x \wedge y}$;

4) $x \downarrow y = \overline{x \vee y}$;

5) $x \oplus y = \overline{x \leftrightarrow y} = \overline{(\,\overline{x} \vee y) \wedge (\,\overline{y} \vee x)} = x\overline{y} \vee \overline{x}y$;

б) перенести все отрицания к переменным по законам де Моргана:
$\overline{x \wedge y} = \overline{x} \vee \overline{y},\ \overline{x \vee y} = \overline{x} \wedge \overline{y}$;

в) используя закон дистрибутивности, преобразовать формулы так, чтобы все конъюнкции выполнялись раньше дизъюнкций:
$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

П р и м е р 2.3.3 Приведём к ДНФ формулу $\varphi = (x \to y) \downarrow \overline{y \to z}$. Для этого заменим $\to, \downarrow$ на $\wedge,\ \vee, \neg$, затем применим закон де Моргана и закон двойного отрицания: $\varphi = (\overline{x} \vee y) \downarrow \left(\overline{\overline{y} \vee z}\right) = \overline{(\overline{x} \vee y) \vee \overline{(\overline{y} \vee z)}} = \overline{(\overline{x} \vee y)} \wedge \overline{\overline{(\overline{y} \vee z)}} =$

$= \left(\overline{\overline{x}} \wedge \overline{y}\right) \wedge (\overline{y} \vee z) = (x \wedge \overline{y}) \wedge (\overline{y} \vee z) = (x \wedge \overline{y} \wedge \overline{y}) \vee (x \wedge \overline{y} \wedge z)$.

Заметим, что последняя формула в примере в некоторых учебниках уже считается ДНФ, в других же считают, что в элементарных конъюнкциях и дизъюнкциях каждая переменная должна встречаться не более одного раза. Для удаления лишних переменных применяют следующие эквивалентности:

а) $x \vee x = x, x \wedge x = x$ (законы идемпотентности);

б) $x \vee \overline{x} = 1$ (закон исключённого третьего), $x \wedge \overline{x} = 0$ (закон противоречия);

в) $x \wedge 1 = x,\ x \vee 1 = 1,\ x \wedge 0 = 0,\ x \vee 0 = x$ - (свойства констант).

Поэтому, используя закон идемпотентности, в последнем примере получим ДНФ: $\varphi = (x \wedge \overline{y}) \vee (x \wedge \overline{y} \wedge z)$.

Приведение формулы к КНФ производится так же, как к ДНФ, только вместо пункта в) применяется пункт в'):

в') используя закон дистрибутивности, преобразовать формулы так, чтобы все дизъюнкции выполнялись раньше конъюнкций, т.е.

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

П р и м е р 2.3.4 Приведём к КНФ формулу $\varphi = (x \to y) \land ((\overline{y} \to z) \to \overline{x})$.
Заменим операцию $\to$, используя формулу $(x \to y) = \overline{x} \lor y$:

$\varphi = (\overline{x} \lor y) \land ((\overline{\overline{y} \to z}) \lor \bar{x}) = (\overline{x} \lor y) \land (\left(\overline{\overline{\overline{y}} \lor z}\right) \lor \overline{x})$ = [закон де Моргана, двойное отрицание]= $(\overline{x} \lor y) \land (\overline{\overline{y}} \land \overline{z}) \lor \overline{x}) = (\overline{x} \lor y) \land ((\overline{y} \land \overline{z}) \lor \overline{x}) = (\overline{x} \lor \lor y) \land (\overline{y} \lor \overline{x}) \land (\overline{z} \lor \overline{x})$ - КНФ.

ДНФ и КНФ имеют тот недостаток, что они не обладают свойством единственности, т.е. одна и та же функция имеет несколько ДНФ и КНФ. Этим недостатком не обладают совершенные нормальные формы.

Определение. Совершенной дизъюнктивной нормальной формой (СДНФ) называется ДНФ, в которой в каждую элементарную конъюнкцию каждая переменная входит ровно один раз, причём, входит либо сама переменная, либо её отрицание, и среди элементарных конъюнкций не должно быть одинаковых.

Определение. Совершенной конъюнктивной нормальной формой (СКНФ) называется КНФ, в которой в каждую элементарную дизъюнкцию каждая переменная входит ровно один раз, причём, входит либо сама переменная, либо её отрицание, и среди элементарных дизъюнкций не должно быть одинаковых.

П р и м е р 2.3.5

а) $x_1\overline{x_2}x_3 \lor \overline{x_1}x_2x_3$ - СДНФ;

б) $(x \lor y \lor \overline{z}) \land (\overline{x} \lor \overline{y} \lor z) \land (\overline{x} \lor y \lor z)$ - СКНФ;

в) $x_1\overline{x_2}x_3 \lor \overline{x_1}x_2x_3 \lor x_1\overline{x_2}x_3$ - не СДНФ, т.к. содержит две одинаковых элементарных конъюнкции;

г) $x_1\overline{x_1}x_3 \lor x_1x_2x_3$ - не СДНФ, т.к. в одной элементарной конъюнкции содержится и переменная и её отрицание: $x_1, \overline{x_1}$.

Теорема. (Существование и единственность СДНФ и СКНФ). Всякая логическая формула единственным образом (с точностью до порядка следования элементарных конъюнкций (дизъюнкций)) может быть представлена в СДНФ (СКНФ).

Для приведения формулы к СДНФ можно использовать один из двух методов:

I метод: приводим формулу к ДНФ; если какая-то элементарная конъюнкция не содержит некоторой переменной y, то добавляем её, используя закон расщепления: $x = (x \land y) \lor (x \land \overline{y})$; убираем одинаковые элементарные конъюнкции, используя закон идемпотентности $x \lor x = x$.

II метод: для данной формулы строим таблицу истинности, потом применяем правило, основанное на теореме Шеннона: СДНФ функции $f(x_1, \ldots, x_n)$ содержит столько элементарных конъюнкций, сколько единиц в столбце значений f; каждому единичному набору нулей и единиц $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ соответствует элементарная конъюнкция всех переменных, в которой x_i взято с отрицанием, если $\sigma_i = 0$ и без отрицания, если $\sigma_i = 1$.

П р и м е р 2.3.6 Для функции $\varphi = x \vee yz \vee \bar{x}\,\bar{y}z$ найти СДНФ.

1 метод: данная формула уже находится в форме ДНФ, поэтому добавляем в элементарные конъюнкции недостающие переменные, используя закон расщепления:

$\varphi = x \vee yz \vee \bar{x}\,\bar{y}z = xy \vee x\bar{y} \vee xyz \vee \bar{x}yz \vee \bar{x}\,\bar{y}z = \underline{xyz} \vee xy\bar{z} \vee x\bar{y}z \vee x\bar{y}\,\bar{z} \vee$ $\underline{xyz} \vee \bar{x}yz \vee \bar{x}\,\bar{y}z =$(убираем одинаковые элементарные конъюнкции, используя закон идемпотентности)$= xyz \vee xy\bar{z} \vee x\bar{y}z \vee x\bar{y}\,\bar{z} \vee \bar{x}yz \vee \bar{x}\,\bar{y}z$ - СДНФ;

2 метод: построим таблицу истинности:

Т а б л и ц а 2.3.1

x	y	z	$\bar{x}$	$\bar{y}$	$y \wedge z$	$\bar{x} \wedge \bar{y}$	$\bar{x}\,\bar{y}z$	$x \vee yz$	φ
0	0	0	1	1	0	1	0	0	0
0	0	1	1	1	0	1	1	0	1
0	1	0	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0	1	1
1	0	0	0	1	0	0	0	1	1
1	0	1	0	1	0	0	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	1	0	0	1	0	0	1	1

Выпишем единичные наборы функции: $1 = \varphi(0,0,1) = \varphi(0,1,1) =$ $= \varphi(1,0,0) = \varphi(1,0,1) = \varphi(1,1,0) = \varphi(1,1,1)$. Тогда по выше приведённому правилу, СДНФ функции имеет вид $\varphi = \bar{x}\,\bar{y}z \vee \bar{x}yz \vee x\bar{y}\,\bar{z} \vee x\bar{y}z \vee xy\bar{z} \vee xyz$.

Приведение формулы к СКНФ аналогично приведению к СДНФ. Также существует два метода:

а) метод эквивалентных преобразований;

б) СКНФ находят по таблице истинности: СКНФ функции $f(x_1, \ldots, x_n)$ содержит столько элементарных дизъюнкций, сколько нулей в столбце значений f; каждому нулевому набору нулей и единиц $(\sigma_1, \sigma_2, \ldots, \sigma_n)$

соответствует элементарная дизъюнкция всех переменных, в которой x_i взято с отрицанием, если $\sigma_i = 1$ и без отрицания, если $\sigma_i = 0$.

Пример 2.3.7 Рассмотрим функцию из предыдущего примера: $\varphi = x \vee yz \vee \overline{x}\,\overline{y}z$. Приведём её к СКНФ двумя способами:

а) $\varphi = \overline{\varphi} = \overline{x \vee yz \vee \overline{x}\,\overline{y}z} = \overline{\overline{x} \wedge \overline{\overline{y}\,\overline{z}} \wedge \overline{\overline{x}\,\overline{y}z}} = \overline{\overline{x} \wedge (\overline{y} \vee \overline{z}) \wedge (x \vee y \vee \overline{z})} =$

$=\overline{(\overline{y} \vee \overline{z})(\overline{x}y \vee \overline{x}\ \overline{z})} = \overline{\overline{x}y\overline{y} \vee \overline{x}\,\overline{z}\,\overline{y} \vee \overline{z}\,\overline{x}y \vee \overline{z}\,\overline{x}} = \overline{\overline{x}\,\overline{z}\,\overline{y} \vee \overline{z}\,\overline{x}y \vee \overline{z}\,\overline{x}} =$

$= \overline{\overline{x}\,\overline{z}\,\overline{y} \vee \overline{z}\,\overline{x}y \vee \overline{z}\,\overline{x}(y \vee \overline{y})} = \overline{\overline{x}\,\overline{z}\,\overline{y} \vee \overline{z}\,\overline{x}y \vee \overline{z}\,\overline{x}y \vee \overline{z}\,\overline{x}\,\overline{y}} = \overline{\overline{x}\,\overline{z}\,\overline{y} \vee \overline{z}\,\overline{x}y} =$

$= \overline{\overline{x}\,\overline{z}\,\overline{y}} \wedge \overline{\overline{z}\,\overline{x}y} = (x \vee z \vee y) \wedge (z \vee x \vee \overline{y})$ - СКНФ;

б) из таблицы истинности найдем $0 = \varphi(0{,}0{,}0) = \varphi(0{,}1{,}0)$ - нулевые наборы: по выше приведённому правилу, $\varphi = (x \vee y \vee z) \wedge (x \vee \overline{y} \vee z)$ - СКНФ.

Минимизация булевых функций в классе ДНФ. Карты Карно

При решении практических задач часто возникает проблема минимизации логических формул, в смысле, например, найти формулу, содержащую наименьшее число переменных, или наименьшее число операций, или наименьшее количество подформул определённого вида и т.д. К настоящему времени наиболее изучена задача отыскания дизъюнктивных форм, минимальных по числу вхождений переменных. Под вхождением переменной понимается место, которое переменная занимает в формуле.

Определение. Минимальной ДНФ (МДНФ) называется ДНФ с наименьшим числом вхождений переменных.

Существует много способов отыскания МДНФ (метод Квайна, неопределённых коэффициентов, с помощью гиперкубов и т.д.). Остановимся на наиболее простом – с использованием карт (диаграмм) Карно.

Карта Карно – это таблица, каждая клетка (ячейка) которой соответствует некоторой элементарной конъюнкции всех переменных. Для функции n переменных $x_1, x_2, \ldots, x_n$ существует 2^n возможных комбинаций их значений, состоящих из 0 и 1. То есть, например, для $n = 2$ имеем $2^2 = 4$ элементарные конъюнкции $xy, x\overline{y}, \overline{x}y, \overline{x}\,\overline{y}$, которым соответствуют следующие наборы 0 и 1: (1,1), (1,0), (0,1), (0,0); для $n = 3$ - $2^3 = 8$ - $xyz, xy\overline{z}, \ldots, \overline{x}\,\overline{y}\,\overline{z}$ - (1,1,1), (1,1,0),…, (0,0,0) и т.д.

Карты Карно строятся в виде таблицы размером $2^{n-k} \times 2^k$ так, что её столбцы соответствуют значениям переменных $x_1, x_2, \ldots, x_k$, строки -

$x_{k+1}, x_{k+2}, \dots, x_n$ (или наоборот); для одной и той же функции может быть построено несколько карт, важно, чтобы соседние ячейки (как по вертикали, так и по горизонтали) отличались только значением одной переменной.

Мы будем рассматривать в основном функции двух, трёх и четырёх переменных. Для них карты Карно имеют следующий вид:

а) для функции двух переменных x, y - рисунок 2.3.1;

б) для функции трёх переменных x, y, z - рисунок 2.3.2;

в) для функции четырёх переменных x, y, z, t - рисунок 2.3.3.

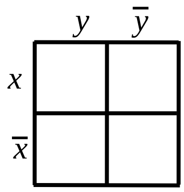


Рисунок 2.3.1

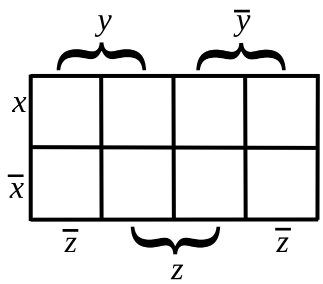


Рисунок 2.3.2

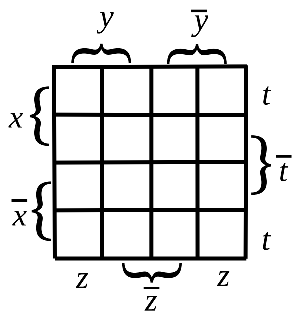


Рисунок 2.3.3

Для определения МДНФ булевой функции, сначала надо найти её СДНФ, затем каждую элементарную конъюнкцию СДНФ отметить единицей в соответствующей ячейке карты Карно.

П р и м е р 2.3.8 Функции $\varphi_1 = xy \vee x\bar{y}$ и $\varphi_2 = xy\bar{z} \vee x\bar{y}z \vee \bar{x}y\bar{z}$ заданы в форме СДНФ. Карта Карно для φ_1 на рисунке 2.3.4; для φ_2 - на рисунке 2.3.5.

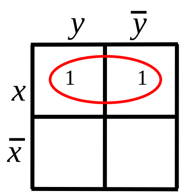


Рисунок 2.3.4

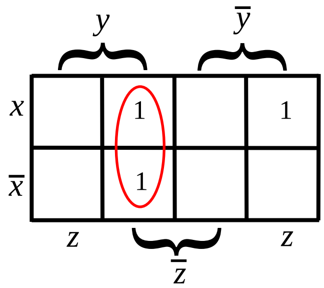


Рисунок 2.3.5

Заметим, что, если в картах Карно две, четыре, восемь (для функции четырёх переменных) соседних ячеек по вертикали или по горизонтали содержат 1, то эти ячейки объединяют в блоки (на картах их отмечают овалами) и соответствующие этим блокам дизъюнкции элементарных конъюнкций можно упростить. Так, в примере 2.3.8 для функции φ_1 имеем блок из двух ячеек, на рисунке он отмечен овалом. Этому блоку соответствует дизъюнкция $xy \vee x\bar{y}$, упрощая которую, получим: $xy \vee x\bar{y} = x \vee (y \wedge \bar{y}) = x \wedge 1 = x$. Таким образом, блоку из двух ячеек функции двух переменных отвечает одна переменная x, а именно та переменная, которая полностью «покрывает» этот блок. Формула упростилась $\varphi_1 = x$.

Для функции φ_2 также имеем один блок из двух ячеек, ему соответствует дизъюнкция элементарных конъюнкций $xy\bar{z} \vee \bar{x}y\bar{z}$, упрощая которую получим $y\bar{z}$, т.е. блоку из двух ячеек функции трёх переменных соответствует конъюнкция двух переменных, «покрывающих» этот блок. Формула упростилась $\varphi_2 = y\bar{z} \vee x\bar{y}z$.

Рассмотрим ещё несколько примеров.

П р и м е р 2.3.9 $\varphi = xyz \vee \bar{x}yz \vee x\bar{y}z \vee \bar{x}\,\bar{y}z$ - СДНФ функции. Её карта Карно на рисунке 2.3.6. Так как z находится на обоих концах карты, то её (карту) можно «скрутить» и считать, что 1 в углах карты образуют блок из четырёх ячеек. Эти четыре ячейки полностью «покрывает» переменная z, т.о., МДНФ функции будет $\varphi = z$.

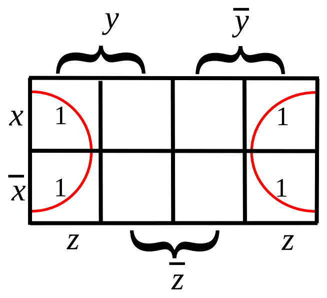


Рисунок 2.3.6

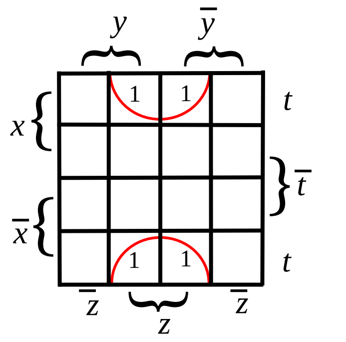


Рисунок 2.3.7

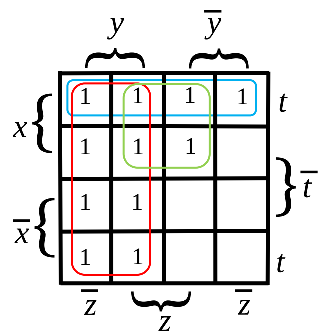


Рисунок 2.3.8

П р и м е р 2.3.10 - $\varphi = xyzt \vee x\bar{y}zt \vee \bar{x}yzt \vee \bar{x}\,\bar{y}zt$ - СДНФ функции. Её карта Карно на рисунке 2.3.7. На карте есть блок из четырёх ячеек, который покрывают переменные z и t, поэтому МДНФ функции будет: $\varphi = zt$.

П р и м е р 2.3.11 – Карта Карно для функции $\varphi = xyzt \vee xyt\bar{z} \vee xyz\bar{t} \vee$ $\vee\, xy\bar{z}\bar{t} \vee yz\bar{x}\bar{t} \vee y\bar{x}\,\bar{z}\bar{t} \vee yzt\bar{x} \vee \bar{x}y\bar{z}t \vee x\bar{y}\,\bar{z}t \vee xzt\bar{y} \vee xz\bar{y}\bar{t}$, заданной в СДНФ на рисунке 2.3.8. На карте имеем: блок из 8 ячеек покрывает переменная y; двум блокам из 4 ячеек соответствуют элементарные конъюнкции xt и xz, поэтому МДНФ будет: $\varphi = y \vee xt \vee xz$.

2.4 Булева алгебра и теория множеств. Коммутационные схемы

Можно легко заметить аналогию между свойствами операций над множествами и свойствами логических операций. Это не случайно.

Множество M вместе с заданными на нём операциями $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ называется алгеброй и обозначается $(M, \varphi_1, \varphi_2, \ldots, \varphi_n)$.

Определение. Всякая алгебра, содержащая две бинарные и одну унарную операции, которые удовлетворяют соотношениям 1) - 9) (см. основные эквивалентные соотношения булевой алгебры, или основные свойства операций над множествами) называется булевой.

Таким образом, булевыми алгебрами будут:

а) $(P_2,\ \wedge,\ \vee, \neg)$ - булева алгебра всех логических функций с операциями конъюнкции, дизъюнкции, отрицания;

б) $(P_2(m),\ \wedge,\ \vee,\ \neg)$ - булева алгебра логических функций m переменных – это подалгебра алгебры $(P_2,\ \wedge,\ \vee,\ \neg)$, т.к. $P_2(m) \subset P_2$;

в) $(\mathscr{P}(U),\ \cap,\ \cup,\ \neg)$ - булева алгебра множеств над U - универсумом, с операциями пересечения, объединения, дополнения;

г) $(B_n,\ \vee,\ \wedge,\ \neg)$ - булева алгебра двоичных векторов длины n с покомпонентными логическими операциями над двоичными векторами, определёнными следующим образом:

$\forall\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n), \beta = (\beta_1, \beta_2, \dots, \beta_n)$ имеет место:

1) $\alpha \wedge \beta = (\alpha_1 \wedge \beta_1, \alpha_2 \wedge \beta_2, \dots, \alpha_n \wedge \beta_n)$, где $\alpha_i \wedge \beta_i = 1$, если $\alpha_i = \beta_i = 1$; в любом другом случае $\alpha_i \wedge \beta_i = 0$.

2) $\alpha \vee \beta = (\alpha_1 \vee \beta_1, \alpha_2 \vee \beta_2, \dots, \alpha_n \vee \beta_n)$, где $\alpha_i \vee \beta_i = 0$, если $\alpha_i = 0, \beta_i = 0$; в любом другом случае $\alpha_i \vee \beta_i = 1$.

3) $\overline{\alpha} = (\overline{\alpha_i}, \dots, \overline{\alpha_n})$, где $\overline{\alpha_i} = 0$, если $\alpha_i = 1$, $\overline{\alpha_i} = 1$, если $\alpha_i = 0$.

Если мощности множеств $\mathscr{P}(U)$, B_n и $P_2(m)$ равны, то между ними можно установить взаимно однозначное соответствие, а соответствующие булевы алгебры будут изоморфны.

Изоморфизм булевых алгебр широко используется в компъютерных вычислениях, например, вместо выполнения операций над множествами или логическими функциями используют их изоморфные аналоги – поразрядные операции над двоичными векторами.

Коммутационные схемы

Возможность применения математической логики к техническим вопросам была обнаружена в 30-х годах XX века. Была замечена, например, связь между электрическими цепями и логическими функциями. Это открытие дало толчок к развитию ЭВМ. Рассмотрим упрощённо эту связь.

Основным элементом релейно-контактных устройств является электромеханическое реле (переключатель p). Реле может размыкать и замыкать цепь. Присвоим p значение 1, когда цепь замкнута (ток проходит), и значение 0, когда цепь разомкнута (ток не проходит).

Рассмотрим электрическую цепь на рисунке 2.4.1. При таком расположении контактов p и q лампочка будет гореть (т.е. схема имеет значение1), если оба переключателя p и q замкнуты (т.е. имеют значения 1). Таким образом, эта схема соответствует логической формуле $p \wedge q$, а такое расположение переключателей называется логическим элементом «p и q» или схемой логического умножения, его часто обозначают на схеме как на рисунке 2.4.2 .

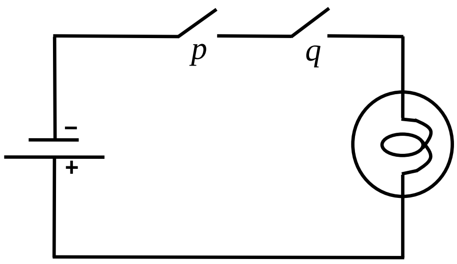


Рисунок 2.4.1

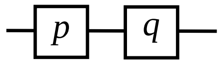


Рисунок 2.4.2

Рассмотрим теперь схему на рисунке 2.4.3. В этой цепи лампочка будет гореть, и значение схемы равно 1, если хотя бы один из двух контактов p или q, или оба, будут замкнуты, т.е. или $p = 1$, или $q = 1$, или оба $p = q = 1$. Таким образом, эта схема соответствует логической формуле $p \vee q$, а такое расположение переключателей называется логическим элементом «p или q» или схемой логического сложения. Этот элемент можно изображать на схемах, как на рисунке 2.4.4.

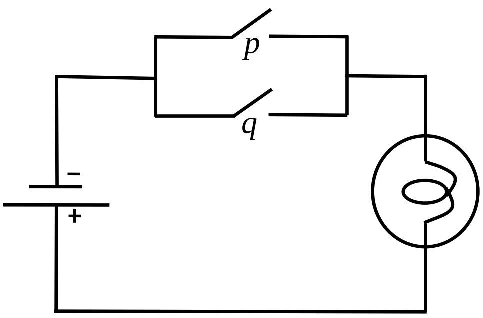


Рисунок 2.4.3

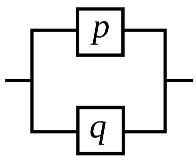


Рисунок 2.4.4

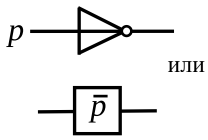


Рисунок 2.4.5

Если имеем схему с одним переключателем p, который обладает свойством, что лампочка загорается тогда и только тогда, когда p разомкнут (т.е. схема имеет значение 1, когда $p = 0$, и значение 0, когда $p = 1$), то эта схема соответствует $\overline{p}$. Такой логический элемент называется «не p» или инвертором, его часто изображают на схемах, как на рисунке 2.4.5.

Рассмотрим примеры, реализующих простейшие логические формулы.

П р и м е р 2.4.1 Схема на рисунке 2.4.6 реализует формулу (переключательную функцию, или функцию проводимости) $(x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_3)$; схема для формулы $(\overline{x_1 \vee x_2}) \vee (x_1 \wedge x_2)$ изображена на рисунке 2.4.7; схема на рисунке 2.4.8 для формулы $(x_1 \vee (\overline{x_2} \wedge x_3)) \vee (\overline{x_1} \wedge x_2)$.

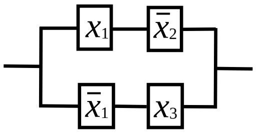


Рисунок 2.4.6

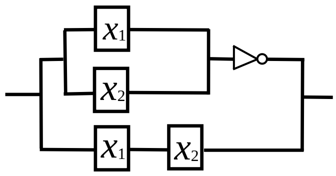


Рисунок 2.4.7

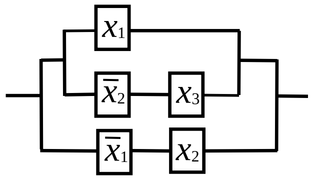


Рисунок 2.4.8

Так как любую логическую формулу можно привести к ДНФ или КНФ, то для неё всегда можно построить контактную схему. Очевидно, что чем проще формула, представляющая функцию проводимости, тем проще схема. Поэтому задача упрощения схемы сводится к задаче упрощения или минимизации соответствующих функций. Эту задачу мы решали выше.

П р и м е р 2.4.2 Упростим схему на рисунке 2.4.9.

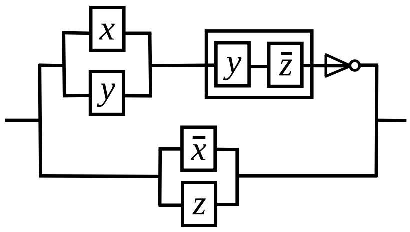


Рисунок 2.4.9

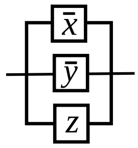


Рисунок 2.4.10

Решение: составим переключательную функцию

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

Упростим эту функцию, используя эквивалентные преобразования:

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

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

Последней формуле соответствует упрощённая схема на рисунке 2.4.10.

3 Элементы теории графов

3.1 Основные понятия и определения

Графы – это один из способов наглядного представления взаимосвязей между различными объектами. На языке теории графов формулируются и решаются многие задачи сетевого планирования и управления – сети железных дорог, телефонные и компьютерные сети, ирригационные системы; эта теория позволяет формализовать, анализировать и решать задачи экономической и производственной практики, производить анализ и синтез функциональных блоков компьютеров, комплексов программ и т.д.

Графы описывают связи между объектами, а связи могут быть «направленными» (например, в генеалогическом дереве) или «ненаправленными» (например, сеть дорог с двусторонним движением). Поэтому в теории графов выделяют два основных типа графов: ориентированные (орграф) и неориентированные (неорграф, н-граф). Иногда рассматривают графы смешанного типа.

Определение. Графом G называется совокупность двух множеств: V – вершин и E – рёбер (для н-графа) или дуг (для орграфа). Обозначается $G = G(V, E)$ или $G = (V, E)$.

Пусть $V = \{v_1, v_2, \dots, v_n\}$ - множество вершин. Тогда для н-графа $E = \{e_1, e_2, \dots, e_m\}$, где ребра $e_k = \{v_i, v_j\}$ – двухэлементные подмножества множества V; для орграфа $E = \{e_1, e_2, \dots, e_m\}$, где дуги $e_k = (v_i, v_j)$ – упорядоченные пары, т.е. $E \subseteq V \times V$.

Если $e = \{u, v\}$ – ребро ($e = (u, v)$ - дуга), то вершины u и v называются концами ребра (u – началом, v – концом дуги); вершины u и v называются инцидентными ребру (дуге) e, а ребро (дуга) e инцидентным (ой) вершинам u и v; вершины u и v называются смежными, если они являются концами одного ребра (дуги).

Ребро (дуга), концевые вершины которого совпадают, называется петлёй.

Рёбра (дуги) инцидентные одной и той же паре вершин, называются параллельными или кратными.

Граф, содержащий кратные рёбра (дуги), называется мультиграфом.

Граф называется конечным, если множетва V и E конечны, пустым – если множество его вершин V (а значит и рёбер) пусто.

Некоторые вершины могут не войти в список пар множества E, они называются изолированными.

Граф, у которого все вершины изолированные, называется нуль-графом; антиподом нуль-графа является полный граф: его рёбрами являются все возможные пары его вершин (у него нет петель и кратных рёбер).

Заметим, что многие теоремы и определения в теории графов могут быть отнесены как к н-графам, так и к орграфам. Часто, когда совершенно ясно, о каком типе графов идёт речь, рёбра, как и дуги, обозначают круглыми скобками, т.е. вместо $e=\{u,v\}$ записывают $e=(u,v)$. Каждому н-графу канонически соответствует орграф с тем же множеством вершин, в котором каждое ребро заменено двумя дугами, инцидентными тем же вершинам и имеющими противоположные направления.

Определение. Степенью (иногда валентностью) вершины v (обозначается $\rho(v)$ или $d(v)$)) называется количество всех инцидентных ей рёбер.

Если m – число рёбер н-графа G, то $\sum_{v\in V}\rho(v)=2m$. Предполагается, что петля даёт вклад 2 в степень вершины. Для вершин орграфа определяются также две локальные степени.

Определение. Степенью выхода вершины v (обозначается $\rho^{+}(v)$, $\rho_1(v)$) называется число дуг с началом в v; степенью входа вершины v (обозначается $\rho^{-}(v)$, $\rho_2(v)$) – число дуг с концом в вершине v; $\rho(v)=\rho_1(v)+\rho_2(v)$. Если m– число дуг орграфа G, то $\sum_{v\in V}\rho_1(v)=\sum_{v\in V}\rho_2(v)=m$, причём петля даёт вклад 1 в обе степени.

Способы задания графов

1. *Задание графа парой множеств* V *и* E, когда каждое ребро (дуга) $e\in$ E , определено парой инцидентных ему вершин.

П р и м е р 3.1.1 $G=G(V,E)$ – граф , где $V=\{a,b,c\}$:

а) $E=\{\{a,b\},\{b,c\}\}$ – для н-графа;

б) $E=\{(a,b),(b,c)\}$ – для орграфа.

2. *Задание графа рисунком*. Вершины обозначаются точками, рёбра – линиями, соединяющими точки; дуги – стрелками. При этом один и тот же граф может быть задан разными рисунками. Например, граф $G=G(V,E)$ из примера 3.1.1, где $V=\{a,b,c\}$, $E=\{\{a,b\},\{b,c\}\}$ можно изобразить так:

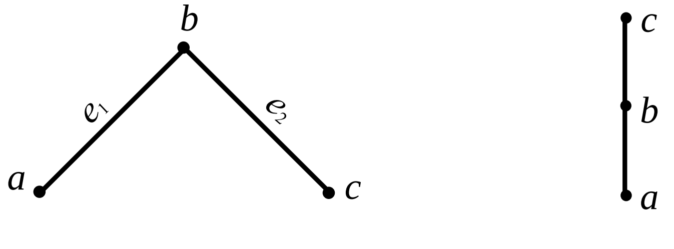


Рисунок 3.1.1

3. *Задание графа матрицей инцидентности.*

Пусть все вершины и рёбра (дуги) графа $G = (V, E)$ пронумерованы: $V = \{v_1, v_2, \dots, v_n\}$, $E = \{e_1, e_2, \dots, e_m\}$. Матрицей инцидентности графа G называется матрица $A = (a_{ij})$ размера $n \times m$, n – число вершин, m – число рёбер (дуг):

а) для н-графа: $a_{ij} = \begin{cases} 1, \text{если вершина } v_i \text{ инцидентна ребру } e_j, \\ 0, \text{в противном случае;} \end{cases}$

б) для орграфа: $a_{ij} = \begin{cases} 1, \text{если из вершины } v_i \text{ дуга } e_j \text{ выходит}, \\ -1, \text{если в вершину } v_i \text{ дуга } e_j \text{ входит}, \\ 2, \text{если } e_j \text{ петля}, \\ 0, \text{в противном случае.} \end{cases}$

Пример 3.1.2 $V = \{v_1, v_2, v_3, v_4\}$ – множество вершин, $E = \{e_1, e_2, e_3, e_4, e_5\}$ – множество рёбер, $A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$ - матрица инцидентности графа на рис. 3.1.2; $V = \{a_1, a_2, a_3\}$ – множество вершин, $E = \{1,2,3,4,5,6\}$ – множество дуг, матрица инцидентности графа изображена на рисунке 3.1.3.

$$A = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & -1 \end{pmatrix}$$

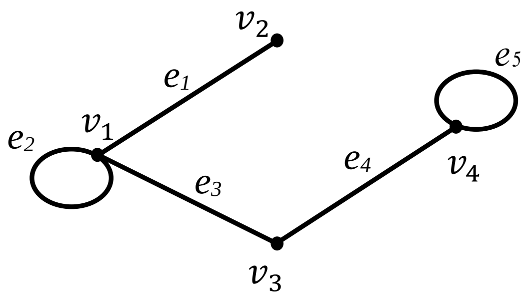


Рисунок 3.1.2

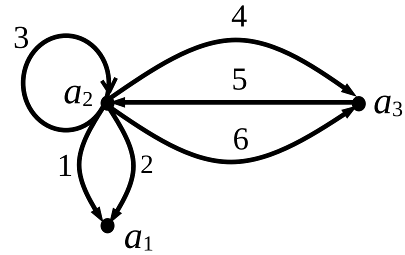


Рисунок 3.1.3

4. *Задание графа матрицей смежности.*

Матрицей смежности $B = (b_{ij})$ графа $G = (V, E)$ называется квадратная матрица размера $n \times n$ (n – число вершин), элементы которой определены следующим образом:

а) для н-графа $b_{ij} = \begin{cases} 1, \text{если } i - \text{ая и } j - \text{ая вершины смежные}, \\ 0, \text{в противном случае;} \end{cases}$

б) для орграфа

$b_{ij} = \begin{cases} 1, \text{если есть дуга с началом в } i - \text{ой вершине и концом в } j - \text{ой}, \\ 0, \text{в противном случае;} \end{cases}$

в) для мультиграфа

$$b_{ij} = \begin{cases} k, \text{если } k \text{ число рёбер, соединяющих } i - \text{ую и } j - \text{ую вершины} \\ \text{(для н} - \text{графа),} \\ \text{или } k \text{ число дуг с началом в } i - \text{ой и концом в } j - \text{ой вершинах,} \\ 0, \text{в противном случае.} \end{cases}$$

П р и м е р 3.1.3 - Матрица

$$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

является матрицей смежности н-графа, изображённого на рисунке 3.1.4; матрица

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- матрица смежности для орграфа на рисунке 3.1.5;

$$B = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

- матрица смежности для мультиграфа на рисунке 3.1.6.

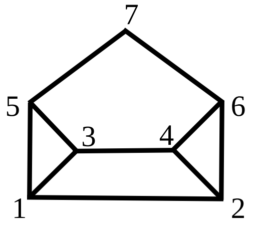


Рисунок 3.1.4

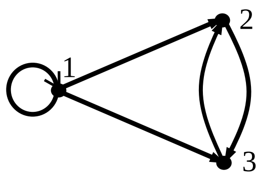


Рисунок 3.1.5

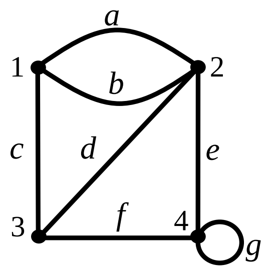


Рисунок 3.1.6

Заметим, что для н-графа матрица смежности симметрическая, т.е. $B = B^T$ причём, если у графа нет петель, то в матрице смежности по главной диагонали стоят нули; для орграфа количество единиц в i-ой строке равно степени выхода $\rho_1(v_i)$ i-ой вершины; количество единиц в j-ом столбце равно $\rho_2(v_i)$ – степени входа j-ой вершины.

5. *Задание графа списком рёбер (дуг).*

Кроме матрицы инцидентности, отношение инцидентности можно определить списком рёбер (дуг), задавая его в виде двух столбцов, в первом перечислены все рёбра (дуги), во втором – инцидентные им вершины. Для орграфа, если число дуг достаточно мало по сравнению с числом вершин, список дуг можно оформить по-другому: задать два набора – $g = (g_1, g_2, \dots, g_m)$ и $h = (h_1, h_2, \dots, h_m)$, где (g_i, h_i) - i-ая дуга графа.

Список дуг орграфа на рисунке 3.1.5 может быть представлен наборами $g = (1,1,1,2,3)$ и $h = (1,2,3,3,2)$.

Пример 3.1.4 - Для графа на рисунке 3.1.6 список рёбер задан таблицей 3.1.1.

Таблица 3.1.1

Ребра	Вершины
a	1 2
b	1 2
c	1 3
d	2 3
e	2 4
f	3 4
g	4 4

Существуют и другие способы задания графов (см., например, [3], с.244), мы ограничимся этими.

Изоморфизм графов

Один и тот же граф, как указывалось выше, можно изображать по разному. Например, на рисунке 3.1.7 изображён один и тот же граф.

Рисунок 3.1.7

Вид матриц и списка рёбер зависит от нумерации вершин и рёбер. Графы, отличающиеся друг от друга только нумерацией вершин и рёбер, называют изоморфными. Приведём точное определение.

Определение. Два графа $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ изоморфны (обозначается $G_1 \sim G_2$), если существует биекция φ: $V_1 \to V_2$, сохраняющая смежность: $e_1 = (u, v) \in \mathrm{E}_1 \Leftrightarrow e_2 = (\varphi(u), \varphi(v)) \in \mathrm{E}_2$.

Теорема. Изоморфизм графов есть отношение эквивалент ности.

Действительно, изоморфизм обладает всеми свойствами отношения эквивалентности:

а) рефлексивностью $(G_1 \sim G_2)$;

б) симметричностью $(G_1 \sim G_2 \Rightarrow G_2 \sim G_1)$;

в) транзитивностью $(G_1 \sim G_2, G_2 \sim G_3 \Rightarrow G_1 \sim G_3)$.

Графы рассматриваются с точностью до изоморфизма. Для того чтобы граф G_1 был изоморфен графу G_2 , необходимо и достаточно существование какого-либо взаимно однозначного соответствия между вершинами графов, а также между их рёбрами. Иногда в простых случаях в изоморфизме графов можно убедиться по их графическому представлению.

Рассмотрим более общее правило установления изоморфизма графов:

а) подсчитываем число вершин графов (если оно разное, то графы не изоморфны);

б) выписываем все вершины обоих графов с их степенями (для н-графов) или с парой степеней выхода и входа (для орграфа);

в) для каждой вершины первого графа ищем такую вершину второго графа, чтобы их степени или пары степеней совпадали, т.е. получим взаимно однозначное соответствие между вершинами графов (если соответствия нет, то графы не изоморфны);

г) проверяем сохранение смежности соответствующих вершин. Практически удобно вершины одного графа располагать над вершинами другого, а соответствующие вершины соединять линией.

Подграфы и части графов. Операции над графами

Определение. Граф $G_1 = (V_1, E_1)$ называется частью графа $G = (V, E)$, если $V_1 \subseteq V$ и $E_1 \subseteq E$; подграфом для графа $G = (V, E)$ называется его часть $\mathrm{G}_1 = (V_1, E_1)$, которой принадлежат все рёбра (дуги) с концами в V_1. На рисунке 3.1.8 изображены граф G, его подграф G_1 и его часть G_1 .

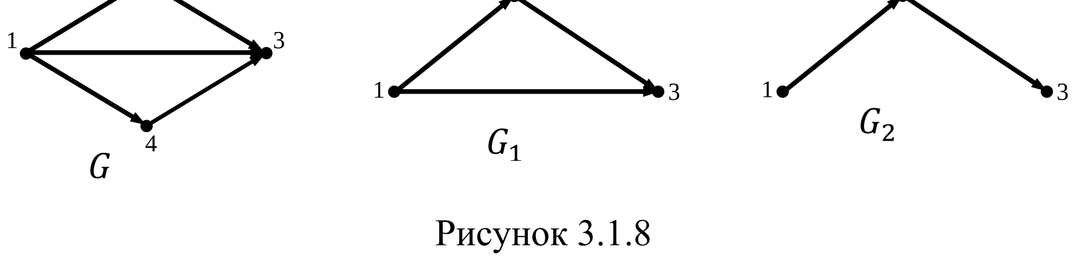


Рисунок 3.1.8

1. Пусть дан граф $G(V,E)$.

а) граф $\overline{G}(V,E)$ называется дополнением графа $G(V,E)$, если для любых $u,v \in V$, ребро $\{u,v\}$ графа $\overline{G}$ существует тогда и только тогда, когда это ребро в G отсутствует;

б) операции добавления к графу G вершины v или ребра $\{u,v\}$ состоит в образовании графов $G'(V \cup \{v\}, E)$ или $G''(V \cup \{u,v\},\ E \cup \{\{u,v\}\})$;

в) операции удаления вершины v из G состоит в удалении вершины вместе с инцидентными ей рёбрами, т.е. образования графа

$G'(V \setminus \{v\},\ E \setminus \{\{v_1,v_2\} \mid = v\})$;

г) в результате операции удаления ребра $\{u,v\}$ получается граф

$G'(V, E \setminus \{\{u,v\}\})$;

д) операция отождествления вершин u и v состоит в удалении из графа этих вершин и добавлении новой вершины v', при этом рёбра, имеющие концами удалённые вершины, заменяются рёбрами, в которых эти концы заменены новой вершиной. В случае, когда u и v соединены ребром, операцию отождествления вершин u и v называют стягиванием ребра $\{u,v\}$.

2. Пусть даны два графа $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$:

а) объединением графов G_1 и G_2 (обозначается $G_1 \cup G_2$) называется граф

$$G = G_1 \cup G_2 = (\,V_1 \cup V_2, E_1 \cup E_2);$$

б) пересечением графов G_1 и G_2 (обозначается $G_1 \cap G_2$) называется граф

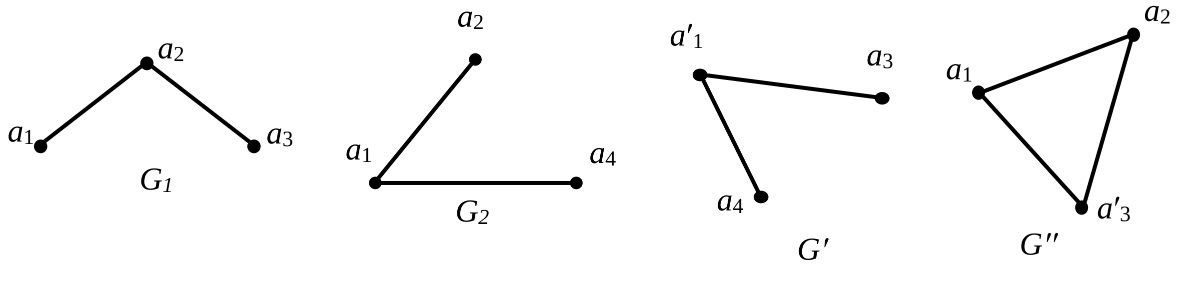
$$G = G_1 \cap G_2 = (\,V_1 \cap V_2, E_1 \cap E_2);$$

в) соединением графов G_1 и G_2 (обозначается $G_1 + G_2$) называется граф

$$G_1 + G_2 = (\,V_1 \cup V_2, E_1 \cup E_2 \cup \ \{\{u,v\} \mid\ u \in V_1\,, v \in V_2, u\ \neq v\}).$$

Кроме этих операций, для графов можно определить также и другие операции: кольцевая сумма, прямое произведение и т.д.

П р и м е р 3.1.5 - На рисунке 3.1.9 изображены графы, представляющие результаты операций над графами G_1, G_2 и $G_1 \cup G_2$. Графы G' и G'' получены из графа $G_1 \cup G_2$ соответственно стягиванием ребра $\{a_1,a_2\}$ и отождествлением вершин a_3 и a_4.



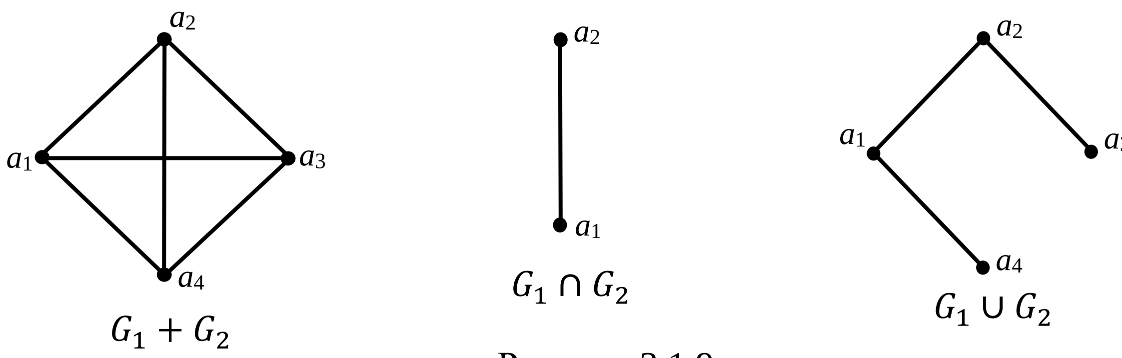


Рисунок 3.1.9

Рассмотрим ещё несколько важных видов графов:

а) полный граф с n вершинами обозначается K_n (н-граф без петель и кратных рёбер называется полным, если любые его две вершины соединены ребром). На рисунке 3.1.10 изображены полные графы.

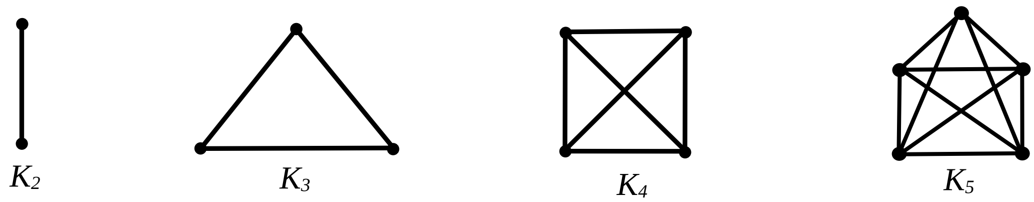


Рисунок 3.1.10

Число рёбер полного графа вычисляется по формуле:

$$m(K_n) = C_n^2 = \frac{n(n-1)}{2};$$

б) n-мерные кубы (n-кубы), обозначаются Q_n. Вершинами Q_n являются всевозможные n-ки, состоящие из 0 и 1. Так как всего таких наборов 2^n, то вершин у Q_n будет 2^n. Рёбра задаются по следующему правилу: вершины смежны тогда и только тогда, когда соответствующие n-ки отличаются ровно одной координатой. На рисунке 3.1.11 изображены кубы $Q_1 = K_2$, Q_2 и Q_3.

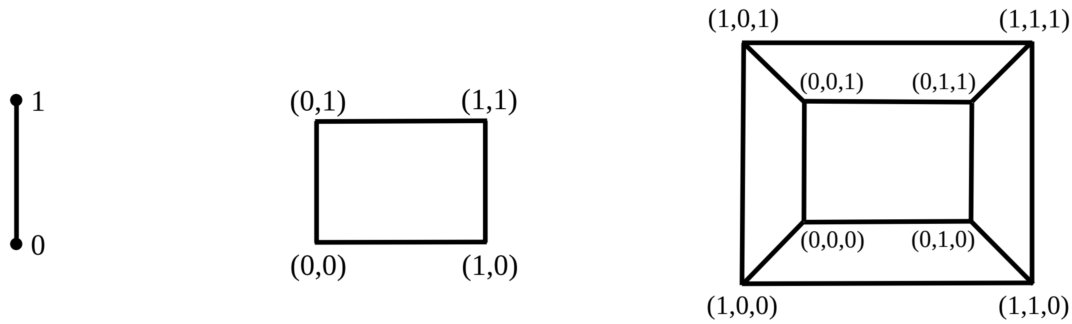


Рисунок 3.1.11

в) граф $G(V,E)$ называется двудольным, если существует такое разбиение множества V его вершин на два подмножества V_1 и V_2 (доли), что концы каждого ребра принадлежат разным подмножествам. Полный двудольный граф обозначают $K_{m,n}$, где m – число вершин в V_1, а n – число вершин в V_2 и для каждого $v \in V_1$, $u \in V_2$ имеем $\{v, u\} \in E$. На рисунке 3.1.12 изображены полные двудольные графы $K_{2,3}$ и $K_{3,3}$.

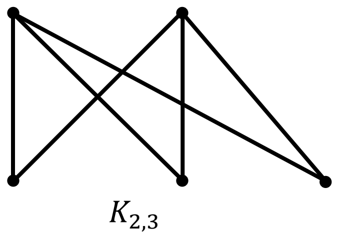


Рисунок 3.1.12

П р и м е р 3.1.6 - Рассмотрим различные графы, иллюстрирующие выше приведённые определения и понятия (см. рисунок 3.1.13):

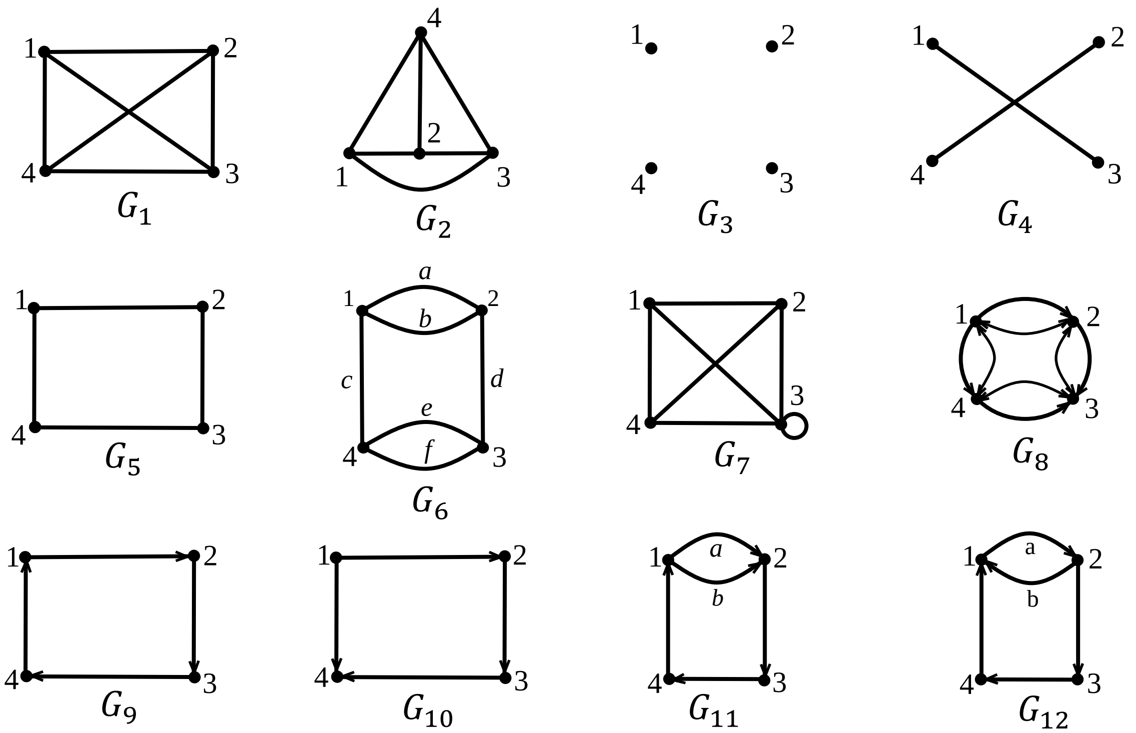


Рисунок 3.1.13

а) G_1- G_7 – н-графы, G_8 - G_{12} – орграфы;

б) G_1 и G_2 – полные графы, причём $G_1 = G_2$;

в) G_7 – не является полным, т.к. имеет одну петлю;

г) G_3 – нуль - граф, т.к. все его вершины изолированные;

д) G_4 и G_5 являются дополнением друг к другу: $G_4 = \overline{G_5}$ и $G_5 = \overline{G_4}$;

е) G_6 – мультиграф, т.к. содержит кратные рёбра a и b, e и f;

ж) G_8 – орграф, канонически соответствующий н-графу G_5;

и) G_9 и G_{10} не являются равными, т.к. имеют отличающиеся дуги: в G_9 – (4,1), в G_{10} – (1,4);

к) G_{11} – ориентированный мультиграф, дуги a и b кратные;

л) G_{12} – не является мультиграфом, т.к. дуги a и b различно ориентированы.

П р и м е р 3.1.7 - Найти степени вершин графов G_1 и G_2 , изображённых на рисунке 3.1.14.

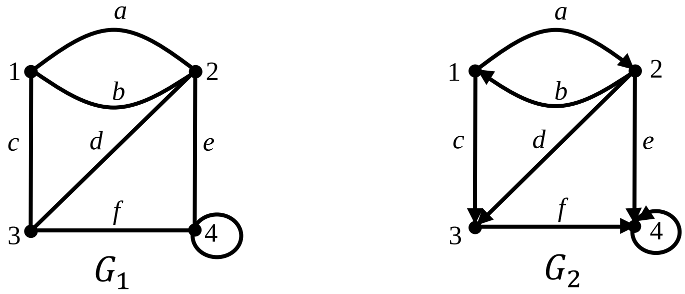


Рисунок 3.1.14

Решение: оба графа имеют по четыре вершины, т.е. $V = \{1,2,3,4\}$ – множество вершин и G_1 и G_2 .

а) степени вершин н-графа G_1 : $\rho(1) = 3,\ \rho(2) = 4,\ \rho(3) = 3,\ \rho(4) = 4.$ Итак, $\sum_{V\in G}\rho(V) = 14 = 2m,\quad m = 7$ – число рёбер;

б) степени вершин орграфа G_2 : степени выхода $\rho_1(1) = 2,\ \rho_1(2) = 3,$ $\rho_1(3) = 1, \rho_1(4) = 1$; степени входа $\rho_2(1) = 1,\ \rho_2(2) = 1,\ \rho_2(3) = 2,\ \rho_2(4) = 3.$ Таким образом, $\sum_{V\in G_2}\rho_2(V) = \sum_{V\in G_2}\rho_2(V) = 7 = m.$

3.2 Маршруты, достижимость, связность

Введём ещё несколько новых понятий одновременно для н-графа и орграфа. Пусть $G(V,E)$ граф. Последовательность $v_1, e_1, v_2, e_2, \dots, e_n, v_{n+1}$, где $v_1, v_2, \dots, v_{n+1} \in V$, $e_1, e_2, \dots, e_n \in E$, называется маршрутом, соединящим вершины v_1 и v_{n+1}. Маршрут можно задать также последовательностью его вершин $v_1, v_2, \dots, v_{n+1}$ или последовательностью его рёбер $e_1, e_2, \dots, e_n$. Вершина v_1 - начало маршрута, v_{n+1} – конец.

Цепь (путь) – маршрут, в котором все рёбра (дуги) различны.

Цикл (контур) – замкнутая цепь (путь), т.е. $v_1 = v_{n+1}$.

Простая цепь (путь), простой цикл (контур) - цепь (путь), цикл (контур) не пересекающие себя, т.е. не содержащие повторяющихся вершин, кроме, может быть, первой и последней.

Граф, не содержащий циклов (контуров), называется ациклическим (бесконтурным).

Число рёбер (дуг) маршрута называется его длиной. Если существует цепь (путь) длины n, соединяющая вершины u и v , то будем писать $u \xrightarrow{n} v.$

Вершина v достижима из вершины u , если существует цепь (путь) с началом в u и концом в v .

Определение. н-граф называется связным, если любые его две несовпадающие вершины соединены маршрутом; орграф называется связным, если соответствующий ему н-граф является связным; орграф называется сильно связным, если любые его две вершины u и v взаимно достижимы, т.е. u достижима из v и v достижима из u .

Определение. Подграф G_1 графа G называется компонентой связности (сильной компонентой связности) графа G (или связной (сильно связной) компонентой), если:

1) G_1 – непустой связный (сильно связный) граф;

2) если G_2 – связный (сильно связный) подграф графа G и $G_1 \subseteq G_2$, то $G_1 = G_2$, т.е. G_1 – максимальный связный (сильно связный) подграф графа G.

Пример 3.2.1 Для н-графа G на рисунке 3.2.1 можно определить следующие понятия:

а) (v_1, v_2, v_3, v_4) – простая цепь;

б) $(v_1, v_3, v_2, v_1, v_3, v_4)$ – маршрут;

в) (v_3, v_1, v_2, v_4) не маршрут (т.к. нет ребра $\{v_2, v_4\}$);

г) (v_1, v_2, v_3, v_1) – простой цикл;

д) вершины v_1, v_2, v_3, v_4 – попарно достижимы; v_5, v_6 – цепь длиной 1, v_7 – изолированная вершина или цепь длиной 0;

е) граф G не является связным, он имеет три компоненты связности с множеством вершин $\{v_1, v_2, v_3, v_4\}$, $\{v_5, v_6\}$, $\{v_7\}$.

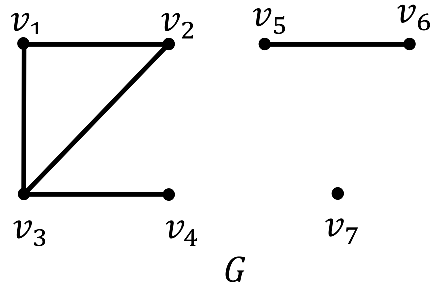


Рисунок 3.2.1

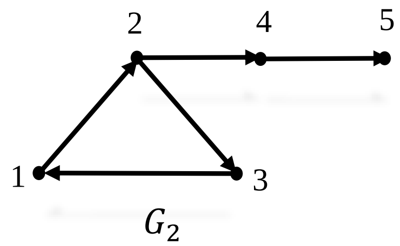


Рисунок 3.2.2

Пример 3.2.2 - Для орграфа G_2 на рисунке 3.2.2 имеет место:

а) 1, 2, 4, 5 или 1, 2, 3 – простой путь;

б) 1, 2, 3, 1 – простой контур;

в) вершина 5 достижима из любой вершины, из вершины 5 не достижима ни одна вершина; вершина 4 достижима из 1, 2, 3 и не достижима из 5;

г) этот граф связный, но не сильно связный. Он имеет три сильные компоненты связности, определяемые множеством вершин $\{1, 2, 3\}$, $\{4\}$, $\{5\}$.

Теорема. Любой граф может быть представлен в виде объединения непересекающихся связных (сильно связных) компонент. Разложение графа на связные (сильно связные) компоненты однозначно.

Множество вершин графа разбивается на подмножества вершин связных (сильно связных) компонент. Таким образом, множество вершин связных (сильно связных) компонент образуют разбиение множества вершин графа.

Число связных (сильно связных) компонент графа G (обозначается $K(G)$ или $C(G)$) определяется однозначно. Если $K(G) = 1$, то граф G связный, если $K(G) > 1$, то G – несвязный граф.

Для орграфа G_2 из примера 3.2.2 число сильно связных компонент $K(G_2) = 3$, а множество вершин $\{1,2,3,4,5\}$ имеет разбиение $\{\{1, 2, 3\},\{4\},\{5\}\}$.

Определение компонент связности и сильных компонент связности

Рассмотрим ещё одну матричную характеристику графа.

Определение. Квадратная матрица $C = (c_{ij})$ порядка n (n– число вершин) называется матрицей достижимости (для н-графа матрицей связности), если

$$c_{ij} = \begin{cases} 1, \text{если } j - \text{ая вершина достижима из } i - \text{ой}, \\ 0, \text{в противном случае}. \end{cases}$$

Матрицу достижимости можно получить следующим образом: пусть A – матрица смежности, составим матрицу $B = (b_{ij}) = E + A + A^2 + \cdots + A^m$, тогда $c_{ij} = \begin{cases} 1, \text{ если } b_{ij} \neq 0 \\ 0, \text{ если } b_{ij} = 0 \end{cases}$.

Заметим, что существуют другие способы определения матрицы достижимости. В простых случаях матрицу C можно составить по рисунку графа, например, для графа G_2 из примера 3.2.2 матрица достижимости будет иметь вид:

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Если все элементы матрицы связности н-графа $c_{ij} = 1$, то граф связный. В общем случае матрица C н-графа является матрицей отношения эквивалентности, соответствующей разбиению множества вершин графа на компоненты. Например, для графа G из примера 3.2.1 матрица связности будет:

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Блоки, состоящие из единиц, соответствуют 3 компонентам связности с множеством вершин $\{v_1, v_2, v_3, v_4\}$, $\{v_5, v_6\}$, $\{v_7\}$, т.е. разбиению множества вершин графа на эти компоненты.

Для орграфа с помощью матрицы достижимости можно найти сильные компоненты связности.

Рассмотрим, кроме матрицы $C = (c_{ij})$, ещё две матрицы Q и S. $Q = C^T = (q_{ij})$,

$$q_{ij} = \begin{cases} 1, \text{если } i - \text{ая вершина достижима из } j - \text{ой} \\ 0, \text{в противном случае.} \end{cases}.$$

Иногда матрицу Q называют матрицей контрдостижимости.

$S = (s_{ij}) = Q * C$, где операция $*$ означает поэлементное произведение матриц C и Q, т.е. $s_{ij} = c_{ij} \cdot q_{ij}$.

Так как вершины v_i и v_j находятся в одной сильно связной компоненте, если они взаимно достижимы, то очевидно, что соответствующий элемент матрицы S равен 1: $s_{ij} = 1$. Итак, сильно связная компонента, содержащая вершину v_i, состоит из вершин v_j, для которых $s_{ij} = 1$.

Пример 3.2.3 - Рассмотрим орграф G_2 из примера 3.2.2. Его матрица достижимости была найдена: $C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$. $Q = C^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ - матрица контрдостижимости. Матрица $S = C * Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$.

В первой строке матрицы S три единицы, которые соответствуют вершинам 1, 2, 3, это значит, что сильная компонента связности, содержащая вершину 1, содержит также вершины 2 и 3, т.е. соответствует множеству вершин $\{1,2,3\}$. Мысленно вычёркиваем из матрицы S строки и столбцы, соответствующие вершинам 1, 2, 3, в оставшейся матрице $S_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ в первой строке одна единица, отвечающая вершине 4, следовательно, вторая сильная компонента связности отвечает множеству $\{4\}$. Вычёркиваем из S_1 строку и столбец, соответствующие вершине 4, в оставшейся матрице $S_1 = (1)$ всего одна единица, которая соответствует вершине 5, т.е. последняя компонента сильной связности будет отвечать множеству $\{5\}$. Таким образом, граф имеет три компоненты сильной связности с множествами вершин $\{1,2,3\}$, $\{4\}$, $\{5\}$.

Исследование маршрутов графа

Важным применением матрицы смежности является возможность по ней определять маршруты фиксированной длины и их количество.

Теорема. Пусть дан граф $G = (V, E)$, где $V = \{v_1, v_2, \dots, v_n\}$ – множество вершин, A – матрица смежности графа G. Рассмотрим матрицу:

$$A^k = A \cdot A \cdot \ldots \cdot A = (a_{ij}).$$

Если $a_{ij} = m$, то из вершины v_i в вершину v_j существует m маршрутов длины k.

Пример 3.2.4 Рассмотрим н-граф, заданный рисунком:

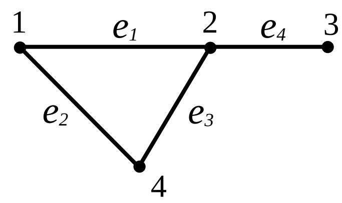


Рисунок 3.2.3

Матрица смежности этого графа: $A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$. По матрице A устанавливаем существование маршрутов длины 1, например, т.к. $a_{13} = 0$, то маршрутов из 1 вершины в 3 длины 1 нет (записывают: маршрута $\left(1 \xrightarrow{1} 3\right)$ нет).

$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}.$$

Т.к. $a_{13}^2 = 1$, то существует один маршрут $\left(1 \xrightarrow{2} 3\right)$, т.к. $a_{22}^2 = 3$, то существует три маршрута $\left(2 \xrightarrow{2} 2\right)$, т.к. $a_{34}^2 = 1$, то существует один маршрут $\left(3 \xrightarrow{2} 4\right)$ и т.д.

$$A^3 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 1 & 3 \\ 4 & 2 & 3 & 4 \\ 1 & 3 & 0 & 1 \\ 3 & 4 & 1 & 2 \end{pmatrix}.$$

Т.к. $a_{13}^3 = 1$, то существует один маршрут $\left(1 \xrightarrow{3} 3\right)$, т.к. $a_{21}^3 = 4$, то существует четыре маршрута $\left(2 \xrightarrow{3} 1\right)$ и т.д.

Последовательность вершин маршрута можно получить, исследуя элементы матриц $A,\ A^2,\ A^3$ и т.д., или, в случае простых графов, по их рисунку. Наиболее наглядный способ состоит в использовании модифицированной матрицы смежности, в которой вместо единиц записаны названия соответствующих рёбер (дуг). В примере 3.2.4 модифицированная матрица смежности будет иметь вид: $A_1 = \begin{pmatrix} 0 & e_1 & 0 & e_2 \\ e_1 & 0 & e_4 & e_3 \\ 0 & e_4 & 0 & 0 \\ e_2 & e_3 & 0 & 0 \end{pmatrix}$.

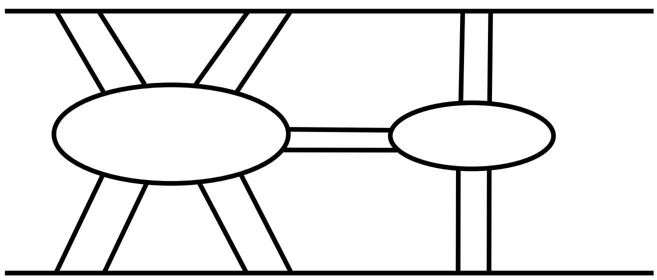
$$\mathrm{A}_1^2 = \begin{pmatrix} 0 & e_1 & 0 & e_2 \\ e_1 & 0 & e_4 & e_3 \\ 0 & e_4 & 0 & 0 \\ e_2 & e_3 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & e_1 & 0 & e_2 \\ e_1 & 0 & e_4 & e_3 \\ 0 & e_4 & 0 & 0 \\ e_2 & e_3 & 0 & 0 \end{pmatrix} = \begin{pmatrix} e_1e_1 + e_2e_2 & e_2e_3 & e_1e_4 & e_1e_3 \\ e_3e_2 & e_1e_1 + e_4e_4 + e_3e_3 & 0 & e_1e_2 \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{pmatrix}.$$

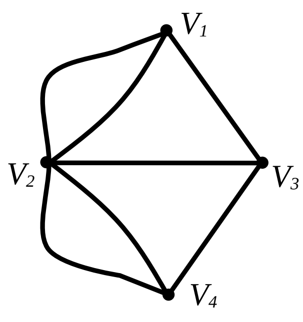
По последней матрице делаем вывод, что т.к. её элемент ${a_{12}}^2 = e_2e_3$, то существует один маршрут длины 2 из вершины 1 в вершину 2, это $(1, e_2, 4, e_3, 2)$; т.к. ${a_{22}}^2 = e_1e_1 + e_4e_4 + e_3e_3$, то существует три маршрута длины 2 из вершины 2 в 2: $(2, e_1, 1, e_1, 2)$, $(2, e_4, 3, e_4, 2)$ и $(2, e_3, 4, e_3, 2)$. По ${A_1}^3$ аналогично можно найти маршруты длины 3 и их количество.

3.3 Эйлеровы и гамильтоновы графы

При решении практических задач часто возникает необходимость обхода вершин или рёбер графа, удовлетворяющих определённым свойствам. Рассмотрим две задачи такого вида, которые отвечают на вопрос, откуда произошли названия «эйлеров цикл», «гамильтонов цикл».

1. Задача о кёнигсберских мостах была сформулирована и решена Эйлером в 1736 году. Она положила начало теории графов как самостоятельной науки. На рисунке 3.3.1 а) изображён план семи мостов через реку Перголи в центре г. Кёнигсберга.





а) б)

Рисунок 3.3.1

Задача заключалась в том, чтобы пройти каждый мост по одному разу и вернуться в исходную точку. На языке теории графов эту задачу можно сформулировать так: дан граф G (см. рисунок 3.3.1 б)), в котором вершины – части города (берег и острова), рёбра – мосты. Вопрос: существует ли в графе G цикл, содержащий все рёбра? Такой цикл в дальнейшем назвали эйлеровым.

2. Название «гамильтонов цикл» произошло от задачи (игры), которую придумал математик У. Гамильтон в 1857 году. Суть этой задачи в следующем: дан граф, представляющий собой укладку на плоскости додекаэдра (см. рисунок 3.3.2).

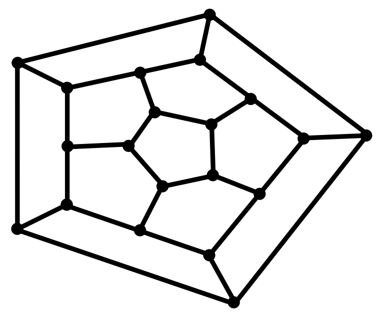


Рисунок 3.3.2

Каждая вершина – город. Требуется обойти города (вершины), посетив каждый город один раз, и вернуться в исходный город. Т.е. найти в графе цикл, проходящий через каждую вершину один раз. Такие циклы в дальнейшем назвали гамильтоновыми. С задачей нахождения гамильтонова цикла связана задача коммивояжера (имеется n городов. Коммивояжер должен посетить каждый по одному разу и вернуться в исходный. Требуется найти кратчайший маршрут) и многие другие важные практические задачи.

Эйлеровым циклом графа называется цикл (не обязательно простой), в котором содержатся все рёбра графа и каждое ребро встречается в нём только один раз. Граф, в котором есть эйлеров цикл, называется эйлеровым. Если граф имеет цепь (не обязательно простую), содержащую все рёбра по одному разу, то такая цепь называется эйлеровой. Эйлеров граф можно нарисовать, не отрывая карандаш от бумаги и не проходя дважды одно и то же ребро. Ясно, что эйлеровым может быть только связный граф, он содержит не только все рёбра (по одному разу), но и все вершины (возможно, по нескольку раз).

Следующая теорема даёт простой способ проверки наличия эйлерова цикла в графе.

Теорема. Связный граф является эйлеровым тогда и только тогда, когда степени всех его вершин чётны (заметим, что в этом случае множество рёбер можно разбить на простые циклы).

Итак, поскольку в графе для кёнигсберских мостов все четыре вершины имеют нечётную степень, то ответ в этой задаче отрицательный: нельзя пройти каждый мост по одному разу и вернуться в исходную точку.

Для эйлеровых графов существует алгоритм, называемый иногда алгоритмом Флери, позволяющий быстро построить один из существующих эйлеровых циклов. Этот алгоритм можно задать следующими правилами:

а) выбираем произвольную вершину v_1 и ребро e_1, инцидентное v_1. Переходим в вершину v_2 по ребру $e_1 = \{v_1, v_2\}$. Присваиваем ребру e_1 номер 1, назовём его пройденным и вычёркиваем;

б) находясь в вершине v_i , не следует выбирать ребро, соединяющее v_i с v_1, если есть возможность другого выбора;

в) находясь в вершине v_i , не следует выбирать ребро, которое является перешейком, т.е. ребром, при удалении которого, граф, образованный не вычеркнутыми рёбрами, распадается на две компоненты связности, каждая из которых имеет хотя бы по одному ребру;

г) после того как в графе будут занумерованы все рёбра, образуется эйлеров цикл, причём мы придём в ту вершину, с которой начали. Порядок нумерации соответствует последовательности обхода рёбер.

П р и м е р 3.3.1 - Найдём эйлеров цикл в графах на рисунках 3.3.3 а) и б).

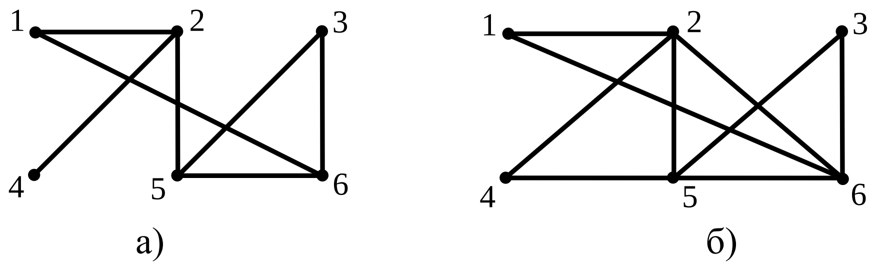


Рисунок 3.3.3

Граф на рисунке 3.3.3 а) не содержит эйлеровых циклов, так как в нём есть вершины, имеющие нечётные степени: $\rho(2) = \rho(5) = \rho(6) = 3,\ \rho(4) = 1$.

Граф на рисунке 3.3.3 б) эйлеров, поскольку $\rho(1) = \rho(3) = \rho(4) = 2$, $\rho(2) = \rho(5) = \rho(6) = 4$. Построим эйлеров цикл в этом графе:

а) выберем вершину 1 и ребро $e_1 = \{1,2\}$, присвоив ему номер 1, перейдём в вершину 2;

б) находясь в вершине 2, не выбираем пройденное ребро e_1, из оставшихся инцидентных этой вершине рёбер ни одно не является перешейком, поэтому выбираем любое, например, $e_2 = \{2,4\}$, присваиваем ему номер 2 и переходим в вершину 4;

в) рассуждая аналогично, обходим оставшиеся рёбра в порядке:

$e_3 = \{4,5\}$, $e_4 = \{5,3\}$, $e_5 = \{3,6\}$, $e_6 = \{6,2\}$, $e_7 = \{2,5\}$, $e_8 = \{5,6\}$, $e_9 = \{6,1\}$.

Итак, получен эйлеров цикл $e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9$.

Граф называется гамильтоновым, если в нём имеется простой цикл, содержащий все вершины графа по одному разу, этот цикл также называется гамильтоновым. Гамильтоновой называется и простая цепь, содержащая все вершины графа. Гамильтонов цикл не обязательно содержит все рёбра графа.

Очевидно, что гамильтоновым может быть только связный граф. Задача нахождения гамильтоновых циклов значительно сложнее аналогичной задачи для эйлеровых циклов. Неизвестен такой же общий простой критерий существования гамильтонова цикла в графе. Известны лишь некоторые достаточные условия его существования. Необходимые условия существования неизвестны за исключением некоторых особых типов графов. Приведём два достаточных признака, простых в практическом применении. Пусть дан связный н-граф G без петель, имеющий $n \geq 3$ вершин.

Теорема. Если для любых двух несмежных вершин u и v графа G выполняется условие $\rho(u) + \rho(v) \geq n$, то в G существует гамильтонов цикл.

Следствие. Если для любой вершины v графа G выполняется условие $\rho(v) \geq n/2$, то в G существует гамильтонов цикл.

П р и м е р 3.3.2 Полный граф K_5 (см. рисунок 3.1.10) является как эйлеровым, так и гамильтоновым, так как выполняются необходимые и достаточные условия эйлеровости и достаточные условия гамильтоновости: число вершин $n = 5$, произвольно обозначив вершины через a, b, c, d, e, имеем $\rho(a) = \rho(b) = \rho(c) = \rho(d) = \rho(e) = 4$ и для любых двух вершин графа выполняется $\rho(u) + \rho(v) = 8 > 5$, где $u, v \in \{a, b, c, d, e\}$.

П р и м е р 3.3.3 - Для графа на рисунке 3.3.3 б), число вершин которого $n = 6$, не выполняется достаточное условие существования гамильтонова цикла: например, $\rho(1) + \rho(3) = 4 < 6$, однако, такой цикл в графе существует, это, например, цикл 1, 2, 4, 5, 3, 6, 1. Таким образом, этот граф как эйлеров, так и гамильтонов.

3.4 Расстояния в графах, взвешенные графы

Пусть $G = (V, E)$ связный н-граф, v_i, v_j – две разные вершины G.

Определение. Расстоянием между вершинами v_i и v_j (обозначается $d(v_i, v_j)$) называется минимальная длина простой цепи с концами в v_i и v_j.

Таким образом, расстояние удовлетворяет всем аксиомам метрики:

а) $d(v_i, v_j) \geq 0$, причём $d(v_i, v_j) = 0 \Leftrightarrow v_i = v_j$;

б) $d(v_i, v_j) = d(v_j, v_i)$;

в) $d(v_i, v_j) \leq d(v_i, v_k) + d(v_k, v_j)$ – неравенство треугольника.

Пусть $V = \{v_1, v_2, \ldots, v_n\}$ – множество вершин графа. Рассмотрим матрицу $D = (d_{ij})$, где $d_{ij} = d(v_i, v_j)$. D называется матрицей расстояний. Очевидно, $D = D^T$, т.е. D – симметрическая матрица.

Введём ещё несколько понятий:

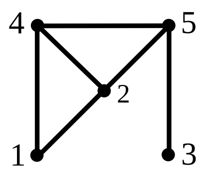
1. Эксцентриситетом вершины v (обозначается $e(v)$) называется $\max\{d(u,v) \mid u \in V\}$, т.о., эксцентриситет равен расстоянию от данной вершины до наиболее удалённой от неё. В матрице расстояний $e(v_i)$ равен наибольшему из чисел, стоящих в i-ой строке.

2. Диаметром графа G (обозначается $d(G)$) называется максимальный из всех эксцентриситетов: $d(G) = \max\{e(v) : v \in V\}$.

3. Радиусом графа G (обозначается $r(G)$) называется минимальный из всех эксцентриситетов: $r(G) = \min\{e(v) : v \in V\}$.

4. Вершина v называется периферийной, если $e(v) = d(G)$; Вершина v называется центральной, если $e(v) = r(G)$. Множество всех центральных вершин графа называется его центром.

П р и м е р 3.4.1 Рассмотрим граф на рисунке 3.4.1.



$$D = \begin{pmatrix} 0 & 1 & 3 & 1 & 2 \\ 1 & 0 & 2 & 1 & 1 \\ 3 & 2 & 0 & 2 & 1 \\ 1 & 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 \end{pmatrix}$$ – матрица расстояний графа.

Рисунок 3.4.1

Наибольшее число в строке – это эксцентриситет соответствующей вершины, поэтому $e(1) = 3, e(2) = 2,\ e(3) = 3,\ e(4) = 2,\ e(5) = 2$. Диаметр графа $d(G) = 3$, как наибольший из эксцентриситетов, радиус – $r(G) = 2$, как наименьший. Вершины 1, 3 – периферийные, вершины 2,4,5 – центральные, множество {2,4,5} – центр графа.

Задача нахождения центральных вершин часто возникает на практике, например, если вершинам графа соответствуют районы города, рёбрам – дороги между ними. Требуется оптимально разместить учреждения общего пользования (больницы, рынки, и т.д.). Очевидно, что в этом случае оптимизация заключается в минимизации расстояний от самых отдалённых районов до этих учреждений. Следовательно, местами размещения должны быть центральные вершины графа. Реальные задачи от этих идеальных отличаются тем, что приходится учитывать и другие обстоятельства: расстояние между районами, стоимость проезда, время проезда и т.д. Для учёта различных обстоятельств используют взвешенные графы.

Определение. Пусть дан граф $G = (V, E)$. Пометкой (или распределением меток) графа называется пара функций: $f: V \to S_V$ и $g: E \to S_E$, где S_V, S_E – множества меток вершин и рёбер (дуг). Четвёрка $((V, E, f, g))$ называется взвешенным или помеченным графом.

Если $v \in V$, то $f(v)$ – вес вершины v, если $e \in E$, то $g(e)$ – вес дуги е. Часто бывают помеченными только вершины или только рёбра (дуги).

П р и м е р 3.4.2 - На рисунке 3.4.2 изображён помеченный граф (V, E, f, g), представляющий схему автомобильных дорог с указанием их длины.

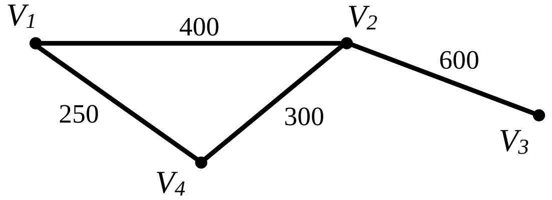


Рисунок 3.4.2

V_1 – Кокчетав, V_2 – Экибастуз, V_3 – Семипалатинск, V_4 – Караганда. $V = \{V_1, V_2, V_3, V_4, V_5\}$, $E = \{\{V_1, V_2\}, \{V_2, V_3\}, \{V_1, V_4\}, \{V_2, V_4\}\}$, $f\colon V \to S_V$: S_V – множество городов, $f(V_1) =$ Кокчетав, $f(V_2) =$ Экибастуз, $f(V_3) =$ Семипалапинск, $f(V_4) =$Караганда. $g\colon V \to S_E$: S_E – множество расстояний, $g(\{V_1, V_2\}) = 400$, $g(\{V_2, V_3\}) = 600$, $g(\{V_1, V_4\}) = 250$, $g(\{V_2, V_4\}) = 300$.

Информацию о весах ребер (дуг) во взвешенном графе можно представлять в виде матрицы весов $W = (w_{ij})$, где w_{ij} – вес ребра (дуги) (v_i, v_j), если эта дуга существует, если не существует, то соответствующий вес обозначают 0 или ∞ (0 - если $i = j$; ∞ - если вершины v_i и v_j не смежные). Для графа из примера 3.4.2 матрица весов имеет вид: $W = \begin{pmatrix} 0 & 400 & \infty & 250 \\ 400 & 0 & 600 & 300 \\ \infty & 600 & 0 & \infty \\ 250 & 300 & \infty & 0 \end{pmatrix}$.

Для взвешенных графов также вводятся понятия взвешенных расстояния, эксцентриситета и т.д. Пусть $G = (V, E)$ взвешенный граф, в котором вес каждой дуги (v_i, v_j) есть некоторое число $\mu\,(v_i, v_{i+1})$, $v_i, v_{i+1} \in$ V.

1. Весом маршрута $((v_1 \leftrightarrow v_{n+1})$ называется число $\mu = \sum_{i=1}^{n} \mu\,(v_i\,, v_{i+1})$.

2. Взвешенным расстоянием $d_w(v_1, v_{n+1})$ между вершинами v_1 и v_{n+1} называется минимальный из весов $v_1 \leftrightarrow v_{n+1}$ маршрутов.

3. Кратчайшим маршрутом между v_1 и v_{n+1} называется маршрут, вес которого равен $d_w(v_1, v_{n+1})$.

4. Взвешенным эксцентриситетом вершины v (обозначается $e_w(v)$) называется число $e_w(v) = max\{d_w(v, u)\colon u \in V\}$.

5. Взвешенной центральной вершиной графа G называется вершина v , для которой $e_w(v) = min\{e_w(u)\colon u \in V\}$.

6. Взвешенным радиусом графа G (обозначается $r_w(G)$) называется взвешенный эксцентриситет центральной вершины.

Нахождение кратчайших маршрутов (путей)

Существуют различные способы (алгоритмы) определения кратчайших маршрутов. Выбор каждого из них зависит от условий задачи (например, весы рёбер отрицательные или нет), от способа решения (вручную или на компьютере) и т.д.

Рассмотрим алгоритм Дейкстры, который можно использовать только для взвешенных графов с неотрицательными весами всех рёбер. Этот алгоритм можно встретить в различных вариантах, например, с использованием и без матрицы весов, для применения в компьютере и т.д. Мы рассмотрим два варианта алгоритма Дейкстры (с использованием и нет матрицы весов).

Алгоритм Дейкстры 1.

Пусть $G = (V, E)$ – взвешенный граф, имеющий n вершин и $W = (w_{ij})$, – его матрица весов ($w_{ij} > 0$). Пусть требуется найти взвешенное расстояние от фиксированной вершины v_i (она называется источником) до некоторой другой вершины (вообще по алгоритму Дейкстры 1 находят сразу расстояния от источника до всех остальных вершин).

Обозначим T_1 множество вершин $V \setminus \{v_i\}$, т.е. T_1 – это множество всех вершин V без источника. Обозначим $D^{(1)} = (d_1^{(1)}, d_2^{(1)}, \dots, d_n^{(1)})$, где $d_1^{(1)} = 0$, $d_j^{(1)} = w_{ij}$, $(i \neq j)$, т.е. $D^{(1)}$ – это i-ая строка в матрице весов.

Пусть на шаге S уже определены множества вершин T_s и строка $D^{(S)} = = (d_1^{(S)}, d_2^{(S)}, \dots, d_n^{(S)})$. Наша задача перейти от T_s к T_{s+1} и от $D^{(S)}$ к $D^{(S+1)}$.

Выберем в T_s вершину v_j такую, что $d_j^{(s)} = min\left\{d_k^{(s)},\ v_k \in T_S\right\}$. Положим $T_{S+1} = T_S \setminus \{v_j\}$, $D^{(S+1)} = (d_1^{(S+1)}, d_2^{(S+1)}, \dots, d_n^{(S+1)})$, где $d_k^{(S+1)} = = d_k^{(S)}$, если $v_k \notin T_{S+1}$, $d_k^{(S+1)} = min\{d_k^{(S)}, d_j^{(S)} + w_{jk}\}$, если $v_k \in T_{S+1}$.

На шаге $S = n - 1$ получим строку $D^{(n-1)} = (d_1^{(n-1)}, d_2^{(n-1)}, \dots, d_n^{(n-1)})$, в которой $d_1^{(n-1)}$ равно взвешенному расстоянию от вершины v_i до вершины v_j: $d_j^{(n-1)} = d_w(v_i, v_j)$.

П р и м е р 3.4.3 Рассмотрим граф на рисунке 3.4.3.

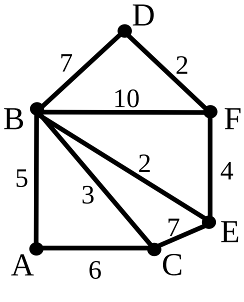


Рисунок 3.4.3

$$W = \begin{pmatrix} 0 & 5 & 6 & \infty & \infty & \infty \\ 5 & 0 & 3 & 7 & 2 & 10 \\ 6 & 3 & 0 & \infty & 7 & \infty \\ \infty & 7 & \infty & 0 & \infty & 2 \\ \infty & 2 & 7 & \infty & 0 & 4 \\ \infty & 10 & \infty & 2 & 4 & 0 \end{pmatrix}$$ - матрица весов.

Пусть требуется найти кратчайшие взвешенные расстояния от вершины A (источник) до остальных вершин. $V = \{A, B, C, D, E, F\}$ – множество вершин. Будем полагать, что $min(a, \infty) = a,\ min(\infty, \infty) = \infty,\ a + \infty = \infty,\ \infty + \infty = \infty.$

1 шаг. A- источник, $T_1 = V - \{A\} = \{B, C, D, E, F\},\ D^{(1)} = \left(0, \underline{\hat{5}}, \hat{6}, \hat{\infty}, \hat{\infty}, \hat{\infty}\right).$

2 шаг. В $D^{(1)}$ выбираем среди элементов, отмеченных ^ (т.е. кроме элемента отвечающего источнику) наименьший, и подчёркиваем его. Из T_1 убираем вершину, соответствующую подчёркнутому элементу (это B). Строим $T_2 = T_1 - \{B\} = \{C, D, E, F\}$. Для определения $D^{(2)}$ делаем вспомогательную запись: из матрицы W выписываем строку, соответствующую B, и прибавляем ко всем её элементам (кроме первого и второго) подчёркнутое число 5. Сравним полученные числа с соответствующими числами в $D^{(1)}$ и в $D^{(2)}$ставим на их места наименьшие числа:

$$+\begin{array}{cccccc} 5 & 0 & 3 & 7 & 2 & 10 \\ & & 5 & 5 & 5 & 5 \\ \hline & & 8 & 12 & 7 & 15, \end{array}$$

$D^{(2)} = \left(0, 5, \underline{\hat{6}}, \widehat{12}, \hat{7}, \widehat{15}\right).$

3 шаг. $T_3 = T_2 \setminus \{C\} = \{D, E, F\},$ $$+\begin{array}{cccccc} 6 & 3 & 0 & \infty & 7 & \infty \\ & & & 6 & 6 & 6 \\ \hline & & & \infty & 13 & \infty, \end{array}$$

$D^{(3)} = \left(0, 5, 6, \widehat{12}, \underline{\hat{7}}, \widehat{15}\right).$

4 шаг. $T_4 = T_3 \setminus \{E\} = \{D, F\},$ $$+\begin{array}{cccccc} \infty & 2 & 7 & \infty & 0 & 4 \\ & & & 7 & 7 & 7 \\ \hline & & & \infty & 7 & 11, \end{array}$$

$D^{(4)} = \left(0, 5, 6, 1\hat{2}, 7, \underline{\widehat{11}}\right).$

5 шаг. $T_5 = T_4 - \{F\} = \{D\},$ $$+\begin{array}{cccccc} \infty & 10 & \infty & 2 & 4 & 0 \\ & & & 11 & & 11 \\ \hline & & & 13 & & 11, \end{array}$$

$D^{(5)} = (0, 5, 6, 12, 7, 11).$

$n = 6,\ n - 1 = 5,$ значит, это последний шаг. По виду $D^{(5)}$ делаем вывод, что взвешенные расстояния от вершины A до остальных вершин равны $d_w(A, A) = 0,\quad d_w(A, B) = 5,\quad d_w(A, C) = 6,\quad d_w(A, D) = 12,\quad d_w(A, E) = 7,$ $d_w(A, F) = 11.$

Алгоритм Дейкстры 2

Этот алгоритм позволяет найти не только вес кратчайшего пути, но и сам путь, его иногда называют алгоритмом расстановки меток. Пусть требуется найти кратчайшее взвешенное расстояние и путь от вершины v_1 до вершины v_n. Каждой вершине поставим в соответствие пару (метку) $(\infty, 0)$. Если вершина v_i имеет метку (w, v_k), то первая координата означает присвоенное расстояние от v_1 до v_i, вторая – предыдущую вершину пути от v_1 до v_i. Метки, как и вершины, могут находиться в двух состояниях – быть временными и постоянными.

Алгоритм содержит следующие шаги:

1) Начинаем с источника v_1. Приписываем этой вершине метку $(0{,}0)$ и делаем постоянной. Каким-либо образом это отмечаем, например, звёздочкой или подчёркиваем: v_1*(0,0). Остальные вершины временные и имеют метки $(\infty, 0)$.

2) Пусть вершина $v_j(w, v_k)$ стала постоянной. Рассмотрим все вершины v_j, смежные с v_i. Прибавим величину w к расстоянию (весу) от v_1 до v_j. Если эта сумма меньше, чем временное расстояние, присвоенное вершине v_j (т.е. её предыдущая первая координата), то заменим этой суммой первую координату v_j, а вторую координату заменим на v_i.

3) Находим минимальное из расстояний, приписанных временным вершинам v_j, смежным с v_j. Вершину с наименьшим весом делаем постоянной.

4) Если вершина v_n не постоянна, то возвращаемся к пункту 2, если v_n постоянна, то расстояние, присвоенное вершине v_n, является кратчайшим взвешенным расстоянием от v_1 до v_n. Алгоритм закончен.

Для нахождения пути от v_1 до v_n надо начать с v_n, по второй координате найти предыдущую ей вершину, для этой вершины аналогично найти предыдущую и т.д., пока не будет достигнута первая вершина пути v_1. Перестановка вершин в обратном порядке даст путь от v_1 до v_n.

Рассмотрим этот метод на том же примере. Для графа на рисунке 3.4.3 найти кратчайшее расстояние и путь от вершины A до вершины F.

Заметим, что этот алгоритм можно оформлять, изображая на каждом шаге граф с указанием вершин с их метками, найденными на этом шаге (см., например, [4], стр.611). Мы обойдёмся без рисунков.

1 шаг. Приписываем источнику A метку $(0{,}0)$, делаем её постоянной и отмечаем $A^*(0{,}0)$, остальные вершины временные и имеют метки $(\infty, 0)$: $B(\infty, 0)$, $C(\infty, 0)$,…, $F(\infty, 0)$.

1-ая итерация.

2 шаг. С вершиной A смежны вершины B и C:

- от $A^*(0{,}0)$ до $B(\infty, 0)$ расстояние 5, $5 + 0 < \infty$, поэтому B меняет метку: $B(5, A)$;

- от $A^*(0,0)$ до $C(\infty, 0)$ расстояние 6, $6 + 0 < \infty$, поэтому C меняет метку: $C(6, A)$;

3 шаг. $min(5,6) = 5$, поэтому B становится постоянной, отмечаем её: $B^*(5, A)$. $B \neq F$, возвращаемся на второй шаг.

2-ая итерация.

2 шаг. С вершиной B смежны вершины C, D, E, F:

- от $B^*(5, A)$ до $C(6, A)$ расстояние 3, $5 + 3 = 8 > 6$, $\to C$ не меняет метку: C(6, A);

- от $B^*(5, A)$ до $D(\infty, 0)$ расстояние 7, $5 + 7 = 12 < \infty$, $\to$ D меняет метку: $D(12, B)$;

- от $B^*(5,\text{A})$ до $E(\infty, 0)$ расстояние 2, $5 + 2 = 7 < \infty$, поэтому E меняет метку: $E(7,\text{B})$;

- от $B^*(5, A)$ до $F(\infty, 0)$ расстояние 10, $5 + 10 = 15 < \infty$, $\to$ F меняет метку: $F(15, B)$;

3 шаг. $min(6{,}12{,}7{,}15) = 6$, поэтому C становится постоянной: $C^*(6, A)$. $C \neq F$, возвращаемся на второй шаг.

3-ая итерация.

2 шаг. С вершиной C смежна вершина E (рассматриваем только временные вершины):

- от $C^*(6, A)$ до $E(7, B)$ расстояние 7, $6 + 7 = 13 > 7$, $\to$ E не меняет метку: $E(7, B)$;

3 шаг. $E^*(7, B)$ - постоянная, $E \neq F$, возвращаемся на второй шаг.

4-ая итерация.

2 шаг. С вершиной E смежна вершина F:

- от $E^*(7, B)$ до $F(15, B)$ расстояние 4, $7 + 4 = 11 < 15$, $\to$ F меняет метку: $F(11, E)$;

3 шаг. $F^*(11, E)$ - постоянная.

Для наглядности эти рассуждения можно оформить следующей схемой:

A - источник $\longrightarrow A^*(0,0)$ постоянная вершина, $\left[\begin{array}{l} B(\infty, 0) \\ C(\infty, 0) \\ D(\infty, 0) \\ E(\infty, 0) \\ F(\infty, 0) \end{array}\right.$ - временные вершины;

1)$A^*(0,0) - \left[\begin{array}{l} \xrightarrow{5} B(\infty, 0) - 0 + 5 = 5 < \infty \to B(5, A) \\ \xrightarrow{6} C(\infty, 0) - 0 + 6 = 6 < \infty \to C(6, A) \end{array}\right.$ - min(5,6)=5 $\to B^*(5, A)$ -пост.;

2) $B^*(5,A) - \left[\begin{array}{l} \xrightarrow{3} C(6,A) - 5 + 3 = 8 > 6 \to C(6,A) \\ \xrightarrow{7} D(\infty,0) - 5 + 7 = 12 < \infty \to D(12,B) \\ \xrightarrow{2} E(\infty,0) - 5 + 2 = 7 < \infty \to E(7,B) \\ \xrightarrow{10} F(\infty,0) - 5 + 10 = 15 < \infty \to F(15,B) \end{array}\right. - min(6,12,7,15) =$
$= 6 \to C^*(6,A)$ - пост.;

3) $C^*(6,A) - \left[\xrightarrow{7} E(7,B) - 6 + 7 = 13 > 7 \to E^*(7,B)\right] \to E^*(7,B)$ – пост.;

4) $E^*(7,B) - \left[\xrightarrow{4} F(15,B) - 7 + 4 = 11 < 15 \to F^*(11,E)\right] \to F^*(11,E)$ – пост.

Итак, вершина F, до которой надо было найти кратчайшее расстояние, стала постоянной, поэтому процесс завершен. 11 есть кратчайшее расстояние от A до F. Если бы совокупность вершин, смежных с постоянной вершиной, была исчерпана до того, как мы достигли вершину F, то задача не имела бы решения, т.к. не было бы пути от A до F.

Для нахождения кратчайшего пути идём последовательно от F к A по указанию второй координаты: $F(11,E)$, $E(7,B)$, $B(5,A)$, A. Переставив вершины в обратном порядке, получим искомый путь: (A,B,E,F).

Потоки в сетях

Примерами нагруженных графов являются графы, представляющие сети. Сети – это системы, которые служат для транспортировки некоторых продуктов из одной точки в другую. Например, система нефтепровода, по которой течёт нефть, транспортная система, по которой перемещаются потоки грузов, водопроводная, компьютерная сети и т.д. Поскольку пропускная способность каналов сетей ограничена, то в практических задачах чаще всего требуется определить максимальные потоки, которые способны пропускать сети. Эта задача успешно решается в теории графов.

Сетью называется связный ориентированный граф $G(V,E)$ без петель. В сети выделяют две вершины s - источник (исток) и t - сток. Каждой дуге $e = (u,v)$ ставится в соответствие число $c(e)$–пропускная способность дуги e.

Аналогично матрице смежности или матрице весов, представляющих граф, сеть можно представить матрицей пропускных способностей $C = (c_{ij})$,

где $c_{ij} = \begin{cases} c(v_i,v_j), \text{если } (v_i,v_j) \text{ существует} \\ 0, \text{если } (v_i,v_j) \text{ не существует} \end{cases}$.

Потоком в сети называется такая функция f, определённая на множестве дуг E ($f\colon \mathrm{E} \to \mathrm{N} \cup \{0\}$), что

1) $\forall e \in \mathrm{E}: 0 \le f(e) \le c(e)$;

2) $\forall v \in V$ такой, что $v \ne s$ и $v \ne t$: $\sum_{(u,v)\in \mathrm{E}} f(u,v) = \sum_{(v,u)\in \mathrm{E}} f(v,u)$.

Первое условие потока очевидно: величина потока в дуге не может превысить её пропускную способность. Второе условие называется условием сохранения потока: в промежуточных вершинах потоки не создаются и не исчезают. Величину $\Delta(u,v) = c(u,v) - f(u,v)$ называют остаточной пропускной способностью дуги (u,v). Если $f(u,v) = c(u,v)$, то дугу (u,v) называют насыщенной.

Важную роль при определении максимальных потоков играет понятие разреза. Пусть множество вершин V разбито на два непустых непересекающихся подмножества S и T: $V = S \cup T$, $S \cap T = \emptyset$.

Множество дуг, начала которых лежат в S, а концы - в T, называется ориентированным разрезом, обозначается $P = (S,T)$. Таким образом,

$P = (S,T) = \{(u,v) | u \in S, v \in T\}$.

Пропускной способностью разреза называется сумма пропускных способностей входящих в него дуг. Поток в сети называется максимальным, если его величина не меньше величины любого другого потока; разрез называется минимальным, если его пропускная способность не больше пропускной способности любого другого разреза сети.

Теорема Форда-Фалкерсона. Максимальный поток в сети равен пропускной способности минимального разреза.

Алгоритм Форда-Фалкерсона нахождения максимального потока основан на следующем:

а) пусть в сети есть путь из s в t, состоящий из ненасыщенных дуг. Тогда поток в сети можно увеличить на величину Δ, равную минимальной из остаточных пропускных способностей дуг, входящих в этот путь. Переберём все возможные такие пути из s в t, проведем в них процедуру увеличения потока. В результате получим полный поток, для которого каждый путь из s в t содержит, по крайней мере, одну насыщенную дугу;

б) рассмотрим произвольный маршрут из s в t, состоящий как из прямых (ориентированных от s к t), так и обратных (ориентированных от t к s) дуг. Пусть в этом маршруте прямые дуги не насыщены, а потоки на обратных дугах положительны. Обозначим Δ_1- минимальную из остаточных пропускных способностей прямых дуг, Δ_2- минимальную из величин потоков на обратных дугах. Тогда поток в сети можно увеличить на величину $\Delta = \min\{\Delta_1, \Delta_2\}$, прибавляя Δ к потокам на прямых дугах и вычитая на обратных. Ясно, что при этом условие сохранения потока для вершин, входящих в рассматриваемый маршрут, сохраняется.

Заметим, что дуги, ставшие насыщенными, не должны входить в рассматриваемые далее маршруты. Для удобства их можно на чертеже вычёркивать. Кроме того, если в процессе перебора маршрутов все дуги, исходящие из источника или входящие в сток, стали насыщенными, то алгоритм закончен и максимальный поток найден.

Пример 3.4.4 Найти максимальный поток из s в t и указать минимальный разрез сети $G=(V,E)$, $V=\{s,v_1,v_2,v_3,v_4,t\}$, заданной матрицей пропускных способностей $C=\begin{pmatrix} 0 & 13 & 11 & 0 & 0 & 0 \\ 0 & 0 & 11 & 6 & 0 & 0 \\ 0 & 0 & 0 & 11 & 13 & 17 \\ 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.

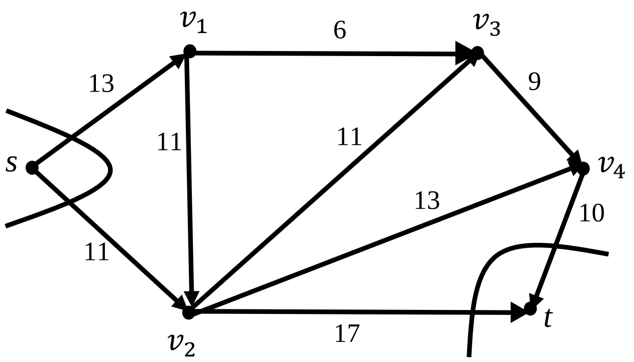


Рисунок 3.4.4

По матрице C построим граф (см. рисунок 3.4.4). Начнём перебирать все пути из s в t, указывая над стрелками поток через данную дугу и её пропускную способность (в скобках), под стрелкой то же после увеличения потока на Δ. Полагаем, что вначале поток через все дуги равен нулю:

1) $s \xrightarrow[11(11)]{0(11)} v_2 \xrightarrow[11(17)]{0(17)} t$, $\Delta = min(11,17) = 11;$

2) $s \xrightarrow[10(13)]{0(13)} v_1 \xrightarrow[10(11)]{0(11)} v_2 \xrightarrow[10(13)]{0(13)} v_4 \xrightarrow[10(10)]{0(10)} t$, $\Delta = min(13,11,13,10) = 10;$

3) $s \xrightarrow[11(13)]{10(13)} v_1 \xrightarrow[11(11)]{10(11)} v_2 \xrightarrow[12(17)]{11(17)} t$, $\Delta = min(13-10,11-10,17-11) = 1;$

Больше прямых путей из s в t без насыщенных дуг нет. Рассмотрим маршруты из s в t, содержащие как прямые, так и обратные дуги:

$$1)\ s \xrightarrow[13(13)]{11(13)} v_1 \xrightarrow[2(6)]{0(6)} v_3 \xleftarrow[-2(11)]{0(11)} v_2 \xrightarrow[14(17)]{12(17)} t,\ \Delta = min(13-11,6,11,17-12) = 2.$$

Обе дуги, выходящие из источника, стали насыщенными. Алгоритм закончен. Максимальный поток равен $f_{max} = f(s, v_1) + f(s, v_2) = 13 + 11 = 24$, или $f_{max} = f(v_4, t) + f(v_2, t) = 10 + 14 = 24$. Минимальный разрез $P = (S, T)$, где $S = \{s\}$, $T = \{v_1, v_2, v_3, v_4, t\}$ или $S = \{s, v_1, v_2, v_3, v_4\}$, $T = \{t\}$.

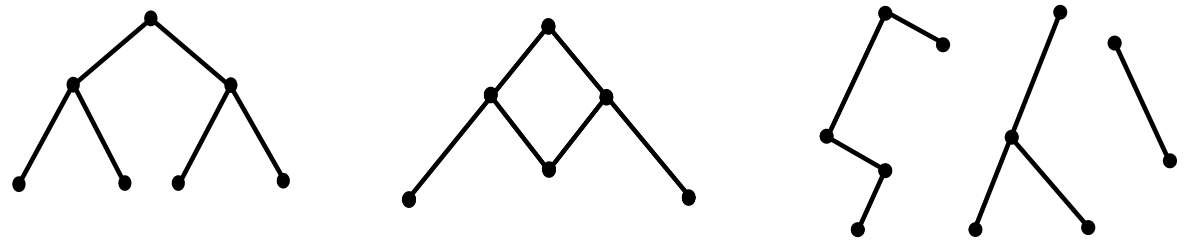
3.5 Деревья, лес. Минимальные остовные деревья

Понятие дерева широко применяется во многих областях математики и информатики. Это самый распространённый и в некотором смысле простейший класс графов. Деревья используют как инструмент при вычислениях, как удобный способ хранения информации, как способ сортировки и поиска данных.

Определение. Деревом называется связный н-граф**,** не содержащий циклов; лесом называется несвязный н-граф без циклов.

Таким образом, компонентами связности любого леса являются деревья.

П р и м е р 3.5.1 - На рисунке 3.5.1 а) изображено дерево; на рисунке 3.5.1 б) граф, не являющийся деревом, так как содержит цикл; на рисунке 3.5.1 в) лес, состоящий из трёх деревьев.



а) б) в)

Рисунок 3.5.1

Теорема (*). Пусть G – н-граф без петель, имеющий n вершин, тогда следующие условия эквивалентны:

а) G – дерево;

б) G – связный граф, содержащий $n-1$ ребро;

в) G – ациклический граф, содержащий $n-1$ ребро;

г) любые две не совпадающие вершины графа G соединяет единственная простая цепь;

д) G – ациклический граф такой, что если какую-либо пару его несмежных вершин соединить ребром, то полученный граф будет содержать ровно один цикл.

Введём несколько новых понятий: любое дерево можно «подвесить» за какую-либо его вершину v , т.е. v расположить на верхнем этаже, смежные с ней вершины этажом ниже и т.д. В этом случае вершина v называется корнем дерева. Если степень вершины v равна 1 ($\rho(v)$=1), то она называется листом (концевой, висячей). На рисунке 3.5.2 вершина v корень дерева; вершины v_3, v_4, v_5,v_7 и v_8- листья.

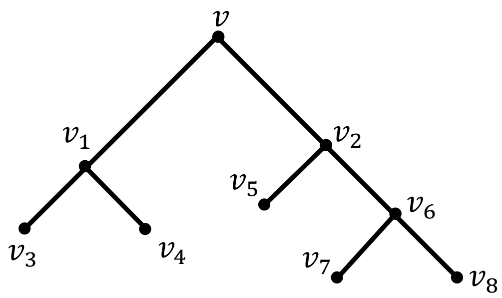


Рисунок 3.5.2

Для каждого неориентированного дерева с корнем можно определить соответствующее ориентированное дерево с корнем: все рёбра такого дерева ориентируются от корня, причём данная ориентация единственная. При выборе другой вершины - корня получается другой орграф-дерево.

Если корень выбран, то уровнем вершины называется длина единственного пути из корня в эту вершину. Высотой дерева называется длина самого длинного пути от корня до листа. Общепринято соглашение о том, что корень находится наверху и все дуги ориентированы сверху вниз, поэтому стрелки на дугах можно не изображать. Таким образом, корень имеет уровень 0, а все вершины одного уровня удобно изображать на одном этаже (ярусе). Поскольку ориентированные деревья являются абстракцией иерархических отношений, то часто встречается следующая терминология:

1) вершины, достижимые из вершины u , называются потомками этой вершины, а вершина u предком этих вершин;

2) если в дереве существует дуга (u, v), то вершина u называется родителем (отцом) v, а вершина v сыном u;

3) если u родитель v_1 и v_2, то v_1 и v_2 называются братьями;

4) если наибольшая из степеней выхода для вершин дерева равна m, то дерево называется m - арным; при $m = 2$ дерево называют бинарным;

5) в бинарном дереве, если у родителя два сына, то один левый сын, другой правый; если сын один, то он либо левый, либо правый.

Пример 3.5.2 Рассмотрим граф на рисунке 3.5.2. Это бинарное дерево. Уровень вершины v_6 равен 2, уровень вершины v_8 равен 3. Высота дерева равна 3. Вершина v_1 является родителем для v_3 и v_4, v_3 – левый сын, v_4– правый сын вершины v_1, также v_3 и v_4 – братья. v_2 – предок v_5, v_6, v_7 и v_8, а v_5, v_6, v_7 и v_8 – потомки v_2.

Определение. Рассмотрим н-граф $G = (V, E)$. Подграф $G' = (V', E')$ графа G называется остовом или каркасом графа G, если $V = V'$ и G'– лес, который на любой связной компоненте графа G образует дерево.

Если G связный граф, то его остов G' есть подграф графа G, являющийся деревом, которое называется остовным деревом.

Очевидно, что в каждом графе существует остов, его можно получить, если разрушать в каждой связной компоненте циклы, удаляя лишние рёбра.

Пример 3.5.3 - Граф G на рисунке 3.5.3 состоит из двух связных компонент. Рёбра обозначены цифрами. В качестве остова G_1 графа можно взять лес с рёбрами 2, 3, 4, 6, 7, удалив рёбра 1, 5, 8. Остов определяется неоднозначно, например, удалив рёбра 3, 5, 6, также получим лес, состоящий из двух остовных деревьев с рёбрами 2, 1, 4 и 7, 8.

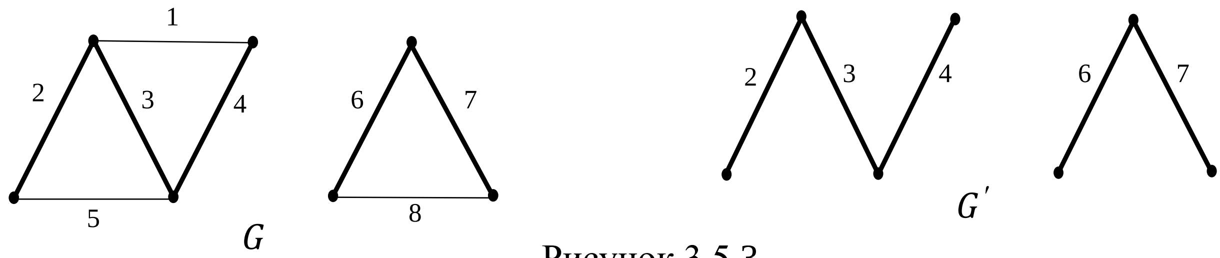


Рисунок 3.5.3

Теорема. Число рёбер произвольного н-графа G, которое необходимо удалить для получения остова, не зависит от последовательности их удаления и равно $m - n + k$, где m – число рёбер, n – число вершин, k – число компонент связности графа.

Доказательство. Пусть C_i – i-ая компонента связности графа G $(i = \overline{1, k})$. Пусть C_i содержит n_i вершин. Тогда по теореме (*) её остовное дерево K_i будет содержать $(n_i - 1)$ рёбер.

Если m_i – число рёбер C_i, $(n_i - 1)$ – число рёбер K_i, то для получения K_i из C_i нужно удалить $m_i - (n_i - 1) = m_i - n_i + 1$ рёбер. Всего компонент связности k, поэтому суммируя все удалённые рёбра по всем компонентам связности графа G получим:

$$\sum_{i=1}^{k}(m_i - (n_i - 1)) = \sum_{i=1}^{k} m_i - \sum_{i=1}^{\mathrm{k}} n_i + \sum_{i=1}^{k} 1 = m - n + k.$$

Определение. Число $v(G) = m - n + k$ называется цикломатическим числом (или цикломатическим рангом) графа G. Число $v*(G) = n - k$ называется коциклическим рангом или корангом. Таким образом, $v*(G)$ – это число рёбер, входящих в любой остов графа G.

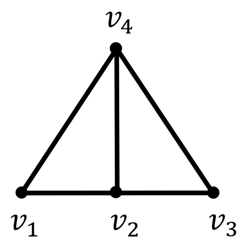
Следствия: а) н-граф G является деревом (лесом) тогда и только тогда, когда $v(G) = 0$; б) н-граф G имеет единственный цикл тогда и только тогда, когда $v(G) = 1$.

Число остовов в графе можно найти с помощью матрицы Кирхгофа, которая определяется так: $K = (k_{ij})$, где

$$k_{ij} = \begin{cases} -1, \text{если } v_i \text{ и } v_j \text{ смежные,} \\ 0, \text{если } v_i \text{ и } v_j \text{ несмежные,} \\ \rho(v_i), i = j, (\rho(v_i) - \text{степень вершины } v_i). \end{cases}$$

Теорема Кирхгофа. Число остовных деревьев в связном графе, имеющем n вершин ($n \geq 2$), равно алгебраическому дополнению любого элемента матрицы Кирхгофа.

П р и м е р 3.5.4 Для графа на рисунке 3.5.4 степени вершин будут равны $\rho(v_1) = \rho(v_3) = 2, \rho(v_2) = \rho(v_4) = 3$.



$$K = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix} - \text{ матрица Кирхгофа.}$$

Рисунок 3.5.4

Рассмотрим, например, алгебраическое дополнение элемента k_{11}:

$A_{11} = (-1)^{1+1} \begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix} = 8$ Таким образом, здесь 8 остовных деревьев графа.

Определение остовного дерева минимального веса

Вес остовного дерева взвешенного графа G равен сумме весов, приписанных его рёбрам. Остов минимального веса (или минимальное остовное дерево) – это такой остов графа G, что его вес меньше или равен весу любого другого остова графа.

Задача определения остова минимального веса (кратчайшего остова) имеет множество практических применений. Она возникает, например, при проектировании линий электропередач, дорог, телеграфных линий, авиалиний и т.д., когда требуется заданные центры (вершины) соединить некоторой системой каналов связи (рёбер), так, чтобы два центра были связаны либо непосредственно, либо через другие центры (цепью) и чтобы общая длина (или стоимость, или другой вес) каналов связи была минимальной. Рассмотрим два способа построения минимального остовного дерева взвешенного графа.

Алгоритм Краскала (Крускала).

Идея этого метода следующая: дан граф $G = (V, E)$, будем формировать дерево $T = (V, E_{n-k})$, выбирая рёбра с наименьшим весом так, чтобы не возникало циклов:

1. Строим граф T_1, состоящий из множества вершин V и ребра e_1, которое имеет наименьший вес (т.е. $T_1 = (V, E_1)$, где $E_1 = \{e_1\}$).

2. Если граф T_i уже построен и $i < v^* = n - k$, то строим граф T_{i+1} добавляя к множеству рёбер T_i ребро e_{i+1}, имеющее наименьший вес среди рёбер, не входящих в T_i и не составляющее циклов с рёбрами T_i (т.е. $T_{i+1} = (V, E_{i+1})$, где $E_{i+1} = E_i \cup \{e_{i+1}\}$).

3. При $i = v^* = n - k$ алгоритм закончен и $T_{n-k} = (V, E_{n-k})$ – есть остов минимального веса, его вес равен сумме весов всех его рёбер.

П р и м е р 3.5.5 Дан взвешенный граф (рисунок 3.5.5). Построить остов минимального веса и найти его вес.

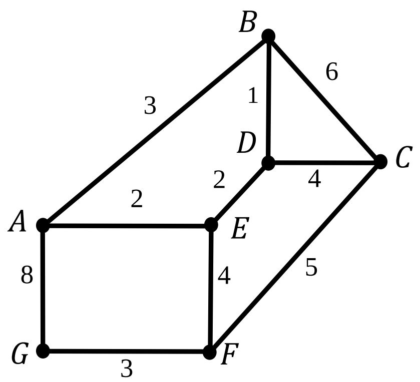


Рисунок 3.5.5

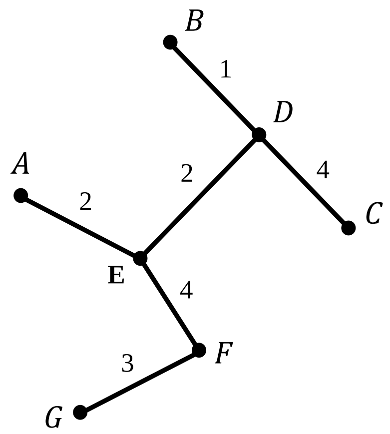


Рисунок 3.5.6

Решение:

а) $T_1 = (V, E_1)$, где $V = \{A, B, C, D, E, F, G\}$, $E_1 = \{\{B, D\}\}$;

б) $T_2 = (V, E_2)$, $E_2 = \{\{B, D\}, \{E, D\}\}$;

в) $T_3 = (V, E_3)$, $E_3 = \{\{B, D\}, \{E, D\}, \{A, E\}\}$;

г) $T_4 = (V, E_4)$, $E_4 = \{\{B, D\}, \{E, D\}, \{A, E\}, \{G, F\}\}$;

д) $T_5 = (V, E_5)$, $E_5 = \{\{B, D\}, \{E, D\}, \{A, E\}, \{G, F\}, \{E, F\}\}$;

е) $T_6 = (V, E_6)$, $E_6 = \{\{B, D\}, \{E, D\}, \{A, E\}, \{G, F\}, \{E, F\}, \{C, D\}\}$.

Число вершин графа $n = 7$, число компонент k=1, число рёбер $m = 10$, коранг $v^* = n - k = 7 - 1 = 6$, значит, остов должен содержать 6 рёбер, удалено $v = m - n + k = 10 - 7 + 1 = 4$ ребра. Алгоритм закончен, построено дерево минимального веса (рисунок 3.5.6).

Его вес равен $1 + 2 + 2 + 4 + 3 + 4 = 16$.

Заметим, что если множество T содержит более одного дерева, то существует ребро, при добавлении которого не возникает цикла – оно соединяет два дерева. Добавленное ребро - кратчайшее возможное (в примере 3.5.5 ребро $\{E, F\}$ соединило два дерева).

Алгоритм Прима (или алгоритм ближайшего соседа).

Принципиальное отличие этого алгоритма от алгоритма Краскала состоит в том, что остовное дерево строится в результате расширения исходного поддерева. По алгоритму Прима можно найти как минимальное, так и максимальное по весу дерево, всё зависит от того, будем ли мы в формуле, записанной ниже, находить минимум или максимум. Алгоритм Прима состоит из двух шагов, выполняющихся $n - 1$ раз для графа, имеющего n вершин.

Пусть дан взвешенный граф $G(V, E)$, $w(v_i, v_j)$ - вес ребра (v_i, v_j). Пусть $\{V', V''\}$ разбиение множества вершин V на два непересекающихся подмножества. Определим пошаговое расстояние между V'и V'' следующим образом: $d(V', V'') = \min \{w(v_i, v_j) \mid v_i \in V', v_j \in V''\}$.

1 шаг. Начинаем с любой вершины, например, v_1. Полагаем

$$V' = \{v_1\}, V'' = V \setminus V',\ E' = \emptyset.$$

2 шаг. Находим ребро (v_i, v_j) такое, что $v_i \in V', v_j \in V''$ и

$$w(v_i, v_j) = d(V', V'') = min\{w(v_i, v_j) \mid v_i \in V', v_j \in V''\}.$$

Полагаем

$$V' = V' \cup \{v_j\}, V'' = V \setminus V', E' = E' \cup \{(v_i, v_j)\}.$$

3 шаг. Если $V' = V$, то $G'(V', E')$- искомый остов, если нет, то переходим ко второму шагу.

Рассмотрим тот же пример 3.5.5, но для решения применим алгоритм Прима: $V = \{A, B, C, D, E, F, G\}$. Начнём с вершины A.

1 шаг. $V' = \{A\}$, $V'' = \{B, C, D, E, F, G\}$, $E' = \emptyset$.

1-ая итерация.

2 шаг. $d(V', V'') = w(A, E) = 2$, $V' = \{A, E\}$, $V'' = \{B, C, D, F, G\}$, $E' = \{(A, E)\}$.

3 шаг. $V' \neq V \;\; \rightarrow \;\; 2$ шаг.

2-ая итерация.

2 шаг. $d(V',V'') = w(E,D) = 2, \quad V' = \{A,E,D\}, \quad V'' = \{B,C,F,G\}, E' = \{(A,E),(E,D)\}$.

3 шаг. $V' \neq V \;\; \rightarrow \;\; 2$ шаг.

3-ая итерация.

2 шаг. $d(V',V'') = w(D,B) = 1, \; V' = \{A,E,D,B\}, \; V'' = \{C,F,G\}$,
$E' = \{(A,E),(E,D),(D,B)\}$.

3 шаг. $V' \neq V \;\; \rightarrow \;\; 2$ шаг.

4-ая итерация.

2 шаг. $d(V',V'') = w(D,C) = 4, V' = \{A,E,D,B,C\}, V'' = \{F,G\}$,
$E' = \{(A,E),(E,D),(D,B),(D,C)\}$

3 шаг. $V' \neq V \;\; \rightarrow \;\; 2$ шаг.

5-ая итерация.

2 шаг. $d(V',V'') = w(E,F) = 4, V' = \{A,E,D,B,C,F\}, V'' = \{G\}$,
$E' = \{(A,E),(E,D),(D,B),(D,C),(E,F)\}$.

3 шаг. $V' \neq V \;\; \rightarrow \;\; 2$ шаг.

6-ая итерация.

2 шаг. $d(V',V'') = w(F,G) = 3, \; V' = \{A,E,D,B,C,F,G\}, \mathrm{V}'' = \emptyset$,
$E' = \{(A,E),(E,D),(D,B),(D,C),(E,F),(F,G)\}$.

3 шаг. $V' = V \;\; \rightarrow G'(V',E')$- остов минимального веса. Его вес $w(G') = 2 + 2 + +1 + 4 + 4 + 3 = 16$.

Алгоритм закончен. Построение остова лучше делать сразу, прибавляя на каждом шаге соответствующее ребро (рисунок 3.5.6).

3.6 Раскраска графов. Планарность

Пусть $G(V,E)$ н-граф без петель. Раскраской графа называется такое приписание цветов (или натуральных чисел) его вершинам, что никакие две смежные вершины не получают одинакового цвета (т.е. если (u,v) ребро, то вершины u и v имеют разные цвета). Хроматическим числом $\chi(G)$ графа G называется минимальное число цветов, требующееся для раскраски G.

Многие практические задачи сводятся к построению раскраски графов: задачи составления расписаний, распределения оборудования, проектирования некоторых технических изделий, раскрашивания географических карт и т.д.

Для некоторых известных графов хроматическое число легко найти, например, $\chi(K_n) = n$, $\chi(\overline{K}_n) = 1$, $\chi(K_{m,n}) = 2$, $\chi(T) = 2$, где K_n– полный граф с n вершинами, $\overline{K}_n$– его дополнение, $K_{m,n}$ – двудольный граф, T - дерево. В общем случае нет формулы, по которой можно было бы вычислить хроматическое число графа. Известны только некоторые оценки этого числа. Наиболее простая оценка имеет вид: $\chi(G) \leq deg(G) + 1$, где $deg(G)$ – максимальная степень вершин графа G.

Поскольку формула для определения хроматического числа неизвестна, то задача нахождения минимальной раскраски труднорешаема. Рассмотрим простой алгоритм построения раскраски графа, который в общем случае не приводит к минимальной раскраске, но приводит к раскраскам, близким к минимальным.

Алгоритм последовательной раскраски.

1. Произвольной вершине v графа G присваиваем цвет 1.
2. Если вершины $v_1, v_2, \ldots, v_i$ раскрашены k цветами 1,2,…,k ($k \leq i$), то новой произвольно взятой вершине v_{i+1} припишем минимальный цвет, не использованный при раскраске вершин из её окружения, т.е. множества вершин v_j таких, что $d(v_{i+1}, v_j) = 1 (j < i)$. Обозначим через $S(v_{i+1})$ окружение вершины v_{i+1}.

П р и м е р 3.6.1 Оценить и найти хроматическое число графа на рисунке 3.6.1. Раскрасить его вершины по алгоритму последовательной раскраски.

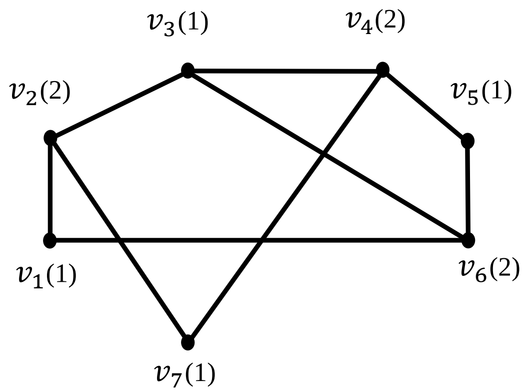


Рисунок 3.6.1

Максимальная степень вершин графа $deg(G) = 3$, отсюда следует, что $\chi(G) \leq 3 + 1 = 4$. Алгоритм последовательной раскраски:

а) выберем вершину v_7 и присвоим ей цвет 1. На рисунке указываем цвет рядом с обозначением вершины. Окружение v_7: $S(v_7) = \{v_2, v_4\}$. Окрасим v_2 и v_4 в цвет 2;

б) выберем вершину v_3, $S(v_3) = \{v_2, v_4, v_6\}$. v_2 и v_4 уже окрашены в цвет 2, v_6 не окрашена. Минимальным цветом, не использованным при раскраске вершин из окружения v_3, является цвет 1, поэтому присваиваем v_3 цвет 1;

в) $S(v_5) = \{v_4, v_6\}$. v_4 окрашена в цвет 2, v_6 не окрашена, поэтому для v_5 выбираем цвет 1;

г) $S(v_6) = \{v_1, v_3, v_5\}$, v_3 и v_5 окрашены в цвет 1, v_1 не окрашена, поэтому присваиваем v_6 цвет 2; д) $S(v_1) = \{v_2, v_6\}$, v_2 и v_6 окрашены в цвет 2, поэтому присваиваем v_1 цвет 1.

Таким образом, получена раскраска графа в два цвета. Эта раскраска минимальна, поэтому $\chi(G) = 2$.

Планарные и плоские графы

н-граф называется плоским, если его вершины – точки на плоскости, а рёбра – плоские линии, и никакие два ребра не имеют общих точек, кроме инцидентной им вершины. Граф, изоморфный плоскому графу, называется планарным. При этом говорят, что планарный граф может быть уложен на плоскости.

П р и м е р 3.6.2 Граф K_4 (рисунок 3.6.2 а)) планарен, т.к. его можно уложить на плоскости. На рисунках 3.6.2 б) и в) изображены два плоских графа, представляющих плоскую укладку K_4.

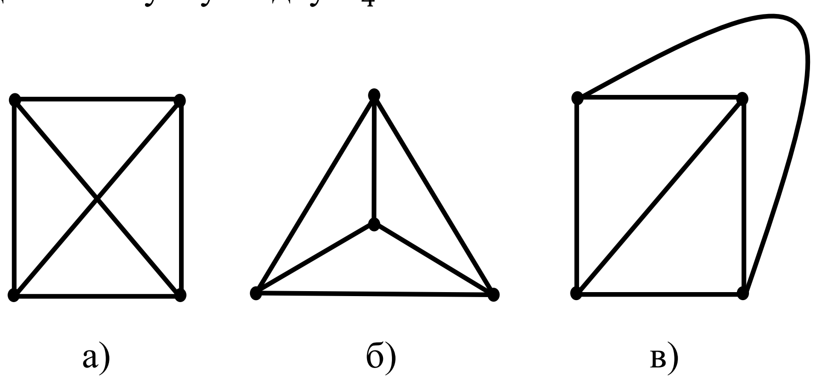


Рисунок 3.6.2

Вопрос возможности укладки графа на плоскости возникает в различных практических задачах, например, при изготовлении интегральных микросхем, которые представляют собой слои миниатюрных микросхем, впечатанных в пластину. При этом важно исключить пересечение проводов. Или при проектировании железнодорожных и других путей, где нежелательны их пересечения и т.д.

Не всякий н-граф является планарным. Для формулировки критерия планарности введём понятие гомеоморфности графов.

Пусть граф $G(V,E)$содержит ребро $\{v_1,v_2\}$, а граф $G'(V',E')$ получен из графа G добавлением новой вершины v в множество V и заменой ребра $\{v_1,v_2\}$ рёбрами $\{v_1,v\}$ и $\{v,v_2\}$. Эта операция называется операцией подразбиения ребра в графе G, а граф G' называется расширением графа G. Два графа G' и G'' называются гомеоморфными, если их можно получить из одного графа G с помощью последовательности подразбиений рёбер.

П р и м е р 3.6.3 На рисунке 3.6.3 графы G' и G'' являются расширением графа G, поэтому эти графы гомеоморфны.

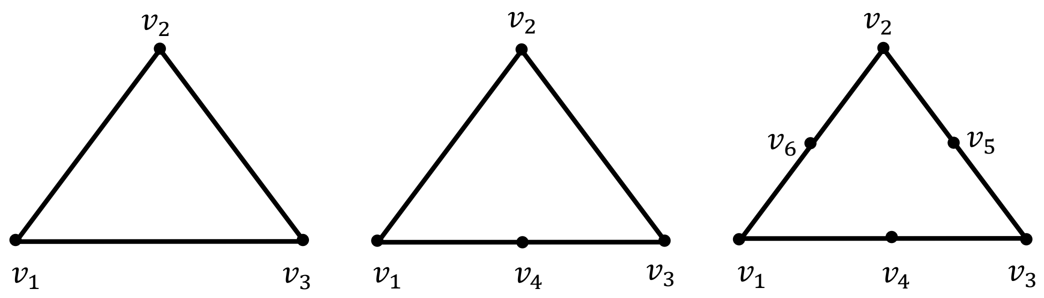


Рисунок 3.6.3

Критерий планарности графов даёт следующая теорема.

Теорема Понтрягина-Куратовского. Граф планарен тогда и только тогда, когда он не содержит подграфа, гомеоморфного K_5 и $K_{3,3}$.

Другая эквивалентная формулировка критерия планарности: граф планарен тогда и только тогда, когда он не содержит подграфов, стягиваемых (т.е. получаемых последовательностью отождествлений вершин, связанных рёбрами) к графу K_5 или $K_{3,3}$.

На практике требуется ответить на вопросы:

а) будет ли граф планарным?

б) если будет, то как уложить его на плоскости?

На первый вопрос отвечает теорема Понтрягина-Куратовского, хотя практически применить её не всегда просто. Ответ на второй можно найти в различных книгах, например, один из методов плоской укладки графа дан в книге С.Д.Шапорева [6], стр. 160.

Если граф не планарен, то, удалив некоторые его рёбра (т.е. перенося их на другую плоскость), можно получить планарный граф. Минимальное число рёбер, которое необходимо удалить из графа для получения его плоского изображения, называется числом планарности графа G (иногда обозначают $sk(G)$). Если после вынесения этих рёбер на вторую плоскость, на ней вновь получают не планарный граф, то удаляют отдельные рёбра на следующую плоскость и т.д. Минимальное число плоскостей m, при котором граф G разбивается на плоские части $G_1, G_2, \ldots, G_m$ называется толщиной графа (толщину иногда обозначают $t(G)$).

П р и м е р 3.6.4 – Толщина любого планарного графа равна 1; число планарности графа K_5 равна $sk(K_5) = 1$. Его толщина $t(K_5) = 2$. Действительно из графа K_5 надо удалить 1 ребро, чтобы получить планарный граф. На рисунке 3.6.4 графы G_1 и G_2 - плоские части графа K_5.

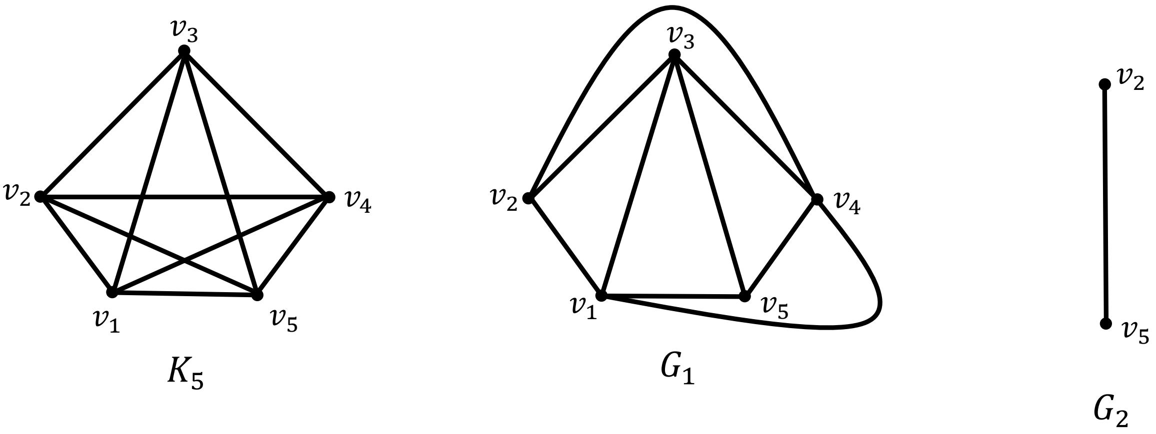


Рисунок 3.6.4

Проблема раскраски плоских графов является одной из самых известных проблем теории графов. Она возникла в связи с раскраской географических карт: любые две соседние страны должны быть окрашены в различные цвета. Известно, что для раскрашивания карты пяти красок достаточно, трёх – нет. Отсюда возникла гипотеза четырёх красок: всякий планарный граф 4-раскрашиваем. На протяжении многих лет проблема четырёх красок оставалась нерешённой. Только в 1890 году Р.Д. Хивуд смог доказать, что произвольную плоскую карту можно раскрасить только пятью цветами, т.е. была доказана теорема: всякий планарный граф можно раскрасить пятью красками.

4 Элементы комбинаторики

4.1 Правила суммы и произведения

Комбинаторика – раздел математики, в котором решаются задачи построения подмножеств некоторых, чаще всего конечных, множеств. Эти подмножества (или выборки, или комбинации, или комбинаторные конфигурации) строятся в соответствии с определёнными правилами. Целью комбинаторики является изучение условий существования комбинаторных конфигураций, алгоритмов их построения, оптимизации таких алгоритмов.

Вот современные задачи, решаемые комбинаторными методами:

а) перечислительные задачи отвечают на вопрос о количестве комбинаторных конфигураций;

б) задачи о маршрутах – отыскание оптимального плана, например, кратчайшего пути;

в) комбинаторные задачи теории графов – задачи сетевого планирования, задача окраски графов, и т. д.

Простейшими примерами комбинаторных конфигураций являются перестановки, размещения, сочетания и разбиения. При их подсчёте используются следующие правила.

1. Правило суммы.

Если из некоторого конечного множества подмножество A (может содержать 1 элемент) можно выбрать n способами, а подмножество B ($A \neq B$) другими m способами, то выбор A или B можно получить $n + m$ способами.

Это правило можно сформулировать и в терминах теории множеств: если мощности множеств (число элементов конечного множества) соответственно равны $|A| = n$, $|B| = m$, то $|A \cup B| = n + m - |A \cap B|$; в случае, когда $A \cap B = \emptyset$ имеем $|A \cup B| = n + m$.

2. Правило произведения.

Если из некоторого конечного множества подмножество A (которое может содержать один элемент) можно выбрать n способами и после каждого такого выбора подмножество B можно выбрать m способами, то последовательный выбор A и B можно осуществить $n \cdot m$ способами.

В терминах теории множеств: если $|A| = n$, $|B| = m$, то $|A \times B| = n \cdot m$, где $A \times B$ - прямое произведение множеств A и B.

Пример - 4.1.1. В классе 14 девочек и 10 мальчиков. Сколькими способами можно выбрать двух учеников одного пола?

Решение: по правилу произведения двух девочек можно выбрать $14 \cdot 13 = 182$ способами, а двух мальчиков $10 \cdot 9 = 90$ способами. Так как следует выбрать двух учеников одного пола, т.е. или двух девочек или двух мальчиков, то по правилу суммы таких способов выбора будет 182 + 90 = 272.

4.2 Перестановки, размещения и сочетания

Рассмотрим задачи, связанные с подсчётом числа способов построения различных комбинаторных конфигураций из элементов конечного множества.

Определение. Пусть дано конечное множество A, состоящее из n элементов ($|A| = n$). Перестановкой элементов множества A называется любой кортеж (т.е. упорядоченное множество), состоящий из n элементов этого множества

Таким образом, перестановки отличаются друг от друга только порядком входящих в них элементов.

В некоторых учебниках (см. [3], с.87) перестановку определяют как взаимно однозначную функцию $f: A \to A$. Если $|A| = n$, то можно считать, что $A = \{1,2,\dots,n\}$, и тогда перестановку удобно задавать таблицей, например, $f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 5 & 2 \end{pmatrix}$, $(n = 5)$. Эта таблица называется подстановкой, по сути подстановки и перестановки одно и то же. Каждой такой подстановке можно поставить в соответствие граф. Так подстановке f, приведённой выше соответствует граф на рисунке 4.2.1.

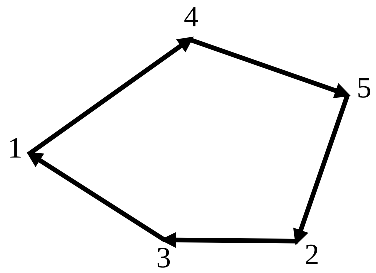


Рисунок 4.2.1

Число перестановок из n элементов обозначается P_n. Легко показать, что $P_n = n!$ (n факториал, $n! = 1 \cdot 2 \dots (n-1) \cdot n$, по определению $0! = 1, 1! = 1$).

Действительно, на первое место в кортеже можно поставить любой из n элементов, на вторе – любой из оставшихся $(n-1)$ - го, на третье – любой из $(n-2)$ оставшихся и т.д. Для последнего места останется только один элемент. Поэтому $P_n = n \cdot (n-1) \cdot (n-2) \dots 2 \cdot 1 = n!$.

П р и м е р 4.2.1 Составить различные перестановки из элементов множества $A = \{a, b, c\}$; подсчитать их число.

Решение:

$|A| = n = 3$. По выше - приведённой формуле: $P_3 = 3! = 1 \cdot 2 \cdot 3 = 6$. Действительно, имеем следующие перестановки: $(a,b,c), (c,a,b), (b,c,a)$, $(b,a,c), (c,b,a), (a,c,b)$.

Определение. Пусть множество A состоит из n элементов ($|A| = n$) и $m \leq n$. Размещением из n элементов по m называется любой кортеж, состоящий из m попарно различных элементов множества A.

Таким образом, размещения отличаются друг от друга либо самими элементами, либо порядком их расположения.

Число размещений из n элементов по m обозначается A_n^m. Доказано, что

$$A_n^m = \frac{n!}{(n-m)!} = n(n-1)(n-2)\ldots(n-m+1).$$

Действительно, для составления каждого размещения нужно выбрать m элементов из n и упорядочить их. На первое место можно поставить любой из имеющихся n элементов (т.е. заполнить первое место можно n способами), на второе место – любой из оставшихся $(n-1)$ элементов, и т.д. После заполнения мест с первого по $(m-11)$ останется $n-(m-1) = n-m+1$ элементов, из которых выбирается элемент на последнее место. Таким образом, по правилу произведения существует $n(n-1)(n-2)\ldots(n-m+1)$ способов выбора m элементов из n, т.е.

$$A_n^m = n(n-1)(n-2)\ldots(n-m+1).$$

Пример 4.2.2 Составить различные размещения из элементов множества $A = \{a, b, c\}$ по два; подсчитать их число.

Решение:

$|A| = n = 3$, $m = 2$, $A_3^2 = \frac{3!}{(3-2)!} = 3 \cdot 2 = 6$. Действительно, таких размещений шесть: $(a, b), (b, a), (a, c), (c, a), (b, c), (c, b)$.

Определение. Пусть множество A состоит из n элементов ($|A| = n$) и $m \leq n$. Сочетанием из n элементов по m называется любое подмножество множества A, состоящее из m элементов.

Таким образом, сочетания отличаются друг от друга хотя бы одним элементом. Число сочетаний из n элементов по m обозначается C_n^m. Легко показать, что

$$C_n^m = \frac{A_n^m}{P_m} = \frac{n(n-1)\ldots(n-m+1)}{1 \cdot 2 \ldots m} = \frac{n!}{m!(n-m)!}.$$

Действительно, число размещений из n элементов по m (A_n^m) можно найти так: выбрать m элементов из n элементов множества A, что возможно сделать C_n^m способами (по определению сочетаний); затем в каждом из полученных сочетаний, содержащих m элементов, произвести все возможные перестановки. Это можно сделать P_m способами. Тогда по правилу произведения $A_n^m = C_n^m \cdot P_m$, откуда $C_n^m = \frac{A_n^m}{P_m}$.

Полезно знать некоторые свойства числа C_n^m:

1) $C_n^m = C_n^{n-m}$;

2) $C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$;

3) $C_0^0 = C_n^0 = C_n^n = 1$, $C_n^1 = n$;

4) Числа C_n^m часто называют биномиальными коэффициентами. Действительно, формула бинома Ньютона имеет вид

$$(a+b)^{\mathrm{n}} = C_n^0 a^n b^0 + C_n^1 a^{n-1} b^1 + \cdots + C_n^n a^0 b^n.$$

Из этой формулы легко получить ещё одно свойство чисел C_n^m: при $a = b = 1$ формула бинома Ньютона превращается в равенство $C_n^0 + C_n^1 + \cdots + C_n^n = 2^n$.

Пример 4.2.3 Составить различные сочетания из элементов множества $A = \{a, b, c\}$ по два; подсчитать их число.

Решение:

$|A| = n = 3, m = 2.$ $C_3^2 = \frac{3 \cdot 2}{1 \cdot 2} = 3.$

Действительно, таких сочетаний три: $(a, b), (a, c), (b, c)$.

4.3 Размещения, сочетания и перестановки с повторениями

При выборе m элементов из n для составления размещений и сочетаний элементы могут не возвращаться в исходное множество (как было выше), а могут возвращаться. В последнем случае мы имеем размещения и сочетания с повторениями.

Определение. Пусть множество A состоит из n элементов ($|A| = n$) и $m \leq n$. Размещением с повторениями из n элементов множества A по m называется кортеж, состоящий из m элементов множества A.

Таким образом, размещения с повторениями могут отличаться друг от друга либо элементами, либо порядком их расположения, либо количеством повторений элементов.

Число размещений с повторениями из n элементов по m обозначается $\overline{A}_n^m$. Легко доказать, что $\overline{A}_n^m = n^m$. Действительно, пусть множество A состоит из n элементов и $(a_1, a_2, \ldots, a_m)$ один из его кортежей (размещений с повторениями). Каждый элемент этого размещения может повторяться в других размещениях, т.е. на каждое место кортежа может претендовать любой из n элементов множества A. Поэтому число размещений из n элементов по m с повторениями $\overline{A}_n^m = \underbrace{n \cdot n \ldots n}_{m} = n^m$.

П р и м е р 4.3.1 Составить размещения с повторениями из элементов множества $A = \{a, b, c\}$ по два; подсчитать их число.

Решение:

$|A| = n = 3, \quad m = 2. \quad \overline{A}_3^2 = 3^2 = 9$. Действительно, это следующие размещения: $(a, a), (a, b), (a, c), (b, b), (b, a), (b, c), (c, c), (c, a), (c, b)$.

Определение. Сочетанием с повторениями из n элементов множества A по m называется любое подмножество множества A, состоящее из m элементов, причём один и тот же элемент может повторяться.

Таким образом, сочетания с повторениями отличаются друг от друга хотя бы одним элементом, но в каждом из них любой элемент может повторяться.

Число сочетаний с повторениями из n элементов по m обозначается $\overline{C}_n^m$. Доказано, что $\overline{C}_n^m = C_{n+m-1}^m$.

П р и м е р 4.3.2 Составить сочетания с повторениями из элементов множества $A = \{a, b, c\}$ по два; подсчитать их число.

Решение:

$|A| = n = 3, \quad m = 2. \quad \overline{C}_3^2 = C_{3+2-1}^2 = C_4^2 = \frac{4\cdot 3}{1\cdot 2} = 6$. Это следующие сочетания: $(a, a), (a, b), (a, c), (b, b), (b, c), (c, c)$.

Определение. Пусть во множестве с n элементами есть m различных элементов. Причём первый элемент повторяется n_1 раз, второй - n_2 раза и т.д., m - ый повторяется n_m раз: $n_1 + n_2 + \cdots + n_{\mathrm{m}} = n$.

Перестановки из n элементов данного множества называются перестановками с повторениями из n элементов.

Число перестановок с повторениями из n элементов обозначается $P_n(n_1, n_2, \dots, n_m)$. Доказано, что

$$P_n(n_1, n_2, \dots, n_m) = \frac{n!}{n_1! n_2! \dots n_m!}.$$

П р и м е р 4.3.3 Сколько различных четырёхзначных чисел можно составить из цифр 1,1,2,4.

Решение.

Данное множество состоит из четырёх цифр ($n = 4$), среди которых 1 повторяется два раза: $n_1 = 2$; 2 – один раз: $n_2 = 1$, 4 - один раз: $n_3 = 1$. По выше приведённой формуле число четырёхзначных чисел из цифр 1,1,2,4 равно $P_4(2,1,1) = \frac{4!}{2!\cdot 1!\cdot 1!} = \frac{1\cdot 2\cdot 3\cdot 4}{1\cdot 2} = 12$.

4.4 Разбиения

Пусть множество A состоит из n элементов, $|A| = n$. Пусть $\{A_1, A_2, \dots, A_k\}$ - разбиение множества A на k подмножеств (определение разбиений см. на стр.6), $|A_i| = m_i$, $\sum_{i=1}^{k} m_i = n$.

Назовём кортеж $(A_1, A_2, \dots, A_k)$ упорядоченным разбиением множества A на k подмножеств.

Обозначим число упорядоченных разбиений $(A_1, A_2, \dots, A_k)$ так: $C(n; m_1, m_2, \dots, m_k)$, где $|A_i| = m_i$, $i = 1,2, \dots, k$. Доказано, что:

$$C(n; m_1, m_2, \dots, m_k) = \frac{n!}{m_1! \cdot m_2! \ldots m_k!}.$$

П р и м е р 4.4.1 В общежитии остались не заселёнными 3 комнаты на 6, 4 и 2 человека. Сколькими способами можно расселить по этим комнатам 12 студентов?

Решение:

$$n = 6 + 4 + 2 = 12, \quad m_1 = 6; \quad m_2 = 4, \quad m_3 = 2.$$

По выше приведённой формуле число способов расселения равно: $C(12; 6,4,2) = \frac{12!}{6! \cdot 4! \cdot 2!} = 13860$.

Понятие производящей функции

При решении комбинаторных задач одним из самых развитых теоретических и практических методов является метод производящих функций. Он предполагает использование некоторых разделов математического анализа, в частности теории функциональных рядов.

Идея этого метода в следующем: пусть дана некоторая числовая последовательность $\{a_i\} = a_0, a_1, \dots, a_n, \dots$ и последовательность функций:

$$\{\varphi_i(x)\} = \varphi_0(x), \varphi_1(x), \dots, \varphi_n(x), \dots$$

Составим формально ряд

$$F(x) = \sum_i a_i \varphi_i(x) = a_0 \varphi_0(x) + a_1 \varphi_1(x) + \dots + a_{\mathrm{n}} \varphi_{\mathrm{n}}(x) + \cdots$$

Функция $F(x)$ называется производящей функцией для последовательности $\{a_i\}$ относительно заданной последовательности $\{\varphi_i(x)\}$.

Часто используют $\varphi_i(x) = x^i$ или $\varphi_i(x) = x^i/_{i!}$.

Производящими функциями во многих случаях оперировать удобнее и проще. Они позволяют определить такие свойства последовательности $\{a_i\}$, которые другими способами получить затруднительно. Производящие функции, и не только они, а вообще комбинаторные методы, широкое применение имеют в теории вероятностей, в той её части, где рассматривают случайные события с конечным числом исходов. Рассмотрим несколько примеров, иллюстрирующих понятие производящей функции и её применение в теории вероятностей.

Пример 4.4.2 – По формуле бинома Ньютона имеем

$$(1+x)^n = \sum_{i=0}^{n} C_n^i\, x^i \, .$$

Поэтому для последовательности биномиальных коэффициентов $\{C_n^i\} = C_n^0, C_n^1, \dots, C_n^n$ производящей функцией будет

$$F(x) = (1+x)^n.$$

Пример 4.4.3 Пусть $\{a_i\} = \{a^i\}$, $\{\varphi_i(x)\} = \{x^i\}$, $i = 0,1,2,\dots$ Тогда

$$\sum_i a_i \varphi_i(x) = \sum_{i=0}^{\infty} a^i x^i = 1 + ax + a^2x^2 + \dots + a^n x^n + \dots$$

- это сумма членов бесконечной геометрической прогрессии со знаменателем $q = ax$. Если $|q| == |ax| < 1$, то последний ряд сходится и его сумма равна:

$$S = \frac{1}{1-ax}.$$

Таким образом,

$$\frac{1}{1-ax} = 1 + ax + a^2x^2 + \dots + a^n x^n + \dots,$$

поэтому функция $F(x) = \frac{1}{1-ax}$ будет производящей для последовательности $\{a^i\} = 1, a, a^2, \dots$.

Пример 4.4.4 Пример из теории вероятностей: пусть производится n независимых испытаний, в каждом из которых вероятность появления события A разная: в первом - p_1, во втором - p_2, …, в n -ом - p_n; соответственно $q_1, q_2, \dots, q_n$ - вероятности не появления события A; $P_n(k)$ - вероятность того, что в n испытаниях события A появится ровно k раз. Заметим, что вероятность $P_n(k)$ находится по формуле Бернулли, если в каждом испытании вероятность появления события A одинаковая. В нашем случае эта вероятность определяется с помощью производящей функции. Производящей функцией вероятностей $P_n(k)$ называют функцию $\varphi_n(z) = (p_1 z + q_1)(p_2 z +$

$+q_2)\ldots(p_n z+q_n)$. Вероятность $P_n(k)$ равна коэффициенту при z^k в разложении производящей функции по степеням z. Например, при $n=2$ имеем

$$\varphi_2(z)=(p_1 z+q_1)(p_2 z+q_2)=p_1 p_2 z^2+(p_1 q_2+p_2 q_1)z+q_1 q_2.$$

Коэффициент при z^2 равен вероятности того, что событие A появится два раза в двух испытаниях:

$$P_2(2)=p_1 p_2;$$

при z^1 - вероятности того, что событие A появится один раз в двух испытаниях:

$$P_2(1)=p_1 q_2+p_2 q_1;$$

при z^0 (т.е. свободный член) - вероятности того, что событие A не появится ни разу в двух испытаниях:

$$P_2(0)=q_1 q_2.$$

Эту функцию можно использовать и в случае, если в различных испытаниях появляются различные события: в первом испытании событие A_1, во втором – A_2, и т.д. Тогда изменяется только истолкование коэффициентов при различных степенях z. Так, в приведённом выше разложении коэффициент при z^2- $p_1 p_2$ будет определять вероятность появления одновременно двух событий A_1 и A_2.

В теории вероятностей производящую функцию можно определить по - другому, если она нужна для других целей. Например, для нахождения числовых характеристик дискретных случайных величин с целыми неотрицательными значениями, производящую функцию определяют так:

$$\varphi(x)=p_0+p_1 x+p_2 x^2+\ldots+p_n x^n+\ldots, \text{ где } p_1, p_2, \ldots, p_n, \ldots$$

соответственно вероятности возможных значений 0,1,2,3,…случайной величины X. Легко показать, что математическое ожидание равно:

$$M(X)=\varphi'(1);$$

Дисперсия:

$$D(X)=\varphi''(1)+\varphi'(1)-(\varphi'(1))^2.$$

5 Элементы теории кодирования

5.1 Предмет теории кодирования

Кодирование – это система представления информации. Например:

1) прямоугольная декартова система координат – это способ представления (кодирования) геометрических объектов числами, т.е. точке соответствует пара чисел – её координаты, линии – её уравнение и т.д.;

2) десятичная система счисления – это способ представления (кодирования) чисел, когда каждому, например, натуральному числу $n \in N$ ставится в соответствие последовательность $a_m a_{m-1} \ldots a_1 a_0$, такая, что:
$n = a_m \cdot 10^m + a_{m-1} \cdot 10^{m-1} + \ldots + a_1 \cdot 10^1 + a_0 \cdot 10^0$, $a_i \in \{0,1,,2,\ldots,9\}$, $i = \overline{0,1,\ldots,m}$; другой способ кодирования чисел – римские цифры;

3) в военных целях секретную информацию также кодируют, её шифруют и дешифруют. Здесь возникает проблема защиты своих кодов и взламывания чужих. Этими вопросами занимается раздел теории кодирования – криптология.

Таким образом, кодирование издавна широко применялось в деятельности человека и, если раньше оно играло вспомогательную роль, то с появлением компьютеров кодирование стало рассматриваться как отдельный раздел математики. Оно стало центральным вопросом при решении практически всех задач программирования. Например:

1) представление данных любой природы в памяти компьютера;
2) защита информации от доступа посторонних лиц;
3) сжатие информации в базе данных и т.д.

Математическое задание, основные понятия и свойства кодирования

Пусть заданы два конечных множества некоторых символов $A = \{a_1, a_2, \ldots, a_n\}$ и $B = \{b_1, b_2, \ldots, b_m\}$, называемых алфавитами, их элементы называют буквами. Последовательность букв из алфавитов, например, $\alpha = a_{i1} a_{i2} \ldots a_{ik}$ из алфавита A называется словом в A, а число k – длиной слова (обозначается $|\alpha| = k$ или $k = l(\alpha)$); если $k = 0$, то слово называется пустым, обозначается Λ или ε. Множество непустых слов в алфавите A обозначим A^*, в алфавите B - B^*.

Пусть определена функция или отображение $f: A^* \to B^*$, область определения $f: D(f) = S,\ S \subset A*$. Слова из S называются сообщениями, слово $\beta = f(\alpha)$ называется кодом сообщения α, где $\alpha \in S, \beta \in B^*$, а процесс перехода от слова α к слову β, т.е. функция f, называется кодированием. Процесс обратного преобразования слова $\beta \in B^*$ в слово $\alpha \in A^*$, т.е. обратная функция f^{-1} (если она существует) называется декодированием. Если мощность множества B равна m ($|B| = m$), то f называется m–ичным кодированием. Наиболее распространён случай $B = \{0,1\}$, т.е. двоичное кодирование.

Формулировка задачи кодирования: при данных алфавитах A и B и множества сообщений S найти такое кодирование f, которое обладает определёнными свойствами.

Основные свойства, предъявляемые кодированию:

1) существование декодирования, т.е. возможность по коду β сообщения α восстановить α; или однозначность кодирования, т.е., если $\alpha_1 \neq \alpha_2$, то $f(\alpha_1) \neq f(\alpha_2)$;

2) помехоустойчивость или исправление ошибок, т.е. использование кодов, исправляющих или обнаруживающих ошибки, которые могут возникнуть при передаче информации;

3) заданная сложность или простота кодирования и декодирования. Например, в криптографии, которая занимается созданием и раскрытием шифров, шифр считается достаточно надёжным, если раскрыть его гораздо труднее, чем создать.

Для сформулированной выше задачи кодирования большое значение имеет природа множества сообщений S. При одних и тех же алфавитах A и B и кодировании f применяются разные методы для описания S: например, $S = \{\alpha | \alpha \in \mathrm{A} * \wedge |\alpha| = n\}$, т.е. теоретико-множественное описание; вероятностное описание, например, $S = \mathrm{A}*$ и заданы вероятности p_i появления букв a_i в сообщении, причём $\sum_i^n p_i = 1$ и т.д.

Существуют различные виды кодов, т.е. способов кодирования (алфавитное кодирование, оптимальное, помехоустойчивое, коды Хемминга, линейные коды, блочное кодирование и т.д.). Выбор способа кодирования зависит от требований, предъявляемых кодированию в данной задаче.

Рассмотрим самое простое - алфавитное кодирование, которое всегда можно ввести на непустых алфавитах.

5.2 Алфавитное кодирование

Кодирование f можно осуществить по разному: можно рассматривать множество сообщений S как единое целое и сопоставить ему некоторый код, или можно разбить S на части и каждую часть кодировать. Самый простой случай – это когда элементарной частью сообщения является одна буква алфавита A. Такое кодирование называется алфавитным (или побуквенным).

Алфавитное кодирование задаётся схемой (таблицей), которая обозначается Σ или σ и записывается: $\Sigma = \langle a_1 \to \beta_1, \dots, a_n \to \beta_n \rangle = \langle a_i \to \beta_i \rangle_{i=1}^{n}$, где $a_i \in A,\ \beta_i \in B^*,\ i = 1,2,\dots,n$ или

$\Sigma =$

a_1	a_2	…	a_n
β_1	β_2	…	β_n

Множество $V = \{\beta_1, \beta_2, \dots, \beta_n\}$ кодов букв алфавита A называется множеством элементарных кодов, а слова $\beta_1, \beta_2, \dots, \beta_n$ называются элементарными кодами или кодовыми словами.

Таким образом, любое слово кодируется следующим образом:

$$f\colon A^* \to B^*,\ \alpha = a_{i1}a_{i2}\dots a_{ik} \in A^*,$$

$$f(\alpha) = f(a_{i1})f(a_{i2}) \dots f(a_{ik}) = \beta_{i1}\beta_{i2}\dots\beta_{ik} = \beta \in B^*.$$

Одним из основных вопросов в кодировании является проблема взаимной однозначности, т. е. возможность по коду β сообщения α однозначно восстановить α. Введём несколько определений:

а) Если $\beta = \beta'\beta_{i1}\beta_{i2}\dots\beta_{ik}\beta''$, то β' называется началом или префиксом слова β, а β'' - окончанием или постфиксом слова β; два слова можно соединять, для этого префикс второго слова должен следовать за постфиксом первого, при этом в новом слове они утрачивают свой статус, если только одно из слов не было пустым.

б) Схема алфавитного кодирования Σ называется обладающей свойствами префикса (или префиксной), если ни один элементарный код не является префиксом другого элементарного кода, например:

1) схема

$\Sigma =$

a_1	a_2	a_3
01	011	10

не является префиксной, т.к. элементарный код $\beta_1 = 01$ является префиксом элементарного кода $\beta_2 = 011$;

2) схема

$\Sigma =$

a_1	a_2	a_3
01	1101	10

обладает свойством префикса, т.к. ни один элементарный код не является префиксом другого элементарного кода.

в) Схема алфавитного кодирования Σ называется разделимой, если любое слово, составленное из элементарных кодов, разлагается на элементарные коды единственным образом, т.е. если $\beta_{i1}\beta_{i2}\ldots\beta_{ik}=\beta_{j1}\beta_{j2}\ldots\beta_{jl}$, то $k=l$ и $\forall t\in\overline{1\ldots k}(i_t=j_t)$.

Доказано:

1) префиксная схема является разделимой, причём, условие префиксности является достаточным, но не необходимым для разделимости схемы;

2) алфавитное кодирование с разделимой схемой допускает декодирование;

3) алфавитное кодирование с префиксной и разделимой схемой допускает взаимно-однозначное кодирование (свойство достаточное, но не необходимое).

5.3 Неравенство Макмиллана

Теорема 1. Если схема $\Sigma=\langle a_i\to\beta_i\rangle_{i=1}^{n}$ разделима, то

$$\sum_{i=1}^{n}\frac{1}{2^{l_i}}\le 1, \qquad (*)$$

где $l_i=|\beta_i|$ - длина слова β_i; неравенство (*) называется неравенством Макмиллана.

Теорема 2. Если числа $l_1,l_2,\ldots,l_n$ удовлетворяют неравенству

$$\sum_{i=1}^{n}\frac{1}{2^{l_i}}\le 1,$$

то существует такая разделимая (даже префиксная) схема алфавитного кодирования $\Sigma=\langle a_i\to\beta_i\rangle_{i=1}^{n}$, что $\forall i(|\beta_i|=l_i)$.

Следствие теорем 1 и 2.

Если схема алфавитного кодирования $\Sigma=\langle a_i\to\beta_i\rangle_{i=1}^{n}$ разделима, то существует префиксная схема $\Sigma'=\langle\alpha_i\to\beta_i'\rangle_{i=1}^{n}$, причём $\forall i(|\beta_i|=|\beta_i'|)$.

Таким образом, неравенство Макмиллана является необходимым условием разделимости схемы (даже в некотором смысле достаточным, т.к. из теорем 1 и 2 следует существование разделимой схемы с заданными длинами слов).

Пример 5.3.1 Пусть $A=\{0,1,2,3,4,5,6,7,8,9\}$, $B=\{0,1\}$.

а) $\Sigma_2=$

0	1	2	3	4	5	6	7	8	9
0	1	10	11	100	101	110	111	1000	1001

Σ_1 - не префиксная схема, т.к. элементарный код, например, 2 ($\beta_3=10$) является префиксом элементарных кодов 4, 5, 8, 9; для Σ_1 неравенство Макмиллана не выполняется:

$$\frac{1}{2^1}+\frac{1}{2^1}+\frac{1}{2^2}+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^4}+\frac{1}{2^4}=1+\frac{1}{2}+\frac{4}{8}+\frac{2}{16}>1,$$

где $l_1 = |\beta_1| = 1$, $l_2 = |\beta_2| = 1$, $l_3 = |\beta_3| = 2, \ldots, l_{10} = |\beta_{10}| = 4$; схема Σ_1 не префиксная и не разделимая, следовательно, однозначное кодирование невозможно, действительно, например, для двух разных слов $\alpha_1 = 77$ и $\alpha_2 = 333$ существует один код $\beta_1 = \beta_2 = 111111$.

б) $\Sigma_2 =$

0	1	2	3	4	5	6	7	8	9
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Σ_2 - префиксная схема, т.к. ни один элементарный код не является префиксом другого элементарного кода, следовательно она разделимая, неравенство Макмиллана выполняется: $\frac{1}{2^4} \cdot 10 = \frac{10}{16} < 1$; итак, схема Σ_2 префиксная и разделимая, поэтому код любого слова однозначно декодируется, например: $\alpha_1 = 77 \leftrightarrow \beta_1 = 01110111$; $\alpha_2 = 333 \leftrightarrow \beta_2 = 001100110011$.

П р и м е р 5.3.2

$\Sigma_3 =$

a_1	a_2
0	01

Σ_3- не префиксная схема, т.к. элементарный код $\beta_1 = 0$ является префиксом элементарного кода $\beta_2 = 01$; для Σ_3- неравенство Макмиллана выполняется: $\frac{1}{2^1} + \frac{1}{2^2} = \frac{3}{4} < 1$ и схема будет разделимой. По следствию из теорем 1 и 2 для неё можно построить префиксную и разделимую схему Σ_3':

$\Sigma_3' =$

a_1	a_2
0	10

П р и м е р 5.3.3

$\Sigma_4 =$

a_1	a_2	a_3
00	100	001

Эта схема не префиксная ($\beta_1 = 00$ является префиксом $\beta_3 =$ 001), неравенство Макмиллана выполняется: ($\frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} = \frac{1}{2} < 1$), но схема не разделима. Коды некоторых слов неоднозначно декодируются. Например, $\alpha = a_1 a_2 a_2 \quad \to \beta = \beta_1 \beta_2 \beta_2 = (00)(100)(100) = 00100100$. Слово β можно декодировать и по – другому, т.к. его можно представить в виде соединения элементарных кодов в другом порядке:

$\beta = 00100100 = (001)(001)(00) = \beta_3\beta_3\beta_1$ или $\beta = 00100100 =$

$= (001)(00)(100) = \beta_3\beta_1\beta_2$.

5.4 Оптимальное кодирование (или кодирование с минимальной избыточностью)

Для практики важно, чтобы коды сообщений имели по возможности наименьшую длину. Например, в компьютере все буквы и другие символы хранятся в виде строк из 0 и 1, поэтому всегда желательно использовать для их хранения как можно меньше места. Очевидно, это станет возможным, если более короткие строки хранили бы часто используемые символы, менее короткие – редко используемые.

Пусть дана разделимая схема алфавитного кодирования:

$$\Sigma = \langle a_i \to \beta_i \rangle_{i=1}^{n}.$$

Длины элементарных кодов β_i в общем случае различны, поэтому длина кода β сообщения $\alpha \to \beta = \beta_{i1}\beta_{i2}\ldots\beta_{ik}$ зависит от состава букв в сообщении α и от того, каким буквам какие элементарные коды приписаны.

Пусть $k_1, k_2, \ldots, k_n$ количество вхождений (т.е. частота) соответственно букв $a_1, a_2, \ldots, a_n$ в сообщении $s \in S$, а $l_1, l_2, \ldots, l_n$ длины их элементарных кодов $\beta_1, \beta_2, \ldots, \beta_n$. Тогда, если $k_i \le k_j$ и $l_i \ge l_j$, то

$$k_i l_i + k_j l_j \le k_i l_j + k_j l_i \text{ (*)}.$$

Действительно, пусть $k_i = k$, $k_j = k + a$ (т.е. $k_i \le k_j$), а $l_i = l + b$, $l_j = l$ (т.е. $l_i \ge l_j$), $a \ge 0, b \ge 0$.

Тогда разность

$$(k_i l_j + k_j l_i) - (k_i l_i + k_j l_j) = [kl + (k + + a)(l + b)]-$$
$$-[k(l + b) + (k + a)l] = \ldots = ab \ge 0,$$

откуда следует неравенство (*).

Из этого рассуждения вытекает принцип приписания буквам алфавита сообщений элементарных кодов, при котором длина кода конкретного сообщения будет минимальной: часто встречающимся буквам должен соответствовать более короткий код.

Известно, что из разделимой схемы $\Sigma = \left\langle a_i \to \beta_i \right\rangle_{i=1}^{n}$ всегда можно получить разделимую и префиксную схему $\Sigma' = \left\langle a_i \to \beta_i' \right\rangle_{i=1}^{n}$, где $\beta_1', \beta_2', \ldots, \beta_n'$ имеют соответственно ту же длину, что и $\beta_1, \beta_2, \ldots, \beta_n$ (см. следствие теорем 1 и 2). При этом должно учитываться следующее: длина кода сообщения $s \in S$ будет минимальной, если переставить буквы s в порядке убывания их частот, а элементарные коды переставить в порядке возрастания их длины, и приписать буквам коды в этом порядке.

Цена (длина) кодирования

Пусть заданы алфавит $A = \{a_1, a_2, \dots, a_n\}$ и вероятности появления букв в сообщении $P = \langle p_1, p_2, \dots, p_n \rangle$ (p_i - вероятность появления буквы $a_i, i = \overline{1 \dots n}$). Полагаем, что буквы упорядочены по убыванию вероятности их появления, т.е. $p_1 \geq p_2 \geq \dots \geq p_n > 0$ и буквы с нулевой вероятностью исключаются т.е. $p_1 + p_2 + \dots + p_n = 1$. Заметим, что здесь рассматривается другой подход к минимизации кодов сообщений – вместо частот букв алфавита A рассматривают вероятности их появления.

Определение. Средней ценой (или длиной) разделимой схемы алфавитного кодирования $\Sigma = \left\langle a_i \to \beta_i \right\rangle_{i=1}^{n}$ называется математическое ожидание длины закодированного сообщения (обозначается l_Σ):

$$l_\Sigma = \sum_{i=1}^{n} p_i l_i, \text{ где } l_i = |\beta_i|.$$

П р и м е р 5.3.4. Пусть заданы алфавиты $A = \{a_1, a_2\}$, $B = \{0,1\}$ и разделимая схема

$\Sigma =$

a_1	a_2
0	10

$l_1 = |\beta_1| = 1, \quad l_2 = |\beta_2| = 2.$

При распределении вероятностей $P_1 = \langle 0{,}5; 0{,}5 \rangle$ цена кодирования равна $l_\Sigma = 0{,}5 \cdot 1 + 0{,}5 \cdot 2 = 1{,}5$; при распределении вероятностей $P_2 = \langle 0{,}9; 0{,}1 \rangle$ цена кодирования равна $l_\Sigma = 0{,}9 \cdot 1 + 0{,}1 \cdot 2 = 1{,}1$. Таким образом, если менее длинным кодам букв соответствует большая вероятность появления этих букв, то цена кодирования будет меньше.

Если вместо вероятностей появления букв алфавита A рассматривать их частоты, то можно определить аналогичное цене кодирования понятие – понятие веса кодирования. Пусть буквы алфавита $A = \{a_1, a_2, \dots, a_n\}$ появляются соответственно с частотами $k_1, k_2, \dots, k_n$. Рассмотрим схему алфавитного кодирования $\Sigma = \langle a_i \to \beta_i \rangle_{i=1}^{n}$, причём $l_i = |\beta_i|$, тогда:

$$\omega = \sum_{i=1}^{n} k_i l_i$$

называется весом кодирования Σ.

Определение. Алфавитное кодирование называется оптимальным (или с минимальной избыточностью), если средняя цена (вес) кодирования будет наименьшим из возможных.

5.5 Алгоритм Хаффмана

Алгоритм Хаффмана строит схему оптимального префиксного алфавитного кодирования для заданного распределения вероятностей или частот появления букв алфавита A.

Мы будем строить этот алгоритм с заданным распределением частот. В этом случае алгоритм легко построить вручную с применением понятия взвешенного дерева – графа с определёнными свойствами. Случай, когда задано распределение вероятностей рассмотрен в книге А.В. Чашкина [19] или в книге Ф.А. Новикова [3], где построена программа алгоритма Хаффмана.

Для наглядности разберём алгоритм Хаффмана на примере. Рассмотрим бинарное дерево G с листьями $V_1, V_2, V_3, V_4, V_5, V_6$. Это дерево ориентировано от корня, для простоты на рисунке не изображаем стрелки и дуги называем рёбрами.

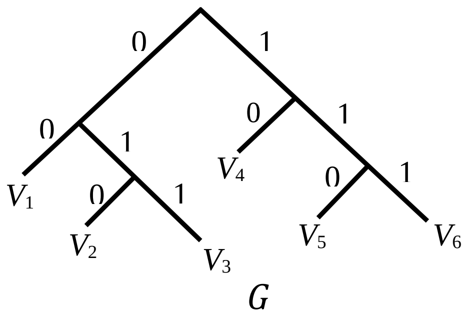


Рисунок 5.5.1

Если рассматривать путь от корня к любому из листьев, то каждое ребро этого пути должно вести к правому или левому сыну предыдущей вершины. Если ребро ведёт к левому сыну, то обозначим его через 0, если к правому – через 1. На рисунке 0 и 1 записываем рядом с соответствующим ребром. Поскольку путь, например, к листу V_1 состоит из двух рёбер к левым сыновьям вершин этого пути, то путь к V_1 обозначим 00. Аналогично пути к V_2, V_3, V_4, V_5, V_6 обозначаются соответственно через 010, 011, 10, 110, 111. Строку из 0 и 1 соответствующую данному листу назовём путевым кодом. Число нулей и единиц в путевом коде листа V_i назовём его длиной, обозначим l_i. Заметим, что т.к. к каждому листу ведёт единственный путь, то путевой код для каждого листа единственный, а путевые коды, соответствующие всем листьям этого дерева образуют префиксное множество кодов.

Теорема. В любом бинарном дереве путевые коды для листьев дерева образуют префиксное множество кодов.

Алфавитное кодирование с алфавитами $A = \{a_1, a_2, \dots, a_n\}$ и $B = \{0,1\}$ можно представить так: каждой букве a_i алфавита A ставится в соответствие лист V_i на бинарном дереве и его путевой код, состоящий из 0 и 1.

Таким образом, получена схема алфавитного кодирования Σ. Для дерева G, рассмотренного выше, эта схема выглядит так:

$\Sigma =$

a_1 (V_1)	a_2 (V_2)	a_3 (V_3)	a_4 (V_4)	a_5 (V_5)	a_6 (V_6)
00	010	011	10	110	111

Она будет, как указано выше, префиксной, а также разделимой. Таким образом, путевые коды – это элементарные коды букв a_i по терминологии, приведённой выше (т.е. $\beta_1 = 00$, $\beta_2 = 010$ и т.д.).

Пусть каждая буква a_i алфавита A появляется с частотой k_i, а l_i - длина путевого кода листа V_i (буквы a_i) или длина элементарного кода этой буквы. Вес кодирования $\omega = \sum_{i=1}^{n} k_i l_i$ был определён выше, эта величина называется также весом соответствующего бинарного дерева. Это понятие не надо путать с понятием взвешенного графа, определённого в разделе графы, где вес приписывается рёбрам (дугам) графа.

Наша задача получить оптимальное кодирование, т.е. кодирование с наименьшим весом (или бинарное дерево с наименьшим весом). Процесс построения такого дерева и называется алгоритмом Хаффмана, а построенный код (схема) – кодом (схемой) Хаффмана, построенное бинарное дерево – деревом Хаффмана.

Алгоритм Хаффмана для алфавита $A = \{a_1, a_2, \dots, a_n\}$ с соответствующими частотами букв $k_1, k_2, \dots, k_n$:

1) расположить частоты в возрастающем порядке;

2) если k_i и k_j две наименьшие частоты, то сформировать бинарное дерево с a_i и a_j в качестве сыновей, где a_i левый сын, a_j - правый ($k_i < k_j$), а k_i+k_j- частота родителя; поместить 0 на ребро левого сына, 1 – на ребро правого; удалить k_i и k_j из множества частот и добавить туда k_i+k_j;

3) снова расположить частоты в возрастающем порядке; удалить две наименьшие частоты из множества частот и сформировать дерево, где в качестве сыновей буквы или деревья, соответствующие этим частотам, причём буква или дерево с меньшей частотой – левый сын, а сумма их частот – частота родителя; если какой либо из сыновей дерево, то удалить метку его частоты из построенного дерева; заменить значения двух наименьших частот их суммой;

4) повторять шаг 3) пока во множестве частот не останется один элемент;

5) удалить частоту из корня последнего дерева, оно будет деревом Хаффмана, т.е. деревом с минимальным весом; по дереву Хаффмана записать схему кодирования Σ, которая будет префиксной, разделимой и оптимальной.

То, что по алгоритму Хаффмана построено дерево минимального веса (или оптимальная схема кодирования) утверждается в следующей теореме.

Теорема. Для заданного алфавита с соответствующими частотами букв дерево Хаффмана является деревом с минимальным весом.

Доказательство этой теоремы см. в книге Д. Андерсона [4], стр.644.

Пример 5.5.1 Построить схему оптимального кодирования для алфавита $A = \{a, b, c, d, e, f, m\}$, буквы которого имеют частоты 7, 10, 3, 5, 8, 2, 1.

Решение: строим алгоритм Хаффмана:

1) располагаем частоты в возрастающем порядке, результат для удобства запишем в таблицу

m	f	c	d	a	e	b
1	2	3	5	7	8	10

2) формируем бинарное дерево, где m и f листья, m – левый сын, f - правый, $1 + 2 = 3$ частота родителя; приписываем левому сыну 0, правому – 1; обозначим это дерево G_1; в списке частот заменяем значения двух наименьших

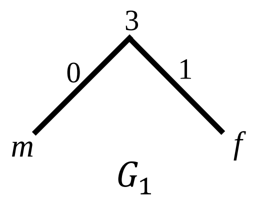


Рисунок 5.5.2

частот их суммой и сразу упорядочиваем частоты по возрастанию; записываем новую таблицу

G_1	c	d	a	e	b
3	3	5	7	8	10

3) формируем дерево, где в качестве сыновей построенное выше дерево G_1 и буква c (как имеющие наименьшие частоты в последней таблице); сумма их частот $3 + 3 = 6$; новое дерево G_2:

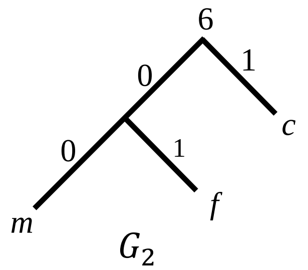
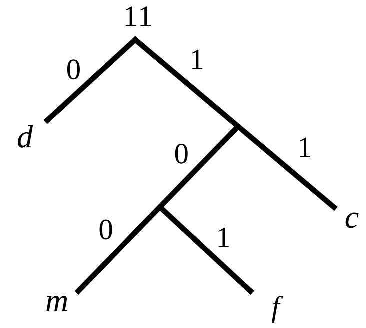


Рисунок 5.5.3

Новая таблица:

d	G_2	a	e	b
5	6	7	8	10

4) формируем дерево, где сыновьями будут d и G_2, d – левый сын, G_2 - правый; сумма их частот $5 + 6 = 11$; новое дерево G_3:



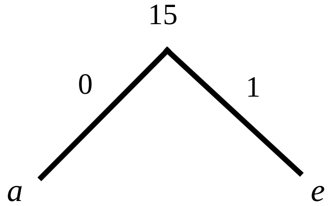
G_3

Рисунок 5.5.4

Новая таблица:

a	e	b	G_3
7	8	10	11

5) по последней таблице строим дерево G_4 (его вес $7 + 8 = 15$):



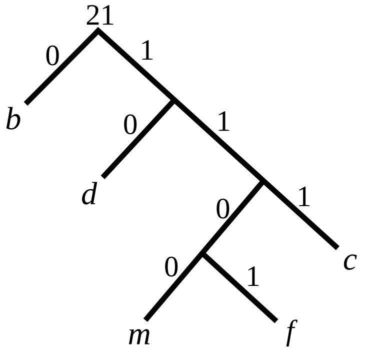
G_4

Рисунок 5.5.5

Новая таблица:

b	G_3	G_4
10	11	15

6) по последней таблице строим дерево G_5 (его вес $10 + 11 = 21$):



G_5

Рисунок 5.5.6

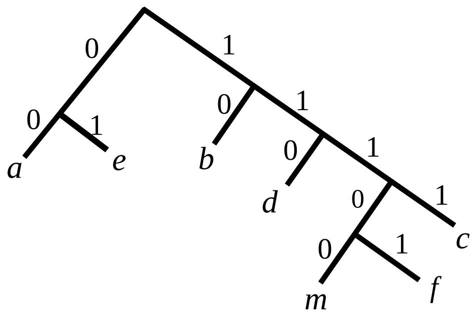
Новая таблица:

G_4	G_5
15	21

7) по последней таблице строим дерево G_6 (его вес $15 + 21 = 36$) – дерево Хаффмана; по дереву Хаффмана записываем искомую схему оптимального кодирования

$\Sigma =$

a	b	c	d	e	f	m
00	10	1111	110	01	11101	11100



G_6

Рисунок 5.5.7

Σ - префиксная, разделимая и оптимальная схема кодирования, её минимальный вес $\omega = 36$.

Заметим, что если две наименьшие частоты в таблице одинаковы (как в этом примере на втором шаге), то выбор левого сына может быть другой. Тогда получится другая схема кодирования. Однако она имеет тот же наименьший вес.

Список литературы

1. Судоплатов С.В., Овчинникова Е.В. Дискретная математика. – М.:, «ЮРАЙТ», 2020 – 280с.
2. Москинова Г.И. Дискретная математика. Математика для менеджера в примерах и упражнениях: Учебное пособие. – М.: Логос, 2004 – 240с.
3. Новиков Ф.А. Дискретная математика для программистов. Учебник для вузов. 3-е изд. – СПб.: Питер, 2009 – 384с.
4. Андерсон, Д. Дискретная математика и комбинаторика.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004 – 960с.
5. Спирина М.С., Спирин П.А. Дискретная математика: учебник. – 9-е изд. - М.: Издат. центр «Академия», 2013 - 368с.
6. Шапорев С.Д. Дискретная математика. Курс лекций и практических занятий. – СПб.: БХВ-Петербург, 2006 – 396с.
7. Чашкин А.В. Дискретная математика. М.: Издат. центр «Академия», 2012 – 352с.
8. Асеев Г.Г., Абрамов О.М., Ситников Д.Э. Дискретная математика: Учеб. пособие. – Ростов н/Д: «Феникс», Харьков: «Торсинг», 2003 – 144с.
9. Плотников А.Д. Дискретная математика.: Учебное пособие. – 2-е изд., испр. и доп. – М.: Новое знание, 2006 – 304с.
10. Оре О. Графы и их применение: пер. с англ./ Под ред. и с предисл. И.М.Яглома. Изд. 3-е, стереотипное. – М.: КомКнига, 2006 – 168с.
11. Просветов Г.И. Дискретная математика: задачи и решения: учеб.-практ. пособие. – 2-е изд., – М.: Альфа-Пресс, 2014 – 240с.
12. Данилов В.Г. и др., Дискретная математика.: Учебное пособие для вузов. - М.: 2008 – 136с.
13. Палий И.А. Дискретная математика.: Курс лекций. – М.: Эксмо, 2008 – 352с.
14. Галушкина Ю.И., Марьямов А.Н. Конспект лекций по дискретной математике. – М.: Айрис-пресс, 2007 – 176с.
15. Шевелев Ю.П. Дискретная математика.: Учебное пособие – СПб.: Издательство «Лань», 2016 – 592с.
16. Шевелев Ю.П., Писаренко Л.А., Шевелев М.Ю. Сборник задач по дискретной математике (для практических занятий в группах) – СПб.: Издательство «Лань», 2013 – 528с.
17. Астраханцева Л.Н. Дискретная математика.: Учебное пособие. – Алматы: АУЭС, 2011. – 78с.
18. Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике. – М.: Высш. шк., 2013 – 400с.
19. Чашкин А.В. Дискретная математика. – М., 2012.

Содержание

Астраханцева Людмила Николаевна
Байсалова Маншук Жумамуратовна

ДИСКРЕТНАЯ МАТЕМАТИКА
Учебное пособие

Редактор Жанабаева Е.Б.

Подписано в печать ___ ___________ 2021.
Тираж 100 экз. Формат 60х84 1/16

Бумага типографская №1
Уч.- изд. лист 6,0. Заказ ___
Цена 3000 тенге.

Некоммерческое АО «АУЭС имени Гумарбека Даукеева »
Алматы, Байтурсынова, 126/1

Копировально-множительное бюро
некоммерческое акционерное общество
«Алматинский университет энергетики и связи »
050013, Алматы, Байтурсынова, 126/1