Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Попов Анатолий Николаевич
Должность: директор
Дата подписания: 16.06.2022 18:11:32
Уникальный программный ключ:
1e0c38dcc0aee73cee1e5c09c1d5873fc7497bc8

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

Приложение 2
к рабочей программе дисциплины

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Дискретная математика
(наименование дисциплины(модуля)

Направление подготовки / специальность

09.03.03 Прикладная информатика
(код и наименование)

Направленность (профиль)/специализация

Прикладная информатика на железнодорожном транспорте
(наименование)

Содержание

1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ОПК-1.1: Применяет методы высшей математики для решения задач профессиональной деятельности	Знает способы задания, свойства множеств, отношений, функций и отображений; канонические формы представления, методы преобразования и минимизации булевых функций; методы осуществления операций над графами и выполнения количественных оценок их характеристик; стандартные и рекурсивные схемы алгоритмов, структуры и потоки данных
	Использует символику дискретной математики для выражения количественных и качественных отношений объектов; методы дискретной математики при решении задач синтеза цифровых устройств и разработке программного обеспечения
	Владеет методами дискретной математики при решении задач синтеза цифровых устройств и разработке программного обеспечения; имеет представление о перспективах использования методов дискретной математики при разработке моделей систем автоматики и вычислительной техники

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ОПК-1.1: Применяет методы высшей математики для решения задач профессиональной деятельности	Использует символику дискретной математики для выражения количественных и качественных отношений объектов; методы дискретной математики при решении задач синтеза цифровых устройств и разработке программного обеспечения	Задания (тесты 1-5)
	Использует символику дискретной математики для выражения количественных и качественных отношений объектов; методы дискретной математики при решении задач синтеза цифровых устройств и разработке программного обеспечения	Задания 1
	Владеет методами дискретной математики при решении задач синтеза цифровых устройств и разработке программного обеспечения; имеет представление о перспективах использования методов дискретной математики при разработке моделей систем автоматики и вычислительной техники	Задания 2

Промежуточная аттестация (зачет) проводится в одной из следующих форм:
1) собеседование;
2) выполнение заданий в ЭИОС СамГУПС.

2. Типовые[1] контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат

Код и наименование индикатора достижения компетенции	Образовательный результат
ОПК-1.1	Знает методы высшей математики для решения задач профессиональной деятельности

1. Перечислите элементы следующего множества B ={x | x$\in$Z и 6x^2 + x – 1 = 0}.
2. Запишите булеан P(A) множества A = {0,-1, -2, -3}.
3. Задайте с помощью характеристического свойства множество A = {2, 5, 8, 11, …}.
4. Для заданного множества $A \subseteq U$ составить характеристический вектор: U={1,2,3,…,20}, A={x| 5|(x+3) }.
5. Доказать равенство множеств, преобразуя множества к одинаковому виду с помощью основных законов алгебры множеств: $(A \cap B) \cup (B \setminus A) \cup (A \setminus B) = B \cup A$.
6. Доказать тождество (тремя способами): $\overline{(\overline{A} \cup \overline{B})} = A \cap B$.
7. Доказать тождество: $(A \cup B) \times C = (A \times C) \cup (B \times C)$
8. Составьте матрицу данного бинарного отношения: $\rho \subseteq \{1,2,3...,7\}^2, (x,y) \in \rho \Leftrightarrow x^2 \geq y^2$
9. Бинарное отношение между множествами A = {1, 2, 3, 4} и B = {a, b, c, d} задано матрицей. $\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ Выпишите элементы этого отношения и постройте его изображение.
10. Дано бинарное отношение $\rho = \{(x,y) | x, y \in N, y | x\}$, найдите $D_\rho, E_{\rho,}, \rho^{-1}, \rho \circ \rho, \rho^{-1} \circ \rho$
11. Является ли отношение ρ рефлексивным, симметричным, антисимметричным, транзитивным? $\rho \subseteq Z^2; (x,y) \in \rho \Leftrightarrow (x^2 + y) \vdots 2$.
12. Бинарное отношение между множествами A = {1, 2, 3, 4} и B = {a, b, c, d} $\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ задано матрицей. Выпишите элементы этого отношения и постройте его изображение. Проверьте, является ли это отношение рефлексивным, симметричным, антисимметричным, транзитивным? Найдите его области определения и значений.
13. Является ли данная функция инъективной? Сюръективной? Биективной? Почему? Постройте ее график. f: R→R, f(x)=lnx-1.
14. B = {1, 2, 3, 4}, $\rho \subseteq B^2$. Изобразите ρ графически. Проверьте с помощью матрицы является ли отношение ρ рефлексивным, симметричным, антисимметричным, транзитивным? $\rho = \{(1,1),(1,2),(1,4),(2,2),(2,4),(3,3),(3,2),(3,4),(4,4)\}$
15. Построить следующие бинарные отношения:

[1]Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

а) рефлексивное, симметричное , не транзитивное;

б) не рефлексивное, антисимметричное, не транзитивное;

в) рефлексивное, не симметричное, транзитивное.

16. Доказать, что отношение {(a, b) | (a – b) – рациональное число} является отношением эквивалентности на множестве натуральных чисел.
17. К каким типам (эквивалентности; строгого, нестрогого, линейного порядка) относятся данные отношения:
    1) отношение равносильности на множестве формул;
    2) отношения ≤ и < на множестве векторов длины n с компонентами из N, определяемые следующим образом:

а) $(a_1, \dots, a_n) \le (b_1, \dots, b_n)$, если $a_1 < b_1, \dots, a_n < b_n$;

б) $(a_1, \dots, a_n) < (b_1, \dots, b_n)$, если $(a_1, \dots, a_n) \le (b_1, \dots, b_n)$ и хотя бы для одной координаты i выполняется $a_i < b_i$;

3) отношение предшествования слов упорядоченного конечного алфавита.

18. Проиллюстрировать диаграммой Эйлера-Венна следующие разбиения множества U:
    1) $\{A, \overline{A}\}$

    2) $\{A \cap B, A \cap \overline{B}, \overline{A} \cap B, \overline{A} \cap \overline{B}\}$

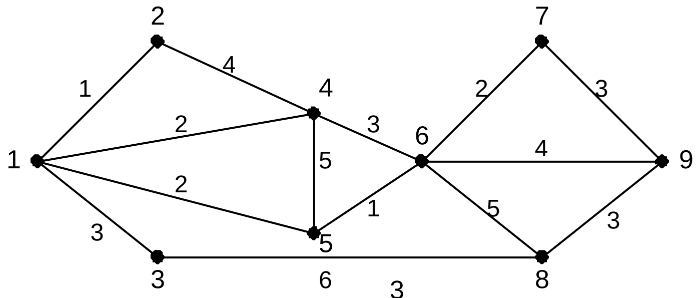
    3) $\{A \setminus B, A \cap B, B \setminus A\}$.

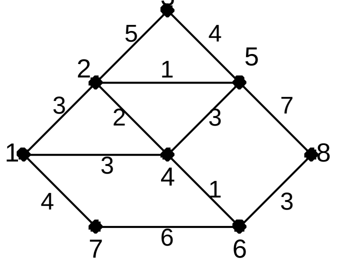
19. Построить диаграмму Хассе для отношения $\subseteq$ на булеане P(A), где A={1, 2, 3}.
20. Сколько пятибуквенных слов, каждое из которых состоит из трех согласных и двух гласных, можно. образовать из букв слова ***уравнение****?*
21. Сколько четырехбуквенных слов можно образовать из букв слова ***сапфир****?* 2) Сколько среди них таких, которые не содержат буквы ***р***? 3) Сколько таких, которые начинаются с буквы ***с*** и оканчиваются буквой ***р****?*
22. Сколько различных перестановок можно образоватьизо всех букв слова ***перестановка****?* Сколько из них начинается с буквы ***п*** и оканчивается буквой **а**?
23. Сколько сигналов можно подать пятью различными флажками, поднимая их в любом количестве и в произвольном порядке?
24. Войсковое подразделение состоит из 5 офицеров, 8 сержантов и 70 рядовых. Сколькими способами можно выделить отряд из 2 офицеров, 4 сержантов и 15 рядовых?
25. Найти число всех подмножеств множества *X*, если *X* содержит *k* элементов.
26. Сколькими способами можно выбрать 2 стандартные и 1 нестандартную детали из 40 деталей, среди которых имеются 10 нестандартных?
27. Сколько существует различных трехзначных чисел?
28. По *n* ящикам случайно распределяются *n* шаров. Считая, что ящики и шары различимы, найти вероятности следующего события: {два ящика пустых} = A_2
29. Саша Иванов - средний студент и обычно дает правильные ответы лишь на половину экзаменационных билетов. На очередном экзамене Саша на билет ответил и получил положительную оценку. Какие события можно считать случайными: а) Саше попался «хороший» билет – событие *A*;
30. б) Саша ответил на билет – событие *B*; в) Саша сдал экзамен – событие *C*.
31. Саша и Маша разыгрывают билет на концерт. Какие из следующих событий можно считать случайными? а) Только Саша выиграл билет – событие *A*; б) Только Маша выиграла билет – событие *B*; в) Саша или Маша выиграли билет – событие *C*; г) Оба выиграли билет – событие *D*.
32. Рассмотрим работу столовой, в плане обслуживания клиентов. Моменты прихода посетителей (событие *A*), время, затрачиваемое клиентами на обед (событие *B*) – можно ли считать случайными событиями; а процесс обслуживания клиентов – случайным процессом?
33. В урне из *n* шаров - *k* красных и (*n* - *k*) черных. Наудачу извлекаем без возвращения *r* шаров. Какова

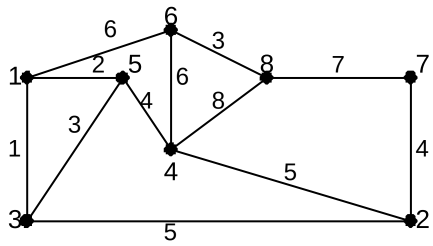
вероятность того, что в выборке из r шаров s шаров – красных?

34. Вычислить вероятность того, что для наудачу взятого значения $x \in [0, 2\pi)$, значение $y = \sqrt{0{,}5 - \sin^2 x}$ существует.
35. Два друга договорились встретиться между 12 и 13 часами. Пришедший первым ждет второго в течении 20 минут, после чего уходит. Найти вероятность, что встреча произойдет, если каждый наудачу выбирает время своего прихода в промежутке от12 до 13 часов.
36. Применяя формулу полной вероятности, вычислить вероятность того, что при подбрасывании симметричного кубика выпадет четная грань.
37. Разложите в сумму выражение $(3y+2)^4$
38. Разложить в ряд $(а + в + с)^3$
39. Сколько положительных целых чисел от 70 до 950 делится ровно на одно из чисел:
    1) 7, 11 или 13 2) 3, 5 или 17.

40. Найти кратчайший путь, построить кратчайшее остовное дерево для следующих графов.







41. По заданной матрице смежности построить изображение графа: $A_G = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$

$$A_G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix} \quad A_G = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

42. По заданной матрице инцидентности построить изображение графа:

$$B_G = \begin{pmatrix} 1 & -1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B_G = \begin{pmatrix} 1 & -1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

43. Построить таблицу истинности, СДНФ, СКНФ, полином Жегалкина для следующих булевых функций, заданных формулами:
    1) $\overline{(\bar{x} \cup y) \cup x \cdot \bar{z}} \downarrow (x \sim y)$,
    2) $\bar{x} \to (\bar{z} \sim (y \oplus x \cdot z))$,
    3) $(x \cup y \cup z) \to (x \cup y) \cdot (x \cup z)$,
    4) $\overline{(x \to y) \cup (x \to z) \cdot y}$,
    5) $x\bar{y} \cdot (\bar{y} \to x\bar{z})$,
    6) $((x_1 \to x_2x_3) \cdot (x_2x_4 \oplus x_3) \to x_1\bar{x}_4) \cup \bar{x}_1$,
    7) $((x_1 \to x_2) \cup \bar{x}_3) | x_1$,
    8) $((x_3 \to x_2) \cup x_1) \cdot (x_2 \to x_1)x_3\bar{x}_1 \oplus x_3$,
    9) $(x_1 \to (x_1 \cup x_2)) \to x_3$,
    10) $\overline{(x \to y) \to xz} \to (y \to z)$,
    11) $(\bar{x} \cup y \cup z) \cdot t \cup \bar{x}y\bar{z}$,
    12) $(x \cup \bar{y} \cup z) \cdot t \cup \bar{x}yz$,
    13) $(x \downarrow y) \to (x \oplus z)$,
    14) $xy \cup xz \cup yt \cup zt$,
    15) $(x \to y) \oplus ((x \downarrow y) | (\bar{x} \sim yz))$,
    16) $(\bar{x} \cup y \cup (y\bar{z} \oplus 1)) \to (y \cup x)$,
    17) $(x_1 \to x_2) \cdot (x_2 \to x_1) \sim x_3$,
    18) $((x \oplus y) \sim z) \,\&\, (x \to yz)$,
    19) $((x \oplus y) \to (x \cup y)) \cdot ((\bar{x} \to y) \to (x \oplus y)) | z$,
    20) $\overline{(x \to y) \to xz} \to (y \to z)$,
    21) $(x \cup \bar{y} \cup z) \cdot t \cup \bar{x}yz$,
    22) $(x \downarrow y) \to (x \oplus z)$,
    23) $(\bar{x} \cup y \cup \bar{z}) \cdot t \cup \bar{x}y\bar{z}$,
    24) $(x_1 \to x_2) \cdot (x_2 \to x_1) \sim x_3$,
    25) $(x_1 | x_2) \to x_3 \downarrow x_2$.
44. Проверить на полноту следующие системы булевых функций:
    1) $\{x \to y, \overline{x \oplus y \oplus z}\}$,
    2) $\{x \to y, (1100001100111100)\}$,
    3) $\{0, xy \cup xz \cup yz, xy \oplus z\}$,
    4) $\{(1011), (1111110011000000)\}$,
    5) $\{x\bar{y}, \bar{x} \sim yz\}$,
    6) $\{0, 1, x(y \sim z) \cup \bar{x}(y \oplus z)\}$,
    7) $\{(01101001), (10001101), (00011100)\}$,
    8) $\{(0010), (1010110111110011)\}$.
45. Используя метод Квайна и карты Карно, найти МДНФ И МКНФ формул:

1) $\bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3$

2) $\bar{x}\,\bar{y}\,\bar{z} \vee x\bar{y}\,\bar{z} \vee x\bar{y}z \vee xy\bar{z} \vee xyz$

3) $\bar{x}\,\bar{y}\,\bar{z} \vee x\bar{y}\,\bar{z} \vee xy\bar{z} \vee xyz$

46. Являются ли схемы данного алфавитного кодирования префиксными? Взаимно однозначными??

1) $(a \to 0, \mathrm{b} \to 10, \mathrm{c} \to 011, \mathrm{d} \to 1011, \mathrm{e} \to 1111)$

2) $(a \to 0, \mathrm{b} \to 10, \mathrm{c} \to 20)$

3) $(a \to 01, \mathrm{b} \to 021, \mathrm{c} \to 12, \mathrm{d} \to 0102, \mathrm{e} \to 10112)$

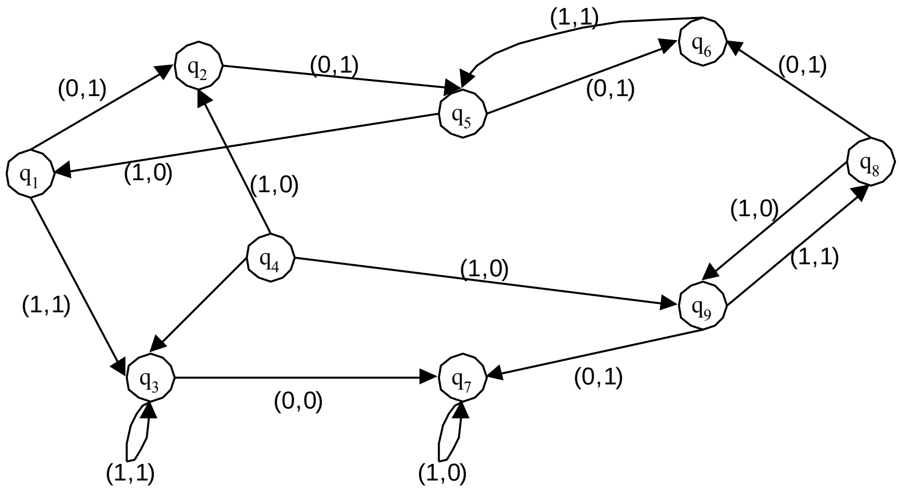
47. Построить префиксное алфавитное кодирование с минимальной избыточностью для алфавита $\{a, b, c, d\}$ со следующими распределениями вероятностей появления букв:

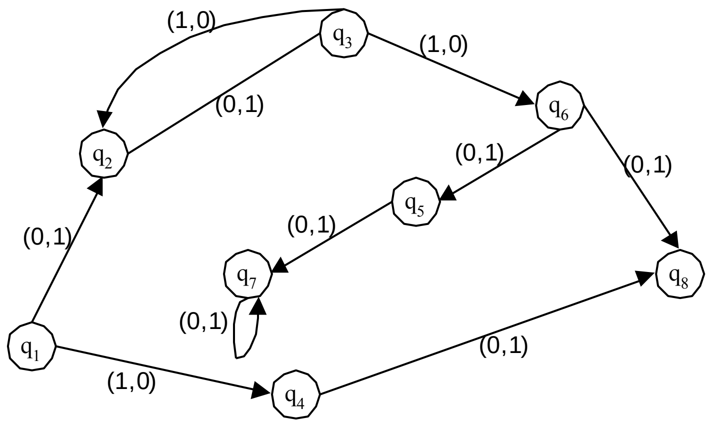
1) $p_0 = ½$, $p_1 = ¼$, $p_3 = p_4 = ⅛$;

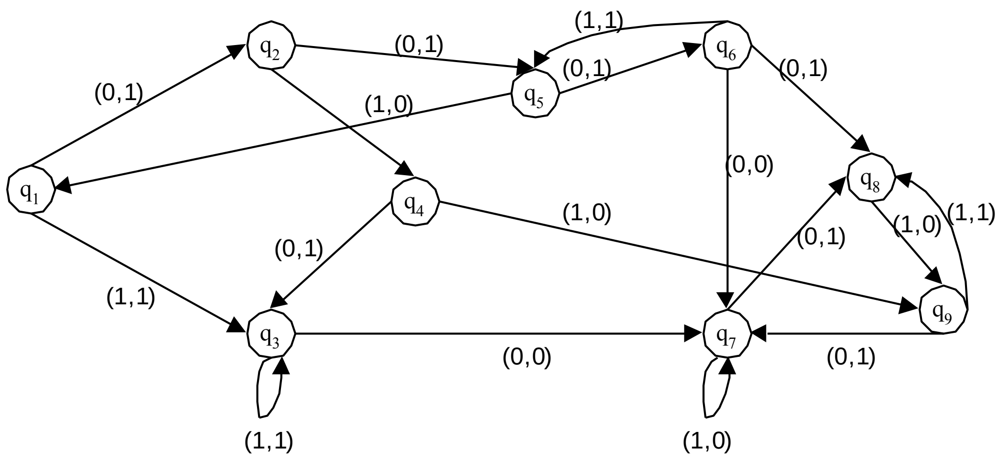
2) $p_0 = 0{,}4$, $p_1 = 0{,}25$, $p_3 = 0{,}2$, $p_4 = 0{,}15$;

3) $p_0 = 0{,}1$, $p_1 = 0{,}45$, $p_3 = 0{,}22$, $p_4 = 0{,}23$.

48. Минимизировать конечный автомат.







2.2 Теоретические сведения и типовые задания:

1. НАЧАЛЬНЫЕ ПОНЯТИЯ ТЕОРИИ МНОЖЕСТВ

1. 1. Элементы и множества

Человеческое мышление устроено так, что мир представляется состоящим из отдельных «объектов». Философам давно ясно, что мир – единое неразрывное целое, и выделение в нем объектов – это не более чем произвольный акт нашего мышления, позволяющий сформировать доступную для рационального анализа картину. Но как бы там ни было, выделение объектов и их совокупностей – естественный способ организации нашего мышления, поэтому не удивительно, что он лежит в основе главного инструмента описания точного знания – математики.

Понятие множества принадлежит к числу фундаментальных неопределяемых понятий математики. О множестве известно как минимум, что оно состоит из элементов. Для определенности остановимся на следующем определении.

*Определение.Под **множествомS** будем понимать любое собрание определенных и различимых между собой объектов, мыслимое как единое целое. Эти объекты называются **элементами множестваS**.*

Определение.Под множеством понимают объединение в единое целое определенных вполне различаемых предметов (объектов), которые при этом называются элементами образуемого ими множества.

Обычно множества обозначают прописными буквами латинского алфавита: *A, B, C*, …; а элементы множеств – строчными буквами: *a, b, c,* … .

Если объект x является элементом множества M, то говорят, что x *принадлежит* M: $x \in M$. В противном случае говорят, что x *не принадлежит* M: $x \notin M$.

В этом интуитивном определении, принадлежащем немецкому математику Г. Кантору, существенным является то обстоятельство, что собрание предметов само рассматривается как один предмет, мыслится как единое целое. Что касается самих предметов, которые могут входить в множество, то относительно них существует значительная свобода.

Пример 1. Это может быть множество студентов, присутствующих на лекции, множество четных чисел и т. д.

Определение. Множество A называется ***подмножеством*** *множества B, если всякий элемент из A является элементом B. Если A является подмножеством B и B не является подмножеством A, то говорят, что A является* ***строгим (собственным) подмножеством*** *B.*

В первом случае обозначают $A \subseteq B$, во втором случае $A \subset B$.

Определение. Множество, не содержащее элементов, называется ***пустым*** $\varnothing$, *оно является подмножеством любого множества. Множество U называется* ***универсальным****, то есть все рассматриваемые множества являются его подмножеством.*

Рассмотрим два определения равенства множеств.

Определение. ***Множества*** *A и B считаются* ***равными****, если они состоят из одних и тех же элементов, пишут $A=B$, $A \neq B$ – в противном случае.*

Определение. ***Множества*** *A и B считаются* ***равными****, если $A \subset B$ и $B \subset A$.*

Способы задания множеств:

- ***перечислением элементов****: $M=\{a_1, a_2, \ldots, a_k\}$, т. е. списком своих элементов;*
- ***характеристическим предикатом****: $M=\{x \mid P(x)\}$(описанием характеристических свойств, которыми должны обладать его элементы);*
- ***порождающей процедурой****: $M=\{ x \mid x=f\}$, которая описывает способ получения элементов множества из уже полученных элементов либо других объектов. В таком случае элементами множества являются все объекты, которые могут быть построены с помощью такой процедуры. Например, множество всех целых чисел, являющихся степенями двойки.*

*Замечание.*При задании множеств перечислением обозначения элементов обычно заключают в фигурные скобки и разделяют запятыми. Перечислением можно задавать только ***конечные множества*** (число элементов множества конечно, в противном случае множество называется бесконечным). Характеристический предикат – это некоторое условие, выраженное в форме логического утверждения или процедуры, возвращающей логическое значение. Если для данного элемента условие выполнено, то он принадлежит определяемому множеству, в противном случае – не принадлежит. Порождающая процедура – это процедура, которая, будучи запущенной, порождает некоторые объекты, являющиеся элементами определяемого множества. ***Бесконечные множества*** задаются характеристическим предикатом или порождающей процедурой.

Пример 2.

1. $M=\{1, 2, 3, 4\}$ – перечисление элементов множества.

2. Ì = {ò |m ∈ N è ò ≤10} - характеристический предикат.

3. Числа Фибоначчи задаются условиями (порождающей процедурой):

$a_1=1$, $a_2=2$, $a_n=a_{n-1}+a_{n-2}$ для $n>2$.

Определение. ***Мощность*** *конечного множества A - это число его элементов.*

Мощность множества обозначают |A|.

Пример 3.

$|\varnothing|=0$, $|\{\varnothing\}|=1$.

Определение. Множества называются ***равномощными****, если их мощности совпадают.*

Определение. Множество всех подмножеств множества A называется ***булеаном*** *P(A).*

Известно, что если множество *A* содержит *n* элементов, то множество *P(A)* содержит 2^n элементов. В связи с этим используется также обозначение множества-степени множества *A* в виде 2^A.

Пример 4.

A={0, 1, 2}, P(A)={ ∅, {0}, {1}, {2}, {0, 1}, {0, 2}, {1, 2}, {0, 1, 2}}.

1.2. Операции над множествами. Диаграммы Эйлера-Венна

Диаграммы Эйлера-Венна – геометрические представления множеств. Построение диаграммы заключается в изображении большого прямоугольника, представляющего универсальное множество *U*, а внутри его – кругов (или каких-нибудь других замкнутых фигур), представляющих множества. Фигуры должны пересекаться в наиболее общем случае, требуемом в задаче, и должны быть соответствующим образом обозначены. Точки, лежащие внутри различных областей диаграммы, могут рассматриваться как элементы соответствующих множеств. Имея построенную диаграмму, можно заштриховать определенные области для обозначения вновь образованных множеств.

Операции над множествами рассматриваются для получения новых множеств из уже существующих.

Определение. ***Объединением*** *множеств A и B называется множество, состоящее из всех тех элементов, которые принадлежат хотя бы одному из множеств A, B (рис. 1):*

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

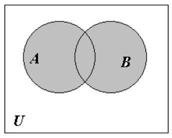


Рис. 1.

Определение. ***Пересечением*** *множеств A и B называется множество, состоящее из всех тех и только тех элементов, которые принадлежат одновременно как множеству A, так и множеству B (рис. 2):*

$$A \cap B = \{x \mid x \in A \text{ и } x \in B\}.$$

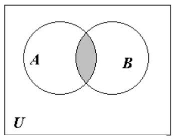


Рис. 2.

Определение. ***Разностью*** *множеств A и B называется множество всех тех и только тех элементов A, которые не содержатся в B (рис. 3):*

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

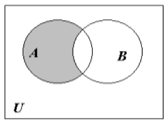


Рис. 3.

Определение. ***Симметрической разностью*** *множеств A и B называется множество элементов этих множеств, которые принадлежат либо только множеству A, либо только множеству B (рис. 4):*

$$A + B = \{x \mid \text{либо } x \in A, \text{ либо } x \in B\}.$$

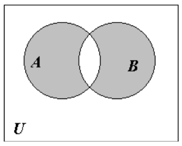


Рис. 4.

Определение. ***Абсолютным дополнением*** *множества A называется множество всех тех элементов, которые не принадлежат множеству A (рис. 5):*

$$\overline{A} = U \setminus A.$$

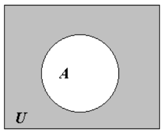
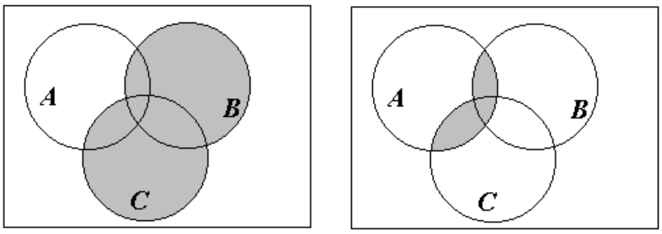


Рис. 5.

Пример 5. С помощью диаграмм Эйлера – Венна проиллюстрируем справедливость



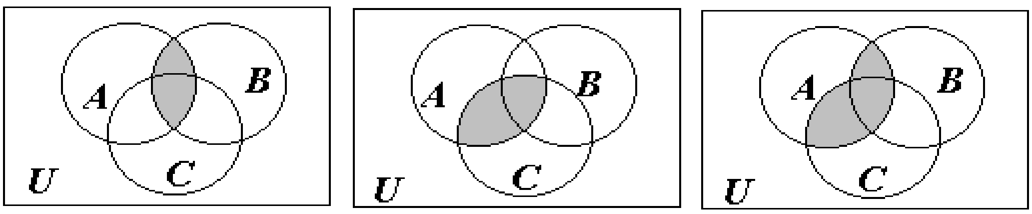
соотношения $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (рис. 6).

$B \cup C$ $\qquad$ $A \cap (B \cup C)$

$A \cap B$ $\qquad$ $A \cap C$ $\qquad$ $(A \cap B) \cup (A \cap C)$

Рис. 6.

Убедились, что в обоих случаях получаем равные множества. Следовательно, исходное



соотношение справедливо.

1.3. Основные тождества алгебры множеств

Для произвольных множеств A, B, и C справедливы следующие соотношения (табл. 1):

Таблица 1

1. Коммутативность объединения $A\cup B=B\cup A$	1’. Коммутативность пересечения $A\cap B=B\cap A$
2. Ассоциативность объединения $A\cup(B\cup C)=(A\cup B)\cup C$	2’. Ассоциативность пересечения $A\cap(B\cap C)=(A\cap B)\cap C$
3. Дистрибутивность объединения относительно пересечения $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$	3’. Дистрибутивность пересечения относительно объединения $A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$
4. Законы действия с пустым и универсальным множествами $A\cup\varnothing=A$ $A\cup\overline{A}=U$ $A\cup U=U$	4’. Законы действия с пустым и универсальным множествами $A\cap U=A$ $A\cap\overline{A}=\varnothing$ $A\cap\varnothing=\varnothing$
5. Закон идемпотентности объединения $A\cup A=A$	5’. Закон идемпотентности пересечения $A\cap A=A$
6. Закон де Моргана $\overline{A\cup B}=\overline{A}\cap\overline{B}$	6’. Закон де Моргана $\overline{A\cap B}=\overline{A}\cup\overline{B}$
7. Закон поглощения $A\cup(A\cap B)=A$	7’. Закон поглощения $A\cap(A\cup B)=A$
8. Закон склеивания $(A\cap B)\cup(A\cap\overline{B})=A$	8’. Закон склеивания $(A\cup B)\cap(A\cup\overline{B})=A$
9. Закон Порецкого $A\cup(\overline{A}\cap B)=A\cup B$	9’. Закон Порецкого $A\cap(\overline{A}\cup B)=A\cap B$
10. Закон двойного дополнения $\overline{\overline{A}}=A$	

Пример 6.

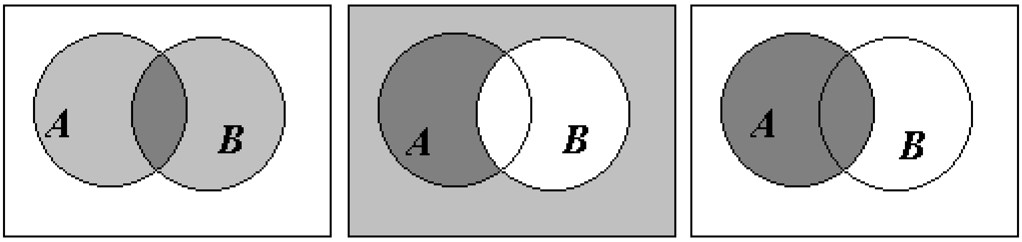
Доказать следующее тождество $(A\cap B)\cup(A\cap\overline{B})=A$.

Решение.

Докажем это тождество двумя способами: аналитически (используя равносильности алгебры множеств) и конструктивно (используя диаграммы Эйлера-Венна).

1. $(A\cap B)\cup(A\cap\overline{B})=(A\cup(A\cap\overline{B}))\cap(B\cup(A\cap\overline{B}))=A\cap((B\cup A)\cap(B\cup\overline{B}))=A\cap((B\cup A)\cap U)=$
$=A\cap(B\cup A)=A$

2. Построим соответствующие диаграммы Эйлера-Венна (рис. 7).



$A\cap B$ $\quad$ $A\cap\overline{B}$ $\quad$ $(A\cap B)\cup(A\cap\overline{B})=A$

Рис. 7.

1.4. Прямое произведение множеств. Отношения и функции

Определение. ***Упорядоченная пара*** *<x, y> интуитивно определяется как совокупность, состоящая из двух элементов x и y, расположенных в определенном порядке. Две пары <x, y>, <u, v> считаются равными тогда и только тогда, когда x=u, y=v.*

Упорядоченная n-ка элементов x_1, ..., x_n обозначается <x_1, ..., x_n>.

Определение. ***Прямым произведением*** *множеств X и Y называется множество* $X\times Y$, *элементами которого являются все возможные упорядоченные пары <x, y>, такие, что* $x\in X, y\in Y$.

Определение. ***Прямым произведением*** *множеств X_1, X_2, ..., X_n называется совокупность всех упорядоченных n-ок <x_1, ..., x_n> таких, что* $x\in X_1,...,x_n\in X_n$. *Если X_1=X_2=...X_n, то пишут* $X_1\times X_2\times...\times X_n=X^n$.

Пример 7.

1. Пусть *X={1, 2, 3}*, *Y={0, 1}*. Тогда $X\times Y=\{\langle 1,0\rangle,\langle 1,1\rangle,\langle 2,0\rangle,\langle 2,1\rangle,\langle 3,0\rangle,\langle 3,1\rangle\}$; $Y\times X=\{\langle 0,1\rangle,\langle 0,2\rangle,\langle 0,3\rangle,\langle 1,1\rangle,\langle 1,2\rangle,\langle 1,3\rangle\}$.

2. Пусть *X* – множество точек отрезка [0, 1], а *Y* – множество точек отрезка [1, 2]. Тогда $X\times Y$ - множество точек квадрата $[0,1]\times[1,2]$ с вершинами в точках (0, 1), (0, 2), (1, 1), (1,2).

Определение. ***Бинарным*** *(или двуместным)* ***отношением ρ*** *называется множество упорядоченных пар.*

Если ρ есть отношение и пара $<x, y>$ принадлежит этому отношению, то наряду с записью $<x, y> \in \rho$ употребляется запись $x\rho y$. Элементы x и y называются координатами (или компонентами) отношения ρ.

Определение. ***N-арным отношением*** *называется множество упорядоченных n-ок.*

Определение. ***Областью определения*** *бинарного отношения ρ называется множество*
$D_\rho = \{x \,|\, \text{ñóùåñòâóåò òàêîå ó, ÷òî } x\,\rho\,y\}$.

Определение. ***Областью значений*** *бинарного отношения ρ называется множество*
$\text{Å}_\rho = \{\text{ó} \,|\, \text{ñóùåñòâóåò òàêîå õ, ÷òî } x\,\rho\,y\}$.

Пусть $\rho \subseteq X \times Y$определено в соответствии с изображением на рисунке 8 . Область определения D_ρ и область значений E_ρ определяются соответственно:

$D_\rho=\{x: (x, y) \in\rho\}, E_\rho=\{y: (x,y)\in\rho\}$.

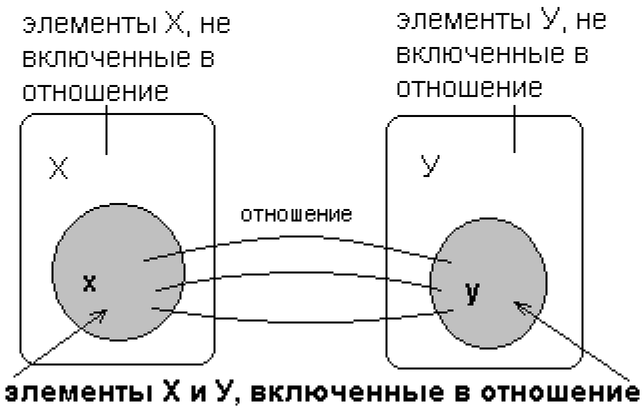


Рис. 8.

Бинарное отношение можно задать любым из способов задания множеств.Помимо этого отношения, определенные на конечных множествах обычно задаются:

1. *списком (перечислением) пар*, для которых это отношение выполняется.
2. *матрицей* – бинарному отношению соответствует квадратная матрица порядка n, в которой элемент c_{ij}, стоящий на пересечении i-той строки и j-го столбца, равен 1, если a_i и a_j имеет место отношение, или 0, если оно отсутствует.

Пример 8.

Пусть M={1, 2, 3, 4, 5, 6}. Задать в явном виде (списком) и матрицей отношение ρ, заданное на множестве $M \times M$, если ρ означает «быть строго меньше».

Отношение ρ как множество содержит все пары элементов a, b из M такие, что $a<b$. Тогда

ρ = *{(1, 2), (1,3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6)}.*

Матрица отношения имеет вид:

$$\rho = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Определение. Бинарное отношение f называется ***функцией****, если из* $<x, y> \in f$ *и* $<x, z> \in f$ *следует, что* $y=z$.

Поскольку функции являются бинарными отношениями, то две функции f и g равны, если они состоят из одних и тех же элементов. Область определения функции обозначается D_f, а область значений – R_f. Определяются они так же, как и для бинарных отношений.

Если f – функция, то вместо $<x, y> \in f$ пишут $y=f(x)$ и говорят, что y – значение, соответствующее аргументу x, или y – образ элемента x при отображении f. При этом x называется прообразом элемента y.

Определение. Назовем ***f n-местной функцией*** *из X в Y если* $f:X^n \to Y$. *Тогда пишем* $y=f(x_1, x_2, \ldots, x_n)$ *и говорим, что y – значение функции при значении аргументов* $x_1, x_2, \ldots, x_n$.

Пусть $f:X \to Y$.

Определение. Функция f называется ***инъективной****, если для любых* x_1, x_2, y *из* $y=f(x_1)$ *и* $y=f(x_2)$ *следует, что* $x_1=x_2$, *то есть каждому значению функции соответствует единственное значение аргумента.*

Определение. Функция f называется ***сюръективной****, если для любого элемента* $y \in Y$ *существует элемент* $x \in X$ *такой, что* $y=f(x)$.

Определение. Функция f называется ***биективной****, если f одновременно сюръективна и инъективна.*

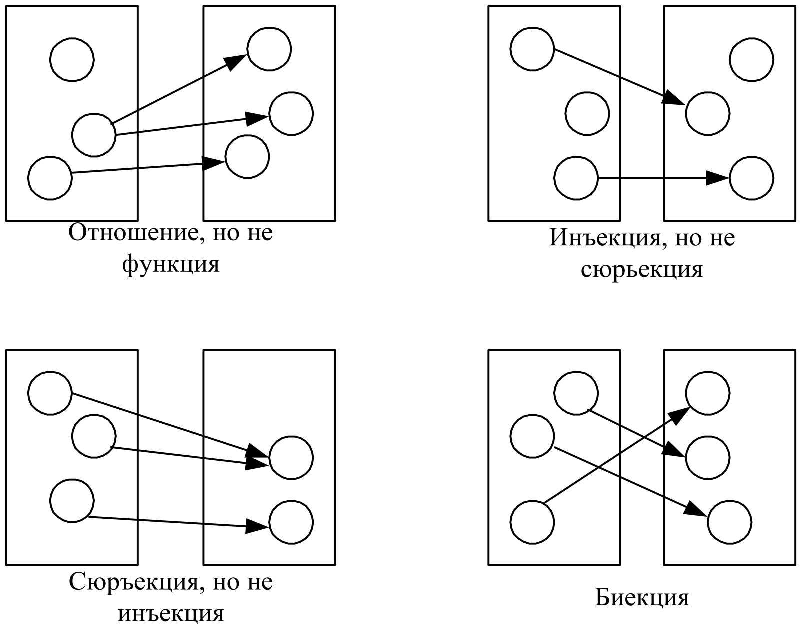


Рис. 9.

Рисунок 9 иллюстрирует понятия отношения, функции, инъекции, сюръекции и биекции.

Пример 9.

Рассмотрим три функции, заданные на множестве действительных чисел и принимающих значение в этом же множестве:

1. функция $f(x)=e^x$ - инъективна, но не сюръективна;
2. функция $f(x)=x^3-x$ – сюръективна, но не инъективна;
3. функция $f(x)=2x+1$ – биективна.

Определение. ***Суперпозиция функций*** *– функция, полученная из системы функций f, f_1, f_2, …, f_k некоторой подстановкой функций f_1, f_2, …, f_k во внешнюю функцию f вместо переменных и переименованиями переменных.*

Пример 10.

Класс элементарных функций есть множество всех суперпозиций так называемых основных элементарных функций (одноместных: степенных, показательных, логарифмических, тригонометрических и обратных тригонометрических) и двуместных функций, представляющих арифметические операции.

1.5. Свойства бинарных отношений. Специальные бинарные отношения

Определение. Отношение ρ на множестве X называется ***рефлексивным****, если для любого элемента $x \in X$ выполняется $x \rho x$.*

Определение. Отношение ρ на множестве X называется ***симметричным****, если для любых x, $y \in X$ из $x \rho y$ следует $y \rho x$.*

Определение. Отношение ρ на множестве X называется ***транзитивным****, если для любых x, y, $z \in X$ из $x \rho y$ и $y \rho z$ следует $x \rho z$.*

Определение. Рефлексивное, симметричное, транзитивное отношение на множестве X называется отношением ***эквивалентности*** *на множестве X.*

Пример 11.

1. Отношение равенства на множестве целых чисел есть отношение эквивалентности.

2. Отношение подобия на множестве треугольников есть отношение эквивалентности.

3. Отношение «строго меньше» на множестве действительных чисел не рефлексивно, не симметрично и транзитивно на этом множестве.

4. Отношение перпендикулярности прямых не рефлексивно, симметрично, не транзитивно.

Пусть ρ - отношение эквивалентности на множестве X.

Определение. ***Классом эквивалентности****, порожденным элементом x, называется подмножество множества X, состоящее из тех элементов $y\in Y$, для которых $x\rho y$. Класс эквивалентности, порожденный элементом x, обозначается через [x]:* $[x]=\{y \mid y\in X \text{ и } x\rho y\}$.

Определение. Отношение ρ на множестве X называется ***антисимметричным****, если для любых $x, y\in X$ из $x\rho y$ и $y\rho x$ следует $x=y$.*

Определение. Рефлексивное, антисимметричное и транзитивное отношение называется ***отношением частичного порядка*** *на множестве X.*

Пример 12.

1. Отношение «$x \le y$» на множестве действительных чисел есть отношение частичного порядка.

2. Схема организации подчинения в учреждении есть отношение частичного порядка на множестве должностей.

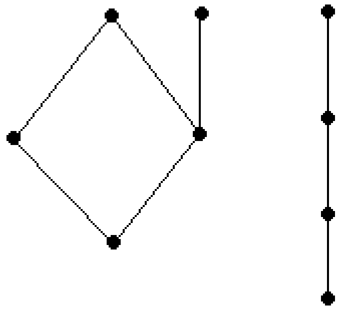


Рис. 10.

Любое частично упорядоченное множество можно представить в виде схемы, в которой каждый элемент изображается точкой на плоскости, и если *y* покрывает *x*, то точки *x* и *y* соединяют отрезком, причем точку, соответствующую *x*, располагают ниже *y*. Такие схемы называются диаграммами Хассе. На рис. 10 показаны две диаграммы Хассе, причем вторая соответствует линейно упорядоченному множеству.

1.6. Операции над бинарными отношениями

Так как отношения на *X* задаются подмножествами $\rho\subseteq X\times Y$, для них определимы те же операции, что и над множествами:

1. *Объединение* $\rho_1\cup\rho_2$: $\rho_1\cup\rho_2=\{(x, y) \mid (x, y)\in\rho_1$ *или* $(x, y)\in\rho_2\}$.
2. *Пересечение* $\rho_1\cap\rho_2$: $\rho_1\cap\rho_2=\{(x, y) \mid (x, y)\in\rho_1$ *и* $(x, y)\in\rho_2\}$.
3. *Разность* $\rho_1\setminus\rho_2$: $\rho_1\setminus\rho_2=\{(x, y) \mid (x, y)\in\rho_1$ *и* $(x, y)\notin\rho_2\}$.
4. *Дополнение* $\overline{\rho}$: $\overline{\rho}=U\setminus\rho$, ãäå $U=M_1\times M_2$ (èëè $U=M^2$).
5. *Обратное отношение* ρ^{-1}: $x\,\rho^{-1}\,y$ тогда и только тогда, когда $y\rho x$, $\rho^{-1}=\{(x, y) \mid (y, x)\in\rho\}$.

Пример 13.

Если ρ - «быть моложе», то ρ^{-1} – «быть старше».

6. *Составное отношение (композиция)* $\rho_1 \bullet \rho_2$. Пусть заданы множества M_1, M_2 и M_3 и отношения $R_1 \subseteq M_1 \times M_2$ и $R_2 \subseteq M_2 \times M_3$. Составное отношение действует из M_1 в M_2 посредством R_1, а затем из M_2 в M_3 посредством R_2, то есть $(a, b) \in R_1 \bullet R_2$, если существует такое $c \in M_2$, что $(a, c) \in R_1$ и $(a, c) \in R_2$.
7. *Транзитивное замыкание* ρ°. Транзитивное замыкание состоит из таких и только таких пар элементов a и b из M, то есть $(a, b) \in \rho^\circ$, для которых в M существует цепочка из $(k+2)$ элементов M, $k \geq 0$, что $a, c_1, c_2, \ldots c_k, b$, между соседними элементами которой выполняется ρ. Другими словами $a \rho c_1$, $c_1 \rho c_2$, ..., $c_k \rho b$.

Пример 14.

Для отношения «быть сыном» транзитивным замыканием является отношение «быть прямым потомком по мужской линии».

1.7. Алгебраические операции

Пусть дано множество M.

Определение. Говорят, что на М определена ***бинарная алгебраическая операция****, если всякой упорядоченной паре элементов множества М по некоторому закону ставится в соответствие вполне определенный элемент этого же множества.*

Примерами бинарных операций на множестве целых чисел являются сложение и умножение. Однако нашему определению не удовлетворяют, например, множество отрицательных чисел относительно умножения и множество действительных чисел относительно деления из-за невозможности деления на нуль.

Среди известных бинарных операций, производимых не над числами, можно отметить векторное умножение векторов пространства, умножение квадратных матриц порядка n, композицию отображений множества X в себя, теоретико-множественное объединение и пересечение множеств.

Таблица 2

	x_1	x_2	x_3	x_4
x_1	x_1	x_2	x_3	x_4
x_2	x_2	x_3	x_1	x_1
x_3	x_2	x_3	x_1	x_2
x_4	x_4	x_2	x_1	x_3

Фактическое задание алгебраической операции на множестве может быть произведено различными методами. Возможно также непосредственное перечисление всех результатов операций для конечных множеств. Его удобно описать с помощью, так называемой таблицы Кэли (табл.2). Слева и сверху квадратной таблицы выписывают все элементы множества. На пересечении строки, соответствующей элементу a, и столбца, соответствующего элементу b, записывают результат операции над a и b.

Будем употреблять следующую терминологию и символику: операцию называть умножением, а результат применения операции к элементам a и b – произведением ab.

Определение. Если для любых элементов a *и* b *множества М справедливо равенство* $ab = ba$, *то операцию называют коммутативной.*

Определение. Если для любых элементов a, b, c множества М справедливо равенство $a(bc) = (ab)c$, то операцию называют ассоциативной.

В ряде случаев множество М, на котором определена алгебраическая операция, обладает единичным элементом, т. е. таким элементом e, что $ae = ea = a$ для всех a из М. Единичный элемент единственен.

Теорема. Если операция, определенная на М, ассоциативна, то результат ее последовательного применения к n элементам множества не зависит от расстановки скобок.

2. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

Математическая логика – современный вид формальной логики, то есть науки, изучающей умозаключения с точки зрения их формального строения.

Вплоть до начала XIX века формальная логика практически не выходила за рамки силлогических умозаключений. Однако, начиная с работ Дж. Буля, можно говорить о превращении ее в математическую логику. Особенности математической логики заключаются в ее математическом аппарате, в преимущественном внимании к умозаключениям, применяемым в самой математике.

Математическая логика – это обширная наука, которая кроме традиционной проблематики занимается вопросами оснований математики и теории алгоритмов и имеет целый ряд приложений.

2. 1. Логика высказываний

2. 1.1. Высказывания. Логические связки

*Определение. Под **высказыванием** принято понимать языковое предложение, о котором имеет смысл говорить, что оно истинно или ложно в данный момент времени.*

Высказывания чаще всего обозначают маленькими латинскими буквами a, b, c, x_1, x_2, …

В логике высказываний интересуются не содержанием, а истинностью или ложностью высказываний. Истинностные значения – истина и ложь – будем обозначать И и Л соответственно. Множество *{И, Л}* называется множеством истинностных значений.

*Определение. Высказывание называют **простым** (элементарным), если оно рассматривается как некое неделимое целое (аналогично элементу множества). **Сложным** (составным) называется высказывание, составленное из простых с помощью логических связок.*

В естественном языке роль связок при составлении сложных предложений из простых играют следующие грамматические средства: союзы «и», «или», «не»; слова «если …, то», «либо … либо», «тогда и только тогда, когда» и др. В логике высказываний логические связки, используемые для составления сложных высказываний, обязаны быть определены точно. Рассмотрим логические связки (операции) над высказываниями, при которых истинностные значения составных высказываний определяются только истинностными значениями составляющих высказываний, а не их смыслом.

В дальнейшем значению «истина» будем ставить в соответствие *1*, а «ложь» - *0*. Каждой логической операции ставится в соответствие ***таблица истинности***. Таблица истинности выражает значения истинности высказываний в зависимости от значений элементарных высказываний. В дальнейшем буден использовать таблицу истинности для установления истинностных значений сложных высказываний при данных значениях входяших в него элементарных высказываний.

Таблица 3

№ набора	a	$\bar{a}$
0	*0*	*1*
1	*1*	*0*

Определение. ***Отрицанием*** *высказывания является новое высказывание, истинное только тогда, когда исходное высказывание ложно (табл. 3).*

Отрицание обозначается через $\bar{a}, \neg a$ и читается как «не *а*», «неверно, что *а*».

Пример 15.

А – «Степан любит танцевать».

Тогда $\bar{a}$ - «Не верно, что Степан любит танцевать».

Таблица 4

№ набора	a	b	$a \wedge b$
0	*0*	*0*	*0*
1	*0*	*1*	*0*
2	*1*	*0*	*0*
3	*1*	*1*	*1*

Определение. ***Конъюнкцией*** *двух высказываний является новое высказывание, которое истинно только тогда, когда оба исходных высказывания истинны (табл. 4).*

Конъюнкция обозначается $a \wedge b$ или $a\&b$ и читается как «*a* и *b*», «*a*, но *b*», «*a*, а *b*».

Пример 16.

a – «Степан любит танцевать»,*b* – «Степан любит петь».

Тогда $a \wedge b$ - «Степан любит танцевать и петь».

Таблица 5

№ набора	a	b	$a \vee b$
0	*0*	*0*	*0*
1	*0*	*1*	*1*
2	*1*	*0*	*1*
3	*1*	*1*	*1*

Определение. ***Дизъюнкцией*** *двух высказываний является новое высказывание, которое ложно только тогда, когда оба исходных высказывания ложны (табл. 5).*

Дизъюнкция обозначается через $a \vee b$ и читается как «*a* или *b*».

Пример 17.

a – «Степан любит танцевать»,*b* – «Степан любит петь».

Тогда $a \vee b$ - «Степан любит танцевать или петь».

Определение. ***Импликацией*** *двух высказываний является новое высказыв ложно только тогда, когда первое истинно, а второе – ложно (табл. 6).*

Таблица 6

№ набора	a	b	a→b
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	1

Импликация обозначается $a \rightarrow b$ и читается как «если *a, тоb»; « из а следует* b». При этом *a* называется посылкой или условием, b – следствием или заключением.

Пример 18.

a – «Степан любит танцевать»,*b* – «Степан любит петь».

Тогда $\grave{a} \rightarrow b$ - «Если Степан любит танцевать, то он любит петь».

Определение. ***Эквиваленцией (или эквивалентностью)*** *двух высказываний является новое высказывание, которое считается истинным, когда оба высказывания либо одновременно истинны, либо одновременно ложны, и ложным во всех остальных случаях (табл. 7).*

Таблица 7

№ набора	a	b	a≈b
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

Эквивалентность обозначается $a \approx b$и читается как «*a эквивалентно* b*»*.

Пример 19.

a – «Степан любит танцевать»,*b* – «Степан любит петь».

Тогда $\grave{a} \approx b$ - «Для того, чтобы Степан любил танцевать, необходимо и достаточно, чтобы он любил петь».

Сведем все сказанное выше в единую таблицу и введем в рассмотрение еще три операции: сумма по модулю два, штрих Шеффера, стрелка Пирса (табл. 8).

Таблица 8

Обозначения логической операции	Другие обозначения логической операции	Набор истинностных значений, отвечающих данной логической операции	Названия логической операции и связки	Как читается выражение, приведенное в первом столбце
$\neg a$		*10*	*отрицание*	*неверно, что a; не a*
$a \wedge b$	$a \,\&\, b$ $a \cdot b$ ab $min(a; b)$	*0001*	*конъюнкция, логическое умножение, логическое «и»*	*a и b*

$a \vee b$	$a+b$ $max(a; b)$	*0111*	*дизъюнкция, логическое сложение, логическое «или»*	*a или b*
$a \rightarrow b$	$a \supset b$ $a \Rightarrow b$	*1101*	*импликация, логическое следование*	*если a, то b;* *a имплицирует b; a влечет b*
$a \approx b$	$a \equiv b$ $a \leftrightarrow b$ $a \Leftrightarrow b$	*1001*	*эквиваленция, эквивалентность, равнозначность, тождественность*	*a тогда и только тогда, когда b; a эквивалентно b*
$a \oplus b$	$a + b$ $a \Delta b$	*0110*	*сумма по модулю два, разделительная дизъюнкция, разделительное «или»*	*a плюс b;* *либо a, либо b*
$a \mid b$		*1110*	*штрих Шеффера, антиконъюнкция*	*неверно, что a и b; a штрих Шеффера b*
$a \downarrow b$	$a \circ b$	*1000*	*стрелка Пирса, антидизъюнкция, функция Вебба, функция Даггера*	*ни a, ни b; a стрелка Пирса b*

2.1.2. Формулы логики высказываний

Определим понятие формулы логики высказываний.

Определение. ***Алфавитом*** *называется любое непустое множество. Элементы этого множества называются* ***символами*** *данного алфавита.* ***Словом*** *в данном алфавите называется произвольная конечная последовательность символов (возможно пустая).*

Алфавит логики высказываний содержит следующие символы:

- высказывательные переменные;
- логические символы;
- символы скобок.

Определение. Слово в алфавите логики высказываний называется ***формулой****, если оно удовлетворяет следующему определению:*

1) *любая высказывательная переменная – формула;*
2) *если A и B – формулы, то* $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \oplus B$, $A \approx B$, $A \mid B$, $A \downarrow B$ *– формулы;*
3) *только те слова являются формулами, для которых это следует из 1) и 2).*

Определение. ***Подформулой*** *формулы A называется любая ее часть, которая сама является формулой.*

Пример 20.

Представить логическими формулами следующие высказывания:

1. «Сегодня понедельник или вторник».
2. «Идет снег или дождь».
3. «Если идет дождь, то крыши мокрые. Дождя нет, а крыши мокрые».
4. «Что в лоб, что по лбу».

Решение.

1. Составное (сложное) высказывание «Сегодня понедельник или вторник» состоит из двух простых:

- ✓ a – «сегодня понедельник»;
- ✓ b – «сегодня вторник».

Высказывания a и b соединены связкой «или» очевидно в разделительном смысле (не допускается одновременное выполнение обоих условий), то есть используется логическая связка «сумма по модулю два». Таким образом, данное высказывание представимо логической формулой: $a \oplus b$.

2. Высказывание «Идет снег или дождь» также состоит из двух простых, соединенных связкой «или»:

- ✓ a – «идет снег»;
- ✓ b – «идет дождь».

Но в отличие от предыдущего связка «или» использована здесь не в разделительном смысле, поэтому – используется логическая связка дизъюнкция и логическая формула имеет вид: $a \vee b$.

3. Сложное высказывание «Если идет дождь, то крыши мокрые. Дождя нет, а крыши мокрые» включает два простых высказывания:

- ✓ a – «идет дождь»;
- ✓ b – «крыши мокрые».

В первом предложении «Если идет дождь, то крыши мокрые» высказывания a, b соединены связкой «если …, то…»: $a \to b$.

Во втором «Дождя нет, а крыши мокрые» союз «а» здесь имеет смысл связки «и» ($\wedge$), и кроме того высказывание a следует взять с отрицанием: $\bar{a} \wedge b$.

Остается объединить представленные выше два высказывания в одно связкой $\wedge$:

$$(a \to b) \wedge (\bar{a} \wedge b).$$

4. Высказывание «Что в лоб, что по лбу», если обозначить:

- ✓ a – «в лоб»,
- ✓ b – «по лбу»,

представимо логической формулой $a \approx b$.

Пример 21.

Представить логической формулой следующий текст:

«Если фирма продолжает выпуск существующего продукта и ориентирована на существующий рынок, то для нее целесообразна стратегия «малого корабля», или экономии издержек. Такая стратегия привлекательна, если интенсивный маркетинг – стратегический хозяйственный фактор, но слабая сторона организации. Если интенсивный маркетинг является стратегическим хозяйственным фактором и сильной стороной фирмы, то фирме следует придерживаться стратегии захвата новых рынков для существующего продукта.»

Введем обозначения простых высказываний, содержащихся в первом предложении:

A– «фирма продолжает выпуск существующего продукта»;

B– «фирма ориентирована на существующий рынок;

C– «для фирмы целесообразна (привлекательна) стратегия «малого корабля»;

D– «для фирмы целесообразна (привлекательна) стратегия экономии издержек»;

С учетом введенных обозначений логическая формула для первого предложения примет

вид:

$$(A \wedge B) \rightarrow (C \sim D).$$

Второе предложение содержит новые простые высказывания:

K – «интенсивный маркетинг является стратегическим хозяйственным фактором организации»;

L – «интенсивный маркетинг является слабой стороной организации».

Логическая формула, представляющая второе предложение:

$$(K \wedge L) \rightarrow (C \sim D).$$

В третьем предложении содержатся новые простые высказывания:

M – «интенсивный маркетинг является сильной стороной организации»;

N – «фирме следует придерживаться стратегии захвата новых рынков для существующего продукта».

Логическая формула для третьего предложения:

$$(K \wedge M) \rightarrow N.$$

Окончательно текст записывается следующей логической формулой:

$$((A \wedge B) \rightarrow (C \sim D)) \wedge ((K \wedge L) \rightarrow (C \sim D)) \wedge ((K \wedge M) \rightarrow N).$$

Для каждой формулы логики высказываний можно построить таблицу истинности.

*Определение. Формула называется **выполнимой (опровержимой)**, если существует такой набор значений переменных, при которых эта формула принимает значение 1 (0).*

Определение. Формула называется ***тождественно-истинной****, или* ***тавтологией*** *(****тождественно-ложной*** *или* ***противоречием****), если эта формула принимает значение 1 (0) при всех наборах значений переменных.*

Пример 22.

Составить таблицы истинности для формул:

1. $(x \wedge y) \vee x$;
2. $(x \to \overline{y}) \to (\overline{x \vee y} \wedge z)$.

Решение.

1. Таблица истинности для формулы $(x \wedge y) \vee x$ имеет вид (табл. 9):

Таблица 9

№	x	y	$x \wedge y$	$(x \wedge y) \vee x$
0	0	0	0	0
1	0	1	0	0
2	1	0	0	1
3	1	1	1	1

2. Таблица истинности для формулы $(x \to \overline{y}) \to (\overline{x \vee y} \wedge z)$ имеет вид (табл. 10):

Таблица 10

№	x	y	z	$\overline{y}$	$x \to \overline{y}$	$x \vee y$	$\overline{x \vee y}$	$\overline{x \vee y} \wedge z$	$(x \to \overline{y}) \to (\overline{x \vee y} \wedge z)$
0	0	0	0	1	1	0	1	0	0
1	0	0	1	1	1	0	1	1	1
2	0	1	0	0	1	1	0	0	0
3	0	1	1	0	1	1	0	0	0
4	1	0	0	1	1	1	0	0	0
5	1	0	1	1	1	1	0	0	0
6	1	1	0	0	0	1	0	0	1
7	1	1	1	0	0	1	0	0	1

2.1.3. Равносильность формул логики высказываний

Определение. Пусть А и В – две формулы, зависящие от одного и того же списка переменных. Будем называть их ***равносильными****, если для любого набора значений переменных они принимают одинаковые значения.*

Рассмотрим основные равносильности логики высказываний.

Пусть *A, B, C* – произвольные формулы. Тогда справедливы следующие свойства логических операций (табл. 11):

Таблица 11

1. Идемпотентность	
$A \wedge A = A$	$A \vee A = A$
2. Коммутативность	
$A \wedge B = B \wedge A$	$A \vee B = B \vee A$
3. Ассоциативность	
$A \wedge (B \wedge C) = (A \wedge B) \wedge C$	$A \vee (B \vee C) = (A \vee B) \vee C$
4. Правила поглощения	
$A \wedge (A \vee B) = A$	$A \vee (A \wedge B) = A$
5. Дистрибутивность	
$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
6. Правила де Моргана	
$\overline{\grave{A} \wedge \hat{A}} = \overline{\grave{A}} \vee \overline{\hat{A}}$	$\overline{\grave{A} \vee \hat{A}} = \overline{\grave{A}} \wedge \overline{\hat{A}}$
7. Свойства констант	
$A \wedge 1 = A$	$A \vee 0 = A$
$A \wedge 0 = 0$	$A \vee 1 = 1$
8. Закон исключения третьего и закон противоречия	
$\grave{A} \wedge \overline{\grave{A}} = 0$	$\grave{A} \vee \overline{\grave{A}} = 1$
9. Снятие двойного отрицания	
$\overline{\overline{\grave{A}}} = \grave{A}$	
10. Формулы расщепления (склеивания)	
$(\grave{A} \wedge \hat{A}) \vee (\grave{A} \wedge \overline{\hat{A}}) = \grave{A}$	$(\grave{A} \vee \hat{A}) \wedge (\grave{A} \vee \overline{\hat{A}}) = \grave{A}$
11. Связь дизъюнкции, конъюнкции, отрицания и импликации	
$\grave{A} \to \hat{A} = \overline{\grave{A}} \vee \hat{A} = \overline{\grave{A} \wedge \overline{\hat{A}}} = \overline{\hat{A}} \to \overline{\grave{A}}$	
12. Выражение эквивалентности	
$\grave{A} \approx \hat{A} = (\overline{\grave{A}} \vee \hat{A}) \wedge (\overline{\hat{A}} \vee \grave{A}) = (\grave{A} \wedge \hat{A}) \vee (\overline{\grave{A}} \wedge \overline{\hat{A}})$	

Любая из этих равносильностей легко может быть доказана с помощью таблицы истинности.

Пример 23.

Рассмотрим одно из правил поглощения $A \wedge (A \vee B) = A$.

Таблица 12

A	*B*	$A \vee B$	$A \wedge (A \vee B)$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

По таблице 12 видно, что результирующий столбец и столбец *A* совпадают на все наборах. Значит, формулы действительно равносильны.

Однако часто равносильность экономнее доказывать без составления полной таблицы истинности, а с помощью приведенных выше равносильностей.

Пример 24.

1. Доказать равносильность формулы, используя логические законы: $\overline{\grave{a}\rightarrow b}\equiv a\wedge\overline{b}$.

2. Упростить формулу: $\left(\overline{\overline{x}\wedge\overline{y}}\vee\overline{x}\right)\wedge\overline{x\vee\overline{\overline{\overline{x}\wedge y}}}$.

3. Определить, является ли формула тавтологией, противоречием или ни тем, ни другим:

   a) $a\wedge\left(\overline{a\vee b}\right)$;

   b) $a\rightarrow\left(a\wedge b\right)$;

   c) $\left(\left(a\rightarrow b\right)\wedge b\right)\rightarrow a$;

   d) $\left(a\rightarrow b\right)\rightarrow a$;

   e) $\overline{a}\rightarrow\left(a\rightarrow b\right)$.

Решение.

1. $\overline{\grave{a}\rightarrow b}\equiv|11|\equiv\overline{\overline{a}\vee b}\equiv|6|\equiv\overline{\overline{a}}\wedge\overline{b}\equiv|9|\equiv a\wedge\overline{b}$.

2. $\left(\overline{\overline{x}\wedge\overline{y}}\vee\overline{x}\right)\wedge\overline{x\vee\overline{\overline{\overline{x}\wedge y}}}\equiv|6|\equiv\left(\overline{\overline{x}}\vee\overline{\overline{y}}\vee\overline{x}\right)\wedge\left(\overline{x}\wedge\overline{\overline{\overline{x}\wedge y}}\right)\equiv|9|\equiv\left(x\vee y\vee\overline{x}\right)\wedge\left(\overline{x}\wedge\overline{x}\wedge y\right)\equiv|2,1|\equiv$

$\equiv\left(\left(x\vee\overline{x}\right)\vee y\right)\wedge\left(\overline{x}\wedge y\right)\equiv|8,3|\equiv\left(1\vee y\right)\wedge\overline{x}\wedge y\equiv|7|\equiv1\wedge\overline{x}\wedge y\equiv|7|\equiv\overline{x}\wedge y.$

3. a) $a\wedge\left(\overline{a\vee b}\right)\equiv|6|\equiv a\wedge\left(\overline{a}\wedge\overline{b}\right)\equiv|3|\equiv a\wedge\overline{a}\wedge\overline{b}\equiv|8|\equiv0\wedge\overline{b}\equiv|7|\equiv0$

Поскольку формула при любых значениях переменных равна нулю, то данная формула является противоречием.

b) $a\rightarrow\left(a\wedge b\right)\equiv|11|\equiv\overline{a}\vee\left(a\wedge b\right)\equiv|5|\equiv\left(\overline{a}\vee a\right)\wedge\left(\overline{a}\vee b\right)\equiv|8|\equiv1\wedge\left(\overline{a}\vee b\right)\equiv|7|\equiv\overline{a}\vee b$.

Исходная формула не является ни тавтологией, ни противоречием, поскольку её значение зависит от значений переменных.

c) $\left(\left(a\rightarrow b\right)\wedge b\right)\rightarrow a\equiv|11|\equiv\left(\overline{\left(\overline{a}\vee b\right)\wedge b}\right)\vee a\equiv|6|\equiv\left(\overline{\overline{a}\vee b}\vee\overline{b}\right)\vee a\equiv|6,3|\equiv\left(\overline{\overline{a}}\wedge\overline{b}\right)\vee\overline{b}\vee a\equiv|4|\equiv$

$\equiv\overline{b}\vee a.$

Исходная формула не является ни тавтологией, ни противоречием, поскольку её значение зависит от значений переменных.

d) $\left(a\rightarrow b\right)\rightarrow a\equiv|11|\equiv\left(\overline{\overline{a}\vee b}\right)\vee a\equiv|6,9|\equiv\left(a\wedge\overline{b}\right)\vee a\equiv|4|\equiv a.$

Исходная формула не является ни тавтологией, ни противоречием, поскольку её значение зависит от значений переменных.

e) $\overline{a}\rightarrow\left(a\rightarrow b\right)\equiv|11,3|\equiv\overline{\overline{a}}\vee\overline{a}\vee b\equiv|9|\equiv a\vee\overline{a}\vee b\equiv|8,7|\equiv1.$

Поскольку формула при любых значениях переменных равна единице, то данная формула является тавтологией.

2.1.4. Нормальные формы

Определение. ***Элементарной конъюнкцией*** *называется конъюнкция, составленная из попарно различных переменных или отрицаний переменных.*

Иногда будем допускать в элементарной конъюнкции наличие повторов элементов.

Определение. ***Дизъюнктивной нормальной формой (ДНФ)*** *называется дизъюнкция попарно различных элементарных конъюнкций.*

Иногда будем допускать в ДНФ наличие повторов элементов.

Пример 25.

Следующие формулы находятся в ДНФ: $x \wedge y$; $\overline{x_1} \wedge x_2 \wedge \overline{x_3}$; $ab \vee a\bar{b}c \vee \bar{a}c$; $x \vee xy$.

Определение. ***Элементарной дизъюнкцией*** *называется дизъюнкция, составленная из попарно различных переменных или отрицаний переменных.*

Иногда будем допускать в элементарной дизъюнкции наличие повторов элементов.

Определение. ***Конъюнктивной нормальной формой (КНФ)*** *называется конъюнкция попарно различных элементарных дизъюнкций.*

Иногда будем допускать в КНФ наличие повторов элементов.

Пример 26.

Следующие формулы находятся в КНФ: $x \vee \bar{y}$; $(x \vee y)(x \vee \bar{y} \vee z)(\bar{x} \vee y \vee \bar{z})$.

2.1.5.Совершенные нормальные формы

Определение. Совершенной дизъюнктивной формулой формулы алгебры высказываний (СДНФ) **называется ДНФ, в которой:**

1. **различны все члены дизъюнкции;**
2. *различны все члены каждой конъюнкции;*
3. *ни одна конъюнкция не содержит одновременно переменную и отрицание этой переменной;*
4. *каждая конъюнкция содержит все переменные, входящие в формулу, т. е. имеет вид*

$$F(x_1, x_2, ..., x_n) = \bigvee_{F(c_1, ..., c_n)=1} x_1^{c_1} ... x_n^{c_n},$$

где дизъюнкция берется по всем наборам $\boldsymbol{c}=(c_1, c_2, ..., c_n)$ *из 0 и 1, для которых* $F(\boldsymbol{c})=1$.

Теорема (о СДНФ). Для всякой не равной тождественному нулю формулы логики высказываний $F(x_1, x_2, ..., x_n)$ *существует такая формула* F_1*, зависящая от того же списка переменных и находящаяся в СДНФ относительно этого списка, что* F_1 *выражает собой формулу* F*. Формула* F_1 *определена однозначно с точностью до перестановки дизъюнктивных членов.*

Определение. ***Совершенной конъюнктивной формулой формулы алгебры высказываний (СКНФ)*** *называется КНФ, в которой:*

1. *различны все члены конъюнкции;*
2. *различны все члены каждой дизъюнкции;*
3. *ни одна дизъюнкция не содержит переменную вместе с отрицанием этой переменной;*
4. *каждая дизъюнкция содержит все переменные, входящие в исходную формулу, т. е. имеет вид*

$$F(x_1, x_2, ..., x_n) = \bigwedge_{F(c_1, ..., c_n)=0} (x_1^{\overline{\tilde{n}_1}} \vee ... \vee x_n^{\overline{c_n}}),$$

где конъюнкция берется по всем наборам $\boldsymbol{c}=(c_1, c_2, ..., c_n)$ *из 0 и 1, для которых* $F(\boldsymbol{c})=0$.

Теорема (о СКНФ). Для всякой не равной тождественной единице формулы логики высказываний $F(x_1, x_2, \ldots, x_n)$ существует такая формула F_1, зависящая от того же списка переменных и находящаяся в СКНФ относительно этого списка, что F_1 выражает собой формулу F. Формула F_1 определена однозначно с точностью до перестановки конъюнктивных членов.

Опишем два способа приведения к совершенным нормальным формам.

1-й способ – аналитический.

Приведение к СДНФ. Алгоритм приведения.

1. привести формулу с помощью равносильных преобразований к ДНФ.
2. удалить члены дизъюнкции, содержащие переменную вместе с ее отрицанием (если такие окажутся);
3. из одинаковых членов дизъюнкции (если такие окажутся) удалить все, кроме одного;
4. из одинаковых членов каждой конъюнкции (если такие окажутся) удалить все, кроме одного;
5. если в какой-нибудь конъюнкции не содержится переменной x_i из числа переменных, входящих в исходную формулу, добавить к этой конъюнкции член $x_i \vee \overline{x_i}$ и применить закон дистрибутивности конъюнкции относительно дизъюнкции;
6. если в полученной дизъюнкции окажутся одинаковые члены, воспользоваться предписанием из п. 3.

Полученная формула и является СДНФ данной формулы.

Пример 27.

Привести следующие формулы к СДНФ с помощью равносильных преобразований:

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

2. $x(\overline{y} \vee z)$;

3. $(x \to y)xy$.

Решение.

1. $(x \vee y)(x \vee \overline{y}) \equiv x \equiv x(y \vee \overline{y}) \equiv xy \vee x\overline{y}$.

2. $x(\overline{y} \vee z) \equiv x\overline{y} \vee xz \equiv x\overline{y}(z \vee \overline{z}) \vee xz(y \vee \overline{y}) \equiv |5| \equiv x\overline{y}z \vee x\overline{y}\overline{z} \vee xzy \vee xz\overline{y} \equiv x\overline{y}z \vee x\overline{y}\overline{z} \vee xyz$.

3. $(x \to y)xy \equiv (\overline{x} \vee y)xy \equiv \overline{x}xy \vee yxy \equiv xy$.

Приведение к СКНФ. Алгоритм приведения.

1. привести формулу с помощью равносильных преобразований к КНФ.
2. удалить члены конъюнкции, содержащие переменную вместе с ее отрицанием (если такие окажутся);
3. из одинаковых членов конъюнкции (если такие окажутся) удалить все, кроме одного;

4. из одинаковых членов каждой дизъюнкции (если такие окажутся) удалить все, кроме одного;
5. если в какой-нибудь дизъюнкции не содержится переменной x_i из числа переменных, входящих в исходную формулу, добавить к этой дизъюнкции член $x_i \wedge \overline{x_i}$ и применить закон дистрибутивности дизъюнкции относительно конъюнкции;
6. если в полученной конъюнкции окажутся одинаковые члены, воспользоваться предписанием из п. 3.

Полученная формула и является СКНФ данной формулы.

Пример 28.

Привести следующие формулы к СКНФ с помощью равносильных преобразований:

1. $x\left(\overline{y} \vee z\right)$;
2. $(x \to y)xy$.

Решение.

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

2. $$(x \to y)xy \equiv \left(\overline{x} \vee y\right)xy \equiv \left(\overline{x} \vee y\right)\left(x \vee y\overline{y}\right)\left(y \vee x\overline{x}\right) \equiv \left(\overline{x} \vee y\right)(x \vee y)\left(x \vee \overline{y}\right)(y \vee x)\left(y \vee \overline{x}\right) \equiv$$
$$\equiv (x \vee y)\left(x \vee \overline{y}\right)\left(\overline{x} \vee y\right).$$

2-й способ – табличный.

Составляем таблицу истинности для данной функции.

Приведение к СДНФ. Алгоритм приведения.

Строим таблицу значений формулы. Рассматриваем только те строки, в которых значение формулы равно единице. Каждой такой строке соответствует конъюнкция всех аргументов (без повторений). Причем, аргумент, принимающий значение 0, входит в нее с отрицанием, значение 1 – без отрицания. Наконец, образуем дизъюнкцию всех полученных конъюнкций.

Пример 29.

Построить СДНФ для данных формул логики высказываний.

1. $F = x\left(\overline{y} \vee z\right)$.
2. $F = (x \to y)xy$.

Решение.

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

Строим таблицу истинности (табл. 13) для формулы F:

Таблица 13

№	x	y	z	$\overline{y}$	$\overline{y} \vee z$	$F = x(\overline{y} \vee z)$
0	0	0	0	1	1	0
1	0	0	1	1	1	0
2	0	1	0	0	0	0
3	0	1	1	0	1	0
4	1	0	0	1	1	1
5	1	0	1	1	1	1
6	1	1	0	0	0	0
7	1	1	1	0	1	1

Рассматриваем только 4, 5 и 7 наборы, так как только на этих наборах формула принимает значение равное единице.

СДНФ имеет вид: $F = x\overline{y}\overline{z} \vee x\overline{y}z \vee xyz$.

2. 2. $F = (x \to y)xy$.

Строим таблицу истинности (табл. 14) для формулы F:

Таблица 14

№	x	y	$x \to y$	$F=(x \to y) \wedge x \wedge y$
0	0	0	1	0
1	0	1	1	0
2	1	0	0	0
3	1	1	1	1

СДНФ (1): *№ 3*:

$F = x\,y$.

Приведение к СКНФ. Алгоритм приведения.

Рассматриваем только те строки таблицы, где формула принимает значение 0. Каждой такой строке соответствует дизъюнкция всех переменных (без повторений). Причем аргумент, принимающий значение 0, берется без отрицания, значение 1 – с отрицанием. Наконец, образуют конъюнкцию полученных дизъюнкций.

Пример 30.

Построить СКНФ для данных формул логики высказываний.

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

2. $F = (x \to y)xy$.

Решение.

1. Строим таблицу значений, используя предыдущий пример (табл. 15).

Таблица 15

№	*x*	*y*	*z*	$F = x(\bar{y} \vee z)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Рассматриваем только наборы, на которых формула принимает значение ноль.

СКНФ (0): № *0, 1, 2, 3, 6*:

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

2. Строим таблицу значений, используя предыдущий пример (табл. 16).

Таблица 16

№	*x*	*y*	$F=(x \to y) \wedge x \wedge y$
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

СКНФ (0): *№ 0, 1, 2*:

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

2.2. Булевы функции

2.2.1. Представление булевой функции формулой логики высказываний

Определение. ***Булевой функцией*** *$f(x_1, x_2, \ldots, x_n)$ называется n-местная функция, аргументы которой принимают значения во множестве {0, 1} и сама функция принимает значения в этом же множестве.*

Всякую булеву функцию от nпеременных можно задать таблицей из 2^n строк, в которой в каждой строке записывают одну из оценок списка переменных, принимающих значение 0 или 1.

Пример 31.

Для $n=3$ булеву функцию можно задать таблицей 17.

Таблица 17

№	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	$f(0, 0, 0)$
1	0	0	1	$f(0, 0, 1)$
2	0	1	0	$f(0, 1, 0)$
3	0	1	1	$f(0, 1, 1)$
4	1	0	0	$f(1, 0, 0)$
5	1	0	1	$f(1, 0, 1)$
6	1	1	0	$f(1, 1, 0)$
7	1	1	1	$f(1, 1, 1)$

Используется также задание булевой функции в виде двоичного слова, длина которого зависит от числа переменных.

Пример 32.

Пусть задана булева функция от трех переменных (табл. 18). Тогда число наборов $2^3 = 8$.

Таблица 18

№ набора	x_1	x_2	x_3	f
0	*0*	*0*	*0*	*0*
1	*0*	*0*	*1*	*0*

2	*0*	*1*	*0*	*1*
3	*0*	*1*	*1*	*0*
4	*1*	*0*	*0*	*1*
5	*1*	*0*	*1*	*1*
6	*1*	*1*	*0*	*0*
7	*1*	*1*	*1*	*1*

Номера наборов всегда нумеруются, начиная с нуля, в таблице приведено стандартное расположение всех наборов функции трех переменных (обратите внимание, что каждый набор представляет собой двоичный код числа, равный номеру соответствующего набора). Первые четыре столбца одинаковы для всех булевых функций от трех переменных. Столбец значений функции задается или вычисляется.

Эту же функцию можно записать $f(x_1, x_2, x_3)=00101101$.

Существует ровно 2^{2^n} различных булевых функций от n переменных. Константы 0 и 1 считают нуль-местными булевыми функциями.

Утверждение. Каждой формуле логики высказываний соответствует некоторая булева функция.

Пример 33.

Построить все булевы функции, зависящие от двух переменных.

Решение.

Поскольку $n=2$, различных булевых функций от двух переменных существует ровно 16 (табл. 19).

Таблица 19

№ функции	*Значение функции*	*Формула, соответствующая функции*
1	$f=0000$	$f=0$
2	$f=0001$	$f=x_1 \wedge x_2$
3	$f=0010$	$f=\overline{x_1 \to x_2}$
4	$f=0011$	$f=x_1$
5	$f=0100$	$f=\overline{x_1} \wedge x_2$

6	f=0101	$f=x_2$
7	f=0110	$f=x_1 \oplus x_2$
8	f=0111	$f=x_1 \vee x_2$
9	f=1000	$f=\overline{x_1 \wedge x_2}$
10	f=1001	$f=x_1 \approx x_2$
11	f=1010	$f=\overline{x_2}$
12	f=1011	$f=x_1 \vee \overline{x_2}$
13	f=1100	$f=\overline{x_1}$
14	f=1101	$f=x_1 \rightarrow x_2$
15	f=1110	$f=\overline{x_1 \vee x_2}$
16	f=1111	$f=1$

Теорема. Пусть $f(x_1, x_2, \ldots, x_k)$ *k-местная булева функция. Если f не равна тождественно нулю, то существует такая формула F, зависящая от списка переменных* $x_1, x_2, \ldots, x_n$ *и находящаяся в СДНФ относительно этого списка, что F выражает собой функцию f. Формула F определена однозначно с точностью до перестановки дизъюнктивных членов.*

Теорема 2. Пусть $f(x_1, x_2, \ldots, x_k)$ *k-местная булева функция. Если f не равна тождественно единице, то существует такая формула F, зависящая от списка переменных* $x_1, x_2, \ldots, x_k$ *и находящаяся в СКНФ относительно этого списка, что F выражает собой функцию f. Формула F определена однозначно с точностью до перестановки конъюнктивных членов.*

Поскольку каждая булева функция представима в виде формулы логики высказываний, то принцип построения СДНФ и СКНФ сохраняется такой же как и для формул логики высказываний.

Пример 34.

Построить СКНФ и СДНФ булевой функции $f(x_1, x_2, x_3)= 00101110$.

Решение.

Строим таблицу значений функции (табл. 20):

Таблица 20

№	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0

1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

СКНФ (0): № 0, 1, 3, 7

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \overline{x_3})(x_1 \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

СДНФ (1): № 2, 4, 5, 6

$$f(x_1, x_2, x_3) = \overline{x_1}\, x_2\, \overline{x_3} \vee x_1\, \overline{x_2}\, \overline{x_3} \vee x_1\, \overline{x_2}\, x_3 \vee x_1\, x_2\, \overline{x_3}.$$

2.2.2. Минимизация нормальных форм

Выше было сказано, что произвольная булева функция может быть представлена формулой в дизъюнктивной и конъюнктивной нормальной форме. Равносильными преобразованиями можно получить формулу, содержащую меньшее, чем исходная, число переменных.

Определение. ***Минимальной ДНФ (МДНФ)*** *функции $f(x_1, x_2, \ldots, x_n)$ называется ДНФ, реализующая функцию f и содержащая минимальное число символов переменных по сравнению со всеми другими ДНФ, реализующими функцию f.*

Минимальную ДНФ данной формулы можно найти, перебрав конечное число равносильных ей ДНФ и выбрав среди них ту, которая содержит минимальное число переменных. Однако при большом числе переменных такой перебор практически невыполним. Существуют эффективные способы нахождения минимальной ДНФ. Рассмотрим два из них.

Каждый из рассмотренных ниже методов состоит из двух этапов:

- построение сокращенной ДНФ;
- построение матрицы покрытий. Построение МДНФ.

Определение. Если для всякого набора $a = (a_1, a_2, \ldots, a_n)$ значений переменных условие $g(a) = 1$ влечет $f(a) = 1$, то функция g называется ***частью функции*** *f (или функция* ***f накрывает функцию g****). Если при этом для некоторого набора $c = (c_1, c_2, \ldots, c_n)$ функция $g(c)=1$, то говорят, что функция g накрывает единицу функции f на наборе c (или что g накрывает конституенту единицы $x_1^{c_1} \ldots x_n^{c_n}$ функции f).*

Конституента единицы функции *f* есть часть функции *f*, накрывающая единственную единицу функции *f*.

Определение. Элементарная конъюнкция К называется ***импликантом*** *функции f, если для всякого набора a = (a_1, a_2, …, a_n) из 0 и 1 условие К(a)=1 влечет f(a)=1.*

Определение. Импликант К функции f называется простым, если выражение, получающееся из него выбрасыванием любых множителей, уже не импликант функции f.

Всякий импликант функции *f*есть часть функции *f*.

Теорема. Всякая функция реализуется дизъюнкцией всех своих простых импликант.

Определение. ***Сокращенная ДНФ*** *функции f есть дизъюнкция всех простых импликант функции f.*

Утверждение. Всякая функция f реализуется своей сокращенной ДНФ. Для всякой функции, не равной тождественно нулю, существует единственная сокращенная ДНФ.

Теорема (Куайна). Если в СДНФ функции f провести все операции неполного склеивания, а затем все операции поглощения и удаления дублирующих членов, то в результате получится сокращенная ДНФ функции f.

2.2.2.1. Алгоритм Куайна построения сокращенной ДНФ

1. Получить СДНФ функции.
2. Провести все операции неполного склеивания.
3. Провести все операции поглощения.

Пример 35.

Минимизировать функцию *f=1111010010101111*.

Решение.

1. Строим таблицу значения для данной функции (табл. 20). Строим СДНФ функции. При этом слагаемые нумеруем и записываем в столбец (табл. 21).

Таблица 20

№	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0

8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

СДНФ (1): № 0, 1, 2, 3, 5, 8, 10, 12, 13, 14, 15.

Таблица 21

№ слагаемого	слагаемое
1	$\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}$
2	$\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,x_4$
3	$\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}$
4	$\overline{x_1}\,\overline{x_2}\,x\ x_4$
5	$\overline{x_1}\,x_2\,\overline{x_3}\,x_4$
6	$x_1\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}$
7	$x_1\,\overline{x_2}\,x_3\,\overline{x_4}$
8	$x_1\,x_2\,\overline{x_3}\,\overline{x_4}$
9	$x_1x_2\overline{x_3}\,x_4$
10	$x_1x_2x_3\overline{x_4}$
11	$x_1x_2x_3x_4$

2. Проводим все операции неполного склеивания.
   Первый этап склеивания (табл. 22):

Таблица 22

Слагаемые	*Склеивание по*	*Результат*	*Новые слагаемые*
1, 2	x_4	$\overline{x_1}\,\overline{x_2}\,\overline{x_3}$	*1*
1, 3	x_3	$\overline{x_1}\,\overline{x_2}\,\overline{x_4}$	*2*
1, 6	x_1	$\overline{x_2}\,\overline{x_3}\,\overline{x_4}$	*3*
2, 4	x_3	$\overline{x_1}\,\overline{x_2}\,x_4$	*4*
2, 5	x_2	$\overline{x_1}\,\overline{x_3}x_4$	***5***

3, 4	x_4	$\overline{x_1}\,\overline{x_2}\,x_3$	*6*
3, 7	x_1	$\overline{x_2}\,x_3\,\overline{x_4}$	*7*
5, 9	x_1	$x_2\,\overline{x_3}x_4$	*8*
6, 7	x_3	$x_1\overline{x_2}\,\overline{x_4}$	*9*
6, 8	x_2	$x_1\overline{x_3}\,\overline{x_4}$	*10*
7, 10	x_2	$x_1x_3\overline{x_4}$	*11*
8, 9	x_4	$x_1x_2\,\overline{x_3}$	*12*
8, 10	x_3	$x_1x_2\,\overline{x_4}$	*13*
9, 11	x_3	$x_1x_2\,x_4$	*14*
10, 11	x_4	$x_1x_2x_3$	*15*

В первом этапе склеивания участвовали все слагаемые СДНФ, значит, ни одно из исходных слагаемых не войдут в сокращенную ДНФ. После первого этапа склеивания (и возможных поглощений) получаем, что

$$f(x_1,x_2,x_3,x_4)=\overline{x_1}\,\overline{x_2}\,\overline{x_3}\vee\overline{x_1}\,\overline{x_2}\,\overline{x_4}\vee\overline{x_2}\,\overline{x_3}\,\overline{x_4}\vee\overline{x_1}\,\overline{x_2}x_4\vee\overline{x_1}\,\overline{x_3}x_4\vee\overline{x_1}\,\overline{x_2}x_3\vee\overline{x_2}x_3\overline{x_4}\vee x_1\overline{x_2}\,\overline{x_4}\vee$$
$$\vee x_2\overline{x_3}x_4\vee x_1\overline{x_3}\,\overline{x_4}\vee x_1x_3\overline{x_4}\vee x_1x_2\overline{x_3}\vee x_1x_2\overline{x_4}\vee x_1x_2x_4\vee x_1x_2x_3.$$

Пронумеруем дизъюнктивные члены в полученной ДНФ в порядке их следования от 1 до 15 (табл. 23).

Второй этап склеивания:

Таблица 23

Слагаемые	*Склеивание по*	*Результат*
1, 6	x_3	$\overline{x_1}\,\overline{x_2}$
2, 4	x_4	$\overline{x_1}\,\overline{x_2}$
2, 9	x_1	$\overline{x_2}\,\overline{x_4}$
3, 7	x_3	$\overline{x_2}\,\overline{x_4}$
9, 13	x_2	$x_1\overline{x_4}$

10, 11	x_3	$x_1\overline{x_4}$
12, 15	x_3	x_1x_2
13, 14	x_4	x_1x_2

В процедуре склеивания на втором этапе не принимали участие слагаемые № 5, 8 с предыдущего шага, поэтому после второго этапа склеивания и последующих поглощений получаем, что

$$f(x_1,x_2,x_3,x_4)=x_1x_2\vee\overline{x_1}\,\overline{x_2}\vee\overline{x_2}\,\overline{x_4}\vee x_1\overline{x_4}\vee\overline{x_1}\,\overline{x_3}\,x_4\vee x_2\overline{x_3}\,x_4.$$

Поскольку дальнейшее склеивания невозможны, то это и будет сокращенная ДНФ исходной функции.

2.2.2.2. Построение сокращенной ДНФ в классе дизъюнктивных нормальных форм

Этот метод не отличается большой эффективностью, но он прост для изложения и не требует введения дополнительных понятий.

Пусть булева функция задана таблицей истинности или СДНФ.

Минимизирующая карта булевой функции представляет собой квадратную матрицу $2^n \times 2^n$, где n – число переменных. Первые столбцы отводят для аргументов, дальнейшие – для их всевозможных конъюнкций по 2, по 3 и т. д. сомножителей, предпоследний - для конъюнкции всех аргументов, последний – для значений функции.

Шаг 1. Столбцы для аргументов, как обычно в таблицах истинности, заполняются всевозможными наборами 0 и 1. В столбцах для конъюнкций проставляются десятичные значения двоичных чисел, соответствующих наборам значений аргументов. Последний столбец заполняется соответственно значению функции.

Далее работа чередуется по строкам, по столбцам.

Шаг 2. Вычеркиваются строки, в которых функция обращается в нуль.

Шаг 3. В каждом столбце из сохранившихся чисел вычеркивают те, равные которым уже вычеркнуты в этом столбце на предыдущем шаге.

Шаг 4. В сохранившихся строках выбирают «значения» наименьших по числу множителей конъюнкций (включая и конъюнкции с одним множителем – переменные) и обводят их кружочками.

Шаг 5. Если в одном столбце обведено несколько одинаковых чисел, то вычеркивают все, кроме одного.

Шаг 6. С помощью оставшихся обведенных чисел образуют конъюнкции. Для этого переводят каждое число в двоичную систему. Переменную, которой соответствует 1, берут сомножителем без отрицания, которой соответствует 0 – с отрицанием.

Шаг 8. Составляют дизъюнкцию полученных конъюнкций. В результате получаем сокращенную ДНФ функции.

Пример 36.

Построить сокращенную ДНФ для функции *f=11100101*.

Решение.

1.Строим минимизационную карту (табл. 24):

Таблица 24

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	0	0	0	0	*1*
1	0	0	1	0	1	1	1	*1*
2	0	1	0	1	0	2	2	*1*
3	0	1	1	1	1	3	3	*0*
4	1	0	0	2	2	0	4	*0*
5	1	0	1	2	3	1	5	*1*
6	1	1	0	3	2	2	6	*0*
7	1	1	1	3	3	3	7	*1*

2. Вычеркиваем строки, в которых функция обращается в нуль (табл. 25):

Таблица 25

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	0	0	0	0	*1*
1	0	0	1	0	1	1	1	*1*
2	0	1	0	1	0	2	2	*1*
3	0	1	1	1	1	3	3	*0*
4	1	0	0	2	2	0	4	*0*
5	1	0	1	2	3	1	5	*1*
6	1	1	0	3	2	2	6	*0*
7	1	1	1	3	3	3	7	*1*

3. В каждом столбце из сохранившихся чисел вычеркиваем те, равные которым уже вычеркнуты в этом столбце на предыдущем шаге (табл. 26):

Таблица 26

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	0	0	0	0	*1*
1	0	0	1	0	1	1	1	*1*
2	0	1	0	1	0	2	2	*1*
3	0	1	1	1	1	3	3	*0*
4	1	0	0	2	2	0	4	*0*
5	1	0	1	2	3	1	5	*1*
6	1	1	0	3	2	2	6	*0*
7	1	1	1	3	3	3	7	*1*

4. В сохранившихся строках выбираем «значения» наименьших по числу множителей конъюнкций (включая и конъюнкции с одним множителем – переменные) и обводим их (табл. 27):

Таблица 27

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	0	0	0	0	*1*
1	0	0	1	0	1	1	1	*1*
2	0	1	0	1	0	2	2	*1*
3	0	1	1	1	1	3	3	*0*
4	1	0	0	2	2	0	4	*0*
5	1	0	1	2	3	1	5	*1*
6	1	1	0	3	2	2	6	*0*
7	1	1	1	3	3	3	7	*1*

5. Если в одном столбце обведено несколько одинаковых чисел, то вычеркиваем все, кроме одного (табл. 28):

Таблица 28

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	0	0	0	0	*1*
1	0	0	1	0	1	1	1	*1*
2	0	1	0	1	0	2	2	*1*
3	0	1	1	1	1	3	3	*0*
4	1	0	0	2	2	0	4	*0*
5	1	0	1	2	3	1	5	*1*
6	1	1	0	3	2	2	6	*0*
7	1	1	1	3	3	3	7	*1*

6. С помощью оставшихся обведенных чисел образуем конъюнкции. Для этого переводим каждое число в двоичную систему. Переменную, которой соответствует 1, берем сомножителем без отрицания, 0 – с отрицанием. Составляем дизъюнкцию полученных конъюнкций.

Сокращенная ДНФ имеет вид:

$$f(x_1, x_2, x_3) = \overline{x_1}\,\overline{x_2} \vee \overline{x_1}\,\overline{x_3} \vee x_1x_3 \vee \overline{x_2}\,x_3.$$

Пример 37.

Построить сокращенную ДНФ функции *f=1111010010101111* с использованием минимизационной карты.

Решение.

Строим минимизационную карту (табл. 29) и пошагово выполняем алгоритм.

Шаг 1.

Таблица 29

№	x_1	x_2	x_3	x_4	x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$	f
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

1	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	1
2	0	0	1	0	0	1	0	1	0	2	1	0	2	2	2	1
3	0	0	1	1	0	1	1	1	1	3	1	1	3	3	3	1
4	0	1	0	0	1	0	0	2	2	0	2	2	0	4	4	0
5	0	1	0	1	1	0	1	2	3	1	2	3	1	5	5	1
6	0	1	1	0	1	1	0	3	2	2	3	2	2	6	6	0
7	0	1	1	1	1	1	1	3	3	3	3	3	3	7	7	0
8	1	0	0	0	2	2	2	0	0	0	4	4	4	0	8	1
9	1	0	0	1	2	2	3	0	1	1	4	5	5	1	9	0
10	1	0	1	0	2	3	2	1	0	2	5	4	6	2	10	1
11	1	0	1	1	2	3	3	1	1	3	5	5	7	3	11	0
12	1	1	0	0	3	2	2	2	2	0	6	6	4	4	12	1
13	1	1	0	1	3	2	3	2	3	1	6	7	5	5	13	1
14	1	1	1	0	3	3	2	3	2	2	7	6	6	6	14	1
15	1	1	1	1	3	3	3	3	3	3	7	7	7	7	15	1

Шаг 2. Вычеркиваем строки, в которых функция обращается в нуль (табл. 30):

Таблица 30

№	x_1	x_2	x_3	x_4	x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$	f
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	1
2	0	0	1	0	0	1	0	1	0	2	1	0	2	2	2	1
3	0	0	1	1	0	1	1	1	1	3	1	1	3	3	3	1
4	0	1	0	0	1	0	0	2	2	0	2	2	0	4	4	0

5	***0***	***1***	***0***	***1***	***1***	***0***	***1***	***2***	***3***	***1***	***2***	***3***	***1***	***5***	***5***	***1***
6	***0***	***1***	***1***	***0***	***1***	***1***	***0***	***3***	***2***	***2***	***3***	***2***	***2***	***6***	***6***	***0***
7	***0***	***1***	***1***	***1***	***1***	***1***	***1***	***3***	***3***	***3***	***3***	***3***	***3***	***7***	***7***	***0***
8	***1***	***0***	***0***	***0***	***2***	***2***	***2***	***0***	***0***	***0***	***4***	***4***	***4***	***0***	***8***	***1***
9	***1***	***0***	***0***	***1***	***2***	***2***	***3***	***0***	***1***	***1***	***4***	***5***	***5***	***1***	***9***	***0***
10	***1***	***0***	***1***	***0***	***2***	***3***	***2***	***1***	***0***	***2***	***5***	***4***	***6***	***2***	***10***	***1***
11	***1***	***0***	***1***	***1***	***2***	***3***	***3***	***1***	***1***	***3***	***5***	***5***	***7***	***3***	***11***	***0***
12	***1***	***1***	***0***	***0***	***3***	***2***	***2***	***2***	***2***	***0***	***6***	***6***	***4***	***4***	***12***	***1***
13	***1***	***1***	***0***	***1***	***3***	***2***	***3***	***2***	***3***	***1***	***6***	***7***	***5***	***5***	***13***	***1***
14	***1***	***1***	***1***	***0***	***3***	***3***	***2***	***3***	***2***	***2***	***7***	***6***	***6***	***6***	***14***	***1***
15	***1***	***1***	***1***	***1***	***3***	***3***	***3***	***3***	***3***	***3***	***7***	***7***	***7***	***7***	***15***	***1***

Шаг 3. В каждом столбце из сохранившихся чисел вычеркиваем те, равные которым уже вычеркнуты в этом столбце на предыдущем шаге (табл. 31):

Таблица 31

№	x_1	x_2	x_3	x_4	x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$	f
0	~~0~~	~~0~~	~~0~~	~~0~~	***0***	~~0~~	~~0~~	~~0~~	***0***	~~0~~	***0***	***0***	~~0~~	***0***	***0***	***1***
1	~~0~~	~~0~~	~~0~~	~~1~~	***0***	~~0~~	~~1~~	~~0~~	~~1~~	~~1~~	***0***	***1***	***1***	~~1~~	***1***	***1***
2	~~0~~	~~0~~	~~1~~	~~0~~	***0***	~~1~~	~~0~~	~~1~~	***0***	~~2~~	***1***	***0***	~~2~~	***2***	***2***	***1***
3	~~0~~	~~0~~	~~1~~	~~1~~	***0***	~~1~~	~~1~~	~~1~~	~~1~~	~~3~~	***1***	***1***	~~3~~	~~3~~	***3***	***1***
4	***0***	***1***	***0***	***0***	***1***	***0***	***0***	***2***	***2***	***0***	***2***	***2***	***0***	***4***	***4***	***0***
5	~~0~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~1~~	~~2~~	~~3~~	~~1~~	~~2~~	~~3~~	***1***	***5***	***5***	***1***
6	***0***	***1***	***1***	***0***	***1***	***1***	***0***	***3***	***2***	***2***	***3***	***2***	***2***	***6***	***6***	***0***

7	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**7**	**7**	**0**
8	*1*	*0*	*0*	*0*	*2*	*2*	**2**	*0*	**0**	*0*	*4*	**4**	**4**	**0**	**8**	**1**
9	**1**	**0**	**0**	**1**	**2**	**2**	**3**	**0**	**1**	**1**	**4**	**5**	**5**	**1**	**9**	**0**
10	*1*	*0*	*1*	*0*	*2*	*3*	**2**	*1*	**0**	*2*	*5*	**4**	**6**	**2**	**10**	**1**
11	**1**	**0**	**1**	**1**	**2**	**3**	**3**	**1**	**1**	**3**	**5**	**5**	**7**	**3**	**11**	**0**
12	*1*	*1*	*0*	*0*	**3**	*2*	**2**	*2*	*2*	*0*	**6**	**6**	**4**	*4*	**12**	**1**
13	*1*	*1*	*0*	*1*	**3**	*2*	*3*	*2*	*3*	*1*	**6**	**7**	*5*	**5**	**13**	**1**
14	*1*	*1*	*1*	*0*	**3**	*3*	**2**	*3*	*2*	*2*	**7**	**6**	**6**	*6*	**14**	**1**
15	*1*	*1*	*1*	*1*	**3**	*3*	*3*	*3*	*3*	*3*	**7**	**7**	*7*	*7*	**15**	**1**

Шаг 4. В сохранившихся строках выбираем «значения» наименьших по числу множителей конъюнкций (включая и конъюнкции с одним множителем – переменные) и обводим их (табл. 32):

Таблица 32

№	x_1	x_2	x_3	x_4	x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$	*f*
0	*0*	*0*	*0*	*0*	**0**	*0*	*0*	*0*	**0**	*0*	*0*	*0*	*0*	*0*	*0*	**1**
1	*0*	*0*	*0*	*1*	**0**	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*1*	*1*	**1**
2	*0*	*0*	*1*	*0*	**0**	*1*	*0*	*1*	**0**	*2*	*1*	*0*	*2*	*2*	*2*	**1**
3	*0*	*0*	*1*	*1*	**0**	*1*	*1*	*1*	*1*	*3*	*1*	*1*	*3*	*3*	*3*	**1**
4	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**2**	**2**	**0**	**2**	**2**	**0**	**4**	**4**	**0**
5	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*2*	*3*	*1*	*2*	*3*	**1**	**5**	*5*	**1**
6	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**3**	**2**	**2**	**3**	**2**	**2**	**6**	**6**	**0**
7	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**7**	**7**	**0**

8	1	0	0	0	2	2	2	0	0	0	4	4	4	0	8	1
9	1	0	0	1	2	2	3	0	1	1	4	5	5	1	9	0
10	1	0	1	0	2	3	2	1	0	2	5	4	6	2	10	1
11	1	0	1	1	2	3	3	1	1	3	5	5	7	3	11	0
12	1	1	0	0	3	2	2	2	2	0	6	6	4	4	12	1
13	1	1	0	1	3	2	3	2	3	1	6	7	5	5	13	1
14	1	1	1	0	3	3	2	3	2	2	7	6	6	6	14	1
15	1	1	1	1	3	3	3	3	3	3	7	7	7	7	15	1

Шаг 5. Если в одном столбце обведено несколько одинаковых чисел, то вычеркиваем все, кроме одного (табл. 33):

Таблица 33

№	x_1	x_2	x_3	x_4	x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$	f
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	1
2	0	0	1	0	0	1	0	1	0	2	1	0	2	2	2	1
3	0	0	1	1	0	1	1	1	1	3	1	1	3	3	3	1
4	0	1	0	0	1	0	0	2	2	0	2	2	0	4	4	0
5	0	1	0	1	1	0	1	2	3	1	2	3	1	5	5	1
6	0	1	1	0	1	1	0	3	2	2	3	2	2	6	6	0
7	0	1	1	1	1	1	1	3	3	3	3	3	3	7	7	0
8	1	0	0	0	2	2	2	0	0	0	4	4	4	0	8	1
9	1	0	0	1	2	2	3	0	1	1	4	5	5	1	9	0

10	**1**	**0**	**1**	**0**	**2**	**3**	**2**	**1**	**0**	**2**	**5**	**4**	**6**	**2**	**10**	**1**
11	**1**	**0**	**1**	**1**	**2**	**3**	**3**	**1**	**1**	**3**	**5**	**5**	**7**	**3**	**11**	**0**
12	**1**	**1**	**0**	**0**	**3**	**2**	**2**	**2**	**2**	**0**	**6**	**6**	**4**	**4**	**12**	**1**
13	**1**	**1**	**0**	**1**	**3**	**2**	**3**	**2**	**3**	**1**	**6**	**7**	**5**	**5**	**13**	**1**
14	**1**	**1**	**1**	**0**	**3**	**3**	**2**	**3**	**2**	**2**	**7**	**6**	**6**	**6**	**14**	**1**
15	**1**	**1**	**1**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**7**	**7**	**7**	**7**	**15**	**1**

Шаг 6. Сокращенная ДНФ имеет вид

$$f = \overline{x_1}\,\overline{x_2} \vee x_1 x_2 \vee x_1 \overline{x_4} \vee \overline{x_2}\,\overline{x_4} \vee \overline{x_1}\,\overline{x_3} x_4 \vee x_2 \overline{x_3} x_4 .$$

2.2.2.3. Построение всех тупиковых ДНФ

Определение. ***Тупиковой ДНФ (ТДНФ)*** *функции f называется такая ДНФ ее простых импликант, из которых нельзя выбросить ни одного импликанта, не изменив функции f.*

Теорема. Всякая минимальная ДНФ некоторой функции является ее тупиковой ДНФ.

Для получения МДНФ функции *f* необходимо построить все ТДНФ функции *f* и выбрать те из них, которые содержат минимальное число букв.

Алгоритм построения всех тупиковых ДНФ.

Пусть $f(x_1, x_2, \ldots, x_n)$ есть булева функция.

Шаг 1. Построим СДНФ функции *f* и пусть $P_1, P_2, \ldots, P_n$ есть ее конституенты (единицы).

Шаг 2. Построим сокращенную ДНФ функции *f* и пусть $K_1, K_2, \ldots, K_m$ – ее простые импликанты.

Шаг 3. Построим матрицу покрытий простых импликант функции *f* ее коституентами единицы (табл. 34), полагая, что

$$a_{ij} = \begin{cases} 1, \text{ åñëè êàæäûé ìíîæèòåëü â } K_i \text{ ÿâëÿåòñÿ ìíîæèòåëåì â } P_j; (P_j \text{ åñòü ÷àñòü } K_i); \\ 0, \text{ â ïðîòèâíîì ñëó÷àå} \end{cases}$$

Таблица 34

N	P_1	P_2	…	P_j	…	P_n

K_1	a_{11}	a_{12}	…	a_{1j}	…	a_{1n}
K_2	a_{21}	a_{22}	…	a_{2j}	…	a_{2n}
K_i	a_{i1}	a_{i2}	…	a_{ij}	…	a_{in}
K_m	a_{m1}	a_{m2}	…	a_{mj}	…	a_{mn}

Шаг 4. Для каждого столбца j $(1 \le j \le n)$ найдем множество E_j всех тех номеров i строк, для которых $a_{ij}=1$. Пусть $E_j = \{e_{j1}, e_{j2}, e_{j_{rj}}\}$. Составим выражение $A = \mathop{\wedge}\limits_{j=1}^{n}\left(e_{j1} \vee e_{j2} \vee \ldots \vee e_{j,rj}\right)$.

Назовем его решеточным выражением. Это выражение можно рассматривать как формулу, построенную в свободной дистрибутивной решетке с образующими $1, 2, \ldots, m$ и с операциями конъюнкции и дизъюнкции.

Шаг 5. В выражении A раскроем скобки приведя выражение A к равносильному выражению $B = \mathop{\vee}\limits_{i}(e_{i1} \wedge e_{i2} \wedge \ldots \wedge e_{in})$, где перечислены все конъюнкции $e_{i1} \wedge e_{i2} \wedge \ldots \wedge e_{in}$ элементы $e_{i1}, e_{i2}, \ldots, e_{in}$ которой взяты из скобок $1, 2, \ldots, n$ соответственно в выражении A.

Шаг 6. В выражении В проведем все операции удаления дублирующих членов и все операции поглощения. В результате получим равносильное выражение С, представляющее собой дизъюнкцию элементарных конъюнкций.

Пример 38.

Построить все минимальные ДНФ для функции $f=1111010010101111$.

Решение.

Сокращенная ДНФ для данной функции имеет вид

$$f(x_1,x_2,x_3,x_4) = x_1x_2 \vee \overline{x_1}\,\overline{x_2} \vee \overline{x_2}\,\overline{x_4} \vee x_1\overline{x_4} \vee \overline{x_1}\,\overline{x_3}\,x_4 \vee x_2\overline{x_3}\,x_4.$$

Строим матрицу покрытий (табл. 35).

Таблица 35

№	*Простые импликанты*				*Конституенты единицы функции f*										
	x_1	x_2	x_3	x_4	*0000*	*0001*	*0010*	*0011*	*0101*	*1000*	*1010*	*1100*	*1101*	*1110*	*1111*
1	*1*	*1*	-	-								+	+	+	+
2	*0*	*0*	-	-	+	+	+	+							
3	-	*0*	-	*0*	+		+			+	+				
4	*1*	-	-	*0*						+	+	+		+	
5	*0*	-	*0*	*1*		+			+						
6	-	*1*	*0*	*1*					+				+		

Пошагово будем выбирать слагаемые, которые войдут в минимальную ДНФ. Если слагаемое нами выбрано, то мы помечаем конституенты единицы функции f, которые будут покрыты (по строке). При этом автоматически исключаем из рассмотрения конституенты единицы, которые уже покрыты, но относятся к другим слагаемым сокращенной ДНФ.

Шаг 1. Выбираем слагаемое 1 (табл. 36):

Таблица 36

№	*Простые импликанты*				*Конституенты единицы функции f*										
	x_1	x_2	x_3	x_4	*0000*	*0001*	*0010*	*0011*	*0101*	*1000*	*1010*	*1100*	*1101*	*1110*	*1111*
1	*1*	*1*	-	-								+	+	+	+
2	*0*	*0*	-	-	+	+	+	+							
3	-	*0*	-	*0*	+		+			+	+				
4	*1*	-	-	*0*						+	+	+		+	
5	*0*	-	*0*	*1*		+			+						
6	-	*1*	*0*	*1*					+				+		

Шаг 2. Выбираем слагаемое 2 (табл. 37):

Таблица 37

№	*Простые импликанты*				*Конституенты единицы функции f*										
	x_1	x_2	x_3	x_4	*0000*	*0001*	*0010*	*0011*	*0101*	*1000*	*1010*	*1100*	*1101*	*1110*	*1111*
1	*1*	*1*	-	-								+	+	+	+
2	*0*	*0*	-	-	+	+	+	+							
3	-	*0*	-	*0*	+		+			+	+				
4	*1*	-	-	*0*						+	+	+		+	
5	*0*	-	*0*	*1*		+			+						
6	-	*1*	*0*	*1*					+				+		

Шаг 3. Выбираем слагаемое 4 (табл. 38):

Таблица 38

№	*Простые импликанты*				*Конституенты единицы функции f*										
	x_1	x_2	x_3	x_4	*0000*	*0001*	*0010*	*0011*	*0101*	*1000*	*1010*	*1100*	*1101*	*1110*	*1111*
1	*1*	*1*	-	-								+	+	+	+
2	*0*	*0*	-	-	+	+	+	+							

3	-	0	-	0	+		+			+	+				
4	1	-	-	0						+	+	+		+	
5	0	-	0	1		+			+						
6	-	1	0	1					+				+		

Шаг 4. Выбираем слагаемое 5 (табл. 39):

Таблица 39

№	*Простые импликанты*				*Конституенты единицы функции f*										
	x_1	x_2	x_3	x_4	*0000*	*0001*	*0010*	*0011*	*0101*	*1000*	*1010*	*1100*	*1101*	*1110*	*1111*
1	1	1	-	-								+	+	+	+
2	0	0	-	-	+	+	+	+							
3	-	0	-	0	+		+			+	+				
4	1	-	-	0						+	+	+		+	
5	0	-	0	1		+			+						
6	-	1	0	1					+				+		

Поскольку все конституенты единицы покрыты, то одна из ТДНФ имеет вид

$$f(x_1,x_2,x_3,x_4)=x_1x_2\vee\bar{x}_1\bar{x}_2\vee x_1\bar{x}_4\vee\bar{x}_1\bar{x}_3x_4.$$

Поскольку выбор включаемых слагаемых произволен, то функция может иметь несколько ТДНФ. Для рассматриваемой функции существует еще несколько ТДНФ:

$$f(x_1,x_2,x_3,x_4)=x_1x_2\vee\bar{x}_1\bar{x}_2\vee\bar{x}_2\bar{x}_4\vee\bar{x}_1\bar{x}_3x_4.$$

$$f(x_1,x_2,x_3,x_4)=x_1x_2\vee\bar{x}_1\bar{x}_2\vee\bar{x}_2\bar{x}_4\vee x_2\bar{x}_3x_4.$$

$$f(x_1,x_2,x_3,x_4)=x_1x_2\vee\bar{x}_1\bar{x}_2\vee x_1\bar{x}_4\vee x_2\bar{x}_3x_4.$$

Все найденные ТДНФ являются минимальными ДНФ.

Пример 39.

Построить одну из МДНФ функции *f=11100101*.

Решение.

Сокращенная ДНФ для данной функции имеет вид

$$f(x_1,x_2,x_3)=\bar{x}_1\bar{x}_2\vee\bar{x}_1\bar{x}_3\vee x_1x_3\vee\bar{x}_2x_3.$$

Строим матрицу покрытий (табл. 40):

Таблица 40

№	*Простые импликанты*	*Конституенты единицы функции f*

	x_1	x_2	x_3	000	001	010	101	111
1	0	0	-	+	+			
2	0	-	0	+		+		
3	1	-	1				+	+
4	-	0	1		+		+	

Шаг 1. Выбираем слагаемое 3 (табл. 41):

Таблица 41

№	Простые импликанты			Конституенты единицы функции f				
	x_1	x_2	x_3	000	001	010	101	111
1	0	0	-	+	+			
2	0	-	0	+		+		
3	1	-	1				+	+
4	-	0	1		+		+	

Шаг 2. Выбираем слагаемое 2 (табл. 42):

Таблица 42

№	Простые импликанты			Конституенты единицы функции f				
	x_1	x_2	x_3	000	001	010	101	111
1	0	0	-	+	+			
2	0	-	0	+		+		
3	1	-	1				+	+
4	-	0	1		+		+	

Шаг 3. Выбираем слагаемое 1(табл. 43):

Таблица 43

№	Простые импликанты			Конституенты единицы функции f				
	x_1	x_2	x_3	000	001	010	101	111
1	0	0	-	+	+			
2	0	-	0	+		+		
3	1	-	1				+	+
4	-	0	1		+		+	

В результате получаем МДНФ:

$$f(x_1, x_2, x_3) = \overline{x_1}\,\overline{x_2} \vee \overline{x_1}\,\overline{x_3} \vee x_1 x_3.$$

2.2.2.4. Алгоритм минимизации функций в классе ДНФ

1. Строим СДНФ функции *f*.

2. Строим сокращенную ДНФ функции *f*.

3. С помощью матрицы покрытий и решеточного выражения строим все ТДНФ функции *f*.

4. Среди построенных ТДНФ выбираем все минимальные дизъюнктивные нормальные формы функции *f*.

Пример 40.

В классе нормальных форм минимизировать функцию *f=01011110*.

Решение.

Для построения сокращенной ДНФ используем алгоритм Куайна.

1. Строим СДНФ для функции *f*. Таблица значений имеет вид (табл. 43):
   Таблица 43

№	x_1	x_2	x_3	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

СДНФ (1): № 1, 3, 4, 5, 6 (табл.44):

Таблица 44

№ слагаемого	*Слагаемое СДНФ*
1	$\overline{x_1}\,\overline{x_2}\,x_3$
2	$\overline{x_1}\,x_2\,x_3$
3	$x_1\,\overline{x_2}\,\overline{x_3}$
4	$x_1\,\overline{x_2}\,x_3$
5	$x_1\,x_2\,\overline{x_3}$

2. Проводим все операции неполного склеивания (табл. 45):

Таблица 45

Слагаемые	*Склеивание по*	*Результат*
1, 2	x_2	$\overline{x_1}\,x_3$

1, 4	x_1	$\overline{x_2}\,x_3$
3, 4	x_3	$x_1\,\overline{x_2}$
3, 5	x_2	$x_1\,\overline{x_3}$

Дальнейшее склеивание невозможно. Все слагаемые предыдущего шага участвовали в операции склеивания, поэтому сокращенная ДНФ имеет вид:

$$f = \overline{x_1}\,x_3 \vee \overline{x_2}\,x_3 \vee x_1\,\overline{x_2} \vee x_1\,\overline{x_3}.$$

3. Строим матрицу покрытий (табл. 46).

Таблица 46

№	Простые импликанты			Конституенты единицы функции f				
	x_1	x_2	x_3	001	011	100	101	110
1	0	-	1	+	+			
2	-	0	1	+			+	
3	1	0	-			+	+	
4	1	-	0			+		+

Последовательно выбираем слагаемые: № 4, 1, 2 (табл. 47).

Таблица 47

№	Простые импликанты			Конституенты единицы функции f				
	x_1	x_2	x_3	001	011	100	101	110
1	0	-	1	+	+			
2	-	0	1	+			+	
3	1	0	-			+	+	
4	1	-	0			+		+

В результате МДНФ имеет вид: $f = \overline{x_1}\,x_3 \vee \overline{x_2}\,x_3 \vee x_1\,\overline{x_3}$.

Пример 41.

Построить МДНФ функции *f=11011011*.

Решение.

Для построения сокращенной ДНФ используем минимизационную карту (табл. 48).

Шаг 1.

Таблица 48

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	0	0	0	0	*1*
1	0	0	1	0	1	1	1	*1*
2	0	1	0	1	0	2	2	*0*
3	0	1	1	1	1	3	3	*1*
4	1	0	0	2	2	0	4	*1*
5	1	0	1	2	3	1	5	*0*
6	1	1	0	3	2	2	6	*1*
7	1	1	1	3	3	3	7	*1*

Шаг 2. Вычеркиваем строки, в которых функция обращается в нуль (табл. 49):

Таблица 49

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*
1	*0*	*0*	*1*	*0*	*1*	*1*	*1*	*1*
2	*0*	*1*	*0*	*1*	*0*	*2*	*2*	*0*
3	*0*	*1*	*1*	*1*	*1*	*3*	*3*	*1*
4	*1*	*0*	*0*	*2*	*2*	*0*	*4*	*1*
5	*1*	*0*	*1*	*2*	*3*	*1*	*5*	*0*
6	*1*	*1*	*0*	*3*	*2*	*2*	*6*	*1*
7	*1*	*1*	*1*	*3*	*3*	*3*	*7*	*1*

Шаг 3. В каждом столбце из сохранившихся чисел вычеркиваем те, равные которым уже вычеркнуты в этом столбце на предыдущем шаге (табл. 50):

Таблица 50

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$

0	0	0	0	**0**	0	**0**	**0**	***1***
1	0	0	1	**0**	**1**	1	**1**	***1***
2	0	1	0	**1**	**0**	**2**	**2**	***0***
3	0	1	1	1	**1**	**3**	**3**	***1***
4	1	0	0	2	**2**	**0**	**4**	***1***
5	1	0	1	**2**	**3**	**1**	**5**	***0***
6	1	1	0	**3**	**2**	2	**6**	***1***
7	1	1	1	**3**	3	**3**	**7**	***1***

Шаг 4. В сохранившихся строках выбираем «значения» наименьших по числу множителей конъюнкций (включая и конъюнкции с одним множителем – переменные) и обводим их (табл. 51):

Таблица 51

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	**0**	0	**0**	0	***1***
1	0	0	1	**0**	**1**	1	1	***1***
2	0	1	0	**1**	**0**	**2**	**2**	***0***
3	0	1	1	1	**1**	**3**	3	***1***
4	1	0	0	2	**2**	**0**	4	***1***
5	1	0	1	**2**	**3**	**1**	**5**	***0***
6	1	1	0	**3**	**2**	2	6	***1***
7	1	1	1	**3**	3	**3**	7	***1***

Шаг 5. Если в одном столбце обведено несколько одинаковых чисел, то вычеркиваем все, кроме одного (табл. 52):

Таблица 52

№	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
0	0	0	0	**0**	0	**0**	0	***1***

1	0	0	1	**0**	**1**	1	1	***1***
2	0	1	0	**1**	**0**	**2**	**2**	***0***
3	0	1	1	1	**1**	**3**	3	***1***
4	1	0	0	2	**2**	**0**	4	***1***
5	1	0	1	**2**	**3**	**1**	**5**	***0***
6	1	1	0	**3**	**2**	2	6	***1***
7	1	1	1	**3**	3	**3**	7	***1***

Шаг 6. Сокращенная ДНФ имеет вид: $f = \overline{x}_1\overline{x}_2 \vee x_1x_2 \vee \overline{x}_1x_3 \vee x_1\overline{x}_3 \vee \overline{x}_2\overline{x}_3 \vee x_2x_3$.

Строим матрицу покрытий (табл. 53).

Таблица 53

№	*Простые импликанты*			*Конституенты единицы функции f*					
	x_1	*x_2*	*x_3*	*000*	*001*	*011*	*100*	*110*	*111*
1	*0*	*0*	-	+	+				
2	*1*	*1*	-					+	+
3	*0*	-	*1*		+	+			
4	*1*	-	*0*				+	+	
5	-	*0*	*0*	+			+		
6	-	*1*	*1*			+			+

Последовательно выбираем слагаемые: № 1, 2, 5, 6 (табл. 54).

Таблица 54

№	*Простые импликанты*			*Конституенты единицы функции f*					
	x_1	*x_2*	*x_3*	*000*	*001*	*011*	*100*	*110*	*111*
1	*0*	*0*	-	+	+				
2	*1*	*1*	-					+	+
3	*0*	-	*1*		+	+			
4	*1*	-	*0*				+	+	
5	-	*0*	*0*	+			+		

6	-	1	1			+			+

В результате МДНФ имеет вид: $f = \bar{x}_1 \bar{x}_2 \vee x_1 x_2 \vee \bar{x}_2 \bar{x}_3 \vee x_2 x_3$.

2.2.3. Полные системы булевых функций

Определение. ***Система булевых функций*** *$f_1, f_2, \ldots, f_n$ называется* ***полной****, если любая булева функция может быть выражена через функции $f_1, f_2, \ldots, f_n$ с помощью суперпозиций.*

Пример 42.

Исходя из определения полной системы булевых функций, следует, что система $\{\wedge, \vee, \neg\}$ является полной, так как любая булева функция может быть представима в виде СДНФ и/либо СКНФ.

Дадим определение суперпозиции функций.

Определение. $K^0 = \{ f_1(x_1, x_2, \ldots, x_{k_1}),\ f_2(x_1, x_2, \ldots, x_{k_2}), \ldots, f_m(x_1, x_2, \ldots, x_{k_m}) \}$ *- конечная система булевых функций. Функция f называется* ***суперпозицией ранга 1*** *(или элементарной суперпозицией) функций $f_1, f_2, \ldots, f_m$, если f может быть получена одним из следующих способов:*

1. *переименованием некоторой переменной x_j какой-нибудь функции f_i, т. е. $f=f_i(x_1, \ldots, x_{j-1}, y, x_{j+1}, \ldots, x_{k1})$, где y может совпасть с любой переменной;*
2. *подстановкой некоторой функции f_l ($1 \le l m$) вместо какой-либо переменной x_j любой из функций $fi \in K^0$, т. е. $f=f_i(x_1, \ldots, x_{j-1}, f_l(x_1, x_2, \ldots, x_{k1}), x_{j+1}, \ldots, x_{ki})$.*

Определение. Суперпозиции ранга 1 образуют класс функций K^1. Класс функций, получающихся из функций класса K^{r-1} суперпозиций ранга r-1 с помощью элементарных суперпозиций, называется ***классом функций K^r суперпозиций ранга r****. Суперпозициями функций из K^0 называются функции, входящие в какой-либо из классов K^r.*

Определение. Класс (множество) K булевых функций называется ***функционально замкнутым****, если вместе с функциями из этого класса он содержит и все их суперпозиции.*

Пример 43.

1. Четыре булевы функции одной переменной ($f_1 = 00$, $f_2 = 11$, $f_3 = 01$, $f_4 = 10$) образуют замкнутый класс.

2. Булевы функции $f_1 = x$ и $f_2 = \bar{x}$ образуют замкнутый класс.

Теорема. Класс ***T_0****$=\{f \mid f(0, 0, \ldots, 0)=0\}$ функций, сохраняющих константу ноль на нулевом наборе, замкнут относительно суперпозиций.*

Теорема. Класс ***T_1****$=\{ f \mid f(1, 1, \ldots, 1)=1\}$ функций, сохраняющих константу один на единичном наборе замкнут относительно суперпозиций.*

2.2.3.1. Двойственные функции

Определение. ***Двойственной*** *для функции f(x_1, x_2, …, x_n) называется функция*

$$f^*(x_1, x_2, ..., x_n) = \overline{f(\overline{x_1}, \overline{x_2}, ..., \overline{x_n})}.$$

Пример 44.

Построить функцию, двойственную данной:

1. $f = x \vee y$;
2. $f = x \rightarrow y$.

Решение.

1. $f^* = \overline{\overline{x} \vee \overline{y}} = \overline{\overline{x}} \wedge \overline{\overline{y}} = x \wedge y.$

2. $f^* = \overline{\overline{x} \rightarrow \overline{y}} = \overline{\overline{\overline{x}} \vee \overline{y}} = \overline{\overline{\overline{x}}} \wedge \overline{\overline{y}} = \overline{x} \wedge y.$

Определение. Функция, совпадающая со своей двойственной, называется ***самодвойственной***.

Утверждение. Если функция f(x_1, x_2, …, x_n) самодвойственна, то функция $\overline{f}$ *тоже самодвойственна.*

Утверждение. Чтобы функция была самодвойственной необходимо и достаточно, чтобы на всяких двух противоположных наборах она принимала разные значения.

Противоположными называются те наборы, которые в сумме дают двоичный код числа (2^n-1).

Пример 45.

Выяснить являются ли функции самодвойственными:

1. $f = (\overline{x} \approx y) \rightarrow \overline{z}$;

2. f = *01110010*.

Решение.

1. Строим таблицу истинности для данной функции $f = (\overline{x} \approx y) \rightarrow \overline{z}$ (табл. 55):

Таблица 55

№	x	y	z	$\overline{x}$	$(\overline{x} \approx y)$	$\overline{z}$	$f = (\overline{x} \approx y) \rightarrow \overline{z}$
0	0	0	0	1	0	1	1
1	0	0	1	1	0	0	1
2	0	1	0	1	1	1	1
3	0	1	1	1	1	0	0
4	1	0	0	0	1	1	1

5	1	0	1	0	1	0	0
6	1	1	0	0	0	1	1
7	1	1	1	0	0	0	1

Так как наборы *(0, 0, 0)* и *(1, 1, 1)* являются противоположными, а *f(0, 0, 0) = f(1, 1, 1)*, то данная функция не является самодвойственной.

2. Строим таблицу значений для функции *f = 01110001* (табл. 56).

Таблица 56

№	x	y	z	f(x, y, z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Перечислим пары противоположных наборов: (0, 7), (1, 6), (2, 5), (3, 4). Легко убедиться по таблице, что на всяких двух противоположных наборах функция принимает разные значения. Следовательно, функция является самодвойственной.

Теорема. Класс ***S*** *= {f | f = f*}самодвойственных функций замкнут относительно суперпозиций.*

2.2.3.2. Линейные функции

Определение. ***Арифметические функции*** *в алгебре логики это сложение по модулю два и умножение (конъюнкция).*

Определение. ***Многочленом Жегалкина*** *называется многочлен, являющийся суммой константы 0 или 1 и различных одночленов, в которые все переменные входят не выше, чем в первой степени:* $\sum X_{t_1} \dots X_{i_k} + a_j$*, причем на каждом наборе <i_1, ..., i_k> все a_{ij} (j = 1, ..., k) различны,* $a_j \in \{0, 1\}$.

Теорема. Всякую булеву функцию можно представить единственным полиномом Жегалкина.

Многочлен Жегалкина можно получить различными способами. Остановимся на рассмотрении построения многочлена Жегалкина с помощью треугольника Паскаля. Рассмотрим алгоритм на примере.

Пример 46.

Построить многочлен Жегалкина для функции *f=10011110*.

Решение.

Алгоритм построения многочлена Жегалкина:

Шаг 1. Строим таблицу (табл. 57). Первый столбец содержит возможные слагаемые полинома Жегалкина. Нулевому набору всегда соответствует слагаемое 1. Остальным наборам соответствует слагаемое, представляющее собой конъюнкцию переменных, которые на данном наборе принимают значение 1. Следующие n столбцов – всевозможные наборы из 0 и 1, соответствующие переменным. Далее столбец значений функции f. Функция g является вспомогательной, поэтому изначально этот столбец не заполнен.

Таблица 57

Слагаемые полинома Жегалкина	x_1	x_2	x_3	f	g	*Треугольник Паскаля*
1	*0*	*0*	*0*	*1*		
x_3	*0*	*0*	*1*	*0*		
x_2	*0*	*1*	*0*	*0*		
x_2x_3	*0*	*1*	*1*	*1*		
x_1	*1*	*0*	*0*	*1*		
$x_1\ x_3$	*1*	*0*	*1*	*1*		
$x_1\ x_2$	*1*	*1*	*0*	*1*		
$x_1\ x_2\ x_3$	*1*	*1*	*1*	*0*		

Шаг 2. Построение треугольника Паскаля. Верхняя сторона треугольника есть функция f. Любой другой элемент треугольника есть сумма по модулю два двух соседних элементов предыдущей строки. Левая сторона треугольника представляет собой значение вспомогательной функции g (табл. 58).

Таблица 58

Слагаемые полинома Жегалкина	x_1	x_2	x_3	f	g	*Треугольник Паскаля*
1	*0*	*0*	*0*	*1*	*1*	*f = 1 0 0 1 1 1 1 0*
x_3	*0*	*0*	*1*	*0*	*1*	*1 0 1 0 0 0 1*
x_2	*0*	*1*	*0*	*0*	*1*	*1 1 1 0 0 1*
x_2x_3	*0*	*1*	*1*	*1*	*0*	*0 0 1 0 1*
x_1	*1*	*0*	*0*	*1*	*0*	*0 1 1 1*
$x_1\ x_3$	*1*	*0*	*1*	*1*	*1*	*1 0 0*
$x_1\ x_2$	*1*	*1*	*0*	*1*	*1*	*1 0*

$x_1\ x_2\ x_3$	1	1	1	0	1	1

Шаг 3. Построение полинома Жегалкина. В полином войдут только те слагаемые, которым соответствует единица во вспомогательной функции g.

Для данной функции многочлен Жегалкина имеет вид:

$$f = 1 + x_3 + x_2 + x_1\, x_3 + x_1\, x_2 + x_1\, x_2\, x_3.$$

Определение. Функция $f(x_1, x_2, \ldots, x_n)$ называется ***линейной****, если многочлен Жегалкина для нее имеет следующий линейный относительно переменных вид:*

$f(x_1, x_2, \ldots, x_n) = a_1x_1 + \ldots + a_nx_n + a_{n+1}$, где каждое a_i равно 0 или 1.

Булева функция из рассмотренного выше примера не является линейной.

Теорема. Класс $\mathbf{L} = \{f \mid f = a_0+a_1x_1+\ldots+a_nx_n,\ a_i\in\{0, 1\}\}$ линейных функций замкнут относительно суперпозиций.

2.2.3.3. Монотонные функции

Определение. Если $\mathbf{a} = (a_1, \ldots, a_n)$ и $\mathbf{b} = (b_1, \ldots, b_n)$ - наборы длины n из 0 и 1, то $\mathbf{a}\le\mathbf{b}$, если $a_1\le b_1, \ldots, a_n\le b_n$.

Пример 47.

Наборы (0, 1, 0) и (1, 1, 0) сравнимы, причем *(0, 1, 0) ≤ (1, 1, 0).*

Наборы (0, 1) и (1, 0) несравнимы. Также несравнимы наборы (0, 1) и (1, 1, 0).

Определение. Функция $f(x_1, x_2, \ldots, x_n)$ называется ***монотонной****, если для всяких наборов $\mathbf{a} = (a_1, \ldots, a_n)$ и $\mathbf{b} = (b_1, \ldots, b_n)$ условие $\mathbf{a}\le\mathbf{b}$ влечет* $\mathbf{f(a) \le f(b)}$.

Утверждение. Функция монотонна тогда и только тогда, когда ее сокращенная ДНФ не содержит отрицаний.

Следствие. Функция монотонна тогда и только тогда, когда ее МДНФ не содержит отрицаний.

Пример 48.

Выяснить, являются ли функции монотонными:

1. *f = 00100110*;
2. *f = 00110111.*

Решение.

1. Сокращенная ДНФ для функции *f = 00100110* имеет вид $f = x_2\overline{x_3} \vee x_1\overline{x_2}x_3$. Поскольку сокращенная ДНФ содержит отрицания, то функция не является монотонной.

2. Сокращенная ДНФ для функции *f = 00110111* имеет вид $f = x_2 \vee x_1x_3$. Поскольку сокращенная ДНФ не содержит отрицаний, то функция является монотонной.

Теорема. Класс $\mathbf{M}= \{f \mid a\le b \Rightarrow f(a)\le f(b)\}$ *монотонных функций замкнут относительно суперпозиций.*

2.2.3.4. Теорема Поста о функциональной полноте

Теорема Поста (признак полноты системы булевых функций). Для того чтобы система булевых функций $\{f_1, \ldots, f_m\}$ *была полной, необходимо и достаточно, чтобы для каждого из пяти функционально замкнутых классов* T_0, T_1, L, M, S *нашлась хотя бы одна функция* f_i *из системы, не принадлежащая этому классу.*

Пример.

Выяснить к каким функционально замкнутым классам принадлежит булева функция $f=01001110$, используя теорему Поста.

Решение.

Строим таблицу значений и треугольник Паскаля (табл. 59):

Таблица 59

Слагаемые полинома Жегалкина	x_1	x_2	x_3	f	g	*Треугольник Паскаля*
1	0	0	0	0	0	f = 0 1 0 0 1 1 1 0
x_3	0	0	1	1	1	1 1 0 1 0 0 1
x_2	0	1	0	0	0	0 1 1 1 0 1
x_2x_3	0	1	1	0	1	1 0 0 1 1
x_1	1	0	0	1	1	1 0 1 0
$x_1 x_3$	1	0	1	1	1	1 1 1
$x_1 x_2$	1	1	0	1	0	0 0
$x_1 x_2 x_3$	1	1	1	0	0	0

Полином Жегалкина имеет вид: $f = x_3 + x_2x_3 + x_1 + x_1x_3$.

1. $f(0, 0, 0) = 0 \Rightarrow f \in T_0$;
2. $f(1, 1, 1) = 1 \Rightarrow f \notin T_1$;
3. $f(0, 0, 0) = f(1, 1, 1)$, а наборы $(0, 0, 0)$ и $(1, 1, 1)$ являются противоположными, то $f \notin S$;
4. так как в полиноме Жегалкина присутствуют слагаемые, представляющие собой конъюнкцию нескольких переменных, то $f \notin L$;
5. сокращенная ДНФ функции имеет вид: $f = x_1 \overline{x_2} \vee x_1 \overline{x_3} \vee \overline{x_2}\, x_3$, так как она содержит отрицания, то $f \notin M$.

Сведем полученные данные:

	T_0	T_1	S	L	M

f	+	-	-	-	-

Пример 49.

Доказать полноту системы $\{+, \vee, 1\}$.

Решение.

Введем обозначения: $f_1 = x_1 + x_2$, $f_2 = x_1 \vee x_2$, $f_3 = 1$. Построим единую таблицу для функций (табл. 60).

Таблица 60

Слагаемые	№	x_1	x_2	$f_1 = x_1+x_2$	*Δ Паскаля*	$f_2 = x_1 \vee x_2$	*Δ Паскаля*	$f_3 = 1$	*Δ Паскаля*
1	0	0	0	0	0 1 1 0	0	0 1 1 1	1	1 1 1 1
x_2	1	0	1	1	1 0 1	1	1 0 0	1	0 0 0
x_1	2	1	0	1	1 1	1	1 0	1	0 0
x_1x_2	3	1	1	0	0	1	1	1	0

Полином Жегалкина:

$$f_1 = x_1 + x_2;$$
$$f_2 = x_2 + x_1 + x_1x_2;$$
$$f_3 = 1.$$

f	T_0	T_1	L	M	S
f_1	+	-	+	-	-
f_2	+	+	-	+	-
f_3	-	+	+	+	-

Поскольку для каждого из пяти функционально замкнутых классов нашлась функция, не принадлежащая этому классу (в каждом столбце имеется хотя бы один минус), то система булевых функций $\{+, \vee, 1\}$ является полной.

2.2.4. Существенные и несущественные переменные. Производная булевой функции первого порядка. Вес переменной

Определение. Булева функция $f \in P_n$ ***существенно зависит*** *от переменной x_i, если существует такой набор значений $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$, что*

$$f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n).$$

В этом случае x_i называют ***существенной переменной****, в противном случае x_i называют* ***несущественной переменной****.*

Определение. ***Производная первого порядка*** $\frac{\partial f}{\partial x_i}$ *от булевой функции f по переменной x_i есть сумма по модулю 2 соответствующих остаточных функций:*

$$\frac{\partial f}{\partial x_i} = f\left(x_1, x_2, ..., x_{i-1}, 0, x_{i+1}, ..., x_n\right) \oplus f\left(x_1, x_2, ..., x_{i-1}, 1, x_{i+1}, ..., x_n\right),$$

где $f(x_1, x_2, ..., x_{i-1}, 1, x_{i+1}, ..., x_n)$ – единичная остаточная функция; $f(x_1, x_2, ..., x_{i-1}, 0, x_{i+1}, ..., x_n)$ – нулевая остаточная функция; $\oplus$ - сумма по модулю 2.

Единичная остаточная функция получается в результате приравнивания переменной x_i единице, нулевая – приравниванием x_i нулю.

Определение. ***Весом производной*** $P(\partial f / \partial x_i)$ *от булевой функции называется число конституент этой производной.*

Утверждение. Чем больше вес производной $P(\partial f / \partial x_i)$, тем больше функция f зависит от переменной x_i.

Пример 50.

Определить переменную x_i, по которой производная $\frac{\partial f}{\partial x_i}$ функции

$$f\left(x_1, ..., x_5\right) = x_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_3 x_4 \vee x_1 x_3 \bar{x}_5 \vee x_1 x_2 x_4 \vee \bar{x}_2 x_3 x_5 \vee \bar{x}_3 \bar{x}_4 x_5$$

имеет минимальный (максимальный) вес, т. е. функция $f(x_1, x_2, x_3, x_4, x_5)$ зависит от нее менее (более) существенно.

Решение.

Определим вес каждой переменной, найдя сначала соответствующую производную.

Имеем

$$\frac{\partial f}{\partial x_1} = \left(\bar{x}_2 x_3 \vee x_3 \bar{x}_5 \vee x_2 x_4 \vee \bar{x}_2 x_3 x_5 \vee \bar{x}_3 \bar{x}_4 x_5\right) \oplus \left(\bar{x}_3 x_4 \vee \bar{x}_2 x_3 x_5 \vee \bar{x}_3 \bar{x}_4 x_5\right).$$

x_2x_3	x_4x_5

Для вычисления веса производной $\frac{\partial f}{\partial x_1}$ зависящей от четырех переменных x_2, x_3, x_4, x_5, представим 4-мерное пространство с образующими $\{x_2, x_3, x_4, x_5\}$ в виде декартова произведения двух двумерных пространств $\{x_2, x_3\} \times \{x_4, x_5\}$ с образующими $\{x_2, x_3\}$ и $\{x_4, x_5\}$ соответственно. Тогда производную $\frac{\partial f}{\partial x_1}$ можно задать в виде двумерной таблицы (табл. 61). Вес производной $\frac{\partial f}{\partial x_1}$ равен числу единиц в этой таблице.

Таблица 61

	00	**01**	**10**	**11**
00	*0*	*0*	*1*	*1*
01	*1*	*0*	*1*	*0*
10	*0*	*0*	*0*	*0*
11	*1*	*0*	*1*	*1*

Итак, $P\left(\frac{\partial f}{\partial x_1}\right) = 7.$

Аналогично вычислим вес производных $\frac{\partial f}{\partial x_i}$ *(i = 2, 3, 4, 5)* (табл. 62, 63, 64, 65). Имеем:

$$\frac{\partial f}{\partial x_2} = \left(\overline{x_1}\,\overline{x_3}x_4 \vee x_1x_3\overline{x_5} \vee x_1x_4 \vee \overline{x_3}\,\overline{x_4}x_5\right) \oplus \left(x_1x_3 \vee \bar{x}_1\bar{x}_3x_4 \vee x_1x_3\bar{x}_5\right),$$

Таблица 62

x_1x_3	x_4x_5			
	00	**01**	**10**	**11**
00	*0*	*0*	*0*	*0*
01	*0*	*1*	*0*	*1*
10	*0*	*0*	*1*	*1*
11	*0*	*1*	*0*	*0*

$$P\left(\frac{\partial f}{\partial x_2}\right) = 5.$$

$$\frac{\partial f}{\partial x_3} = \left(x_1\bar{x}_2 \vee x_1\bar{x}_5 \vee x_1x_2x_4 \vee \bar{x}_2x_5\right) \oplus \left(\bar{x}_1x_4 \vee x_1x_2x_4 \vee \bar{x}_4x_5\right),$$

Таблица 63

x_1x_2	x_4x_5			
	00	**01**	**10**	**11**
00	*0*	*0*	*1*	*0*
01	*0*	*1*	*1*	*1*
10	*1*	*0*	*1*	*1*

11	*0*	*1*	*0*	*0*

$$P\left(\frac{\partial f}{\partial x_3}\right)=8.$$

$$\frac{\partial f}{\partial x_4}=\left(x_1\bar{x}_2x_3\vee\bar{x}_1\bar{x}_3\vee x_1x_3\bar{x}_5\vee x_1x_2\vee\bar{x}_2x_3x_5\right)\oplus\left(x_1\bar{x}_2x_3\vee x_1x_3\bar{x}_5\vee\bar{x}_2x_3x_5\vee\bar{x}_3x_5\right),$$

Таблица 64

x₁x₂	***x₃x₅***			
	00	***01***	***10***	***11***
00	*1*	*0*	*0*	*0*
01	*1*	*0*	*0*	*0*
10	*0*	*1*	*0*	*0*
11	*1*	*0*	*0*	*1*

$$P\left(\frac{\partial f}{\partial x_4}\right)=5.$$

$$\frac{\partial f}{\partial x_5}=\left(x_1\bar{x}_2x_3\vee\bar{x}_1\bar{x}_3x_4\vee x_1x_2x_4\vee\bar{x}_2x_3\vee x_3\bar{x}_4\right)\oplus\left(x_1\bar{x}_2x_3\vee\bar{x}_1\bar{x}_3x_4\vee x_3x_1\vee x_1x_2x_4\right),$$

Таблица 65

x₁x₂	***x₃x₄***			
	00	***01***	***10***	***11***
00	*1*	*0*	*1*	*1*
01	*1*	*0*	*0*	*0*
10	*1*	*0*	*0*	*0*
11	*1*	*0*	*1*	*0*

$$P\left(\frac{\partial f}{\partial x_5}\right)=7.$$

Выяснили, что минимальное значение $\min P\left(\frac{\partial f}{\partial x_i}\right)$ получено при дифференцировании функции f по переменным x_2 и x_4, максимальное значение $\max P\left(\frac{\partial f}{\partial x_i}\right)$ получено при дифференцировании функции f по переменной x_3.

2.3. Исчисление высказываний

Таблицы истинности позволяют ответить на многие вопросы, касающиеся формул логики высказываний, например на вопрос о тождественной истинности формулы или на вопрос о равносильности двух формул. Однако более сложные вопросы логики высказываний уже не могут быть решены с помощью таблиц истинности.

Наряду с алфавитом и правилами построения сложных высказываний – логических формул, языки логики высказываний содержат правила преобразования логических формул. Правила преобразования реализуют общелогические законы и обеспечивают логически правильные рассуждения. Корректность допустимых в логике преобразований является фундаментальным свойством формальной (математической) логики.

Если описание системы (процесса, явления и т.п.) представлено совокупностью сложных высказываний – логических формул, истинных для данной системы (в данной интерпретации ее простых высказываний), то с помощью допустимых преобразований имеющихся логических представлений о системе может быть выполнен их анализ (синтез), могут быть получены новые представления, характеризующие указанную систему (истинные для данной системы) и т.п. Таким образом, с помощью допустимых в логике преобразований появляется возможность получения новых знаний из имеющихся.

Определение. Процесс получения новых знаний, выраженных высказываниями, из других знаний, также выраженных высказываниями, называется ***рассуждением*** *(умозаключением). Исходные высказывания называются* ***посылками*** *(гипотезами, условиями), а получаемые высказывания –* ***заключением*** *(следствием).*

Логика – это наука о способах доказательства.

В логике высказываний все доказательства строятся на отношении порядка, т.е. на отношении, которое существует между причиной и следствием. Отдельные звенья цепи связаны символом импликации «→» при логическом выводе мы будем заменять на символ «⇒», подобно тому, как используются два символа эквивалентности «≈» и «=». Во избежании путаницы вместо конъюнкции «∧» будем использовать символ запятой «,», а вместо дизъюнкции «∨» - символ точка с запятой «;». Тогда утверждение, которое требуется доказать, в логике высказываний оформляется в виде следующего причинно-следственного отношения:

$$P_1, P_2, \ldots, P_{n-1}, P_n \Rightarrow C,$$

где P_i – посылка, C – заключение. Условимся формальную запись такого рода называть ***клаузой.*** Смысловой текст, отвечающий некоторой конкретной клаузе, будем называть её ***легендой***.

Пример 51.

Для данной легенды построить соответствующую клаузу:

« Если фирма приглашает на работу крупного специалиста в области новейшей технологии, то она считает ее привлекательной и разворачивает работы по изменению технологии производства своего традиционного продукта или начинает разработку нового продукта. Конкурирующая фирма пригласила на работу крупного специалиста в области новейшей технологии. Следовательно, она разворачивает работы по изменению технологии производства выпускаемого продукта или разработке нового продукта».

Решение.

Выделим простые высказывания и введем обозначения:

A – «фирма приглашает на работу крупного специалиста в области новейшей технологии»;

B – «фирма считает данную новейшую технологию привлекательной»;

C – «фирма разворачивает работу по изменению технологии производства своего традиционного продукта»;

D - «фирма начинает разработку нового продукта».

С учетом принятых обозначений умозаключение примет вид:

«Если A, то B и (C или D). A. Следовательно, C или D.»

Используя логические связки, получим окончательно:

$$((A \to (B \wedge (C \vee D))) \wedge A) \Rightarrow (C \vee D).$$

Используя равносильности логики высказываний получаем:

$$\overline{A} \vee (B \wedge C) \vee (B \wedge D), A \Rightarrow C \vee D.$$

Если клауза верна, то она является некоторой логической теоремой. В логике высказываний существует несколько способов доказательств теорем. В логике высказываний существуют аксиоматический и конструктивный подходы доказательств логических выражений. Рассмотренные ниже метод Вонга и метод резолюций относится к смешанной стратегии доказательств.

2.3.1. Основные схемы логически правильных рассуждений

Приведем примеры наиболее употребимых схем логически правильных рассуждений (некоторые их них приведем без пояснений) (табл. 62):

Таблица 62

№			
1	*Правило заключения – утверждающий модус (ModusPonens)*	*Если из высказывания A следует высказывание B и справедливо (истинно) высказывание A, то справедливо B*	$\frac{\grave{A}\to\hat{A},\grave{A}}{\hat{A}}$
2	*Правило отрицания – отрицательный модус (ModusTollens)*	*Если из A следует B, но высказывание B неверно, то неверно A*	$\frac{\grave{A}\to\hat{A},\overline{\hat{A}}}{\overline{\grave{A}}}$
3	*Правила утверждения-отрицания (ModusPonendo-Tollens)*	*Если справедливо или высказывание A, или высказывание B (в разделительном смысле) и истинно одно из них, то другое ложно*	$\frac{\grave{A}\oplus\hat{A},\grave{A}}{\overline{\hat{A}}}$; $\frac{\grave{A}\oplus\hat{A},\hat{A}}{\overline{\grave{A}}}$
4	*Правила отрицания-утверждения (ModusTollen-Ponens)*	*Если истинно или A, или B (в разделительном смысле) и неверно одно из них, то истинно другое*	$\frac{\grave{A}\oplus\hat{A},\overline{\grave{A}}}{\hat{A}}$; $\frac{\grave{A}\oplus\hat{A},\overline{\hat{A}}}{\grave{A}}$
		Если истинно A или B (в неразделительном смысле) и неверно одно из них, то истинно другое	$\frac{\grave{A}\vee\hat{A},\overline{\grave{A}}}{\hat{A}}$; $\frac{\grave{A}\vee\hat{A},\overline{\hat{A}}}{\grave{A}}$
5	*Правило транзитивности*	*Если из A следует B, а из B следует C, то из A следует C*	$\frac{\grave{A}\to\hat{A},\hat{A}\to\tilde{N}}{\grave{A}\to\tilde{N}}$
6	*Закон противоречия*	*Если из A следует B и* $\overline{\hat{A}}$*, то неверно A*	$\frac{\grave{A}\to\hat{A},\grave{A}\to\overline{\hat{A}}}{\overline{\grave{A}}}$
7	*Правило контрапозиции*	*Если из A следует B, то из того, что неверно B, следует, что неверно A*	$\frac{\grave{A}\to\hat{A}}{\overline{\hat{A}}\to\overline{\grave{A}}}$
8	*Правило сложной контрапозиции*	*Если из A и B следует C, то из A и* $\overline{\tilde{N}}$ *следует* $\overline{\hat{A}}$	$\frac{(\grave{A}\wedge\hat{A})\to\tilde{N}}{(\grave{A}\wedge\overline{\tilde{N}})\to\overline{\hat{A}}}$
9	*Правило сечения*	*Если из A следует B, а из B и C следует D, то из A и C следует D*	$\frac{\grave{A}\to\hat{A},(B\wedge C)\to D}{(A\wedge C)\to D}$
10	*Правило импортации (объединения посылок)*		$\frac{\grave{A}\to(\hat{A}\to\tilde{N})}{(\grave{A}\wedge\hat{A})\to\tilde{N}}$

11	*Правило экспортации (разъединения посылок)*		$\dfrac{(\grave{A}\wedge\hat{A})\to\tilde{N}}{\grave{A}\to(\hat{A}\to\tilde{N})}$
12	*Правила дилемм*		$\dfrac{\grave{A}\to\tilde{N},\hat{A}\to\tilde{N},\grave{A}\vee\hat{A}}{\tilde{N}}$
			$\dfrac{\grave{A}\to\hat{A},\grave{A}\to\tilde{N},\overline{\hat{A}}\vee\overline{\tilde{N}}}{\overline{\grave{A}}}$
			$\dfrac{\grave{A}\to\hat{A},C\to D,A\vee C}{B\vee D}$
			$\dfrac{A\to B,C\to D,\overline{B}\vee\overline{D}}{\overline{A}\vee\overline{C}}$

Пример 52.

Следующие рассуждения не являются правильными:

$$\frac{\grave{A}\to\hat{A},\hat{A}}{\grave{A}},\quad \frac{\grave{A}\to\hat{A},\overline{\grave{A}}}{\overline{\hat{A}}},\quad \frac{\grave{A}\vee\hat{A},\grave{A}}{\overline{\hat{A}}}.$$

2.3.2. Метод Вонга

Пусть дана клауза в своей наиболее общей форме:

$B_1, B_2, \ldots, B_n \Rightarrow A_1, A_2, \ldots, A_n$

*Шаг 1.*Снятие отрицаний с посылок и заключений. С этой целью нужно опустить знак отрицаний у A_i и B_j и перенести их в противоположные стороны относительно символа$\Rightarrow$.

Шаг 2. Если слева от символа $\Rightarrow$ встречается конъюнкция, а справа дизъюнкция, то их следует заменить на запятые.

Шаг 3. Если после предыдущих шагов оказалось, что связкой, расположенной слева от$\Rightarrow$, является дизъюнкция, а справа – конъюнкция, то образуются две новые клаузы, каждая из которых содержит одну из двух подформул, заменяющих исходную клаузу.

Шаг 4. Если одна и та же буква находится с обеих сторон символа$\Rightarrow$, то такая строка считается доказанной. Исходная клауза является теоремой, если все ветви оканчиваются истинными клаузами. В противном случае переходим к шагу 3.

Пример 53.

Выяснить, является ли клауза теоремой:

$$P\vee Q,\overline{R\wedge S},\overline{Q},P\vee R\Rightarrow S,\overline{P}.$$

Решение.

Шаг 1. $P\vee Q,\overline{R\wedge S},\overline{Q},P\vee R\Rightarrow S,\overline{P}$.

Избавляемся от отрицаний. В результате получаем: $P \vee Q, P, P \vee R \Rightarrow S, R \wedge S, Q$.

Шаг 2. Поскольку слева от символа $\Rightarrow$ не встречается конъюнкция, а справа не встречается дизъюнкция, то шаг 2 как таковой отсутствует.

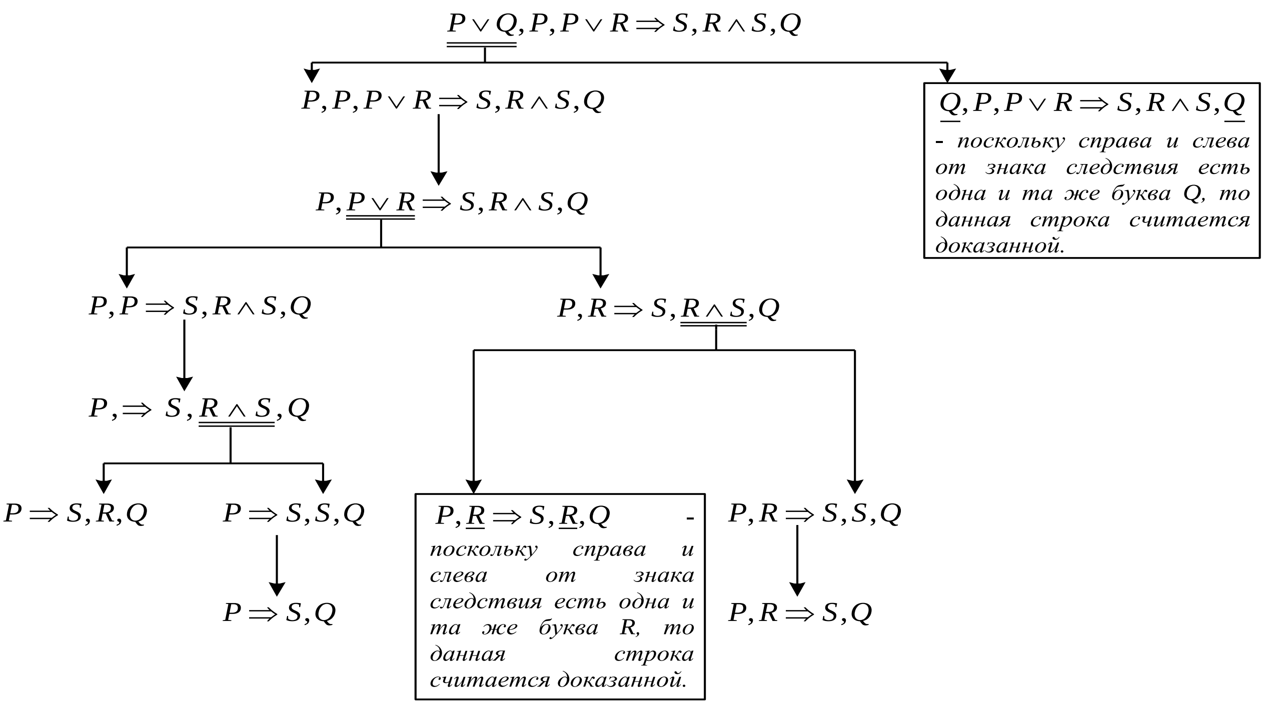


Рис. 11.

Шаг 3. Построим дерево доказательств (рис. 11):

Так как есть не доказанные строки, то исходная клауза теоремой не является.

Пример 54.

Выяснить, является ли клауза теоремой:

$\overline{P} \vee Q, \overline{Q} \vee R, \overline{R} \vee S, \overline{T} \vee \overline{S} \Rightarrow \overline{P}, \overline{T}$.

Решение.

Представим ход доказательства в виде дерева (рис. 12). Поскольку все строки доказаны, то исходная клауза является теоремой.

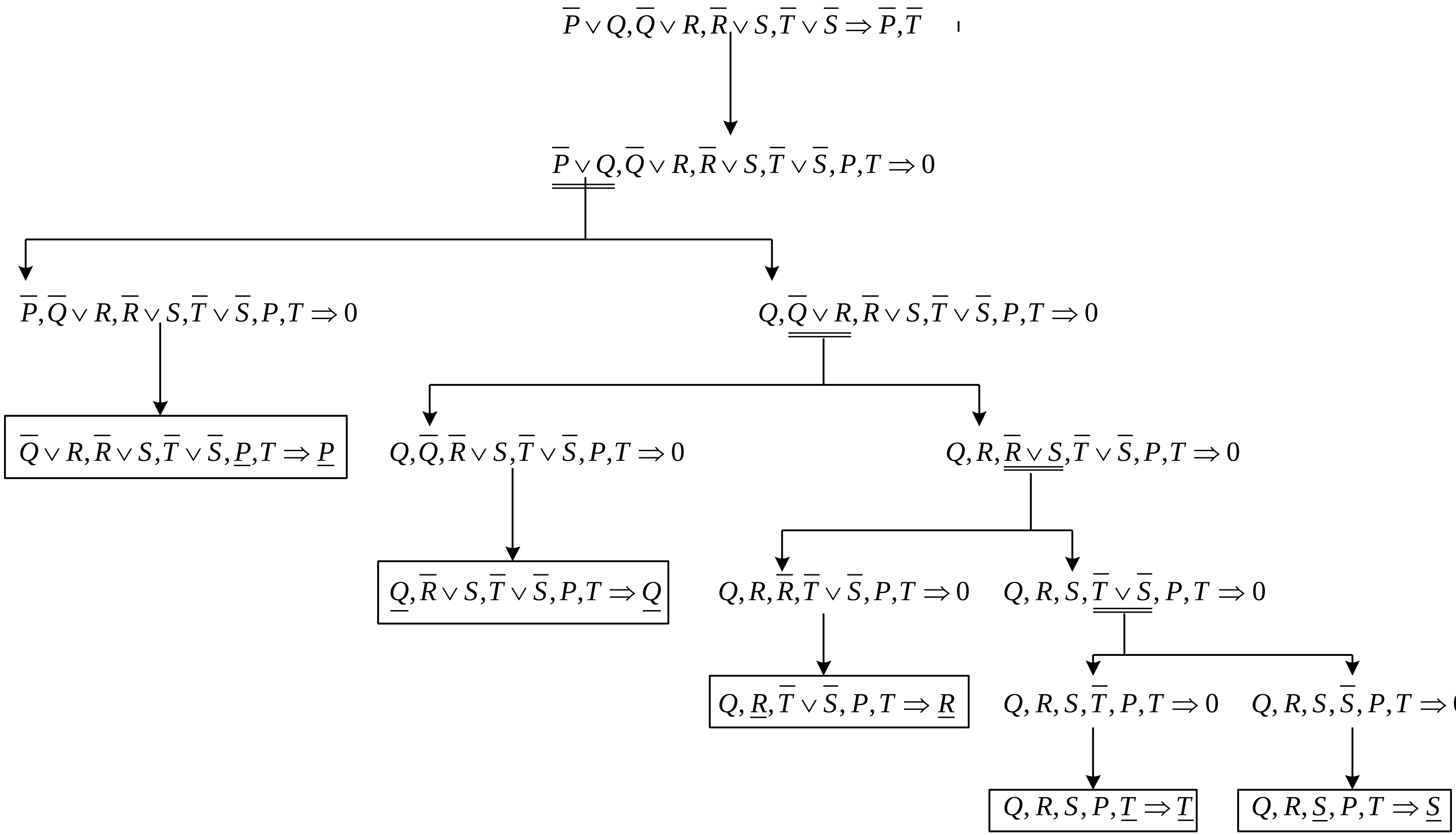


Рис. 12.

2.3.3. *Метод резолюций*

Методику продемонстрируем на примере. Пусть требуется доказать:

$$\overline{A} \to \hat{A}, \overline{C} \to A \Rightarrow \left(B \to \overline{C}\right) \to A.$$

Сначала поступают точно так же, как и по методике Вонга, только необходимо преобразовать клаузу таким образом, чтобы слева от символа $\Rightarrow$ был ноль $\varnothing$:

$$A \vee B, C \vee A \Rightarrow \overline{\left(B \wedge C\right) \wedge \overline{A}}$$
$$A \vee B, C \vee A, \overline{B \wedge C}, \overline{A} \Rightarrow \varnothing$$
$$A \vee B, C \vee A, \overline{B} \vee \overline{C}, \overline{A} \Rightarrow \varnothing$$

Затем из дизъюнктов составляют резолюции до тех пор, пока не получится ноль.

Выпишем по порядку все посылки и далее начнем их «склеивать». Дизъюнкты можно перебирать автоматически в соответствии с возрастанием порядковых номеров. Такая стратегия поиска нуля очень непродуктивна. К решению данной задачи можно подойти творчески.

В итоге получим:

1. $A \vee B$	*5. (1; 4)* $\hat{A}$
2. $C \vee A$	*6. (2; 4)* C
3. $\overline{\hat{A}} \vee \overline{\tilde{N}}$	*7. (3; 5)* $\overline{C}$
4. $\overline{\hat{A}}$	*8. (6; 7)* $\varnothing$

Иначе, произведенные раннее преобразования, можно представить в следующем виде:

$$\left(A \vee B\right) \wedge \left(C \vee A\right) \wedge \left(\overline{B} \vee \overline{C}\right) \wedge \overline{A} \equiv \left(\left(A \vee B\right) \wedge \overline{A}\right) \wedge \left(C \vee A\right) \wedge \left(\overline{B} \vee \overline{C}\right) \equiv \overline{A} \wedge B \wedge \left(C \vee A\right) \wedge \left(\overline{B} \vee \overline{C}\right) \equiv$$
$$\equiv \left(\overline{A} \wedge \left(C \vee A\right)\right) \wedge \left(B \wedge \left(\overline{B} \vee \overline{C}\right)\right) \equiv \overline{A} \wedge C \wedge B \wedge \overline{C} \equiv \varnothing.$$

2.4. *Логика предикатов*

2.4.1. *Основные понятия, связанные с предикатами*

В высказывании все четко: это конкретное утверждение о конкретных объектах – истинное или ложное. Предикат – предложение, похожее на высказывание, но все же им не являющееся: о нем нельзя судить истинно оно или ложно.

Логика предикатов представляет собой развитие логики высказываний. С помощью формул логики высказываний можно описать и исследовать структуру сложных высказываний, установить их истинность или ложность в зависимости от истинности или ложности входящих в нее простых высказываний. Для описания внутренней логической структуры простых высказываний используется понятие предиката.

*Определение.**N-местным предикатом**, определенном на множествах M_1, M_2, …, M_n, называется предложение, содержащее n переменных x_1, x_2, …, x_n, превращающееся в* *высказывание при подстановке вместо этих переменных любых конкретных элементов из множеств M_1, M_2, …, M_n соответственно.*

Обозначение. Чаще всего предикаты обозначают большими латинскими буквами, а число переменных указывает на его размерность: $P(x_1, x_2, \ldots, x_n)$.

Предикат также называют функцией-высказыванием.

Пример 55.

Рассмотрим три высказывания:

А – «Рубль – валюта России»;

В – «Доллар – валюта России»;

С – «Доллар – валюта США».

Высказывания *А* и *С* – истинны, *В* – ложно. Если вместо конкретных наименований валюты в выражениях *А, В* подставить предметную переменную *x* и определить ее на множестве наименований денежных единиц *М*, то получим одноместный предикат

$$P(x): \text{«}x \text{ – валюта России»}, \text{ где } x \in M.$$

Если же в высказывания подставить не только предметную переменную *x*, определенную на множестве *М*, но и вместо наименований стран ввести предметную переменную *y*, определенную на множестве названий стран *У*, то получим двуместный предикат:

$$Q(x, y): \text{«}x \text{ - валюта страны } y\text{»}, \text{ где } x \in M,\ y \in У.$$

Чаще всего предикаты задают высказывательными формами, как показано выше. Однако предикат можно задать таблицей. Такой способ пригоден только для предикатов, область определения которых – конечное множество.

Пример 56.

Пусть задан одноместный предикат $P(x)$, $x \in M$, где $M = \{1, 2, 3, 4, 5\}$. Значение предиката можно задать таблицей (табл. 63):

Таблица 63

x	1	2	3	4	5
P(x)	*и*	*и*	*л*	*и*	*л*

Пример 57.

Предикат задан высказывательной формой *P(x): «в слове x буква «а» встречается не более двух раз», x* $\in$ *M*, где *M = {конь, стол, карандаш, зал, чаша, барабан}*.

Построим таблицу значений для данного предиката (табл. 64):

Таблица 64

x	*конь*	*стол*	*карандаш*	*зал*	*чаша*	*барабан*
P(x)	*и*	*и*	*л*	*и*	*и*	*л*

Определение. ***Множеством истинности*** *предиката $P(x_1, x_2, \ldots, x_n)$, заданного на множествах M_1, M_2, …, M_n, называется совокупность всех упорядоченных n-систем $(a_1, a_2, \ldots, a_n)$, в* *которых* $\grave{a}_1 \in \grave{I}_1, \grave{a}_2 \in \grave{I}, ..., a_n \in \grave{I}_n$ *таких, что данный предикат обращается в истинное высказывание при подстановке* $x_1=a_1$, $x_2=a_2$, …, $x_n=a_n$.

Это множество будем обозначать P^+.

Пример 58.

Определить множество истинности предикатов, заданных на соответствующих множествах:

1. *P(x): «x кратно 3»,* $x \in M$, *где* $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$;
2. *G(x, y):* «$x^2 + y^2 < 0$», $(x, y) \in R \times R$;
3. *Q(x):* «$\sin^2 x + \cos^2 x = 1$», $x \in R$.

R – множество действительных чисел.

Решение.

1. $P^+ = \{3, 6, 9\}$;
2. $G^+ = \varnothing$;
3. $Q^+ = R$.

2.4.2. Классификация предикатов

Определение. Предикат $P(x_1, x_2, …, x_n)$, *заданный на множествах* $M_1, M_2, …, M_n$, *называется:*

1. ***тождественно-истинным**, если при любой подстановке вместо переменных* $x_1, x_2, …, x_n$ *любых конкретных предметов* $a_1, a_2, …, a_n$ *из множеств* $M_1, M_2, …, M_n$ *соответственно он превращается в истинное высказывание* $P(a_1, a_2, …, a_n)$;
2. ***тождественно-ложным,** если при любой подстановке вместо переменных* $x_1, x_2, …, x_n$ *любых конкретных предметов из множеств* $M_1, M_2, …, M_n$ *соответственно он превращается в ложное высказывание;*
3. ***выполнимым (опровержимым),** если существует, по крайней мере, один набор конкретных предметов, при подстановке которого вместо соответствующих переменных в предикат, последний обращается в истинное (ложное) высказывание.*

С точки зрения множества истинности предиката истинны следующее утверждение.

Утверждение.

1. *Если предикат* $P(x_1, x_2, …, x_n)$, *заданный на множествах* $M_1, M_2, …, M_n$ *является тождественно-истинным, то его множество истинности* $P^+ = M_1 \times M_2 \times … \times M_n$.
2. *Если предикат* $P(x_1, x_2, …, x_n)$, *заданный на множествах* $M_1, M_2, …, M_n$ *является тождественно-ложным, то его множество истинности* $P^+ = \varnothing$.
3. *Если предикат* $P(x_1, x_2, …, x_n)$, *заданный на множествах* $M_1, M_2, …, M_n$ *является выполнимым, то его множество истинности* $P^+ \neq \varnothing$.
4. *Если предикат* $P(x_1, x_2, …, x_n)$, *заданный на множествах* $M_1, M_2, …, M_n$ *является опровержимым, то его множество истинности* $P^+ \neq M_1 \times M_2 \times … \times M_n$.

*Определение. **Два n-местных предиката*** $P(x_1, x_2, …, x_n)$ *и* $Q(x_1, x_2, …, x_n)$, *заданных над одними и теми же множествами* $M_1, M_2, …, M_n$, *называются **равносильными**, если набор элементов* $\grave{a}_1 \in \grave{I}_1, \grave{a}_2 \in \grave{I}, ..., a_n \in \grave{I}_n$ *превращает первый предикат в истинное высказывание* $P(a_1, a_2, …, a_n)$ *в том и только в том случае, когда этот набор превращает в истинное высказывание* $Q(a_1, a_2, …, a_n)$ *второй предикат.*

Утверждение о равносильности двух предикатов *P* и *Q* символически будем записывать так: $P \Leftrightarrow Q$.

Пример.

Необходимо решить уравнение (или, другими словами, найти множество истинности предиката): $4x - 2 = -3x - 9$.

Решение.

Делая равносильные преобразования, найдем множество истинности предиката:

$4x - 2 = -3x - 9 \Leftrightarrow 4x + 3x = -9 + 2 \Leftrightarrow x = -1$.

Определение. Предикат $Q(x_1, x_2, \ldots, x_n)$, заданный над множествами $M_1, M_2, \ldots, M_n$, называется ***следствием предиката*** *$P(x_1, x_2, \ldots, x_n)$, заданного над теми же множествами, если он превращается в истинное высказывание на всех наборах значений предметных переменных на соответствующих множествах, на которых в истинное высказывание превращается предикат $Q(x_1, x_2, \ldots, x_n)$.*

Другими словами (в терминах множеств истинности), можно сказать, что предикат Q является следствием предиката P тогда и только тогда, когда $P^+ \subseteq Q^+$.

Теорема. Каждые два тождественно истинных (тождественно ложных)предиката, заданных на одних и тех же множествах, равносильны. Обратно, всякий предикат, равносильный тождественно истинному (тождественно ложному) предикату, сам является тождественно истинным (тождественно ложным) предикатом.

Теорема. Каждый тождественно истинный n-местный предикат является следствием любого другого n-местного предиката, определенного на тех же множествах. Каждый n-местный предикат является следствием любого тождественно ложного n-местного предиката, определенного на тех же множествах.

2.4.3. Логические операции над предикатами

Над предикатами можно проделывать те же самые логические операции, что и над высказываниями. Рассмотрим основные три операции в их связи с операциями над множествами.

Определение. ***Отрицанием****n – местного предиката $P(x_1, x_2, \ldots, x_n)$, определенного на множествах $M_1, M_2, \ldots, M_n$, называется новый n-местный предикат, определенный на тех же множествах, обозначаемый $\neg P(x_1, x_2, \ldots, x_n)$, который превращается в истинное высказывание при всех тех значениях предметных переменных, при которых исходный предикат превращается в ложное высказывание.*

Теорема. Для n-местного предиката $P(x_1, x_2, \ldots, x_n)$, определенного на множествах $M_1, M_2, \ldots, M_n$, множество истинности его отрицания $\neg P(x_1, x_2, \ldots, x_n)$ совпадает с его дополнением множества истинности данного предиката:

$$\left(\overline{P}\right)^+ = \overline{P^+} \text{ или } \left(\overline{P}\right)^+ = \left(M_1 \times M_2 \times \ldots \times M_n\right) \setminus P^+ .$$

Определение. ***Конъюнкцией*** *n – местного предиката $P(x_1, x_2, \ldots, x_n)$, определенного на множествах $M_1, M_2, \ldots, M_n$, и m-местного предиката $Q(y_1, y_2, \ldots, y_m)$, определенного на множествах $N_1, N_2, \ldots, N_m$, называется новый $(n + m)$-местный предикат, определенный на множествах $M_1, M_2, \ldots, M_n, N_1, N_2, \ldots, N_m$, обозначаемый $P(x_1, x_2, \ldots, x_n) \wedge Q(y_1, y_2, \ldots, y_m)$, который превращается в истинное высказывание при всех тех и только тех значениях* *предметных переменных, при которых оба исходных предиката превращаются в истинные высказывания.*

Теорема. Для n-местных предикатов $P(x_1, x_2, \ldots, x_n)$ и $Q(x_1, x_2, \ldots, x_n)$, определенных на множествах $M_1, M_2, \ldots, M_n$, множество истинности конъюнкции $P(x_1, x_2, \ldots, x_n) \wedge Q(x_1, x_2, \ldots, x_n)$, совпадает с пересечением множеств истинности исходных предикатов:

$$\left(\text{Đ} \wedge Q\right)^{+} = P^{+} \cap Q^{+}.$$

Определение. ***Дизъюнкцией*** *n – местного предиката $P(x_1, x_2, \ldots, x_n)$, определенного на множествах $M_1, M_2, \ldots, M_n$, и m-местного предиката $Q(y_1, y_2, \ldots, y_m)$, определенного на множествах $N_1, N_2, \ldots, N_m$, называется новый (n + m)-местный предикат, определенный на множествах $M_1, M_2, \ldots, M_n, N_1, N_2, \ldots, N_m$, обозначаемый $P(x_1, x_2, \ldots, x_n) \vee Q(y_1, y_2, \ldots, y_m)$, который превращается в истинное высказывание при всех тех и только тех значениях предметных переменных, при которых в истинное высказывание превращается по меньшей мере один исходный предикат.*

Теорема. Для n-местных предикатов $P(x_1, x_2, \ldots, x_n)$ и $Q(x_1, x_2, \ldots, x_n)$, определенных на множествах $M_1, M_2, \ldots, M_n$, множество истинности дизъюнкции $P(x_1, x_2, \ldots, x_n) \vee Q(x_1, x_2, \ldots, x_n)$, совпадает с объединением множеств истинности исходных предикатов:

$$\left(\text{Đ} \wedge Q\right)^{+} = P^{+} \cup Q^{+}.$$

2.4.4. Кванторные операции над предикатами

Специфическая природа предикатов, позволяет ввести над ними такие операции, которые не имеют аналогов среди операций над высказываниями. Имеются в виду две кванторные операции над предикатами.

Квантор общности

Для превращения одноместного предиката в высказывание нужно вместо его переменной подставить какой-нибудь конкретный предмет из области задания предиката. Имеется еще один способ для такого превращения – это применение к предикату операций связывания квантором общности или квантором существования. Каждая из этих операций ставит в соответствие одноместному предикату некоторое высказывание, истинное или ложное в зависимости от исходного предиката.

Определение. ***Операцией связывания квантором общности*** *называется правило, по которому каждому одноместному предикату $P(x)$, определенному на множестве M, сопоставляется высказывание, обозначаемое $(\forall x)(P(x))$, которое истинно в том и только в том случае, когда предикат $P(x)$ тождественно истинен, и ложно в противном случае, то есть*

$$(\forall x)(P(x)) = \begin{cases} 1, \textit{если} P(x) - \textit{тождественно} - \textit{истинный} \ \ \textit{предикат}, \\ \quad 0, \ \textit{если} \ \ P(x) - \textit{опровержимый} \ \textit{предикат}. \end{cases}$$

Словесным аналогом квантору общности $\forall$ является: «для любого», «для каждого», «для всякого» и т.п.

В выражении $(\forall x)(P(x))$ переменная x уже перестает быть переменной в обычном смысле этого слова, то есть вместо нее невозможно подставить какие бы то ни было конкретные значения. Говорят, что переменная x ***связанная***.

Если одноместный предикат $P(x)$ задан на конечном множестве $M = \{a_1, a_2, \ldots, a_n\}$, то высказывание $(\forall x)(P(x))$ эквивалентно конъюнкции $P(a_1) \wedge P(a_2) \wedge \ldots \wedge P(a_n)$.

Пример 59.

Пусть x определен на множестве людей M, а $P(x)$ – предикат *«x – смертен»*. Дать словесную формулировку предикатной формулы $(\forall x)(P(x))$.

Решение.

Выражение $(\forall x)(P(x))$ означает «все люди смертны». Оно не зависит от переменной x, а характеризует всех людей в целом, т. е. выражает суждение относительно всех x множества M.

Определение. ***Операцией связывания квантором общности*** *по переменной x_1 называется правило, по которому каждому n-местному ($n \geq 2$) предикату $P(x_1, x_2, \ldots, x_n)$, определенному на множествах $M_1, M_2, \ldots, M_n$, сопоставляется новый (n-1)-местный предикат, обозначаемый $(\forall x_1)(P(x_1, x_2, \ldots, x_n))$, который для любых предметов $a_2 \in M_2, \ldots, a_n \in M_n$, превращается в высказывание $(\forall x_1)(P(x_1, a_2, \ldots, a_n))$, истинное в том и только в том случае, когда одноместный предикат $P(x_1, a_2, \ldots, a_n)$, определенный на множестве M_1, тождественно истинен, и ложное в противном случае, то есть:*

$$(\forall x_1)(P(x_1, a_2, \ldots, a_n)) = \begin{cases} 1, \textit{если } P(x_1, a_2, \ldots, a_n) - \textit{тождественно} - \textit{истинный предикат от } x_1, \\ 0, \textit{если } P(x_1, a_2, \ldots, a_n) - \textit{опровержимый предикат от } x_1. \end{cases}$$

Квантор существования

Определение. ***Операцией связывания квантором существования*** *называется правило, по которому каждому одноместному предикату P(x), определенному на множестве M, сопоставляется высказывание, обозначаемое $(\exists x)(P(x))$, которое ложно в том и только в том случае, когда предикат P(x) тождественно ложен, и истинно в противном случае, то есть*

$$(\exists x)(P(x)) = \begin{cases} 0, \textit{если } P(x) - \textit{тождественно} - \textit{ложный предикат}, \\ 1, P(x) - \textit{выполнимый предикат}. \end{cases}$$

Словесным аналогом квантору существования $\exists$ является: «существует», «найдется» и т.п.

Подобно выражению $(\forall x)(P(x))$, в выражении $(\exists x)(P(x))$переменная x также перестает быть переменной в обычном смысле этого слова: это - ***связаннаяпеременная***.

Если одноместный предикат $P(x)$ задан на конечном множестве $M = \{a_1, a_2, \ldots, a_n\}$, то высказывание $(\exists x)(P(x))$ эквивалентно дизъюнкции $P(a_1) \vee P(a_2) \vee \ldots \vee P(a_n)$.

Пример 60.

Пусть $P(x)$ – предикат *«x – четное число»*, определенный на множестве N. Дать словесную формулировку высказыванию $(\exists x)(P(x))$, определить его истинность.

Решение.

Исходный предикат *P(x): «x – четное число»* является переменным высказыванием: при подстановке конкретного числа вместо переменной *x* он превращается в простое высказывание, являющееся истинным или ложным, например при подстановке числа 5 – ложным, при подстановке числа 10 – истинным. Высказывание $(\exists x)(P(x))$ означает «во множестве натуральных чисел *N* существует четное число». Поскольку множество *N* содержит четные числа, то высказывание $(\exists x)(P(x))$истинно.

Определение. ***Операцией связывания квантором существования*** *по переменной x_1 называется правило, по которому каждому n-местному (n≥ 2) предикату $P(x_1, x_2, \ldots, x_n)$, определенному на множествах $M_1, M_2, \ldots, M_n$, сопоставляется новый (n-1)-местный предикат, обозначаемый $(\exists x_1)(P(x_1, x_2, ..., x_n))$, который для любых предметов $a_2 \in M$ $,..., a_n \in M_n$, превращается в высказывание $(\exists x_1)(P(x_1, a_2, ..., a_n))$, ложное в том и только в том случае, когда одноместный предикат $P(x_1, \grave{a}_2, ..., \grave{a}_n)$, определенный на множестве M_1, тождественно ложен, и истинное в противном случае, то есть:*

$$(\exists \tilde{o}_1)(\text{Ð}(\tilde{o}_1, a_2, ..., a_n)) = \begin{cases} 0, \text{åñëè Ð}(\tilde{o}_1, a_2, ..., a_n) - \text{òîæäåñòâåí íî} - \text{ëîæíûé ïðåäèêàò îò } \tilde{o}_1, \\ 1, \text{åñëè Ð}(\tilde{o}_1, a_2, ..., a_n) - \text{âûïîëíèìûé ïðåäèêàò îò } \tilde{o}_1. \end{cases}$$

Выше уже было сказано, что переменная, на которую навешен квантор, называется связанной, несвязанная квантором переменная называется ***свободной***. Выражение, на которое навешивается квантор, называется ***областью действия квантора*** и все вхождения переменной, на которую навешен квантор, в это выражение являются связанными. На многоместные предикаты можно на разные переменные навешивать различные кванторы, нельзя на одну и ту же переменную навешивать сразу два квантора.

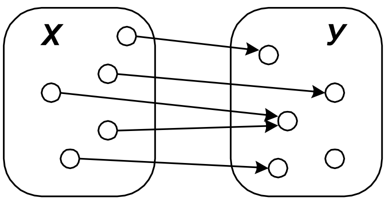
Пример 61.

Пусть предикат *P(x, y)* описывает отношение «x любит y» на множестве людей. Рассмотреть все варианты навешивания кванторов на обе переменные. Дать словесную интерпретацию полученных высказываний.

Решение.

Обозначим предикат *«x любит y»* через *ЛЮБИТ(x, y)*. Предложения, соответствующие различным вариантам навешивания кванторов, проиллюстрированы на рис. 999, где *x* и *y* показаны на разных множествах, что является условностью и предпринято только для объяснения смысла предложений (реальные множества переменных *x* и *y*, очевидно, должны совпадать):

1. $\forall x\ \exists y$ *ЛЮБИТ*(x, y)



$\forall \tilde{o}\ \exists \acute{o}\ \text{ËÞÁÈÒ}\ (\tilde{o}, \acute{o})$ - «для любого человека *x* существует человек *y*, которого он любит» или «всякий человек кого-нибудь любит» (рис. 13).

Рис. 13.

2. ∃*ó* ∀*õ* *ËÞÁÈÒ* (*õ*, *ó*)

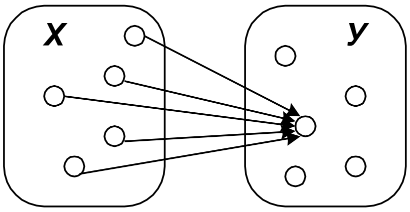


Рис. 14.

∃*ó* ∀*õ* *ËÞÁÈÒ* (*õ*, *ó*) - «существует такой человек *у*, которого любит всякий человек *х*» (рис. 14).

3. ∀*õ* ∀*ó* *ËÞÁÈÒ* (*õ*, *ó*)

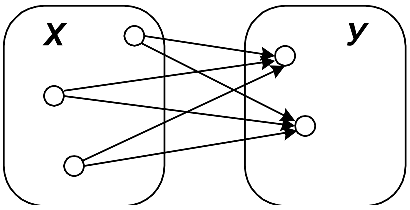


Рис. 15.

∀*õ* ∀*ó* *ËÞÁÈÒ* (*õ*, *ó*) - ***«все люди любят всех людей» (рис. 15).***

4. ∃*õ* ∃*ó* *ËÞÁÈÒ* (*õ*, *ó*)

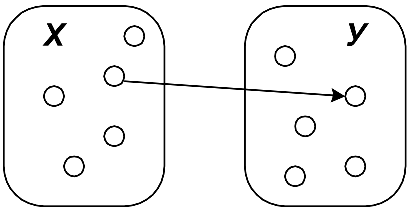


Рис. 16.

∃*õ* ∃*ó* *ËÞÁÈÒ* (*õ*, *ó*) - ***«существует человек, который кого-то любит» (рис. 16).***

5. ∃*õ* ∀*ó* *ËÞÁÈÒ* (*õ*, *ó*)

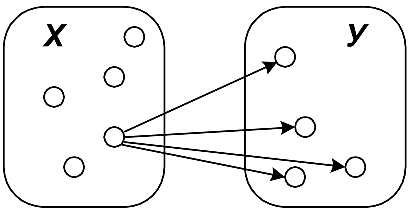


Рис. 17.

∃õ ∀ó ËÞÁÈÒ (õ, ó) - ***«существует человек, который любит всех людей» (рис. 17).***

6. ∀ó ∃õ ËÞÁÈÒ (õ, ó)

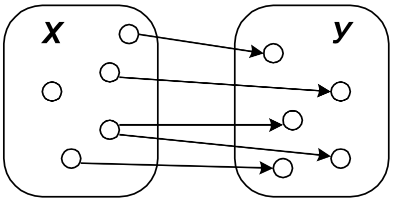


Рис. 18.

∀ó ∃õ ËÞÁÈÒ (õ, ó) - «для всякого человека существует человек, который его любит» или «каждого человека кто-то любит» (рис. 18).

Из приведенного выше примера можно сделать вывод о том, что перестановка кванторов общности и существования меняет смысл высказывания, т.е. кванторы общности и существования не обладают в общем случае свойством коммутативности. Итак, одноименные кванторы можно менять местами, разноименные кванторы менять местами нельзя.

2.4.5. Численные кванторы

В математике часто встречаются выражения вида «по меньшей мере *n*» («хотя бы *n*»), «не более чем *n*», «*n* и только *n*», где *n*– натуральное число.

Эти выражения, называемые численными кванторами, имеют чисто логический смысл; они могут быть заменены равнозначными выражениями, не содержащими числительных и состоящими только из логических терминов и знака =, обозначающего тождество (совпадение) объектов.

Пусть$n = 1$.

Предложение «По меньшей мере один объект обладает свойством *P*» имеет тот же смысл, что и предложение «Существует объект, обладающий свойством *P*», т.е.

$$\exists x\ (P(x)). \quad (1)$$

Предложение «Не более чем один объект обладает свойством *P*» равнозначно предложению «Если есть объекты, обладающие свойством *P*, то они совпадают», т.е.

$$\forall x\ \forall y\ ((P(x) \wedge P(y)) \rightarrow x = y). \quad (2)$$

Предложение «Один и только один объект обладает свойсвом Р» равнозначно конъюнкции предложений (1) и (2), т.е.

$$\exists x\ (P(x)) \wedge \forall x\ \forall y\ ((P(x) \wedge P(y)) \rightarrow x = y).$$

Рассмотрим случай $n = 2$.

Предложение «По меньшей мере два объекта обладают свойством Р» означает то же, что и предложение «Существуют несовпадающие объекты, обладающие свойством Р», т.е.

$$\exists x\ \exists y\ (P(x) \wedge P(y) \wedge x \neq y). \quad (3)$$

Предложение «Не более чем два объекта обладают свойством Р» равнозначно предложению «Каковы бы ни были объекты x, y, z, если все они обладают свойством Р, то по меньшей мере два из них совпадают», т.е.

$$\forall x\ \ \forall y\ \ \forall z\ ((P(x) \wedge P(y) \wedge P(z)) \to (x = y \vee x = z \vee y = z)). \quad (4)$$

Предложение «Два и только два объекта обладают свойством Р» совпадают по смыслу с конъюнкцией предложений (3) и (4).

Совершенно аналогично обстоит дело с численными кванторами при $n > 2$.

2.4.6. Формулы логики предикатов

Напомним некоторые из определений и введем понятие формулы логики предикатов аналогично тому, как это было сделано в логике высказываний.

Зададим сначала алфавит символов, их которых будем составлять формулы:

- предметные переменные: x, y, z, x_i, y_i, z_i (i – натуральное число)$
- предикатные буквы: P, Q, R, …;
- символы операций – отрицания, конъюнкции, дизъюнкции, импликации, эквиваленции, суммы по модулю два;
- кванторы общности и существования;
- вспомогательные символы – скобки, запятая.

Определение.

1. *Всякий нуль-местный предикатный символ – формула.*
2. *Всякий n–местный предикатный символ – формула.*
3. *Если F – формула, а* ξ *- предметная переменная, то* $\forall\xi(F)$ *и* $\exists\xi(F)$ *– формулы.*
4. *Если* F_1 *и* F_2 *– формулы, то* $\overline{F}, F_1 \wedge F_2, F_1 \vee F_2, F_1 \to F_2, F_1 \approx F_2, F_1 \oplus F_2$ *- формулы.*
5. *Никаких других формул в логике предикатов нет.*

Определение. Формулы, определенные в п. 1 и 2, называются элементарными. Формулы, не являющиеся элементарными, называют составными.

Пример 62.

1. P; $Q(x, y, z)$; $R(x_1, x_2)$ – элементарные формулы.
2. $\forall x\ (P(x, y, z)$; $\forall x\ (\exists y\ (P(x, y, z)))$; $(\overline{\forall \tilde{o}(Đ(\tilde{o}, \acute{o})} \wedge Q)$ - составные формулы.

Формула F в формулах вида $\forall\xi(F)$ и $\exists\xi(F)$ называется соответственно ***областью действия квантора*** $\forall\xi$ или $\exists\xi$.

Определение. Вхождение переменной в формулу называется ***связанным****, если оно находится в области действия квантора по этой переменной или является вхождением в этот квантор;*

вхождение, не являющееся связанным, называется ***свободным*** *(область действия квантора всегда однозначно определяется по виду формулы).*

Определение. Переменная называется ***свободной*** *в формуле, если хотя бы одно ее вхождение в этой формуле свободно.*

Определение. Формулы без свободных предметных переменных называются ***замкнутыми****, а формулы, содержащие свободные переменные –* ***открытыми.***

2.4.7. Интерпретация формул логики предикатов

Формулы логики высказываний всегда можно рассматривать как высказывательные формы с высказывательными переменными либо как высказывания. Формулы логики предикатов становятся высказывательными формами с предметными переменными или высказываниями, если задать непустое множество *M* значений, которые можно приписывать предметным переменным, входящим в формулу, а каждому *n*-местному предикатному символу поставить в соответствие *n*-местный предикат, определенный на множестве *M* (причем двум различным *n*-местным предикатным символам с одинаковыми предикатными буквами ставится в соответствие один и тот же предикат); нуль-местным предикатным символам независимо от выбора множества *M* приписывается нуль-местный предикат, т.е. одно из значений истинности *{И, Л}*.

Если формула не содержит свободных предметных переменных, то, задав множество *M* и приписав предикатным символам конкретные предикаты, мы получим высказывание (точнее говоря, значение истинности). Если же в формуле есть свободные вхождения предметных переменных, то получим высказывательную форму от этих переменных, которая станет высказыванием, если подставить вместо свободных вхождений переменных элементы множества *M*.

Обращение формулы в высказывание описанным выше способом будем называть ***интерпретацией*** этой формулы.

Интерпретация замкнутой формулы состоит из следующих шагов:

1. *задается множество M;*
2. *каждой предикатной букве, входящей в n-местный предикатный символ ставится в соответствие n-местный предикат, определенный на множестве M;*
3. *каждому нуль-местному предикатному символу приписывается одно из значений истинности.*

Если формула – ***открытая***, то добавляется еще один шаг:

4. *каждому свободному вхождению переменной ставится в соответствие элемент множества M.*

Пример 63.

Дать интерпретацию формуле $\exists y\,\forall x P(x, y) \to (Q(x) \wedge R)$.

Решение.

Данная формула $\exists y\,\forall x P(x, y) \to (Q(x) \wedge R)$ является открытой, следовательно интерпретация будет состоять из четырех шагов:

1. Пусть *M = {1, 2}*.
2. Предикатной букве *P* поставим в соответствие двуместный предикат, заданный таблицей (табл. 65):

Таблица 65

(1; 1)	*(1; 2)*	*(2; 1)*	*(2; 2)*
и	*л*	*и*	*л*

а предикатной букве *Q* – предикат, принимающий следующие значения:

1	*2*
и	*л*

3. Предикатному символу *R* припишем значение *и*.
4. Свободному вхождению переменной *x* припишем значение *1*.

При такой интерпретации данная формула обращается в *истинное высказывание*.

В самом деле, посылка данной импликации принимает значение *и*, так как, согласно таблице *P*, высказывание *P(1; 1)* и P(2; 1) – истинные, т.е. существует значение *y* (равное 1) такое, что при всяком значении *x* (равном 1 или 2) *P(x; y)* истинно. Заключение также принимает значение *и*, так как *Q(1)* и *R* истинны.

Если же, например, переменной *x* приписать значение *2*, либо символу *R* – значение *л*, либо букве *Q* – предикат *«быть четным числом»*, оставляя все остальное без изменения, то всякий раз данная формула будет получать значение *л*.

2.4.8. Классификация формул логики предикатов

Сформулируем классификационные определения для формул логики предикатов. Рассмотрим некоторую интерпретацию с множеством М.

Определение. Формула A ***выполнимав данной интерпретации****, если существует набор $<a_1, \ldots, a_n>$, $a_i \in M$, значений свободных переменных $x_{i1}, \ldots, x_{in}$ формулы A такой, что $A|<a_1, \ldots, a_n>=И$.*

Определение. Формула A ***истиннав данной интерпретации****, если она принимает значение И на любом наборе $<a_1, \ldots, a_n>$, $a_i \in M$, значений своих свободных переменных $x_{i1}, \ldots, x_{in}$.*

Определение. Формула A ***выполнима*** *(в логике предикатов), если существует интерпретация, в которой A выполнима.*

Определение. Формула A, истинная при любой интерпретации, называется ***общезначимой*** *или тождественно-истинной (в логике предикатов).*

Теорема Черча. Не существует алгоритма, который для любой формулы логики предикатов устанавливает, общезначима она или нет.

Аналогично вводятся понятия опровержимого и тождественно-ложного предиката.

Пример 64.

Выяснить является ли формула выполнимой и опровержимой: $\exists$õĐ(õ, ó)$\to\forall$õĐ(õ, ó)..

Решение.

Поскольку на переменную x навешены кванторы, то она является связной. В свою очередь переменная y является свободной. Формула не имеет вхождений нуль-местных предикатов. Значит, интерпретация будет состоять из трех шагов.

1. Для того чтобы выяснить является ли формула выполнимой, достаточно привести одну интерпретацию, которая обращает исходную формулу в истинное высказывание.

1. Зададим множество $M = \{0\}$.
2. Зададим предикат *P(x, y): «x = y»*.
3. Поскольку заданное множество M имеет единственные элемент, то свободному вхождению переменной y припишем значение 0.

При такой интерпретации данная формула обращается в *истинное высказывание*.

Заданное множество M имеет единственные элемент, поэтому вместо переменной x мы можем подставлять только его. Действительно, посылка данной импликации ∃õĐ(õ, ó) принимает значение *и*. Заключение импликации ∀õĐ(õ, ó) также принимает значение *и*.

Значит, исходная формула является *выполнимой*.

2. Для того чтобы выяснить является ли формула опровержимой, достаточно привести одну интерпретацию, которая обращает исходную формулу в ложное высказывание.

4. Зададим множество $M = \mathbf{N}$.
5. Зададим предикат *P(x, y): «x < y»*.
6. Свободному вхождению переменной y припишем значение 5.

При такой интерпретации данная формула обращается в *ложное высказывание*.

Действительно, посылка данной импликации ∃õĐ(õ, ó) принимает значение *и*, т.к. действительно во множестве натуральных чисел $\mathbf{N}$ найдутся числа меньше числа 5. Заключение импликации ∀õĐ(õ, ó) принимает значение *л*, т.к. не верно, что любое натуральное число меньше числа 5.

Значит, исходная формула является *опровержимой*.

Пример 65.

Доказать общезначимость формулы $\forall x\, P(x) \to \exists x\, P(x)$.

Решение.

Допустим, что P поставлен некоторый предикат на множестве M. Данная формула представляет собой импликацию. Вспомним, что импликация ложна только тогда, когда посылка истинна, а заключение ложно. В нашем случае такая ситуация невозможна. Поскольку, если не для любого элемента $x \in M$ выполняется предикат P, то автоматически исходная формула обращается в истинное высказывание (независимо от того какое значение примет заключение импликации). Если же для любого элемента $x \in M$ выполняется предикат P, то естественно заключение верно, т.е. найдется $x \in M$ такой, что выполняется предикат P.

Таким образом, исходная формула $\forall x\, P(x) \to \exists x\, P(x)$*общезначима*.

2.4.9. Равносильность формул логики предикатов

Пусть формулы A и B имеют одно и то же множество свободных переменных.

Определение.Формулы A и B ***равносильны в данной интерпретации****, если на любом наборе значений свободных переменных они принимают одинаковые значения (т. е. если формулы выражают в данной интерпретации один и тот же предикат).*

Определение.Формулы A и B ***равносильны на множестве M****, если они равносильны во всех интерпретациях, заданных на множестве M..*

Определение.Формулы A и B ***равносильны в логике предикатов****, если они равносильны на всех множествах (A≡B).*

Укажем несколько правил перехода от одних формул к другим, им равносильным.

Для формул логики предикатов сохраняются все равносильности и правила равносильных преобразований логики высказываний.

Утверждение. Всякую формулу логики предикатов, содержащую символы → и ≈, можно преобразовать в равносильную ей формулу, не содержащую этих символов.

Кроме этого, существуют следующие правила:

1. *Перенос квантора через отрицание*
$\neg(\forall x)A(x) \equiv (\exists x)\neg A(x), \qquad \neg(\exists x)A(x) \equiv (\forall x)\neg A(x).$

2. *Вынос квантора за скобки*
$$(\exists x)(A(x) \wedge B) \equiv (\exists x)A(x) \wedge B,$$
$$(\forall x)(A(x) \wedge B) \equiv (\forall x)A(x) \wedge B,$$
$$(\exists x)(A(x) \vee B) \equiv (\exists x)A(x) \vee B,$$
$$(\forall x)(A(x) \vee B) \equiv (\forall x)A(x) \vee B.$$

$$(\forall x)(A(x) \wedge B(x)) \equiv (\forall x)A(x) \wedge (\forall x)B(x),$$
$$(\exists x)(A(x) \vee B(x)) \equiv (\exists x)A(x) \vee (\exists x)B(x).$$

3. *Перестановка одноименных кванторов*
$\forall x\ \forall y\ A(x, y) \equiv \forall y\ \forall x\ A(x, y),$

$\exists x\ \exists y\ A(x, y) \equiv \exists y\ \exists x\ A(x, y).$

4. *Переименование связанных переменных.*
Заменяя связанную переменную формулы A другой переменной, не входящей в эту формулу, в кванторе и всюду в области действия квантора получаем формулу, равносильную A.

Определение.Формула A, равносильная формуле B, и не содержащая символов →, ≈, а также составных формул под знаком отрицания, называется ***приведенной формой*** *формулы B.*

Теорема. Для любой формулы существует равносильная ей приведенная формула, причем множества свободных и связанных переменных этих формул совпадают.

Пример 66.

Преобразовать в приведенную форму формулу $\forall\tilde{o}\exists\acute{o}\text{Đ}(\tilde{o},\acute{o})\to Q(x)$.

Решение.

$$\forall\tilde{o}\exists\acute{o}\text{Đ}(\tilde{o},\acute{o})\to Q(x)\equiv\overline{\forall\tilde{o}\exists\acute{o}\text{Đ}(\tilde{o},\acute{o})}\vee Q(x)\equiv\exists\tilde{o}\forall\acute{o}\overline{\text{Đ}(\tilde{o},\acute{o})}\vee Q(x).$$

Определение.Приведенная формула называется ***нормальной(ПНФ)****, если она не содержит символов кванторов или все кванторы стоят в ее начале, а область действия каждого из них распространяется до конца формулы.*

Пример 67.

Преобразовать в ПНФ формулы:

1. $\exists\tilde{o}\forall\acute{o}\text{Đ}(\tilde{o},\acute{o})\vee\overline{\forall\tilde{o}\exists z\,Q(x,z)}$;

2. $\exists x((\exists yP(y)\wedge Q(x))\to\forall z(R(z)\wedge K(x,z)))$.

Решение.

1. $$\exists\tilde{o}\forall\acute{o}\text{Đ}(\tilde{o},\acute{o})\vee\overline{\forall\tilde{o}\exists z\,Q(x,z)}\equiv\exists\tilde{o}\forall\acute{o}\text{Đ}(\tilde{o},\acute{o})\vee\exists\tilde{o}\forall z\,\overline{Q(x,z)}\equiv\exists\tilde{o}\forall\acute{o}\text{Đ}(\tilde{o},\acute{o})\vee\exists t\,\forall z\,\overline{Q(t,z)}\equiv$$
$$\equiv\exists x\forall y\exists t\forall z(P(x,y)\vee\overline{Q(t,z)}).$$

2. $$\exists x((\exists yP(y)\wedge Q(x))\to\forall z(R(z)\wedge K(x,z)))\equiv\exists x(\overline{(\exists yP(y)\wedge Q(x))}\vee\forall z(R(z)\wedge K(x,z)))\equiv$$
$$\equiv\exists x(\forall y\overline{P(y)}\vee\overline{Q(x)}\vee\forall z(R(z)\wedge K(x,z)))\equiv\exists x\forall y\forall z(\overline{P(y)}\vee\overline{Q(x)}\vee(R(z)\wedge K(x,z))).$$

3. ТЕОРИЯ ГРАФОВ

Графические представления в широком смысле – любые наглядные отображения исследуемой системы, процесса, явления на плоскости. К ним могут быть отнесены рисунки, чертежи, графики зависимостей характеристик, планы-карты местностей, блок-схемы процессов, диаграммы и т.п. Такие изображения наглядно представляют различные взаимосвязи и взаимообусловленности: топологические, хронологические, логические, структурные, причинно-следственные и др.

В последние несколько лет теория графов стала важнейшим математическим инструментом, широко используемым во многих областях науки, начиная с исследования операций и лингвистики и кончая химией и генетикой. В то же самое время она выросла в самостоятельную математическую дисциплину. Для ее понимания требуется только знание элементарной теории множеств и теории матриц.

3.1. Основные определения

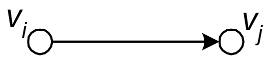
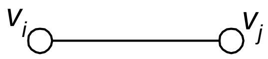
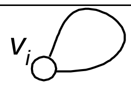
Определение. Пара (V(G), E(G)) называется ***простым графом****, если V(G) – непустое конечное множество элементов, называемых* ***вершинами*** *(или узлами, или точками), а E(G) – конечное множество неупорядоченных пар различных элементов из V(G), называемых* ***ребрами*** *(или линиями). В простом графе данную пару вершин может соединять не более чем одно ребро.*

Между элементами множества вершин и множества ребер определено отношение *инцидентности*. Говорят, что ребро *e* инцидентно вершинам v_1, v_2, если оно соединяет эти вершины и наоборот, каждая из вершин v_1, v_2 инцидентна ребру *e*.

Рассмотрим графическое представление графов (табл. 66).

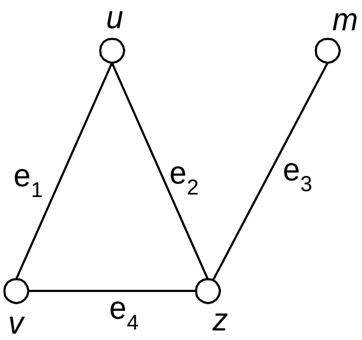
Таблица 66

Графическое представление графов

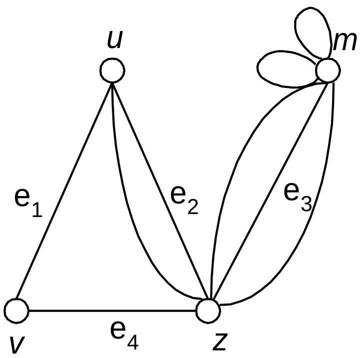
№	*Элементы графов*	*Геометрические элементы*
1.	$x \in X$ – *вершина*	○ - точка в пространстве
2.	$(v_i, v_j) \in V \wedge (v_i, v_j) \notin V$	 - направленный отрезок, ориентированное ребро
3.	$(v_i, v_j) \in V \wedge (v_i, v_j) \in V$	 - отрезок, неориентированное ребро
4.	$(v_i, v_i) \in V$	 - замкнутый отрезок, петля

Многие результаты, полученные для простых графов, без труда модно перенести на более общие объекты, в которых две вершины могут быть соединены более чем одним ребром. Кроме того, часто бывает удобно снять ограничение, состоящее в том, что ребро должно соединять две различные вершины, и допустить существование петель. Получающийся при этом объект, в котором могут быть и кратные ребра и петли называется ***графом (псевдографом)***. Псевдограф без петель называется ***мультиграфом***.

Пример 68.



Простой граф



Граф (псевдограф)

Определение. Если ребро задаётся упорядоченной парой вершин, то оно является ***ориентированным****. Если каждое ребро графа ориентированное, то граф называется* ***ориентированным*** *или* ***орграфом****.*

Иногда удобно преобразовать неориентированный граф в ориентированный – заменой одного неориентированного ребра парой ребер с противоположной ориентацией.

Приведем ряд понятий и определений для ориентированных и неориентированных графов. Там, где это специально не оговорено, те же понятия и определения переносятся без изменений на ориентированные и неориентированные псевдографы.

3.1.1. Смежность, инцидентность, степени

*Определение. Если e = {v, w} – ребро графа, то вершины v, w называются **концами ребра e**; в этом случае также говорят, что **ребро** e **соединяет вершины**v, w.*

*Определение. Если e = {v, w} – дуга орграфа, то вершина v называется **началом**, а вершина w – **концом дуги** e; в этом случае также говорят, что дуга e исходит из вершины v и заходит в вершину w.*

Между элементами множества вершин и множества ребер определено отношение *инцидентности*. Говорят, что ребро *e* инцидентно вершинам *v*, *w*, если оно соединяет эти вершины и наоборот, каждая из вершин *v*, *w* инцидентна ребру *e*.

*Определение. Две вершины называются **смежными**, если существует ребро, концами которого они являются. Два ребра называются смежными, если они имеют общую вершину.*

*Определение. **Степенью** вершины v графа G называется число $\delta(v)$ ребер графа, которым инцидентна эта вершина.*

*Определение. Вершина, локальная степень которой равна 0, называется **изолированной**; степень которой равна 1 – **висячей**.*

Замечание. В случае неориентированного псевдографа обычно считается, что вклад каждой петли, инцидентной некоторой вершине v, равен 2 (тогда как вклад любого другого ребра, инцидентного вершине v, равен 1).

*Определение. **Полустепенью исхода (захода)** вершины v орграфа D называется число $\delta^+(v)$ ($\delta^-(v)$) дуг орграфа D, исходящих из вершины v (заходящих в вершину v).*

Замечание. В случае ориентированного псевдографа вклад каждой петли, инцидентной некоторой вершине v, равен 1, как в $\delta^+(v)$, так и в $\delta^-(v)$.

Количество вершин и ребер в графе G обозначим соответственно через *n(G)* и *m(G)*, а количество вершин и дуг в орграфе D – через *n(D)* и *m(D)*.

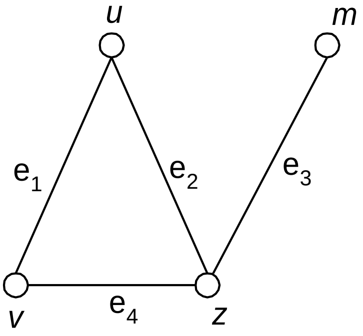
Утверждение. Для любого псевдографа G выполняется равенство $\sum_{v\in V}\delta(v)=2m(G)$.

Утверждение. Для любого ориентированного псевдографа D выполняется равенство $\sum_{v\in V}\delta^+(v)=\sum_{v\in V}\delta^-(v)=m(G)$.

Пример 69.

Найти локальные степени графа (рис. 19) и орграфа (рис. 20).

Решение.



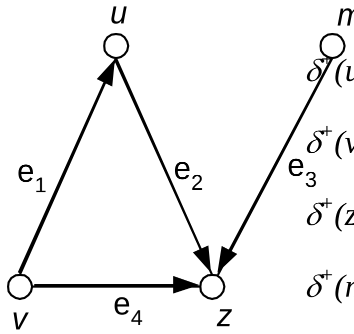
$\delta(u) = 2;$

$\delta(v) = 2;$

$\delta(z) = 3;$

$\delta(m) = 1.$

Рис. 19.



$\delta^+(u) = 1;\quad \delta^-(u) = 1;$

$\delta^+(v) = 2;\quad \delta^-(v) = 0;$

$\delta^+(z) = 0;\quad \delta^-(z) = 3;$

$\delta^+(m) = 1;\quad \delta^-(m) = 0.$

Рис. 20.

3.1.2. Изоморфизм, гомеоморфизм

Определение. Два графа G_1 и G_2 называются ***изоморфными****, если существует взаимно однозначное соответствие между множествами их вершин, обладающее тем свойством, что число ребер, соединяющих любые две вершины в G_1, равно число ребер, соединяющих соответствующие вершины в G_2.*

Из определения следует, что изоморфные графы можно одинаково изображать графически и отличаться они будут только метками вершин.

Изоморфизм графов есть отношение эквивалентности.

Пример 70.

На рисунке 21 изображены три изоморфных графа. На рисунке 22 изображены два неизоморфных графа.

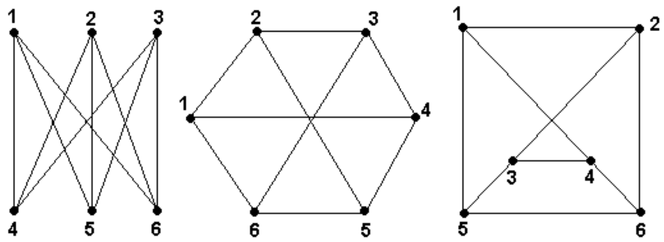


Рис. 21.

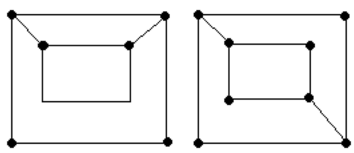


Рис. 22.

Графы рассматриваются с точностью до изоморфизма, то есть рассматриваются классы эквивалентности по отношению изоморфизма.

Пример 71.

С точностью до изоморфизма существует ровно четыре простых графа с тремя вершинами и одиннадцать с четырьмя вершинами. Постройте эти графы.

Решение.

1. Построим четыре простых графа с тремя вершинами с точностью до изоморфизма (рис. 23):

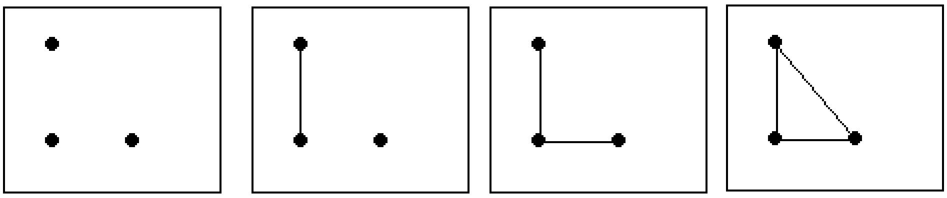


Рис. 23.

2. Построим одиннадцать простых графов с четырьмя вершинами с точностью до изоморфизма (рис. 24):

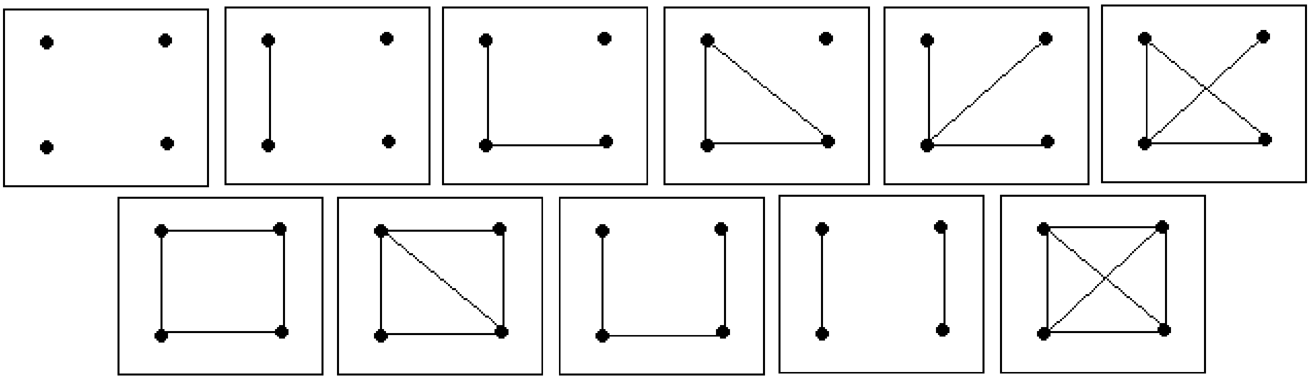


Рис. 24.

*Определение.**Операция подразбиения (измельчения) дуги** (u, v) в орграфе D = (V, E) состоит в удалении из E дуги (u, v), добавлении к V новой вершины w и добавлении к E | {(u, v)} двух дуг (u, v), (w, v). Аналогично определяется операция подразбиения ребра в графах.*

*Определение. Орграф D_1 называется **подразбиением** орграфа D_2, если орграф D_1 можно получить из D_2 путем последовательного применения операции подразбиения дуг. Аналогично определяется подразбиение графа.*

*Определение. Орграфы D_1, D_2 называются **гомеоморфными**, если существуют их подразбиения, являющиеся изоморфными.*

3.1.3. Матричное задание графов. Матрицы смежности, инцидентности

Пусть $D = (V, X)$ – ***орграф***, где $V=\{v_1, v_2, \ldots, v_n\}$, $X=\{x_1, x_2, \ldots, x_m\}$.

*Определение.**Матрицей смежности** орграфа D называется квадратная матрица $A(D)=[a_{ij}]$ порядка n, у которой*

$$a_{ij} = \begin{cases} 1, \text{если } (v_i, v_j) \in X; \\ 0, \text{ если } (v_i, v_j) \notin X. \end{cases}$$

*Определение. **Матрицей инцидентности** орграфа D называется $(n \times m)$ –матрица $B(D)=[b_{ij}]$, у которой*

$$b_{ij} = \begin{cases} 1, \text{ если веошина } v_i \text{ является концом дуги } x_j; \\ -1, \text{ если вершина } v_i \text{ явяляется началом дуги } x_j; \\ 0, \text{ если вершина } v_i \text{ не инцидентна дуге } x_{j.} \end{cases}$$

Введем также матрицы смежности и инцидентности для неориентированных графов. Пусть $G = (V, X)$ – граф, где $V=\{v_1, v_2, \ldots, v_n\}$, $X=\{x_1, x_2, \ldots, x_m\}$.

Определение. ***Матрицей смежности*** *графа G называется квадратная матрица $A(G)=[a_{ij}]$ порядка n, у которой*

$$a_{ij} = \begin{cases} 1, \textit{если } (v_i, v_j) \in X; \\ 0, \textit{ если } (v_i, v_j) \notin X. \end{cases}$$

Определение. ***Матрицей инцидентности*** *графа G называется (n×m) –матрица $B(G)=[b_{ij}]$, у которой*

$$b_{ij} = \begin{cases} 1, \textit{ если веошина } v_i \textit{ инцидентно ребру } x_j; \\ \\ 0, \textit{ если вершина } v_i \textit{ не инцидентна ребру } x_j. \end{cases}$$

С помощью введенных матриц удобно задавать графы для обработки на ЭВМ. Используя матрицу смежности легко определить локальные степени вершин графа: сумма элементов матрицы по строке равна локальной степени соответствующей вершины. Для орграфов по строке определяются полустепени исхода, по столбцам – полустепени захода.

Пример 72.

Построить матрицы смежности и инцидентности для графа $G = (V, X)$ (рис. 25).

Решение.

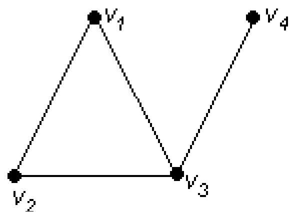


Рис. 25.

Матрица смежности имеет вид

$$\grave{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Поскольку граф не имеет петель, то на главной диагонали стоят все нули. Для любого графа матрица смежности симметрична относительно главной диагонали.

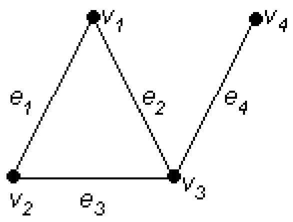


Рис. 26.

Для того, чтобы построить матрицу инцидентности необходимо пронумеровать ребра графа (рис. 26). Матрица инцидентности имеет вид:

$$\hat{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Напомним, что в строках указываются вершины, в столбцах – ребра. Матрица инцидентности может быть как квадратной, так и прямоугольной.

Пример 73.

Построить матрицы смежности и инцидентности для орграфа $D= (V, X)$ (рис. 27).

Решение.

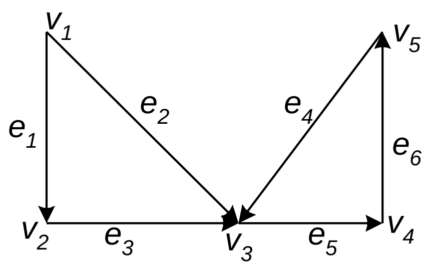


Рис. 27.

Матрица смежности имеет вид:

$$\grave{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Матрица инцидентности имеет вид

$$\hat{A} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

3.1.4. Примеры графов. Операции над графами

Рассмотрим некоторые важные типы графов.

Определение. Граф, у которого множество ребер пусто, называется ***вполне несвязным*** *(или* ***пустым****) графом. Вполне несвязный граф обозначают N_n.*

Заметим, что у вполне несвязного графа все вершины изолированы.

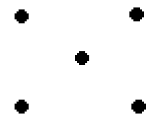
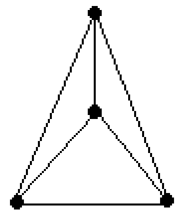


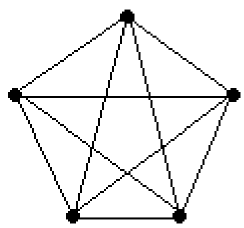
Рис. 28. N_5

Определение. Простой граф, в котором любые две вершины смежны, называется ***полным****. Полный граф обозначают K_n.*

Пример 74.



K_4



K_5

Заметим, что для полного графа выполняется равенство $m = \frac{n(n-1)}{2}$, где m – число ребер, n – число вершин графа.

Определение. Граф, у которого все вершины имеют одну и ту же локальную степень n, называется ***регулярным*** *(или* ***однородным****) степени n.*

Регулярные графы степени 3 называются *кубическими* (или *трехвалентными*).

Пример 75.

Известным примером кубического графа является граф Петерсона (рис. 29).

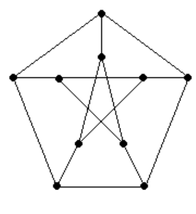
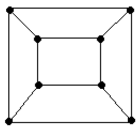


Рис. 29.

Среди регулярных графов особенно интересны так называемые *платоновы графы* – графы, образованные вершинами и ребрами пяти правильных многогранников – Платоновых тел: тетраэдра, куба, октаэдра, додекаэдра и икосаэдра.

Пример 76.



На рис. 30 приведен граф, соответствующий кубу.

Рис. 30.

Определение. Допустим, что множество вершин графа G можно разбить на два непересекающихся подмножества V_1 и V_2 так, что каждое ребро в G соединяет какую-нибудь вершину из V_1 с какой-нибудь вершиной из V_2, тогда данный граф называется ***двудольным****.*

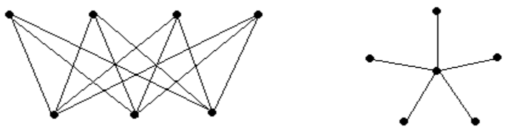
Двудольный граф можно определить и по-другому – в терминах раскраски его вершин двумя цветами, скажем красным и синим. При этом граф называется двудольным, если каждую его вершину можно окрасить красным или синим цветом так, чтобы каждое ребро имело один конец красный, а другой – синий.

Определение. Если в двудольном графе каждая вершина из V_1 соединена с каждой вершиной из V_2, то граф называется ***полным двудольным****.*

Обозначение. $K_{m.\ n}$

Заметим, что граф $K_{m.\ n}$ имеет ровно $m + n$ вершин и mn ребер.

Пример 77.



$K_{4,\ 3}$ $K_{1,5}$

Рис. 31.

На рис. 31 изображены двудольные графы.

Определение. ***Объединением*** *графов* $G_1 = (V_1, X_1),\ G_2 = (V_2,\ X_2)$ *называется граф* $G_1 \cup G_2 = (V_1 \cup V_2,\ X_1 \cup X_2)$.

Определение. ***Пересечением*** *графов* $G_1 = (V_1, X_1),\ G_2 = (V_2,\ X_2)$, *где* $V_1 \cap V_2 \neq \varnothing$, *называется граф* $G_1 \cap G_2 = (V_1 \cap V_2,\ X_1 \cap X_2)$.

Определение. ***Соединением*** *графов G_1 и G_2 является новый граф, у которого $V=V_1 \cup V_2$, а множеством ребер являются все ребра первого и второго графа и ребра, соединяющие между собой каждую вершину первого графа с первой вершиной второго графа.*

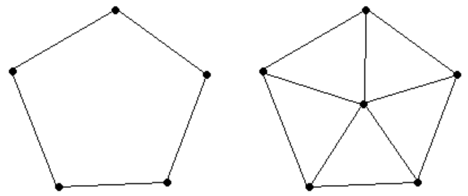
Определение. Граф называется ***связным****, если его нельзя представить в виде объединения двух графов, и несвязным в противном случае.*

Очевидно, что всякий несвязный граф можно представить в виде объединения конечного числа связных графов – каждый из таких связных графов называется *компонентой связности* графа.

Определение. Связный регулярный граф степени 2 называется ***циклическим*** *графом. Обозначается* C_n *(рис. 32).*

Определение. Соединение графов N_1 *и* C_{n-1} $(n\geq 3)$ *называется* ***колесом*** *с n вершинами. Обозначается* W_n *(рис. 32).*

Пример 78.



$C_5 W_6$

Рис. 32.

Определение. ***Дополнением*** *простого графа G называется простой граф с множеством вершин V(G), в котором две вершины смежны тогда и только тогда, когда они не смежны в исходном графе.*

Обозначение. $\overline{G}$

Другими словами, дополнением графа является граф, содержащий все вершины исходного графа и только те ребра, которых не хватает исходному графу для того, чтобы он стал полным.

Определение. ***Подграфом*** *графа G называется граф, все вершины и ребра которого содержатся среди вершин и ребер графа G. Подграф называется* ***собственным****, если он отличен от самого графа.*

3.1.5. Маршруты, цепи, циклы

Введем понятие маршрута для графа $G = (V, E)$ (и соответственно понятие пути для орграфа $D = (V, E)$).

Определение. ***Маршрутом*** *в данном графе G называется конечная последовательность ребер вида* $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{m-1}, v_m\}$ *(обозначаемая также* $v_0 \to v_1 \to v_2 \to \ldots \to v_m$*).*

Каждому маршруту соответствует последовательность вершин $v_0 v_1 v_2 \ldots v_m$; v_0 – начальная вершина, аv_m – конечная вершина маршрута.Одна и та же вершина может одновременно оказаться начальной, конечной и внутренней. Таким образом, мы будем говорить о маршруте из v_0 в v_m.

Определение. ***Длиной маршрута*** *называется число ребер в нем.*

Определение. Маршрут называется ***цепью****, если все его ребра различны, и* ***простой цепью****, если все вершины тоже различны (кроме, может быть, начальной и конечной вершин).*

Определение. Цепь или простая цепь ***замкнуты****, если начальная и конечная вершины совпадают.*

Определение. Замкнутая простая цепь, содержащая, по крайней мере, одно ребро, называется ***циклом****.*

Определение. ***Обхватом*** *графа называется длина его кратчайшего цикла.*

3.1.6. Связность. Компоненты связности

Определение. ***Вершина****w орграфа D (графа G)* ***достижима*** *из вершины v, если либо v=w, либо существует путь из v в w(маршрут, соединяющий v, w).*

Дадим более удобное определение связных графов.

Определение. Граф называется ***связным****, если для любых двух его вершин v, w существует простая цепь из v в w.*

Определение. Граф (орграф) называется ***связным (сильно связным)****, если для любых двух его вершин v, w существует маршрут (путь), соединяющий v, w (из v и w).*

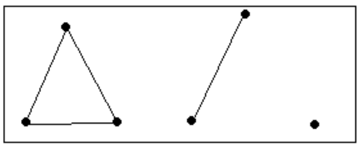
Определение. Орграф называется ***односторонне связным****, если для любых его двух вершин, по крайней мере, одна достижима из другой.*

Определение. Если граф не является связным, то он называется ***несвязным****.*

Определение. ***Компонентой связности*** *графа называется его связный подграф, не являющийся собственным подграфом никакого другого связного подграфа.*

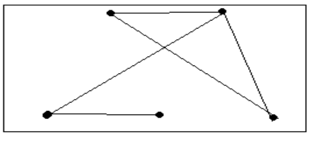
В дальнейшем количество компонент связности графа будем обозначать k.

Пример 79.



Данный граф не является связным: $k = 3$.

Данный граф является связным: $k = 0$.



Теорема. Пусть G – простой граф с n вершинами и k компонентами. Тогда число m его ребер удовлетворяет неравенствам

$$n-k \le m \le \frac{(n-k)(n-k+1)}{2}.$$

Следствие. Любой простой граф с n вершинами и более чем (m-1)(m-2)/2 ребрами связен.

При исследовании графов возникает вопрос: насколько сильно связен связный граф? Этот вопрос можно сформулировать и так: сколько ребер нужно удалить из графа, чтобы он перестал быть связным? Под операцией удаления вершин из графа будем понимать операцию, заключающуюся в удалении некоторой вершины вместе с инцидентными ей ребрами.

Определение. Вершина графа, удаление которой увеличивает число компонент связности, называется ***разделяющейся****.*

Определение. ***Разделяющим множеством*** *связного графа G называется такое множество его ребер, удаление которого приводит к несвязному графу.*

Определение. ***Разрезом*** *называется такое разделяющее множество, никакое собственное подмножество которого не является разделяющим.*

Определение. Разрез, состоящий из одного ребра, называется ***мостом*** *(перешейком).*

Пример 80.

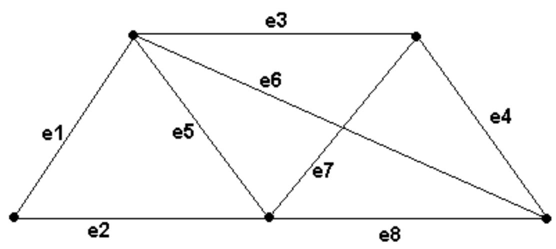


Рис. 33.

Для графа, изображенного на рис.33, каждое из множеств $\{e_1, e_2, e_5\}$ и $\{e_3, e_6, e_7, e_8\}$ является разделяющим.

Разрезом является множество ребер $\{e_1, e_2\}$.

В графе возможно выделить несколько разделяющих множеств и разрезов.

3.2. Задачи поиска маршрутов (путей) в графе (орграфе)

3.2.1. Поиск путей (маршрутов) с минимальным числом дуг (ребер)

При решении некоторых прикладных задач нередко возникает необходимость найти маршрут, соединяющий заданные вершины в графе *G*. Данная задача решается при использовании алгоритма Тэрри.

Рассмотрим более сложную задачу: нахождение пути (маршрута) с минимальным числом дуг (ребер).

Определение.Путь в орграфе D из вершины v в вершину w, где v≠w, называется ***кратчайшим****, если он имеет минимальную длину среди всех путей орграфа D из v и w.Аналогично определяется и кратчайший маршрут в графе G.*

Утверждение. Любой кратчайший путь (маршрут) является простой цепью.

Рассмотрим задачу поиска минимального пути (маршрута). Введем некоторые обозначения. Пусть $D=(V, X)$ – орграф, $v\in V$, $V_1\subset V$. Обозначим $D(v)=\{w\in V|\ (v, w)\in X\}$ – ***образ вершины*** v; $D^{-1}(v)=\{w\in V|\ (w,v)\in X\}$ – ***прообраз вершины***v; $D(V_1)=\bigcup\limits_{v\in V_1} D(v)$ - образ множества вершин V_1; $D^{-1}(V_1)=\bigcup\limits_{v\in V_1} D^{-1}(v)$ - прообраз множества вершин V_1. Пусть G=(V, X) – граф, $v\in V$,

$V_1 \subset V$. Обозначим $G(v)=\{w \in V | \{v, w\} \in X\}$ – образ вершины v; $G(V_1) = \bigcup_{v \in V_1} G(v)$ - образ множества вершин V_1.

Пусть $D=(V, X)$ – орграф с $n \geq 2$ вершинами и v, w – заданные вершины из V, где $v \neq w$. Опишем алгоритм поиска кратчайшего пути из v в w в орграфе D.

Алгоритм фронта волны.

Шаг 1. Помечаем вершину v индексом 0. Затем помечаем вершины, принадлежащие образу вершины v, индексом 1. Множество вершин с индексом 1 обозначаем $FW_1(v)$. Полагаем $k=1$.

Шаг 2. Если $FW_k(v)=\varnothing$ или выполняется $k=n-1, w \notin FW_k(v)$, то вершина w не достижима из v, и работа алгоритма на этом заканчивается. В противном случае переходим к шагу 3.

Шаг 3. Если $w \notin FW_k(v)$, то переходим к шагу 4. В противном случае существует путь из v в w длины k, и этот путь является кратчайшим. Последовательность вершин

$v w_1 w_2 \dots w_{k-1} w$, где

$w_{k-1} \in FW_{k-1}(v) \cap D^{-1}(w)$;

$w_{k-2} \in FW_{k-2}(v) \cap D^{-1}(w_{k-1})$;

....

$w_1 \in FW_1(v) \cap D^{-1}(w_2)$,

и есть искомый кратчайший путь из v и w. На этом работа алгоритма заканчивается.

Шаг 4. Помечаем индексом $k+1$ все непомеченные вершины, которые принадлежат образу множества вершин с индексом k. Множество вершин с индексом $k+1$ обозначаем $FW_{k+1}(v)$. Присваиваем $k:=k+1$ и переходим к шагу 2.

*Замечание.*Множество $FW_k(v)$ обычно называют фронтом волны k-го уровня.

Замечание. Вершины $w_1 w_2 \dots w_{k-1}$, вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных кратчайших путей из v и w.

Пример 81.

Определить кратчайший путь из v_1 и v_6 в орграфе D, заданном матрицей смежности (табл. 67).

Таблица 67

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	0	0	1	1	0
v_2	1	0	0	1	1	1
v_3	1	1	0	1	1	1
v_4	0	1	1	0	1	0
v_5	1	1	1	1	0	0
v_6	1	1	1	1	1	0

Решение.

Согласно алгоритму Фронта волны, последовательно определяем

$FW_1(v_1) = \{v_4, v_5\}$;

$FW_2(v_1) = D(FW_1(v_1))\backslash\{v_1, v_4, v_5\} = \{v_2, v_3\}$;

$FW_3(v_1) = D(FW_2(v_1))\backslash\{v_1, v_2, v_3, v_4, v_5\} = \{v_6\}$.

Таким образом, $v_6 \in FW_3(v_1)$, а значит, существует путь из v_1 в v_6 длины 3, и этот путь является кратчайшим. Найдем теперь кратчайший путь из v_1 в v_6. Определим множество

$$FW_2(v_1) \cap D^{-1}(v_6) = \{v_2, v_3\} \cap \{v_2.v_3\} = \{v_2, v_3\}.$$

Выберем любую вершину из найденного множества, например вершину v_3. Определим далее множество

$$FW_1(v_1) \cap D^{-1}(v_3) = \{v_4, v_5\} \cap \{v_4.v_5, v_6\} = \{v_4, v_5\}.$$

Выберем любую вершину из найденного множества, например вершину v_5. Тогда $v_1v_5v_3v_6$– искомый кратчайший путь из v_1 в v_6 (длины 3) в орграфе D.

3.2.2. Расстояния в графе. Диаметр, центр, радиус графа

Утверждение. Если для двух вершин существует маршрут, связывающий их, то обязательно найдется минимальный маршрут, соединяющий эти вершины. Обозначим длину этого маршрута через $d(v, w)$.

*Определение. Величину $d(v, w)$ (конечную или бесконечную) будем называть **расстоянием между вершинами v, w**. Это расстояние удовлетворяет аксиомам метрики:*

1. *$d(v, w) \geq 0$, причем $d(v, w) = 0$ тогда и только тогда, когда $v=w$;*
2. *$d(v, w) = d(w, v)$;*
3. *$d(v, w) \leq d(v, u) + d(u, w)$.*

*Определение. **Диаметром** связного графа называется максимально возможное расстояние между двумя его вершинами.*

*Определение. **Центром** графа называется такая вершина, что максимальное расстояние между ней и любой другой вершиной является наименьшим из всех возможных; это расстояние называется **радиусом** графа.*

Пример 82.

Для графа G, изображенного на рисунке 34, найти радиус, диаметр и центры.

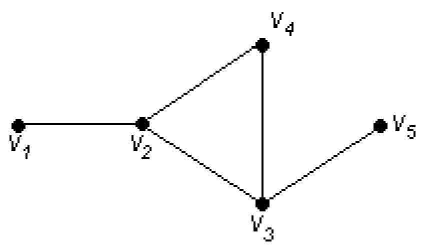


Рис. 34.

Решение.

Чтобы определить центры, радиус, диаметр графа G, найдем матрицу $D(G)$ расстояний между вершинами графа, элементами d_{ij} которой будут расстояния между вершинами v_i и v_j. Для этого воспользуемся графическим представлением графа. Заметим, что матрица $D(G)$ симметрична относительно главной диагонали. $D(G)=\begin{bmatrix} 0 & 1 & 2 & 2 & 3 \\ 1 & 0 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 3 & 2 & 1 & 2 & 0 \end{bmatrix}$.

С помощью полученной матрицы для каждой вершины графа G определим наибольшее удаление из выражения: $r(v_i)=\max\limits_j d(v_i,v_j)$ для $i, j = 1, 2, ..., 5$. В результате получаем: $r(v_1) = 3$, $r(v_2) = 2$, $r(v_3) = 2$, $r(v_4) = 2$, $r(v_5) = 3$. Минимальное из полученных чисел является радиусом графа G, максимальное – диаметром графа G. Значит, $R(G) = 2$ и $D(G) = 3$, центрами являются вершины v_2, v_3, v_4.

3.2.3. Нахождение минимального пути в нагруженном графе

*Определение. Назовём орграф $D=(V,X)$ **нагруженным**, если на множестве дуг X определена некоторая функция $l: X \to R$, которую часто называют **весовой** функцией.*

Тем самым и нагруженном орграфе D каждой дуге $x \in X$ поставлено в соответствие некоторое действительное число $l(x)$. Значение $l(x)$ будем называть *длиной дуги* x.

Для любого пути π нагруженного орграфа D обозначим через $l(\pi)$ сумму длин входящих в π дуг, при этом каждая дуга учитывается столько раз, сколько она входит в путь. Величину $l(\pi)$ будем называть *длиной пути* π в нагруженном орграфе D. Ранее так называлось количество дуг в пути π. В связи с этим заметим, что если длины дуг выбраны равными 1, то $l(\pi)$ выражает введенную ранее длину пути π в ненагруженном орграфе. Следовательно, любой ненагруженный орграф можно считать нагруженным с длинами дуг, равными 1. Аналогично определяется и нагруженный граф, а также длина маршрута в нем.

*Определение. Путь в нагруженном орграфе D из вершины υ в вершину ϖ, где $\upsilon \neq \varpi$, называется **минимальным**, если он имеет минимальную длину среди всех путей орграфа D из υ в ϖ. Аналогично определяется и минимальный маршрут в нагруженном графе G.*

Рассмотрим теперь задачу поиска минимальных путей (маршрутов) в нагруженном орграфе (графе). При этом для определенности рассуждения будем проводить для орграфа (для графа они аналогичны).

Пусть $D=(V,X)$ - нагруженный орграф, $V=\{\upsilon_1,...,\upsilon_n\}$, $n \geq 2$. Введем величины $\lambda_i^{(k)}$, где $i=1, ..., n$, $k=1, 2, ...$ Для каждых фиксированных i и k величина $\lambda_i^{(k)}$ равна длине минимального пути среди путем из υ_1 в υ_i, содержащих не более k дуг; если же таких путей нет, то $\lambda_i^{(k)} = \infty$. Кроме того, если произвольную вершину $\upsilon \in V$ считать путем из υ в υ нулевой длины, то величины $\lambda_i^{(k)}$ можно ввести также и для $k=0$, при этом

$$\lambda_1^{(0)}=0, \lambda_i^{(0)}=\infty,\ i=2,...,n. \qquad (1)$$

Введем также в рассмотрение квадратную матрицу $C(D)=\lfloor c_{ij} \rfloor$ порядка n с элементами

$$c_{ij} = \begin{cases} l(\upsilon_i, \upsilon_j), \text{ если } (\upsilon_i, \upsilon_j) \in X; \\ \infty, \text{ если } (\upsilon_i, \upsilon_j) \overline{\in} X, \end{cases}$$

которую будем называть *матрицей длин дуг* нагруженного орграфа D.

Следующее утверждение дает простые формулы для вычисления величин $\lambda_i^{(k)}$.

Утверждение. При $i = 2,...,n,\ k \geq 0$ *выполняется равенство* $\lambda_i^{(k+1)} = \min\limits_{1 \leq j \leq n}\left\{\lambda_j^{(k)} + c_{ji}\right\}$ (2),

а при $i = 1,\ k \geq 0$ *справедливо равенство* $\lambda_1^{(k+1)} = \min\limits_{1 \leq j \leq n}\left\{\lambda_j^{(k)} + c_{j1}\right\}$ (3).

Используя данное утверждение, нетрудно описать алгоритм нахождения таблицы значений величин $\lambda_i^{(k)}$ (будем записывать её в виде матрицы, где i - номер строки, $k+1$ - номер столбца). Действительно, используя рекуррентные соотношения (2), (3) и исходя из (1), последовательно определяем набор величин $\lambda_1^{(k)},...,\lambda_n^{(h)}$ (($k+1$)-й столбец матрицы), начиная с $k+1$, а затем шаг за шагом увеличиваем значение k до любой необходимой величины.

Алгоритм Форда – Беллмана нахождения минимального пути в нагруженном орграфе D *из* υ_1 *в* υ_{i_1} $(i_1 \neq 1)$

Шаг 1. Пусть мы уже составили таблицу величин $\lambda_i^{(k)},\ i = 1, 2, ..., n,\ k = 0, 1, ..., n-1$. Если $i \in \{2,...,n\}$ не достижима из υ_1 (предполагаем, что все величины $l(x)$, $x \in X$, конечны). В этом случае работа алгоритма заканчивается.

Шаг 2. Пусть $\lambda_{i_1}^{(n-1)} < \infty$. Тогда число $\lambda_{i_1}^{(n-1)}$ выражает длинны любого минимального пути из υ_1 в υ_{i_1} в нагруженном орграфе D. Определим минимальное число $k_1 \geq 1$, при котором выполняется равенство $\lambda_{i_1}^{(k_1)} = \lambda_{i_1}^{(n-1)}$. По определению чисел $\lambda_i^{(k)}$ получим, что k_1 - минимальное число дуг в пути среди всех минимальных путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Шаг 3. Последовательно определяем номера $i_2, ..., i_{k_1+1}$ такие что

$$\lambda_{i_2}^{(k_1-1)} + c_{i_2,i_1} = \lambda_{i_1}^{(k_1)};$$

$$\lambda_{i_3}^{(k_1-2)} + c_{i_3,i_2} = \lambda_{i_2}^{(k_1-1)};$$

$$\ldots\ldots \qquad (4)$$

$$\lambda_{i_{k_1+1}}^{(0)} + c_{i_{k_1+1},i_{k_1}} = \lambda_{i_{k_1}}^{(1)}$$

Из (4) с учётом того, что $\lambda_{i_1}^{(k_1)} = \lambda_{i_1}^{(n-1)} < \infty$, имеем $c_{i_2,i_1} < \infty,...,c_{i_{k_1+1},i_{k_1}} < \infty$, $\lambda_{i_{k_1+1}}^{(0)} < \infty$, откуда, используя (1), получаем

$$\upsilon_{i_2}, \upsilon_{i_1}, ..., (\upsilon_{i_{k_1+1}}, \upsilon_{i_{k_1}}) \in X, \quad l(\upsilon_{i_2}, \upsilon_{i_1}) = c_{i_2,i_1}, ..., l(\upsilon_{i_{k_1+1}}, \upsilon_{i_{k_1}}) = c_{i_{k_1+1},i_{k_1}};\ \lambda_{i_{k_1+1}}^{(0)} = 0, i_{k_1+1} = 1, \upsilon_{i_{k_1+1}} = \upsilon_1. \quad (5)$$

Складывая равенства (4) и учитывая (5), имеем

$$l(\upsilon_1\upsilon_{k_1}\ldots\upsilon_{i_2}\upsilon_{i_1}) = \lambda_{i_1}^{(k_1)},$$

т.е. $\upsilon_1\upsilon_{k_1}\ldots\upsilon_{i_2}\upsilon_{i_1}$ - искомый минимальный путь из υ_1 в υ_{i_1} в нагруженном орграфе D. Заметим, что в этом пути ровно k_1 дуг Следовательно, мы определили путь с минимальным числом дуг среди всех минимальных путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Замечание. Номера $i_2, i_3, ..., i_{k_1}$, удовлетворяющие (4) вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных путей из υ_1 в υ_{i_1} с минимальным числом дуг среди минимальных, путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Замечание. Алгоритм можно модифицировать, с тем чтобы определить минимальный путь из υ_1 в заданную вершину υ_{i_1} среди путей из υ_1 в υ_{i_1}, содержащих не более k_0 дуг, где k_0 - заданное число, $k_0 \geq 1$. Для этого в алгоритме вместо $\lambda_{i_1}^{(n-1)}$ следует воспользоваться $\lambda_{i_1}^{(k_0)}$.

Пример 83.

Определить минимальный путь из v_1вv_6 в нагруженном орграфе *D*, изображенном на рисунке 35.

Решение.

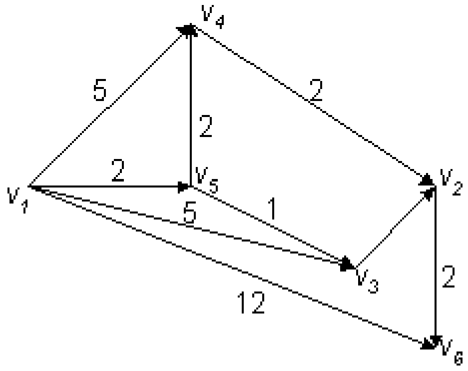


Рис. 35.

Составим матрицу *C(D)* длин дуг нагруженного орграфа *D*(табл. 68). Справа от матрицы *C(D)* припишем шесть столбцов, которые будем определять, используя рекуррентное соотношение (2) и исходя из (1).

Величина $\lambda_6^{(5)} = 7$ выражает длину минимального пути из v_1 в v_6 в нагруженном орграфе *D*. Найдем минимальное число $k_1 \geq 1$, при котором выполняется равенство $\lambda_6^{(k_1)} = \lambda_6^{(5)}$. Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех минимальных путей из v_1вv_6 в нагруженном графе *D* равняется 4. Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5, где $i_1 = 6$, удовлетворяющих (4) (для этого используем формулу (2)).

Таблица 68

	v_1	v_2	v_3	v_4	v_5	v_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
v_1	∞	∞	5	5	2	12		0	0	0	0	0	0
v_2	∞	∞	∞	∞	∞	2		∞	∞	7	5	5	5
v_3	∞	2	∞	∞	∞	∞		∞	5	3	3	3	3
v_4	∞	2	∞	∞	∞	∞		∞	5	4	4	4	4

v_5	∞	∞	1	2	∞	∞		∞	2	2	2	2	2
v_6	∞	∞	∞	∞	∞	∞		∞	12	12	9	7	7

Получаем, что в качестве такой последовательности надо взять номера 6, 2, 3, 5, 1, так как

$$\lambda_2^{(3)} + \tilde{n}_{2,6} = 5 + 2 = 7 = \lambda_6^{(4)};$$
$$\lambda_3^{(2)} + c_{3,2} = 3 + 2 = 5 = \lambda_2^{(3)};$$
$$\lambda_5^{(1)} + c_{5,3} = 2 + 1 = 3 = \lambda_3^{(2)};$$
$$\lambda_1^{(0)} + c_{1,5} = 0 + 2 = 2 = \lambda_5^{(1)}.$$

Тогда $v_1v_5v_3v_2v_6$ – искомый минимальный путь из v_1вv_6 в нагруженном орграфе D, причем он содержит минимальное число дуг среди всех возможных минимальных путей из v_1вv_6 .

3.2.4. Эйлеровы цепи и циклы

Классической в теории графов является следующая задача. В городе Кенигсберге имеется два острова, соединенных семью мостами с берегами реки Преголь и друг с другом так, как показано на рисунке 36. Задача состоит в следующем: осуществить прогулку по городу таким образом, чтобы, пройдя по одному разу по каждому мосту, вернуться обратно. Решение этой задачи сводится к нахождению некоторого специального маршрута в графе.

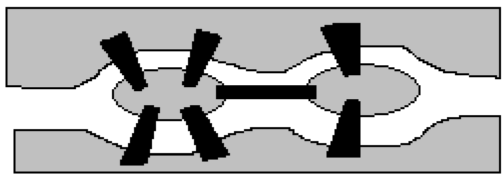


Рис. 36.

Пусть G – псевдограф.

Определение. Цепь (цикл) в G называется ***эйлеровой (эйлеровым)****, если она (он) проходит по одному разу через каждое ребро псевдографа G.*

Поставим в соответствие схеме мультиграф G, изображенный на рисунке 37, в котором каждой части суши соответствует вершина, а каждому мосту – ребро, соединяющее соответствующие вершины. На языке теории графов задача звучит следующим образом: найти эйлеров цикл в мультиграфе G.

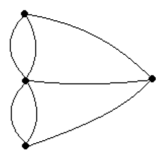


Рис. 37.

Определение. Граф является ***эйлеровы****м, если он содержит эйлеров цикл.*

Рассмотрим вопрос о наличии эйлеровой цепи и цикла в псевдографе.

Теорема. Связный граф является эйлеровым тогда и только тогда, когда каждая вершина имеет четную локальную степень.

Теорема. Связный граф содержит эйлерову цепь тогда и только тогда, когда ровно две вершины имеют нечетную локальную степень.

Завершая рассмотрение эйлеровых графов, рассмотрим алгоритм построения эйлеровой цепи в данном эйлеровом графе. Этот метод известен под названием ***алгоритма Флёри***.

Теорема. Пусть G – эйлеров граф; тогда следующая процедура всегда возможна и приводит к эйлеровой цепи графа G. Выходя из произвольной вершины u, идем по ребрам графа произвольным образом, соблюдая лишь следующие правила:

- *стираем ребра по мере их прохождения и стираем также изолированные вершины, которые при этом образуются;*
- *на каждом этапе идем по мосту только тогда, когда нет других возможностей.*

Пример 84.

Любой простой полный граф с нечетным количеством вершин является эйлеровым. Любой циклический граф является эйлеровым. Граф, являющийся колесом, не является эйлеровым.

3.2.5. Гамильтоновы цепи и циклы

Пусть G – псевдограф.

*Определение. Цепь (цикл) в G называется **гамильтоновой (гамильтоновыми)**, если она (он) проходит по одному разу через каждую вершину псевдографа G.*

*Определение. Граф является **гамильтоновым**, если он содержит гамильтонов цикл.*

С понятием гамильтоновых циклов тесно связана так называемая задача коммивояжера: в нагруженном графе *G* определить гамильтонов цикл минимальной длины (иными словами, коммерсант должен совершить поездку по городам и вернуться обратно, побывав в каждом городе ровно один раз, и при этом стоимость такой поездки должна быть минимальной).

Приведем ***теорему Дирака***, которая отвечает на вопрос: существует ли в графе гамильтонов цикл.

Теорема. Если в простом графе с n (≥ 3) вершинами локальная степень каждой вершины не менее n/2, то граф является гамильтоновым.

Пример 85.

Любой простой полный граф является гамильтоновым. Любой циклический граф является гамильтоновым. Граф, являющийся колесом, является гамильтоновым.

3.3. Деревья и циклы

3.3.1. Определение и свойства деревьев

*Определение. Граф G называется **деревом**, если он является связным и не имеет циклов. Граф G, все компоненты связности которого являются деревьями, называется **лесом**.*

Пример 86.

Граф G_1 является деревом (рис.38). Граф G_2 является лесом (рис. 38), он содержит три связные компоненты, каждая из которых является деревом.

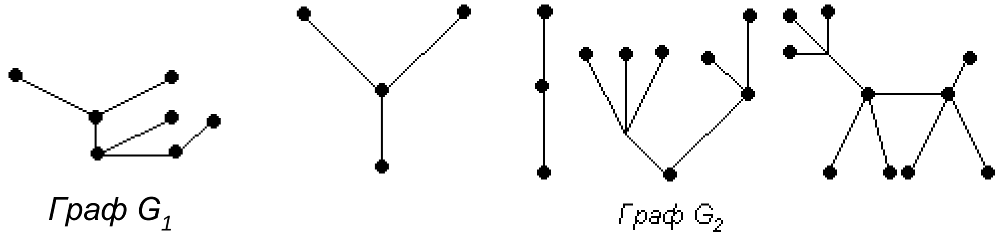


Рис. 38.

Следующие утверждения эквивалентны:

- граф *G* есть дерево;
- граф *G* является связным и не имеет простых циклов;
- граф *G* является связным и число его ребер ровно на единицу меньше числа вершин;
- любые две различные вершины графа *G* можно соединить единственной (и притом простой) цепью;
- граф *G* не содержит циклов, но, добавляя к нему любое новое ребро, получаем ровно один (с точностью до направления обхода и начальной вершины обхода) и притом простой цикл (проходящий через добавляемое ребро).

Утверждение. Если у дерева G есть, по крайней мере, одно ребро, то у него обязательно найдется висячая вершина.

Утверждение. Пусть G – дерево. Тогда любая цепь в G будет простой.

3.3.2. Остовное дерево связного графа

Определение. ***Остовным деревом*** *связного графа G называется любой его подграф, содержащий все вершины графа G и являющийся деревом.*

Пусть *G* связный граф. Тогда остовное дерево графа *G* (если оно существует) должно содержать $n(G) - 1$ ребер. Таким образом, любое остовное дерево графа *G* есть результат удаления из *G* ровно $m(G)-(n(G)-1)=m(G)-n(G)+1$ ребер.

Определение. Число $m(G)-n(G)+1$ называется ***цикломатическим числом*** *связного графа G и обозначается через $v(G)$.*

Замечание. Понятие остовного дерева и цикломатического числа аналогичным образом определяется и для произвольного связного псевдографа G.

Покажем существование остовного дерева для произвольного связного псевдографа $G=(V,X)$, описав алгоритм его выделения.

Шаг 1. Выбираем в *G* произвольную вершину u_1, которая образует подграф G_1 псевдографа *G*, являющийся деревом. Полагаем $i=1$.

Шаг 2. Если $i=n$, где $n=n(G)$, то задача решена, и G_i – искомое остовное дерево псевдографа *G*. В противном случае переходим к шагу 3.

Шаг 3. Пусть уже построено дерево G_i, являющееся подграфом псевдографа *G* и содержащий некоторые вершины $u_1, \ldots, u_i$, где $1 \le i \le n-1$. Строим граф G_{i+1}, добавляя к графу G_i новую вершину $u_{i+1} \in V$, смежную в *G* с некоторой вершиной u_j графа G_i, и новое ребро (u_{i+1}, u_j) (в силу связности *G* и того обстоятельства, что $i<n$, указанная вершина u_{i+1} обязательно

найдется). Согласно утверждению граф G_{i+1} также является деревом. Присваиваем $i:=i+1$ и переходим к шагу 2.

Пример 87.

Используя алгоритм, выделим остовное дерево графа G, изображенного на рисунке 39.

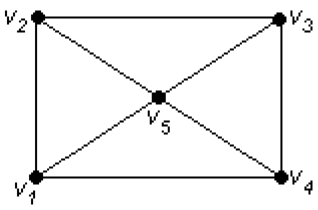


Рис. 39.

Решение.

На рисунке 40 приведена последовательность графов G_i, $i=1, 2, 3, 4, 5$, получаемых в результате выполнения алгоритма.

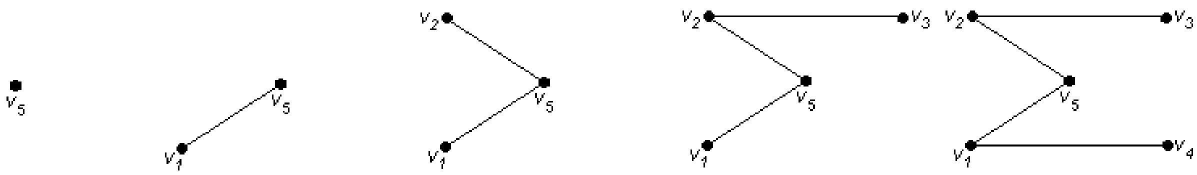


Рис. 40.

При этом в силу того, что $n(G)=5$, G_5 - остовное дерево графа G.

Замечание. Остовное дерево связного графа может быть выделено, вообще говоря, не единственным способом.

Общее число остовных деревьев связного графа может оказаться весьма большим. Например, для полного графа с n вершинами оно равно n^{n-2}.

В общем случае обозначим через G произвольный граф с n вершинами, m ребрами и k компонентами. Как уже было сказано, применяя выше описанный алгоритм, получим в результате граф, являющийся остовным лесом.

Определение. Число ребер, удаленных при построении остовного леса произвольного графа G, называется ***цикломатическим числом*** *(или цикломатическим рангом) графа G и обозначается через* $\gamma(G) = m - n + k$.

Пример 88.

Циклический ранг дерева равен нулю, а циклический ранг циклического графа равен единице.

Определение. ***Коциклическим рангом*** *графа G называется число ребер в его остовном дереве.*

С понятием остовного леса T графа G тесно связано понятие фундаментальной системы циклов, ассоциированной с T.

Определение. Если добавить к T любое не содержащиеся в нем ребро графа G, то получим единственный цикл; множество всех циклов, получаемых таким способом (т.е. путем *добавления по отдельности каждого ребра из G, не содержащегося в T), называется **фундаментальной системой циклов, ассоциированной с T**.*

3.3.3. Минимальные остовные деревья нагруженных графов

Пусть теперь каждому ребру $x \in X$связного графа $G=(V,X)$ с непустым множеством ребер X поставлена в соответствие величина $l(x)$ – длина ребра x, т.е. граф G является нагруженным. Приведем алгоритм, позволяющий найти остовное дерево графа G с минимальной суммой длин содержащихся в нем ребер (по сравнению со всеми другими остовными деревьями графа G).

*Определение. Остовное дерево связного нагруженного граф G с минимальной суммой длин содержащихся в нем ребер будем называть **минимальным остовным деревом** (МОД) графа G.*

Алгоритм выделения МОД нагруженного связного графа G:

Шаг 1. Выберем в графе G ребро минимальной длины. Вместе с инцидентными ему вершинами оно образует подграф G_2 графа G. Положим $i=2$.

Шаг 2. Если $i=n$, где $n=n(G)$, то задача решена, и G_i – искомое МОД графа G. В противном случае переходим к шагу 3.

Шаг 3. Строим граф G_{i+1}, добавляя к графу G_i новое ребро минимальной длины, выбранное среди всех ребер графа G, каждое из которых инцидентно к какой-нибудь вершине графа G_i и одновременно инцидентно какой-нибудь вершине графа G, не содержащейся в G_i. Вместе с этим ребром включаем в G_{i+1} и инцидентные ему вершины, не содержащуюся в G_i. Присваиваем $i:=i+1$ и переходим к шагу 2.

Пример 89.

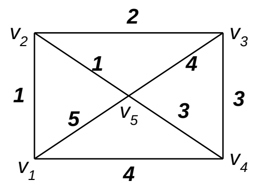


Рис. 41

Определить МОД нагруженного графа G, изображенного на рисунке 41, используя алгоритм.

Решение.

На рисунке 42 приведена последовательность графов G_i, $i=2,3,4,5$, получаемых в результате выполнения алгоритма. При этом в силу того, что $n(G)=5$, G_5 - МОД

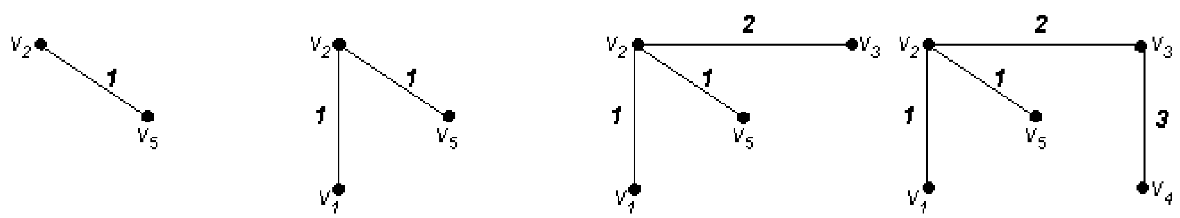


Рис.42.

графа G. Причем, $МОД(G) = 7$.

Замечание. Возможно выделить в графе несколько остовных деревьев, каждое из которых будет являться минимальным, при этом величина МОД для всех этих деревьев будет равной.

Замечание. Для выделения МОД нагруженного псевдографа G следует предварительно удалить из G петли, из кратных ребер оставить лишь ребра минимальной длины.

3.4. Транспортные сети

Определение. ***Транспортной сетью*** *называется орграф D = (V, X) с множеством вершин V, для которого выполняются условия:*

1) *существует одна и только одна вершина v_1, называемая* ***источником****, такая, что $D^{-1}(v_1) = 0$ (т.е. ни одна дуга не заходит в v_1);*
2) *существует одна и только одна вершина v_n, называемая* ***стоком****, такая, что $D(v_n) = 0$ (т.е. из v_n не исходит ни одной дуги);*
3) *каждой дуге $x\in X$поставлено в соответствие целое число $c(x) \geq 0$, называемое* ***пропускной способностью дуги****.*

Определение. Вершины в транспортной сети, отличные от источника и стока, называются ***промежуточными****.*

3.4.1. Поток в транспортной сети

Определение. Функция $\varphi(x)$, определенная на множестве X дуг транспортной сети D, и принимающая целочисленные значения, называется ***допустимым потоком*** *(или просто потоком) в транспортной сети D, если:*

1) *для любой дуги $x\in X$величина $\varphi(x)$, называемая* ***потоком по дуге****x, удовлетворяет условию $0 \leq \varphi(x) \leq c(x)$;*

2) *для любой промежуточной вершины v выполняется равенство*
$$\sum_{\varpi\in D^{-1}(v)}\varphi(\omega,v)-\sum_{\varpi\in D(v)}\varphi(v,\omega)=0$$
т.е. сумма потоков по дугам, заходящими в v, равна сумме потоков по дугам, исходящими из v.

Определение. ***Величиной потока φ*** *в транспортной сети D называется величина φ, равная сумме потоков по всем дугам, заходящим в v_n, или, что то же самое, величина, равная сумме потоков по всем дугам, исходящим из v_1,* т.е.

$$\overline{\varphi}=\sum_{v\in D^{-1}(v_n)}\varphi(v,v_n)=\sum_{v\in D(v_1)}\varphi(v_1,v)$$

Пусть φ - допустимый поток в транспортной сети *D*.

Определение. Дуга$x\in X$ называется ***насыщенной****, если поток по ней равен ее пропускной способности, т.е. если $\varphi(x) = c(x)$. Поток φ называется* ***полным****, если любой путь в D из v_1 в v_nсодержит, по крайней мере, одну насыщенную дугу.*

Определение. Поток φ называется ***максимальным****, если его величина принимает максимальное значение по сравнению с другими допустимыми потоками в транспортной сети D.*

Очевидно, что максимальный поток φ обязательно является полным. Обратное неверно. Существуют полные потоки, не являющиеся максимальными. Тем не менее, полный поток можно рассматривать как некоторое приближение к максимальному потоку.

Алгоритм построения полного потока в транспортной сети D:

Шаг 1. Полагаем $\forall x \in X \varphi(x) = 0$ (т.е. начинаем с нулевого потока). Кроме того, полагаем $D` = D$.

Шаг 2. Удаляем из орграфа $D`$ все дуги, являющиеся насыщенными при потоке $\varphi$в транспортной сети $D$. Полученный орграф снова обозначаем через $D`$.

Шаг 3. Ищем в $D`$ простую цепь η из v_1 в v_n. Если такой цепи нет, то φ - искомый полный поток в транспортной сети D. В противном случае переходим к шагу 4.

Шаг 4. Увеличиваем поток $\varphi(x)$ по каждой дуге x из η на одинаковую величину $\boldsymbol{a} > 0$ такую, что, по крайней мере, одна дуга из η оказывается насыщенной, а потоки по остальным дугам из η не превышают их пропускных способностей. При этом величина потока также увеличивается на $\boldsymbol{a}$, а сам поток φ в транспортной сети Dостается допустимым (поскольку в каждую промежуточную вершину, содержащуюся в η, дополнительно вошло $\boldsymbol{a}$ единиц потока и из нее вышло также $\boldsymbol{a}$ единиц потока). После этого переходим к шагу 2.

Пример 90.

Построить полный поток в транспортной сети, изображенной на рисунке 43.

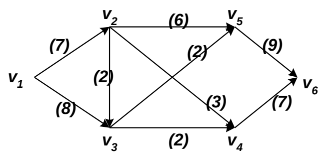


Рис. 43.

Решение.

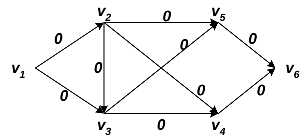
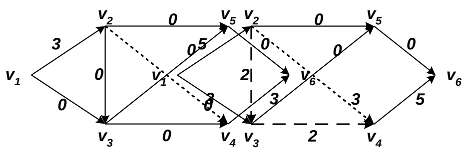


Рис. 44.

Начинаем с нулевого потока (рис. 44). Пошагово выделяем простые цепи и увеличиваем поток по ним таким образом, чтобы хотя бы одна дуга в каждой из них стала насыщенной. Полученную насыщенную дугу помечаем пунктиром, помня, что по насыщенной дуге больше идти нельзя.



Ри

Рис. 46.

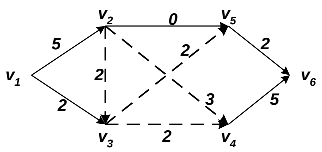


Рис. 47.

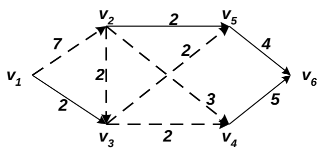


Рис. 48.

1. $\eta_1 = v_1v_2v_4v_6$, $a_1 = min\{7, 3, 7\} = 3$ (рис. 45).

2. $\eta_2 = v_1v_2v_3v_4v_6$, $a_2 = min\{7\text{-}3, 2, 2, 7\text{-}3\} = 2$ (рис. 46).

3. $\eta_3 = v_1v_3v_5v_6$, $a_3 = min\{8, 2, 9\} = 2$ (рис. 47).

4. $\eta_4 = v_1v_2v_5v_6$, $a_4 = min\{7\text{-}5, 6, 9\text{-}2\} = 2$ (рис. 48).

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и $\varphi = 3+2+2+2=9$.

3.4.2. Орграф приращений

Выделим для заданной транспортной сети D и допустимого потока φв этой сети орграф приращений $I(D, \varphi)$, имеющий те же вершины, что и сеть D. Каждой дуге $x = (v, \omega) \in X$

транспортной сети D в орграфе приращений $I(D,\ \varphi)$ соответствуют две дуги: x и $x` = (\omega,v)$ - дуга, противоположная по направлению дуге $x$. Припишем дугам $x = (v,\ \omega)\in X$, $x` = (\omega,\ v)$ орграфа приращений $I(D,\ \varphi)$ длину l:

$$l(x) = \begin{cases} 0, \text{ если } \varphi(x) < c(x); \\ \infty, \text{ если } \varphi(x) = c(x). \end{cases}$$

$$l'(x) = \begin{cases} 0, \text{ если } \varphi(x) > 0; \\ \infty, \text{ если } \varphi(x) = 0. \end{cases}$$

т.е. орграф $I(D,\ \varphi)$ является нагруженным. При этом, очевидно, что длинна любого пути из v_1 в v_n орграфе $I(D,\ \varphi)$ равна либо 0, либо ∞.

Пример 91.

Построить орграф приращений для данной транспортной сети и построенного полного потока в этой сети.

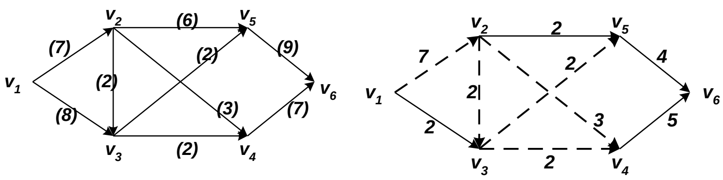


Рис. 49.

Решение.

Напомним, что орграф приращений имеет в два раза больше дуг, чем исходная транспортная сеть.

Для дуги прямой направленность вес равен 0, если дуга не является насыщенной, ∞ - в противном случае. Для дуги обратной направленности вес равен 0, если поток по ней не равен нулю, ∞ - в противном случае. На рисунке 50 изображен орграф приращений, соответствующий данному потоку в исходной транспортной сети.

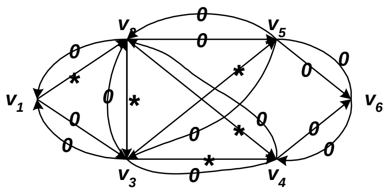


Рис. 50.

3.4.3. Алгоритм построения максимального потока в транспортной сети

Алгоритм построения максимального потока в транспортной сети D:

Шаг 1. Полагаем $i = 0$. Пусть φ_0 - любой допустимый поток в транспортной сети D. (например, полный; можно начинать с нулевого потока: $\varphi_0(x),\ x\in X$).

Шаг 2. По сети D и потоку φ_i строим орграф приращений $I(D,\ \varphi_i)$.

Шаг 3. Находим простую цепь η_i, являющуюся минимальным путем из v_1 в v_nв нагруженном орграфе $I(D,\ \varphi_i)$ (например, используя алгоритм Форда - Беллмана). Если длина этой цепи равна ∞, то поток φ_i максимален, и работа алгоритма закончена. В противном случае увеличиваем поток вдоль цепи η_i на максимально допустимую величину $a_i > 0$, где $a_i \in Z$(прибавляя ее для каждой дуги $x\in X$, через которую проходит цепь η_i, к уже имеющейся величине потока по дуге x, если направления x и η_i совпадают, и, вычитая, если направления x и η_i противоположны).

Пример 92.

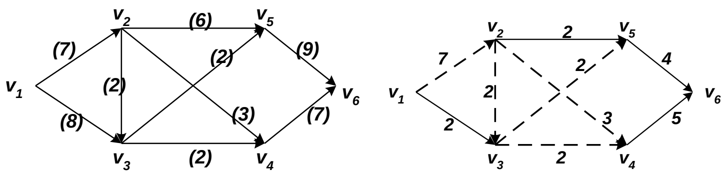


Рис. 51.

Выяснить является ли полный поток максимальным (рис. 51), если нет, то дополнить его до максимального.

Решение.

Для решения используем алгоритм Форда-Беллмана нахождения минимального пути в нагруженном орграфе.

Построим матрицу длин дуг *C(D)* и λ-матрицу (табл. 69).

Таблица 69

	v_1	v_2	v_3	v_4	v_5	v_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
v_1	∞	∞	0	∞	∞	∞		0	0	0	0	0	0
v_2	0	∞	∞	∞	0	∞		∞	∞	0	0	0	0
v_3	0	0	∞	0	0	∞		∞	0	0	0	0	0
v_4	∞	0	0	∞	∞	0		∞	∞	∞	∞	∞	0
v_5	∞	0	0	∞	∞	0		∞	∞	∞	0	0	0
v_6	∞	∞	∞	0	0	∞		∞	∞	∞	∞	0	0

Поскольку $\lambda_6^{(5)} = 0$, то существует нулевой путь из источника v_1 в сток v_6. Значит, полный поток не является максимальным. Дополним его до максимального. Для этого найдем путь нулевой длины: $\lambda_6^{(5)} = 0 = \lambda_6^{(4)}$

Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех нулевых путей из v_1вv_6 в орграфе приращений равняется 4. Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5,где $i_1 = 6$.

Получаем, что в качестве такой последовательности надо взять номера 1, 3, 2, 5, 6, так как

$$\lambda_5^{(3)} + \tilde{n}_{5,6} = 0 + 0 = 0 = \lambda_6^{(4)};$$
$$\lambda_2^{(2)} + c_{2,5} = 0 + 0 = 0 = \lambda_5^{(3)};$$
$$\lambda_3^{(1)} + c_{3,2} = 0 + 0 = 0 = \lambda_2^{(2)};$$
$$\lambda_3^{(0)} + c_{1,3} = 0 + 0 = 0 = \lambda_3^{(1)}.$$

Тогда $v_1v_3v_2v_5v_6$ – искомый нулевой путь из v_1вv_6 . Дуги, совпадающие по направлению с дугами исходной транспортной сети помечаем знаком «+», не совпадающие – знаком «-».

Получаем, $\underset{+}{v_1 \to} \underset{-}{v_3 \to} \underset{+}{v_2 \to} \underset{+}{v_5 \to} v_6$. Теперь необходимо найти величину, которую будем перемещать по полученному контуру. Для этого, каждому ребру в контуре поставим в соответствие число $\alpha(i, j)$, которое находим по следующему правилу: если направление ребра *(i, j)* в контуре совпадает с направлением ребра *x* в транспортной сети, то $\alpha(i, j)=c(x)-\varphi(x)$; если направление ребра в контуре не совпадает с направлением ребра в транспортной сети, то $\alpha(i, j)=\varphi(x)$. Итак, из чисел $\alpha(i, j)$ найдем минимальное:

$$min\{8-2=6, 2, 6-2=4, 9-4=5\}=2.$$

Перемещаем по контуру 2.

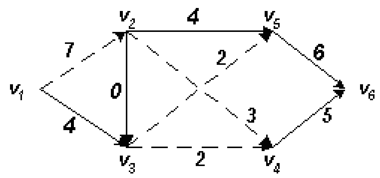


Рис. 52.

В результате получаем поток, изображенный на рисунке 52.

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и $\varphi = 11$. Проверим, является ли он максимальным.

Построим матрицу длин дуг *C(D)* и λ-матрицу (табл. 70).

Таблица 70

	v_1	v_2	v_3	v_4	v_5	v_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
v_1	∞	∞	0	∞	∞	∞		0	0	0	0	0	0
v_2	0	∞	0	∞	0	∞		∞	∞	∞	∞	∞	∞
v_3	0	∞	∞	∞	∞	∞		∞	0	0	0	0	0
v_4	∞	0	0	∞	∞	0		∞	∞	∞	∞	∞	∞
v_5	∞	0	0	∞	∞	0		∞	∞	∞	∞	∞	∞
v_6	∞	∞	∞	0	0	∞		∞	∞	∞	∞	∞	∞

Так как $\lambda_6^{(5)} = \infty$, то нулевого пути из v_1 в v_6 не существует. Значит, поток $\overline{\varphi} = 11$ является максимальным.

2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации ЗАДАНИЕ (практическое) **к зачету:**

1. Дискретная математика как наука. Области ее применения.
2. Понятие множества. Мощность множества. Способы задания множества.
3. Операции над множествами (объединение, пересечение, разность, дополнение). Диаграммы Эйлера-Венна (изображение операций над множествами).
4. Свойства операций над множествами. Доказательства свойств.
5. Булеан множества. Теорема: $|P(X)| = 2^n$ с доказательством. Алгоритм построения булеана.

6. Декартово произведение множеств. N-местное отношение. Бинарное отношение. Примеры.
7. Область определения бинарного отношения. Область значений бинарного отношений. Обратное отношение для бинарного отношения. Образ множества. Прообраз множества.
8. Матрица бинарного отношения. Ее свойства.
9. Свойства бинарных отношений. Определение свойств бинарных отношений матричным методом.
10. Функция. Частичная функция. Инъекция. Сюръекция. Биекция.
11. Отношение эквивалентности. Классы эквивалентности.
12. Покрытие множества. Разбиение множества. Фактор-множество.
13. Теорема о связанности разбиения множества и отношения эквивалентности с доказательством.
14. Классификация отношений. Отношения частичного порядка. Топологическая сортировка. Диаграммы Хассе.
15. Комбинаторика, ее основные задачи. Правило суммы. Правило произведения.
16. Число размещений без повторений. Доказательство. Число размещений с повторениями. Доказательство.
17. Число сочетаний без повторений. Доказательство. Число сочетаний с повторениями. Доказательство.
18. Биномиальные коэффициенты. Элементарные свойства биномиальных коэффициентов.
19. Бином Ньютона и треугольник Паскаля. Способы их использования.
20. Формула включения и исключения. Форма записи формулы включения и исключения с использованием свойств элементов множества.
21. Размещения заданного состава. Полиномиальная теорема.
22. Числа Фибоначчи, их свойства
23. Основные определения теории вероятностей. Классическое определение вероятности.
24. Условные вероятности. Формула полной вероятности. Формула Байеса и ее использование.
25. Случайные величины и распределения вероятностей.
26. Математическое ожидание и дисперсия, их основные свойства.
27. Определение энтропии случайной схемы, ее свойства. Аксиоматическое определение энтропии.
28. Строковые данные в различных разделах математики и приложениях. Основные операции над строками.
29. Лексикографическое сравнение строк. Типичные задачи, решаемые со строками. Методы поиска образца в строке. Классификация функций от строк.
30. Графы. Основные понятия и определения. Способы представления.
31. Представление графов матрицами инцидентности и смежности. Свойства данных матриц.
32. Понятие связного графа, компоненты связности и сильной связности. Метрические характеристики графов.
33. Нагруженные графы. Постановка задачи коммивояжера.
34. Задача о кратчайшем пути в графе (алгоритм фронта волны, алгоритмы Форда-Беллмана и Дейкстры).
35. Деревья. Характеристическое свойство дерева. Алгоритм нахождения кратчайшего остовного дерева (алгоритм Краскала).
36. Алгоритм нахождения максимального потока. Теорема Форда-Фалкерсона.
37. Функциональные системы с операциями. Булева алгебра. Способы задания булевых функций. Элементарные функции булевой алгебры.
38. Булева алгебра Формулы булевой алгебры. Равносильные формулы. Двойственные функции. Принцип двойственности.
39. Представление булевых функций в классе СДНФ, СКНФ. Алгоритм построения СДНФ, СКНФ.

40.Полнота в алгебре логики. Примеры полных систем. Принцип суперпозиции.
41.Замкнутые классы алгебры логики (TO, T1, S, M, L).
42.Критерий функциональной полноты в алгебре логики. Теорема Поста.
43.Минимизация булевых функций в классе ДНФ. Интервал, максимальный интервал, простая импликанта. Сокращенная ДНФ. Минимальная ДНФ.Примеры использования алгоритмов минимизации.
44.Функциональная система: *k*-значная логика. Элементарные функции *k*-значной логики.
45.Определение кодирования. Свойства кодирования. Код сообщения. Побуквенное кодирование. Элементарные коды. Алфавитный код. Равномерное кодирование. Разделимый код.
46.Схема кодирования. Префиксный код. Взаимно однозначное кодирование. Критерий взаимной однозначности алфавитного кодирования.
47.Взаимно однозначное кодирование. Неравенство Крафта-Макмиллана. Коды с минимальной избыточностью.
48.Кодовое дерево. Насыщенная вершина кодового дерева. Насыщенное кодовое дерево. Исключительная вершина. Порядок ветвления исключительной вершины.
49.Бинарный код Хэмминга. Схема кодирования. Схема декодирования. Самокорректирующиеся коды Хэмминга.
50.Конечный автомат. Определение. Использование конечных автоматов в программировании.
51.Марковская цепь. Основные определения. Граф переходов. Классификация состояний марковской цепи.
52.Процесс принятия решений. Модель динамического программирования. Уравнение Беллмана. Процессы в информатике.
53.Производящие функции. Асимптотика.

3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90% от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76% от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы –75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60% от общего объёма заданных вопросов.

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

«Неудовлетворительно/не зачтено» – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения* *задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

- негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.

- недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.

Критерии формирования оценок по зачету с оценкой

«Отлично/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний, не допустил логических и фактических ошибок

«Хорошо/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний; допустил незначительные ошибки и неточности.

«Удовлетворительно/зачтено» – студент допустил существенные ошибки.

«Неудовлетворительно/не зачтено» – студент демонстрирует фрагментарные знания изучаемого курса; отсутствуют необходимые умения и навыки, допущены грубые ошибки.

Экспертный лист
оценочных материалов для проведения промежуточной аттестации по дисциплине «Дискретная математика»

Направление подготовки / специальность

09.03.01. «Информатика и вычислительная техника»
(код и наименование)

Направленность (профиль)/специализация

____
(наименование)

Бакалавр
квалификация выпускника

<table>
<tr><td colspan="4">1. Формальное оценивание</td></tr>
<tr><td colspan="2">Показатели</td><td>Присутствуют</td><td>Отсутствуют</td></tr>
<tr><td colspan="2">Наличие обязательных структурных элементов:</td><td>+</td><td></td></tr>
<tr><td colspan="2">–титульный лист</td><td>+</td><td></td></tr>
<tr><td colspan="2">–пояснительная записка</td><td>+</td><td></td></tr>
<tr><td colspan="2">– типовые оценочные материалы</td><td>+</td><td></td></tr>
<tr><td colspan="2">–методические материалы, определяющие процедуру и критерии оценивания</td><td>+</td><td></td></tr>
<tr><td colspan="4">Содержательное оценивание</td></tr>
<tr><td>Показатели</td><td>Соответствует</td><td>Соответствует частично</td><td>Не соответствует</td></tr>
<tr><td>Соответствие требованиям ФГОС ВО к результатам освоения программы</td><td>+</td><td></td><td></td></tr>
<tr><td>Соответствие требованиям ОПОП ВО к результатам освоения программы</td><td>+</td><td></td><td></td></tr>
<tr><td>Ориентация на требования к трудовым функциям ПС (при наличии утвержденного ПС)</td><td>+</td><td></td><td></td></tr>
<tr><td>Соответствует формируемым компетенциям, индикаторам достижения компетенций</td><td>+</td><td></td><td></td></tr>
</table>

Заключение: ФОС рекомендуется/ не рекомендуется к внедрению; обеспечивает/ не обеспечивает объективность и достоверность результатов при проведении оценивания результатов обучения; критерии и показатели оценивания компетенций, шкалы оценивания обеспечивают/ не обеспечивают проведение всесторонней оценки результатов обучения.