Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Ярославский государственный университет им. П. Г. Демидова
Университетский колледж

Дискретная математика

Практикум

Рекомендовано
Научно-методическим советом университета для студентов, обучающихся по специальности Автоматизированные системы обработки информации и управления (по отраслям)

Ярославль 2010

УДК 517.929
ББК В 174я72
Д 48

Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2009/10 года

Рецензент
педагогический совет Университетского колледжа

Составитель И. А. Сурикова

Д 48 **Дискретная математика:** практикум / сост. И. А. Сурикова; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль : ЯрГУ, 2010. – 76 с.

Предназначен для студентов, обучающихся по специальности 230103.51 Автоматизированные системы обработки информации и управления (по отраслям) (дисциплина «Дискретная математика», блок ОПД), очной формы обучения.

УДК 517.929
ББК В 174я72

Учебное издание

Дискретная математика

Практикум

Составитель **Сурикова** Ирина Александровна

Редактор, корректор И. В. Бунакова
Верстка Е. Л. Шелехова

Подписано в печать 04.02.10. Формат 60×84 $^{1}/_{16}$.
Бум. офсетная. Гарнитура "Times NewRoman".
Усл. печ. л. 4,42. Уч.-изд. л. 2,63.
Тираж 50 экз. Заказ

Оригинал-макет подготовлен
в редакционно-издательском отделе Ярославского
государственного университета им. П. Г. Демидова.
Отпечатано на ризографе.
Ярославский государственный университет им. П. Г. Демидова.
150000, Ярославль, ул. Советская, 14.

Пояснительная записка

Практикум по дискретной математике предназначен для самостоятельной работы студентов. При изучении учебной дисциплины «Дискретная математика» у студентов формируется базовый уровень знаний для освоения других общепрофессиональных и специальных дисциплин. Данные материалы используются при изучении таких дисциплин, как «Основы алгоритмизации и программирования», «Архитектура ЭВМ и вычислительных систем», «Базы данных», «Теория вероятностей и математическая статистика», «Математические методы», «Технология разработки программных продуктов», «Разработка и эксплуатация удаленных баз данных», «Пакеты прикладных программ».

Дискретная математика является фундаментом математической кибернетики. Аппарат дискретной математики необходим при создании и эксплуатации современных ЭВМ, средств передачи и обработки информации, автоматизированных систем управления и проектирования; поэтому знание основ данной дисциплины абсолютно необходимо для современного специалиста в области информатики и вычислительной техники.

В результате изучения дисциплины *студент должен*

знать:

– аппарат алгебры логики и теорию булевых функций;

– основы теории множеств;

– логику предикатов и бинарных отношений;

– алгебру подстановок;

– основы алгебры вычетов;

– простейшие криптографические шифры;

– метод математической индукции;

– методику генерирования основных комбинаторных объектов;

– основы теории графов и теории автоматов;

уметь:

– строить таблицы истинности для формул логики и упрощать формулы логики;

– представлять булевы функции в виде формул заданного типа, проверять множество булевых функций на полноту;

– выполнять операции над множествами, применять аппарат теории множеств для решения задач;

– выполнять операции над предикатами, записывать области истинности предикатов, формализовывать предложения с помощью логики предикатов;

– исследовать бинарные отношения на заданные свойства;

– выполнять операции над подстановками;

– выполнять операции в алгебре вычетов;

– применять простейшие криптографические шифры;

– доказывать утверждения с помощью метода математической индукции;

– генерировать основные комбинаторные объекты;

– находить характеристики графов, выделять их структурные особенности, исследовать на заданные свойства, строить для них структурные представления заданных типов;

– строить автоматы с заданными свойствами.

Раздел 1. **Формулы логики**

В этом разделе студенты знакомятся с элементами математической логики. В ее основе лежит логика высказываний, имеющая дело с истинностью (или ложностью) простых описательных утверждений.

Высказывание – связное повествовательное предложение (утверждение), о котором можно сказать, истинно оно или ложно.

Основные логические операции

Отрицание высказывания a – высказывание, истинное, когда высказывание a ложно, и ложное – в противном случае.

Обозначение: $\bar{a}$, $\neg a$ (читается: “не a”, “неверно, что a”).

Отрицание определяется следующей таблицей:

a	$\bar{a}$
0	1
1	0

Конъюнкцией двух высказываний a и b называется высказывание, истинное, когда оба высказывания истинны, и ложное – во всех других случаях.

Обозначение: $a\&b$; $a\wedge b$; ab; $a\cdot b$ (читается: “a и b”).

Конъюнкция определяется следующей таблицей:

a	b	$a\wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкцией двух высказываний a и b называется высказывание, ложное в случае, когда оба высказывания ложны, и истинное – во всех других случаях.

Обозначение: $a \vee b$ (читается a или b).

Дизъюнкция определяется следующей таблицей:

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Импликацией двух высказываний a и b называется высказывание, ложное, когда a истинно, а b ложно; во всех других случаях – истинное.

Обозначение: $a \rightarrow b$ (читается: “если a то b”, “из a следует b”).

Импликация определяется следующей таблицей:

a	b	$a \rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Эквиваленцией двух высказываний a и b называется высказывание, истинное, когда a и b имеют одинаковые значения истинности, и ложное – в противном случае.

Обозначение: $a \sim b$ (читается: “a эквивалентно b”, “a тогда и только тогда, когда b”).

Эквиваленция определяется следующей таблицей:

a	b	$a \sim b$
0	0	1
0	1	0
1	0	0
1	1	1

Алгоритм построения таблицы истинности

1. Подсчитать количество переменных в формуле n.
2. Определить количество строк в таблице, которое равно 2^n.

3. Подсчитать количество логических операций в формуле и определить количество столбцов в таблице, которое равно количеству переменных плюс количество операций.

4. Записать названия столбцов таблицы в соответствии с последовательностью выполнения логических операций с учетом скобок и приоритетов ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\sim$, где самая сильная операция – отрицание).

5. Заполнить столбцы переменных наборами значений в порядке возрастания от (00…0) до (11…1), используя метод "последовательного половинного деления столбцов":

a) разделить столбец первой переменной пополам и заполнить верхнюю половину нулями, а нижнюю половину – единицами;

b) разделить каждую половину второго столбца пополам и заполнить полученные половины нулями и единицами и т. д.

6. Провести заполнение таблицы истинности по столбцам.

Пример 1.1. Построить таблицу истинности для формулы логики: $x \wedge \bar{y} \rightarrow (x \vee y) \wedge \bar{z}$.

1. Определим порядок действий в формуле:

$$x \overset{2}{\wedge} \bar{y} \overset{1}{\rightarrow} \overset{6}{\left(x \overset{3}{\vee} y \right)} \overset{5}{\wedge} \overset{4}{\bar{z}}$$

2. Пользуясь определениями операций $\neg$, $\wedge$, $\vee$ и $\rightarrow$, заполним таблицу:

x	y	z	$\bar{y}$	$x \wedge \bar{y}$	$x \vee y$	$\bar{z}$	$(x \vee y) \wedge \bar{z}$	$x \wedge \bar{y} \rightarrow (x \vee y) \wedge \bar{z}$
0	0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0	1
0	1	0	0	0	1	1	1	1
0	1	1	0	0	1	0	0	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	0	0	1	0	0	1

Дизъюнктивная и конъюнктивная нормальные формы

Элементарной конъюнкцией называется произвольная конъюнкция формул, каждая из которых есть переменная или отрицание переменной.

Дизъюнктивная нормальная форма (ДНФ) – это произвольная дизъюнкция элементарных конъюнкций.

Элементарной дизъюнкцией называется произвольная дизъюнкция формул, каждая их которых есть переменная или отрицание переменной.

Конъюнктивная нормальная форма (КНФ) – это произвольная конъюнкция элементарных дизъюнкций.

Методика построения таблицы истинности для ДНФ

Элементарная конъюнкция принимает значение “1”, если значения всех входящих в неё формул равны “1” (переменная без отрицания имеет значение “1”, переменная с отрицанием имеет значение “0”), и принимает значение “0” во всех остальных случаях.

Произвольная дизъюнкция формул принимает значение “0”, если значения всех входящих в нее формул равны “0” и принимают значение “1” во всех остальных случаях.

Пример 1.2. Построить таблицу истинности для ДНФ: $xy \vee x\bar{y}z \vee \bar{x}\,\bar{y}\,\bar{z}$.

x	y	z	xy	$x\bar{y}z$	$\bar{x}\,\bar{y}\,\bar{z}$	$xy \vee x\bar{y}z \vee \bar{x}\,\bar{y}\,\bar{z}$
0	0	0	0	0	1	1
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	0	0	1

Законы логики

1. Ассоциативность:

а) $a(bc)=(ab)c=abc$; б) $a \vee (b \vee c) = (a \vee b) \vee c = a \vee b \vee c$.

2. Коммутативность:

а) $ab=ba$; б) $a \vee b = b \vee a$.

3. Дистрибутивность:

а) $a(b \vee c) = ab \vee ac$; б) $a \vee bc = (a \vee b)(a \vee c)$.

4. Идемпотентность:

а) $aa = a$; б) $a \vee a = a$.

5. Закон двойного отрицания:

$\bar{\bar{a}} = a$.

6. Свойства констант 0 и 1:

а) $a \cdot 1 = a$; б) $a \cdot 0 = 0$; в) $a \vee 1 = 1$;

г) $a \vee 0 = a$; д) $\bar{0} = 1$; е) $\bar{1} = 0$.

7. Правила де Моргана:

а) $\overline{a \wedge b} = \bar{a} \vee \bar{b}$; б) $\overline{a \vee b} = \bar{a} \wedge \bar{b}$.

8. Закон противоречия:

$a \wedge \bar{a} = 0$.

9. Закон исключенного третьего:

$a \vee \bar{a} = 1$.

10. Зависимости между операциями:

$a \to b = \bar{a} \vee b$;

$a \sim b = (a \to b)(b \to a) = (\bar{a} \vee b)(a \vee \bar{b}) = ab \vee \bar{a}\bar{b}$.

11. Законы поглощения:

а) $a \vee ab = a$; *б)* $a(a \vee b) = a$.

12. Законы полупоглощения:

а) $a \vee \bar{a}b = a \vee b$; б) $\bar{a} \vee ab = \bar{a} \vee b$;

в) $a(\bar{a} \vee b) = ab$; г) $\bar{a}(a \vee b) = \bar{a}b$.

Методика упрощения формулы логики с помощью равносильных преобразований

1. С помощью зависимостей между операциями перейти к формуле, содержащей только дизъюнкцию, конъюнкцию и отрицание.

2. Пользуясь законами де Моргана и законом двойного отрицания, “спустить” все отрицания до переменных, т.е. перейти к формуле, содержащей отрицание не выше, чем над переменными.

3. Раскрыть скобки, пользуясь дистрибутивными законами и ассоциативностью конъюнкции и дизъюнкции.

4. Удалить лишние конъюнкции и повторения переменных в конъюнкциях, используя идемпотентность, законы противоречия, исключенного третьего, поглощения и полупоглощения.

5. Удалить константы с помощью свойств констант.

Пример 1.3. Упростить формулу логики с помощью равносильных преобразований $(x \sim y) \to \overline{xy}$.

1. $(x \sim y) \to \overline{xy} = (xy \vee \bar{x}\,\bar{y}) \to \overline{xy} = \overline{xy \vee \bar{x}\,\bar{y}} \vee \overline{xy} =$
2. $= \overline{xy} \cdot \overline{\bar{x}\,\bar{y}} \vee (\bar{x} \vee \bar{y}) = (\bar{x} \vee \bar{y}) \cdot (\bar{\bar{x}} \vee \bar{\bar{y}}) \vee (\bar{x} \vee \bar{y}) = (\bar{x} \vee \bar{y}) \cdot (x \vee y) \vee (\bar{x} \vee \bar{y}) =$
3. $= \bar{x}x \vee \bar{x}y \vee \bar{y}x \vee \bar{y}y \vee \bar{x} \vee \bar{y} =$
4. $= 0 \vee 0 \vee \bar{x} \vee \bar{y} =$
5. $= \bar{x} \vee \bar{y}$

Упражнения

1. Построить таблицу истинности для формулы логики:

а) $(x \sim y) \to \overline{(x \vee z)y}$;
б) $\left(\overline{xz} \vee (x \to y)\right) \sim z$;
в) $(x \vee y)\,(y \to z) \sim \bar{z}$;
г) $((x \to y) \sim y \vee \bar{x})\,z$;
д) $(x \sim y)\,z \vee (x \to \bar{z})$;
е) $(y \sim x) \vee z\,(x \to y)$;
ж) $(x \vee y \to z) \sim \overline{yz}$;
з) $(xy \sim z) \to (\bar{x} \vee z)$.

2. Построить таблицу истинности для ДНФ:

а) $x\bar{y} \vee xz \vee \bar{x}y\bar{z}$;
б) $\bar{x}y \vee \bar{y}z \vee x\bar{y} \cdot \bar{z}$;
в) $x\bar{y} \vee xz \vee \bar{x}y\bar{z}$;
г) $\bar{x}y \vee \bar{y}z \vee x\bar{y} \cdot \bar{z}$;
д) $\bar{x}z \vee \bar{y} \cdot \bar{z} \vee xy\bar{z}$;
е) $\bar{x} \cdot \bar{y} \vee x\bar{z} \vee \bar{x}yz$;
ж) $\bar{x}z \vee \bar{y} \cdot \bar{z} \vee xy\bar{z}$;
з) $x\bar{z} \vee y\bar{z} \vee \bar{x}yz$.

3. Упростить формулу логики с помощью равносильных преобразований:

а) $\overline{\overline{xy}} \vee (x \to y)x$;
б) $\overline{(x \to y)\left(y \to \bar{x}\right)}$;
в) $(x \vee y)\,(x \sim y)$;
д) $xy\,(x \sim y)$;
е) $(x \to \bar{y})\,(x \sim y)$;
ж) $\left(x \to \bar{y}\right) \vee \overline{x \vee y}$;

г) $\overline{xy(x \to y)}$; з) $\left(\overline{\bar{x} \vee y} \to (x \vee y)\right)y$.

Раздел 2. **Булевы функции**

Изучение исчисления высказываний как алгебраической системы составляет предмет алгебры логики, или булевой алгебры. Мы освоим язык алгебры логики, ее законы, научимся строить и упрощать булевы функции и выполнять операции над сложными высказываниями.

Совершенная ДНФ

ДНФ называется **совершенной** и обозначается **СДНФ**, если каждая переменная формулы входит в каждую элементарную конъюнкцию ровно один раз с отрицанием или без отрицания и в ДНФ нет одинаковых элементарных конъюнкций.

Построение СДНФ по таблице истинности

1. Выделить строки таблицы, на которых значение функции равно единице.
2. Для каждой такой строки выписать элементарные конъюнкции всех переменных.
3. Над переменными, значения которых равны нулю, поставить отрицания.
4. Все элементарные конъюнкции соединить знаками дизъюнкции.

Пример 2.1.

x	y	z	f	
0	**0**	**0**	**1**	$\bar{x}\,\bar{y}\,\bar{z} \vee$
0	**0**	**1**	**1**	$\bar{x}\,\bar{y}z \vee$
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	**0**	**1**	**1**	$x\bar{y}z \vee$
1	1	0	0	
1	**1**	**1**	**1**	xyz

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

Совершенная КНФ

КНФ называется **совершенной** и обозначается **СКНФ**, если каждая переменная формулы входит в каждую элементарную дизъюнкцию ровно один раз с отрицанием или без отрицания и в КНФ нет одинаковых элементарных дизъюнкций.

Построение СКНФ по таблице истинности

1. Выделить строки таблицы, на которых значение функции равно нулю.

2. Для каждой такой строки выписать элементарные дизъюнкции всех переменных.

3. Над переменными, значения которых равны единице, поставить отрицания.

4. Все элементарные дизъюнкции соединить знаками конъюнкции.

Пример 2.2.

x	y	z	f	
0	0	0	1	
0	0	1	1	
0	**1**	**0**	**0**	$x \vee \bar{y} \vee z$
0	**1**	**1**	**0**	$x \vee \bar{y} \vee \bar{z}$
1	**0**	**0**	**0**	$\bar{x} \vee y \vee z$
1	0	1	1	
1	**1**	**0**	**0**	$\bar{x} \vee \bar{y} \vee z$
1	1	1	1	

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

Сокращенные ДНФ

Получение сокращенной ДНФ из произвольной ДНФ (метод Блейка)

1. Применить операции обобщенного склеивания по правилу: $xK_1 \vee \bar{x}K_2 = xK_1 \vee \bar{x}K_2 \vee K_1K_2$.

2. Применить операции поглощения по правилу: $K_1 \vee K_1K_2 = K_1$.

Пример 2.3. Найти сокращенную ДНФ методом Блейка для функции $f = x_1x_2 \vee \overline{x_1}x_3 \vee \overline{x_2}x_3$.

$$f = x_1x_2 \vee \overline{x_1}x_3 \vee \overline{x_2}x_3 \overset{1}{=} x_1x_2 \vee \overline{x_1}x_3 \vee x_2x_3 \vee \overline{x_2}x_3 \vee x_1x_3 \vee x_3 \overset{2}{=} x_1x_2 \vee x_3.$$

Получение сокращенной ДНФ из КНФ (метод Нельсона)

1. Раскрыть скобки, пользуясь законом дистрибутивности.

2. Вычеркнуть из получившейся ДНФ буквы и слагаемые, используя правила $x\bar{x} = 0$, $xx = x$, $x \vee x = x$, $0 \vee x = x$, $K_1 \vee K_1K_2 = K_1$.

Пример 2.4. Найти сокращенную ДНФ для функции $f = (x_1 \vee x_2)(\overline{x_1} \vee x_2 \vee x_3)$.

$$f = (x_1 \vee x_2)(\overline{x_1} \vee x_2 \vee x_3) \overset{1}{=} x_1\overline{x_1} \vee x_1x_2 \vee x_1x_3 \vee \overline{x_1}x_2 \vee x_2x_2 \vee x_2x_3 \overset{2}{=} x_1x_3 \vee x_2.$$

Получение сокращенной ДНФ геометрически

1. На единичном *n*-мерном кубе отметить двоичные наборы, на которых значение функции равно единице (множество N_f).

2. Выписать грани, содержащиеся в N_f и не содержащиеся в других гранях, составленных из вершин множества N_f.

3. Каждой из полученных граней сопоставить элементарную конъюнкцию.

4. Записать дизъюнкцию полученных конъюнкций.

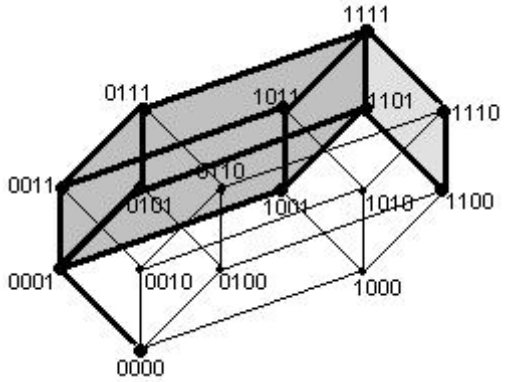
Пример 2.5. Найти сокращенную ДНФ геометрически f = (1101 0101 0101 1111).

Сначала выпишем значения функции в таблицу.

x_1	x_2	x_3	x_4	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1

1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

На единичном четырехмерном кубе отметим двоичные наборы, на которых значение функции равно единице (множество N_f).



Выпишем грани, содержащиеся в N_f и не содержащиеся в других гранях, составленных из вершин множества N_f.

N_1={(0001), (0011), (0111), (0101), (1001), (1011), (1111), (1101)} – трехмерная грань (четвертая координата каждого двоичного набора равна 1).

N_2={(1100), (1101), (1111), (1110)} – двумерная грань (первая и вторая координаты каждого двоичного набора равны 1).

N_3={(0000), (0001)} – одномерная грань (первая, вторая и третья координаты каждого двоичного набора равны 0).

Каждой из граней сопоставим элементарную конъюнкцию:

$K_1 = x_4$;

$K_2 = x_1 x_2$;

$K_3 = \bar{x}_1 \bar{x}_2 \bar{x}_3$.

Запишем дизъюнкцию полученных конъюнкций:

$f = x_4 \vee x_1 x_2 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3$.

Получение сокращенной ДНФ с помощью минимизирующей карты (карты Карно)

Карта Карно – плоскостная интерпретация четырехмерного куба. Считаем, что левый край склеен с правым, а верхний – с

нижним. Каждой ячейке соответствует определенный набор переменных и указывается, какое значение на этом наборе принимает функция. Причем соседние ячейки отличаются значением только одной переменной, что позволяет минимизировать ДНФ, используя закон склеивания, объединяя ячейки по 2n в прямоугольники, которые называются контурами. При создании контуров одну и ту же ячейку (конъюнкцию) можно включать в несколько контуров. Объединяя клетки, соответствующие единичным значениям функции, в максимальные контуры и сопоставляя им элементарные конъюнкции, получим сокращенную ДНФ.

Пример 2.6. Найти сокращенную ДНФ для функции из примера 2.5.

f = (1101 0101 0101 1111) с помощью минимизирующей карты (карты Карно).

x_1x_2 \ x_3x_4	00	01	11	10
00	1	1	1	0
01	0	1	1	0
11	1	1	1	1
10	0	1	1	0

Каждому из контуров сопоставим элементарную конъюнкцию:

$K_1 = x_4$;

$K_2 = x_1x_2$;

$K_3 = \bar{x}_1\bar{x}_2\bar{x}_3$.

Запишем дизъюнкцию полученных конъюнкций:

$f = x_4 \vee x_1x_2 \vee \bar{x}_1\bar{x}_2\bar{x}_3$.

Операция двоичного сложения. Многочлен Жегалкина

Операция двоичного сложения двух высказываний a и b – высказывание, истинное, когда истинностные значения a и b не совпадают и ложное – в противном случае.

Обозначение: $a \oplus b$; $a+b$ (читается: “либо a, либо b”, “или a, или b”).

Операция двоичного сложения определяется таблицей:

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	0

Свойства:

$a + b = b + a$; $(a + b)c = ac + bc$; $a + a = 0$; $a + 0 = a$;

$(a + b) + c = a + (b + c)$; $a + b = \overline{a \sim b} = \bar{a}b \vee a\bar{b} = (a \vee b)(\bar{a} \vee \bar{b})$; $a + \bar{a} = 1$; $a + 1 = \bar{a}$.

Многочлен Жегалкина для функции, содержащей две переменные:

$$P = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 xy .$$

Многочлен Жегалкина для функции, содержащей три переменные:

$$P = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z + \beta_4 xy + \beta_5 xz + \beta_6 yz + \beta_7 xyz .$$

Методика представления булевой функции в виде многочлена Жегалкина

1 способ. Метод неопределенных коэффициентов.

1. По таблице истинности составить систему уравнений (вместо переменных в многочлен подставить их соответствующие значения, в левой части уравнения – соответствующее этому набору значение функции).

2. Пользуясь таблицами истинности для двоичного сложения и конъюнкции, вычислить коэффициенты β_i.

3. Подставить в многочлен значения коэффициентов.

Пример 2.7. Методом неопределенных коэффициентов построить многочлен Жегалкина для функции $f(x, y) = x \vee y$.

$$P = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 xy .$$

Выписываем систему уравнений для коэффициентов β_0, β_1, β_2, β_3:

$$\begin{cases} f(0,0) = 0 = \beta_0 + \beta_1 \cdot 0 + \beta_2 \cdot 0 + \beta_3 \cdot 0 \cdot 0; \\ f(0,1) = 1 = \beta_0 + \beta_1 \cdot 0 + \beta_2 \cdot 1 + \beta_3 \cdot 0 \cdot 1; \\ f(1,0) = 1 = \beta_0 + \beta_1 \cdot 1 + \beta_2 \cdot 0 + \beta_3 \cdot 1 \cdot 0; \\ f(1,1) = 1 = \beta_0 + \beta_1 \cdot 1 + \beta_2 \cdot 1 + \beta_3 \cdot 1 \cdot 1; \end{cases}$$

или

$$\begin{cases} \beta_0 = 0; \\ \beta_0 + \beta_2 = 1; \\ \beta_0 + \beta_1 = 1; \\ \beta_0 + \beta_1 + \beta_2 + \beta_3 = 1. \end{cases}$$

Получаем $\beta_0 = 0$, $\beta_1 = \beta_2 = \beta_3 = 1$. Следовательно, $x \vee y = x + y + xy$.

2 способ. Преобразование формул.

1. Построить некоторую формулу Ф над множеством связок $\{\wedge, \neg\}$, реализующую заданную функцию f.
2. Заменить подформулы вида $\overline{A}$ на A+1.
3. Раскрыть скобки, пользуясь дистрибутивным законом $(a + b)c = ac + bc$.
4. Применить эквивалентности $a \cdot a = a$, $a \cdot 1 = a$, $a + a = 0$, $a + 0 = a$.

Пример 2.8. Построить многочлен Жегалкина для функции $f(x, y) = x \vee y$.

$x \vee y = \overline{\overline{x} \cdot \overline{y}} = (x + 1)(y + 1) + 1 = xy + x + y + 1 + 1 = xy + x + y$.

Важнейшие замкнутые классы. Теорема Поста

1. Класс функций, сохраняющих 0.

$T_0 = \{ f \mid f(0,...,0) = 0 \}$ – класс функций, обращающихся на нулевом векторе в 0.

2. Класс функций, сохраняющих 1.

$T_1 = \{ f \mid f(1, \ldots, 1) = 1 \}$ – класс функций, обращающихся на единичном векторе в 1.

3. Класс самодвойственных функций.

У самодвойственной функции на противоположных наборах значения противоположны.

4. Класс монотонных функций.

$M = \{ f \mid \alpha \le \beta \Rightarrow f(\alpha) \le f(\beta) \}$, где $\alpha = (a_1,...,a_n), \beta = (b_1,...,b_n)$, $a_i, b_i \in E_2$.

$\alpha \le \beta$ означает, что для любого i $a_i \le b_i$.

5. Класс линейных функций.

Функция называется линейной, если ее многочлен Жегалкина не содержит ни одной конъюнкции переменных. $L = \{ f \mid f = c_0 + c_1 x_1 + ... + c_n x_n \}$, где $c_i \in E_2$.

Если в векторе значений функции число 0 и 1 различно, то функция не является линейной, а если число нулей совпадает с числом единиц, то функция может быть линейной, а может быть и нелинейной. В этом случае для проверки линейности функции нужно построить многочлен Жегалкина.

Пример 2.9. Проверить булеву функцию $f = (x \to y) \to yz$ на принадлежность к классам T_0, T_1, S, L, M.

Построим таблицу истинности:

x	y	z	$x \to y$	yz	f
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	1	1	1

На наборе (000) функция принимает значение 0. Значит, $f \in T_0$.

На наборе (111) функция принимает значение 1. Значит, $f \in T_1$.

На противоположных наборах (011) и (100) функция принимает одинаковые значения. Значит, $f \notin S$.

$f \notin M$, так как (1, 0, 0) ≤ (1, 1, 0), но $f(1, 0, 0) > f(1, 1, 0)$.

В столбце значений функции f одинаково число 0 и 1, поэтому для проверки линейности функции нужно построить многочлен Жегалкина.

$$P = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z + \beta_4 xy + \beta_5 xz + \beta_6 yz + \beta_7 xyz .$$

$$\begin{cases} \beta_0 = 0; \\ \beta_0 + \beta_3 = 0; \\ \beta_0 + \beta_2 = 0; \\ \beta_0 + \beta_2 + \beta_3 + \beta_6 = 1; \\ \beta_0 + \beta_1 = 1; \\ \beta_0 + \beta_1 + \beta_3 + \beta_5 = 1; \\ \beta_0 + \beta_1 + \beta_2 + \beta_4 = 0; \\ \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 + \beta_7 = 1. \end{cases}$$

$$\beta_0 = 0,\ \beta_3 = 0,\ \beta_2 = 0,\ \beta_6 = 1, \beta_1 = 1, \beta_5 = 0, \beta_4 = 1, \beta_7 = 0 .$$

$$P = x + xy + yz .$$

Многочлен Жегалкина содержит конъюнкцию переменных – *yz*. Поэтому $f \notin L$.

Система функций $\{f_1, f_2, \ldots\}$ называется **полной**, если любая булева функция может быть записана в виде формулы через функции этой системы

Теорема Поста

Система булевых функций полна тогда и только тогда, когда для каждого из замкнутых классов T_0, T_1, S M и L в этой системе найдется хотя бы одна функция, которая этому классу не принадлежит, т. е. система содержит хотя бы одну функцию, не сохраняющую нуль, хотя бы одну функцию, не сохраняющую единицу, хотя бы одну несамодвойственную функцию, хотя бы одну немонотонную функцию и хотя бы одну нелинейную функцию.

Пример 2.10. Проверить множество булевых функций {0, 1, $xy \vee xz \vee yz$} на полноту.

Составим таблицу:

	T_0	T_1	S	M	L
0	+	-	-	+	+
1	-	+	-	+	+
$xy \vee xz \vee yz$	+	+	+	+	-

x	y	z	xy	xz	yz	$xy \vee xz \vee yz$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

$$P = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z + \beta_4 xy + \beta_5 xz + \beta_6 yz + \beta_7 xyz .$$

$$\begin{cases} \beta_0 = 0; \\ \beta_0 + \beta_3 = 0; \\ \beta_0 + \beta_2 = 0; \\ \beta_0 + \beta_2 + \beta_3 + \beta_6 = 1; \\ \beta_0 + \beta_1 = 0; \\ \beta_0 + \beta_1 + \beta_3 + \beta_5 = 1; \\ \beta_0 + \beta_1 + \beta_2 + \beta_4 = 1; \\ \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 + \beta_7 = 1. \end{cases}$$

$\beta_0 = 0$, $\beta_3 = 0$, $\beta_2 = 0$, $\beta_6 = 1, \beta_1 = 0, \beta_5 = 1, \beta_4 = 1, \beta_7 = 0$.

$P = xy + xz + yz$.

В столбце M нет ни одного минуса. Все функции множества {0, 1, $xy \vee xz \vee yz$} являются монотонными. Поэтому система неполная.

Упражнения

1. Представить булеву функцию в виде совершенной ДНФ:

а) $(x \sim y)\,(y \sim z)\,(z \sim x)$;
б) $\overline{(x \to y)}(x \sim \overline{yz})$;
в) $\overline{(x \to (y \to z))} \sim (x \to y)$;
г) $(x \vee y)(y \vee z) \to (x \vee z)$;
д) $(x \vee y \vee z)(x \to y)$;
е) $\overline{xy} \vee (x \to y) \sim z$;
ж) $(x \to (y \to z)) \sim \overline{yz}$;
з) $(x \to y\ \bar{z}) \to (x \sim y)$.

2. Представить булеву функцию в виде совершенной КНФ:
а) $(x \to z) \to (x \vee \bar{y})$;
б) $(x \vee y) \to (x \to z)$;
в) $x \vee y \vee z \to (x \vee y)z$;
г) $(x \vee y)(y \to z)(z \sim x)$;
д) $((x \to y) \sim (y \to \bar{x}))z$;
е) $(x \to y) \sim (\bar{x} \to (\bar{y} \vee z))$;
ж) $\overline{(x \sim y)(\bar{z} \vee y)}$;
з) $\overline{x \vee y} \vee z \sim xz$.

3. Найти сокращенную ДНФ методом Блейка для функции:
а) $f = x_1\overline{x_4} \vee \overline{x_1}x_2\overline{x_3} \vee x_2\overline{x_3}x_4$;
б) $f = x_1\overline{x_2} \vee \overline{x_1}\,\overline{x_3}\,\overline{x_4} \vee x_2\overline{x_3}\,\overline{x_4}$;
в) $f = x_3x_4 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3} \vee \overline{x_1}\,\overline{x_2}\,\overline{x_4}$;
г) $f = \overline{x_1}\,\overline{x_3} \vee x_1x_2x_4 \vee x_2x_3x_4$;
д) $f = x_2\overline{x_4} \vee \overline{x_1}\,\overline{x_2}x_3 \vee \overline{x_1}\,x_3x_4$;
е) $f = \overline{x_2}\,\overline{x_4} \vee x_1x_2x_3 \vee x_1x_3x_4$;
ж) $f = \overline{x_2}x_3 \vee \overline{x_1}x_2x_4 \vee \overline{x_1}\,\overline{x_3}x_4$;
з) $f = \overline{x_1}\,x_2 \vee \overline{x_2}\,\overline{x_3}x_4 \vee x_1\overline{x_3}x_4$.

4. Найти сокращенную ДНФ из КНФ:
а) $f = (x_1 \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee \overline{x_4})(x_2 \vee x_3 \vee \overline{x_4})$;
б) $f = (\overline{x_1} \vee x_2 \vee x_4)(\overline{x_1} \vee x_3 \vee \overline{x_4})(x_2 \vee \overline{x_3})$;
в) $f = (\overline{x_1} \vee x_2 \vee \overline{x_3})(\overline{x_3} \vee x_4)(x_1 \vee x_2 \vee \overline{x_4})$;
г) $f = (\overline{x_1} \vee \overline{x_2} \vee x_4)(x_3 \vee \overline{x_4})(x_1 \vee \overline{x_2} \vee x_3)$;
д) $f = (x_1 \vee \overline{x_2} \vee x_4)(x_1 \vee x_3)(x_2 \vee \overline{x_3} \vee x_4)$;
е) $f = (\overline{x_1} \vee x_2 \vee \overline{x_3})(\overline{x_2} \vee x_4)(x_1 \vee \overline{x_3} \vee x_4)$;
ж) $f = (\overline{x_1} \vee \overline{x_4})(\overline{x_1} \vee x_2 \vee \overline{x_3})(\overline{x_2} \vee \overline{x_3} \vee x_4)$;
з) $f = (\overline{x_2} \vee x_3 \vee \overline{x_4})(\overline{x_1} \vee \overline{x_4})(x_1 \vee x_2 \vee x_3)$.

5. Найти сокращенную ДНФ геометрически:
а) f = (0001 1111 1100 1111);
б) f = (0111 0011 1011 1011);
в) f = (1110 1011 1010 1011);
г) f = (1111 1111 0100 0110);
д) f = (1100 1101 1110 1101);
е) f = (0111 0111 1101 0101);
ж) f = (1000 1111 0011 1111);
з) f = (1111 1000 1111 1001).

6. Найти сокращенную ДНФ с помощью минимизирующей карты (карты Карно):

а) f = (1100 1100 0100 1100);
б) f = (1111 1100 0100 1100);
в) f = (1111 1011 0000 0000);
г) f = (1011 1011 1000 1000);
д) f = (1000 1101 1101 1000);
е) f = (1010 1010 0010 1010);
ж) f = (1111 0000 0000 1011);
з) f = (1000 1000 1011 1011);
и) f = (1001 1011 1001 1001);
к) f = (1101 1101 1000 1000).

7. Представить булеву функцию в виде многочлена Жегалкина, используя 2 способа:

а) $\overline{x \to y} \vee (\bar{x} \to \bar{y})$;
б) $x\bar{y} \to (\bar{y} \to \bar{x})$;
в) $x \vee (x \sim y)$;
г) $xy \to (\bar{x} \vee \bar{y})$;
д) $x \vee y \to xy$;
е) $\bar{x}y \to x\bar{y}$;
ж) $\bar{x} \to (y \to x)$;
з) $\bar{x}y \vee \overline{x \to y}$.

8. Проверить булеву функцию на принадлежность к классам T_0, T_1, S, L, M:

а) $((x \vee y)\bar{x} \to y)\bar{z}$;
б) $\overline{xz \to x} \vee y$;
в) $y(\overline{\bar{x}z} \to x)$;
г) $\bar{x}(y \sim z) \vee x(y + z)$;
д) $(x \vee (y + z))(\bar{x} \vee (y \sim z))$;
е) $y \to \bar{z} \vee \overline{x(y + z)}$;
ж) $x\,(y \sim z) \vee (y \to z)$;
з) $\overline{y(x \to (x \vee y)) + z}$.

9. Проверить множество булевых функций на полноту:

а) $\{x \cdot \bar{y}, x \sim yz\}$;
б) $\{x \to y, x \to \bar{y}z\}$;
в) $\{x \sim y, x + y + z, x \vee y\}$;
г) $\{0, x + y + z + 1, x\,(y + z\}$;
д) $\{xy, x + y + z, \bar{x}\}$;
е) $\{1, xy + z\}$;
ж) $\{x + y + z, xy, x \vee y \vee z\}$;
з) $\{x \sim (y \vee z), y \to x \cdot z\}$.

Раздел 3. **Основы теории множеств**

Множество – одно из основных понятий современной математики, с которым каждый человек знаком со школьной скамьи. В этом разделе мы введем понятие множества и опишем различные способы комбинирования разных множеств для получения новых. Здесь рассматриваются исключительно конечные множества, а тонкие и сложные вопросы, связанные с рассмотрением бесконечных множеств, сознательно опущены.

Множество – это любая определенная совокупность объектов.

U – универсальное множество.

$\varnothing$ **–** пустое множество.

Операции над множествами

Объединением множеств A и B (A $\cup$ B) называется множество, состоящее из всех тех элементов, которые принадлежат хотя бы одному из множеств A, B.

Пересечением множеств A и B (A $\cap$ B) называется множество, состоящее из всех элементов, которые принадлежат и A и B.

Разностью множеств A и B (A \ B) называется множество всех тех элементов A, которые не содержатся в B.

Дополнением (до U**)** множества A называется множество всех элементов, не принадлежащих A (но принадлежащих U): $\overline{A} = U \setminus A$.

Пример 3.1. $A=\{a, b, \underline{c}, \underline{d}\}$, $B=\{\underline{c}, \underline{d}, e, f, g, h\}$

$A \cup B = \{a,b,c,d,e,f,g,h\}$, $A \cap B = \{c,d\}$, $A \setminus B = \{a,b\}$, $B \setminus A = \{e,f,g,h\}$.

Свойства операций над множествами

1. Ассоциативность:

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

2. Коммутативность:

$A \cup B = B \cup A$; $\quad A \cap B = B \cap A$.

3. Дистрибутивность:

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

4. Идемпотентность:

$A \cup A = A$; $\quad A \cap A = A$.

5. Закон двойного дополнения:

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

6. Законы ∅:

$A \cap \varnothing = \varnothing$; $A \cup \varnothing = A$.

7. Законы U:

$A \cup U = U$; $A \cap U = A$.

8. Законы де Моргана:

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

9. Свойства дополнения:

$A \cup \overline{A} = U$; $A \cap \overline{A} = \varnothing$.

10. Поглощение:

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

11. Выражение для разности:

$A \setminus B = A \cap \overline{B}$.

Пример 3.2. Доказать, что $(A \setminus C) \setminus (B \setminus C) = (A \setminus B) \setminus C$.

$(A \setminus C) \setminus (B \setminus C) = (A \cap \overline{C}) \setminus (B \cap \overline{C}) =$

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

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

$= ((A \setminus B) \cap \overline{C}) \cup \varnothing = (A \setminus B) \setminus C$.

Формула количества элементов в объединении трех конечных множеств

Пусть A, B, C – конечные множества, тогда

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

Пример 3.3.

В классе учится 36 человек. 18 человек посещают математический кружок, 14 человек – физический, 10 человек – химический, 8 человек – математический и физический, 5 человек – математический и химический, 3 человека – физический и химический, 2 человека – все три кружка. Сколько человек не посещает ни одного кружка?

1. $|M \cap \Phi \cup X| = |M| + |\Phi| + |X| - |M \cap \Phi| - |M \cap X| - |M \cap \Phi| + |M \cap \Phi \cap X| =$

$= 18+14+10-8-5-3+2=28$ (человек) – посещают кружки.

2. 36-28=8 (человек) – не посещают кружки.

Упражнения

1. Выполнить над множествами A и B операции: $A \cup B$, $A \cap B$, $A \setminus B$:

а) $A = \{a, b, c, d\}$, $B = \{1, 2, b, d\}$;
б) $A = \{1, 2, 3, x, y\}$, $B = \{x, y, z, t\}$;
в) $A = \{1, 2, 3, 4, 5\}$, $B = \{1, 3, 5, 7, 9\}$;
г) $A = \{a, 1, b, 2, c\}$, $B = \{1, 2, 3, 4\}$;
д) $A = \{a, b, c, y, z\}$, $B = \{a, b, c, d\}$;
е) $A = \{1, 2, 4, 8, 16\}$, $B = \{1, 2, 3, 4\}$;
ж) $A = \{a, b, 1, 2, 3\}$, $B = \{1, 2, c, d\}$;
з) $A = \{x, y, z, t\}$, $B = \{x, y, 1, 2\}$.

2. Доказать равенства:

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

3. Решить задачи:

3.1. В отделе научно-исследовательского института работают несколько человек, причем каждый из них знает хотя бы один иностранный язык. Шестеро знают английский, шестеро – немецкий, семеро – французский. Четверо знают английский и немецкий, трое – немецкий и французский, двое – французский и английский. Один человек знает все три языка. Сколько человек работает в отделе?

3.2. В ожесточенном бою 70 из 100 пиратов потеряли глаз, 80 – руку, 85 – ногу, 40 – глаз и руку, 50 – глаз и ногу, 55 – руку и ногу. Сколько пиратов потеряли глаз, руку и ногу?

3.3. Исследователь рынка сообщает следующие данные. Из 1000 опрошенных 811 нравится шоколад, 752 – мармелад, 418 – зефир, 570 – шоколад и мармелад, 356 – шоколад и зефир, 348 – мармелад и зефир, а 297 – все три вида сладостей. Показать, что в этой информации есть ошибки.

3.4. На загородную прогулку поехали 92 человека. Бутерброды с колбасой взяли 48 человек, с сыром – 38 человек, с ветчиной – 42 человека, с сыром и колбасой – 28 человек, с колбасой и ветчиной – 31 человек, с сыром и ветчиной – 26 человек. 25 человек взяли с собой все три вида бутербродов, а несколько человек вместо бутербродов взяли пирожки. Сколько человек взяли с собой пирожки?

3.5. На вступительном экзамене по математике были предложены три задачи: по алгебре, планиметрии и стереометрии. Из 1000

абитуриентов задачу по алгебре решили 800, по планиметрии – 700, а по стереометрии – 600 абитуриентов. При этом задачи по алгебре и планиметрии решили 600 абитуриентов, по алгебре и стереометрии – 500, по планиметрии и стереометрии – 400. Все три задачи решили 300 абитуриентов. Существуют ли абитуриенты, не решившие ни одной задачи, и если да, то сколько их?

Раздел 4. **Предикаты. Бинарные отношения**

Для строгого математического описания любых связей между элементами двух множеств мы введем понятие бинарного отношения, а также обсудим некоторые свойства. Понятие предиката является обобщением понятия высказывание.

N-местным предикатом называется связное повествовательное предложение, содержащее n переменных и обладающее следующим свойством: при фиксации значений всех переменных о предложении можно сказать, истинно оно или ложно.

Пусть P(x) – некоторый предикат на множестве M.

Высказывание «для любого x истинно P(x)» обозначается $\forall x$ P(x). Знак $\forall x$ называется **квантором всеобщности**.

Высказывание «найдется такой x, что истинно P(x)» обозначается $\exists x$ P(x). Знак $\exists x$ называется **квантором существования**.

Построение отрицаний к предикатам, содержащим кванторные операции

Для построения отрицаний высказываний, содержащих квантор существования, нужно заменить его квантором всеобщности, а предложение, стоящее после квантора, заменить его отрицанием.

Для построения отрицаний высказываний, содержащих квантор всеобщности, нужно заменить его квантором существования, а предложение, стоящее после квантора, заменить его отрицанием.

Пример 4.1. Построить отрицание к предикату $\exists x\big((\forall y\ x^2 = 2y) \to (\exists y\ x + y = 6)\big)$.

$$\overline{\exists x\left(\left(\forall y\ x^2=2y\right)\to\left(\exists y\ x+y=6\right)\right)}=\forall x\overline{\left(\left(\forall y\ x^2=2y\right)\to\left(\exists y\ x+y=6\right)\right)}=$$
$$\forall x\overline{\left(\left(\forall y\ x^2=2y\right)\vee\left(\exists y\ x+y=6\right)\right)}=\forall x\left(\overline{\left(\forall y\ x^2=2y\right)}\wedge\overline{\left(\exists y\ x+y=6\right)}\right)=$$
$$\forall x\left(\left(\forall y\ x^2=2y\right)\wedge\overline{\left(\exists y\ x+y=6\right)}\right)=\forall x\left(\left(\forall y\ x^2=2y\right)\wedge\left(\forall y\ \overline{x+y=6}\right)\right)=$$
$$\forall x\left(\left(\forall y\ x^2=2y\right)\wedge\left(\forall y\ x+y\neq 6\right)\right).$$

Формализация предложений с помощью логики предикатов

1. Выделить простейшие высказывания; ввести подходящие предикаты на соответствующих множествах.

2. С помощью логических операций и кванторов построить формулы.

Пример 4.2. Формализировать предложения с помощью логики предикатов: Всякая функция, непрерывная на [0,1] сохраняет знак или принимает нулевое значение.

X – функция; Q(x) – x непрерывна на [0,1], P(x) – x сохраняет знак, R(x) – x принимает нулевое значение – $(\forall x)$ (Q(x))→P(x)∨R(x)).

Бинарные отношения

Декартовым произведением двух множеств A и B называется множество, обозначаемое $A\times B$, элементами которого являются упорядоченные пары (x, y), где $x\in X, y\in Y$.

Бинарным отношением R называется подмножество пар $(a;b)\in R$ декартово произведения $A\times B$, т. е. $R\subset A\times B$.

Часто рассматривают отношения R между парами элементов одного и того же множества A, тогда R $\subset A\times A$ (R – есть отношение на множестве A).

Если a и b находятся в отношении R, это часто записывается так $a\ R\ b$.

Свойства бинарных отношений

Пусть R – отношение на множестве M, $R\subset M\times M$.

1. R – рефлексивно, если имеет место aRa для любого $a\in M$.

2. R – симметрично, если aRb влечёт bRa.

3. R – транзитивно, если aRb и bRc влекут aRc.

Пример 4.3. Исследовать бинарное отношение на рефлексивность, симметричность и транзитивность:

$R = \{(m; n) \mid m < n + 1\}$.

1. $m < m + 1$ – рефлексивно;

2. $1 < 3 + 1$; но $3 > 1 + 1$ – не симметрично;

3. $3 < 2 + 1$ и $2 < 1 + 1 \Rightarrow 3 < 1 + 1$ – не транзитивно.

Отношение эквивалентности – это бинарное отношение, которое рефлексивно, симметрично и транзитивно.

Отношение эквивалентности R разбивает множество M, на котором оно задано, на непересекающиеся непустые подмножества так, что элементы одного и того же подмножества находятся в отношении R, а между элементами из разных подмножеств отношение R отсутствует.

В таком случае говорят, что отношение R задает **разбиение** на множестве M, или **систему классов эквивалентности** по отношению R.

Пример 4.4. Разбить множество $M = \{x, y, z, t\}$ на классы эквивалентности, если на множестве M задано отношение эквивалентности $R = \{(x, x), (y, y), (z, z), (t, t), (x, z), (z, x), (y, t), (t, y)\}$.

Отношение R разбивает множество M на два класса эквивалентности: $\{x, z\}$ и $\{y, t\}$.

Упражнения

1. Записать область истинности предиката:

а) $x^2 > 29, x \in N$;
б) $x^2 - 2x - 3 > 0, x \in N$;
в) $(x > 3) \vee (x < -1)(x \in R)$;
г) $(x > 2) \vee (x < 2)(x \in R)$;
д) $|x| > 5, x \in N$;
е) $(|x| > 2) \rightarrow (|x| < 3)(x \in R)$;
ж) $(x > 2) \rightarrow (x < 2)(x \in R)$;
з) $(|x| < 3) \wedge (x \geq 2)(x \in R)$.

2. Определить логическое значение следующих высказываний $(x, y \in R)$:

а) $\exists x \ln x < 0$; $\forall x \sqrt{x^2 + 2x + 1} = x + 1$; $\exists x \forall y\, y^2 > x$;
б) $\exists x \sqrt{x + 2} = 1 - x$, $\forall x\, x > 0$; $\exists x \forall y\, x^2 > \cos y$;
в) $\exists x\, x^2 + 5x + 6 = 0$, $\forall x \quad \frac{x^3 - 3x^2 + 3x - 1}{x - 1} = x^2 - 2x + 1$; $\exists x \forall y\, |y| > x$;
г) $\exists x\, |x - 1| = x + 1$; $\forall x\, x^2 + 2x + 3 > 0$; $\exists x \forall y\, x^2 + y^2 > 25$;

д) $\exists x\, x^2=25\ \forall x\ \frac{x^2-1}{x-1}=x+1\ \exists x \forall y\ (x+y)^2<0$;

е) $\exists x\, x^2+x+1=0\ \forall x\, x^2=25\ \exists x \forall y\, y^2-x^2\geq 0$;

ж) $\exists x\ 2^x\leq 0\ \forall x\, x^2+2x+1=(x+1)^2\ \exists x \forall y\ \sqrt{x}$ - $|y|\leq 0$;

з) $\exists x\ \sqrt{x}\leq 0\ \forall x\ \sqrt{x^2}=x\ \exists x \forall y\ |\sin y|\leq x$.

3. Построить отрицание к предикатам:

а) $\exists x\ ((\forall y\, x^2+y^2>4)\rightarrow(\forall y\, y^2>x))$;

б) $\forall x\ ((\exists y\, y>3x)\wedge(\forall z\, z\leq 5x))$;

в) $\exists x\ ((\exists y\, y=x^2+5)\wedge(\forall y\, y-x=2))$;

г) $\forall x\ ((\exists y\, y=x^3)\wedge(\exists y\, y=\sqrt{x}))$;

д) $\exists y\ ((\forall x\, x\geq 3)\wedge(\exists z\, z^2+y^2=1))$;

е) $\forall x\ ((\exists y\, x<y)\rightarrow\forall z\, z^2+x^2>1))$;

ж) $\exists x\ ((\forall y\, x+y=5)\wedge(\exists y\, x+y<0))$;

з) $\forall x\ ((\forall y\, y=x^2)\rightarrow(\exists y\, y=2x))$.

4. Формализировать предложения с помощью логики предикатов:

а) Все рыбы, кроме акул, добры к детям.

б) Некоторые остроумны только, когда пьяны.

в) Всякий, в ком есть упорство, может изучить логику.

г) Есть такие люди, которые спят днем, и нет человека, который не спит ночью.

д) Всякий человек любит кого-нибудь, и никто не любит всех.

е) У всех людей, которые вечером долго сидят за компьютером, утром болит голова или плохое настроение.

5. Исследовать бинарное отношение на рефлексивность, симметричность и транзитивность:

а) $R=\{(m,n)\mid m,n\in N\ \&\ m=n^2\}$;

б) $R=\{(x,y)\mid x,y\in R\ \&\ x\leq y\}$;

в) $R=\{(x,y)\mid x,y\in R\ \&\ x=y\}$;

г) $R=\{(m,n)\mid m,n\in N\ \&\ m=2n\}$;

д) $R=\{(x,y)\mid x,y\in R\ \&\ x>y\}$;

е) $R=\{(m,n)\mid m,n\in Z\ \&\ m=n+2\}$;

ж) $R=\{(m,n)\mid m,n\in Z\ \&\ |m|=|n|\}$;

з) $R=\{(m,n)\mid m,n\in N\ \&\ m^3=n^3\}$.

6. Разбить множество $M=\{x, y, z, t\}$ на классы эквивалентности, если на множестве M задано отношение эквивалентности R:

а) $R=\{(x,z),(x,x),(z,z),(z,x),(y,y),(t,t)\}$;

б) $R=\{(x,z),(z,x),(x,y),(y,x),(y,z),(z,y),(t,t)\ (x,x)\ (y,y),(z,z)\}$;

в) $R=\{(y,z),(z,y),(x,x),(y,y),(z,z),(t,t)\}$;

г) $R=\{(x,y),(y,x),(x,x),(y,y),(z,z),(t,t)\}$;

д) R = {(*x, t*), (*t, x*), (*x, x*), (*y, y*), (*z, z*), (*t, t*)};
е) R = {(*x, z*), (*z, x*), (*y, t*), (*t, y*), (*x, x*), (*y, y*), (*z, z*), (*t, t*)};
ж) R = {(*x, y*), (*y, x*), (*z, t*), (*t, z*), (*x, x*), (*y, y*), (*z, z*), (*t, t*)};
з) R = {(*y, z*), (*z, t*), (*t, y*), (*y, t*), (*t, z*), (*z y*), (*x, x*), (*y, y*), (*z, z*), (*t, t*)}.

Раздел 5. **Алгебра подстановок**

Понятие «функции» является одним из основополагающих в математике. Функции играют центральную роль в математике, где они используются для описания любых процессов, при которых элементы одного множества каким-то образом переходят в элементы другого. Подстановка является функцией, заданной на конечном множестве.

Пусть X и Y – множества, а f – отображение X на Y, т. е. правило, сопоставляющее каждому элементу x вполне определенный элемент $f(x) \in Y$.

Взаимно однозначное отображение f: $X \to X$ называется **подстановкой** на X.

Замечание. Если множество X конечно ($|X| = n$), то можно считать, что $X = 1..n$. В этом случае подстановку f: $1..n \to 1..n$ удобно задавать таблицей из двух строк. В первой строке – значения аргументов, во второй – соответствующие значения функции.

Пример 5.1.

$$f = \begin{vmatrix} 12345 \\ 52143 \end{vmatrix}; \quad g = \begin{vmatrix} 12345 \\ 41235 \end{vmatrix}.$$

Произведением подстановок f и g называется их композиция $f \circ g$.

Пример 5.2. Перемножим подстановки из примера 5.1.

$$f = \begin{vmatrix} 12345 \\ 52143 \end{vmatrix}$$

$$g = \begin{vmatrix} 52143 \\ 51432 \end{vmatrix}$$

$$fg = \begin{vmatrix} 12345 \\ 51432 \end{vmatrix}$$

Тождественная подстановка – это подстановка e, такая что $e(x) = x$.

Пример 5.3.

$$e = \begin{vmatrix} 12345 \\ 12345 \end{vmatrix}.$$

Свойство. Подстановка не изменится, если ее умножить на тождественную: $fe = ef = f$.

Обратная подстановка – это обратная функция.

Пример 5.4. $f = \begin{vmatrix} 12345 \\ 52143 \end{vmatrix}$; $f^{-1} = \begin{vmatrix} 52143 \\ 12345 \end{vmatrix} = \begin{vmatrix} 12345 \\ 32541 \end{vmatrix}$.

Свойство. $f \cdot f^{-1} = f^{-1} \cdot f = e$.

Замечание. Таблицу обратной подстановки можно получить, если поменять местами строки исходной таблицы.

Графическое представление подстановки

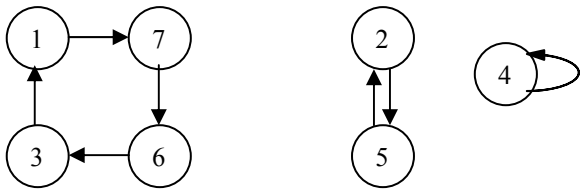
Подстановки удобно представлять в графической форме, проводя стрелки от каждого элемента x к элементу $f(x)$.

Цикл – это последовательность элементов x_0, …, x_k, такая что

$$f(x_i) = \begin{cases} x_{i+1}, & 0 \le i < k, \\ x_0, & i = k. \end{cases}.$$

Разложением подстановки на циклы будем называть представление подстановки в виде композиции циклов.

$$f = \begin{vmatrix} 1234567 \\ 7514236 \end{vmatrix}$$



$f = [1763][25][4]$

Решение простейших уравнений в алгебре подстановок

Пример 5.5.

$$a = \begin{vmatrix} 12345 \\ 52143 \end{vmatrix}, \; b = \begin{vmatrix} 12345 \\ 41235 \end{vmatrix}, \; c = \begin{vmatrix} 12345 \\ 34215 \end{vmatrix}.$$

1) $ax = b$

$$x = a^{-1} b = \begin{vmatrix} 52143 \\ 12345 \end{vmatrix} \cdot \begin{vmatrix} 12345 \\ 41235 \end{vmatrix} = \begin{vmatrix} 52143 \\ 41235 \end{vmatrix} = \begin{vmatrix} 12345 \\ 21534 \end{vmatrix}.$$

2) $xa = b$

$$x = ba^{-1} = \begin{vmatrix} 12345 \\ 41235 \end{vmatrix} \cdot \begin{vmatrix} 52143 \\ 12345 \end{vmatrix} = \begin{vmatrix} 12345 \\ 41235 \end{vmatrix} \cdot \begin{vmatrix} 41235 \\ 43251 \end{vmatrix} = \begin{vmatrix} 12345 \\ 43251 \end{vmatrix}.$$

3) $axb = c$

$$x = a^{-1}cb^{-1} = \begin{vmatrix} 52143 \\ 12345 \end{vmatrix} \cdot \begin{vmatrix} 12345 \\ 34215 \end{vmatrix} \cdot \begin{vmatrix} 41235 \\ 12345 \end{vmatrix} = \begin{vmatrix} 52143 \\ 34215 \end{vmatrix} \cdot \begin{vmatrix} 41235 \\ 12345 \end{vmatrix} = \begin{vmatrix} 12345 \\ 24513 \end{vmatrix} \cdot \begin{vmatrix} 24513 \\ 31524 \end{vmatrix} = \begin{vmatrix} 12345 \\ 31524 \end{vmatrix}.$$

Четные и нечетные подстановки

Перестановками называются комбинации, состоящие из одних и тех же различных элементов и отличающиеся только порядком их расположения.

Два элемента образуют **инверсию**, если $a_i > a_j$ при $i < j$.

Перестановка называется **четной (нечетной)**, если суммарное количество всех инверсий I, образуемых элементами этой перестановки, четно (нечетно).

Пример 5.6.

123 – четная перестановка I=0;

312 – четная перестановка I=2 (3>1,3>2);

321 – нечетная перестановка I=3 (3>2, 3>1, 2>1).

Подстановка называется **четной** (**нечетной**), если суммарное число инверсии верхней и нижней строк четно (нечетно).

Пример 5.7. Определить четность подстановки $A = \begin{vmatrix} 52413 \\ 32514 \end{vmatrix}$.

I_1=4+1+2=7;

I_2=2+1+2=5;

I_1+I_2=7+5=12.

Значит, подстановка является четной.

Упражнения

1. Записать циклическое разложение подстановки и представить её в графической форме:

а) $A=\begin{vmatrix}1234567\\3741526\end{vmatrix}$;

б) $A=\begin{vmatrix}1234567\\3417256\end{vmatrix}$;

в) $A=\begin{vmatrix}1234567\\2563417\end{vmatrix}$;

г) $A=\begin{vmatrix}1234567\\5627134\end{vmatrix}$;

д) $A=\begin{vmatrix}1234567\\4157362\end{vmatrix}$;

е) $A=\begin{vmatrix}1234567\\6427513\end{vmatrix}$;

ж) $A=\begin{vmatrix}1234567\\5716342\end{vmatrix}$;

з) $A=\begin{vmatrix}1234567\\1756432\end{vmatrix}$.

2. Решить уравнения: 1) $A\cdot X=B$; 2) $X\cdot A=B$; 3)$A\cdot X\cdot B=E$ и найти $A\cdot B$; B^{-1}, если

а) $A=\begin{vmatrix}12345\\14523\end{vmatrix}$; $B=\begin{vmatrix}12345\\23514\end{vmatrix}$;

б) $A=\begin{vmatrix}12345\\41253\end{vmatrix}$; $B=\begin{vmatrix}12345\\34152\end{vmatrix}$;

в) $A=\begin{vmatrix}12345\\51324\end{vmatrix}$; $B=\begin{vmatrix}12345\\25134\end{vmatrix}$;

г) $A=\begin{vmatrix}12345\\35214\end{vmatrix}$; $B=\begin{vmatrix}12345\\52431\end{vmatrix}$;

д) $A=\begin{vmatrix}12345\\31524\end{vmatrix}$; $B=\begin{vmatrix}12345\\23451\end{vmatrix}$;

е) $A=\begin{vmatrix}12345\\43125\end{vmatrix}$; $B=\begin{vmatrix}12345\\13521\end{vmatrix}$;

ж) $A=\begin{vmatrix}12345\\25143\end{vmatrix}$; $B=\begin{vmatrix}12345\\24351\end{vmatrix}$;

з) $A=\begin{vmatrix}12345\\23514\end{vmatrix}$; $B=\begin{vmatrix}12345\\35241\end{vmatrix}$.

3. Определить четность подстановки:

а) $A=\begin{vmatrix}45213\\31524\end{vmatrix}$;

б) $A=\begin{vmatrix}25143\\53421\end{vmatrix}$;

в) $A=\begin{vmatrix}35241\\42513\end{vmatrix}$;

г) $A=\begin{vmatrix}52431\\42315\end{vmatrix}$;

д) $A=\begin{vmatrix}53241\\42513\end{vmatrix}$;

е) $A=\begin{vmatrix}41352\\35421\end{vmatrix}$;

ж) $A=\begin{vmatrix}34512\\21453\end{vmatrix}$;

з) $A=\begin{vmatrix}23145\\15324\end{vmatrix}$.

Раздел 6. **Основы алгебры вычетов и их приложение к простейшим криптографическим шифрам**

Криптография, область знаний о шифрах, методах их создания и раскрытия, известна с глубокой древности и использует самые разнообразные шифры. Для описания методов шифрования необходимы некоторые факты из теории чисел, основы алгебры вычетов. В данной теме все операции выполняются над целыми числами.

Число a **сравнимо по модулю** m с числом b, если a и b при делении на m дают один и тот же остаток, где m – натуральное число.

Обозначение: $a \equiv b \pmod{m}$.

Отношение сравнимости рефлексивно, симметрично и транзитивно, следовательно, является отношением эквивалентности.

Вычетами (по модулю m) называются классы эквивалентности по отношению сравнимости (по модулю m).

Обозначение: Z_m – множество вычетов по модулю m.

Обычно из каждого класса выбирают одного представителя:

$Z_m = \{0, 1, 2, \ldots, m\text{-}1\}$ – система наименьших неотрицательных вычетов (все возможные остатки при делении на m).

Вычет a является **обратимым вычетом** по модулю m, если для a найдется такой вычет a^{-1}, что $a *_m a^{-1}=1$.

a^{-1} – **обратный вычет**.

Пример 6.1. Рассмотрим операцию умножения в Z_4

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

$1^{-1}=1$, так как $1*_4 1=1$

$3^{-1}=3$, так как $3*_4 3=1$

Класс 2 не имеет обратного, то есть не обратим.

Пример 6.2. Выполнить операции в алгебре вычетов в $\boldsymbol{Z}_4$:
$1:3+2\cdot 3-2\cdot 3^2=3+2-2\cdot 1=3$.

Критерий обратимости вычета

Для того чтобы вычет $a \in Z_m$ был обратим, необходимо и достаточно, чтобы он был взаимно прост с модулем.

Замечание: числа a и b называются **взаимно простыми**, если их наибольший общий делитель равен 1.

Обозначение: Z_m^* – система обратимых вычетов по модулю m.

Функция $\varphi(m) = \left| Z_m^* \right|$ называется **функцией Эйлера**.

Если p – простое число, то $\varphi(p)=p-1$.

$\varphi(m)=m\left(1-\frac{1}{p_1}\right)...\left(1-\frac{1}{p_k}\right)$, где $p_1, ..., p_k$ – все простые делители m.

Пример 6.3.

Пусть $m = 12$, тогда $\varphi(12) = 12\cdot\left(1-\frac{1}{2}\right)\cdot\left(1-\frac{1}{3}\right)=12\cdot\frac{1}{2}\cdot\frac{2}{3}=4$.

$Z_{12}^*=\{1, 5, 7, 11\}$.

Пусть $m = 15$, тогда $\varphi(15)=15\cdot\left(1-\frac{1}{3}\right)\cdot\left(1-\frac{1}{5}\right)=15\cdot\frac{2}{3}\cdot\frac{4}{5}=8$.

Z_{15}^* $\{1, 2, 3, 7, 8, 11, 13, 14\}$.

Пусть $m = 7$, тогда $\varphi(7) = 7 - 1 = 6$; $Z_7^* =\{1,2,3,4,5,6\}$.

Шифрование

Шифр Виженера с постоянным ключом

Пример 6.4. Расшифровать текст, зашифрованный шифром Виженера с постоянным ключом (Key): wmlew

1 способ.

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
K	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
E	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*
Y	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*

Алгоритм дешифрования

1. Берем первую букву в зашифрованном тексте.

2. Находим ее в строке, которая помечена первой буквой ключа.

3. Заменяем на букву, которая находится в том же столбце и верхней строке.

4. Берем вторую букву, находим ее в строке, которая помечена второй буквой ключа, и заменяем ее на букву в том же столбце и верхней строке. Ключ используется циклически.

wmlew – minus

2 способ.

C – зашифрованный текст.

T – открытый текст.

Ключ: $K=K_1...K_n$, n – длина ключа.

Дешифрирование: $T_i = C_i - K_{i \bmod n}$ (mod 26).

K E Y
10 4 24
$C=\overset{W}{22}\ \overset{M}{12}\ \overset{L}{11}\ \overset{E}{4}\ \overset{W}{22}$
$T_1=22-10=12$ M
$T_2=12-4=8$ I
$T_3=11-24=-13=13$ N
$T_4=4-10=-6=20$ U
$T_5=22-4=18$ S

Шифр Виженера с бегущим ключом

В качестве ключа этой криптосистемы используется либо сам открытый текст, либо зашифрованный текст и дополнительные односимвольные затравочные ключи(t_0,c_0).

1. Затравочный ключ: t_0.

Дешифрование: $t_i=c_i-t_{i-1}$ (mod m).

Пример 6.5.

Пусть $t_0=22$

$C=\overset{i}{8}\ \overset{u}{20}\ \overset{v}{21}\ \overset{h}{7}\ \overset{m}{12}$
$t_1=8-22=-14=12$ m
$t_2=20-12=8$ i
$t_3=21-8=13$ n
$t_4=7-13=-6=20$ u
$t_5=12-20=-8=18$ s

2. Затравочный ключ: C_0.

Дешифрование: $t_i=C_i-C_{i-1} \pmod m$.

Пример 6.6.

Пусть $C_0=10$

$C=\overset{w}{22}\ \overset{e}{4}\ \overset{r}{17}\ \overset{l}{11}\ \overset{d}{3}$

$t_1=22-10-12$ m

$t_2=4-22=-18=8$ i

$t_3=17-4=13$ n

$t_4=11-17=-6=20$ u

$t_5=3-11=-8=18$ s

Упражнения

1. Выполнить операции в алгебре вычетов в $\boldsymbol{Z}_5$:

а) $3^4-4:3-2\cdot3^2+2:4$;

б) $2^4-3:4+3^2\cdot4^3-1:3$;

в) $4^7\cdot2^3+3:2-2\cdot3^3+3^5$;

г) $3^3\cdot2^5+2:3-3\cdot4^2+4^5$.

Выполнить операции в алгебре вычетов в $\boldsymbol{Z}_6$:

д) $3+5^4-4^5+2:5$;

е) $2^5\cdot5^3-4:5+3^{10}-5\cdot4^2$;

ж) $3^8\cdot5^5-1:5+3:5+4^8\cdot3^9$;

з) $3:5-3^2\cdot5^5+2^4\cdot4^2-5\cdot4\cdot3$.

2. Написать систему обратимых вычетов по заданному модулю:

а) 40; в) 44; д) 50; ж) 52;

б) 42; г) 48; е) 51; з) 54.

3. Расшифровать текст, зашифрованный шифром Виженера с постоянным ключом (Key):

а) ayccxgyr; г) mslfipqi; ж) citovyv;

б) zsqcmzvc; д) wiycypo; з) srdsrgdi.

в) cioeilmi; е) zvmnyad;

4. Расшифровать текст, зашифрованный шифром Виженера с бегущим ключом:

а) ewgvbwb ($t_0=12$); г) iifnzzk ($c_0=22$); ж) jqbfltng ($t_0=7$);

б) hldswxq ($c_0=16$); д) nvgxkxrl ($t_0=5$); з) xmqhhaof ($c_0=9$) .

в) inbhalv ($t_0=8$); е) ysdwetec ($c_0=12$);

Раздел 7. **Алфавитное кодирование**

Для передачи, хранения и переработки информации в различных системах необходимы знания теории кодирования. Всякое сообщение, записанное с использованием символов некоторого алфавита, можно представить в виде некоторой последовательности из нулей и единиц. Для практики важно, чтобы коды сообщений имели по возможности наименьшую длину. Использование информации о распределении вероятности появления букв в сообщении позволяет строго поставить и решить задачу построения оптимального алфавитного кодирования.

Метод Хаффмена построения оптимального кода

1. Упорядочить по невозрастанию список вероятностей.

2. Две последние вероятности исключаются из списка, а их сумма вставляется в список таким образом, чтобы в получившемся новом списке вероятности не возрастали. Процедура повторяется до тех пор, пока не получится список из двух вероятностей.

3. Одной из вероятностей приписывается символ 0, а другой – символ 1 (оптимальный код для двухбуквенного алфавита сообщений при любом распределении вероятностей).

4. Затем строится оптимальный код для трех букв при соответствующем списке вероятностей и т. д. до тех пор, пока не получится оптимальный код при исходном списке вероятностей.

Цена кодирования: $l(P)=\sum_{i=1}^{n} l_i p_i$, где l_i – длина кодового слова.

Для оптимального кода цена кодирования является минимальной.

Пример 7.1. Используя метод Хаффмена, построить оптимальный код по распределению вероятностей: P = {0,4; 0,2; 0,2; 0,1; 0,1}.

i	p_i							
1	0,4	0,4	**0,4**	**0,6**	**0**	**1**	00	00
2	0,2	**0,2**	0,4	0,4	1	00	**01**	10
3	0,2	0,2	0,2			01	10	11
4	0,1	0,2					11	010
5	0,1							011

$l(P)=2\cdot 0{,}4+2\cdot 0{,}2+2\cdot 0{,}2+3\cdot 0{,}1+3\cdot 0{,}1=2{,}2$.

Метод Фано построения кодов, близких к оптимальным

1. Упорядоченный по невозрастанию список вероятностей делится на две последовательные части так, чтобы суммы вероятностей, входящих в эти части, различались как можно меньше.

2. Каждой букве алфавита сообщений, соответствующей вероятности из первой части, сопоставляется символ 0, а остальным буквам – символ 1.

Далее точно так же поступают с каждой из частей, если она содержит по крайней мере две вероятности. Процесс продолжается до тех пор, пока весь список не разобьется на части, содержащие по одной вероятности.

Пример 7.2. Используя метод Фано, построить код для распределения вероятностей из примера 7.1. Выясните, является ли код оптимальным.

i	p_i		
1	0,4	1	00
2	0,2	01	01
3	0,2	000	10
4	0,1	0010	110
5	0,1	0011	111

Цена кодирования равна 2,2. Значит, код является оптимальным.

Метод Шеннона построения кодов, близких к оптимальным

Пусть $P = (p_1, p_2, \ldots, p_m)$ – упорядоченный по невозрастанию список вероятностей.

Тогда каждой букве алфавита сообщений сопоставляется кодовое слово длины $L_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$, составленное из первых после запятой цифр разложения числа $q_{i-1} = \sum_{j=1}^{i-1} p_j$ в бесконечную двоичную дробь (с недостатком).

Пример 7.3. Используя метод Шеннона, построить код для распределения вероятностей из примера 7.1. Выясните, является ли код оптимальным.

$P = \{0{,}4;\ 0{,}2;\ 0{,}2;\ 0{,}1;\ 0{,}1\}$.

Найдем длины кодовых слов:

$$L_1 = \left\lceil \log_2 \frac{1}{0{,}4} \right\rceil = \lceil \log_2 2{,}5 \rceil = 2.$$

$$L_2 = L_3 = \left\lceil \log_2 \frac{1}{0{,}2} \right\rceil = \lceil \log_2 5 \rceil = 3.$$

$$L_4 = L_5 = \left\lceil \log_2 \frac{1}{0{,}1} \right\rceil = \lceil \log_2 10 \rceil = 4.$$

Найдем кодовые слова:

$0{,}4 + 0{,}2 = 0{,}6$ $\quad$ $0{,}4 + 0{,}2 + 0{,}2 = 0{,}8$ $\quad$ $0{,}4 + 0{,}2 + 0{,}2 + 0{,}1 = 0{,}9$

0,	4
	×2
0	8
	×2
1	6
	×2
1	2

0,	6
	×2
1	2
	×2
0	4
	×2
0	8

0,	8
	×2
1	6
	×2
1	2
	×2
0	4
	×2
0	8

0,	9
	×2
1	8
	×2
1	6
	×2
1	2
	×2
0	4

Код: {00, 011, 100, 1100, 1110}.

Цена кодирования равна 2,8. Значит, код не является оптимальным.

Упражнения

1. Используя метод Хаффмена, постройте оптимальный код по распределению вероятностей:

а) $P = \{0{,}35;\ 0{,}18;\ 0{,}18;\ 0{,}16;\ 0{,}15\}$;
б) $P = \{0{,}31;\ 0{,}22;\ 0{,}18;\ 0{,}17;\ 0{,}16\}$;
в) $P = \{0{,}33;\ 0{,}31;\ 0{,}16;\ 0{,}14;\ 0{,}06\}$;
г) $P = \{0{,}25;\ 0{,}24;\ 0{,}24;\ 0{,}21;\ 0{,}06\}$;
д) $P = \{0{,}33;\ 0{,}33;\ 0{,}12;\ 0{,}11;\ 0{,}11\}$;
е) $P = \{0{,}25;\ 0{,}23;\ 0{,}23;\ 0{,}23;\ 0{,}06\}$;
ж) $P = \{0{,}33;\ 0{,}21;\ 0{,}16;\ 0{,}15;\ 0{,}15\}$;
з) $P = \{0{,}24;\ 0{,}22;\ 0{,}22;\ 0{,}16;\ 0{,}16\}$;
и) $P = \{0{,}34;\ 0{,}19;\ 0{,}16;\ 0{,}16;\ 0{,}15\}$;
к) $P = \{0{,}44;\ 0{,}23;\ 0{,}11;\ 0{,}11;\ 0{,}11\}$.

2. Используя методы Фано и Шеннона, постройте коды для распределения вероятностей из упражнения 1. Выясните, являются ли коды оптимальными.

Раздел 8. **Метод математической индукции**

Множество натуральных чисел бесконечно. Проверить справедливость гипотезы напрямую при бесконечном множестве исследуемых объектов невозможно. В этом случае применяется метод рассуждений, заменяющий полный перебор всех вариантов, который также дает достоверный вывод. Этот метод носит название метода математической индукции. В этом разделе мы рассмотрим применение метода математической индукции для построения доказательств.

Принцип метода математической индукции

1. Базис (или база).

Проверяется справедливость утверждения для $n = 1$.

2. Индукционный шаг.

Допускается справедливость утверждения для $n = k$. На основании этого допущения доказывается справедливость утверждения для $n = k + 1$.

3. Вывод.

На основании метода математической индукции это утверждение будет справедливо для любого натурального n.

Пример 8.1. Сумма первых n нечетных чисел равна квадрату их количества

$1 + 3 + 5 + \ldots + (2n - 1) = n^2$.

1. База.

$n = 1 \quad 1 = 1^2$

2. Индукционный шаг.

Предполагаем, что для всех n, не превосходящих k, $1 + 3 + 5 + \ldots + (2k - 1) = k^2$ истинно.

Докажем, что для $n = k + 1$ выполняется:

$1 + 3 + 5 + \ldots + (2k - 1) + (2(k+1) - 1) = (k + 1)^2$

$k^2 + 2k + 1 = (k + 1)^2$.

3. Вывод: для любого натурального n сумма n нечетных чисел равна n^2 на основании принципа математической индукции.

Упражнения

1. Доказать равенства методом математической индукции:

а) $1^2+3^2+5^2+\ldots+(2n-1)^2=\frac{n(2n-1)(2n+1)}{3}$;

б) $1^2+2^2+3^2+\ldots+n^2=\frac{n(n+1)(2n+1)}{6}$;

в) $2^2+4^2+6^2+\ldots+(2n)^2=\frac{2n(n+1)(2n+1)}{3}$;

г) $\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\ldots+\frac{1}{n(n+1)}=\frac{n}{n+1}$;

д) $\frac{1}{4\cdot 5}+\frac{1}{5\cdot 6}+\ldots+\frac{1}{(n+3)(n+4)}=\frac{n}{4(n+4)}$;

е) $\frac{1}{1\cdot 3}+\frac{1}{3\cdot 5}+\ldots+\frac{1}{(2n-1)(2n+1)}=\frac{n}{2n+1}$;

ж) $\frac{1}{1\cdot 5}+\frac{1}{5\cdot 9}+\ldots+\frac{1}{(4n-3)(4n+1)}=\frac{n}{4n+1}$;

з) $\frac{1}{1\cdot 4}+\frac{1}{4\cdot 7}+\ldots+\frac{1}{(3n-2)(3n+1)}=\frac{n}{3n+1}$.

Раздел 9. **Алгоритмическое перечисление (генерирование) комбинаторных объектов**

В данном разделе рассмотрены способы получения всех перестановок данных чисел и всех подмножеств заданного множества. Алгоритмы приведены в словесной форме, но могут быть переведены на один из языков программирования.

Генерирование перестановок заданной длины

Количество перестановок длины n равно $n!$

1 способ.

Пусть длина перестановки равна n.

1. Записываем последовательность чисел в возрастающем порядке:

$c_1, \ldots, c_n$.

2. «До какого числа меняем».

Просматривая последовательность справа налево, находим число, которое меньше числа, стоящего справа от него. Пусть это будет число *b*.

3. «Какое ставим».

Просматривая снова последовательность справа налево, находим число, которое больше числа *b*. Пусть это будет число *a*. Меняем *a* и *b* местами.

4. Меняем конец, т. е. меняем местами числа, стоящие после *a*.

5. Процесс заканчивается, когда получим убывающую последовательность:

$c_n, \ldots, c_1$.

Пример 9.1. Сгенерировать перестановки чисел 1, 7, 8, 9.

1, 7, 8, 9	7, 1, 8, 9	8, 1, 7, 9	9, 1, 7, 8
1, 7, 9, 8	7, 1, 9, 8	8, 1, 9, 7	9, 1, 8, 7
1, 8, 7, 9	7, 8, 1, 9	8, 7, 1, 9	9, 7, 1, 8
1, 8, 9, 7	7, 8, 9, 1	8, 7, 9, 1	9, 7, 8, 1
1, 9, 7, 8	7, 9, 1, 8	8, 9, 1, 7	9, 8, 1, 7
1, 9, 8, 7	7, 9, 8, 1	8, 9, 7, 1	9, 8, 7, 1

2 способ.

1. Записываем возрастающую последовательность $c_1, \ldots, c_n$.

2. Если число c_n стоит в конце перестановки, то:

а) меняем местами число c_n и число, стоящее слева от него, до тех пор, пока число c_n не будет стоять в начале перестановки;

б) меняем местами два числа, стоящие в конце перестановки.

3. Если число c_n стоит в начале перестановки, то:

а) меняем местами число c_n и число, стоящее справа от него, до тех пор, пока c_n не будет стоять в конце перестановки;

б) меняем местами два числа, стоящие в начале перестановки.

4. Процесс продолжается до тех пор, пока не получим последовательность $c_1, \ldots, c_n$.

Пример 9.2. Сгенерировать перестановки чисел 1, 7, 8, 9.

1, 7, 8, **9**	**9**, 1, 8, 7	8, 1, 7, **9**	**9**, 8, 7, 1	7, 8, 1, **9**	**9**, 7, 1, 8
1, 7, **9**, 8	1, **9**, 8, 7	8, 1, **9**, 7	8, **9**, 7, 1	7, 8, **9**, 1	7, **9**, 1, 8
1, **9**, 7, 8	1, 8, **9**, 7	8, **9**, 1, 7	8, 7, **9**, 1	7, **9**, 8, 1	7, 1, **9**, 8
9, 1, 7, 8	1, 8, 7, **9**	**9**, 8, 1, 7	8, 7, 1, **9**	**9**, 7, 8, 1	7, 1, 8, **9**

Генерирование всех подмножеств заданного множества

1 способ.

Выписываем все к-элементные подмножества заданного множества, где к = 0, 1,…, n.

Количество *к*-элементных подмножеств *n*-элементного множества равно числу сочетаний из *n* по *k*:

$$C_n^к = \frac{n!}{к!(n-к)!}$$

Пример 9.3. Пусть дано множество {1, 7, 8, 9,}. Найдём все его подмножества.

k=0 Ø
k=1 {1}, {7}, {8}, {9}.
k=2 {1, 7}, {1, 8}, {1, 9}, {7, 8}, {7, 9}, {8, 9}
k=3 {1, 7, 8}, {1, 7, 9}, {1, 8, 9}, {7, 8, 9}
k=4 {1, 7, 8, 9}

2 способ.

Множество $X = \{x_1, \ldots, x_n\}$ имеет 2^n подмножеств.

Каждому подмножеству $Y \subset X$ можно сопоставить двоичный набор $b_1, \ldots, b_n$, определяемый следующим образом:

$$b_i = \begin{cases} 0, \text{ если } x_i \notin Y; \\ 1, \text{ если } x_i \in Y. \end{cases}$$

Пример 9.4. Пусть дано множество {1, 7, 8, 9}. Найдём все его подмножества.

0000	Ø	1000	{1}
0001	{9}	1001	{1, 9}
0010	{8}	1010	{1, 8}
0011	{8, 9}	1011	{1, 8, 9}
0100	{7}	1100	{1, 7}
0101	{7, 9}	1101	{1, 7, 9},
0110	{7, 8}	1110	{1, 7, 8},
0111	{7, 8, 9}	1111	{1, 7, 8, 9}

Упражнения

1. Сгенерировать перестановки чисел двумя способами:

а) 1, 3, 5, 7;	в) 2, 3, 5, 6;	д) 1, 2, 8, 9;	ж) 1, 2, 4, 8;
б) 2, 4, 6, 8;	г) 4, 5, 6, 7;	е) 1, 3, 6, 9;	з) 2, 3, 7, 8.

2. Сгенерировать все подмножества заданного множества двумя способами:

а) 1, 3, 5, 7;	в) 2, 3, 5, 6;	д) 1, 2, 8, 9;	ж) 1, 2, 4, 8;
б) 2, 4, 6, 8;	г) 4, 5, 6, 7;	е) 1, 3, 6, 9;	з) 2, 3, 7, 8.

Раздел 10. **Основы теории графов**

В этом разделе описывается универсальный и наглядный язык – графический, который применяется во многих областях науки и техники. Понятие «граф» ввел в 1936 г. венгерский математик Денни Кёниг. Но первая работа по теории графов принадлежит перу великого Леонарда Эйлера и была написана еще в 1736 г. С помощью графов изображаются схемы различных дорог, линии воздушных сообщений, газопроводов, теплотрасс, электросетей, а также микросхемы, химические структурные формулы и другие диаграммы и схемы.

Неориентированные графы

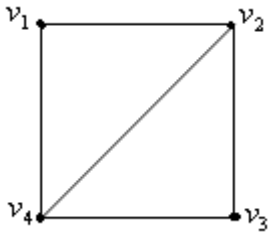
Граф *G*(V, E) состоит из непустого множества **вершин** V и множества **ребер** E. Каждое ребро соединяет пару вершин. Если ребро соединяет вершины V_1 и V_2, то говорят, что ребро E и вершины V_1 и V_2 **инцидентны.** Степенью d(v) вершины v называется количество ребер, инцидентных этой вершине.

Обозначение: p – число вершин графа, q – число ребер.

Матрица смежности – это квадратная матрица размера p×p, где

$$M(i,j) = \begin{cases} 1, & \text{если имеется ребро, соединяющее вершины } v_i \text{ и } v_j, \\ 0, & \text{в противном случае.} \end{cases}$$

Пример 10.1.



Граф

	v_1	v_2	v_3	v_4
v_1	0	1	0	1
v_2	1	0	1	1
v_3	0	1	0	1
v_4	1	1	1	0

Матрица смежности

Маршрутом в графе называется чередующаяся последовательность вершин и ребер, в которой любые два соседних элемента инцидентны.

Маршрут называется **замкнутым**, если он начинается и заканчивается в одной вершине, в противном случае – открытым.

Цепь – это маршрут, в котором нет повторяющихся ребер.

Простой цепью называется маршрут, в котором все вершины и ребра различны, кроме, может быть, первой и последней.

Цикл – замкнутая цепь.

Простым циклом называется замкнутая простая цепь.

Расстоянием между вершинами U и V называется длина кратчайшей цепи, соединяющей вершины U и V.

Нахождение расстояния между вершинами U и V

1. Разметим вершины:

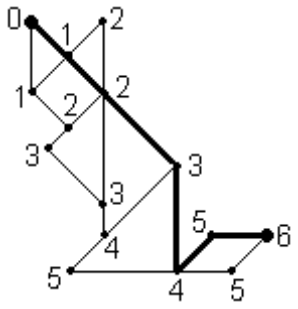
а) вершину U пометим нулем;

б) если на некотором шаге разметки имеются вершины U_j, помеченные числом j, то все еще не помеченные вершины, смежные с U_j, помечаем числом j+1;

в) как только вершина V окажется помечена, разметка прекращается.

2. Строим кратчайшую цепь, соединяющую вершины U и V.

Пример 10.2. Найдем расстояние между вершинами



Расстояние равно 6.

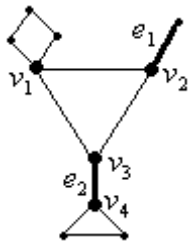
Под **операцией удаления вершины** из графа *G* понимается операция, заключающаяся в удалении некоторой вершины вместе с инцидентными ей ребрами.

Операция удаления ребра из графа *G* заключается в удалении соответствующей пары из *E*, при этом вершины сохраняются.

Вершина графа называется **разделяющей вершиной** (**точкой сочленения**), если ее удаление увеличивает число компонент связности.

Мост – ребро, удаление которого увеличивает число компонент связности.

Пример 10.3.



e_1, e_2 – мосты,

v_1, v_2, v_3, v_4 – точки сочленения.

Две вершины в графе **связаны**, если существует соединяющая их цепь.

Граф, в котором все вершины связаны, называются **связным**.

Компоненты связности графа – это его максимальные связные подграфы.

Методика выделения компонент связности

Пример 10.4.

Пусть граф задан матрицей смежности:

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
a	0	1	0	0	0	0	0	0
b	1	0	1	0	0	1	0	0
c	0	1	0	0	0	1	0	0
d	0	0	0	0	1	0	1	0
e	0	0	0	1	0	0	1	0
f	0	1	1	0	0	0	0	1
g	0	0	0	1	1	0	0	0
h	0	0	0	0	0	1	0	0

Составим вспомогательную таблицу:

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
1	2	3	0	0	3	0	4

Вершину *a* пометим единицей.

Вершины, смежные с *a*, пометим числом 2 (*b*).

Вершины, смежные с *b* и еще не помеченные, пометим числом 3 (*c*, *f*).

Вершины, смежные с *c* и *f* и еще не помеченные, пометим числом 4 (*h*).

На этом процесс обрывается, так как с вершиной *h* смежна только вершина *f*, но она уже помечена, получаем первую компоненту связанности: *a*, *b*, *c*, *f*, *h*.

Рассмотрим оставшиеся вершины.

d	*e*	*g*
1	2	2

Вершину *d* пометим единицей.

С вершиной *d* смежны вершины *e* и *g*, пометим их числом 2.

Получаем вторую компоненту связности: *d*, *e*, *g*.

Полные графы

Полным называется граф, в котором каждая пара вершин соединена ребром.

Обозначение: K_p.

Двудольные графы

Двудольный граф – это граф *G* (V, E), такой, что множество V разбито на два непересекающихся множества V_1 и V_2, причем всякое ребро из E инцидентно вершине из V_1 и вершине из V_2.

Множества V_1 и V_2 называются **долями** двудольного графа.

Если двудольный граф содержит все рёбра, соединяющие множества V_1 и V_2 , то он называется **полным двудольным графом**.

Обозначение: $K_{m,n}$, где m – количество вершин в доле V_1 , n – количество вершин в доле V_2 .

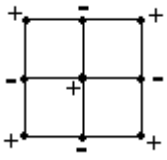
Методика проверки графа на двудольность

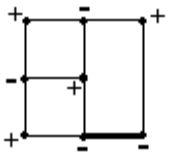
1. Выберем одну из вершин и пометим ее знаком «+».

2. Берём уже помеченную вершину X_i и помечаем все смежные с ней и ещё непомеченные вершины знаком, противоположным тому, которым помечена вершина X_i. Продолжаем эту операцию до тех пор, пока не будут помечены все вершины.

3. Если каждое ребро соединяет две вершины, помеченные противоположными знаками, то граф двудолен. Если найдется ребро, соединяющее вершины, помеченные одинаковыми знаками, то граф не двудолен.

Пример 10.5.

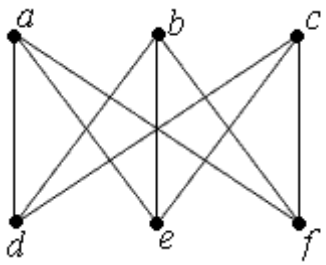

граф двудолен.

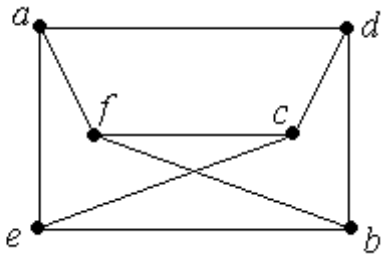

граф не двудолен.

Изоморфные графы

Графы G_1 и G_2 **изоморфны**, если существует взаимно однозначное соответствие между их вершинами и ребрами такое, что соответствующие ребра соединяют соответствующие вершины.

Пример 10.6.





Эйлеровы графы

Если граф имеет цикл, содержащий все ребра графа по одному разу, то такой цикл называется **эйлеровым циклом**, а граф – **эйлеровым графом**.

Методика нахождения эйлерова цикла в эйлеровом графе

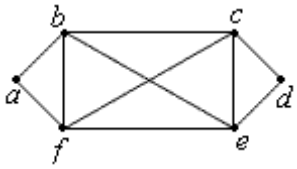
1. Убедимся в том, что граф является эйлеровым (степени всех вершин четны).

2. Начиная с произвольной вершины, строим цепь, удаляя рёбра и запоминая вершины, пока не придём в ту же вершину, с которой начали.

3. Берём следующую вершину и продолжаем процесс, пока не будут удалены все рёбра.

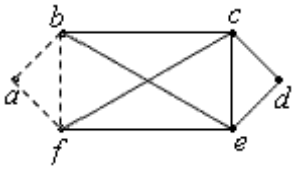
4. Объединяем полученные последовательности в одну.

Пример 10.7.

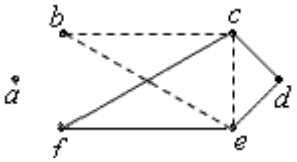


$d(a)=d(d)=2$

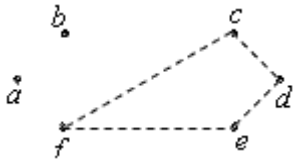
$d(b)=d(c)=d(e)=d(f)=4$ – граф эйлеров



abfa



bceb



cdefc

abcebfa

abcdefcebfa

Гамильтоновы циклы

Если граф имеет простой цикл, содержащий все вершины графа (по одному разу), то такой цикл называется **гамильтоновым**, а граф – **гамильтоновым**.

Плоские графы

Граф **укладывается** на некоторой поверхности, если его можно нарисовать на этой поверхности так, чтобы рёбра графа при этом не пересекались.

Граф называется **планарным**, если его можно уложить на плоскости.

Плоский граф – это граф, уже уложенный на плоскости.

Область, ограниченная рёбрами в плоском графе и не содержащая внутри себя вершин и рёбер, называется **гранью**.

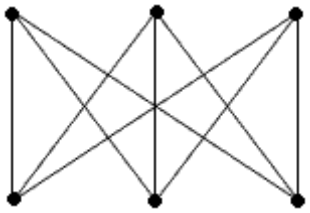
Обозначение: r– число граней плоского графа G.

Замечание. Внешняя часть плоскости также образует грань.

Теорема (формула Эйлера). В связном планарном графе справедливо следующее:

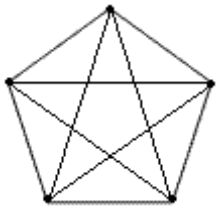
$$p - q + r = 2$$

Теорема.
Полный двудольный граф $K_{3,3}$ непланарен.



Доказательство.
Имеем $p = 6$, $q = 9$. В этом графе нет треугольников. Значит, если этот граф планарен, то в его плоской укладке каждая грань ограничена не менее чем четырьмя рёбрами, каждое ребро ограничивает не более двух граней. Следовательно, $4r \le 2q$ или $2r \le q$. По формуле Эйлера, $6 - 9 + r = 2$, откуда $r = 5$. Имеем $2r = 10 \le q = 9$ – противоречие.

Теорема.
Полный граф K_5 непланарен.



Доказательство.
Имеем $p = 5$, $q = p(p - 1)/2=10$. Если граф планарен, то в его плоской укладке каждой ребро ограничивает не более двух граней, каждая грань ограничена по крайней мере тремя рёбрами, отсюда $3r \le 2q$. По формуле Эйлера, $5 - 10 + r = 2$, откуда $r = 7$. Имеем $3r = 21 \le 2q = 20$ – противоречие.

Деревья

Граф называется **деревом,** если он связен и не имеет циклов.

Кодирование Прюфера для деревьев с пронумерованными вершинами

Пусть T – дерево с множеством вершин $\{ \upsilon_1, \ldots, \upsilon_n \}$.

Будем считать, что номер вершины равен i. Сопоставим дереву T последовательность

$\{a_1,\ldots, a_{n-2}\}$ по следующему правилу:

1. Полагаем i=1.

2. В последовательности 1,…,n (*) путём просмотра слева направо ищем номер первой висячей вершины. Пусть это будет b_i.

3. Ищем, с какой вершины смежна b_i. Пусть это будет a_i. Запоминаем a_i.

4. В последовательности (*) вычёркиваем b_i .

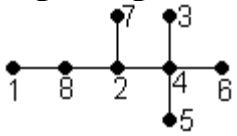
5. Из дерева T удаляем вершину b_i .

6. Полагаем i:=i+1.

7. Если i<n-1, то переходим к шагу 2. Если i=n-1, то выдаём последовательность

$[a_1,\ldots, a_{n-2}]$. Это и есть код Прюфера.

Пример 10.8.

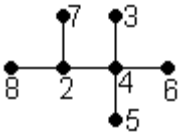


Просматривая последовательность 1, 2, 3, 4, 5, 6, 7, 8 слева направо, находим номер первой висячей вершины (1).

Вершина 1 смежна с вершиной 8. Запоминаем 8: [8].

В последовательности вычёркиваем 1: ~~1~~, 2, 3, 4, 5, 6, 7, 8.

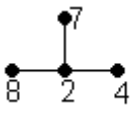
Из дерева удаляем вершину 1:



Вершины 3, 5, 6 висячие и смежны с вершиной 4. Получаем:

код [8, 4, 4, 4],

последовательность ~~1~~, 2, ~~3~~, 4, ~~5~~, ~~6~~, 7, 8,

дерево 

.

Вершины 4, 7 висячие и смежны с вершиной 2. Получаем:
код [8, 4, 4, 4, 2, 2],
последовательность ~~1~~, 2, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, 8,
дерево 

.
[8, 4, 4, 4, 2, 2] – код Прюфера.

Восстановление дерева по коду Прюфера

Пусть A=[a_1,…, $a_к$] – код Прюфера. Будем считать, что номер вершины равен i. Сопоставим последовательности A дерево T={ υ_1,…, $\upsilon_{к+2}$} по следующему правилу:

1. В последовательности 1,…,к+2 (*) путём просмотра слева направо ищем число *b*, не равное числам кода.

2. В T добавляем ребро, соединяющее вершины, соответствующие числу *b* и первому числу кода.

3. В последовательности (*) вычерчиваем b_1.

4. Из кода вычерчиваем первое число.

5. Если в последовательности (*) больше двух чисел, переходим к шагу 1. если в последовательности (*) два числа, добавляем в T ребро, соединяющее вершины, соответствующие этим числам.

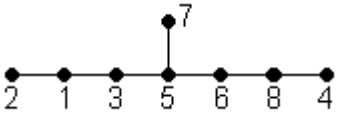
Пример 10.9.

[1, 3, 5, 8, 5, 6], k=6 – длина кода.

1. В последовательности 1, 2, 3, 4, 5, 6, 7, 8 путём просмотра слева направо находим число, не равное числам кода (2).

2. В дерево добавляем ребро, соединяющее вершины, соответствующие числу 2 и первому числу кода (1):

3. В последовательности вычерчиваем 2: 1, ~~2~~, 3, 4, 5, 6, 7, 8.
Из кода вычерчиваем первое число: [~~1~~, 3, 5, 8, 5, 6].
Продолжая данный процесс, получаем:



Кратчайший остов

Пусть *G* (*V*, *E*) – граф. **Остовной** подграф графа *G* (*V*, *E*) – это подграф, содержащий все вершины.

Остов – остовной подграф, являющийся деревом.

Ребра графа могут быть помечены числами. Эти числа обычно называют **длинами** рёбер.

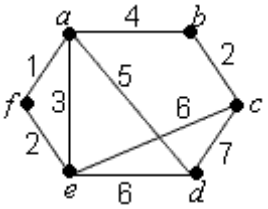
Кратчайший остов – остов, сумма длин рёбер которого минимальна.

Алгоритм Краскала нахождения кратчайшего остова

1. Пусть даны рёбра графа с их длинами. Упорядочим рёбра в порядке возрастания их длин.

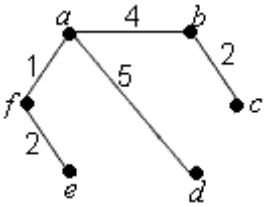
2. Просматриваем последовательность рёбер слева направо. Если при добавлении ребра получаем цикл, то пропускаем это ребро, иначе добавляем данное ребро в остов.

Пример 10.10.



1. $af = 1$, $bc = ef = 2$, $ae = 3$, $ab = 4$, $ad = 5$, $ce = de = 6$, $cd = 7$

2.



Раскраска графов

Раскраска графа – такое приписывание цветов (натуральных чисел) его вершинам, что никакие две смежные вершины не получают одинаковый цвет.

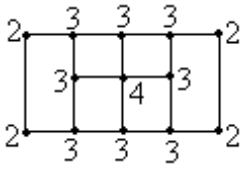
Улучшенный алгоритм последовательного раскрашивания

1. Упорядочим вершины в порядке невозрастания.

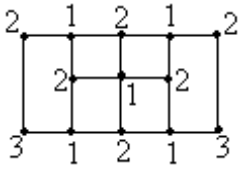
2. Начинаем раскраску с вершины, имеющей наибольшую степень. Красим всё, что можно в цвет 1, затем красим всё, что можно в цвет 2, и т. д.

Пример 10.11. Построим раскраску графа.

Найдем степени вершин графа:



Раскраска графа:



Ориентированные графы

Ориентированным графом (или **орграфом**) называется граф, в котором элементами множества рёбер являются упорядоченные пары.

В орграфе элементы множества V – **узлы**, а элементы множества E – **дуги**.

Полустепень исхода $d^-(v)$ – число дуг, исходящих из вершины v.

Полустепень захода $d^+(v)$ – число дуг, исходящих в вершины v.

Источник – вершина, полустепень захода которой равна нулю.

Сток – вершина, полустепень исхода которой равна нулю.

Способы задания орграфа

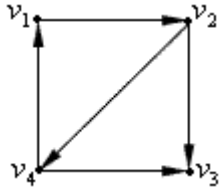
1. Графический способ. Узлы орграфа изображают точками, а дуги – стрелками.

2. Матрица смежности – квадратная матрица размера p × p, где

$$M[i,j] = \begin{cases} 1, \text{ если имеется дуга с началом в узле } v_i \text{ и концом в узле } v_j, \\ 0, \text{ в противном случае.} \end{cases}$$

Пример 10.12.

Граф:



Полустепени исхода и захода:

$d^-(v_1)=1,\ d^+(v_1)=1$

$d^-(v_2)=2,\ d^+(v_2)=1$

$d^-(v_3)=0,\ d^+(v_3)=2$

$d^-(v_4)=2,\ d^+(v_4)=1$

Матрица смежности:

	v_1	v_2	v_3	v_4
v_1	0	1	0	0
v_2	0	0	1	1
v_3	0	0	0	0
v_4	1	0	1	0

Маршруты, пути, контуры

Маршрут – чередующаяся последовательность узлов и дуг v_0, e_1, v_1, …, e_n, v_n, в которой каждая дуга e_i есть дуга с началом в узле v_{i-1} и концом в узле v_i.

Путь – маршрут, в котором все дуги различны

Замкнутым называются маршрут, в котором первые и последние узлы совпадают.

Контур – замкнутый путь.

Достижимость

Вершина v_j в орграфе *G* **достижима** из вершины v_i, если существует путь из v_i в v_j.

Отношение достижимости можно представить квадратной **матрицей** размера $p \times p$, где

$$T[i,j]=\begin{cases}1, \text{ если вершина } v_j \text{ достижима из вершины } v_i\\ 0, \text{ если вершина } v_j \text{ не достижима из вершины } v_i\end{cases}$$

Вершины v_i и v_j **сильно связаны** в орграфе *G*, если существует путь из v_i в v_j и из v_j в v_i.

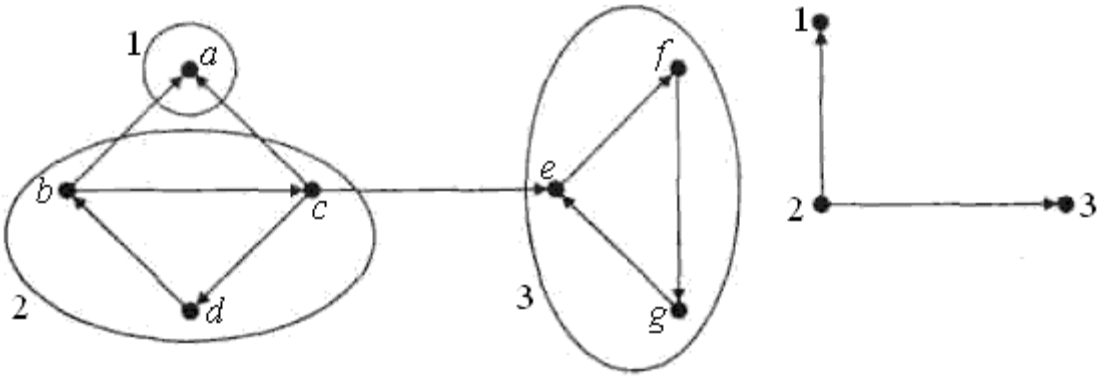
Если все вершины в орграфе сильно связаны, то орграф называется **сильно связным**.

Компоненты сильной связности (КСС) орграфа *G* – это его максимальные сильно связанные подграфы.

Каждая вершина орграфа принадлежит только одной КСС. Если вершина не связана с другими, то считаем, что она сама образует КСС.

Графом Герца называется орграф, который получается стягиванием в одну вершину каждой КСС орграфа G.

Пример 10.13.



Граф и его граф Герца

Матрица достижимости:

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	0	0	0	0	0	0
b	1	1	1	1	1	1	1
c	1	1	1	1	1	1	1
d	1	1	1	1	1	1	1
e	0	0	0	0	1	1	1
f	0	0	0	0	1	1	1
g	0	0	0	0	1	1	1

Эйлеровы орграфы

Эйлеровый контур в орграфе G – это, контур, в котором каждая дуга орграфа встречается ровно по одному разу.

Орграф называется **эйлеровым**, если в нём есть эйлеровый контур.

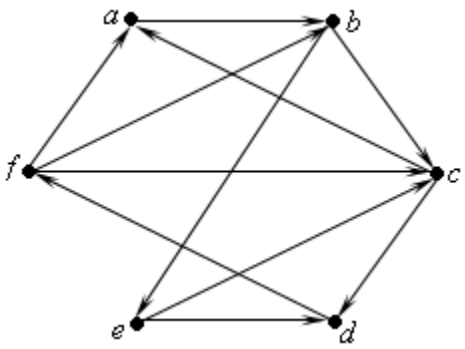
Теорема. Связный орграф эйлеров тогда и только тогда, когда у каждой его вершины полустепень исхода равна полустепени захода.

Гамильтоновы орграфы

Орграф называется **гамильтоновым**, если в нём существует контур, включающий каждую его вершину по одному разу (контур называется гамильтоновым).

Методика нахождения гамильтонова контура (поиск с возвратами)

Пример 10.14.



Составим вспомогательную таблицу. В каждом столбце таблицы перечислим вершины, смежные из данной.

a	*b*	*c*	*d*	*e*	*f*
b	*c* *e*	*a* *d*	*f*	*c* *d*	*a* *b* *c*

Построим гамильтонов контур.

1. a.
2. *ab* (возьмём вершину из столбца *a*).
3. *abc* (возьмём вершину из столбца *b*).
4. *abcd* (возьмём вершину из столбца *c*: *a* взять не можем, поэтому берём *d*).
5. *abcdf* (из столбца *d* берём *f*).
6. *abcd* (из столбца *f* не можем ничего взять, возвращение).
7. *abc*(в столбце *d* нет вершины после *f*, возвращение).
8. *ab* (в столбце с нет вершины после *d*, возвращение).
9. *abe* (из столбца *b* берём вершину *e*).
10. *abec* (из столбца *e* берём вершину *c*).
11. *abecd* (из столбца с не можем взять a, поэтому берём *d*).
12. *abecdf* (из столбца *d* берём *f*).
13. *abecdfa* (в столбце *f* есть вершина *a*).
14. *abecdfa* – гамильтонов контур.

Ориентированное дерево

Ориентированным деревом (или **ордеревом**) называется орграф со следующими свойствами:

1) существует единственный узел (**корень**), полустепень захода которого равна нулю;

2) полустепень захода всех остальных узлов равна 1;

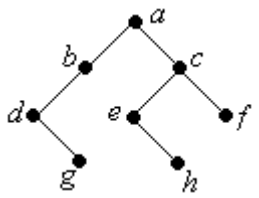
3) каждый узел достижим из корня

Бинарное дерево – это дерево, полустепень исхода каждой вершины которого не более двух.

Кодирование бинарных деревьев

В коде бинарного дерева фиксируется “размеченная степень” каждого узла (0 означает, что это лист, 1 – есть левая связь, но нет правой, 2 – есть правая связь, но нет левой, 3 – есть обе связи). Все узлы поддерева данного узла располагаются вслед за этим узлом. Если у данного узла есть обе связи, то сначала рассматриваем левую связь.

Пример 10.15.

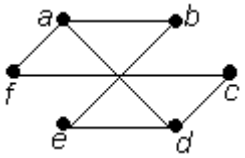


[*a*3 *b*1 *d*2 *g*0 *c*3 *e*2 *h*0 *f* 0].

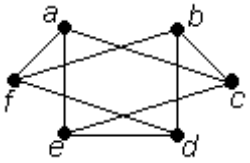
Упражнения

1. Построить матрицу смежности для данного графа:

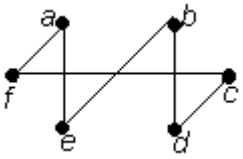
а)



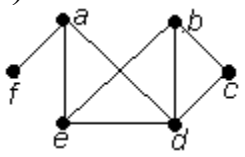
б)



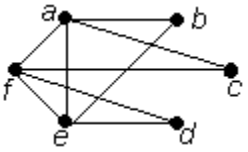
в)



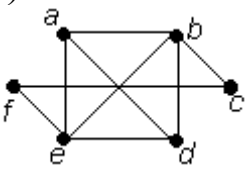
г)



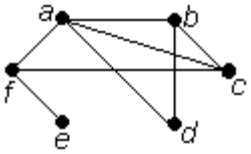
д)



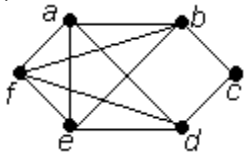
е)



ж)

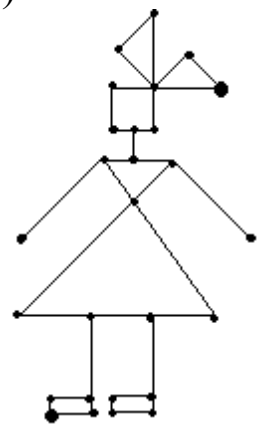


з)

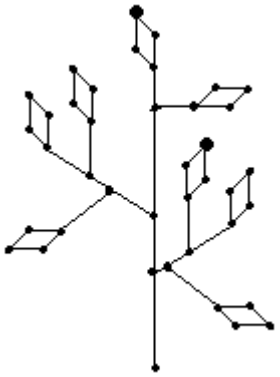


2. Дан граф. Найти все мосты и точки сочленения. Найти расстояние между двумя вершинами графа.

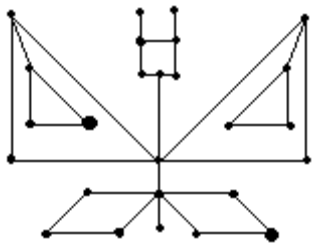
а)



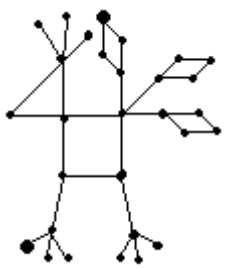
б)



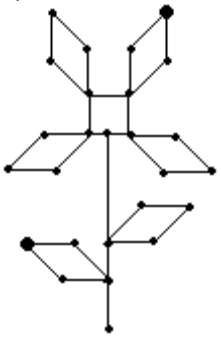
в)



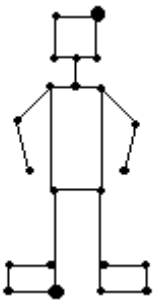
г)



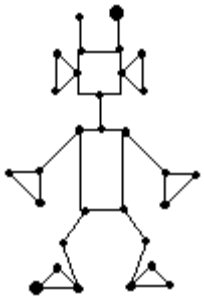
д)



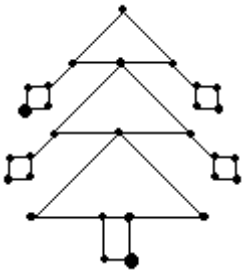
е)



ж)



з)



3. Выделить компоненты связности в графе, заданном матрицей смежности:

а)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	0	0	1	1	0	0
b	0	1	0	0	0	0	0
c	0	0	0	0	0	1	0
d	1	0	0	0	1	0	0
e	1	0	0	1	0	0	0
f	0	0	1	0	0	0	1
g	0	0	0	0	0	1	0

б)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	0	0	1	0	0	1
b	0	0	1	0	0	1	0
c	0	1	0	0	0	1	0
d	1	0	0	0	0	0	0
e	0	0	0	0	0	0	1
f	0	1	1	0	0	0	0
g	1	0	0	0	1	0	0

в)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	1	0	0	0	0	0
b	1	0	0	0	0	0	0
c	0	0	0	0	1	0	1
d	0	0	0	0	0	1	0
e	0	0	1	0	0	0	1
f	0	0	0	1	0	0	0
g	0	0	1	0	1	0	0

г)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	0	0	0	1	0	0
b	0	1	0	0	0	1	0
c	0	0	0	1	1	0	0
d	0	0	1	0	0	0	0
e	1	0	1	0	0	0	0
f	0	1	0	0	0	0	1
g	0	0	0	0	0	1	0

д)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	1	0	1	0	0	0
b	1	0	0	1	0	0	0
c	0	0	0	0	1	0	1
d	1	1	0	0	0	1	0
e	0	0	1	0	0	0	1
f	0	0	0	1	0	0	0
g	0	0	1	0	1	0	0

е)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	0	0	0	0	1	0
b	0	0	1	1	0	0	0
c	0	1	0	1	0	1	0
d	0	1	1	0	0	0	0
e	0	0	0	0	0	0	1
f	1	0	1	0	0	0	0
g	0	0	0	0	1	0	0

ж)

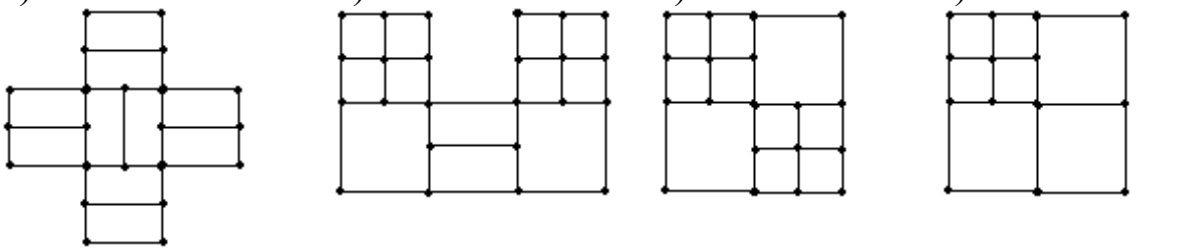
	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	1	1	0	1	0	0
b	1	0	1	0	1	0	0
c	1	1	0	0	1	0	1
d	0	0	0	0	0	1	0
e	1	1	1	0	0	0	1
f	0	0	0	1	0	0	0
g	0	0	1	0	1	0	0

з)

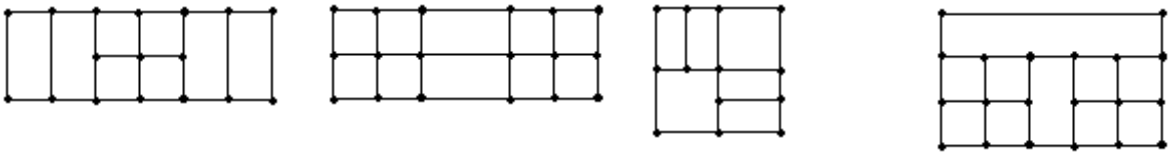
	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	0	0	0	0	1	1
b	0	0	0	0	0	1	1
c	0	0	0	1	1	0	0
d	0	0	1	0	1	0	0
e	0	0	1	1	0	0	0
f	1	1	0	0	0	0	0
g	1	1	0	0	0	0	0

4. Проверить, является ли данный граф двудольным:

а) б) в) г)

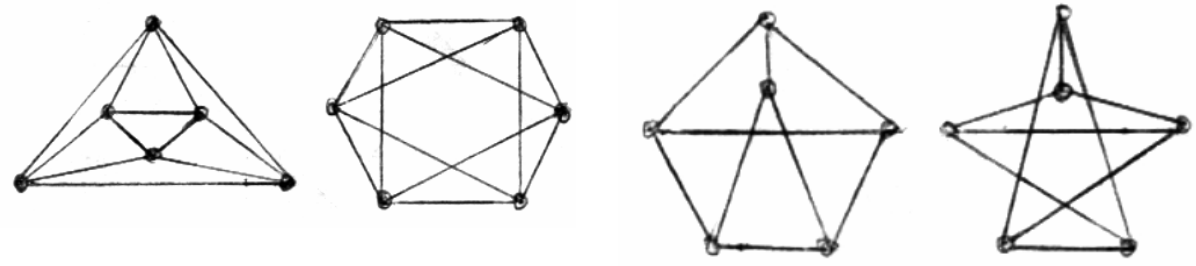


д) е) ж) з)

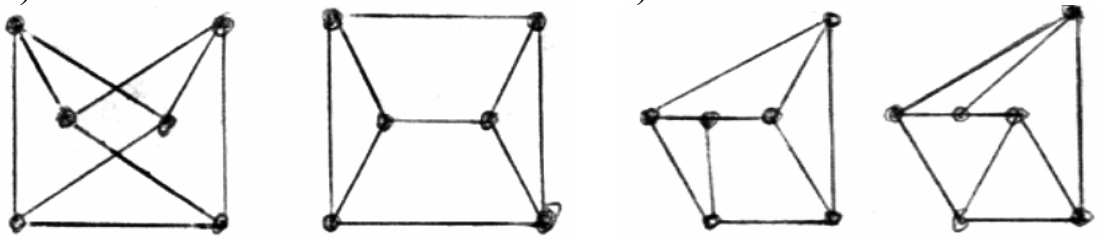


5. Проверить, являются ли два данных графа изоморфными:

а) б)

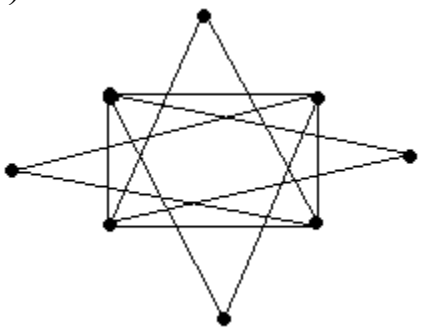


в) г)

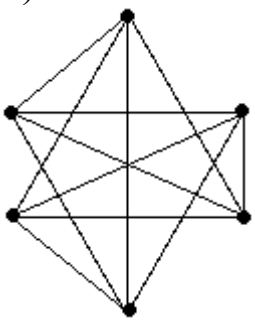


6. Доказать, что данный граф является эйлеровым и найти в нем эйлеров цикл:

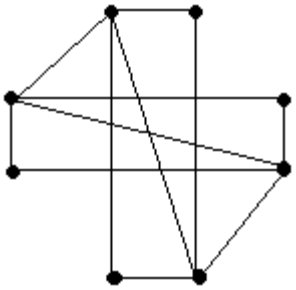
а)



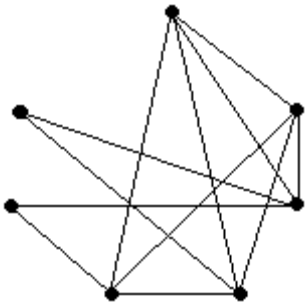
б)



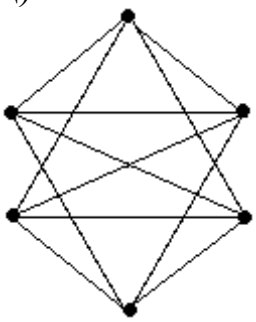
в)



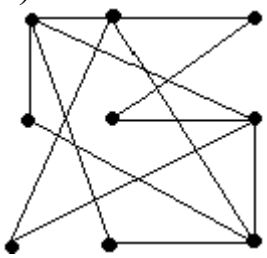
г)



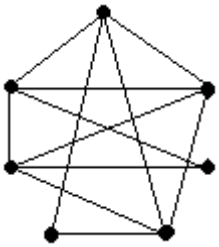
д)



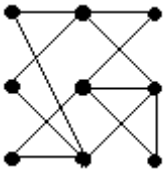
е)



ж)

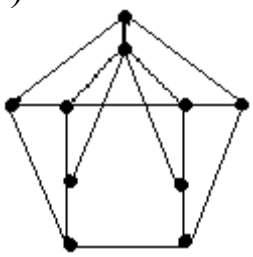


з)

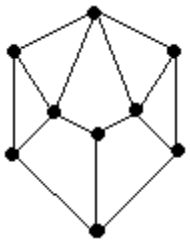


7. Проверить, является ли данный граф гамильтоновым:

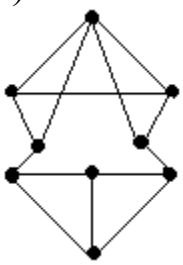
а)



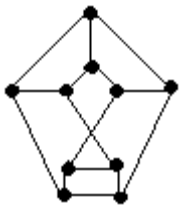
б)



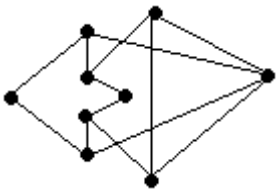
в)



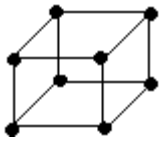
г)



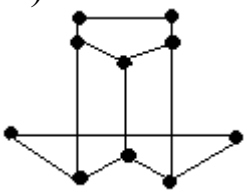
д)



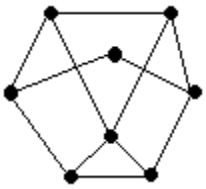
е)



ж)



з)

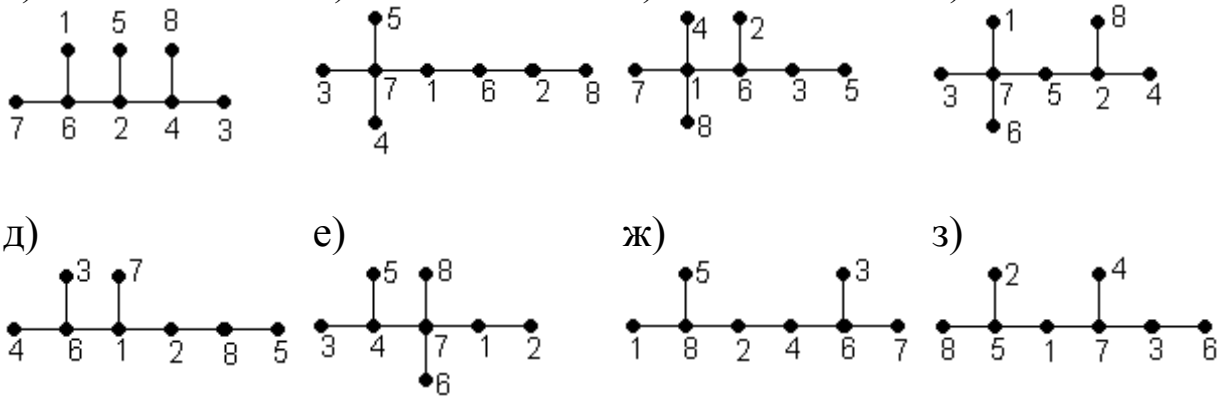


8. Проверить, является ли данный граф планарным:

а) $K_{7;}$ б) $K_{4,5}$**;** в) K_8; г) $K_{3,4}$;
д) $K_{4,4}$; е) K_9; ж) K_6; з) $K_{5,5}$

9. Записать для дерева с пронумерованными вершинами его код Прюфера:

а) б) в) г)



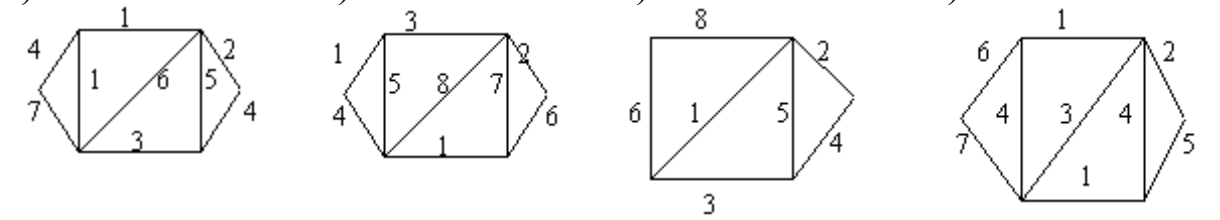
д) е) ж) з)

10. Восстановить по коду Прюфера дерево с пронумерованными вершинами:

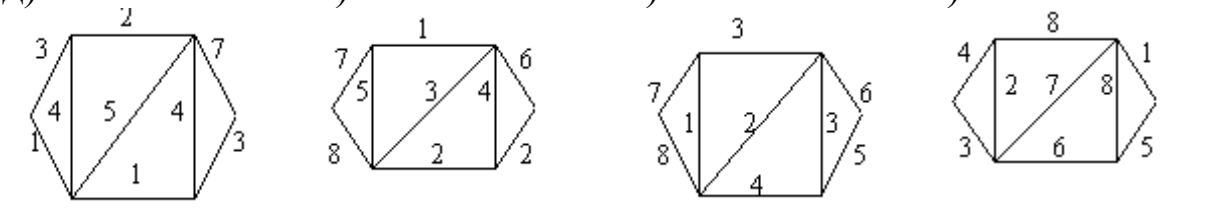
а) [2,2,4,4,6,6] б) [5,2,3,2,4,5] в) [3,8,4,2,8,3] г) [6,6,2,4,7,4]
д) [2,2,4,5,5,7] е) [4,4,2,8,3,3] ж) 2,3,3,3,5,6] з) [2,2,4,5,5,5]

11. Построить кратчайший остов графа:

а) б) в) г)

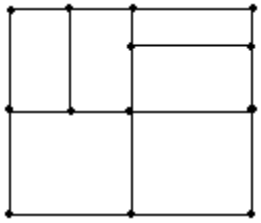


д) е) ж) з)

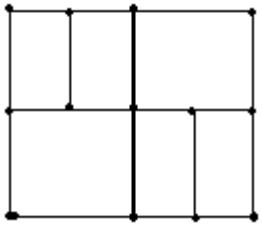


12. Построить раскраску графа:

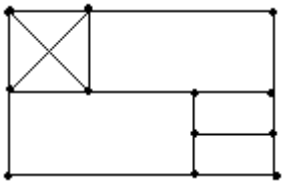
а)



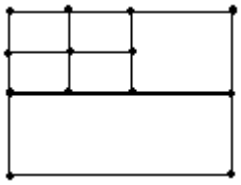
б)



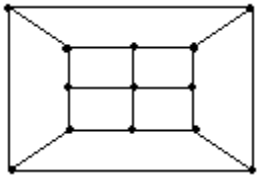
в)



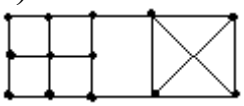
г)



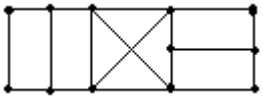
д)



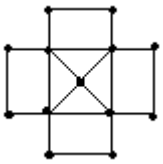
е)



ж)

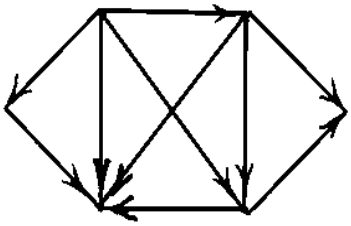


з)

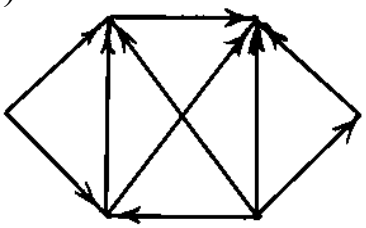


13. Записать матрицу смежности для орграфа, найти степени входа и выхода вершин, выделить в орграфе источники и стоки:

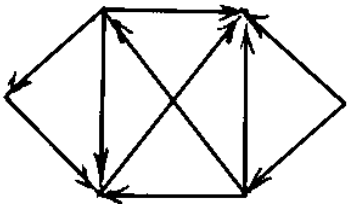
а)



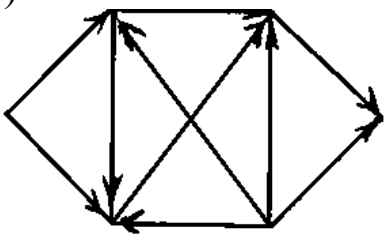
б)



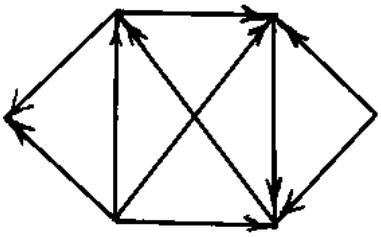
в)



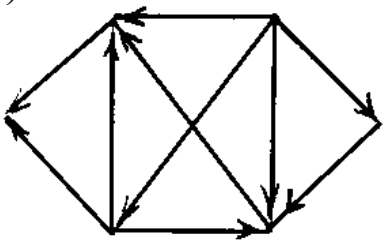
г)



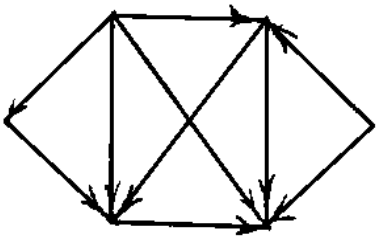
д)



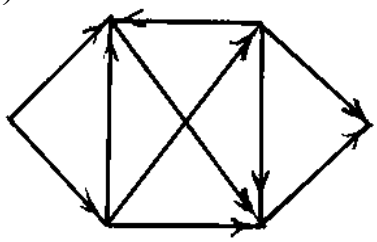
е)



ж)

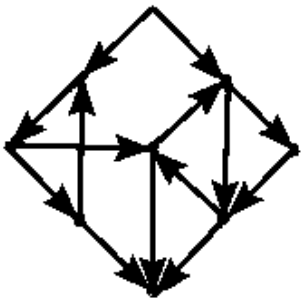


з)

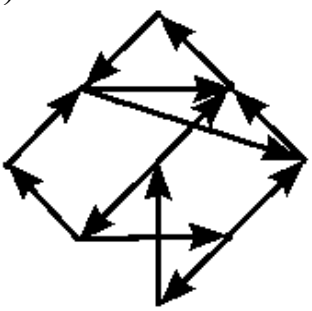


14. Записать матрицу достижимости орграфа, выделить классы эквивалентных вершин в орграфе, построить для орграфа его диаграмму Герца:

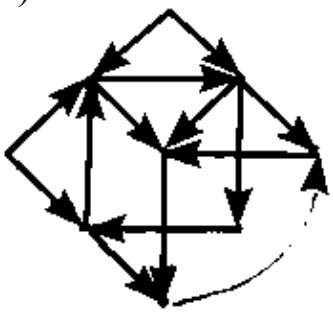
а)



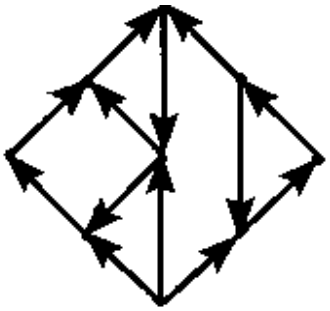
б)



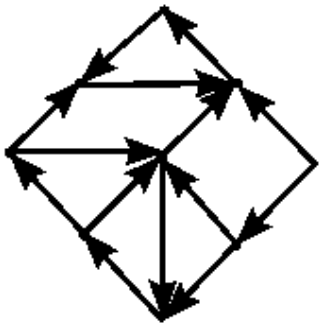
в)



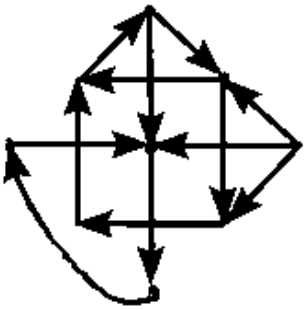
г)



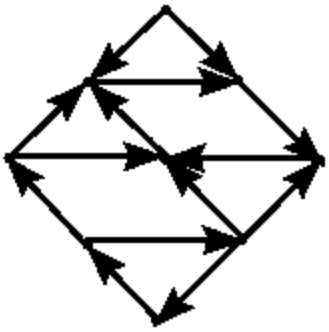
д)



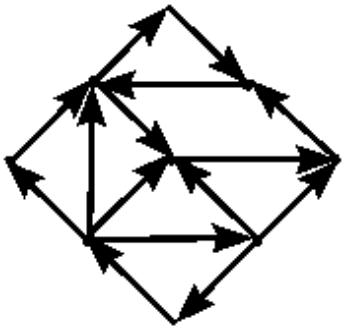
е)



ж)

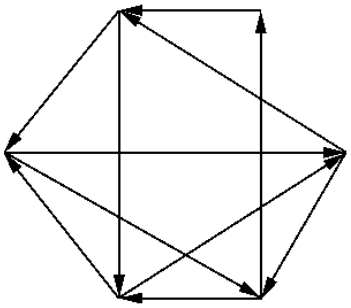


з)

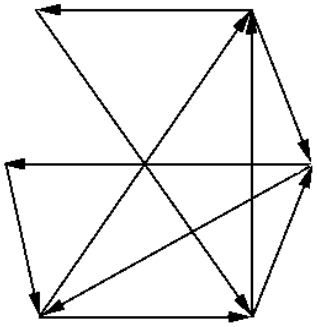


15. Проверить, является ли данный орграф эйлеровым и гамильтоновым:

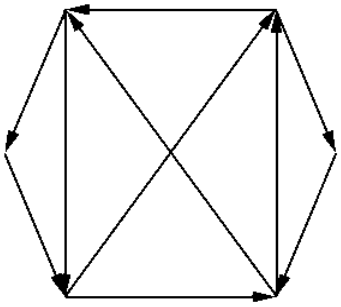
а)



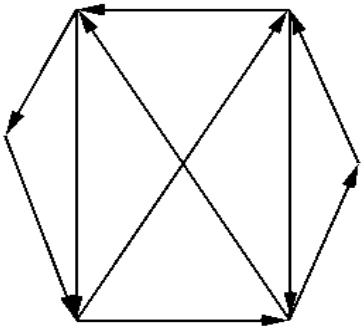
б)



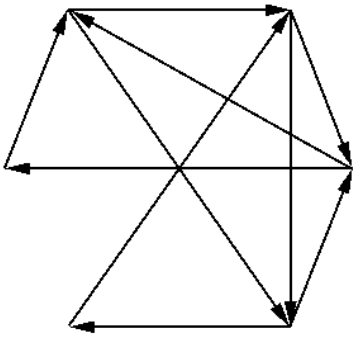
в)



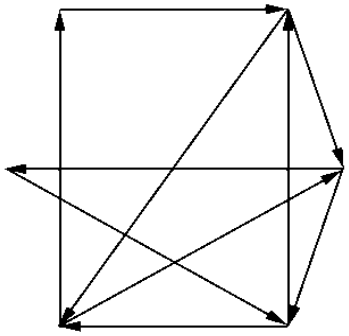
г)



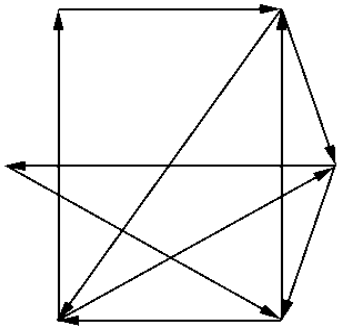
д)



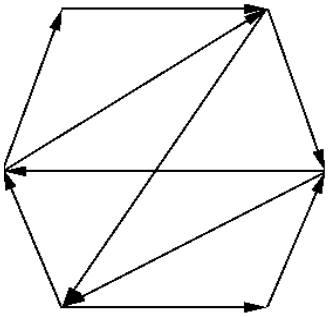
е)



ж)

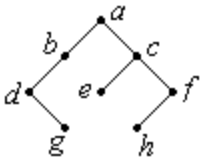


з)

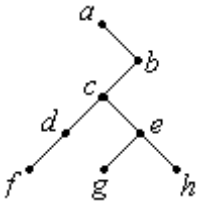


16. Записать код бинарного дерева:

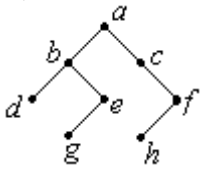
а)



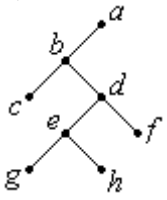
б)



в)



г)

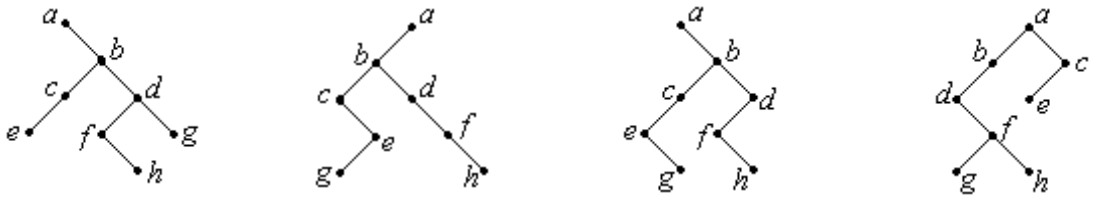


д)

е)

ж)

з)



17. По коду восстановить бинарное дерево:

а) [*a* 3 *b* 2 *c* 1 *d* 2 e 0 *f* 1 *g* 2 *h* 0];
б) [*a* 2 *b* 3 *c* 1 *d* 0 e 3 *f* 2 *g* 0 *h* 0];
в) [*a* 1 *b* 1 *c* 3 *d* 0 *e* 3 *f* o *g* 2 *h* 0];
г) [*a* 1 *b* 3 *c* 2 *d* 1 e 0 *f* 1 *g* 0 *h* 0];
д) [*a* 1 *b* 2 *c* 1 *d* 3 e 0 *f* 3 *g* 0 *h* 0];
е) [*a* 2 *b* 2 *c* 3 *d* 1 e 0 *f* 1 *g* 2 *h* 0];
ж) [*a* 3 *b* 1 *c* 0 *d* 3 e 2 *f* 0 *g* 1 *h* 0];
з) [*a* 3 *b* 3 *c* 2 *d* 0 e 1 *f* 0 *g* 2 *h* 0].

Раздел 11. **Элементы теории автоматов**

Мы постоянно работаем с разного рода автоматами, такими, как калькуляторы, телефонные коммутаторы, переключательные схемы лифтов. Все они имеют некие общие черты, а именно: автомат представляет собой устройство, которое может находиться в разных состояниях; эти состояния могут переходить в другие состояния под влиянием внешнего воздействия. Часто автомат реагирует, продуцируя выходы, в качестве которых могут фигурировать, например, мелкие монеты или результаты вычислений.

Автоматом называется набор V = (*A*, Q, *B*, φ, ψ), где

A, Q, *B* – конечные множества;

φ – функция, определенная на множестве Q×*A* и принимающая значения из Q: $\varphi(q_i, a_j) \in Q$;

ψ – функция, определенная на множестве Q×A и принимающая значения из B: $\psi(q_i, a_j) \in B$.

Множество A– **входной алфавит**.

Множество Q – **алфавит состояний**.

Множество B – **выходной алфавит**.

Функция φ – **функция переходов**.

Функция ψ – **функция выходов**.

Входные слова автомата V – произвольная конечная последовательность символов алфавита A.

Выходные слова автомата V – конечная последовательность символов алфавитов B.

Л – пустое слово, не имеющее ни одного символа.

Способы задания автоматов

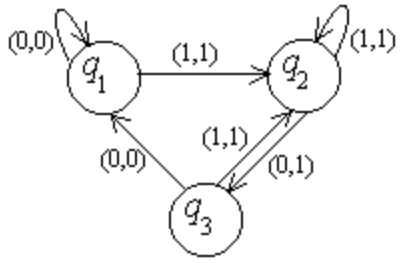
1 способ. Конечные множества *A, B, Q* можно задавать непосредственным перечислением их элементов. Функция φ и ψ можно задавать при помощи прямоугольных таблиц. Каждая строка таблиц взаимно однозначно сопоставляется символу q_i алфавита *Q*, $Q = \{q_1,..., q_n\}$. Каждый столбец таблиц взаимно однозначно сопоставляется символу a_j алфавита *A*, $A = \{a_1,..., a_m\}$. На пересечении строки, соответствующей символу q_i, и столбца, соответствующего символу a_j, содержится символ $\varphi(q_i, a_j)$ ($\psi(q_i, a_j)$).

2 способ. Автоматы можно задавать с помощью диаграмм Мура, которые строятся следующим образом:

1. На плоскости размещается *n* кругов. Внутри *i*-го круга записывается символ q_i алфавита $Q = \{q_1,..., q_n\}$.

2. Рассматриваются всевозможные пары (q_i, a_j), где q_i принадлежит *Q*, a_j принадлежит *A*, $A = \{a_1,..., a_m\}$. Для каждой такой пары от круга, в котором записан символ q_i, проводится стрелка к кругу, в котором записан символ $\varphi(q_i, a_j)$. Этой стрелке приписывается пара $(a_j, \psi(q_i, a_j))$. Из каждого круга диаграммы выходит ровно *m* стрелок *(m* – количество символов алфавита *A)*.

Пример 11.1.



φ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_2

ψ	0	1
q_1	0	1
q_2	1	1
q_3	0	1

Диаграмма Мура *Таблицы автомата*

Рассмотрим **принцип работы** данного автомата.

Пусть дано входное слово: α = 0110100. Найти выходное слово β.

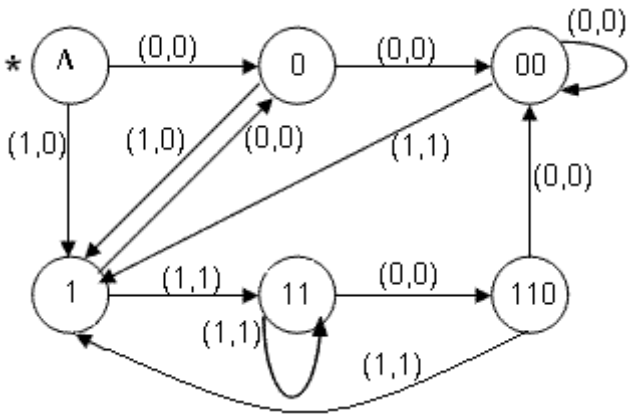
	α	q	β
В начале работы автомат находится в состоянии q_1.		q_1	
1 шаг. $\alpha_1 = 0$. Автомат остается в состоянии q_1 и выдает $\beta_1 = 0$.	0	q_1	0
2 шаг. $\alpha_2 = 1$. Автомат переходит в состояние q_2 и выдает $\beta_2 = 1$	1	q_2	1
3 шаг. $\alpha_3 = 1$. Автомат остается в состоянии q_2 и выдает $\beta_3 = 1$.	1	q_2	1
4 шаг. $\alpha_4 = 0$. Автомат переходит в состояние q_3 и выдает $\beta_4 = 1$	0	q_3	1
5 шаг. $\alpha_5 = 1$. Автомат переходит в состояние q_2 и выдает $\beta_5 = 1$.	1	q_2	1
6 шаг. $\alpha_6 = 0$. Автомат переходит в состояние q_3 и выдает $\beta_6 = 1$	0	q_3	1
7 шаг. $\alpha_7 = 0$. Автомат переходит в состояние q_1 и выдает $\beta_7 = 0$.	0	q_1	0

Получаем выходное слово β = 0111110.

Построение автоматов

Пример 2. Требуется построить автомат, распознающий во входной последовательности подслова одного из видов 001, 11, 1101 и выдающий на выходе единицу в точности в те моменты, когда на входе появляется последняя буква одного из этих подслов.

Решение. Рассмотрим слова 00, 1, 110, получающиеся из указанных подслов отбрасыванием последней буквы. Для каждого из начал Л, 0, 00, 1, 11, 110 этих слов выделим состояние автомата, в котором он будет оказываться, как только конец входного слова совпадает с соответствующим началом (выбирается самое длинное из таких начал, если их несколько).



Упражнения

1. По таблице автомата построить его диаграмму:

а)

φ	0	1
q_1	q_2	q_3
q_2	q_1	q_4
q_3	q_3	q_4
q_4	q_1	q_2

ψ	0	1
q_1	0	1
q_2	1	0
q_3	0	1
q_4	1	0

б)

φ	*0*	*1*
q_1	q_3	q_4
q_2	q_1	q_2
q_3	q_3	q_1
q_4	q_2	q_4

ψ	0	1
q_1	0	1
q_2	1	1
q_3	1	0
q_4	0	0

в)

φ	0	1
q_1	q_1	q_4
q_2	q_3	q_2
q_3	q_4	q_1
q_4	q_2	q_3

ψ	0	1
q_1	0	1
q_2	1	0
q_3	0	0
q_4	1	1

г)

φ	*0*	*1*
q_1	q_4	q_3
q_2	q_2	q_1
q_3	q_4	q_1
q_4	q_2	q_3

ψ	0	1
q_1	1	1
q_2	0	0
q_3	0	0
q_4	1	1

д)

φ	0	1
q_1	q_2	q_1
q_2	q_4	q_1
q_3	q_4	q_3
q_4	q_3	q_2

ψ	0	1
q_1	0	0
q_2	0	1
q_3	1	0
q_4	1	1

е)

φ	*0*	*1*
q_1	q_4	q_1
q_2	q_2	q_3
q_3	q_1	q_2
q_4	q_3	q_4

ψ	0	1
q_1	0	0
q_2	1	1
q_3	0	0
q_4	1	1

ж)

φ	0	1
q_1	q_1	q_2
q_2	q_4	q_3

ψ	0	1
q_1	1	0
q_2	1	1

з)

φ	*0*	*1*
q_1	q_3	q_2
q_2	q_4	q_3

ψ	0	1
q_1	1	0
q_2	0	0

q_3	q_3	q_4
q_4	q_2	q_1

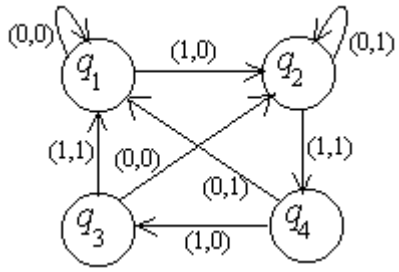
q_3	0	1
q_4	1	1

q_3	q_2	q_1
q_4	q_1	q_4

q_3	1	1
q_4	0	1

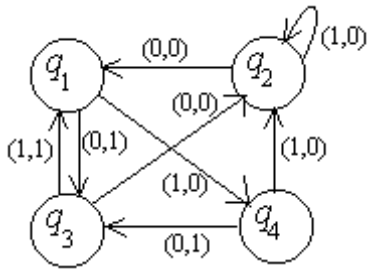
2. По диаграмме автомата записать его таблицу. По заданному входному слову записать соответствующее выходное слово.

а)



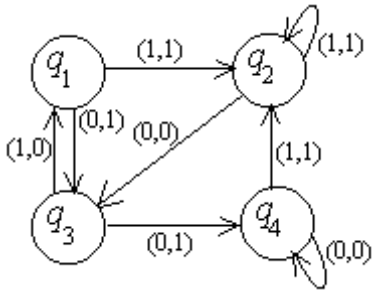
[1 0 1 1 1 0 0 1]

б)



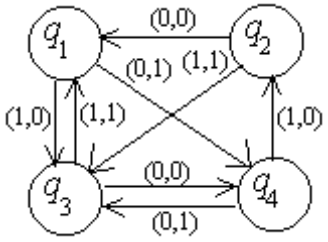
[1 1 0 1 1 0 0 1]

в)



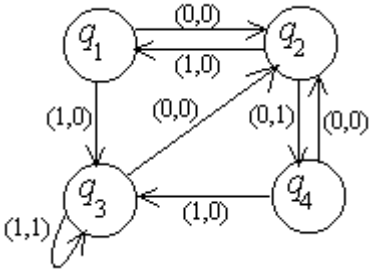
[0 0 0 1 1 0 1 0]

г)



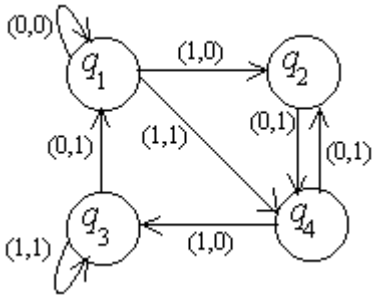
[1 1 0 1 1 1 0 1]

д)



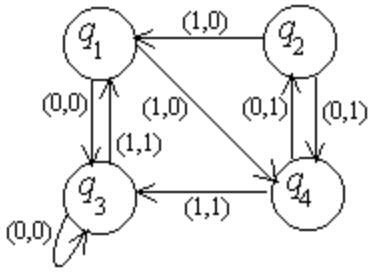
[1 0 1 0 0 1 0 1]

е)



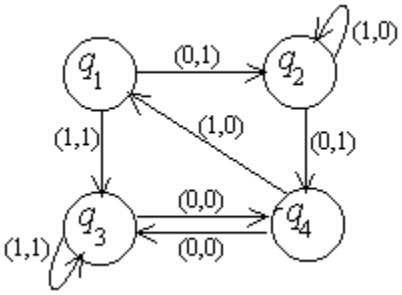
[0 1 0 0 1 1 0 1]

ж)



[0 1 0 1 1 0 0 1]

з)



[0 0 1 1 0 1 0 1]

3. Построить автомат, распознающий во входной последовательности подслова одного из видов и выдающий на выходе единицу в точности в те моменты, когда на входе появляется последняя буква одного из этих подслов:

а) 00, 101, 1011;
б) 10, 111, 0101;
в) 00, 011, 1001;
г) 10, 110, 1001;
д) 01, 100, 0100;
е) 11, 010, 1100;
ж) 01, 001, 0110;
з) 11, 100, 1101.

Рекомендуемая литература

Основная

1. Белов, Ю. А. Элементы теории множеств и математической логики: учеб. пособие / Ю. А. Белов. – Ярославль: ЯрГУ, 2002. – 60 с.

2. Белова, Л. Ю. Элементы теории множеств и математической логики. Теория и задачи: учеб. пособие / Л. Ю. Белова, В. А. Башкин, Ю. А. Белов. – Ярославль: ЯрГУ, 2005. – 79 с.

3. Задачи по дискретной математике. – Ярославль: ЯрГУ, 2005. – 30 с.

4. Кузнецов, О. П. Дискретная математика для инженера: учебник для вузов / О. П. Кузнецов. – 4-е изд., стереотип. – СПб.: Лань, 2005. – 400 с.

5. Ломазова, И. А. Дискретная математика. Математические основы обработки информации: учеб. пособие / И. А. Ломазова. – Ярославль: ЯрГУ, 2000. – 80 с.

6. Новиков, Ф. А. Дискретная математика для программистов / Ф. А. Новиков. – СПб.: Питер, 2001. – 304 с.

7. Спирина, М. С. Дискретная математика / М. С. Спирина. – М.: Академия, 2004. – 367 с.

Дополнительная

1. Асеев, Г. Г. Дискретная математика / Г. Г. Асеев и др. – Ростов н/Д.: Феникс; Харьков: Торсинг, 2003. – 142 с.

2. Белоусов, А. И. Дискретная математика / А. И. Белоусов. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2004. – 743 с.

3. Гаврилов, Г. П. Задачи и упражнения по дискретной математике / Г. П. Гаврилов, А. А. Сапоженко. – М.: Физматлит, 2006. – 400 с.

4. Евстигнеев, В. А. Теория графов: алгоритмы обработки деревьев / В. А. Евстигнеев. – Новосибирск, 1994. – 354 с.

5. Ерусалимский, Я. М. Дискретная математика: теория, задачи, приложения / Я. М. Ерусалимский. – М.: Вузовская книга, 2001. – 280 с.

6. Кристофидес, Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – М.: Мир, 1978. – 432 с.

7. Кудрявцев, В. Б. Введение в теорию автоматов / В. Б. Кудрявцев и др.. – М.: Наука, 1985. – 319 с.

8. Лекции по теории графов. – М.: Наука, 1990. – 382 с.

9. Микони, С. В. Элементы дискретной математики / С. В. Микони. – СПб.: Изд-во ПГУПС, 1999. – 124 с.

10. Москинова, Г. И. Дискретная математика / Г. И. Москинова. – М.: Логос, 2004. – 238 с.

11. Нечаев, В. И. Элементы криптографии / В. И. Нечаев. – М.: Высшая школа, 1999. – 109 с.

12. Оре, О. Графы и их применение / О. Оре. – Новокузнецк: Новокузнецкий физико-математический ин-т, 2000. – 173 с.

13. Плотников, А. Д. Дискретная математика / А. Д. Плотников. – М.: Новое знание, 2005. – 287 с.

14. Хаггарти, Р. Дискретная математика для программистов / Р. Хаггарти. – М.: Техносфера, 2005. – 399 с.

15. Яблонский, С. В. Введение в дискретную математику / С. В. Яблонский. – М.: Высшая школа, 2001. – 384 с.

Оглавление

Дискретная математика