Джандигулов А.Р.

ДИСКРЕТНАЯ МАТЕМАТИКА В ПРИМЕРАХ И ЗАДАНИЯХ

учебное пособие

Астана, 2017

УДК 519.95 (075.8)
ББК 518
Г 12

Джандигулов А.Р. Дискретная математика в примерах и заданиях: Часть 1.Учеб. пособие. — Астана, Изд-во ____ , 2017. – 90 с. ISBN

В пособие включены задания и упражнения по алгебре логики, теории алгоритмов, теории автоматов, теории графов, теории кодирования, комбинаторике, различным представлениям булевых функций, теории предикатов. Исходные данные для заданий и упражнений генерируются из исходной таблицы, что позволяет достичь уникальности задания для каждого студента. Каждый раздел также снабжен кратким теоретическим курсом, примерами выполнения заданий и контрольным тестом.

Для студентов и преподавателей университетов и технических вузов, в которых изучается дискретная математика. Библиогр. 8 назв.

Указания к пользованию учебным пособием.

Настоящее пособие предназначено для формирования самостоятельных и контрольных заданий студентов. Для достижения уникальности индивидуальных заданий исходные данные и/или вариант задания выбирается в зависимости от персональных данных студентов. При этом первоначально необходимо заполнить персональную таблицу Т3.

Необходимо определить и записать в персональную таблицу Т3 значения булевых переменных a_1 - a_{45} *(нули и единицы), исходя из следующих параметров: Первые буквы фамилии и имени переводятся в числа (номер в алфавите) в соответствии с таблицей Т1.*

Ф	первая буква фамилии	
И	первая буква имени	
N	порядковый номер по списку в журнале	

Таблица Т1. Таблица перевода букв алфавита в числа (номера).

Буква	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	Қ	К	Л	М	Н	О
Номер	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Буква	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Ц	Ч	Ш	Щ	Ы	І	Э	Ю	Я
Номер	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38

Таблица Т2. Таблица определения значений булевых переменных a_1 - a_{45}

Число	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Значение	0	1	1	0	0	0	1	1	1	0	0	0	1	0	1	0	0	1	1
Число	20	21	22	23	24	25	26	27	28	29	30	31	32	34	35	36	37	38	39
Значение	1	1	0	0	1	1	0	1	0	1	1	0	1	0	0	1	1	0	1

Таблица Т3. Персональная таблица студента

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

Алгоритм заполнения таблицы Т3. Значения a_1 - a_{45} выбираются из таблицы Т2 имеющей разделенной на 38 пронумерованных ячеек./

Выбор значений a_1 - a_{45} производится по следующему правилу:

a_1 - a_{15} – 15 чисел (нулей и единиц) подряд, начиная с позиции *Ф*;

$a_{15} – a_{30}$ – 15 чисел подряд по часовой стрелке, начиная с позиции *И*;

a_{31} - a_{45} – 15 чисел подряд по часовой стрелке, начиная с позиции *N*.

При этом, если какая-то переменная a_i определяется значением 38-й ячейки, то значения следующих переменных берутся начиная с 1-й ячейки.

Пример заполнения: Абдрахманов Мырзабек

Ф	первая буква фамилии	1
И	первая буква имени	17
N	порядковый номер по списку в журнале	11

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
0	1	1	0	0	0	1	1	1	0	0	0	1	0	1
a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}
0	1	1	1	1	0	0	1	1	0	1	0	1	1	0
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
0	0	1	0	1	0	0	1	1	1	1	0	0	1	1

В каждой из нижеследующих заданий определенным образом осуществляется выбор булевых переменных 0, 1 из заполненной таблицы Т3; из этих цифр составляются многозначные двоичные числа, которые затем используются в качестве исходных данных (в виде двоичных чисел или переводятся в десятичную систему). Правильное выполнение этих арифметических операций наряду с правильным исполнением инструкции в условии заданий, является неотъемлемой частью решения.

Содержание

Предисловие 7
1. СПОСОБЫ ЗАДАНИЯ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ И ИХ ПРОСТЕЙШИЕ СВОЙСТВА 8
  1.1. Функциональные соответствия и отношения. Операции над множествами. 8
  1.2. Функция алгебры логики 19
  1.3. Дизъюнктивные нормальные формы 23
  1.4. Функционально полные системы булевых функций. Критерий полноты системы булевых функций 26
2. ПРЕДИКАТЫ 33
3. ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ 38
4. ЭЛЕМЕНТЫ ТЕОРИИ КОДИРОВАНИЯ 48
  4.1. Алфавитное кодирование 48
  4.2. Оптимальное кодирование 51
  4.3. Коды с обнаружением и исправлением ошибок 53
5. ЛОГИЧЕСКИЕ СЕТИ И АВТОМАТЫ 62
  5.1. Логические сети 62
  5.2. Конечные автоматы 62
6. ЭЛЕМЕНТЫ КОМБИНАТОРИКИ 70
  6.1. Основные комбинаторные конфигурации 70
  6.2. Формулы пересчета числа комбинаторных конфигураций 71
7. АЛГОРИТМЫ И ВЫЧИСЛИМОСТЬ 76
  7.1 Рекурсивные функции 76
  7.2. Примитивно рекурсивные функции. 76
  7.3. Оператор минимизации. 77
  7.4. Машины Тьюринга 79
ГЛОССАРИЙ 83
Библиографический список: 89
ПРИЛОЖЕНИЯ 90

Предисловие

Дискретная математика занимается исследованиями структур и задач на конечных множествах. Общепринято деление математики на непрерывную и дискретную. Специфика задач дискретной математики в первую очередь предполагает отказ от основных понятий классической математики – предела и непрерывности. Данное обстоятельство не говорит о том, что это разделение сопоставляет эти части. Наоборот, они друг друга дополняют, например, для задач ДМ обычные средства классического анализа являются вспомогательными. Понятия и методы одной часто используются в другой. Один и тот же объект может рассматриваться с двух точек зрения и в зависимости от этого выбирается непрерывная или дискретная модель.

Сегодня ДМ является важным звеном математического образования. Умение проводить анализ, композицию и декомпозицию информационных комплексов и информационных процессов – необходимый элемент культуры выпускника технических специальностей.

Почти все разделы классического анализа так или иначе проецируются в дискретную математику, вместе с тем есть и специфические разделы, присущи исключительно дискретной математике. Среди них: теория функциональных систем, теория графов, теория автоматов, теория кодирования, теория алгоритмов и др.

Важным моментом усвоения математики и овладения её методами является самостоятельная работа учащегося. Правильная организация самостоятельной работы способствует более глубокому освоению курса и отработке приёмов решения задач.

Автор выражает благодарность коллегам из кафедры «Алгебра и геометрия» Евразийского национального университета имени Л. Гумилева за ценные замечания и поправки.

1. СПОСОБЫ ЗАДАНИЯ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ И ИХ ПРОСТЕЙШИЕ СВОЙСТВА

1.1. Функциональные соответствия и отношения. Операции над множествами.

Множество может быть задано различными способами, например, перечислением его элементов (список студентов, перечень товаров и т.д.), либо указанием характерного свойства, которым обладают элементы только данного множества (четные числа, студенты конкретной группы, товары, произведенные конкретной фирмой и т.п.).

На практике, определить, обладает ли тот или иной объект заданным свойством, а тем более найти все такие объекты, может стать сложной задачей. В частности, нахождение множества корней уравнения представляет собой решение уравнения. А вопрос о том, существует ли процедура распознавания тех или иных свойств математических объектов, относится к проблемам теории алгоритмов.

Операции объединения, пересечения, дополнения до универсального множества, а также разности двух множеств также определяют новые множества.

При решении ряда задач важно определить понятие **универсального множества *U*.**

Универсальное множество *U* — это множество, содержащее все объекты и все множества, рассматриваемые в решении конкретной задания.

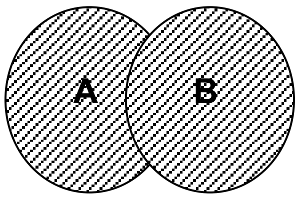
Как только в универсальном множестве *U* мы задаем, какое либо множество *A*, то мы разбиваем его на две не пересекающие части, собственно множество *A* и его **дополнение** $\overline{A} = U \setminus A$.

Разбиение множества *U* – система непустых подмножеств $\{A_\alpha\}$ множества *U* такая, что их объединение равно *U* (полнота разбиения), а все попарные пересечения – пусты (чистота разбиения). Сами A_α называются классами, или блоками разбиения.

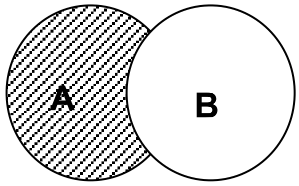
Пример. Система курсов данного факультета есть разбиение множества его студентов; система групп есть другое разбиение того же множества.

Графическое представление

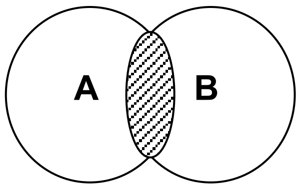
(Диаграммы Эйлера, Венна)



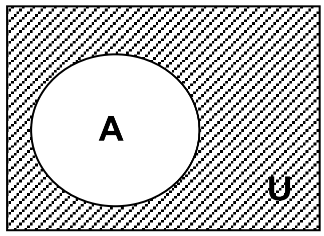
$A \cup B$



$A \setminus B$



$A \cap B$



$\overline{A}$

Конечная последовательность любых объектов, среди которых могут быть и повторяющиеся, называется **вектором**. Сами объекты называются **компонентами** вектора.

Декартовым (прямым) произведением множеств $A \times B$ называется множество M всех пар (a,b), таких, что $a \in A$ и $b \in B$

Если $A=B$, то такое произведение записывается так A^2

Таким же образом вводится операция прямого произведения большего числа множеств.

$A_1 \times A_2 \times \ldots \times A_n$ как множество векторов $(a_1, a_2, \ldots, a_n)$, $a_i \in A_i$.

И, если $A_1, A_1 \ldots, A_n$ одинаковы, $A_1 = A_2 = \ldots = A_n = A$ то получаем A^n

Если множества конечные, мощность (количество элементов) произведений $A \times B$ равна мощности произведений $|A \times B| = |A| \cdot |B|$

Рассмотрим другой пример задания числового множества порождающей процедурой:

(1) $4 \in M$;

(2) если $x \in M$, то $1/x \in M$;

(3) если $x \in M$, то $(1-x) \in M$.

Это множество конечно и состоит из 6 элементов, а именно

$M = \{4, \quad 1/4, \quad -3, \quad -1/3, \quad 3/4, \quad 4/3\}.$

Если же в правиле (3) заменить $(1-x)$ на $(3-x)$, то порождаемое множество будет бесконечным.

Декартовым (прямым) произведением множеств $A\, и\, B$ называется множество M всех пар ($a \cdot b$), таких, что $a \in A\ и\ b \in B$

Если A=B, то такое произведение называется A^2

Аналогично можно вывести операцию прямого произведения большего числа множеств.

$A_1 \times A_2 \times \ldots \times A_n$ как множество векторов $(a_1 \times a_2 \times \ldots a_n)$

Если в частности $A_1, A_1 \ldots A_n$ одинаковы $A_1 = A_2 = \ldots A_n$ то получаем A^n

Если множества конечные, мощность произведений $A \times B$ равна мощности произведений $|A \times B| = |A| \cdot |B|$

Суперпозиция функций

Соответствием G между множествами A и B называется подмножество $G \subseteq A \times B$. Если $(a,\ b) \in G$, то говорят, что b соответствует a. Множество всех $b \in B$, соответствующих элементу a, называется **образом** элемента a. Множество всех $a \in A$, которым соответствует элемент b, называется **прообразом** элемента b.

Множество пар $(b,\ a)$ таких, что $(a,\ b) \in G$ называется обратным по отношению к G и обозначается G^{-1}. Понятия образа и прообраза для G и G^{-1} взаимнообратны.

Функциональное соответствие $A \ \rightarrow\ B$, или **отображение** – соответствие, при котором образом любого элемента $a \in A$ является **единственный** элемент $b \in B$. Например, площадь геометрической фигуры или объем пространственного тела суть их отображения в множество неотрицательных чисел. Если A, B – числовые множества, то соответствие называется функцией (хотя иногда функциями называют и не числовые соответствия).

n **-местная функция (функция** n **переменных)** - функция типа $A_1 \times A_2 \times \ldots \times A_n \to B$ другая форма записи: $f(a_1, a_2, \ldots, a_n) = b$, где $a_i \in A_i$, $b \in B$. Сложение, вычитание, умножение, деление, возведение в степень и функции выбора максимума и минимума являются двуместными функциями.

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

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

В суперпозиции функций могут измениться как сами переменные, так и их число. Заметим также, что, выполняя подстановки, мы преобразовывали формулы, выражающие функции. **Формула** – это выражение, описывающее суперпозицию и содержащее функциональные знаки, символы независимых переменных (аргументов) и констант (параметров).

Задание 1.1.1. Перевести в десятичную систему двоичное число $a_1\, a_2\, a_3\, a_4$ и прибавить N [1]. Если полученное число больше 20, то нужно отнять 20. Полученное число i определяет номер варианта.

Определите порядок действий при вычислении значения суперпозиции элементарных функций и постройте соответствующую схему из функциональных элементов.

1. $f(X, Y, Z) = \max(X + 2 \cdot Z, 2 \cdot Y) / \sin(2 - Z)$
2. $f(X, Y, Z) = X^3 + \ln(3 \cdot Z^2 - 2 \cdot Y)$

[1] Здесь и далее порядковый номер по списку в журнале.

3. $f(X,Y,Z)=3^{X-Y}\cdot\sqrt{\cos^3(Z-3)}$

4. $f(X,Y,Z)=2^{X-Y}\cdot\ln(X+Y+Z)$

5. $f(X,Y,Z)=\min(X+Z,Y)\cdot\cos(X-Z)$

6. $f(X,Y,Z)=X^2+3\cdot Z^2-3\cdot Y$

7. $f(X,Y,Z)=e^{X+Y+Z}\cdot\sqrt{X+2\cdot Y}$

8. $f(X,Y,Z)=\sqrt{\cos(Z+X-3Y)}$

9. $f(X,Y,Z)=\cos(X+Z)/\min(2-Z,Y)$

10. $f(X,Y,Z)=X+Y\cdot\ln(Z-Y)$

11. $f(X,Y,Z)=5^{X-Y}\cdot\cos(X+Z)$

12. $f(X,Y,Z)=\sqrt{Z+2X-3Y^2}$

13. $f(X,Y,Z)=\max(X+2\cdot Y,Z)-\cos(2+Z)$

14. $f(X,Y,Z)=X^2+3\cdot Z^2-2\cdot Z\cdot Y$

15. $f(X,Y,Z)=\sqrt{X^2+Y^2+Z^2}$

16. $f(X,Y,Z)=XY+XZ+YZ$

17. $f(X,Y,Z)=X^2+Y^2+Z^2+2\cdot X\cdot Z+2\cdot X\cdot Y+2\cdot Y\cdot Z$

18. $f(X,Y,Z)=X^3+3\cdot Z^2-2\cdot Y^4$

19. $f(X,Y,Z)=\frac{1}{3}\cdot Z\cdot\sqrt{X^2+Y^2}$

20. $f(X,Y,Z)=X\Big/\sqrt{Y^2+Z^2}$

Пример выполнения задания 1.1.1. Определите порядок действий при вычислении значения суперпозиции элементарных функций и постройте соответствующую схему из функциональных элементов.

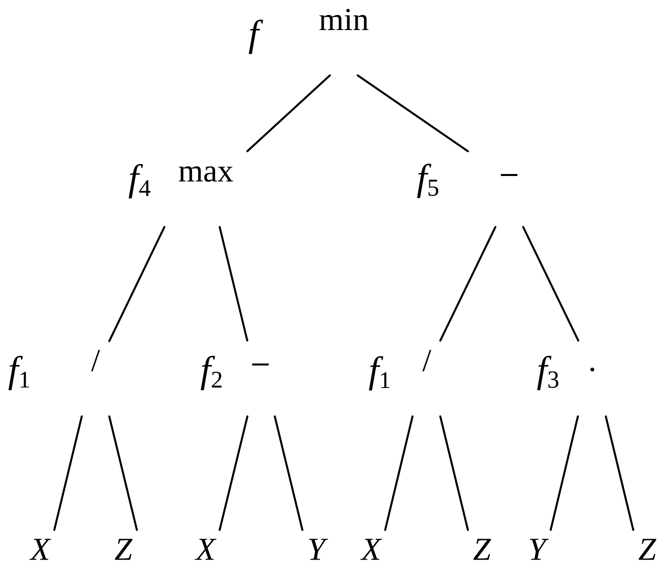
[2] $f(X,Y,Z)=\min(\max(X/Z,X-Y),X/Z-Y\cdot Z)$

Решение:

Составим иерархическую схему последовательности действий. Для этого определим сначала состав переменных и констант в формуле, которая задает функцию: переменные X, Y, Z констант нет.

[2] Пример взят из [1].

Определим внешнюю функцию и основные подформулы и построим иерархическую схему, определяющую порядок действий.



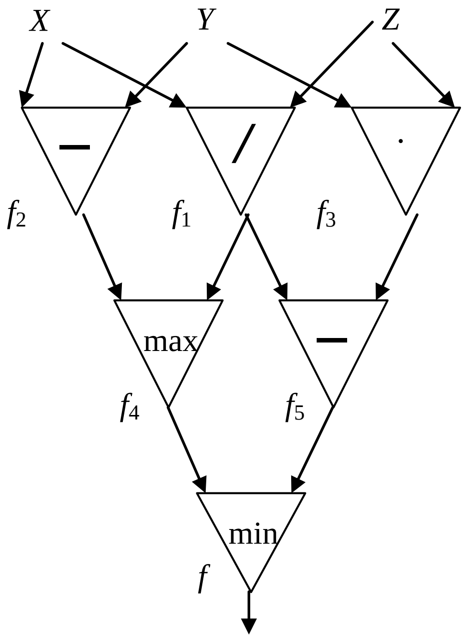
Определим совокупность используемых элементов с одним и двумя входами:

$f_1 = X/Z$; $f_2 = X - Y$; $f_3 = Y \cdot Z$

$f_4 = \max(f_1, f_2)$; $f_5 = f_1 - f_3$

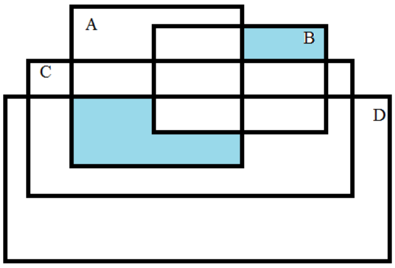
$f = \min(f_4, f_5)$

Осуществим необходимые соединения:

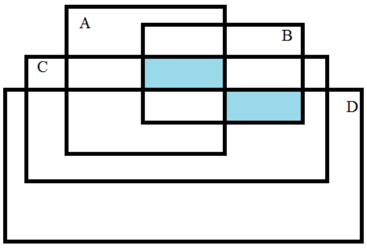


Задание 1.1.2. Перевести в десятичную систему двоичное число $a_{25}\, a_{26}\, a_{27}\, a_{28}$ и прибавить N. Если полученное число больше 20, то нужно отнять 20. Полученное число i определяет номер варианта.

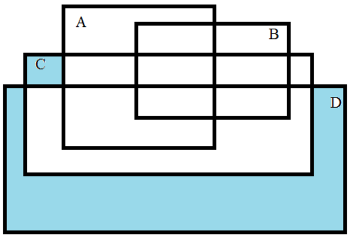
Выразить через множества A, B, C, D множество E, которому соответствует закрашенная область.



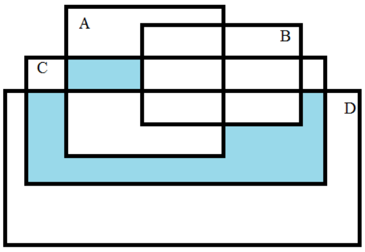
1.



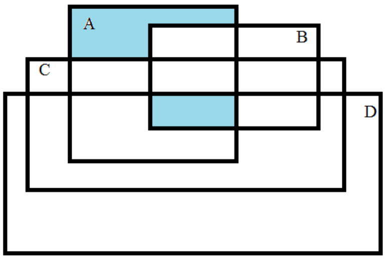
2.



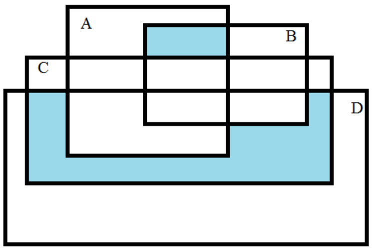
3.



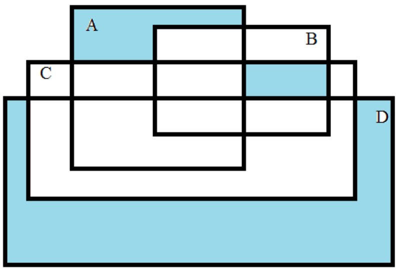
4.



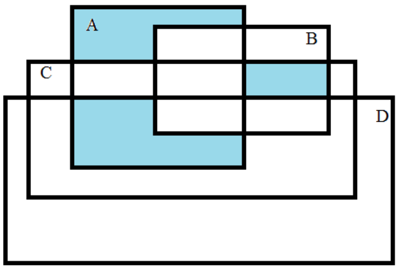
5.



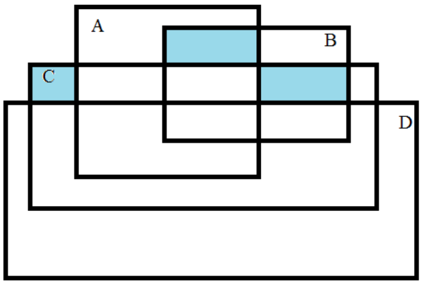
6.



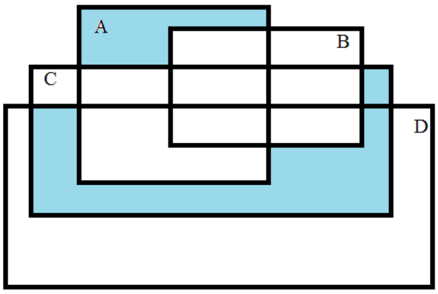
7.



8.

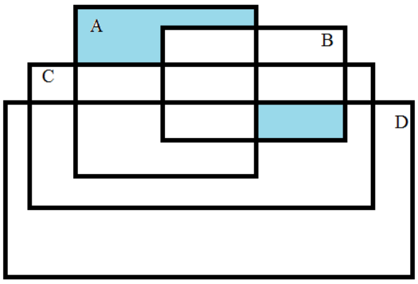


9.

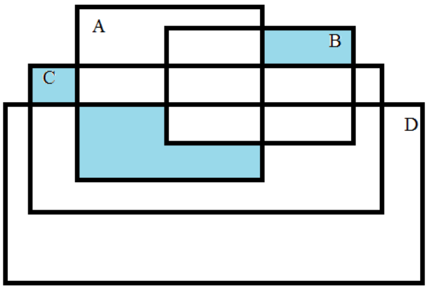


10.

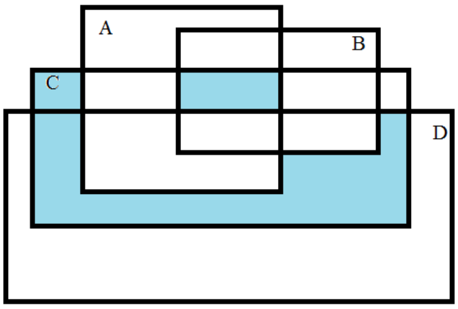
11.



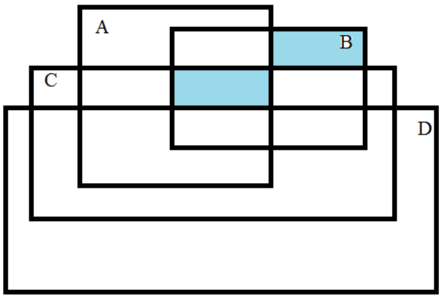
12.



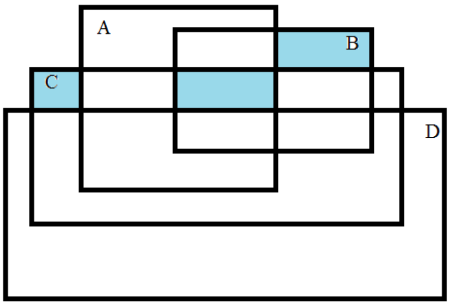
13.



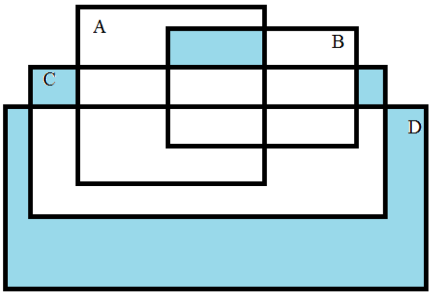
14.



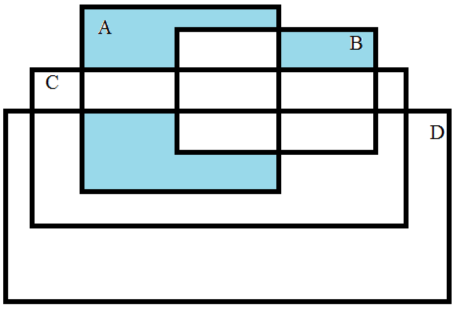
15.



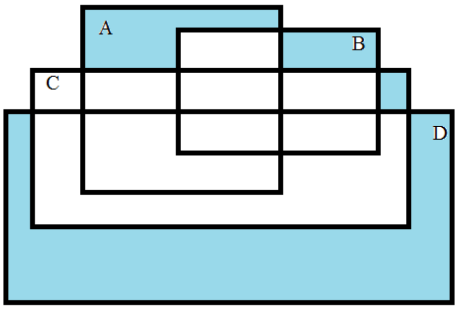
16.



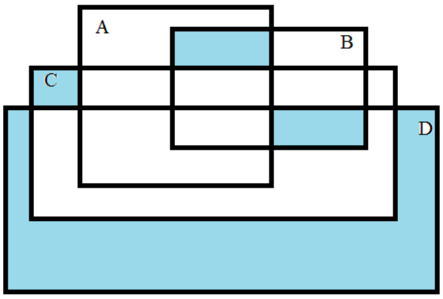
17.



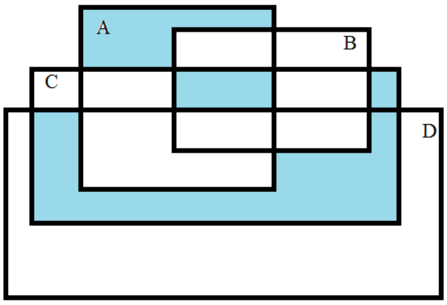
18.



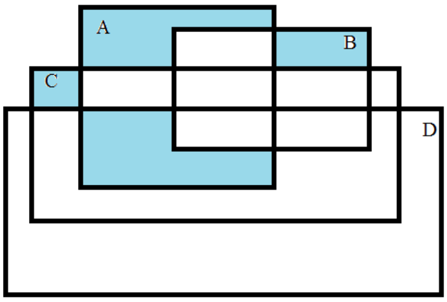
19.



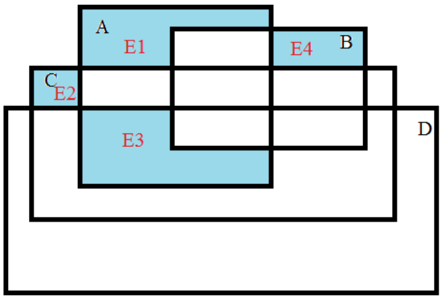
20.



Пример выполнения задания 1.1.2. Выразить через множества A, B, C, D множество E, которому соответствует закрашенная область.



Наша область E состоит из четырех непересекающихся частей (подмножеств), которые обозначим, соответственно E1,E2,E3 и E4.



Выразим каждое подмножество через множества A, B, C и D.

$$E1 = A \setminus (C \cup B)$$
$$E2 = C \setminus (D \cup A)$$
$$E3 = (D \cap A) \setminus B$$
$$E4 = B \setminus (C \cup A)$$

Тогда искомое множество E будет объединением E1,E2,E3 и E4.

$$E = (A \setminus (C \cup B)) \cup (C \setminus (D \cup A)) \cup ((D \cap A) \setminus B) \cup (B \setminus (C \cup A)).$$

Примечание. Данное представление неединственное, в силу наличие равносильных формул теории множеств. Например E1 можно представить в виде:

$$E1 = (A \setminus C) \setminus B$$

Задание 1.1.3. Перевести в десятичную систему двоичное число $a_2\ a_3\ a_4\ a_5$ и прибавить N. Если полученное число больше 20, то нужно отнять 20. Полученное число i определяет номер варианта.

Множество M задается следующей порождающей процедурой:

1) $10 \in M$;

2) если $x \in M$, то $3x \in M$;

3) если $x \in M$, то $(x-4) \in M$.

Выпишите из персональной таблицы последовательность чисел

$(a_7\ a_8\ a_9\ a_{10}\ a_{11}\ a_{12}\ a_{13}\ a_{14})$.

Прибавьте каждому числу число 2. Полученные числа буду определять последовательность операций порождающей процедуры множества M. Найдите результат этой последовательности операций.

Пример выполнения задания 1.1.3.

Результатом выполнения последовательности операций 2,2,3,3,3,2,3,2 будет:

$$10 \xrightarrow[2]{} 30 \xrightarrow[2]{} 90 \xrightarrow[3]{} 86 \xrightarrow[3]{} 82 \xrightarrow[3]{} 78 \xrightarrow[2]{} 234 \xrightarrow[3]{} 230 \xrightarrow[2]{} 690.$$

Ответ: 690.

Задание 1.1.4. Перевести в десятичную систему двоичное число $a_{12}\ a_{13}\ a_{14}\ a_{15}$ и прибавить N. Если полученное число больше 20, то нужно отнять 20. Полученное число i определяет номер варианта.

Пусть U={1, 2, 3, 4, 5, 6, 7, 8, 9}, A={1, 2, 3, 5, 7, 9}, B={2, 4, 6, 7}, C={1, 3, 5, 8}, D={1, 2, 4, 5, 8, 9}. Выразить через известные множества A, B, C, D и их дополнения следующие множества или доказать, что это сделать невозможно.

Варианты заданий.

1. {1; 3; 4; 5; 6; 7; 8};	2. {1; 3; 6};
3. { 1; 3; 4; 5; 6; 7; 8; 9};	4. {1; 2; 4; 5; 7; 9};
5. {4; 6; 8; 9};	6. {1; 3; 4; 6; 8; 9};
7. {1; 5; 6; 7};	8. {2; 3; 4; 6; 7; 8};
9. {1; 2; 3; 6; 7};	10. {2; 3; 5; 7; 8; 9};
11. {1; 2; 3; 5; 6; 7; 8; 9};	12. {3; 4; 6; 7; 9};
13. {1; 2; 6; 7; 9};	14. {1; 3; 4; 8; 9};
15. {1; 2; 7; 9};	16. {1; 2; 7; 8; 9};
17. {1; 2; 3; 4; 6; 7; 9};	18. {1; 3; 6; 7};
19. {1; 2; 6};	20. {2; 4; 7; 9};

Пример выполнения задания 1.1.4.

Выразить через известные множества A, B, C, D следующие множество E={3; 4; 7; 8; 9} или доказать, что это сделать невозможно.

Заметим, что A∪D={1;2; 3; 4;5;7; 8; 9}, то есть F= U \ (A∪D) = {6}.

Также B\A = {6,7}, тогда F1 = (B\A)\ F = **{7}.** Аналогично находим

F2 = (U \ C) \ B = **{9}.**

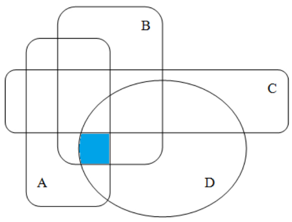
F3 = (D \ A) = **{4, 8}.**

F4 = (C \ D) = **{3}.**

Тогда E = F1 ∪ F2 ∪ F3 ∪ F4.

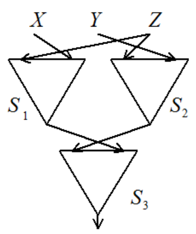
Контрольные тесты:

1. **Декартовым произведением** $A\times B$ **множеств** $A=\{3,4\}$ $B=\{4,5,6\}$ **является**
   A) $\{(4,3),(4,4),(5,3),(5,4),(6,3),(6,4)\}$
   B) $\{(3,4),(3,5),(3,6),(4,4),(4,5),(4,6)\}$
   C) $\{12,15,18,16,20,24\}$
   D) $\{12,15,16,18,20,24\}$
2. **Разность** $A\setminus B$ **множеств** $A=\{3,4,7\} B=\{4,5,6\}$ **является**
   A) $\{3,4\}$
   B) $\{3,7,\}$
   C) $\{-1,-1,1\}$
   D) $\{3,-5,-6,7\}$
3. **Какое множество изображено на картинке**
   A) $A\cap B\cap D\cap C$
   B) $A\cap B\cap D\cap \overline{C}$
   C) $A\cap \overline{B}\cap D\cap C$
   D) $A\cup B\cup D\cap C$



4. **Если** $V=\{1,2,3,...,10\}$**,** $A=\{n\in V: 1\le n\le 7\}$**,** $B=\{n\in V: n\ -\textit{кратное}\ 3\}$**, то найдите** $V\setminus(A\bigcup B)$
   A) $\{9,10\}$
   B) $\{3,6\}$
   C) $\{1,2,3,4,5,6\}$
   D) $\{8,10\}$
   E) $\{10\}$

5. **Схема из трех функциональных элементов** S_1, S_2, S_3**, где** $S_1=A-B$**,** $S_2=A/B$**,** $S_3=A\cdot B$**, реализует функцию**



   A) $(Y/Z)\cdot(X-Z)$
   B) $(Z/Y)\cdot(X-Z)$
   C) $(Z-X)\cdot(Y/Z)$
   D) $(Z/Y)\cdot(Z-X)$
   E) $(Z/Y)\cdot(X-Y)$

1.2. Функция алгебры логики

Логические переменные – это переменные, принимающие значения из двухэлементного множества B{0;1}.

Функция алгебры логики (логическая функция, булева функция) – это функция одной или нескольких переменных $Z= f\ (X_1,X_2,\ldots,\ X_n)$, где $f_S\ (X_1,X_2,\ldots,\ X_n)$, Z – логические переменные, т.е. и значения аргументов, и значение функции – ноль или единица. Тем самым, булева функция n переменных есть функция на B^n – множестве n -мерных векторов $\sigma = (\sigma_1, \sigma_2,\ldots, \sigma_n)$, компоненты которых равны 0 или 1: $\sigma_i \in B$.

Для задания логической функции нужно указать каким-либо образом, какое значение принимает функция на тех или иных наборах значений аргументов. Поэтому естественным является **табличное представление булевых функций.**

Следующая таблица содержит все булевы функции двух переменных.

Таблица 1

X	Y	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Функция	Формула	Название
g_0 =	0	константа 0
g_1 =	$X\&Y = X\cdot Y$	конъюнкция
g_2 =	$\neg(X\rightarrow Y) = X\&\neg Y$	отрицание имплкации
g_3 =	X	первая переменная
g_4 =	$\neg(Y\rightarrow X) = \neg X\&Y$	отрицание обратной импликации
g_5 =	Y	вторая переменная
g_6 =	$X\oplus Y = \neg\ X\&Y\vee\ X\&\neg Y$	сумма по модулю 2
g_7 =	$X\vee Y$	дизъюнкция
g_8 =	$\neg(X\vee Y) = \neg X\&\neg Y$	отрицание дизъюнкции
g_9 =	$X\sim Y = \neg\ X\&\neg Y\vee\ X\&Y$	эквивалентность
g_{10} =	$\neg Y$	отрицание второй переменной
g_{11} =	$Y\rightarrow X = X\vee\neg Y$	обратная импликация
g_{12} =	$\neg X$	отрицание первой переменной
g_{13} =	$X\rightarrow Y = \neg X\vee Y$	импликация
g_{14} =	$X\|Y = \neg X\vee\neg Y = \neg\ (X\&Y)$	штрих Шеффера
g_{15} =	1	константа 1

При решении задач необходимо научиться использовать эквивалентные (равносильные) формулы, т.е. формулы, представляющие одну и ту же функцию.

Примеры эквивалентных формул

№ п/п	Формула	№ п/п	Формула
1.	$X \vee Y = Y \vee X$	2.	$X \wedge Y = Y \wedge X$
3.	$(X \vee Y) \vee Z = X \vee (Y \vee Z)$	4.	$(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$
5.	$(X \vee Y) \wedge Z = (X \wedge \mathrm{Z}) \vee (Y \wedge Z)$	6.	$(X \wedge Y) \vee Z = (X \vee \mathrm{Z}) \wedge (Y \vee Z)$
7.	$X \vee X = X$	8.	$X \wedge X = X$
9.	$X \vee (X \wedge Y) = X$	10.	$X \wedge (X \vee Y) = X$
11.	$\neg(X \vee Y) = \neg X \wedge \neg Y$	12.	$\neg(X \wedge Y) = \neg X \vee \neg Y$
13.	$\neg X \vee X = 1$	14.	$\neg X \wedge X = 0$
15.	$0 \vee X = X$	16.	$0 \wedge X = 0$
17.	$1 \vee X = 1$	18.	$1 \wedge X = X$
19.	$\neg 0 = 1$ $(\neg 1 = 0)$	20.	$\neg(\neg X) = X$
21.	$X \wedge Y \vee X \wedge \neg Y = X$	22.	$X \vee \neg X \wedge Y = X \vee Y$

Конъюнкцию, дизъюнкцию и отрицание можно рассматривать как алгебраические операции над булевыми функциями. Свойства 1-2 выражают коммутативность, свойства 3-4 – ассоциативность операций $\vee$ и $\wedge$; свойства 5-6 – взаимную дистрибутивность этих операций, что дает возможность раскрывать скобки в формулах. Свойства 7-8 – устранение кратности; свойства 9-10 называют **правилами поглощения**; они являются следствиями остальных свойств, что показано ниже. Свойства 11-12 – законы де Моргана. Свойство 13 называют законом исключенного третьего; свойство 14 – законом противоречия. Свойства 15-19 характеризуют основные операции над переменной и константами 0 и 1. Свойство 20 – снятие двойного отрицания. Свойства 21-22 называются **правилами склеивания.**

Совокупность всех булевых функций P_2 с тремя данными операциями есть алгебра (P_2 ; $\wedge$;$\vee$; $\neg$). Она называется алгеброй булевых функций, алгеброй логики, а также **булевой алгеброй**.

Задание 1.2.1. В формуле $f(X,\ Y,\ Z) = ((X\ \boldsymbol{A}\ \bar{Y})\ \boldsymbol{B}\ (Y\ \boldsymbol{C}\ \bar{Z}))\ \boldsymbol{D}\ (\bar{X}\ \ \boldsymbol{E}\ Z)$ заменить двоичные числа $\boldsymbol{A} = a_{33}\, a_{34}$, $\boldsymbol{B} = a_{35}\, a_{36}$, $\boldsymbol{C} = a_{37}\, a_{38}$, $\boldsymbol{D} = a_{39}\, a_{40}$, $\boldsymbol{E} = a_{41}\, a_{42}$ символами логических операций:

00 – & (конъюнкция); 01 – $\vee$ (дизъюнкция);

10 – $\oplus$ (сумма по модулю 2); 11 – $\rightarrow$ (импликация);

Построить таблицу истинности для булевой функции $f(X,\ Y,\ Z)$. Произвести разложение функции по переменной X. Изобразить функцию на единичном кубе.

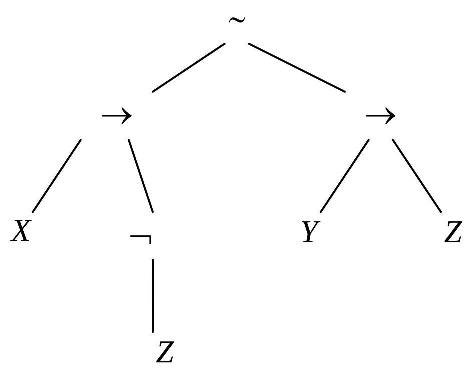
Пример выполнения задания 1.2.1.

Построить таблицу истинности для булевой функции, заданной логической формулой

$f\,(X,\, Y,\, Z) = (X \rightarrow \overline{Z}) \sim (Y \rightarrow Z)\,.$

Произвести разложение функции по переменной X.

Решение. Функция f зависит от трех переменных. Построим иерархическую схему, определяющую порядок действий:



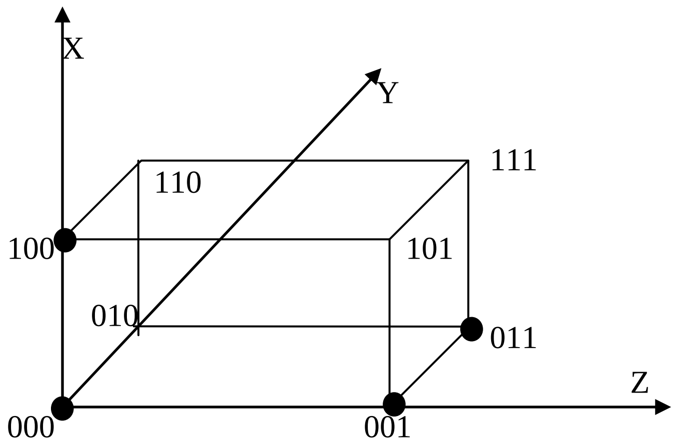
Построим таблицу наборов значений переменных X , Y, Z в порядке возрастания. Присоединим к ней справа столбцы для вычисления значений промежуточных функций, выражаемых подформулами, в последовательности, определяемой иерархической схемой. Последовательно для каждой строки таблицы вычислим значения промежуточных функций, начиная с внутренних, пользуясь таблицами истинности основных логических функций:

X	Y	Z	¬Z	X→¬Z	Y→Z	*f*
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	1	0

Произведем разложение функции по первой переменной *X*.

$$f(X, Y, Z) = \overline{X} \cdot [1 \quad 1 \quad 0 \quad 1]^T \vee X \cdot [1 \quad 0 \quad 0 \quad 0]^T = \overline{X} \cdot (Y \to Z) \vee X \cdot \overline{Y} \cdot \overline{Z}$$

Изобразим функцию на единичном кубе:



Задание 1.2.2. Перевести в десятичную систему двоичное число $a_{15}\, a_{16}\, a_{17}\, a_{18}$ и прибавить 1. Полученное число i определяет номер варианта.

Для заданной функции построить таблицу истинности и определить какие переменные «лишние» (несущественные) [2, с.33]

Варианты функций:

1. f(x, y, z) = (x V y) → (z ⊕ x)
2. f(x, y, z) = (x | y) → (x ⊕ z)
3. f(x, y, z) = xy ⊕ (y → z)
4. f(x, y, z) = xy ⊕ zx
5. f(x, y, z) = (x | y) → (x V z· x)
6. f(x, y, z) = (x → y) V (z ~ x)
7. f(x, y, z) = (x V z) ⊕ (x ↓ y)
8. f(x, y, z) = (z → y) ↓ (x ∧ y)
9. f(x, y, z) = (x | y) ~ (z → x)
10. f(x, y, z) = (x ~ y) V (x ⊕ z)
11. f(x, y, z) = (z | x) → (xy V z)
12. f(x, y, z) = (x ⊕ y) | (y → z)
13. f(x, y, z) = (x → y) ~ (x V (z | y))
14. f(x, y, z) = (x → $\overline{y}$) ⊕ z
15. f(x, y, z) = x & (y V z)
16. f(x, y, z) = (x↓ y) V z

1.3. Дизъюнктивные нормальные формы

Важным примером эквивалентности является **разложение булевой функции по переменной** – представление функции $Z = f(X_1, X_2, \ldots, X_n)$ в виде

$$f(X_1, X_2, \ldots, X_n) = \overline{X}_1 \cdot f(0, X_2, \ldots, X_n) \vee X_2 \cdot f(1, X_2, \ldots, X_n).$$

Справедливость этого тождества следует из того, что оба слагаемых, связанных знаком дизъюнкции, не могут одновременно равняться 1, так как один из сомножителей $\overline{X_1}$ или X_1 равняется 0. При подстановке в левую часть равенства константы 0 на место X_1 второе слагаемое в правой части обращается в 0, а при подстановке константы 1 – первое.

Функции (n-1) переменных $f(0, X_2, \ldots X_n)$ и $f(1, X_2, \ldots X_n)$ имеют в качестве столбцов значений соответственно верхнюю и нижнюю половины столбца значений $f(X_1, X_2, \ldots X_n)$.

В каждом из слагаемых функции от переменных $X_2, \ldots X_n$ могут быть таким же образом разложены по переменному X_2 и т.д.

Элементарная конъюнкция – конъюнкция нескольких переменных и их отрицаний, в которую каждая переменная входит не более одного раза.

Элементарная конъюнкция, соответствующая набору (σ_1, σ_2,…, σ_n) – конъюнкция $X_1^{\sigma_1} \cdot X_2^{\sigma_2} \cdot \cdots \cdot X_n^{\sigma_n}$. Для n -мерного набора конъюнкция содержит ровно n множителей с отрицаниями или без них. Как функция n переменных она принимает значение 1 только на наборе (σ_1, σ_2,…, σ_n).

Совершенная дизъюнктивная нормальная форма (СДНФ) – представление функции $Z = f(X_1, X_2, \ldots, X_n)$ в виде дизъюнкции всех элементарных конъюнкций, соответствующих наборам значений (σ_1, σ_2,…, σ_n), на которых Z =1:

$$f(X_1, X_2, \ldots, X_n) = \bigvee_{f(\sigma_1, \sigma_2, \ldots, \sigma_n)=1} X_1^{\sigma_1} \cdot X_2^{\sigma_2} \cdot \cdots \cdot X_n^{\sigma_n}$$

СДНФ содержит ровно столько n -членных элементарных конъюнкций, сколько единиц в столбце значений функции $Z = f(X_1, X_2, \ldots, X_n)$. На каждом из 2^n наборов либо все логические слагаемые СДНФ обращаются в 0 (если функция f на этом наборе равна 0), либо ровно одна конъюнкция обращается в 1 (если f равна 1). Не имеет СДНФ единственная функция – тождественно равная нулю (константа 0).

Отсюда – простой способ выражения любой функции (кроме константы 0), заданной таблично, в виде СДНФ. По таблице значений составляются соответствующие элементарные конъюнкции и связываются знаком дизъюнкции.

Задание 1.3.1. Построить СДНФ для булевой функции, заданной формулой

$$f(X, Y, Z, T) = (X^{a25}\, Y^{a26} \vee Z^{a27}\, T^{a28}) \rightarrow ((X^{a29} \vee T^{a30}) \oplus X^{a31}\, Y^{a32}\, Z^{a33})$$

с параметрами $a_{25} - a_{33}$, с использованием, где необходимо, знака отрицания в соответствии со значениями $a_{25} - a_{33}$.

Произвести разложение функции f по переменным X,Y. Упростить формулу, используя основные свойства булевых функций и приемы поглощения и склеивания.

Пример выполнения задания 1.3.1.

Для решения задания 1.3.1 необходимо сначала поострить таблицу истинности по аналогии с заданием 1.2.1 с тем отличием, что здесь уже 4 переменных. Предположим, что столбец значений функции получился следующим *f(X,Y,Z,T)* = (1011 0001 0011 1001).

1. Для каждой строки таблицы истинности, в которой значение функции f равно 1, составить соответствующую ей элементарную конъюнкцию четырех переменных

№ п/п	X	Y	Z	T	*f (X,Y,Z,T)*	Элементарная конъюнкция
0	0	0	0	0	1	$\overline{X} \cdot \overline{Y} \cdot \overline{Z} \cdot \overline{T}$
1	0	0	0	1	0	
2	0	0	1	0	1	$\overline{X} \cdot \overline{Y} \cdot Z \cdot \overline{T}$
3	0	0	1	1	1	$\overline{X} \cdot \overline{Y} \cdot Z \cdot T$
4	0	1	0	0	0	
5	0	1	0	1	0	

6	0	1	1	0	0	
7	0	1	1	1	1	$\overline{X}\cdot Y\cdot Z\cdot T$
8	1	0	0	0	0	
9	1	0	0	1	0	
10	1	0	1	0	1	$X\cdot\overline{Y}\cdot Z\cdot\overline{T}$
11	1	0	1	1	1	$X\cdot\overline{Y}\cdot Z\cdot T$
12	1	1	0	0	1	$X\cdot Y\cdot\overline{Z}\cdot\overline{T}$
13	1	1	0	1	0	
14	1	1	1	0	0	
15	1	1	1	1	0	

2. В столбце значений функции 7 единиц. Следовательно, СДНФ есть дизъюнкция 7 конъюнкций.

$$f=\underset{(1)}{\overline{X}\cdot\overline{Y}\cdot\overline{Z}\cdot\overline{T}}\vee\underset{(2)}{\overline{X}\cdot\overline{Y}\cdot Z\cdot\overline{T}}\vee\underset{(3)}{\overline{X}\cdot\overline{Y}\cdot Z\cdot T}\vee\underset{(4)}{\overline{X}\cdot Y\cdot Z\cdot T}\vee\underset{(5)}{X\cdot\overline{Y}\cdot Z\cdot\overline{T}}\vee\underset{(6)}{X\cdot\overline{Y}\cdot Z\cdot T}\vee\underset{(7)}{X\cdot Y\cdot\overline{Z}\cdot\overline{T}}$$

Дизъюнктивные члены пронумерованы – эти обозначения будут в дальнейшем использованы.

3. Составим разложение функции по двум первым переменным. Общий вид разложения функции 4 переменных по двум первым переменным:

$$f(X,Y,Z,T)=\overline{X}\cdot\overline{Y}\cdot f(0,0,Z,T)\vee X\cdot\overline{Y}\cdot f(1,0,Z,T)\vee\overline{X}\cdot Y\cdot f(0,1,Z,T)\vee X\cdot Y\cdot f(1,1,Z,T).$$

Пользуясь таблицей истинности функции *f*, запишем разложение по *X*,*Y*, задавая функции переменных *Z*,*T* столбцами значений:

$$f(X,Y,Z,T)=\overline{X}\cdot\overline{Y}\cdot[1,0,1,1]^T\vee X\cdot\overline{Y}\cdot[0,0,0,1]^T\vee\overline{X}\cdot Y\cdot[0,0,1,1]^T\vee X\cdot Y\cdot[1,0,0,0]^T.$$

Функции оставшихся переменных *Z*,*T*, заданные 4 столбцами значений, можно записать формулами, используя таблицу 1:

$$f(X,Y,Z,T)=\overline{X}\cdot\overline{Y}\cdot(T\to Z)\vee X\cdot\overline{Y}\cdot Z\cdot T\vee\overline{X}\cdot Y\cdot Z\vee X\cdot Y\cdot(\overline{Z}\vee\overline{T}).$$

4. Упростим формулу, используя основные эквивалентности и приемы поглощения и склеивания. Будем исходить из СДНФ. Объединим некоторые пары элементарных конъюнкций и вынесем за скобки общие множители:

а) (3) и (4): $\overline{X}\cdot\overline{Y}\cdot Z\cdot T\vee\overline{X}\cdot Y\cdot Z\cdot T=\overline{X}\cdot(Y\vee\overline{Y})\cdot Z\cdot T=\overline{X}\cdot Z\cdot T.$

Обозначим полученную конъюнкцию (8)

б) (5) и (6): $X\cdot\overline{Y}\cdot Z\cdot\overline{T}\vee X\cdot\overline{Y}\cdot Z\cdot T=X\cdot\overline{Y}\cdot Z\cdot(\overline{T}\vee T)=X\cdot\overline{Y}\cdot Z$.

в) (1) и (7): $\overline{X}\cdot\overline{Y}\cdot\overline{Z}\cdot\overline{T}\vee X\cdot Y\cdot\overline{Z}\cdot\overline{T}=(X\cdot Y\vee\overline{X}\cdot\overline{Y})\cdot\overline{Z}\cdot\overline{T}=(X\sim Y)\cdot\overline{Z}\cdot\overline{T}$.

г) (2) и (8): $\overline{X}\cdot\overline{Y}\cdot Z\cdot\overline{T}\vee\overline{X}\cdot Z\cdot T=\overline{X}\cdot Z\cdot\left(\overline{Y}\cdot\overline{T}\vee T\right)=$[применяя склеивание]=
$\overline{X}\cdot Z\cdot\left(\overline{Y}\vee T\right)$

В результате получаем формулу

$$f=\overline{X}\cdot Z\cdot\left(\overline{Y}\vee T\right)\vee X\cdot\overline{Y}\cdot Z\vee\left(X\sim Y\right)\cdot\overline{Z}\cdot\overline{T}.$$

1.4. Функционально полные системы булевых функций. Критерий полноты системы булевых функций

Выше показано, что любая функция может быть выражена в виде ДНФ, т.е. формулой, использующей функциональные знаки $\{\cdot,\ \vee,\ \neg\}$ и символы переменных. Таким образом, система функций $\{\cdot,\ \vee,\ \neg\}$ является функциональной полной.

Еще один интересный пример дает система функций

$$\{1, X\oplus Y, X\&Y\}. \qquad (*)$$

Выше показано, что $\overline{X}=X\oplus 1$. Используя это равенство и закон де Моргана (свойство 11), выразим дизъюнкцию через функции системы (*).

$$X\vee Y=\overline{\overline{X}\cdot\overline{Y}}=(X\oplus 1)\cdot(Y\oplus 1)\oplus 1=XY\oplus X\oplus Y\oplus 1\oplus 1=XY\oplus X\oplus Y$$

Поэтому в любой ДНФ можно выразить дизъюнкцию и отрицание через функции системы (*).

Таким образом, любую функцию можно представить в виде формулы, представляющей сумму по модулю 2 некоторых конъюнкций переменных (в каждой конъюнкции переменные не повторяются, но, в отличие от элементарных конъюнкций – без отрицаний) и, быть может, константы 1.

Полученная формула, порожденная логическими константами 0 и 1 и функциями и называется **многочленом Жегалкина**. Можно показать, что имеет место единственность (с точностью до перестановки слагаемых в сумме и сомножителей в конъюнкциях) представления булевой функции многочленом Жегалкина.

Предполные классы

1) **Класс T_0** – класс функций, сохраняющих 0, т.е. функций, для которых f (0,0,…,0) = 0 . Замкнутость класса T_0 очевидна: если в функцию $Z=f$ $(X_1, X_2, \ldots, X_m)$ вместо некоторых переменных подставить функции, принадлежащие T_0, то на нулевом наборе аргументов все они имеют значение 0, и для внешней функции f набор ее переменных будет также нулевым, откуда $Z = 0$.

2) **Класс T_1** – класс функций, сохраняющих 1, т.е. функций, для которых f (1, 1,…,1) =1. Замкнутость T_1 устанавливается аналогично предыдущему.

Примерами функций, принадлежащих классам T_0 и T_1, служат функции X&Y и $X \vee Y$; отрицание $\neg X$ не принадлежит ни T_0 ни T_1 ; функция $X \oplus Y$ принадлежит T_0 , но не принадлежит T_1 ; импликация $X \rightarrow Y$ напротив, не принадлежит T_0, но принадлежит T_1 .

3) Для определения следующего класса введем понятие двойственности. **Двойственная функция** для функции $Z = f$ $(X_1, X_2, \ldots, X_m)$ – функция $Z^* = \overline{f(\overline{X_1}, \overline{X_2}, \ldots, \overline{X_n})}$. Если на наборе $\tilde{\sigma} = (\sigma_1 \sigma_2, \ldots, \sigma_n)$ функция f принимает значение α , то двойственная ей функция f^* на противоположном наборе $\overline{\tilde{\sigma}} = (\overline{\sigma}_1, \overline{\sigma}_2, \ldots, \overline{\sigma}_n)$ принимает противоположное значение $\overline{\alpha}$. Например, Функции дизъюнкции и конъюнкции являются двойственными друг к другу.

Для булевых функций справедлив **принцип двойственности** – если в формуле F , представляющей функцию f , все знаки функций заменить соответственно на знаки двойственных функций, то полученная формула F^* будет представлять функцию f^* , двойственную f .

Класс S самодвойственных функций – то есть функций таких, что $f(X_1, X_2, \ldots, X_n) = f^*(X_1, X_2, \ldots, X_n)$.

4) Подмножеством множества многочленов является **класс L линейных функций** – функции вида $X_{i1} \oplus X_{i2} \oplus \ldots \oplus X_{in} \oplus \sigma$. Здесь $X_{i_1}, X_{i_2}, X_{i_n}$ – переменные, σ – булева константа (0 или 1).

5) Введем отношение частичного порядка для булевых векторов: X≤Y (где X=$(\alpha_1,\alpha_2,...,\alpha_n)$, Y=$(\beta\ \beta_2,...,\beta_n)$– если $\alpha_i \le \beta_i$ для I=1,2,…,n. Заметим, что для булевых переменных строгое неравенство $\alpha < \beta$ означает, что $\alpha = 0, \beta = 1$, поскольку других возможностей нет. Равенство $\alpha = \beta$ добавляет варианты $\alpha = \beta = 0$ и $\alpha = \beta = 1$. Поэтому неравенству $\alpha \le \beta$ удовлетворяют 3 пары (α, β): (0, 0), (0, 1), (1, 1) и не удовлетворяет только пара (1, 0). Можно заметить, что $(\alpha \le \beta)$ эквивалентно $(\alpha \to \beta)$.

Класс М монотонных функций – это класс функций таких, что если X≤Y , то f(X)≤f(Y) , т.е. функция на большем наборе принимает не меньшее значение.

Конъюнкция и дизъюнкция являются монотонными.

Критерий полноты системы булевых функций (теорема Поста) – система Σ полна в том и только в том случае, если для каждого из классов T_0 , T_1 ,S, L, M в системе существует функция, не принадлежащая этому классу; иначе говоря, система Σ полна, если выполнены 5 условий:

1) в системе Σ есть $f_1 \notin T_0$,

2) в системе Σ есть $f_2, \notin T_1$,

3) в системе Σ есть $f_3, \notin S$,

4) в системе Σ есть $f_4 \notin L$,

5) в системе Σ есть $f_5 \notin M$.

Функции – не обязательно различные.

Задание 1.4.1. Представить многочленом Жегалкина функцию, заданную ДНФ:

$$f(X, Y, Z) = X^{a1} Y^{a2} Z^{a3} \vee Y^{a4} Z^{a5}$$

По полученной формуле построить схему из функциональных элементов, используя в качестве элементов конъюнкторы и элементарные сумматоры.

Пример выполнения задания 1.4.1

Пусть $a_1\, a_2\, a_3$ = 001; $a_4\, a_5$ = 10.

$f(X, Y, Z) = X^{0}Y^{0}Z^{1} \vee Y^{1}Z^{0} = \overline{X}\,\overline{Y}Z \vee Y\overline{Z} = (\overline{X}\,\overline{Y}Z)\ \&\ (Y\overline{Z}) \oplus \overline{X}\,\overline{Y}Z \oplus Y\overline{Z}$ = [первое слагаемое равно 0, поскольку перемножаются Z и $\overline{Z}$] = $(X \oplus 1)\,(Y \oplus 1)\,Z \oplus Y\,(Z \oplus 1) = (X\,Y\,Z \oplus X\,Z \oplus \boldsymbol{Y\,Z} \oplus Z) \oplus \boldsymbol{Y\,Z} \oplus Y =$

$= X\,Y\,Z \oplus X\,Z \oplus Z \oplus Y$

Задание 1.4.2. Перевести в десятичную систему двоичное число $a_{11}\, a_{12}\, a_{13}\, a_{14}$ и прибавить 1. Полученное число i определяет номер варианта. Представить многочленом Жегалкина заданную функцию

1. $f(x, y,z) = ((x \vee y) \downarrow (y \to \bar{x}))((x \oplus \bar{y}) \to (z \,|\, (\overline{x \vee y})))$.
2. $f(x, y,z) = (x \downarrow (\bar{y} \oplus (y \to \bar{x}))(x \vee (\bar{y} \,|\, \bar{z} \oplus \overline{xy})))$.
3. $f(x, y,z) = (x \to (y \downarrow z)) \leftrightarrow ((x \to y) \downarrow (x \to z))$.
4. $f(x, y,z) = \overline{(z \to x) \leftrightarrow (y \,|\, x)}$.
5. $f(x, y,z) = ((x \wedge y) \leftrightarrow ((y \downarrow \bar{x})(x \to \bar{y}))) \,|\, (z \oplus (\overline{x \vee y}))$.
6. $f(x, y,z) = (x \leftrightarrow (y \oplus z)) \vee ((x \leftrightarrow y) \oplus (x \leftrightarrow z))$.
7. $f(x, y,z) = (\bar{x} \vee y) \to \overline{(z \leftrightarrow x)}$.
8. $f(x, y,z) = ((x \oplus y) \,|\, (y \downarrow \bar{x})) \downarrow ((x \leftrightarrow \bar{y}) \to (z \vee (\overline{x \vee y})))$.
9. $f(x, y,z) = (x \downarrow (y \oplus z)) \,|\, ((x \downarrow y) \oplus (x \downarrow z))$.
10. $f(x, y,z) = \overline{((x \downarrow y) \to z) \leftrightarrow x}$.
11. $f(x, y,z) = x \oplus (\bar{y} \to (y \leftrightarrow \bar{x})) \wedge (x \downarrow (\bar{y} \,|\, (z \vee \overline{xy})))$
12. $f(x, y,z) = (x \vee (y \oplus z)) \to ((x \vee y) \oplus (x \vee z))$
13. $f(x, y,z) = \overline{(x \,|\, y) \oplus (\bar{z} \to y)}$.
14. $f(x, y,z) = (x|y) \to ((y \oplus \bar{x})(x \wedge \bar{y}) \to (z \leftrightarrow \overline{x \downarrow y}))$.
15. $f(x, y,z) = (x \to (y|z)) \oplus ((x \to y) \,|\, (x \to z))$.
16. $f(x, y,z) = (\bar{x} \vee y) \to \overline{(z \leftrightarrow x)}$.

Задание 1.4.3. Перевести в десятичную систему двоичное число $a_{25}\,a_{26}\,a_{27}$ и прибавить N. Если полученное число больше 18, то нужно отнять 18. Полученное число i определяет номер варианта.

Перевести в десятичную систему двоичное число и прибавить N. Полученное число определяет номер варианта.

а) определить принадлежность функций f1 и f2 к предполным классам.

б) проверить полноту системы функций { f_1 ,f_2 } по теореме Поста.

№	Варианты заданий	№	Варианты заданий	№	Варианты заданий
1	f_1=(0 1 0 1 1 0 1 1) $f_2=(x \vee y)\to(z \oplus x)$	7	f_1=(0 1 1 0 1 1 1 0) $f_2=x \oplus y \vee z$	13	f_1=(0 1 1 1 1 1 0 1) $f_2=(x \vee y)\to xyz$
2	f_1=(0 1 0 1 1 0 0 1) $f_2=(x \to y)\ V\ \bar{z}$	8	f_1=(0 1 1 0 0 0 1 1) $f_2=(x \oplus y)\to(y \vee x)$	14	f_1=(1 0 0 1 0 1 0 0) $f_2=(x\ V\ y)\to(z \oplus x)$
3	f_1=(0 1 0 1 0 1 1 0) $f_2=x\ \&\ (\bar{y} \vee z)$	9	f_1=(0 1 1 0 1 1 0 1) $f_2=(x \oplus z)\to(x\vee y)$	15	f_1=(1 0 0 1 1 0 0 1) $f_2=(x \to y)\ \&\ z$
4	f_1=(0 1 0 1 0 0 1 1) $f_2=(x \to \bar{y}) \oplus z$	10	f_1=(0 1 1 1 0 1 0 0) $f_2=x\&y \oplus z\&x$	16	f_1=(1 0 0 1 0 1 1 0) $f_2=x \vee yz$
5	F_1=(0 1 0 1 1 1 0 1) $f_2=x\,y \vee z\,(x \oplus \bar{y})$	11	f_1=(0 1 1 1 1 0 0 1) $f_2=\bar{x}\,yz\to(z\vee x)$	17	f_1=(1 0 0 1 0 0 1 1) $f_2=(x \to y) \oplus z$
6	f_1=(0 1 1 0 0 1 0 0) $f_2=x\,y \oplus (y \to z)$	12	f_1=(0 1 1 1 0 1 1 0) $f_2=(x\to y)\oplus(z\oplus y)$	18	f_1=(1 0 0 1 1 1 0 1) $f_2=xy \vee z(x \oplus y)$

Пример выполнения задания 1.4.3

Для выполнения поставленного задания удобнее всего работать с таблицей истинности. Поэтому мы разберем вопрос принадлежности к предполным классам одной функции, заданной столбцом значений. Построив таблицу истинности для второй функции, можно будет выяснить полноту системы в целом.

Пусть функция задана столбцом значений $f(x,y,z)$ = (0 0 1 1 1 0 1 0)

				Построение многочлена Жегалкина							*Нахождение двойственной функции*	
x	*y*	*z*	*f(x,y,z)*	1	2	3	4	5	6	7	$f(\bar{x},\bar{y},\bar{z})$	*f*(x,y,z)*
0	0	0	**0**	0	1	0	1	1	1	0	0	1
0	0	1	**0**	1	1	1	0	0	1		1	0
0	1	0	**1**	0	0	1	0	1			0	1
0	1	1	**1**	0	1	1	1				1	0
1	0	0	**1**	1	0	0					1	0
1	0	1	**0**	1	0						1	0
1	1	0	**1**	1							0	1
1	1	1	**0**								0	1
			1	*z*	*y*	*yz*	*x*	*xz*	*xy*	*xyz*		
коэффициенты			0	0	1	0	1	1	1	0		
		f=			*y⊕*		*x⊕*	*xz⊕*	*xy*			

Строка коэффициентов вычисляется по такой же схеме, как и треугольник Паскаля, только берется сумма по модулю 2.

Предполные классы	принадлежность к классам (+/-)	Пояснение
T_0	+	*f(0,0,0) = 0, первая строка таблицы*
T_1	-	*f(1,1,1)= 0 ≠ 1, последняя строка*
S	-	*f(x,y,z) ≠ f*(x,y,z) соответствующие столбцы отличаются*
L	-	*в многочлене Жегалкина есть конъюнкции (xz и xy)*
M	-	*В столбце значений после 1 идет снова ноль.*

Построив аналогичную таблицу для второй функции мы смотрим, есть ли строка, т.е. предполный класс, к которой принадлежат обе функции. Если такая есть, то система неполная, и полная в противном случае.

Задание 1.4.4. Перевести в десятичную систему двоичное число $a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15}$ и прибавить 1. Полученное число *i* определяет номер варианта. Доопределить функции *f(x,y,z), g(x,y,z), h(x,y,z)* так, чтобы

$$f \in M, g \in L, h \in S.$$

Если построение какой-либо функции невозможно, докажите это. Выясните вопрос о принадлежности построенных функций к классам T_0, T_1.

№	*f*	*g*	*h*
1	x10x 1xxx	x10x x0x0	x0xx 11x1
2	xxx0 1xx1	0xxx 110x	11xx 10xx
3	xxx0 x10x	xxx0 0x10	x1xx 01x0
4	x0xx xx11	01x0 x1xx	101x 1xxx
5	xxx0 x01x	01x1 xxx1	xx10 xx01
6	xx1x x0xx	x01x x1x1	x1x0 x1x0
7	x00x 1xxx	1xxx 001x	x00x 1xx1
8	x1x1 xx00	xxx1 1x01	x1xx 10x0
9	x01x x0xx	10x1 x0xx	0xxx 101x
10	xxx0 1x1x	1x01 xxx0	1xx1 x00x
11	x1xx xx01	1x1x xx00	1x10 xx1x
12	0xx0 1xxx	xx00 1x0x	xx10 xx00
13	0x1x x0xx	xx10 1x1x	x10x 0xx1
14	01xx xx0x	x00x 1x1x	11x1 x0xx
15	xxx0 1xx1	0xxx 001x	x010 xxx1

Контрольный тест

1. **Булева функция $X \to 0$ тождественно равна функции**

A) X;

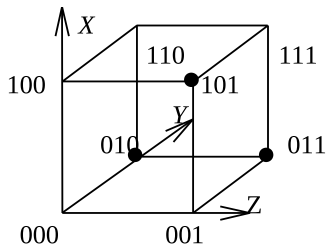
B) 0

C) $\neg X$

D) 1

E) нет правильного ответа

2. **Функция, заданная на трехмерном единичном кубе E^3, имеет СДНФ**



A) $\bar{X}YZ \vee X\bar{Y}Z \vee XY\bar{Z}$

B) $\bar{X}Y\bar{Z} \vee \bar{X}YZ \vee X\bar{Y}Z$

C) $\bar{X}Y\bar{Z} \vee X\bar{Y}Z \vee X\bar{Y}\bar{Z}$

D) $\bar{X}Y\bar{Z} \vee X\bar{Y}Z \vee XY\bar{Z}$

E) $\bar{X}\bar{Y}Z \vee \bar{X}YZ \vee X\bar{Y}Z$

3. **Булевы функции $f(X,Y)$ и $g(X,Y)$ задаются столбцами значений $f = [0111]^T$ и $g = [0101]^T$. Столбцом значений функции $(f \& \neg g)$ является**

A) $[0100]^T$

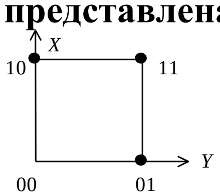
B) $[1001]^T$

C) $[1101]^T$

D) $[0101]^T$

E) $[1100]^T$

4. **Функция, заданная на двумерном единичном кубе E^2, может быть представлена формулой**



A) $X \& Y$

B) $\neg X \vee \neg Y$

C) $X \oplus Y \oplus 1$

D) $X \sim \neg Y$

E) $X \geq Y$

5. **Функция, заданная СДНФ $f = \bar{X}Y\bar{Z} \vee X\bar{Y}Z \vee XY\bar{Z}$, имеет столбец значений**

A) $[00100110]^T$

B) $[00101100]^T$

C) $[00101010]^T$

D) $[00010110]^T$

E) $[00011110]^T$

2. ПРЕДИКАТЫ

Предикат – функция (отображение) P типа: $M_1 \times M_2 \times ... \times M_n \rightarrow B$, где $B=\{0,1\}$, M_i – **произвольные** множества, т.е. функция P, сопоставляющая вектору $(m_1, m_2, ..., m_n)$ значение 0 или 1. Прямое произведение $M_1 \times M_2 \times ... \times M_n$ будем называть областью определения (**предметной областью)** предиката $P(m_1, m_2, ..., m_n)$; $m_1, m_2, ..., m_n$ – **предметными переменными**, P – предикатным символом; число n называется **местностью** предиката.

Область истинности предиката P – подмножество $I_p \subseteq M_1 \times M_2 \times ... \times M_n$ предметной области предиката P, на элементах которого значения предиката равны 1.

Над предикатами на M можно производить логические операции и получать новые предикаты.

Кроме того, над предикатами определяются две специфические операции, называемые **навешиванием кванторов**, которые превращают одноместный предикат P(X) в 0-местный:

квантор всеобщности – высказывание: “для всех X выполнено P(X) ”, обозначение$\forall X{:}P(X)$;

квантор существования – высказывание: “существуетX, для которого выполненоP(X) ”, обозначение $\exists X{:}\ P(X)$.

Процедура навешивания кванторов на предикат применима не только к одноместным, но и к предикатам большей местности: в этом случае говорят также, что квантор навешивается на определенную переменную данного предиката.

Если, например, навесить квантор по переменной X на трехместный предикат P(X,Y,Z), то получится двуместный предикат, зависящий от (Y,Z). При этом, говорят, что переменная X в указанном предикате **связана квантором**, а переменные Y,Z – **свободные**.

Область действия квантора – выражение, на которое навешивается квантор. Для устранения разночтений оно может быть заключено в скобки.

Задание 2.1. Перевести в десятичную систему двоичное число a_{41} a_{42} a_{43} a_{44} и прибавить 1. Полученное число i определяет номер предиката $P_i(X, Y)$ и множества значений $\{X\}$, $\{Y\}$. Заполнив таблицу, найти область истинности предиката $P_i(X, Y)$.

Таблица

X: / Y:	X_1	X_2	X_3	X_4
Y_1				
Y_2				
Y_3				

1) $P(X,Y)$: $\max(X,Y)$ – четное число; $X = \{1, 2, 3, 4\}$, $= \{0, 2, 3\}$.

2) $P(X,Y)$: $\max(X,Y)$ – нечетное число; $X = \{1, 2, 3, 4\}$, $Y = \{0, 2, 3\}$.

3) $P(X,Y)$: $\min(X,Y)$ – четное число; $X = \{1, 2, 3, 4\}$, $Y = \{0, 2, 3\}$.

4) $P(X,Y)$: $\min(X,Y)$ – нечетное число; $X = \{1, 2, 3, 4\}$, $Y = \{0, 2, 3\}$.

5) $P(X,Y)$: $\max(X,Y)$ – четное число; $X = \{3, 4, 5, 6\}$, $Y = \{2, 3, 4\}$.

6) $P(X,Y)$: $\max(X,Y)$ – нечетное число; $X = \{3, 4, 5, 6\}$, $Y = \{2, 3, 4\}$.

7) $P(X,Y)$: $\max(X,Y)$ – четное число; $X = \{3, 4, 5, 6\}$, $Y = \{2, 4, 5\}$.

8) $P(X,Y)$: $\max(X,Y)$ – нечетное число; $X = \{3, 4, 5, 6\}$, $Y = \{2, 4, 5\}$.

9) $P(X,Y)$: $\max(X,Y)$ – четное число; $X = \{2, 3, 4, 5\}$, $Y = \{2, 4, 5\}$.

10) $P(X,Y)$: $\max(X,Y)$ – нечетное число; $X = \{2, 3, 4, 5\}$, $Y = \{2, 4, 5\}$.

11) $P(X,Y)$: $\max(X,Y)$ – четное число; $X = \{2, 3, 4, 5\}$, $Y = \{3, 4, 5\}$.

12) $P(X,Y)$: $\max(X,Y)$ – нечетное число; $X = \{2, 3, 4, 5\}$, $Y = \{3, 4, 5\}$.

13) $P(X,Y)$: $(X - Y)$ – четное число; $X = \{1, 2, 3, 4\}$, $Y = \{0, 2, 3\}$.

14) $P(X,Y)$: $(X - Y)$ – нечетное число; $X = \{1, 2, 3, 4\}$, $Y = \{0, 2, 3\}$.

15) $P(X,Y)$: $\min(X,Y)$ – четное число; $X = \{2, 5, 6, 8\}$, $Y = \{3, 6, 9\}$.

16) $P(X,Y)$: $\min(X,Y)$ – нечетное число; $X = \{2, 5, 6, 8\}$, $Y = \{3, 6, 9\}$.

Определить значения высказываний:

а) $\forall X\, \exists Y\, (P(X, Y) = 1)$

б) $\forall Y\, \exists X\, (P(X, Y) = 1)$

в) $\exists X\, \forall Y\, (P(X, Y) = 1)$

г) $\exists Y\, \forall X\, (P(X, Y) = 1)$

д) $\exists X\, \forall Y\, (P(X, Y) = 0)$

е) $\forall Y\, \exists X\, (P(\mathrm{X}, Y) = 0)$

Задание 2.2. Заполнить 15 числами таблицу, представляющую двуместный предикат. Построить таблицы истинностей одноместных предикатов, получаемых из заданного предиката навешиванием кванторов существования и всеобщности по переменным X и Y.

Определить значения высказываний:

а) $\forall X\, \exists Y : (P(X, Y) = 0)$

б) $\forall Y\, \exists X : (P(X, Y) = 0)$

в) $\exists X\, \forall Y : (P(X, Y) = 1)$

г) $\exists Y\, \forall X : (P(X, Y) = 1)$

д) $\exists X\, \forall Y : (P(X, Y) = 1)$

е) $\forall Y\, \exists X : (P(\mathrm{X}, Y) = 1)$

Таблица

X: / Y:	X_1	X_2	X_3	X_4	X_5
Y_1	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
Y_2	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}
Y_3	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

Пример выполнения задания 2.2.

X: / Y:	X_1	X_2	X_3	X_4	X_5
Y_1	1	0	0	1	1
Y_2	0	1	1	0	1
Y_3	0	1	1	0	1

	∀ X (P(X,Y)=1)
Y_1	0
Y_2	0
Y_3	0
	∃ X (P(X,Y)=0)
Y_1	1
Y_2	1
Y_3	1

	∃ X (P(X,Y)=1)
Y_1	1
Y_2	1
Y_3	1

X:	X_1	X_2	X_3	X_4	X_5
∃ Y (P(X,Y)=1)	1	1	1	1	1

X:	X_1	X_2	X_3	X_4	X_5
∃ Y (P(X,Y)=0)	1	1	1	1	0

X:	X_1	X_2	X_3	X_4	X_5
∀ Y (P(X,Y)=1)	0	0	0	0	1

а) ∀X ∃Y (P(X,Y)=0) = 0

б) ∀Y ∃X (P(X,Y)=0) = 1

в) ∃X ∀Y (P(X,Y)=1) = 1

г) ∃Y ∀X (P(X,Y)=1) = 0

д) ∃ X ∀Y(P(X,Y)=1) = 1

е) ∀Y ∃X (P(X,Y)=1) = 1

Контрольный тест

1. **Для множеств $X = \{0,3,5\}$ и $Y = \{0,3\}$ предикат $P(X,Y)$: "$\max(X,Y)$ – четное число" может быть представлен таблицей**

A)

Y \ X	0	3	5
0	1	0	0
3	0	0	0

B)

Y \ X	0	3	5
0	1	1	1
3	0	0	1

C)

Y \ X	0	3	5
0	1	1	0
3	0	1	0

D)

Y \ X	0	3	5
0	0	1	0
3	1	0	0

2. **Предикатная формула $\exists Z\, P(X,Y,Z)$ представляет собой**
   A) двуместный предикат $Q(X,Y)$
   B) ложное высказывание
   C) истинное высказывание
   D) одноместный предикат $Q(Z)$

3. **Подстановка константы 1 вместо Y превращает функцию $f(X,Y)$ в**
   A) логическую константу
   B) функцию одной переменной $g(Y)$
   C) 1
   D) функцию одной переменной $g(X)$

4. **Предикатная формула $\exists X\ (3X = 6)$ на предметной области действительных чисел R представляет собой**
   A) одноместный предикат
   B) истинное высказывание
   C) ложное высказывание
   D) линейное уравнение

5. **Определите неправильную формулу**
   A) $\exists X : (A(X) \,\&\, B) \equiv (\exists X : A(X)) \,\&\, B;$
   B) $\forall X : (A(X) \,\&\, B) \equiv (\forall X : A(X)) \,\&\, B$
   C) $\exists X : (A(X) \vee B) \equiv (\exists X : A(X)) \vee B;$
   D) $\forall X : (A(X) \vee B) \equiv (\forall X : A(X)) \vee B$
   E) $\forall X : (A(X) \vee B) \equiv (\exists X : A(X)) \vee B$

3. ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

Определение 1. *Графом* называется произвольное множество элементов V, называемых вершинами графа, и произвольное множество E пар из V. Обозначение: $G = (V, E)$.

В дальнейшем будем рассматривать конечные графы, то есть графы с конечными множествами элементов и пар.

Определение 2. Если элементы из E рассматривать как неупорядоченные пары, то граф называется *неориентированным*, а пары называются *рёбрами*. Если же элементы из E рассматривать как упорядоченные, то граф *ориентированный*, а пары — *дуги*.

Определение 3. Пара вида (a, a) называется *петлёй*, если пара (a, b) встречается в семействе E несколько раз, то она называется *кратным ребром* (*кратной дугой*).

В дальнейшем условимся уточнять наличие петель и кратных ребер соответствующими названиями:

петли (если есть – «+», нет – «–»)	кратные ребра (если есть – «+», нет – «–»)	Названия графа
-	-	(просто) *граф*
-	+	*мультиграф*
+	+	*псевдограф*

Определение 4. Две вершины графа, соединенные ребром, называются *смежными*. Причем вершины и ребро называются инцидентными.

Определение 5. *Степенью вершины* (deg v) называется количество рёбер, инцидентных данной вершине. Для псевдографа полагают учитывать петлю дважды.

В любом графе (псевдографе) сумма степеней вершин равно удвоенному количеству ребер.

Определение 6. *Путём* в графе $G = (V, E)$ называется любая последовательность вида

$$v0, (v0, v1), v1, (v1, v2), \dots, vn - 1, (vn - 1, vn), vn \qquad (3.1)$$

Число n в данных обозначениях называется *длиной пути*.

Определение 7. *Цепью* называется путь, в котором нет повторяющихся рёбер.

Определение 8. *Простой цепью* называется путь без повторения вершин.

Из каждой пути P из v_k в v_l ($v_k \neq v_l$) можно выделить подпуть из v_k в v_l, являющийся простой цепью.

Определение 9. Путь типа (3.1) называется *замкнутым*, если $v_0 = v_n$.

Определение 10. Путь называется *циклом*, если он замкнут, и рёбра в нём не повторяются.

Определение 11. Путь называется *простым циклом*, если $v_0 = v_n$ и вершины не повторяются.

Определение 12. Граф $G = (V, E)$ называется *связным*, если для любых вершин $v_i, v_j \in V$ ($vi \neq vj$) существует путь из v_i в v_j. В противном случае граф состоит из нескольких частей, являющиеся, в свою очередь связными графами (подграфами исходного графа) и которые называются *компонентами связности*.

Определение 13. *Деревом* называется связный граф без циклов.

Следующие пять определений эквивалентны (p — число вершин, q — число рёбер):

1) G — дерево;

2) G — без циклов и $q = p - 1$;

3) G — связный граф и $q = p - 1$;

4) G — связный граф, но при удалении любого ребра становится несвязным;

5) G — без циклов, но при добавлении любого ребра на тех же вершинах появляется цикл.

Определение 14. Подграф $G1 = (V1, E1)$ графа $G = (V, E)$, называется *остовным деревом* в графе $G = (V, E)$, если $G1 = (V1, E1)$ — дерево и $V1 = V$. Любой связный граф содержит хотя бы одно остовное дерево.

Относительно остова D все ребра подграфа G \ D называются хордами. Каждая хорда связывает две вершины остова. Таким образом, относительного любого остова хорды определяют циклы. Множество этих циклов составляют базис в пространстве всевозможных циклов исходного графа *G*. Количество циклов в базисе равно цикломатическому числу, который вычисляется по формуле

то $\nu = p - b + k$,

Здесь

$$p=\sum_{i=1}^{k} p_i \quad b=\sum_{i=1}^{k} b_i$$

k – (количество) компонент связности; p_i и b_i – соответственно количество ребер и вершин в i-ой компоненте.

Существует несколько способов задания графов, связанных с различной формой задания функции Г. Вот некоторые из них для конечных графов.

1) Перечисление (список) ребер графа с указанием их концов и добавлением списка изолированных вершин.

2) **Матрица инциденций графа с b вершинами и p ребрами** - прямоугольная матрица $A = \| a_{ij} \|$ с b строками и p столбцами, строки которой соответствуют вершинам графа, а столбцы - ребрам, причем для неориентированного графа элемент матрицы a_{ij} равен 1, если вершина v_i и ребро e_j инцидентны, и равен 0 в противном случае. Для ориентированного графа $a_{ij} = -1$, если v_i является началом дуги e_j, и $a_{ij} = +1$, если v_i - конец дуги e_j. В каждом столбце матрицы инциденций - два ненулевых элемента, если ребро - не петля. Петле соответствует элемент, равный 2.

3) **Матрица соседства (смежности) вершин графа с b вершинами** - квадратная матрица $B = \| b_{ij} \|$ размерности b, строки и столбцы которой соответствуют вершинам графа, причем неотрицательный элемент b_{ij} равен числу ребер, идущих из вершины v_i в вершину v_j (b_{ij} не равно, вообще говоря, b_{ji}, однако для неориентированных графов матрица соседства -

симметричная). Для несмежных вершин соответствующий элемент матрицы равен 0.

Если матрица инциденций задает граф однозначно, то матрица соседства вершин определяет граф с точностью до замены любого неориентированного ребра парой противоположно направленных дуг между теми же вершинами. Однако для графов без кратных ребер задание графа и этой матрицей однозначно, элементы матрицы соседства равны в этом случае 0 или 1.

4) Для наглядности граф часто представляют в виде геометрического объекта: вершинам соответствуют различные выделенные точки в пространстве (на плоскости), ребрам - отрезки кривых, связывающие соответствующие точки и не проходящие через выделенные точки, отличные от их концов. Отношению инцидентности вершин и ребер графа соответствует при этом геометрическая инцидентность выделенных точек и линий. Кроме того, предполагается, что кривые попарно не пересекаются во внутренних точках. Такое представление графа называется **реализацией**.

Задание 3.1. Ориентированный граф G с множеством вершин V = {1, 2, 3, 4, 5, 6, 7} задан списком дуг. Список дуг определяется из персональной таблицы параметрами $a_1 - a_{45}$. Вычисляются сначала 15 чисел b_k (k=0-14) по формулам $b_k = (a_{k+1}\ a_{k+2}\ a_{k+3})_{10}$ последняя десятка внизу означает перевод двоичного числа в десятичную систему. Если при каком то k *получается* $a_{k+1}\ a_{k+2}\ a_{k+3}$ =000, то принять соответствующее b_k = 7.

Затем список дуг определяется так:

E = {(1, b_0), (1, b_1), (2, b_2), (2, b_3), (3, b_4), (3, b_5), (4, b_6), (4, b_7), (4, b_8), (5, b_9), (5, b_{10}), (6, b_{11}), (6, b_{12}) (7, b_{13}), (7, b_{14})}.

Построить реализацию графа G, матрицу инциденций графа и матрицу соседства вершин графа G.

Для соотнесенного неориентированного графа построить матрицу соседства вершин.

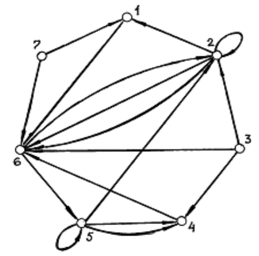
Пример выполнения задания 3.1.

Ориентированный граф G с множеством вершин V = {1, 2, 3, 4, 5, 6, 7} задан списком дуг E = {(1, 4) (1, 6) (2, 1) (2, 2) (2, 6) (2, 6) (3, 2) (3, 4) (4, 6) (5, 2) (5, 4) (5, 4) (5, 5) (6, 2) (6, 5) (7, 1) (7, 6)}.

Построить реализацию графа G, матрицу инциденций графа и матрицу соседства вершин графа G.

Для соотнесенного неориентированного графа построить матрицу соседства вершин.

1. Построим реализацию графа G. Изображения вершин 1,2,…,7 разместим в вершинах правильного 7-угольника. Соединим пары вершин в соответствии со списком дуг, избегая пересечения трех ребер в одной точке. Ориентацию дуг обозначим стрелками.



2. Построим матрицу инциденций графа G. Число ребер графа -18. В прямоугольной матрице $A = \|a_{ij}\|$ размерности 7x18 последовательно заполняем столбцы в соответствии со списком дуг: каждой дуге $e_k = (i, j)$, где $i \neq j$ соответсвуют элементы a_{ik} = -1 и a_{jk} = +1. Петле $e_k = (i, i)$ соответствует a_{ik} = 2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-1	-1	1														1	
2			-1	2	-1	-1	1				1				1			
3							-1	-1	-1									
4	1							1		-1		1	1					
5											-1	-1	-1	2		1		
6		1			1	1			1	1					-1	-1		1
7																	-1	-1

3. Построим матрицу соседства вершин графа G. В квадратной матрице $B = \|b_{ij}\|$ размерности 7x7 каждой дуге $e_k = (i, j)$, соответствует элемент b_{ij} = 1, кроме кратных дуг, котором соответствует элемент равный кратности дуг.

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

4. Для соотнесенного неориентированного графа построим матрицу соседства вершин. Квадратная матрица $C = \|c_{ij}\|$ размерности 7х7 строится из предыдущей матрицы соседства вершин $B = \|b_{ij}\|$ по формулам:

$c_{ij} = c_{ji} = b_{ij} + b_{ji}$, для $i \neq j$,

и $c_{ii} = b_{ii}$.

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

Задание 3.2.

Неориентированный граф G с множеством вершин V = {1, 2, 3, 4, 5, 6, 7} задан списком ребер.

Список дуг определяется из персональной таблицы параметрами $a_1 - a_{28}$. Вычисляются сначала 10 чисел b_k (k=0-9) по формулам $b_k = (a_{k+1}\ a_{k+2})_{10}$ последняя десятка внизу означает перевод двоичного числа в десятичную систему.

Затем список ребер определяется так:

E = {(1, 1+b_0), (1, 2+b_1), (1, 3+b_2), (1, 4+b_3), (2, 2+b_4), (2, 3+b_5), (2, 4+ b_6), (3, 3+ b_7), (4, 4+b_8), (4, 7), (5, 4+b_9), (5, *7*), (6, *7*)}. Если получается пара (x,x) то следует заменить ее на (x,x+1)

Построить реализацию графа G.

Найти цикломатическое число графа G.

Выбрать остов графа G.

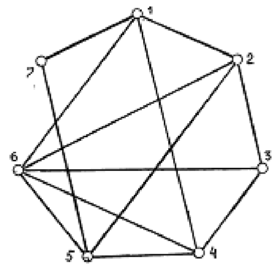
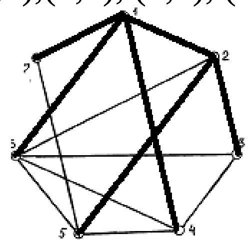
Построить базис циклов графа G.

Пример выполнения задания 3.2.

Неориентированный граф G с множеством вершин V = {1, 2, 3, 4, 5, 6, 7} задан списком ребер E = {(1, 2), (1, 4), (1, 6), (1, 7), (2, 3), (2, 5), (2, 6), (3, 4), (3, 6), (4, 5), (4, 6), (5, 6), (5, 7)}.

Построить реализацию графа G. Найти цикломатическое число графа G.

Выбрать остов графа G. Построить базис циклов графа G.

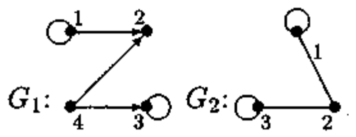
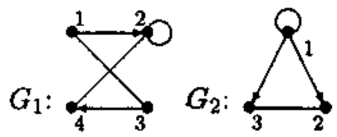
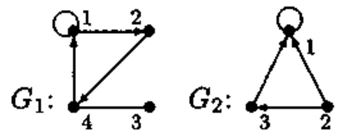
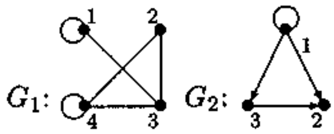
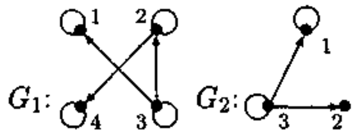
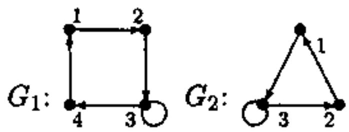
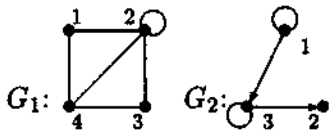
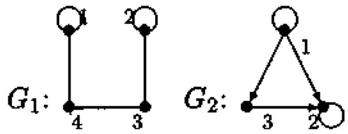
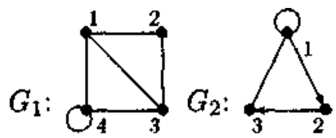
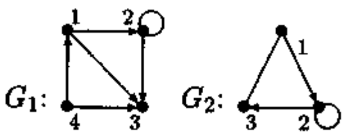
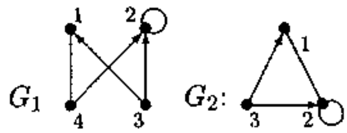
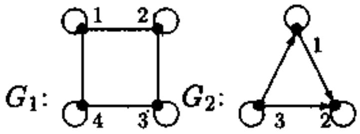
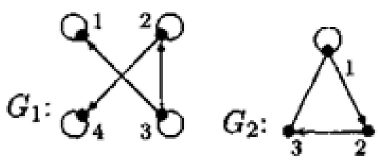
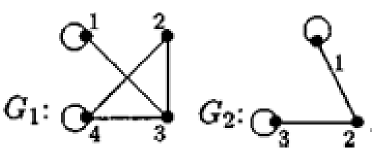
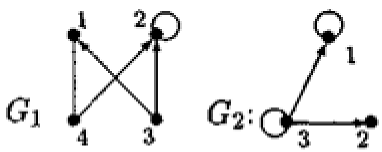
<table>
<tr>
<td>1. Построим реализацию графа G. Изображения вершин 1,2,…,7 разместим в вершинах правильного 7-угольника. Соединим пары вершин в соответствии со списком дуг, избегая пересечения трех ребер в одной точке.</td>
<td></td>
</tr>
<tr>
<td>2. Найдем цикломатическое число графа G. Оно равно размерности базиса циклов и находится по формуле ν = p – b + k, где p – число ребер, b – число вершин, k – связность графа.</td>
<td>p = 13, b = 7, k = 1, так как граф связный.
ν = 13 -7 +1 =7</td>
</tr>
<tr>
<td>3. Выберем остов графа G. Поскольку число вершин графа равно 7, в любом его остове должно быть 6 ребер. При заданном упорядочении вершин 1-7, начиная с вершины 1, последовательно добавляем ребра, присоединяющие к строящемуся дереву новую вершину.</td>
<td>(1,2), (1,4),(1,6), (1,7), (2,3),(2,5).
</td>
</tr>
<tr>
<td>4. Построим базис циклов графа G. Составим список хорд относительно построенного остова (все оставшиеся ребра). Число хорд ложно быть равно цикломатическому числу, т.е. 7. Поочередно присоединяя к остову по одной хорде, выделяем элементарный цикл, состоящий из этой хорды и части ребер остова – единственный цепи, связывающей в остове концы хорды.
Выразим в построенном базисе элементарный цикл C =(6,2,1,4,5,6). Он получается сложением по модулю тех базисных циклов, которые содержат хорды, принадлежащие C (отмечены *).</td>
<td>(2,6),(3,4),(3,6),(4,5),(4,6), (5,6),(6,7).
<table>
<tr><th>Хорда</th><th>Добавляемая цепь</th></tr>
<tr><td>* (2, 6)</td><td>(6, 1), (1, 2)</td></tr>
<tr><td>(3, 4)</td><td>(4, 1), (1, 2), (2, 3)</td></tr>
<tr><td>(3, 6)</td><td>(6, 1), (2, 1), (2, 3)</td></tr>
<tr><td>* (4, 5)</td><td>(5, 2), (2, 1), (1, 4)</td></tr>
<tr><td>(4, 6)</td><td>(6, 1), (1, 4)</td></tr>
<tr><td>* (5, 6)</td><td>(6, 1), (1, 2), (2, 5)</td></tr>
<tr><td>(6, 7)</td><td>(7, 1), (1, 6)</td></tr>
</table></td>
</tr>
</table>

Задание 3.3.

Перевести в десятичную систему двоичное число a_{41} a_{42} a_{43} a_{44} и прибавить 1. Полученное число *i* определяет номер варианта.

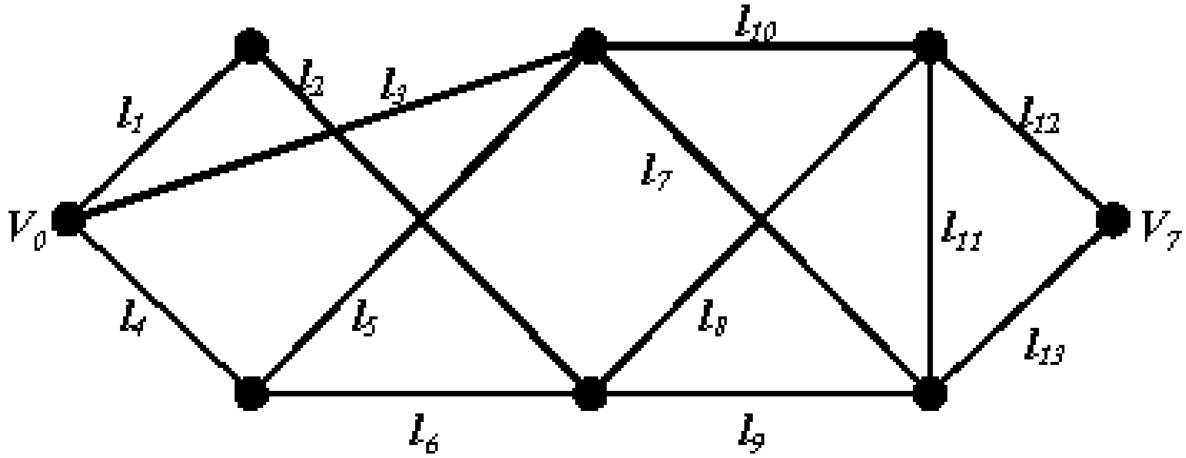
Даны графы G_1 и G_2. Найдите $G_1 \cup G_2$, $G_1 \cap G_2$, $G_1 \oplus G_2$, аналитически и изобразить результат графически.

Для графа $G_1 \cup G_2$ найдите матрицу смежности, матрицу инцидентности, простые цепи, простые циклы, исходящие из вершины 1.

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

Задача 3.3. (Задача о кратчайшем пути). Перевести в десятичную систему двоичное число a_{25} a_{26} a_{27} a_{28} и прибавить 1. Полученное число определяет номер варианта.

В каждом варианте определены веса ребер $L=(l_1, l_2 ..., l_{13})$ заданного графа. Найдите кратчайший путь из вершины V_0 и V_7.



Варианты заданий.

1. L = (3, 7, 12, 8, 24, 9, 13, 5, 4, 2, 16, 3, 6)
2. L = (5, 41, 23, 1, 7, 27, 42, 92, 6, 9, 33, 55,4)
3. L = (51, 4, 52, 9, 5, 2, 11, 3, 42, 6, 9, 22, 8)
4. L = (5, 41, 2, 49, 25, 2, 1, 3, 39, 7, 10, 21, 3)
5. L = (7, 3, 2, 19, 7, 12, 52, 7, 2, 9, 9, 31, 12)
6. L = (27, 14, 35, 71, 4, 1, 1, 13, 21, 16,49, 4, 8)
7. L = (6, 32, 12, 4, 5, 2, 11, 3, 42, 6, 9, 22, 3)
8. L = (41, 5, 2, 19, 35, 14, 1, 23, 12, 3, 8, 72, 3)
9. L = (72, 35, 2, 3, 6, 13, 41, 4, 21, 21, 6, 5, 7)
10. L = (1, 7, 5, 1, 8, 4, 7, 12, 4, 8, 6, 24, 3)
11. L = (1, 44, 35, 21, 61, 1, 31, 2, 4, 1, 5, 32, 82)
12. L = (7, 32, 2, 31, 9, 2, 17, 9, 3, 56,19, 2, 17)
13. L = (6, 23, 32, 6, 9, 12, 41, 5, 24, 6, 8, 6, 9)
14. L = (5, 24, 2, 5, 9, 1, 61, 53,22, 3, 1, 61, 2)
15. L = (6, 34, 21, 81, 2, 7, 31, 6, 19, 4, 2, 2, 1)
16. L = (20, 5, 2,19, 31, 7, 19, 4, 2, 8, 3, 2, 5)

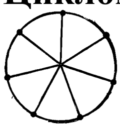
Контрольный тест

1. **Цепью** называется
   A) последовательность вершин и ребер, образующая путь в соотнесенном неориентированном графе.
   B) путь, в которой начальная вершина совпадает с конечной.
   C) последовательность рассуждений
   D) ориентированный граф
   E) остов графа

2. **Вершина, не инцидентная ни одному ребру, называется**
   A) изолированной
   B) концевой
   C) висячей
   D) петлей
   E) одинокой

3. **Вершина, инцидентная ровно одному ребру**
   A) изолированной
   B) концевой
   C) смежной
   D) равновесной
   E) одинокой

4. **Цикломатическое число графа равно**

   A) 14
   B) 7
   C) 15
   D) 2
   E) 12

5. **Число ребер в 5-мерном единичном кубе E^5 равно**
   A) 64
   B) 80
   C) 128
   D) 100
   E) 32

4. ЭЛЕМЕНТЫ ТЕОРИИ КОДИРОВАНИЯ

4.1. Алфавитное кодирование

1. Кодом называется система условных знаков (символов) для передачи, обработки и хранения (запоминания) различной информации. Предметом исследования теории кодирования являются отображения конечных или счетных множеств объектов произвольной природы в множества последовательностей из цифр 0, 1,..., r–1, где r – некоторое целое положительное число (в частности, r = 2). Такие отображения называются **кодированиями**.

Пусть A – произвольный алфавит. Элементы алфавита A будем называть *буквами*, а конечные последовательности (кортежи), составленные из букв, – *словами в A. Длина* (число букв) слова a обозначается через l(a), а слово длины 0 (**пустое слово**) обозначается символом L. *Соединение* слов a_1 и a_2 обозначим через a_1a_2, а соединение n одинаковых слов a – через a^n (a^0 = L).

Пусть U – произвольное множество слов в A. Через U^n (n = 0,1,...) обозначим множество всех слов в A^n, представимых в виде соединения n слов из U. В частности, через A^n (n = 0,1,...) обозначим множество всех слов длины n в алфавите A, а через A* – множество всех слов произвольной длины в A.

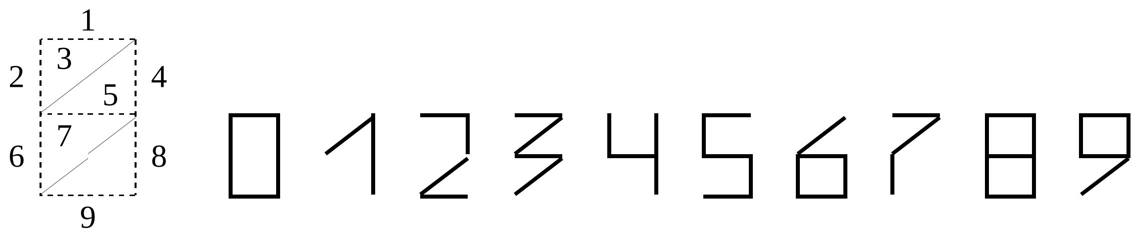
Множество $U^* = \bigcup_{n=0}^{\infty} U^n$ – не всевозможные слова в алфавите A; это зависит от выбранного множества U. Так, если U состоит из двух слов 00 и 11, то любое их соединение, во-первых, имеет четную длину и, во-вторых, нули и единицы расположены парами; например, слово 0110 не принадлежит множеству U^2.

Слово a_1 называют **началом** слова a, если существует слово a_2 такое, что $a = a_1a_2$; при этом слово a_1 называют **собственным началом** слова a, если $a_2 \neq L$. Слово a_2 называют **окончанием** слова a, если существует слово a_1

такое, что a = a_1a_2; при этом слово a2 называют **собственным окончанием** слова a, если a1 № L. В частности, пустое слово является началом и окончанием любого слова a, причем собственным, если $a \neq L$. Множество начал слов из множества U обозначим через $\vec{U}$, а множество окончаний – через $\overleftarrow{U}$.

2. Рассмотрим алфавит $B_{(r)}$ = {0, 1,..., r–1}, где r і 2, и произвольное множество G. В частности, G может быть конечным алфавитом, множеством натуральных чисел, схем или формул определенного типа, множеством слов в некотором алфавите и т.п. Произвольное отображение множества G в множество слов в алфавите B называется **r-ичным кодированием множества G**. Мы будем рассматривать только случай r = 2, т.е. **двоичные** кодирования, и в связи с этим под log X будем понимать $\log_2 X$.

В качестве примера рассмотрим отображение, применяемое при написании цифр почтового индекса. Каждая десятичная цифра, выделенная отправителем из девяти отрезков шаблона, кодируется для автоматического распознавания словом длины 9 в алфавите B: символами 1 отмечаются номера использованных линий (например, цифре 2 соответствует слово 100100101; цифре 5 – слово 110010011).



Важную роль в теории кодирования играет кодирование множества G, состоящего из всех (или выделенной части) слов в некотором алфавите A. Элементы кодируемого множества (т. е. выделенные слова в A) при этом называют **сообщениями**, имея в виду возможную передачу кода по каналу связи. В общем случае не накладывается никаких условий на процесс вычисления (сам алгоритм, его эффективность) для сообщения соответствующего ему двоичного слова. Однако для многих вопросов достаточно ограничиться рассмотрением более узких классов кодирований слов, так называемых побуквенных кодирований.

Пусть A = $\{a_i\}$ – конечный алфавит, буквы которого занумерованы натуральными числами. В этом случае кодирование букв алфавита A можно задать двоичными словами V = $\{v_i\}$, где v_i есть образ буквы a_i . Слова V = $\{v_i\}$ будем называть **кодами** (алфавита A). Кодирование слов в алфавите A, при котором каждому слову (сообщению) $a_{i_1}a_{i_2}...a_{i_k}$ ставится в соответствие слово $v_{i_1}v_{i_2}...v_{i_k}$, будем называть **побуквенным кодированием** и обозначать через $K^{a_i}_{v_i}$ (или K^A_V).

Если коды всех букв алфавита имеют одинаковую длину, то код называется равномерным (пример – кодирование почтового индекса).

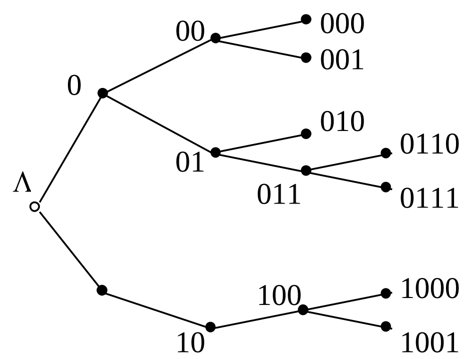
4. Можно считать, что в коде (множестве слов) V = $\{v_i\}$, содержатся только различные слова. Назовем код V = $\{v_i\}$ **разделимым**, если из каждого равенства в алфавите B = {0, 1} вида $v_{i_1}v_{i_2}...v_{i_k} = v_{j_1}v_{j_2}...v_{j_l}$ следует, что k = *l* и $i_t = j_t$, t = 1, 2,...,k.

Побуквенное кодирование является взаимно однозначным тогда и только тогда, когда оно задается с помощью разделимого кода. Код V = $\{v_i\}$ будем называть **префиксным**, если никакое слово v_k не является началом никакого слова v_j, $j \neq k$. *Префиксность является достаточным условием разделимости*. Каждое слово префиксного кода заменить наименьшим его началом, которое не является началом других кодовых слов. При этом полученный код также будет префиксным. Такую операцию будем называть **усечением** префиксного кода.

Простейшим примером префиксного кода является равномерный код. Он не требует разделителей, поскольку код каждой передаваемой буквы заканчивается, а код следующей буквы начинается через определенное одинаковое число кодирующих символов.

Для произвольного кода V, состоящего из различных слов, можно построить кодовое дерево:

i	l_i	v_i
0	3	000
1	3	001
2	3	010
3	4	0110
4	4	0111
5	4	1000
6	4	1001



В дальнейшем для конечного кода $V = \{v_0, v_1, ..., v_{m-1}\}$ будем считать, что $m \geq 2$, и пользоваться обозначением $\lambda_{max} = \max\limits_{0<i<m-1} \lambda(v_i)$.

Теорема 1. Пусть l_0, $l_1, ..., l_{m-1}$ – произвольный набор натуральных чисел ($m \geq 2$). Для того чтобы существовал разделимый код $V = \{v_0, v_1, ..., v_{m-1}\}$ с длинами $\lambda(v_i) = l_i$, $i = 0, 1, ..., m-1$, необходимо и достаточно выполнение **неравенства Крафта-Макмиллана для разделимых кодов**:

$$\sum_{i=0}^{m-1} \frac{1}{2^{l_i}} \leq 1.$$

Для кода, заданного в таблице выполнено: $3 \cdot \frac{1}{2^3} + 4 \cdot \frac{1}{2^4} = \frac{3}{8} + \frac{4}{16} = \frac{5}{8} < 1$.

Следствие. Для любого разделимого кода $V = \{v_0, v_1, ..., v_{m-1}\}$ существует префиксный код с таким же набором длин кодовых слов, что и у кода V.

4.2. Оптимальное кодирование

1. Здесь мы разберем построение эффективных взаимно однозначных кодирований при некоторых простейших предположениях относительно статистических свойств источника сообщений. При этом более эффективными считаются кодирования, у которых в среднем на одну букву сообщения приходится меньшее число двоичных цифр.

Рассматривается источник, который случайным образом (т.е. в случайном порядке) последовательно порождает буквы алфавита A = {a_0, a_1,..., a_{m-1}). Предполагается, что последовательные появления букв алфавита A статистически независимы и подчинены распределению вероятностей

$$P = \{p_0, p_1, ..., p_{m-1}\} \ (p_i \geq 0, \sum_{i=0}^{m-1} p_i = 1).$$

Каждый код V = {v_0, v_1,..., v_{m-1}} в алфавите B = {0, 1) характеризуется средним числом

$$L_V(P) = \sum_{i=0}^{m-1} p_i \cdot \lambda(v_i)$$

двоичных цифр, приходящихся на одну букву алфавита A при побуквенном кодировании $K_{v_0}^{a_0}, K_{v_1}^{a_1}, ..., K_{v_{m-1}}^{a_{m-1}}$. Величину $L_V(P)$ называют **стоимостью** кода V при распределении P. Пусть $L(P) = \inf L_V(P)$, где нижняя грань берется по (счетному) множеству префиксных кодов V, состоящих из m слов. Префиксный код V = {v_0, v_1,..., v_{m-1}}будем называть **оптимальным** при распределении P = {p_0, p_1,..., p_{m-1}), если $L_V(P) = L(P)$. Основная задача, которая здесь возникает, состоит в отыскании методов построения оптимальных или близких к ним по стоимости кодов и в оценке величины $L(P)$.

Известны два метода построения кодов, близких к оптимальному, принадлежащие К. Шеннону и Р. Фано. Метод Фано, отличающийся чрезвычайной простотой конструкции, заключается в следующем. Упорядоченный (в порядке убывания вероятностей) список букв делится на две (последовательные) части так, чтобы суммы вероятностей входящих в них букв как можно меньше отличались друг от друга. Буквам из первой части приписывается символ 0, а буквам из второй части – символ 1. Далее точно так же поступают с каждой из полученных частей, если она содержит по крайней мере две буквы. Этот дихотомический процесс продолжается до тех пор, пока весь список не разобьется на части, содержащие по одной букве. Каждой букве ставится в соответствие последовательность символов, приписанных в результате этого процесса данной букве. Легко видеть, что полученный код является префиксным. Пример кода, построенного по методу Фано, приводится в следующей таблице.

сообщение	вероятность											коды	вес
a	0,30	0,48	0,30				0	0				00	2
b	0,18		0,18					1				01	2
c	0,14	0,52	0,28	0,14			1	0	0			100	3
d	0,14			0,14					1			101	3
e	0,11		0,24	0,11				1	0			110	3
f	0,06			0,13	0,06				1	0		1110	4
g	0,05				0,07	0,05				1	0	11110	5
h	0,02					0,02					1	11111	5

Стоимость кода:

$$2 \cdot (0{,}30 + 0{,}18) + 3 \cdot (0{,}14 + 0{,}14 + 0{,}11) + 4 \cdot 0{,}06 + 5 \cdot (0{,}05 + 0{,}02) = 2{,}72.$$

Следует заметить, что деление пополам списка букв может при определенном соотношении частот осуществляться неоднозначно.

4.3. Коды с обнаружением и исправлением ошибок

Рассмотрим множество B^n слов длины n в алфавите B = {0, 1}. Каждое слово $x_1x_2...x_n$ в алфавите B отождествим с n-мерным вектором (x1,x2,...,xn). Суммой векторов $(x_1x_2...x_n)$ и $(y_1,y_2,...,y_n)$ из B^n назовем вектор $(x_1 \oplus y_1, x_2 \oplus y_2,..., x_n \oplus y_n)$. Напомним, что знак $\oplus$ означает сумму по модулю 2. Если ввести еще умножение произвольного вектора $X = (x_1,x_2,...,x_n)$ на элемент $b \in B$ следующим образом: $bX = (bx_1,bx_2,...,bx_n)$, то множество B^n можно рассматривать как n-мерное векторное пространство.

В векторном пространстве B^n можно определить **расстояние Хемминга d(X,Y)** между векторами X и Y как число несовпадающих компонент векторов X и Y и **норму ||X||** вектора X как расстояние между X и нулевым n-мерным вектором (0,0,...,0). Очевидно, что для любого слова $X = x_1x_2...x_n$ из B^n:

$$\|X\| = \sum_{j=1}^{n} x_j$$

и для любых X и Y из B^n:

$$d(X, Y) = \|X \oplus Y\|.$$

Метрическое пространство B^n допускает уже рассматривавшуюся нами простую геометрическую интерпретацию: множеству B^n соответствует множество вершин n-мерного единичного куба, а расстояние между двумя элементами из B^n равно минимальному числу ребер в цепи, соединяющей соответствующие вершины куба.

Для произвольного слова $X = x_1x_2...x_n \in B^n$ определим его *числовое значение* N(X):

$$N(X) = \sum_{j=1}^{n} x_j \cdot 2^{n-j}.$$

В дальнейшем в этом разделе будут рассматриваться лишь коды, состоящие из $m \geq 2$ двоичных слов фиксированной длины, причем в тех случаях, когда нумерация слов кода не существенна, мы не будем ее указывать. Для произвольного кода $V = \{v_0, v_1, ..., v_{m-1}\} \subseteq B^n$ положим:

$$d(V) = \min_{i \neq j} d(v_i, v_j).$$

Величина d(V) называется **кодовым расстоянием**.

Рассмотрим некоторые виды преобразований двоичных слов, называемых **ошибками**.

Одиночной ошибкой вида $0 \to 1$ ($1 \to 0$) в слове X называют результат замены одного из символов 0 (соответственно 1) символом 1 (соответственно 0). Одиночные ошибки этого вида называют также **замещениями** символов, или **аддитивными ошибками**.

Одиночной ошибкой вида 0 $\to$L (1 $\to$L) в слове X называют результат удаления одного из символов 0 (соответственно 1); при этом длина слова уменьшается на единицу. Одиночные ошибки этого вида называются **выпадениями символов**.

Одиночной ошибкой вида L $\to$0 (L $\to$1) называют результат *вставки символов* перед некоторым символом слова или после его последнего символа; при этом длина слова увеличивается на единицу.

Типом ошибки будем называть некоторое множество видов одиночных ошибок. Например, ошибка типа {0 $\to$L, 1 $\to$L, L $\to$0, L $\to$1} есть выпадение или вставка произвольного символа. Число одиночных ошибок в некоторой их последовательности будем называть **кратностью** ошибки. Так, в результате ошибки кратности 3 (вставка 1 перед первым символом, затем выпадение 0 перед пятым и замещение последнего символа) слово 0001101 переводится в слово 1001100.

В дальнейшем особое внимание будет уделено ошибкам типа {(0 $\to$1), (1 $\to$0)}, т.е. замещениям символов.

При постановке различных задач обнаружения и исправления ошибок заданного типа исходят из того или иного допущения о законе образования ошибок. Эти законы могут иметь как вероятностный, так и комбинаторный характер. Говорят, что тот или иной закон образования ошибок задается каналом передачи (или хранения) двоичных слов. Наиболее часто рассматриваются следующие два типа каналов:

а) в любом символе с вероятностью p ($0 < p < 1/2$) происходит ошибка заданного типа, причем ошибки в различных символах статистически независимы;

б) в каждом слове длины n может произойти любая ошибка заданного типа кратности не более s.

Основная идея помехоустойчивого кодирования состоит в следующем. Вместо сообщений X, Y, Z,... по каналу связи посылают их определенным образом закодированные образы X', Y', Z',..., которые обеспечивают обнаружение ошибок или правильное декодирование даже при наличии ошибок допустимого типа.

Возможность исправления ошибок при использовании кода $V = \{v_0, v_1, ..., v_{m-1}\} \in B^n$ можно пояснить следующим образом. Обозначим через T(X) множество слов, которые можно получить из слова X в результате ошибок, *допустимых в рассматриваемом канале*, в частности $T(v_i)$ – множество слов, в которые может превратиться в результате ошибок кодовое слово v_i. Произвольное однозначное отображение D множества $\bigcup_{i=0}^{m-1} T(v_i)$ на V будем называть **декодированием**. Задание декодирования D равносильно разбиению множества $\bigcup_{i=0}^{m-1} T(v_i)$ на непересекающиеся подмножества (**окрестности**) $D^{-1}(v_i)$, i = 0, 1,..., m–1, где каждая окрестность $D^{-1}(v_i)$ состоит из прообразов слова v_i при отображении D. Естественно считать, что при декодировании D в произвольном слове $v_i \in V$ исправляются те и только те ошибки, которые преобразуют слово v_i в некоторое слово из $T(v_i) \cap D^{-1}(v_i)$. В частности, для существования декодирования D, при котором исправляются все ошибки, допустимые в рассматриваемом канале, необходимо и достаточно, чтобы множества $T(v_i)$, i = 0, 1,..., m–1, попарно не пересекались. Тогда, получая по каналу связи некоторое сообщение Y, мы определяем, окрестности какого кодового слова X оно принадлежит, и делаем вывод, что послано именно слово X. Код, обладающий таким свойством, называется кодом, исправляющим ошибки данного типа.

Для кода $V \subseteq B^n$ с исправлением s ошибок типа замещения множество T(X) совпадает с *метрической окрестностью радиуса s слова X*, т.е. множеством точек, удаленных от X на расстояние не более s. Поэтому условие непересечения множеств $T(v_i)$ равносильно тому, что кодовое расстояние $d(V) > 2s$. Таким образом, *код V является* **кодом с исправлением s замещений** *тогда и только тогда, когда $d(V) > 2s$*, т.е. $d(V) \geq 2s + 1$, поскольку d(V) – целое число.

Наряду с задачами исправления ошибок можно рассматривать и более слабые задачи обнаружения ошибок. Очевидно, при использовании кода V нельзя обнаружить лишь те ошибки, которые преобразуют кодовые слова в другие кодовые слова. Отсюда, в частности, следует, что *код V с кодовым расстоянием d всегда позволяет обнаружить (d – 1) или менее одиночных ошибок типа замещения.*

Из сказанного выше можно заключить, что для обеспечения помехоустойчивости кодирования приходится применять более длинные коды, в которых кроме информации о самом сообщении содержится дополнительная информация, позволяющая при возникновении ошибок в процессе передачи (или хранения) сообщения обнаруживать или восстанавливать переданное сообщение. Этот факт можно сформулировать таким образом: за надежность приходится платить избыточностью кодирования.

Простейший способ построения помехоустойчивых кодов – дублирование. Повторение каждого символа передаваемого слова (2s + 1) раз позволяет не только обнаружить, но и исправить до s ошибок типа замещения.

Далее мы будем рассматривать только случай одиночной ошибки типа замещения. При использовании описанных простейших приемов для возможности обнаружения и исправления одиночной ошибки длина передаваемого слова утраивается.

Однако возможно значительно более экономное помехоустойчивое кодирование. Идея таких методов заключается в следующем. На множестве двоичных слов рассматривается некоторая функция. Искомый код определяется как множество слов из B^n, на которых эта функция принимает некоторое фиксированное значение. Функция подбирается таким образом, чтобы в результате любой одиночной ошибки значение функции изменялось и чтобы по этому изменению и, быть может, самому полученному слову можно было однозначно определить вид и место ошибки.

Мы рассмотрим один пример такого кодирования – код Хэмминга.

Зафиксируем число n и найдем число l такое, что $2^{l-1} \leq n < 2^l$. В этом случае $l = [\log n] + 1$. Например, $l(5) = l(7) = 3$, $l(8) = l(10) = l(13) = 4$.

Для произвольного слова $X = x_1x_2...x_n \in B^n$ положим

$$H(X) = x_1e_l(1) \oplus x_2e_l(2) \oplus ... \oplus x_ne_l(n).$$

H(X) представляет собой вектор длины l, полученный в результате сложения векторов, являющихся двоичными записями (с помощью l цифр) номеров единичных символов слова X.

Теорема. **Код Хэмминга** H_n, состоящий из всех слов $X = x_1x_2...x_n \in B^n$ таких, что

$$H(X) = (0, 0,..., 0), \qquad (*)$$

является кодом с исправлением одного замещения.

Пример построения кода Хемминга разберем при выполнении следующего задания.

Для заданного сообщения X = 0 1 1 0 1 0 1 построить код Хэмминга X'. Внести одиночную ошибку замещения и произвести декодирование.

<table>
<tr>
<td>Подготовим строку для сообщения X', отводя места с номерами, равными 2^k (1,2,4,8,…)для проверочных символов, а остальные разряды заполним символами из X.</td>
<td>
<table>
<tr><td>1</td><td></td><td>2</td><td></td><td>3</td><td></td><td>4</td><td></td><td>5</td><td></td><td>6</td><td></td><td>7</td><td></td><td>8</td><td></td><td>9</td><td></td><td>1
0</td><td></td><td>1
1</td></tr>
<tr><td>п</td><td></td><td>п</td><td></td><td>и</td><td></td><td>п</td><td></td><td>и</td><td></td><td>и</td><td></td><td>и</td><td></td><td>п</td><td></td><td>и</td><td></td><td>и</td><td></td><td>и</td></tr>
<tr><td></td><td></td><td></td><td></td><td>0</td><td></td><td></td><td></td><td>1</td><td></td><td>1</td><td></td><td>0</td><td></td><td></td><td></td><td>1</td><td></td><td>0</td><td></td><td>1</td></tr>
</table>
</td>
</tr>
<tr>
<td>Построим таблицу с двоичными номерами разрядов и внесем в информационные позиции знаки сообщения X. Для каждого из проверочных разрядов 2^k определим строки, формирующие проверочные знаки кода X':
1) информационный знак кода X'равен 1;
2) в двоичном представлении номера строки k –ый разряд равен 1.
Вычислим значения проверочных разрядов. Результаты подсчета в предпоследнем столбце.</td>
<td>
<table>
<tr><td>№ строки</td><td>Двоичное представление номера строки</td><td>инф. знаки</td><td>Провер. знаки</td><td>Номера строк формирующих провер. знаки</td></tr>
<tr><td>1</td><td>0001</td><td></td><td>1</td><td>5+9+11</td></tr>
<tr><td>2</td><td>0010</td><td></td><td>0</td><td>6+11</td></tr>
<tr><td>3</td><td>0011</td><td>0</td><td></td><td></td></tr>
<tr><td>4</td><td>0100</td><td></td><td>0</td><td>5+6</td></tr>
<tr><td>5</td><td>0101</td><td>1</td><td></td><td></td></tr>
<tr><td>6</td><td>0110</td><td>1</td><td></td><td></td></tr>
<tr><td>7</td><td>1111</td><td>0</td><td></td><td></td></tr>
<tr><td>8</td><td>1000</td><td></td><td>0</td><td>9+11</td></tr>
<tr><td>9</td><td>1001</td><td>1</td><td></td><td></td></tr>
<tr><td>10</td><td>1010</td><td>0</td><td></td><td></td></tr>
<tr><td>11</td><td>1011</td><td>1</td><td></td><td></td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td>Объединим информационные и проверочные разряды и получим код Хэмминга.</td><td>$X' = 1\,0\,0\,0\,1\,1\,0\,0\,1\,0\,1$</td></tr>
<tr><td>Проверим правильность кодирования, вычислив значение H(X). Выпишем в столбец двоичные номера строк, в которых знак сообщения X' равен 1. Сумма по модулю 2 знаков номеров в каждом разряде должна быть равна 0.</td><td><table>
<tr><th>№ строки</th><th>Двоичное представление номера строки</th><th>знаки сообщения X'</th></tr>
<tr><td>1</td><td>0001</td><td>1</td></tr>
<tr><td>5</td><td>0101</td><td>1</td></tr>
<tr><td>6</td><td>0110</td><td>1</td></tr>
<tr><td>9</td><td>1001</td><td>1</td></tr>
<tr><td>11</td><td>1011</td><td>1</td></tr>
<tr><td></td><td>0000</td><td></td></tr>
</table></td></tr>
<tr><td>Внесем ошибку замещения в один из разрядов и произведем декодирование. Пусть при передаче сообщения X' произошла ошибка в 7-ом разряде, т.е. полученное сообщение
$X'' = 1\,0\,0\,0\,1\,1\,\mathbf{1}\,0\,1\,0\,1$
Вычислим значение H(X''). Выпишем в столбец двоичные номера строк, в которых знак сообщения X'' равен единице. Суммируем по модулю 2 знаки номеров в каждом разряде. Полученное число 0111 равно номеру разряда 7, в котором произошла ошибка. Заменяя в сообщении X'' значение 7-го разряда на противоположное, восстанавливаем X'. Вычеркивая из X' проверочные разряды получаем искомое сообщение X.</td><td><table>
<tr><th>№ строки</th><th>Двоичное представление номера строки</th><th>знаки сообщения X'</th></tr>
<tr><td>1</td><td>0001</td><td>1</td></tr>
<tr><td>5</td><td>0101</td><td>1</td></tr>
<tr><td>6</td><td>0110</td><td>1</td></tr>
<tr><td>7</td><td>0111</td><td>1</td></tr>
<tr><td>9</td><td>1001</td><td>1</td></tr>
<tr><td>11</td><td>1011</td><td>1</td></tr>
<tr><td></td><td>0111</td><td></td></tr>
</table></td></tr>
</table>

Задание 4.1. Перевести в десятичную систему 4-значное двоичное число $a_{12}\ a_{13}\ a_{14}\ a_{15}$ и прибавить 1. Полученное число определяет одну из перечисленных ниже четверок десятичных цифр. Найти все попарные расстояния между их кодами, применяемыми при автоматическом распознавании цифр почтового индекса.

1) 1, 2, 3, 4;	5) 5, 6, 7, 8;	9) 2, 4, 7, 8;	13) 3, 4, 8, 0;
2) 2, 3, 4, 5;	6) 6, 7, 8, 9;	10) 3, 6, 7, 9;	14) 1, 4, 6, 0;
3) 3, 4, 5, 6;	7) 7, 8, 9, 0;	11) 2, 5, 7, 8;	15) 2, 5, 7, 9;
4) 4, 5, 6, 7;	8) 1, 3, 5, 7;	12) 1, 5, 6, 9;	16) 3, 5, 7, 8.

Задание 4.2. Перевести в десятичную систему 4-значное двоичное число a_{15} a_{16} a_{17} a_{18} прибавить 1. Полученное число k определяет столбец в нижеследующей таблице. Построить код Фано для алфавита {a, b, c, d, e, f} с набором частот, представленным в столбце с номером k. Вычислить стоимость L полученного кода и построить его кодовое дерево.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a:	0.12	0.22	0.32	0.08	0.04	0.09	0.16	0.07	0.11	0.17	0.14	0.13	0.08	0.18	0.20	0.28
b:	0.27	0.11	0.10	0.28	0.12	0.18	0.07	0.25	0.29	0.22	0.18	0.22	0.20	0.28	0.17	0.04
c:	0.06	0.30	0.20	0.06	0.11	0.11	0.28	0.31	0.17	0.06	0.06	0.16	0.31	0.06	0.10	0.12
d:	0.10	0.15	0.18	0.25	0.28	0.27	0.20	0.17	0.03	0.20	0.30	0.31	0.13	0.27	0.24	0.12
e:	0.11	0.13	0.12	0.18	0.13	0.20	0.17	0.14	0.10	0.11	0.10	0.08	0.18	0.14	0.07	0.33
f:	0.34	0.09	0.18	0.15	0.32	0.15	0.12	0.06	0.30	0.24	0.22	0.10	0.10	0.07	0.22	0.11

Задание 4.3. Для сообщения $X = a_{27}\, a_{28}\, a_{29}\, a_{30}\, a_{31}\, a_{32}\, a_{33}\, a_{34}\, a_{35}\, a_{36}\, a_{37}\, a_{38}$

построить код Хэмминга. Перевести в десятичную систему 3-значное двоичное число a_{24} a_{25} a_{26} прибавить 4. Полученное число k определяет разряд, в которой нужно ввести одиночную ошибку замещения. Произведя декодирование, подтвердите место ошибки.

Контрольный тест

1. **В коде a: 01; b: 100; c: 101 словом 10101100 закодировано сообщение**

   A) cac

   B) cca

   C) cba

   D) cab

   E) ccc

2. **При передаче сообщения 0100101 произошла ошибка вида 0 $\to \Lambda$ в 4-ом разряде. На приемнике получено сообщение**

   A) 010101

   B) 010001

   C) 0101001

   D) 000011

   E) 0101101

3. **Из кодов, представленных в таблице, префиксным(и)**

A) являются (1) и (3)

B) являются (1), (2), (3)

C) являются (2) и (3)

D) ни один не является

E) являются (1) и (2)

(1) a: 1001 b: 110 c: 100 d: 00	(2) a: 100 b: 1010 c: 00 d: 11010	(3) a: 001 b: 10 c: 11 d: 000

4. **При передаче сообщения 10110001 произошла ошибка вида 1 $\to$ Λ в 3-м разряде и вида 0 $\to$ 1 в 6-м разряде. На приемнике получено сообщение**

A) 10110011

B) 1010101

C) 10110001

D) 1010001

E) 11110001

5. **Значения компонент с номерами 2^0, 2^1,..., 2^{n-1} в коде Хэмминга соответствуют**

A) проверочным позициям

B) информационным позициям

C) нечетным позициям

D) двоичным позициям

E) четным позициям

5. ЛОГИЧЕСКИЕ СЕТИ И АВТОМАТЫ

5.1. Логические сети

Здесь мы будем рассматривать схемы из функциональных элементов, реализующих булевы функции. В силу полноты ряда систем функций мы будем использовать только некоторые из них:

конъюнктор - элемент с двумя входами, реализующий конъюнкцию;

дизъюнктор - элемент с двумя входами, реализующий дизъюнкцию;

инвертор - элемент с одним входом, реализующий отрицание.

элементарный сумматор, реализующий сумму по модулю 2.

Расширим понятие схемы из функциональных элементов. Определим ее функционирование в дискретном времени t = 0, 1, 2,... Считается, что в каждый момент времени на входы схемы подаются значения аргументов: для n-входовой схемы

$$X(t) = (x(t), x_2(t), \dots, x_n(t)).$$

Введем новый тип элемента с одним входом, **D –триггер** (*delay* – задержать) который задерживает вход на один такт. Будем обозначать символом “D”: если x(t) - входной аргумент элемента задержки, то D(t+1) = x(t). Доопределим значение D(0) = 0.

Включение элементов задержки в схему позволяет связывать значения, подаваемые на входы и вычисляемые внутри схемы (как выходные, так и промежуточные) в разные моменты времени.

Определение 1. Схема из функциональных элементов и задержек, действующая в тактовом (дискретном) режиме, называется **логической сетью**. При этом допустимы ориентированные циклы (контуры), если в них присутствует хотя бы один элемент задержки.

5.2. Конечные автоматы

1. **Конечным автоматом** называется система (A, B, Q), где $A = \{a_1, a_2, \dots, a_p\}$ - входной алфавит, $B = \{b_1, b_2, \dots, b_r\}$ - выходной алфавит, $Q = \{q_1, q_2,$

... , q_w } - множество внутренних состояний (q_1 называется начальным состоянием); функция выходов $\varphi : Q\times A\rightarrow B$ и функция переходов $\psi: Q\times A\rightarrow Q$ определены на множестве $Q\times A$.

Функционирование автомата происходит в тактовом дискретном режиме для t = 0, 1, 2,... На вход автомата поступает входная последовательность a(0), a(1), a(2),... $a(i)\in A$). Пара функций φ и ψ образует **системy канонических уравнений**:

$$b(t) = \varphi (q(t), a(t)),$$

$$q(t+1) = \psi (q(t), a(t)),$$

при этом считается, что $q(0) = q_1$.

Первое уравнение задает выходные значения в момент времени t как функцию входного значения в тот же момент времени и текущего внутреннего состояния автомата. Второе уравнение определяет переход в очередное внутреннее состояние, в котором окажется автомат в следующий момент времени. Совокупность (q(t), a(t)), полностью определяющая текущее значение выхода и следующее внутреннее состояние автомата, называют полным состоянием автомата.

Поскольку функции и определены на конечных множествах, их можно задать таблицами, которые обычно сводятся в одну таблицу для их декартова произведения $\varphi\times\psi :Q\times A \rightarrow Q\times B$, называемую **таблицей (или матрицей) переходов автомата**; она имеет w строк и p столбцов.

Автомат можно задавать также **графом** (или **диаграммой) переходов** – ориентированным графом (Q, E), где Q - множество вершин, а дуги соответствуют клеткам таблицы переходов: если на пересечении строки q_i и столбца a_j в таблице содержится пара (q_k / b), то этой клетке соответствует дуга (q_i , q_k), которой приписана упорядоченная пара (a_j; b).

Из каждой вершины графа переходов исходят p дуг, соответствующих столбцам таблицы переходов. Однако для большей наглядности несколько кратных дуг, которым приписаны пары (a_{i_1} ; b), ... , (a_{i_r} ; b) с одинаковым выходным значением, могут быть склеены в одну дугу, которой приписывается кортеж (a_{i_1} , ... , a_{i_r} ; b).

Задание 5.2.1 Состояние автомата закодировано двузначными двоичными числами; входы и выходы автомата – булевы переменные. Для автомата A:

(1) построить граф переходов;

(2) вычислить 7 первых выходных значений *b(t)* для *t* = 1, 2,..., 6 при заданных входных значениях *a(t)*.

A_0 :

X =	0	1
q_1 : 01	$a_1\, a_2 / a_3$	$a_4\, a_5 / a_6$
q_2 : 10	$a_7\, a_8 / a_9$	$a_{10}\, a_{11} / a_{12}$
q_3 : 11	$a_{13}\, a_{14} / a_{15}$	$a_{16}\, a_{17} / a_{18}$
q_4 : 00	$a_{19}\, a_{20} / a_{21}$	$a_{22}\, a_{23} / a_{24}$

$a(t) = a_{30}, a_{31}, a_{32}, a_{33}, a_{34}, a_{35}, a_{36}$.

Задание 5.2.2. Состояние автомата закодировано двузначными двоичными числами; входы и выходы автомата – булевы переменные. Для автомата A:

(1) построить граф переходов;

(2) вычислить 5 первых выходных значений *b(t)* для *t* = 1, 2,..., 5 при заданных входных значениях *(X(t), Y(t))*.

A_1 :

X, Y =	0 0	0 1	1 0	1 1
q_1 : 1	$a_7 / a_8\, a_9$	$a_{10} / a_{11}\, a_{12}$	$a_{13} / a_{14}\, a_{15}$	$a_{16} / a_{17}\, a_{18}$
q_2 : 0	$a_{19} / a_{20}\, a_{21}$	$a_{22} / a_{23}\, a_{24}$	$a_{25} / a_{26}\, a_{27}$	$a_{28} / a_{29}\, a_{30}$

$X(t) = a_{35}, a_{36}, a_{37}, a_{38}, a_{39}, a_{40}$

$Y(t) = a_{40}, a_{41}, a_{42}, a_{43}, a_{44}, a_{45}$

Задание 5.2.3. Перевести в десятичную систему 4-значное двоичное число $a_{15}\, a_{16}\, a_{17}\, a_{18}$, прибавить 1. Полученное число k определяет автомат A_k (k = 1, 2,..., 16) с одним входом X, задаваемый условием:

$$Z(t) = \begin{cases} 1, \textit{ если выполнено условие } P_k \\ 0 - \textit{в противном случае} \end{cases}$$

P_1: $X(t\text{-}2) = X(t\text{-}1)$

P_2: $X(t\text{-}2) \neq X(t\text{-}1)$

P_3: $X(t\text{-}2) = X(t\text{-}1) = 0$

P_4: $X(t\text{-}2) = X(t\text{-}1) = 1$

P_5: $X(t\text{-}2) \leq X(t\text{-}1)$

P_6: $X(t\text{-}2) \geq X(t\text{-}1)$

P_7: $(X(t\text{-}2) = 0)\ \&\ (X(t\text{-}1) = 1)$

P_8: $(X(t\text{-}2) = 1)\ \&\ (X(t\text{-}1) = 1)$

P_9: $X(t\text{-}2) = X(t)$

P_{10}: $X(t\text{-}2) \neq X(t)$

P_{11}: $X(t\text{-}2) = X(t) = 0$

P_{12}: $X(t\text{-}2) = X(t) = 1$

P_{13}: $X(t\text{-}2) \geq X(t)$

P_{14}: $X(t\text{-}2) \leq X(t)$

P_{15}: $(X(t\text{-}2) = 1)\ \&\ (X(t) = 0)$

P_{16}: $(X(t\text{-}2) = 0)\ \&\ (X(t) = 1)$

Для автомата A_k:

1) построить граф переходов автомата;

2) построить таблицу переходов автомата;

3) закодировать состояния автомата двоичными наборами;

4) составить табличное представление канонических уравнений автомата;

5) составить канонические уравнения автомата в виде формул, использующих булевы функции;

6) построить логическую сеть, реализующую автомат.

Указание: Различные состояния автомата A_k соответствуют различным входным значениям в моменты времени (t - 1) и (t - 2).

Примеры логических сетей

1. Выходное значение логической сети Z(t) определяется равенством

$Z(t) = (X(t) \oplus Y(t))\ \&\ D(Y(t))$.

Построить логическую сеть и вычислить первые 5 значений Z(t), если заданы X(t) и Y(t). В этой логической сети нет контуров (рис 5.1). Результаты работы логической сети запишем в табличном виде

Таблица 5.1.

t	0	1	2	3	4	5	…
X	0	1	1	1	0	0	…
Y	1	0	1	0	1	1	…
Z	0	1	0	1	0	1	…

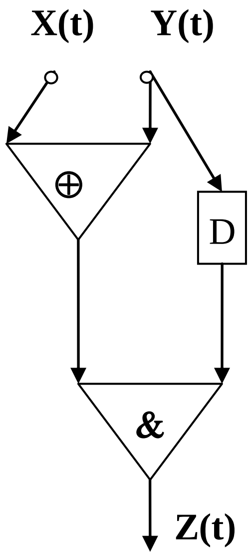


рис 5.1

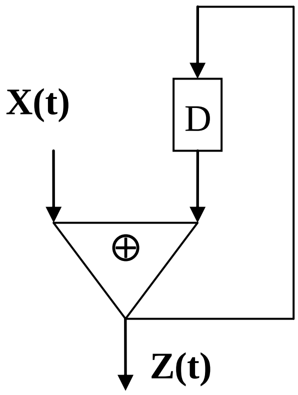


рис 5.2

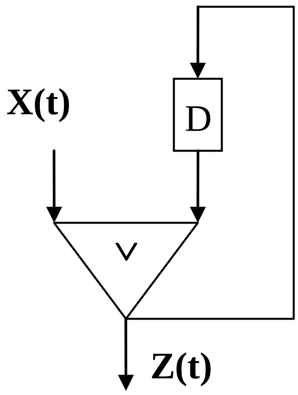


рис 5.3

2. Z(t) = X(t) ⊕ з(t) = X(t) ⊕ Z(t-1) (рис. 5.2).

Z(t) = X(t) ⊕ Z(t-1) = X(t) ⊕ X(t-1) ⊕ Z(t-2) = ... = X(t) ⊕ X(t-1) ⊕ ... ⊕ X(0) = $\bigoplus\limits_{\tau=0}^{t} X(\tau)$. Итак, выходное значение меняется после каждой следующей входной единицы, или, по-другому, Z(t) определяется четностью числа единиц в начальном отрезке входной последовательности. Можно, например, интерпретировать автомат как некоторый переключатель (выключатель). «Нажатие кнопки определяет появление значения X(t) = 1, и переводит состояние Z(t) из одного состояния в другое. Как бы включается и выключается»

3. Z(t) = X(t) ∨ з(t) = X(t) ∨ Z(t-1). (рис. 5.3).

Аналогично, Z(t-1) = X(t-1) ∨ Z(t-2). Продолжая эту рекурсию, получаем:

Z(t) = X(t) ∨X(t-1) ∨X(t-2) ∨ ... ∨X(0) = $\bigvee\limits_{\tau=0}^{t} X(\tau)$.

Исходя из определения дизъюнкции $Z(t) = \begin{cases} 0, \ \textit{если} \ \forall_{\tau \le t} : X(\tau) = 0, \\ 1, \ \textit{в противном случае,} \\ \quad \textit{т.е. если} \ \exists_{\tau \le t} : X(\tau) = 1. \end{cases}$

Таким образом, $Z(t)$ тождественно равно 0 до появления на входе первой единицы, после чего тождественно равно 1.

Пример конченого автомата.

Пусть конечный автомат с алфавитами A = {a, b, c}, B = {0, 1}, Q = {q_1, q_2, q_3, q_4} задается таблицей переходов (Таблица 5.2). задает функции переходов и выходов для автомата символы q_i и b_j разделены знаком /.

Таблица 5.2

	a	b	c
q_1	q_3 / 0	q_3 / 1	q_2 / 0
q_2	q_4 / 0	q_1 / 0	q_1 / 0
q_3	q_2 / 0	q_3 / 0	q_3 / 0
q_4	q_4 / 0	q_2 / 1	q_1 / 1

Граф переходов для таблицы 5.2 изображен на рис. 5.4.

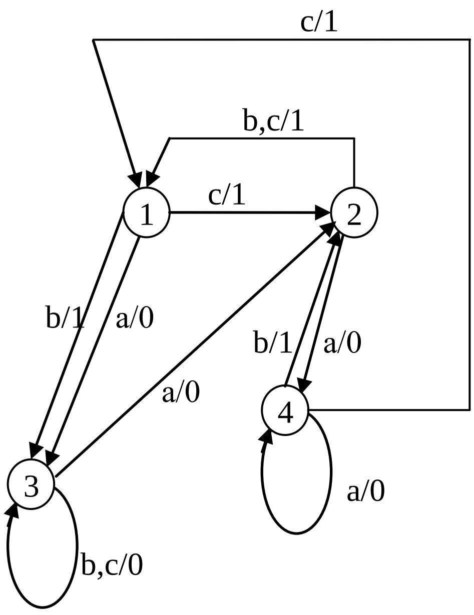


Рис 5.4.

4. Автомат A с одним входом X, задаваемый условием:

$$Z(t) = \begin{cases} 1, & \text{при} \quad (X(t-1) = 1) \wedge (X(t) = 0) \\ 0, & \text{в противномслучае} \end{cases}$$

Закодируем возможные состяния автомата двоичными наборами. Поскольку все зависит от того какое значение было в предыдущем значении времени, т.е. X(t-1), а его возможные значения – это 0 или 1, то ограничимся двумя состояниями

X(t-1)	
0	q_1
1	q_2

Тогда действие автомата можно записать таблицей переходов:

	0	1
q_1	q_1 / 0	q_2 / 0
q_2	q_1 / 1	q_2 / 0

Соответствующий граф переходов будет следующим

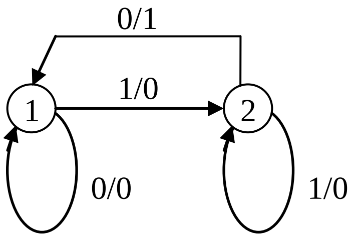


Рис 5.5.

Канонические уравнения автомата в виде формул, использующих булевы функции, будут иметь вид:

$$Z(t) = X(t-1) \wedge \overline{X}(t)$$

Тогда логическую сеть, реализующая автомат будет следующим Рис 5.6.

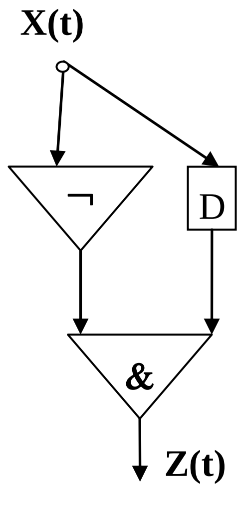


Рис 5.6.

Контрольный тест

1. **Число вершин в графе переходов автомата с входным алфавитом {a, b, c}, выходным алфавитом {d, e} и 4 состояниями равно**

A) 4
B) 24
C) 12
D) 5
E) 13

2. **Матрица переходов автомата с входным алфавитом {a, b, c}, выходным алфавитом {d, e} и 3 состояниями имеет размерность**

A) 3x4
B) 5x3
C) 4x3
D) 3x3
E) 4x4

3. **В логической сети выход элемента задержки может быть присоединен к (1) выходу другого элемента задержки, (2) входу функционального элемента, (3) входу другого элемента задержки, (4) выходу сети. Верными являются утверждения**

A) (1), (2), (4)
B) (1), (2)
C) (1), (3)
D) (2), (3), (4)
E) (1), (3), (4)

4. **Число вершин в графе переходов автомата с входным алфавитом {a, b, c}, выходным алфавитом {c, d} и 5 состояниями равно**

A) 9
B) 20
C) 30
D) 5
E) 25

5. **Канонические уравнения автомата выражают внутреннее состояние автомата в следующий момент через**

A) текущее значение на входе и предыдущее внутреннее состояние
B) предыдущее значение на входе и текущее внутреннее состояние
C) текущее значение на входе и текущее внутреннее состояние
D) текущее значение на входе и последующее внутреннее состояние
E) предыдущее значение на входе и предыдущее внутреннее состояние

6. ЭЛЕМЕНТЫ КОМБИНАТОРИКИ

6.1. Основные комбинаторные конфигурации

Комбинаторная конфигурация - это расположение конечного множества элементов, удовлетворяющее ряду специальных свойств. К основным комбинаторным конфигурациям относятся *сочетания, размещения и перестановки.* Ряд других конфигураций может быть сведен к ним.

Набор элементов ($x_{i_1}, x_{i_2}, \ldots, x_{i_k}$), составленный из элементов множества X = ($x_1, x_2, \ldots x_n$), называется **выборкой объема k из n элементов**, или (**n, k**)-**выборкой**. Выборки, различающиеся составом элементов, всегда считаются различными.

Выборка называется **упорядоченной**, если порядок элементов в ней задан (т.е. она представляет из себя кортеж). Две упорядоченные выборки, различающиеся только порядком следования элементов, считаются различными. Если порядок элементов в выборке несуществен, то выборка называется **неупорядоченной**.

Упорядоченная (n, k)-выборка, в которой элементы могут повторяться, называется (**n, k**)-**размещением с повторениями**, или **размещением с повторениями из n элементов по k**. Если элементы (n, k)-выборки попарно различны, то она называется (**n, k**)-**размещением без повторений**, или **размещением без повторений из n элементов по k**.

(**n, n**)-**размещения без повторений** называются **n-перестановками**, или **перестановками из n элементов**.

Неупорядоченные (n, k)-выборки называются **сочетаниями: с повторениями** или **без повторений**.

В комбинаторике выделяются два основных правила:

1. Пусть X - конечное множество из n элементов. Тогда говорят, что один объект из X можно выбрать n способами, и пишут |X|=n Если X и Y -

непересекающиеся множества |X|=n и |Y|=m то |X+Y|=m+n. Свойство может быть распространено на большее число множеств.

Пусть $\{X_i\}$ - система попарно не пересекающихся множеств: $X_i \cap X_j = \varnothing$, если $i \neq j$. Тогда

$$\left|\bigcup_{i=1}^{n} X_i\right| = \sum_{i=1}^{n} |X_i|.$$

Это **правило суммы**.

Если объект $x \in X$ может быть выбран n способами и после каждого из таких выборов объект $y \in Y$ может быть выбран m способами, то выбор упорядоченной пары (x, y) может быть осуществлен mn способами. Это **правило произведения**. Оно также распространяется на большее число множеств. Для системы множеств $\{X_i\}$, i = 1, 2,..., k, где $|X_i| = m_i$, выбор упорядоченной последовательности из k объектов $(x_1, x_2, \ldots, x_k)$, $x_i \in X_i$, может быть осуществлен $m_1 \times m_2 \times \ldots \times m_k$ способами.

6.2. Формулы пересчета числа комбинаторных конфигураций

1. Число (n, k)-размещений без повторений A_n^k может быть определено с помощью правила произведения. Первый из k элементов размещения может быть выбран n способами, второй - (n - 1) способами, поскольку элемент, выбранный первым, не должен быть повторен; аналогично, для третьего (если k>2) элемента остается (n-2) способов и т.д. Всего k элементов могут быть выбраны =n(n–1)...(n–k+1) способами: произведение k убывающих на 1 сомножителей, начиная с n. По-другому, используя обозначение $n! = 1 \cdot 2 \ \ldots \cdot n$, можно записать:

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

Число n-перестановок P_n равно $n!$.

(n, k)-размещения с повторениями - это кортежи (по-другому, слова) длины k в n-элементном алфавите. Используя правило произведения, нетрудно показать, что число различных (n, k)-размещений с повторениями

$$\hat{A}_n^k = n^k.$$

2. Числа (n, k)-сочетаний без повторений обозначаются символами

$$C_n^k = \frac{n!}{k!\cdot(n-k)!} \text{ или } \binom{n}{k}$$

и называются также **биномиальными коэффициентами**, поскольку совпадают с коэффициентами формулы бинома Ньютона для *n*-й степени двучлена x + y:

$$(x+y)^n = \sum_{k=0}^{n} C_n^k \cdot x^k \cdot y^{n-k}.$$

Число (n, k)-сочетаний с повторениями вычисляются по формуле:

$$\hat{C}_n^k = C_{n+k-1}^k = C_{n+k-1}^{n-1}$$

3. Рассмотрим всевозможные размещения с повторениями $a_1a_2...a_n$ из n элементов с дополнительным условием: в каждом из них b_1 элементов первого рода, b_2 элементов второго рода и вообще b_i элементов i-го рода (i = 1, 2,..., r). Естественно, выполнено равенство $b_1 + b_2 +...+ b_r = n$. Заменим b_i элементов i-го рода различающимися элементами $b_{i_1},..., b_{i_r}$ так, чтобы все элементы стали различными, и получим n! перестановок. Каждое исходное размещение дает $b_1!\, b_2! \,...\, b_r!$ перестановок. Следовательно, число исходных размещений равно

$$\frac{n!}{b_1!\cdot b_2!\cdot\ldots\cdot b_r!}, \quad \text{где } b_1 + b_2 + \ldots + b_r = n.$$

Это число называется **полиномиальным коэффициентом**: оно равно коэффициенту при произведении в разложении по степеням переменных полинома $(x_1 + x_2 +...+ x_r)^n$. Биномиальные коэффициенты представляют частный случай при r = 2.

Задание 6.1. Перевести в десятичную систему двоичное число $a_5\, a_6\, a_7\, a_8$ и прибавить N [3]. Если полученное число больше 20, то нужно отнять 20. Полученное число i определяет номер варианта.

Для заданных n и m б вычислите $\hat{A}_n^k$, A_n^m, , $\hat{C}_n^k$ C_n^m, P_n.

1. n =5, m =3	6. n =6, m =3	11. n =7, m =3	16. n =7, m =4
2. n =6, m =4	7. n =5, m =3	12. n =4, m =2	17. n =8, m =2
3. n =5, m =4	8. n =5, m =3	13. n =8, m =6	18. n =4, m =3
4. n =6, m =2	9. n =5, m =3	14. n =7, m =5	19. n =7, m =2
5. n =5, m =2	10. n =6, m =5	15. n =7, m =4	20. n =8, m =3

Задание 6.2. Перевести в десятичную систему двоичное число $a_{16}\, a_{17}\, a_{18}\, a_{19}$ и прибавить 1. Полученное число i определяет номер варианта.

Из данной пропорции найти x и y

№	**пропорция**
1	$C_{x+1}^{y+1} : C_{x+1}^{y} : C_{x+1}^{y-1} = 5:4:2$
2	$C_{x}^{y+1} : C_{x}^{y} : C_{x}^{y-1} = 3:3:2$
3	$C_{x}^{y+2} : C_{x}^{y+1} : C_{x}^{y} = 42:35:20$
4	$C_{x+1}^{y+2} : C_{x+1}^{y+1} : C_{x+1}^{y} = 3:4:3$
5	$C_{x+1}^{y+1} : C_{x+1}^{y} : C_{x+1}^{y-1} = 4:5:4$
6	$C_{x}^{y+1} : C_{x}^{y} : C_{x}^{y-1} = 21:14:6$
7	$C_{x}^{y+2} : C_{x}^{y+1} : C_{x}^{y} = 3:5:5$
8	$C_{x+1}^{y+2} : C_{x+1}^{y+1} : C_{x+1}^{y} = 2:4:5$
9	$C_{x+1}^{y+1} : C_{x+1}^{y} : C_{x+1}^{y-1} = 2:3:3$
10	$C_{x}^{y+1} : C_{x}^{y} : C_{x}^{y-1} = 14:8:3$
11	$C_{x}^{y+2} : C_{x}^{y+1} : C_{x}^{y} = 5:3:1$
12	$C_{x+1}^{y+2} : C_{x+1}^{y+1} : C_{x+1}^{y} = 5:6:5$
13	$C_{x+1}^{y+1} : C_{x+1}^{y} : C_{x+1}^{y-1} = 14:7:2$
14	$C_{x}^{y+1} : C_{x}^{y} : C_{x}^{y-1} = 6:14:21$
15	$C_{x}^{y+2} : C_{x}^{y+1} : C_{x}^{y} = 24:9:2$
16	$C_{x}^{y+2} : C_{x}^{y+1} : C_{x}^{y} = 5:3:1$

[3] Здесь, как и ранее порядковый номер по списку в журнале.

Задание 6.3. Перевести в десятичную систему двоичное число a_{16} a_{17} a_{18} a_{19} и прибавить 1. Полученное число i определяет номер варианта.

Пусть n и m - целые положительные числа. С использованием тождества

$$(1+t)^n = \sum_{k=0}^{n} \binom{n}{k} t^k$$

или иным способом доказать следующие равенства [2, с. 259]

1) $\sum_{k=0}^{n} \binom{n}{k} = 2^n$; 2) $\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0$; 3) $\sum_{k=1}^{n} k\binom{n}{k} = n2^{n-1}$;

4) $\sum_{k=2}^{n} k(k-1)\binom{n}{k} = n(n-1)2^{n-2}$; 5) $\sum_{k=0}^{n} (2k+1)\binom{n}{k} = (n+1)2^n$;

6) $\sum_{k=0}^{n} \frac{1}{k+1}\binom{n}{k} = \frac{1}{n+1}(2^{n+1}-1)$; 7) $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1}\binom{n}{k} = \frac{1}{n+1}$;

8) $\sum_{k=1}^{n} \frac{(-1)^{k-1}}{k}\binom{n}{k} = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$;

9) $\sum_{r=0}^{k} \binom{m}{r}\binom{n}{k-r} = \binom{n+m}{k}$; 10) $\sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n}$;

11) $\sum_{k=0}^{n} \frac{(2n)!}{(k!)^2((n-k)!)^2} = \binom{2n}{n}^2$; 12) $\sum_{k=0}^{n}\sum_{r=0}^{n-k} \binom{n}{k}\binom{n-k}{r} = 3^n$;

13) $\sum_{r=k}^{n} (-1)^{k-r}\binom{n}{r} = \sum_{r=0}^{n-k} (-1)^{n-k-r}\binom{n}{r}$;

14) $\sum_{r=0}^{n-k} \binom{n}{k+r}\binom{m}{r} = \binom{m+n}{n-k}$;

15) $\sum_{k=n}^{m} (-1)^{k-n}\binom{k}{n}\binom{m}{k} = \begin{cases} 0 & \text{при} \quad m \neq n, \\ 1 & \text{при} \quad m = n. \end{cases}$

Контрольный тест

1. **Установите соответствие**

$A_n^k = \dfrac{n!}{(n-k)!}$		полиномиальный коэффициент
$\hat{A}_n^k = n^k$		число (n, k)-сочетаний без повторений
$\dfrac{n!}{b_1! \cdot b_2! \cdot \ldots \cdot b_r!}$,		число (n, k)-размещений без повторений
$(x+y)^n = \sum_{k=0}^{n} C_n^k \cdot x^k \cdot y^{n-k}$.		Бином Ньютона
$C_n^k = \dfrac{n!}{k! \cdot (n-k)!}$		число (n, k)-размещений с повторениями
$P_n = n!$		число (n, k)-сочетаний с повторениями
$\hat{C}_n^k = C_{n+k-1}^k = C_{n+k-1}^{n-1}$		число перестановок

2. **Формула** $\left|\bigcup_{i=1}^{n} X_i\right| = \sum_{i=1}^{n} |X_i|$. **называется**
   A) Правилом суммы
   B) Правилом произведений
   C) Правилом разности
   D) Правилом упорядочивания
   E) Правилом деления

3. **Число** $A_5^3 =$
   A) 15 B) 5 C) 6 D) 10 E) 20

4. **Число** $C_5^3 =$
   A) 15 B) 5 C) 6 D) 10 E) 20

5. **В электронной библиотеке имеется 10 статей по заданной теме. Сколько есть различных вариантов записи на диск по три различных статьи?**
   A) 1000 B) 100 C) 120 D) 140 E) 10000

7. АЛГОРИТМЫ И ВЫЧИСЛИМОСТЬ

7.1 Рекурсивные функции

Числовой функцией называется функция вида $y = f(x_1,..., x_n)$, где $x_1,..., x_n$, $y \in N^0$, где $N^0 = \{0, 1,..., n,...\}$ - множество натуральных (целых неотрицательных) чисел. Такие функции называются **числовыми**. ; таким образом, и аргументы, и функция принимают натуральные значения.

Следующие функции называются **исходными**:

1) $Z(x) = 0$ - нуль-функция (“zero”), т.е. константа 0;

2) $N(x) =$ “$x + 1$” - функция следования (“next”), обозначение: x';

3) $I_k(x1,..., xn) = x_k$ - селекторные функции, где $k = 1, 2,..., n$.

Замечание: Функция N(x) выражает число, следующее в натуральном ряду за x. Она представляет собой функцию одной переменной, поэтому запись ее с помощью знака “+” служит только для пояснения (что такое сумма - еще не определено, см. ниже). Каждая из функций (x) равна одному из своих аргументов: индекс k выделяет k-ю переменную.

Говорят, что функция h получается применением **оператора суперпозиции S** к функциям $f, g_1,..., g_n$ (обозначение: $h = S(f, g_1,..., g_n)$), если

$$h(x_1,..., x_m) = f(g_1(x_1,..., x_m),..., g_n(x_1,..., x_m)).$$

7.2. Примитивно рекурсивные функции.

Говорят, что функция f от (n+1) переменных получается применением **оператора примитивной рекурсии R** к функциям g от n переменных и h от (n+2) переменных (обозначение: $f = R(g, h)$), если

1) $f(x_1,..., x_n, 0) = g(x_1,..., x_n)$,

2) $f(x_1,..., x_n, k+1) = h(x_1,..., x_n , k, f(x_1,..., x_n, k))$, $k \in N^0$.

Рекурсия в этом определении ведется по последней переменной функции f и выражает встречавшийся ранее индуктивный способ определения: зная значение функции при $k = 0$, мы можем вычислить значение функции для $k = 1$, затем для $k = 2$ и т.д.

Функция называется **примитивно рекурсивной** (сокращенно **п/р**), если она может быть получена из исходных применением конечного числа операторов суперпозиции и примитивной рекурсии.

Отметим, что п/р функции определены при всех $x \in N^0$.

Рассмотрим некоторые примеры примитивно рекурсивных функций.

Функции, выражающие основные арифметические действия сложения и умножения, являются п/р.

А) Сумма $S(x, y) = x + y$ может быть определена так:

1) $S(x, 0) = x$.

Точнее, в обозначениях определения 1, $S(x, 0) = I_1(x)$,

2) $S(x, y + 1) = N(S(x, y)) = (S(x, y))'$, т.е. сумма $x + (y + 1)$ равна числу, следующему за $(x + y)$: $(x + y) + 1$.

В) Произведение $P(x, y) = x \cdot y$ может быть определено так:

1) $P(x, 0) = Z(x)$, т.е. $x \cdot 0 = 0$

2) $P(x, y + 1) = S(P(x, y), x)$, т.е. $x \cdot (y + 1) = x \cdot y + x$

7.3. Оператор минимизации.

Свойство 1. Если $f(x_1,..., x_n, y)$ - п/р функция, то ограниченная сумма $g(x_1,..., x_n, a) = \sum\limits_{y<a} f(x_1,..., x_n, y)$, при $f(x_1,..., x_n, 0) = 0$, тоже является п/р функцией.

Свойство 2.Если $f(x_1,..., x_n, y)$-п/р функция, то ограниченное произведение $g(x_1,..., x_n, a) = \prod\limits_{y<a} f(x_1,..., x_n, y)$, при $f(x_1,..., x_n, 0) = 1$, тоже является п/р функцией.

Результатом применения ***операция минимизации*** по i-той переменной функции $g(x_1,..., x_n)$ будет функция $f(x_1,..., x_{i-1}, y, x_{i+1}..., x_n) = \mu_y(g(x_1,..., x_{i-1}, y, x_{i+1}..., x_n) = x_i)$ определяемая так:

Соотношение $g(x_1,..., x_{i-1}, y, x_{i+1}..., x_n) = x_i$ *(*)*,будем рассматривать, как уравнение относительно *y*.

Это уравнение будем решать подбором, подставляя вместо у последовательно числа 0,1,2, и т.д. Возможны случаи:

1) на некотором шаге левая часть соотношения (*) не определена. В этом случае считаем, что на наборе (x_1,..., x_n) операция минимизации, а значит и функция f(x_1,..., x_{i-1},у, x_{i+1}..., x_n) не определена.

2) На каждом шаге левая часть соотношения (*) определена, но ни при каких у равенство не выполнится. В этом случае также считаем, что на наборе (x_1,..., x_n) операция минимизации, а значит и функция f(x_1,..., x_{i-1},у, x_{i+1}..., x_n) не определена.

3) Левая часть соотношения (*) определена при $y = 0$, $y = 1$,..., $y = z$-l, $y = z$, но при $y<z$ равенство (*) не выполнялось, а при $y = z$ оно выполняется. В этом случае число z считается значением операции минимизации и f(x_1,..., x_{i-1},у, x_{i+1}..., x_n) на наборе (x_1,..., x_n).

Частично-рекурсивной называется числовая функция, которая может быть получена из исходных с помощью применения конечного числа раз операций суперпозиции, примитивной рекурсии и минимизации.

Оказывается, что понятие частично рекурсивной функции является в некотором смысле алгоритмически достаточным.

Тезис Черча: всякая эффективно вычислимая функция является частично рекурсивной.

Частично рекурсивная функция называется общерекурсивной (о/р), если она определена всюду на N^0.

Свойство 3. Всякая п/р функция является о/р.

П/р, ч/р и о/р функции называются рекурсивными функциями (р. ф.).

7.4. Машины Тьюринга

1. **Машина Тьюринга (МТ или м/т)** задается тройкой T = (A, Q, P), где $A = \{a_1,..., a_n\}$ - внешний алфавит, часто выбирают $A = \{0, 1\}$; $Q = \{q_0, q_1,..., q_m\}$ - внутренний алфавит, элементы которого обозначают состояния управляющего устройства, обычно q_0 - заключительное состояние, q_1 - начальное состояние; P - **программа**, состоящая из команд вида $q_ia_j \rightarrow q_ka_lS$, где S - один из символов сдвига: L (налево), R (направо), E (на месте); $1\leq q_i \leq m$, $0 \leq q_k \leq m$, $a_j, a_l \in A$.

Машина функционирует в дискретном времени t = 0, 1, 2,... следующим образом.

Имеется бесконечная в обе стороны лента, разбитая на ячейки (т.е. множество ячеек, упорядоченное по типу Z - множества целых чисел). В начальный момент в конечном числе ячеек записаны символы из A, в остальных ячейках записан пустой символ, обозначаемый через 0. Машина Тьюринга имеет читающую-пишущую головку, которая в любой момент дискретного времени обозревает некоторую ячейку (т.е. находится в ячейке и воспринимает написанный в ней символ) - рис. 7.1.

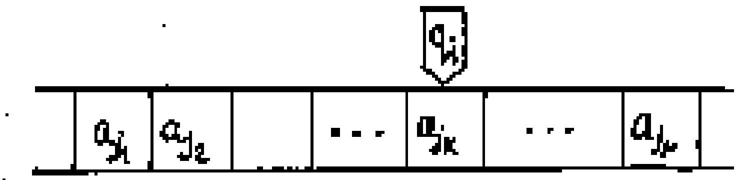


Рис 7.1.

МТ имеет управляющее устройство (автомат), которое в каждый момент времени находится в одном из состояний $q_i \in Q$; q_1 – начальное состояние, q_0 – заключительное состояние.

Конфигурацией (машинным словом) называется слово K вида Uq_iV, где U - слово на ленте левее головки, V - слово правее слова U, начиная с символа, находящегося в той же ячейке, где расположена головка. Стандартная начальная конфигурация – q_1V, а стандартная заключительная - q_0V. Это значит, что в начале работы головка находится в начальном состоянии в самой левой из заполненных ячеек; заполненные ячейки могут не составлять связного куска, а перемежаться незаполненными участками; заключительная конфигурация: головка в заключительном состоянии также находится в самой левой из заполненных ячеек.

Говорят, что машина Т **применима** к слову А, если Т, начав работу с начального состояния q_1А, остановится через конечное число шагов на заключительном состоянии q_0В. Если при этом пишут Т(А) = В. Если Т не останавливается, или остановится посередине какой-то конфигурации, то говорят, что Т **неприменима** к слову А.. М/т считается применимой к множеству слов $\{A_i\}$, если она применима к каждому слову A_i.

Задание 7.1. Перевести в десятичную систему двоичное число a_{26} a_{27} a_{28} a_{29} и прибавить 1. Полученное число i определяет номер варианта.

Найти функцию *f(x,y)* полученную из функций *g(x)* и *h(x,y,z)* с помощью оператора примитивной рекурсии

№	*g(x)*	*h(x,y,z*
1	x^2	xz
2	x	$x + y – z$
3	x	$x + z$
4	0	$y + z +1$
5	x^4	$\sqrt{z} \cdot x^3$
6	1	z^{y+1}
7	2	x/z
8	$x + 1$	xyz

№	*g(x)*	*h(x,y,z*
9	x	$(z + y)^2$
10	$2x$	$x + 2z$
11	0	$x – y + 3z$
12	x	$zy + x$
13	x^2	$4y + z$
14	1	$2x + 3y – z$
15	0	$x + z +$y
16	$2x$	$2x + z$

Задание 7.2. Перевести в десятичную систему двоичное число a_6 a_7 a_8 a_9 и прибавить 1. Полученное число i определяет номер варианта.

Найти функции, получаемые из данной функции $g(x_1,..., x_n)$ с помощью операции минимизации по k–той переменной

№	k	n	$g(x_1,\ldots, x_n)$		№	k	n	$g(x_1,\ldots, x_n)$
1	2	4	$x_1 x_2 + x_3$		9	3	4	$\sqrt{x_2} + 2x_3 + x_1^2 + x_4$
2	2	3	$(x_1)^2 + 1/x_3$		10	1	3	$(x_1)^5 - x_2 + 2x_3$
3	1	3	$x_1 (x_2 + 2)/ x_3$		11	2	3	$(x_1 - x_2)^{1/2} + 2x_3$
4	3	3	$x_2 - 2x_3$		12	1	2	$x_1^{x_2}$
5	1	2	$(x_1)^2 + 4x_2$		13	2	3	$(x_1 - 2x_2)/ x_3$
6	2	3	$x_1 - x_2 + 2x_3$		14	1	2	$2x_1x_2$
7	3	3	$x_1 - 2x_2/ x_3$		15	2	3	$(x_1 + 2x_2)^3 + x_3$
8	2	2	$3(x_1 - x_2)$		16	1	3	$x_1 - x_2 + x_3$

Задание 7.3.

Перевести в десятичную систему двоичное число $a_{11}\ a_{12}\ a_{13}\ a_{14}$ и прибавить 1. Полученное число i определяет номер варианта.

Составить программу для машины Тьюринга которая выполняет заданное действие. Внешний алфавит A = $\{a,b,c\}$

1. Приписывает слева к заданному слову V символ a : $V \to aV$
2. Приписывает справа к заданному слову V символы ab: $V \to aV$
3. Заменяет на b 2-й и 3-й символы, если они есть. Если нет, то ничего не меняет.
4. Удаляет первый символ слова V
5. Удаляет последний символ слова V
6. Удаляет все символы c если они есть.
7. Если слово V совпадает с сочетанием ab , то ничего не делает, в противном случае удаляет слово.
8. Если в слове V есть символ c, то все символы a заменяет на b, *иначе* выводит один символ c.
9. Если слово V имеет четную длину, то выводит 1, иначе 0.
10. Оставить средний символ слова V нечетной длины.
11. Удалить из слова V сочетания ab, если они есть. Если таких нет, то вывести 0.

12. Приписать слева слова *V*, его первый символ. если слово *V* было пустым, то вывести *abc*.

13. Удвоить слово *V*, поставив между ним и его копией знак «=»

14. За первым словом непустого слова *V* поставить символ *a*.

15. Удалить из слова *V* последнее вхождение символа *a*.

16. Если первый и последний символы одинаковы, то вывести 1 иначе 0.

Контрольный тест

1. **Значение суперпозиции I_1 (N(6), Z(3)) исходных п/р функций и констант 6, 3 равно**

   A) 5 B) 3 C) 0 D) 7 E) 6

2. **Тезис Черча**
   A) перечисляет основные требования к точному понятию алгоритма
   B) устанавливает сводимость рекурсивных функций к машинам Тьюринга
   C) устанавливает сводимость машин Тьюринга к рекурсивным функциям
   D) декларирует связь интуитивного понятия алгоритма с рекурсивными функциями

3. **Тезис Тьюринга**
   A) перечисляет основные требования к точному понятию алгоритма
   B) устанавливает сводимость рекурсивных функций к машинам Тьюринга
   C) устанавливает сводимость машин Тьюринга к рекурсивным функциям
   D) декларирует связь интуитивного понятия алгоритма с машинами Тьюринга

4. **М/т неприменима к конфигурации К в том случае, если**
   A) правая часть всех команд ее программы содержит символ, не присутствующий в К
   B) левая часть некоторых команд ее программы содержит символ, не присутствующий в К
   C) левая часть всех команд ее программы содержит символ, не присутствующий в К
   D) правая часть некоторых команд ее программы содержит символ, не присутствующий в К

5. **Значение суперпозиции Z(I_1 (4, 2)) исходных п/р функций и констант 4, 2 равно**

   A) 0 B) 5 C) 3 D) 1 E) 4

ГЛОССАРИЙ

№ п/п	новые понятия	содержание
1.	**Разбиение множества** U	– система непустых подмножеств $\{A_\alpha\}$ множества U такая, что их объединение равно U (полнота разбиения), а все попарные пересечения – пусты (чистота разбиения). Сами A_α называются классами, или блоками разбиения
2.	**Декартовым (прямым) произведением множес тв** $A\,u\,B$	называется множество M всех пар ($a \cdot b$), таких, что $a \in A\ u\ b \in B$
3.	**Суперпозиция функций**	– функция, полученная из системы функций f, f_1, f_2, … , f_k, некоторой подстановкой функций f_1, f_2, … , f_k во внешнюю функцию f вместо переменных и переименованиями переменных
4.	**Характеристическая функция множества** $M \subseteq U$	– отображение $\chi_M : U \to B$, ставящее элементам множества M единицу, а элементам дополнения ноль
5.	**Алфавит**	– это кортеж попарно различных символов, называемых буквами алфавита
6.	**Слово в алфавите** A	– кортеж из символов алфавита A
7.	**Лексико-графическое (алфавитное) упорядочение**	– для различных слов α,β в алфавите A с упорядоченными символами $a_1 \prec a_2 \prec \ldots \prec a_n$ устанавливается упорядочение: $\alpha \prec \beta$, если возможно представление $\alpha = \gamma\ a_i\ \delta'$, $\beta = \gamma\ a_j\ \delta''$, при котором либо $a_i \prec a_j$ (подслово γ может быть пустым), либо $a_i = a_j$ и δ' – пустое подслово
8.	**Логические переменные**	– это переменные, принимающие значения из двухэлементного множества B{0;1}
9.	**Булева функция (логическая функция, функция алгебры логики)**	– это функция одной или нескольких переменных $Z= f(X_1,X_2,\ldots, X_n)$, где $f_S\ (X_1,X_2,\ldots, X_n)$, Z – логические переменные, т.е. и значения аргументов, и значение функции – ноль или единица
10.	**Совершенная дизъюнк тивная нормальная форма (СДНФ)**	– представление функции $Z = f(X_1, X_2, \ldots, X_n)$ в виде дизъюнкции всех элементарных конъюнкций, соответствующих наборам значений (σ_1, σ_2,…, σ_n), на которых Z =1: $f(X_1, X_2, \ldots, X_n) = \bigvee\limits_{f(\sigma_1,\sigma_2,\ldots,\sigma_n)=1} X_1^{\sigma_1} \cdot X_2^{\sigma_2} \cdot \ldots \cdot X_n^{\sigma_n}$
11.	**Предикат**	– функция Р типа: $M^n \to B$, где B={0,1}, М – **произвольное** множество, т.е. функция Р, сопоставляющая вектору ($m_1,m_2,\ldots,m_n$) значение 0 или 1. Множество М называется **предметной областью** предиката Р($m_1,m_2,\ldots,m_n$); $m_1,m_2,\ldots,m_n$ – **предметными переменными**, Р – предикатным символом.
12.	**Теорема Черча**	Не существует алгоритма, который для любой формулы логики предикатов устанавливал бы, общезначима она или нет
13.	**Комбинаторная**	- это расположение конечного множества элементов,

	конфигурация	удовлетворяющее ряду специальных свойств
14.	**выборкой объема k из n элементов**, или **(n, k)-выборкой**	называется набор (множество или кортеж) элементов ($x_{i_1}, x_{i_2}, \ldots, x_{i_k}$), составленный из элементов множества X = ($x_1, x_2, \ldots x_n$),
15.	**(n, k)-размещением с повторениями**, или **размещением с повторениями из n элементов по k**	называется упорядоченная (n, k)-выборка, в которой элементы могут повторяться
16.	**(n, k)-размещением без повторений**, или **размещением без повторений из n элементов по k**.	называется упорядоченная (n, k)-выборка, в которой элементы (n, k)-выборки попарно различны
17.	**n-перестановками**, или **перестановками из n элементов**.	называются **(n, n)-размещения без повторений**
18.	**сочетаниями**	называются неупорядоченные (n, k)-выборки
19.	**Число (n, k)-размещений без повторений**	$A_n^k = \dfrac{n!}{(n-k)!}$
20.	**Число различных (n,k)-размещений с повторениями**	$\hat{A}_n^k = n^k$
21.	**Числа (n, k)-сочетаний без повторений**	$C_n^k = \dfrac{n!}{k!\cdot(n-k)!}$
22.	**Число (n, k)-сочетаний с повторениями вычисляются по формуле**	$\hat{C}_n^k = C_{n+k-1}^k = C_{n+k-1}^{n-1}$
23.	**полиномиальным коэффициентом**	называется $\dfrac{n!}{b_1!\cdot b_2!\cdot\ldots\cdot b_r!}$, где $b_1 + b_2 + \ldots + b_r = n$.
24.	**Неориентированный граф** (соответственно **ориентированный граф**, или **орграф**) G	система G = (V,E,Г), состоящая из множества элементов V={v}, называемых вершинами графа, множества элементов E={e}, называемых ребрами, и отображения Г: E $\to V^2$, ставящего в соответствие каждому элементу $e\in E$ неупорядоченную (соответственно упорядоченную) пару элементов v_1, $v_2\in V$, называемых концами ребра e
25.	**Матрица инциденций графа с b вершинами и p ребрами**	прямоугольная матрица A = $\| a_{ij} \|$ с b строками и p столбцами, строки которой соответствуют вершинам графа, а столбцы - ребрам, причем для неориентированного графа элемент матрицы a_{ij} равен 1, если вершина v_i и ребро e_j инцидентны, и равен 0 в противном случае.
26.	**Матрица соседства (смежности) вершин графа с b вершинами**	квадратная матрица B = $\| b_{ij} \|$ размерности b, строки и столбцы которой соответствуют вершинам графа, причем неотрицательный элемент b_{ij} равен числу ребер,

		идущих из вершины v_i в вершину v_j
27.	**Полным** графом	называется граф без кратных ребер (и иногда без петель), в котором любые две вершины соединены ребром
28.	**Двудольным** графом	называется граф, вершины которого разбиты на два непересекающихся класса: $V = V_1$ И V_2, а ребра связывают вершины только из разных классов - не обязательно все пары
29.	**Путем [** v_{i_0} **,** v_{i_n} **] из вершины** v_{i_0} **в вершину** v_{i_n} ,	называется последовательность вершин и ребер графа G $v_{i_0}e_1v_{i_2}e_2v_{i_2}...v_{i_{n-1}}e_nv_{i_n}$, $\Gamma(e_k) = (v_{i_{k-1}}, v_{i_k})$ для $k = 1, 2,..., n$. Вершина v_{i_0} называется **началом**, а v_{i_n} - **концом пути**; число n называется **длиной пути.**
30.	**Цепью**	называется последовательность вершин и ребер, образующая путь в соотнесенном неориентированном графе
31.	**контуром** (соответственно **циклом**)	называется путь (соответственно цепь), в которой начальная вершина совпадает с конечной.
32.	Путь, цепь, контур, цикл называются **простыми** (соответственно **элементарными**)	если каждое их ребро (соответственно каждая вершина и каждое ребро) входит в последовательность ровно один раз (не считая последней вершины в записи цикла)
33.	**компонентами связности** графа (или **связными компонентами**)	называются подграфы, натянутые на классы эквивалентности, на которые разбиваются множество вершин графа, причем любые две вершины из одной связной компоненты соединены хотя бы одной цепью, вершины же из разных компонент не связаны цепью.
34.	Граф называется **связным**	если он имеет ровно одну компоненту связности, т.е. если любые две его вершины связаны цепью
35.	Ребро е произвольного графа G называется **циклическим**	, если оно принадлежит хотя бы одному элементарному циклу в графе, и **ациклическим** в противном случае
36.	**деревом**	называется связный граф без циклов
37.	Граф называется **четным**	если степени всех его вершин - четные числа
38.	**Цикломатическим числом графа**	называется число $\nu = p - b + k$, где $p=\sum_{i=1}^{k} p_i$, $b=\sum_{i=1}^{k} b_i$, k – связность графа
39.	**Кодом**	называется система условных знаков (символов) для передачи, обработки и хранения (запоминания) различной информации
40.	**Кодированием**	называется отображение конечных или счетных множеств объектов произвольной природы в множества последовательностей из цифр 0,1,...,r–1, где r–некоторое целое положительное число (в частности, r = 2).
41.	код $V = \{v_i\}$ называется **разделимым**	если из каждого равенства в алфавите $B = \{0, 1\}$ вида $v_{i_1}v_{i_2}...v_{i_k} = v_{j_1}v_{j_2}...v_{j_l}$ следует, что $k = l$ и $i_t = j_t$, $t = 1, 2,...,k$.
42.	Код $V=\{v_i\}$ называется **префиксным**	если никакое слово v_k не является началом никакого слова v_j, $j \neq k$
43.	**Неравенство Крафта-Макмиллана для**	$\sum_{i=0}^{m-1} \frac{1}{2^{l_i}} \leq 1.$ необходимо и достаточно для того, чтобы

	разделимых кодов	существовал разделимый код $V = \{v_0, v_1,..., v_{m-1}\}$ с длинами $\lambda(v_i) = l_i$, $i = 0, 1,..., m-1$, где $l_0, l_1, ... , l_{m-1}$ – произвольный набор натуральных чисел ($m \geq 2$)
44.	**Стоимостью** кода $V = \{v_0, v_1,..., v_{m-1}\}$ при распределении $P = \{p_0, p_1, ..., p_{m-1}\}$ $(p_i \geq 0, \sum_{i=0}^{m-1} p_i = 1)$.	называется величина $L_V(P) = \sum_{i=0}^{m-1} p_i \cdot \lambda(v_i)$
45.	**Оптимальным** при распределении $P^* = \{p_0, p_1,..., p_{m-1}\}$	называется префиксный код $V = \{v_0, v_1,..., v_{m-1}\}$, если его стоимость $L_V(P^*) = L(P) = \inf L_V(P)$
46.	**Расстоянием Хемминга** между векторами X и Y в векторном пространстве B^n	называется величина **d(X,Y), равное** числу несовпадающих компонент векторов X и Y
47.	**Нормой** $\|X\|$ вектора X в векторном пространстве B^n	называется расстояние между X и нулевым n-мерным вектором (0,0,...,0), т.е. $\|X\| = \sum_{j=1}^{n} x_j$ и $d(X,Y) = \|X \oplus Y\|$
48.	**Кодовым расстоянием** для произвольного кода $V = \{v_0,v_1,...,v_{m-1}\} \subseteq B^n$	называется величина $d(V) = \min_{i \neq j} d(v_i, v_j)$.
49.	**Одиночной ошибкой вида 0 →1 (1 →0)** в слове X (замещением или **аддитивной ошибкой)**	называют результат замены одного из символов 0 (соответственно 1) символом 1 (соответственно 0).
50.	**Одиночной ошибкой вида 0 →L (1 →L)** в слове X **(выпадением символа)**	называют результат удаления одного из символов 0 (соответственно 1); при этом длина слова уменьшается на единицу. . Одиночные ошибки вида $+2^i$ и -2^i называются
51.	**Одиночной ошибкой вида L →0 (L →1)**	называют результат *вставки символов* перед некоторым символом слова или после его последнего символа; при этом длина слова увеличивается на единицу.
52.	**Метрической окрестностью** радиуса s слова X	называется множеством точек, удаленных от X на расстояние не более s
53.	Код V является **кодом с исправлением s замещений**	тогда и только тогда, когда $d(V) > 2s$, т.е. $d(V) \geq 2s + 1$, где $d(V)$ – кодовое расстояние.
54.	**Теорема**	Код V с кодовым расстоянием d всегда позволяет обнаружить (d – 1) или менее одиночных ошибок типа замещения.
55.	**В ходе Хэмминга**	компоненты, номера которых отличны от $2^0, 2^1,..., 2^{l-1}$ называются **информационными** позициями, значения компонент с номерами $2^0, 2^1,..., 2^{l-1}$ **проверочными** позициями
56.	В логической сети	называется **элемент,** функционирование которого

	задержкой	определяется так: если x(t) - входной аргумент элемента задержки, то D(t+1) = x(t), т.е. значение на выходе задержки в каждый момент времени равно значению на входе элемента в предыдущий момент времени
57.	**Логической сетью**	называется схема из функциональных элементов и задержек, действующая в тактовом (дискретном) режиме
58.	**Существенное свойство логической сети**:	если задать входную последовательность X(0), X(1),..., то однозначно определяется выходная последовательность Z(0), Z(1),...
59.	**Конечным автоматом**	называется система (A, B, Q,), где $A = \{a_1, a_2, \dots, a_p\}$ - входной алфавит, $B = \{b_1, b_2, \dots, b_r\}$ - выходной алфавит, $Q = \{q_1, q_2, \dots, q_w\}$ - множество внутренних состояний (q_1 называется начальным состоянием); функция выходов $\varphi : Q\times A\to B$ и функция переходов ψ: $Q\times A\to Q$ определены на множестве $Q\times A$.
60.	**Система канонических уравнений автомата**	это уравнения: $$b(t) = \varphi(q(t), a(t)),$$ $$q(t+1) = \psi(q(t), a(t)),$$ при этом считается, что $q(0) = q_1$.
61.	**Таблицей (матрицей) переходов автомата**	называется таблицу из w строк и p столбцов, определяющая значения декартова произведения $\varphi\times\psi : Q\times A \to Q\times B$
62.	**Графом** (**диаграммой**) **переходов** автомат	называется – ориентированный граф (Q, E), где Q - множество вершин, а дуги соответствуют клеткам таблицы переходов: если на пересечении строки q_i и столбца a_j в таблице содержится пара (q_k / b), то этой клетке соответствует дуга (q_i , q_k), которой приписана упорядоченная пара (a_j; b)
63.	**Алгоритмом**	называется точное предписание, которое задает вычислительный в широком смысле процесс, называемый в этом случае алгоритмическим
64.	**Вычислимой функцией**	называется функция, для вычисления которой существует алгоритм
65.	**Исходными**	называются функции: 1) Z(x) = 0 - нуль-функция (“zero”), т.е. константа 0; 2) N(x) = “x + 1” - функция следования (“next”), обозначение: x’; 3) $I_k(x1,\dots, xn) = x_k$ - селекторные функции, где k = 1, 2,..., n.
66.	Применением **оператора суперпозиции S** к функциям f, $g_1,\dots, g_n$	называется подстановка функций $g_1,\dots, g_n$ во внешнюю функцию f вместо переменных и переименованиями переменных $h(x_1,\dots, x_m) = f(g_1(x_1,\dots, x_m),\dots, g_n(x_1,\dots, x_m))$. (обозначение: $h = S(f, g_1,\dots, g_n)$)
67.	Применением **оператора примитивной рекурсии R** к функциям g от n переменных и h от (n+2) переменных	называется формирование функции f от (n+1) переменных по следующим правилам: 1) $f(x_1,\dots, x_n, 0) = g(x_1,\dots, x_n)$, 2) $f(x_1,\dots, x_n, k+1) = h(x_1,\dots, x_n , k, f(x_1,\dots, x_n, k))$, $k \in N^0$. (обозначение: f = R(g, h))

68.	**Примитивно рекурсивной** (сокращенно **п/р)** функцией	называется функция, получаемая из исходных применением конечного числа операторов суперпозиции и примитивной рекурсии
69.	Примитивно рекурсивным предикатом $P(x_1,...,x_n)$	называется предикат, если его характеристическая функция $\chi_p(x_1,\ldots,x_n)=\begin{cases}1, \text{если } P(x_1,\ldots,x_n) \text{ истинно}\\ 0, \text{если } P(x_1,\ldots,x_n) \text{ ложно}\end{cases}$ которой является п/р функцией
70.	Результатом применения **ограниченного оператора минимизации (μ-оператора)** к предикату $P(x_1,..., x_n, y)$	называется функция $f(x_1,..., x_n, z)$, равная такому значению $y\in\{0, 1,..., z\}$, при котором $P(x_1,..., x_n, y) = 1$ и $\forall_{t<y}\, t$ $P(x_1,..., x_n, t) = 0$; если же $P(x_1,..., x_n, y) = 0$ для всех $y \in\{0, 1,..., z\}$, то полагают $f(x_1 ,..., x_n, z) = z + 1$, чтобы функция f была всюду определенной. (Обозначение: $f(x_1,..., x_n, z) = \mu_{y\le z} P(x1,..., xn, y)$.)
71.	**Функция называется частично рекурсивной (ч/р)**	если она может быть получена из исходных применением конечного числа операторов S (суперпозиции), R (примитивной рекурсии) и неограниченных μ -операторов.
72.	**Тезис Черча**	всякая эффективно вычислимая функция является частично рекурсивной.
73.	**Машина Тьюринга (МТ или м/т)**	задается тройкой T = (A, Q, P), где $A = \{a_1,..., a_n\}$ - внешний алфавит, часто выбирают A = {0, 1}; $Q = \{q_0, q_1,..., q_m\}$ - внутренний алфавит, элементы которого обозначают состояния управляющего устройства, обычно q_0 - заключительное состояние, q_1 - начальное состояние; P - **программа**, состоящая из команд вида $q_ia_j \to q_ka_lS$, где S - один из символов сдвига: L (налево), R (направо), E (на месте); $1\le q_i \le m$, $0 \le q_k \le m$, $a_j, a_l \in A$.
74.	**Конфигурацией** (машинным словом)	называется слово К вида Uq_iV, где U - слово на ленте левее головки, V - слово правее слова U, начиная с символа, находящегося в той же ячейке, где расположена головка
75.	**Протокол работы машины Тьюинга**	это детерминированная последовательность $K_{i_1}, K_{i_2}, \cdots, K_{i_n} \cdots$ конфигураций получаемых после каждого шага работы м/т
76.	Говорят, что машина T **правильно вычисляет** функцию $f(x_1,..., x_n)$ с областью определения $D \subseteq N^0\times N^0\times...\times N^0$ (сокращенно f(X))	если для $X\in D$ верно, что $T: q_1 01^{x_1} 01^{x_2} 0 \ldots 01^{x_n} 0 \mapsto q_0 01^{f(X)} 0$, а для $X\notin D$ верно, что $T: q_1 01^{x_1} 01^{x_2} 0 ... 01^{x_n} 0$, т.е. неприменима.
77.	Функция называется **правильно вычислимой**	если есть МТ, которая ее правильно вычисляет
78.	**Теорема Тьюринга**	Все частично рекурсивные функции правильно вычислимы.

Библиографический список:

1. Ветухновский Ф.Я. Дискретная математика. Рабочий учебник СГУ. Часть 1. Отношения. Булевы функции. Предикаты. М.:СГУ, 2001
2. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике: Учеб. пособие. — 3-е изд., перераб. — М.: ФИЗМАТЛИТ, 2005.
3. Ветухновский Ф.Я. Дискретная математика. Рабочий учебник СГУ. Часть 2. Графы и сети. Кодирование. Автоматы и алгоритмы. М. :СГУ, 2001
4. Тишин В. В. Дискретная математика в примерах и задачах. — СПб.: БХВ - Петербург, 2008. — 352 с.
5. Андерсон, Джеймс А. Дискретная математика и комбинаторика. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 960 с.
6. Аляев Ю.А., Тюрин С.Ф.. Дискретная математика и математическая логика: учебник. – М. Финансы и статистика, 2006. – 368 с.
7. Нефедов В.Н., Осипова В.А. Курс дискретной математики: Учеб. пособие для студентов специальности. “Прикладная математика”. М., МАИ, 1992. — 264 с.
8. Галушкина Ю.И., Марьямов А.Н. Конспект лекций по дискретной математике. – М.: Айрис-пресс, 2007. – 176 с.

ПРИЛОЖЕНИЯ

Приложение 1

Таблица всевозможных булевых функций двух переменных.

X	Y	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Функция	Формула	Название
g_0 =	0	константа 0
g_1 =	$X\&Y = X\cdot Y$	конъюнкция
g_2 =	$\neg(X\to Y) = X\&\neg Y$	отрицание имплкации
g_3 =	X	первая переменная
g_4 =	$\neg(Y\to X) = \neg X\&Y$	отрицание обратной импликации
g_5 =	Y	вторая переменная
g_6 =	$X\oplus Y = \neg X\&Y\vee X\&\neg Y$	сумма по модулю 2
g_7 =	$X\vee Y$	дизъюнкция
g_8 =	$\neg(X\vee Y) = \neg X\&\neg Y$	отрицание дизъюнкции
g_9 =	$X\sim Y = \neg X\&\neg Y\vee X\&Y$	эквивалентность
g_{10} =	$\neg Y$	отрицание второй переменной
g_{11} =	$Y\to X = X\vee\neg Y$	обратная импликация
g_{12} =	$\neg X$	отрицание первой переменной
g_{13} =	$X\to Y = \neg X\vee Y$	импликация
g_{14} =	$X\vert Y = \neg X\vee\neg Y = \neg(X\&Y)$	штрих Шеффера
g_{15} =	1	константа 1

Приложение 2

Эквивалентные формулы логики высказываний

№ п/п	Формула	комментарии
1.	$X \vee Y = Y \vee X$	*Коммутативность дизъюнкции*
2.	$X \wedge Y = Y \wedge X$	*Коммутативность конъюнкции*
3.	$(X \vee Y) \vee Z = X \vee (Y \vee Z)$	*ассоциативность дизъюнкции*
4.	$(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$	*ассоциативность конъюнкции*
5.	$(X \vee Y) \wedge Z = (X \wedge Z) \vee (Y \wedge Z)$	*взаимная дистрибутивность дизъюнкции и конъюнкции*
6.	$(X \wedge Y) \vee Z = (X \vee Z) \wedge (Y \vee Z)$	
7.	$X \vee X = X$	*устранение кратности*
8.	$X \wedge X = X$	
9.	$X \vee (X \wedge Y) = X$	*правила поглощения*
10.	$X \wedge (X \vee Y) = X$	
11.	$\neg(X \vee Y) = \neg X \wedge \neg Y$	*законы де Моргана*
12.	$\neg(X \wedge Y) = \neg X \vee \neg Y$	
13.	$\neg X \vee X = 1$	*закон исключенного третьего*
14.	$\neg X \wedge X = 0$	*закон противоречия*
15.	$0 \vee X = X$	*основные операции над переменной и константами 0 и 1*
16.	$0 \wedge X = 0$	
17.	$1 \vee X = 1$	
18.	$\neg 0 = 1$ $(\neg 1 = 0)$	
19.	$1 \wedge X = X$	
20.	$\neg(\neg X) = X$	*снятие двойного отрицания*
21.	$X \wedge Y \vee X \wedge \neg Y = X$	*правила склеивания*
22.	$X \vee \neg X \wedge Y = X \vee Y$	