РОСЖЕЛДОР

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный университет путей сообщения»**

(ФГБОУ ВО РГУПС)

Е.В. Голубенко

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ. ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

Учебное пособие

Утверждено учебно-методическим советом университета

Ростов-на-Дону
2019

УДК 519.6(07) + 06

Рецензенты: кандидат технических наук, доцент
В.В. Храмов (ИУБиП);
кандидат физико-математических наук, доцент
В.А. Богачёв (РГУПС)

Голубенко, Е.В.

Теоретические основы информационных и компьютерных технологий. Основы математической логики: учеб. пособие / Е.В. Голубенко; ФГБОУ ВО РГУПС. – Ростов н/Д, 2019. – 70 с.: ил. – Библиогр.: с. 69.

ISBN 978-5-88814-889-1

Приведены справочные данные, упражнения и задачи по дисциплине «Теоретические основы информационных и компьютерных технологий» для направлений подготовки бакалавров 09.03.01 «Информатика и вычислительная техника» и 09.03.02 «Информационные системы и технологии».

Предназначено для студентов и магистрантов направлений «Информатика и вычислительная техника», «Информационные системы и технологии» и «Механотроника и робототехника», а также для студентов, аспирантов и магистрантов всех специальностей, изучающих дисциплину «Математическая логика» и смежные дисциплины.

Одобрено к изданию кафедрой «Вычислительная техника и автоматизированные системы управления».

ISBN 978-5-88814-889-1

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Дисциплина «Теоретические основы информационных и компьютерных технологий» включает в себя три крупных раздела:

1) математическая логика;

2) дискретная математика;

3) алгебраические системы и теория кодирования.

Предлагаемое учебное пособие посвящено разделу «Математическая логика». В пособии представлены следующие темы: множество, булевы функции, нормальные формы и их минимизация, двойственные функции, теорема Поста, исчисление высказываний, логика предикатов, вычислимые функции, машина Тьюринга, модальные и темпоральные логики, функциональные схемы.

По каждой теме приводится необходимый минимум теоретических сведений, рассматривается ряд примеров и предлагается ряд задач для самостоятельного решения.

Доказательства большинства теорем и других полученных результатов не приводятся, пытливый читатель может найти их в литературе, представленной в библиографическом списке.

1 МНОЖЕСТВА

1.1 Элементы теории множеств

Понятие *«множество»* было введено в математику немецким математиком Кантором. Теория множеств получила признание в конце XIX века и оказала огромное влияние на последующее развитие математики.

В настоящее время множество является одним из основных понятий математики. Оно относится к фундаментальным понятиям, не подлежащим определению. Объекты, составляющие множество, называются его элементами. Говорят, что они принадлежат множеству. Основные положения классической теории множеств приведены в табл. 1 и 2.

В табл. 1 даны определения для отношений ($\subset$, $=$) и операций с множествами ($\overline{}$, $\cap$, $\cup$, $\setminus$, Δ).

Здесь приняты следующие обозначения:

x – произвольный элемент множества;

A, B – некоторые множества;

$x \in A$ – элемент x принадлежит множеству A;

$x \notin A$ – элемент x не принадлежит множеству A;

$\rightarrow$ – логическое следование (импликация);

$\leftrightarrow$ – двустороннее следование (эквивалентность);

$\wedge$ – логическое «И» (конъюнкция);

$\vee$ – логическое «ИЛИ» (дизъюнкция).

В табл. 2 приводятся основные соотношения для операций булевой алгебры множеств ($\overline{}$, $\cap$, $\cup$).

Здесь: U – универсальное множество, содержащее все элементы; $\varnothing$ – пустое множество, не содержащее ни одного элемента.

Таблица 1

Отношения и операции на множествах

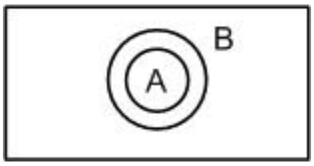
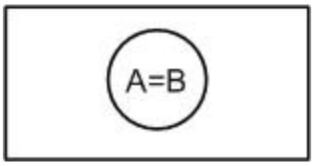
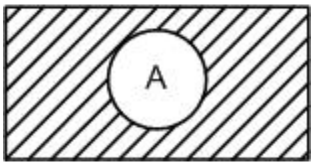
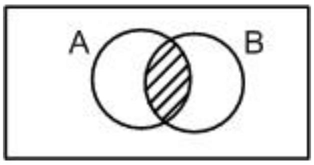
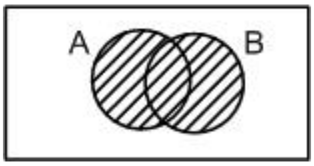
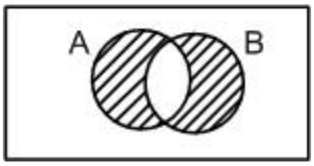
Наименование	Символ	Соотношение	Диаграмма
Подмножество	$\subset$	$A \subset B \leftrightarrow (x \in A \to x \in B)$	
Равенство	$=$	$A = B \leftrightarrow A \subset B \wedge B \subset A$	
Дополнение	$\overline{A}$	$x \in \overline{A} \leftrightarrow x \notin A$	
Пересечение	$\cap$	$x \in A \cap B \leftrightarrow x \in A \wedge x \in B$	
Объединение	$\cup$	$x \in A \cup B \leftrightarrow x \in A \vee x \in B$	
Разность	$\setminus$	$x \in A \setminus B \leftrightarrow x \in A \wedge x \notin B$; $A \setminus B = A \cap \overline{B}$	
Симметрическая разность	Δ	$x \in A \Delta B \leftrightarrow$ $(x \in A \wedge x \notin B) \vee$ $\vee (x \in B \wedge x \notin A)$; $A \Delta B = (A \setminus B) \cup (B \setminus A)$	

Таблица 2

Свойства теоретико-множественных операций

Свойство (закон)	Соотношения
Идемпотентность	$A \cup A = A, A \cap A = A$
Коммутативность	$A \cup B = B \cup A, A \cap B = B \cap A$
Ассоциативность	$A \cup (B \cup C) = (A \cup B) \cup C,$ $A \cap (B \cap C) = (A \cap B) \cap C$
Дистрибутивность	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Поглощение	$(A \cap B) \cup A = A, (A \cup B) \cap A = A$
Свойства $\varnothing$	$A \cup \varnothing = A, A \cap \varnothing = \varnothing$
Свойства U	$A \cup U = U, A \cap U = A$
Инволютивность	$\overline{\overline{A}} = A$
Законы де Моргана	$\overline{A \cap B} = \overline{A} \cup \overline{B}, \overline{A \cup B} = \overline{A} \cap \overline{B}$
Свойства дополнения	$A \cup \overline{A} = U, A \cap \overline{A} = \varnothing$

Теоретико-множественные тождества, в частности, приведенные в табл. 1 и 2, доказываются по следующему алгоритму.

Алгоритм 1.

1 Строятся диаграммы Венна (Эйлера) для левой и правой частей тождества (для основных операций такие диаграммы есть в табл. 1). Если диаграммы совпадают, переходим к шагу 2; в противном случае тождества нет – выход из алгоритма.

2 Берём произвольный элемент из левой части тождества (левого множества) и с помощью цепочки следований ($\to$), используя определения операций из табл. 1 и (или) соотношения из табл. 2, показываем, что этот элемент принадлежит и правой части тождества. Доказано, что левая часть тождества является подмножеством правой.

3 Далее для произвольного элемента из правой части доказываем вложение в левую часть с помощью цепочки обратных следований ($\leftarrow$), как и на шаге 2. Доказано, что правая часть является подмножеством левой.

Таким образом, доказано совпадение (равенство) левой и правой частей тождества как множеств.

4 Выход из алгоритма.

Если последовательности цепочек следований в обе стороны идентичны, шаги 2 и 3 объединяются в одну цепочку с двухсторонним следованием ($\leftrightarrow$). При использовании свойств операций из табл. 2, рассматриваемых как аксиомы, вместо символа $\leftrightarrow$ используется равенство (=).

Пример 1.1

Доказать: $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$.

Шаг 1. Убеждаемся, что диаграммы для левой и правой части тождества совпадают (рис. 1).

Шаги 2 и 3. Цепочки следований для доказательства вложений в левую и правую сторону идентичны, поэтому шаги 2 и 3 объединяем в один:

$x \in A \setminus (B \cup C) \leftrightarrow x \in A \cap \overline{B \cup C} \leftrightarrow x \in A \cap (\overline{B} \cap \overline{C}) \leftrightarrow x \in (A \cap \overline{B}) \cap$

$\cap (A \cap \overline{C}) \leftrightarrow x \in (A \setminus B) \cap (A \setminus C)$.

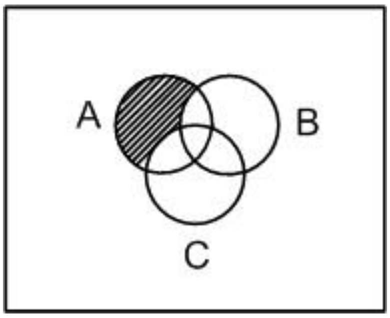


Рис. 1. Трилистник

Данное тождество можно доказать также, используя законы де Моргана, идемпотентности и ассоциативности:

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

1.2 Нечёткие множества

Нечёткие множества описываются функцией принадлежности элемента x множеству A:

$$0 \le \mu_A(x) \le 1.$$

Для обычных множеств:

$$\mu_A(x) = \begin{cases} 1, x \in A; \\ 0, x \notin A. \end{cases}$$

Основные соотношения для нечётких подмножеств представлены в таблице 3.

Свойства операций $\bar{\ }, \cap, \cup$ приведены в табл. 4, а операций $\bar{\ }, \otimes, \oplus$ – в табл. 5. В данных таблицах:

$\varnothing$ – пустое множество ($\mu_{\varnothing}(x) \equiv 0$);

U – универсальное множество ($\mu_U(x) \equiv 1$).

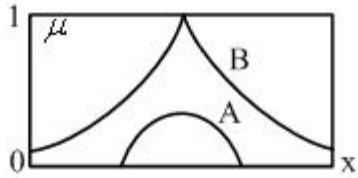
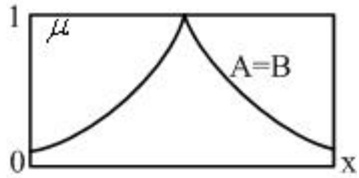
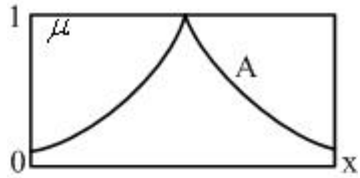
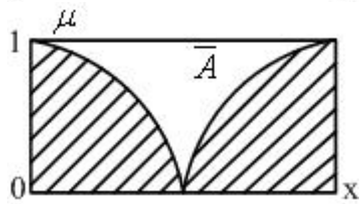
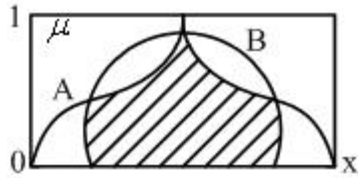
Из табл. 4 и 5 следует, что нечёткие алгебры с сигнатурами $\{\,^{-},\ \cap,\ \cup\}$ и $\{\,^{-}, \otimes, \oplus\}$ не являются булевыми.

Связь этих сигнатур устанавливают законы:

$$A \otimes (B \cap C) = (A \otimes B) \cap (A \otimes C),$$
$$A \otimes (B \cup C) = (A \otimes B) \cup (A \otimes C),$$
$$A \oplus (B \cap C) = (A \oplus B) \cap (A \oplus C),$$
$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C).$$

Таблица 3

Нечёткие отношения и операции

Наименование	Символ	Соотношение	Диаграмма
Подмножество	$\subset$	$A \subset B \leftrightarrow \mu_A(x) \le \mu_B(x)$	
Равенство	$=$	$A = B \leftrightarrow \mu_A(x) = \mu_B(x)$	
Дополнение	$\overline{A}$	$\mu_{\overline{A}}(x) = 1 - \mu_A(x)$	 
Пересечение	$\cap$	$\mu_{A \cap B}(x) = \min(\mu_A(x),$ $\mu_B(x))$	

Окончание табл. 3

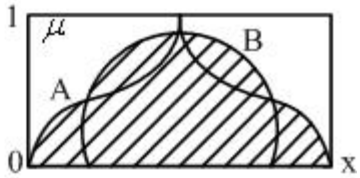
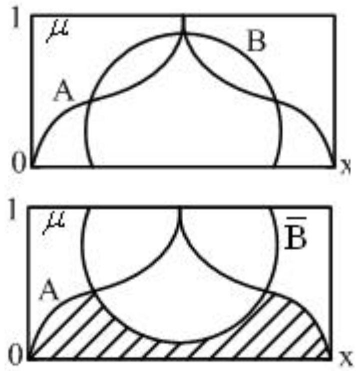
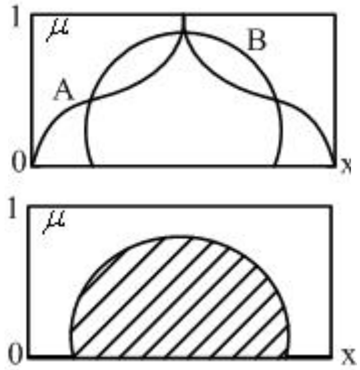
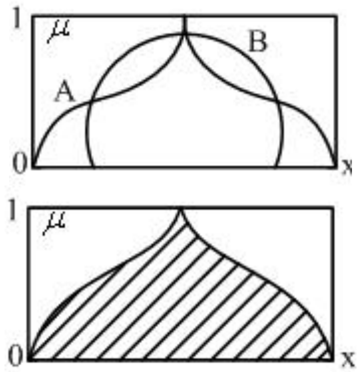
Наиме-нование	Символ	Соотношение	Диаграмма
Объеди-нение	$\cup$	$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$	
Разность	\	$\mu_{A\setminus B}(x) = \min(\mu_A(x), 1\text{-}\mu_B(x))$	
Произ-ведение	$\otimes$	$\mu_{A\otimes B}(x) = \mu_A(x)\mu_B(x)$	
Сумма	$\oplus$	$\mu_{A\oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$	

Таблица 4

Свойства нечётких операций $\overline{\ }$, $\cap$, $\cup$

Свойство	Соотношения
Коммутативность	$A \cup B = B \cup A,\ A \cap B = B \cap A$
Ассоциативность	$A \cup (B \cup C) = (A \cup B) \cup C,\ A \cap (B \cap C) = (A \cap B) \cap C$
Идемпотентность	$A \cup A = A,\ A \cap A = A$
Дистрибутивность	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Свойства $\varnothing$	$A \cup \varnothing = A,\ A \cap \varnothing = \varnothing$
Свойства U	$A \cup U = U,\ A \cap U = A$
Инволютивность	$\overline{\overline{A}} = A$
Законы де Моргана	$\overline{A \cap B} = \overline{A} \cup \overline{B},\ \overline{A \cup B} = \overline{A} \cap \overline{B}$
Закон поглощения	$A \cup (A \cap B) = A,\ A \cap (A \cup B) = A$

Таблица 5

Свойства нечётких операций $\overline{\ }$, $\otimes$, $\oplus$

Свойство	Соотношения
Коммутативность	$A \otimes B = B \otimes A,\ A \oplus B = B \oplus A$
Ассоциативность	$A \otimes (B \otimes C) = (A \otimes B) \otimes C,\ A \oplus (B \oplus C) = (A \oplus B) \oplus C$
Свойства $\varnothing$	$A \otimes \varnothing = \varnothing,\ A \oplus \varnothing = A$
Свойства U	$A \otimes U = A,\ A \oplus U = U$
Инволютивность	$\overline{\overline{A}} = A$
Законы де Моргана	$\overline{A \otimes B} = \overline{A} \oplus \overline{B},\ \overline{A \oplus B} = \overline{A} \otimes \overline{B}$

Доказательство нечётких тождеств, в том числе, приведённых в табл. 4 и 5, сводится к следующему алгоритму.

Алгоритм 2.

Нечёткое тождество справедливо, если функции принадлежности для левой и правой части тождества совпадают как вещественнозначные функции.

Пример 1.2.

Доказать закон поглощения: $A \cup (A \cap B) = A$.

Функция принадлежности для левой части имеет вид
$\mu_{A \cup (A \cap B)} = \max(\mu_A(x), \min(\mu_A(x), \mu_B(x)))$.

При $\mu_A(x) \geq \mu_B(x)$ имеем $\max(\mu_A(x), \mu_B(x)) = \mu_A(x)$.

При $\mu_A(x) < \mu_B(x)$ получаем $\max(\mu_A(x), \mu_A(x)) = \mu_A(x)$.

Таким образом, $\mu_{A \cup (A \cap B)} = \mu_A(x)$.

Пример 1.3.

Доказать закон де Моргана: $\overline{A \otimes B} = \overline{A} \oplus \overline{B}$.

Пусть $\mu_A(x) = a, \mu_B(x) = b.$ Тогда $\mu_{\overline{A \otimes B}} = 1 - ab,$ $\mu_{\overline{A} \oplus \overline{B}} = (1 - a) +$ $+ (1 - b) - (1 - a)(1 - b) = 1 - ab$, т. е. тождество выполняется.

Пример 1.4.

Доказать дистрибутивность: $A \otimes (B \oplus C) = (A \otimes B) \oplus (A \otimes C)$.

Пусть $\mu_A(x) = a, \mu_B(x) = b, \mu_C(x) = c.$

Тогда для левой части имеем:

$$\mu_{A \otimes (B \oplus C)} = a(b + c - bc) = ab + ac - abc,$$

а для правой –

$$\mu_{(A \otimes B) \oplus (A \otimes C)} = ab + \quad + ac - a^2 bc.$$

Следовательно, закон дистрибутивности не выполняется, так как $a^2 \neq a.$

2 БУЛЕВЫ ФУНКЦИИ

2.1 Высказывания

Под высказыванием понимают утверждение, относительно которого можно сказать, истинно оно или ложно.

Пример 2.1. Высказывание «Сочи – столица зимних Олимпийских игр 2014 года» истинно.

Элементарное высказывание – это одно утверждение. Обозначим элементарные высказывания строчными буквами латинского алфавита x, y, z, …, истинное высказывание – цифрой 1, ложное высказывание – цифрой 0.

Рассмотрим функции на множестве переменных x и y, при этом как переменные, так и функции могут принимать значения 0 и 1. Такие функции называются булевыми. Значения булевой функции можно задать при помощи таблицы истинности, показывающей, чему равна функция на всех возможных наборах значений её переменных.

2.2 Операции над высказываниями

Отрицание

Отрицанием высказывания A называется высказывание, истинное, если A ложно и ложное, если A истинно.

Обозначения: $\neg$A, $\overline{\text{A}}$, not A.

Читается «не A».

Таблица истинности:

A	$\neg$A
0	1
1	0

Как видим, отрицание меняет значение переменной на противоположное.

Конъюнкция (логическое умножение)

Конъюнкцией двух высказываний A и B называется высказывание, истинное в том и только том случае, когда оба высказывания A и B истинны.

Обозначения: A $\wedge$ B, A & B, A and B.

Читается «A и B».

Таблица истинности:

A	B	A ~ B
0	0	0
0	1	1
1	0	1
1	1	1

Дизъюнкция (логическое сложение)

Дизъюнкцией двух высказываний A и B называется высказывание, ложное в том и только том случае, когда оба высказывания A и B ложны.

Обозначения: $A \vee B$, A or B.

Читается «A или B».

Таблица истинности:

A	B	A ~ B
0	0	0
0	1	1
1	0	1
1	1	1

Импликация

Импликацией двух высказываний A и B называется высказывание, ложное в том и только том случае, когда высказывание A истинно, а высказывание B ложно.

Обозначения: $A \Rightarrow B$, $A \rightarrow B$.

Читается «если A, то B».

Таблица истинности:

A	B	A ~ B
0	0	1
0	1	1
1	0	0
1	1	1

Эквиваленция

Эквиваленцией двух высказываний A и B называется высказывание истинное в том и только том случае, когда оба высказывания A и B имеют одинаковое логическое значение.

Обозначения: $A \Leftrightarrow B$, $A \sim B$, $A \equiv B$.

Читается «A эквивалентно B» или «A тогда и только тогда, когда B».

Таблица истинности:

A	B	A ~ B
0	0	1
0	1	0
1	0	0
1	1	1

Рассмотренные пять операций: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция образуют алгебру высказываний или булеву алгебру.

С помощью названных операций можно строить произвольные булевы функции. Порядок выполнения операций в выражениях для булевых функций указывается скобками и соглашениями о приоритетах операций:

1) для отрицания скобки опускаются;

2) конъюнкция имеет приоритет над дизъюнкцией, импликацией и эквиваленцией;

3) дизъюнкция имеет приоритет над импликацией и эквиваленцией.

Булева функция полностью определяется своей таблицей истинности.

Пример 2.2. Построить таблицу истинности булевой функции $(x \to y) \wedge (\overline{x \vee y})$.

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

Искомая таблица истинности:

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

2.3 Равносильные функции

Две функции называются равносильными, если они принимают одинаковые значения на любом наборе значений входящих в эти функции переменных. У этих функций одинаковые таблицы истинности.

Из предыдущего примера видно, что функции $(\overline{x \vee y})$ и $(x \to y) \wedge (\overline{x \vee y})$ равносильны.

Основные равносильности (законы булевой алгебры):

1. $A \wedge B = B \wedge A$
2. $A \vee B = B \vee A$ (коммутативные законы)

3. $A \wedge (B \wedge C) = (A \wedge B) \wedge C$
4. $A \vee (B \vee C) = (A \vee B) \vee C$ (ассоциативные законы)
5. $A \wedge (B \vee C) = A \wedge B \vee A \wedge C$
6. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ (дистрибутивные законы)
7. $A \wedge A = A$
8. $A \vee A = A$ (законы идемпотентности)
9. $A \wedge И = A$
10. $A \vee Л = A$
11. $A \wedge Л = Л$
12. $A \vee И = И$ (свойства констант И и Л)
13. $A \wedge \neg A = Л$ (закон противоречия)
14. $A \vee \neg A = И$ (закон исключенного третьего)
15. $\neg(A \wedge B) = (\neg A) \vee (\neg B)$
16. $\neg(A \vee B) = (\neg A) \wedge (\neg B)$ (законы де Моргана)
17. $A \wedge (A \vee B) = A$
18. $A \vee A \wedge B = A$ (законы поглощения)
19. $A \wedge (\neg A \vee B) = A \wedge B$
20. $A \vee (\neg A) \wedge B = A \vee B$ (законы вычёркивания)
21. $\neg(\neg A) = A$ (закон отрицания отрицания или двойного отрицания)

Сформулированные законы алгебры логики могут быть обоснованы с помощью таблиц истинности. Законы 17–20 могут быть выведены из предыдущих. Следующие формулы описывают взаимосвязь логических операций.

22. $A \to B = \neg A \vee B$
23. $A \sim B = (A \to B) \wedge (B \to A)$

 23а. $A \sim B = (A \vee \neg B) \wedge (\neg A \vee B)$

 23б. $A \sim B = A \wedge B \vee (\neg A) \wedge B$

Равносильности 1–23б используются для различных преобразований:

1 Упрощение логического выражения.

2 Установление равносильности логических выражений.

3 Установление тождественной истинности или ложности логических выражений.

Если некоторое выражение тождественно истинно, то оно называется выводимым, в противном случае – не выводимым. Если выражение может принимать как истинное, так и ложное значение, оно называется выполнимым. Если выражение принимает только ложные значения, оно называется невыполнимым.

2.4 Логические предикаты и операции над ними

Логический предикат – это предложение, содержащее одну или несколько предметных переменных, которое превращается в высказывание, если вместо предметных переменных подставить их конкретное значение.

Пример 2.3. x есть простое число.

Обозначения: P(x), P(x, y). Здесь x, y – предметные переменные, P – предикатная переменная. Переменные, которыми обозначают высказывания, называются пропозициональными.

A $\wedge$ P(x) – здесь три переменных: пропозициональная (A), предикатная (P), предметная (x).

Если предикат содержит одну переменную, он называется одноместным, если две – двухместным и т.д.

Примеры математических предикатов:

$$2x^2 + 3x - 8 = 0$$
$$(x + y)^2 = x^2 + 2xy + y^2 = 0$$
$$x^2 < 0$$
$$f(x, y) = x^2 + y^2.$$

Среди предикатов можно рассматривать тождественно-истинные и тождественно-ложные.

На предикаты можно распространить логические операции, определённые для высказываний.

Под отрицанием предиката понимают такой предикат, который превращается в истинное высказывание для тех и только тех значений x, при которых данный предикат превращается в ложное высказывание.

Аналогичные определения строятся для остальных логических функций.

Высказывание можно рассматривать как частный случай предиката.

Существуют операции, характерные только для предикатов. Они превращают предикаты в высказывания, т.е. в предложения, не зависящие от предметных переменных. Они называются кванторами.

В математической логике используются два квантора:

1 Квантор всеобщности.

2 Квантор существования.

Квантор всеобщности есть такая операция, которая преобразует предикат P(x) в высказывание истинное тогда и только тогда, когда предикат P(x) принимает значение истинности для всех значений x.

Обозначение: $\forall x$ P(x) (читается – для всех x P(x)).

$\forall x$ P(x) ~ P(x) ≡ И.

Квантор существования есть такая операция, которая преобразует предикат P(x) в высказывание истинное тогда и только тогда, когда существует по крайней мере одно значение предметной переменной $x = x_0$, при котором предикат P(x) превращается в истинное высказывание.

Обозначение: $\exists x$ P(x) (читается – существует x , для которого P(x)).

$\exists x$ P(x) ~ $\overline{P(x)}$ ≡ Л.

Операции кванторов распространяются и на многоместные предикаты.

$\forall x\ \forall y$ P(x, y)
$\exists y\ \exists x$ P(x, y)
$\forall y\ \exists x$ P(x, y) ≠ $\forall x\ \exists y$ P(x, y).

Из пропозициональных переменных при помощи логических операций и скобок строятся составные высказывания.

Составные высказывания называются равносильными или логически эквивалентными, если они принимают одинаковые значения истинности при любом наборе значений составных частей.

Составные высказывания, принимающие истинные значения при любых значениях своих компонент, называются тавтологией.

Высказывание $\bar{Q} \rightarrow \bar{P}$ называется противоположным или контрапозитивным к высказыванию P → Q.

Пример 2.4.

Доказать, что высказывание $P_1 = A \vee (B \wedge C)$ равносильно высказыванию $P_2 = (A \vee B) \wedge (A \vee C)$.

Решение. Задача решается составлением таблицы истинности для P_1 и P_2.

A	B	C	A∨B	A∨C	B∧C	P_1	P_2
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

Столбцы P_1 и P_2 совпадают, следовательно, высказывания P_1 и P_2 равносильны.

Пример 2.5.
Является ли тавтологией высказывание

$$P = (A\wedge(B\vee C)) \to (A \vee (B\wedge C)) \ ?$$

Решение.
Составим таблицу истинности для P:

A	B	C	$B\vee C$	$A\wedge(B\vee C)$	$B\wedge C$	$A \vee (B\wedge C)$	P
0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	1
0	1	0	1	0	0	0	1
0	1	1	1	0	1	1	1
1	0	0	0	0	0	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

P принимает значение «истина» на всех наборах переменных, т.е. является тавтологией.

Пример 2.6.

Построить высказывание, противоположное к заданному: «Если сегодня понедельник, то у меня есть пятая пара».

Решение.

Если у меня нет пятой пары, то сегодня не понедельник.

Пример 2.7.

Запишите с помощью кванторов утверждение:

Существует единственное значение переменной x, для которого предикат $P(x)$ принимает истинное значение.

Решение.

$$\exists x(P(x) \wedge (\forall y\ ((P(y)) \to (y = x))))$$

2.5 Булевы функции от *n* переменных

Определение 1. Пусть $E_2 = \{0,1\}$ – множество, состоящее из двух элементов. Тогда положим $E_2^n = \{a_1, a_2, \ldots, a_n\} \mid \forall i\ (a_i \in E_2)$ – множество наборов длины n с элементами из E_2. Булевой функцией (функцией алгебры логики) назовём отображение $f(x_1, \ldots, x_n)$: $E_2^n \to E_2$. Множество всех булевых функций обозначим P_2. Булеву функцию можно задать таблично. Например, для $n = 1$:

x	f_0 0	f_1 1	f_2 x	f_3 $\bar{x}$
0	0	1	0	1
1	0	1	1	0

Первые две функции это константы 0 и 1, x – тождественная функция, $\bar{x}$ – отрицание x.

Теорема. Число булевых функций от *n* переменных равно 2^{2^n}.

Для *n* = 2 количество булевых функций равно 16.

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

f_0 – константа 0;

f_1 – стрелка Пирса $x_1 \downarrow x_2 = \bar{x}_1\bar{x}_2$

f_2 – $\bar{x}_1 x_2 = \overline{x2 \rightarrow x1}$

f_3 – отрицание $x_1 = \bar{x}_1$

f_4 – $x_1\bar{x}_2 = \overline{x1 \rightarrow x2}$

f_5 – отрицание $x_2 = \bar{x}_2$

f_6 – сложение по модулю 2 $x_1 \oplus x_2 = \overline{x1 \sim x2}$

f_7 – штрих Шеффера $x_1 | x_2 = \overline{x1 x2}$

f_8 – конъюнкция $x_1\, x_2$

f_9 – эквивалентность $x_1 \sim x_2$

f_{10} – тождественная функция по x_2 = x_2

f_{11} – импликация $x_1 \rightarrow x_2$

f_{12} – тождественная функция по x_1 = x_1

f_{13} – обратная импликация $x_2 \rightarrow x_1$

f_{14} – дизъюнкция $x_1 \vee x_2$

f_{15} – константа 1.

2.6 Формулы

При задании булевой функции с помощью таблицы истинности зависимость значений функции от значений переменных даётся в самом простом виде. Но очень часто на практике возникает ситуация, когда требуется установить связь между значениями различных булевых функций. Таблицы истинности здесь мало чем могут помочь. В этом случае используют формулы.

Пусть F – множество булевых функций. Формула над F определяется следующим образом:

1) любая функция из F есть формула над F,

2) если $A(x_1, x_2,\ldots, x_m)$ есть формула над F, и $A_1, \ldots, A_m$ есть формулы над F, то $A(A_1, A_2,\ldots, A_m)$ есть формула над F (то есть подставляя в формулы над F вместо переменных другие формулы над F, снова получим формулы над F).

Каждая формула над F реализует некоторую булеву функцию. Поэтому будем отождествлять формулу с реализуемой функцией.

3 НОРМАЛЬНЫЕ ФОРМЫ

Для любой булевой функции можно построить её таблицу истинности. Но и по таблице истинности можно восстановить булеву функцию. Покажем, как это делается.

3.1 Совершенная дизъюнктивная нормальная форма

Возьмём наборы переменных, на которых функция равна единице. Если значение переменной в этом наборе равно 0, то эта переменная берётся с отрицанием. Если значение переменной в этом наборе равно 1, то эта переменная берётся без отрицания.

Соединив все переменные, соответствующие этому набору, через знак &, (причём сам знак & для краткости будем опускать), мы получим *элементарную конъюнкцию.* Тогда дизъюнкция всех элементарных конъюнкций, соответствующих наборам значений переменных, где функция равна единице, и восстанавливает исходную функцию. Полученное представление называется *совершенной дизъюнктивной нормальной формой (СДНФ)* булевой функции.

СДНФ можно получить по таблице истинности с помощью следующего алгоритма.

Алгоритм 3

1 Положить номер строки в таблице истинности $i = 1$, номер элемента в строке $j = 1$.

2 Выбрать из таблицы набор переменных (n) с номером i. Положить $F_i =$ $= 1$. Если значение функции $f_i = 1$, то перейти к п.3, если $f_i \neq 1$, перейти к п. 5.

3 Выбрать элемент строки с номером j и сформировать элементарную конъюнкцию F_i. Если

$$x_i = \begin{cases} 0, \ F_i := F_i \wedge \bar{x}_j; \\ 1, \ F_i := F_i \wedge x_j. \end{cases}$$

4 Положить $j = j + 1$. Если $j < n$, перейти к п. 3, если $j \geq n - k$ перейти к п. 5.

5 Положить $i = i + 1$. Если $i < 2^n$, перейти к п. 2, в противном случае перейти к п. 6.

6 Записать $\bigvee_1 F_i$

7 Останов.

Пример 3.1. Построим СДНФ для функции, таблица истинности которой имеет вид:

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

Функция принимает значение 1 на наборах 010, 100, 101, 110.

Набору 010 соответствует элементарная конъюнкция $\bar{x}y\bar{z}$, набору 100 – $x\bar{y}\bar{z}$, набору 101 – $x\bar{y}z$, набору 110 – $xy\bar{z}$.

Получаем СДНФ $f = \bar{x}y\bar{z} \vee x\bar{y}\bar{z} \vee x\bar{y}z \vee xy\bar{z}$.

При построении СДНФ требуется, чтобы функция была отлична от тождественного нуля

3.2 Совершенная конъюнктивная нормальная форма

Возьмём наборы переменных, на которых функция равна нулю. Если значение переменной в этом наборе равно 1, то эта переменная берётся с отрицанием. Если значение переменной в этом наборе равно 0, то эта переменная берётся без отрицания.

Соединив все переменные, соответствующие этому набору, через знак $\vee$, мы получим *элементарную дизъюнкцию.* Тогда конъюнкция всех элементарных дизъюнкций, соответствующих наборам значений переменных, где функция равна нулю, и восстанавливает исходную функцию. Полученное представление называется *совершенной конъюнктивной нормальной формой (СКНФ)* булевой функции. Знак & для краткости будем опускать.

СКНФ вычисляется по аналогии с алгоритмом 3.

Алгоритм 4.

1 Положить номер строки в таблице истинности $i = 1$, номер аргумента в строке $j = 1$.

2 Выбрать из таблицы набор переменных с номером i. Положить $\Phi_i = 0$. Если значение функции $f_i = 0$, то перейти к п. 3, в противном случае перейти к п. 5.

3 Выбрать элемент строки с номером j и сформировать элементарную конъюнкцию Φ_i. Если

$$x_i = \begin{cases} 0, \ \Phi_i := \Phi_i \vee x_j; \\ 1, \ \Phi_i := \Phi_i \vee \bar{x}_j. \end{cases}$$

4 Положить $j = j + 1$. Если $j < n$, перейти к п. 3, в противном случае – к п. 5.

5 Положить $i = i + 1$. Если $i < 2^n$, перейти к п. 2, в противном случае перейти к п. 6.

6 Записать $\bigwedge_0 \Phi_i$.
7 Останов.

Пример 3.2. Построим СКНФ для функции из предыдущего примера.

Функция принимает значение 0 на наборах 000, 001, 011, 111.

Набору 000 соответствует элементарная дизъюнкция $x \vee y \vee z$, набору 001 – $x \vee y \vee \bar{z}$, набору 011 – $x \vee \bar{y} \vee \bar{z}$, набору 111 – $\bar{x} \vee \bar{y} \vee \bar{z}$.

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

При построении СКНФ требуется, чтобы функция была отлична от тождественного нуля.

3.3 Минимизация нормальных форм

Очень часто СДНФ и СКНФ, построенные по таблице истинности, оказываются весьма сложными. Поэтому возникает проблема построения минимальных нормальных форм для заданной функции.

3.3.1 Минимизация дизъюнктивных нормальных форм

Импликант. Булеву функцию g назовём импликантом булевой функции f, если для любых наборов из 0 и 1 из равенства $g = 1$ следует равенство $f = 1$.

Если отбрасывание любой переменной импликанта приводит к тому, что полученная функция перестаёт быть импликантом, то такой импликант называется простым.

Сокращённая ДНФ функции f (сокр. ДНФ f) есть дизъюнкция всех простых импликантов функции f.

Всякая функция реализуется своей сокращённой ДНФ. Для всякой функции, не равной тождественно нулю, существует единственная сокращённая ДНФ.

Алгоритм 5. Построение сокращённой ДНФ с помощью СКНФ:

1 По таблице истинности строим СКНФ функции.

2 В СКНФ раскрываем скобки, удаляем дублирующие элементы ($A \& A = A$, $A \vee A = A$) и элементы, которые содержат переменную вместе с её отрицанием ($A \& \bar{A} = 0$).

3 Проводим поглощение ($A \vee AB = A$) и удаляем дублирующие элементы. Сокращённая ДНФ получена.

Пример 3.3. Для функции f из разделов 3.1 и 3.2 построим сокращённую ДНФ.

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

Раскроем скобки. Перемножение лучше начинать со скобок, которые отличаются всего одной переменной (например, x и $\bar{x}$). Перемножим 1-ю и 2-ю скобки, а также 3-ю 4-ю скобки.

$(xx \vee xy \vee x\bar{z} \vee xy \vee yy \vee y\bar{z} \vee xz \vee yz \vee z\bar{z})(x\bar{x} \vee x\bar{y} \vee x\bar{z} \vee \bar{x}\bar{y} \vee \bar{y}\bar{y} \vee \bar{y}\bar{z} \vee \vee \bar{x}\bar{z} \vee \bar{y}\bar{z} \vee \bar{z}\bar{z})$.

В 1-й скобке слагаемое x поглощает все слагаемые, содержащие x, а слагаемое y поглощает все слагаемые, содержащие y. Во 2-й скобке слагаемое $\bar{y}$ поглощает все слагаемые, содержащие $\bar{y}$, а слагаемое $\bar{z}$ поглощает все слагаемые, содержащие $\bar{z}$.

Получим $(x \vee y)(\bar{y} \vee \bar{z}) = (x\bar{y} \vee x\bar{z} \vee y\bar{y} \vee y\bar{z}) = (x\bar{y} \vee x\bar{z} \vee 0 \vee y\bar{z}) =$

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

Метод неопределённых коэффициентов

Согласно теореме Жегалкина, любая булева функция может быть представлена полиномом вида:

$$f(x_1, x_2, \ldots, x_n) = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus \ldots \oplus$$
$$\oplus a_{n-1,n}x_{n-1}x_n \oplus \ldots \oplus \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} a_{i_1 i_2 \ldots i_k} x_{i_1} x_{i_2} \ldots x_{i_k} \oplus \ldots \oplus a_{12\ldots n}x_1x_2\ldots x_n,$$

где $a_0, a_1, \ldots a_n, \ldots, a_{i1i2\ldots ik}, \ldots, a_{123\ldots n} \in \{0,1\}$

Заменяя в полиноме Жегалкина $\oplus$ на $\vee$, получим ДНФ (в силу структурной инвариантности операций $\oplus$ и $\vee$ относительно ДНФ).

Например, для функций трёх переменных ДНФ может быть представлена в следующем виде:

$$f(x_1, x_2, x_3) = k_1^1x_1 \vee k_1^0\overline{x_1} \vee k_{21}^1x_2 \vee k_2^0\overline{x_2} \vee k_3^1x_3 \vee k_3^0\overline{x_3} \vee$$
$$\vee k_{12}^{11}x_1x_2 \vee k_{12}^{10}x_1\overline{x_2} \vee k_{12}^{01}\bar{x}_1x_2 \vee k_{12}^{00}\overline{x_1x_2} \vee$$
$$\vee k_{13}^{11}x_1x_3 \vee k_{13}^{10}x_1\overline{x_3} \vee k_{13}^{01}\bar{x}_1x_3 \vee k_{13}^{00}\overline{x_1x_3} \vee$$
$$\vee k_{23}^{11}x_2x_3 \vee k_{23}^{10}x_2\overline{x_3} \vee k_{23}^{01}\bar{x}_2x_3 \vee k_{23}^{00}\overline{x_2x_3} \vee$$
$$\vee k_{123}^{111}x_1x_2x_3 \vee k_{123}^{110}x_1x_2\overline{x_3} \vee k_{123}^{101}x_1\bar{x}_2x_3 \vee k_{123}^{100}x_1\overline{x_2x_3} \vee$$
$$\vee k_{123}^{011}\bar{x}_1x_2x_3 \vee k_{123}^{010}\bar{x}_1x_2\overline{x_3} \vee k_{123}^{001}\bar{x}_1\bar{x}_2x_3 \vee k_{123}^{000}\overline{x_1x_2x_3},$$

где k_{ijk}^{lmn} – неопределённые коэффициенты из множества {0, 1}.

Идея алгоритма заключается в подборе неопределённых коэффициентов, обеспечивающих минимальность ДНФ, на основании следующей системы уравнений (для $f(x_1, x_2, x_3)$):

$$k_1^0 \vee k_2^0 \vee k_3^0 \vee k_{12}^{00} \vee k_{13}^{00} \vee k_{23}^{00} \vee k_{123}^{000} = f_0(0,0,0);$$

$$k_1^0 \vee k_2^0 \vee k_3^1 \vee k_{12}^{00} \vee k_{13}^{01} \vee k_{23}^{01} \vee k_{123}^{001} = f_1(0,0,1);$$

$$k_1^0 \vee k_2^1 \vee k_3^0 \vee k_{12}^{01} \vee k_{13}^{00} \vee k_{23}^{10} \vee k_{123}^{010} = f_2(0,1,0);$$

$$k_1^0 \vee k_2^1 \vee k_3^1 \vee k_{12}^{01} \vee k_{13}^{01} \vee k_{23}^{11} \vee k_{123}^{011} = f_3(0,1,1);$$

$$k_1^1 \vee k_2^0 \vee k_3^0 \vee k_{12}^{10} \vee k_{13}^{10} \vee k_{23}^{00} \vee k_{123}^{100} = f_4(1,0,0);$$

$$k_1^1 \vee k_2^0 \vee k_3^1 \vee k_{12}^{10} \vee k_{13}^{11} \vee k_{23}^{01} \vee k_{123}^{101} = f_5(1,0,1);$$

$$k_1^1 \vee k_2^1 \vee k_3^0 \vee k_{12}^{11} \vee k_{13}^{10} \vee k_{23}^{10} \vee k_{123}^{110} = f_6(1,1,0);$$

$$k_1^1 \vee k_2^1 \vee k_3^1 \vee k_{12}^{11} \vee k_{13}^{11} \vee k_{23}^{11} \vee k_{123}^{111} = f_7(1,1,1).$$

Рассмотрим алгоритм минимизации на полученной таблице (матрице системы уравнений).

Алгоритм 6.

1 Выбрать очередную строку, в которой $f_i = 0$. Все коэффициенты в этой строке приравнять к нулю.

2 Если все нулевые строки просмотрены, перейти к п. 3, в противном случае – к п. 1.

3 Просмотреть строки, в которых $f_i = 1$, и вычеркнуть из них все коэффициенты, встречающиеся в строках с $f_i = 0$.

4 Переписать в память все модифицированные уравнения (строки).

5 Выбрать очередную строку с $f_i = 1$ и вычеркнуть (обнулить) максимально возможное количество коэффициентов так, чтобы ранг остающихся членов был минимальным.

В этом алгоритме использовались следующие свойства логических операций: если $x_1 \vee x_2 \vee \ldots \vee x_n = 0$, то $x_1 = x_2 = \ldots = x_n = 0$; если $x_1 \vee x_2 \vee \ldots \vee x_n = 1$, то хотя бы один член уравнения равен 1.

Пример 3.4. Рассмотрим функцию:

Получим систему для определения коэффициентов:

$$k_1^0 \vee k_2^0 \vee k_3^0 \vee k_{12}^{00} \vee k_{13}^{00} \vee k_{23}^{00} \vee k_{123}^{000} = 1;$$

$$k_1^0 \vee k_2^0 \vee k_3^1 \vee k_{12}^{00} \vee k_{13}^{01} \vee k_{23}^{01} \vee k_{123}^{001} = 0;$$

$$k_1^0 \vee k_2^1 \vee k_3^0 \vee k_{12}^{01} \vee k_{13}^{00} \vee k_{23}^{10} \vee k_{123}^{010} = 0;$$

$$k_1^0 \vee k_2^1 \vee k_3^1 \vee k_{12}^{01} \vee k_{13}^{01} \vee k_{23}^{11} \vee k_{123}^{011} = 0;$$

$$k_1^1 \vee k_2^0 \vee k_3^0 \vee k_{12}^{10} \vee k_{13}^{10} \vee k_{23}^{00} \vee k_{123}^{100} = 1;$$

$$k_1^1 \vee k_2^0 \vee k_3^1 \vee k_{12}^{10} \vee k_{13}^{11} \vee k_{23}^{01} \vee k_{123}^{101} = 1;$$

$$k_1^1 \vee k_2^1 \vee k_3^0 \vee k_{12}^{11} \vee k_{13}^{10} \vee k_{23}^{10} \vee k_{123}^{110} = 1;$$

$$k_1^1 \vee k_2^1 \vee k_3^1 \vee k_{12}^{11} \vee k_{13}^{11} \vee k_{23}^{11} \vee k_{123}^{111} = 1.$$

Из уравнений в силу свойств дизъюнкции вытекает, что

$$k_1^0 = k_2^0 = k_2^1 = k_3^0 = k_3^1 = k_{12}^{00} = k_{13}^{00} = k_{13}^{01} = k_{23}^{01} = k_{13}^{01} \vee k_{23}^{10} =$$
$$= k_{23}^{11} = k_{123}^{001} = k_{123}^{010} = k_{123}^{011} = 0.$$

После этого данная система примет вид:

$$\begin{aligned} k_{23}^{00} \vee k_{123}^{000} &= 1; \\ k_1^1 \vee k_{12}^{10} \vee k_{13}^{10} \vee k_{23}^{00} \vee k_{123}^{100} &= 1; \\ k_1^1 \vee k_{12}^{10} \vee k_{13}^{11} \vee k_{123}^{101} &= 1; \\ k_1^1 \vee k_{12}^{11} \vee k_{13}^{10} \vee k_{123}^{110} &= 1; \\ k_1^1 \vee k_{12}^{11} \vee k_{13}^{11} \vee k_{123}^{111} &= 1. \end{aligned}$$

Приравняем к нулю в каждом уравнении все коэффициенты, кроме тех, которые отвечают дизъюнкциям, содержащим наименьшее число переменных:

$$k_{12}^{11} = k_{12}^{10} = k_{13}^{11} = k_{13}^{10} \vee k_{123}^{111} = k_{123}^{110} = k_{123}^{101} = k_{123}^{100} = k_{123}^{000} = 0.$$

Получим систему:

$$\begin{aligned} k_{23}^{00} &= 1; \\ k_1^1 \vee k_{23}^{00} &= 1; \\ k_1^1 &= 1; \\ k_1^1 &= 1; \\ k_1^1 &= 1. \end{aligned}$$

Отсюда минимальная ДНФ равна: $f(x_1, x_2, x_3) = x_1 \vee \bar{x}_2 \bar{x}_3.$

Метод минимизирующих карт

Метод минимизации булевых функций с использованием карт Карно применяют при числе переменных не более пяти.

Упрощение функции по картам Карно начинается с записи в клетки значений функции при соответствующих наборах переменных.

Если исходная функция задана в виде формулы, наборы, на которых функция равна 1, определяют методом перебора переменных, при этом обеспечивая развёртывание формулы в совершенную ДНФ. В заполненной карте Карно наглядно отображаются соседние конституенты: им соответствуют единицы, расположенные в соседних клетках. Все клетки, содержащие 1, объединяются в замкнутые области. Каждая область должна представлять собой прямоугольник с числом клеток 2, 4, 8. Области могут пересекаться, и одни и те же клетки могут входить в разные области. Соседними клетками являются не только клетки, расположенные рядом по горизонтали и вертикали, но и клетки, находящиеся на противоположных границах карты.

При охвате клеток замкнутыми областями следует стремиться к минимальному числу областей, каждая из которых содержала бы возможно большее число клеток. Каждый член МДНФ составляют лишь из тех переменных, которые для соответствующей области имеют одно значение: без инверсии или с инверсией. Если переменная для одной клетки области имеет значение без инверсии, а для другой клетки той же области – с инверсией, она в соответствующем члене МДНФ отсутствует.

Карты Карно для частных случаев функций от двух и трёх переменных приведены на рисунке 3.1.

а)

	q	$\overline{q}$
p	$p \wedge q$	$p \wedge \overline{q}$
$\overline{p}$	$\overline{p} \wedge q$	$\overline{p} \wedge \overline{q}$

б)

	q		$\overline{q}$	
p	$p \wedge q \wedge r$	$p \wedge q \wedge \overline{r}$	$p \wedge \overline{q} \wedge \overline{r}$	$p \wedge \overline{q} \wedge r$
$\overline{p}$	$\overline{p} \wedge q \wedge r$	$\overline{p} \wedge q \wedge \overline{r}$	$\overline{p} \wedge \overline{q} \wedge \overline{r}$	$\overline{p} \wedge \overline{q} \wedge r$
	r	$\overline{r}$		r

Рис. 3.1 Карты Карно: *а*) для $n = 2$, *б*) для $n = 3$

При упрощении булевой функции с помощью карты Карно применяют следующий алгоритм:

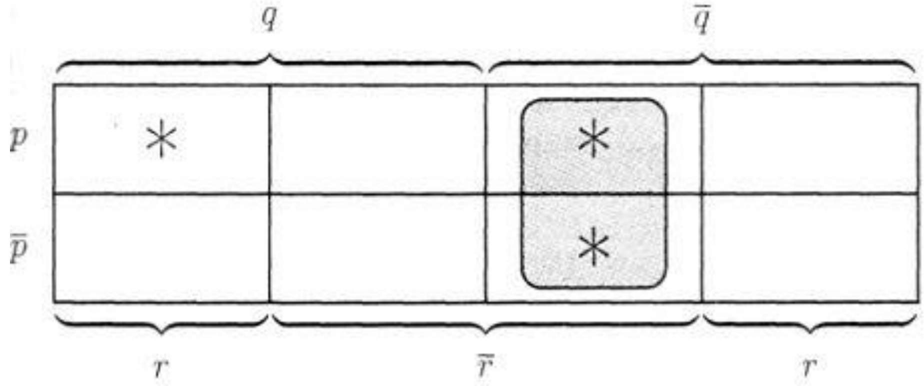
1) ячейки таблицы, соответствующие элементарным конъюнкциям, входящим в ДНФ функции f, обозначить каким-либо символом, например «*»;

2) сформировать из помеченных ячеек прямоугольные блоки максимального размера, в совокупности покрывающие все ячейки с символами «*».

3) записать выражения, соответствующие сформированным блокам, и объединить их операцией дизъюнкции.

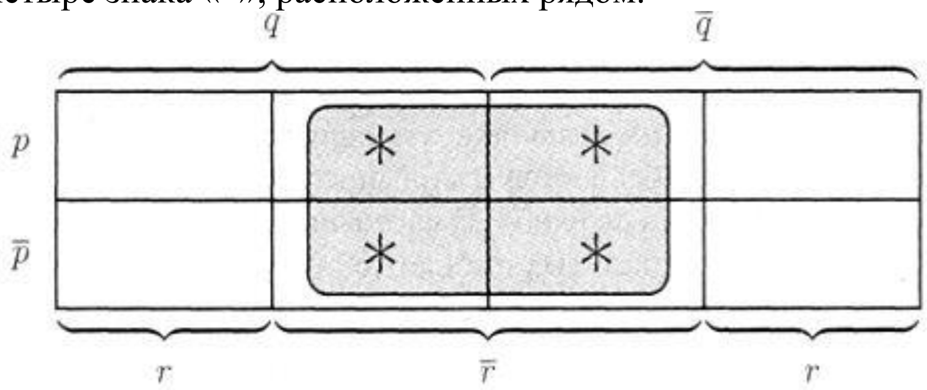
Проиллюстрируем метод работы с Картой Карно на примерах.

Пример 3.5. $f(p,q,r) = pqr \vee p\bar{q}\bar{r} \vee \bar{p}\bar{q}\bar{r}$. Соответствующая карта Карно имеет вид:



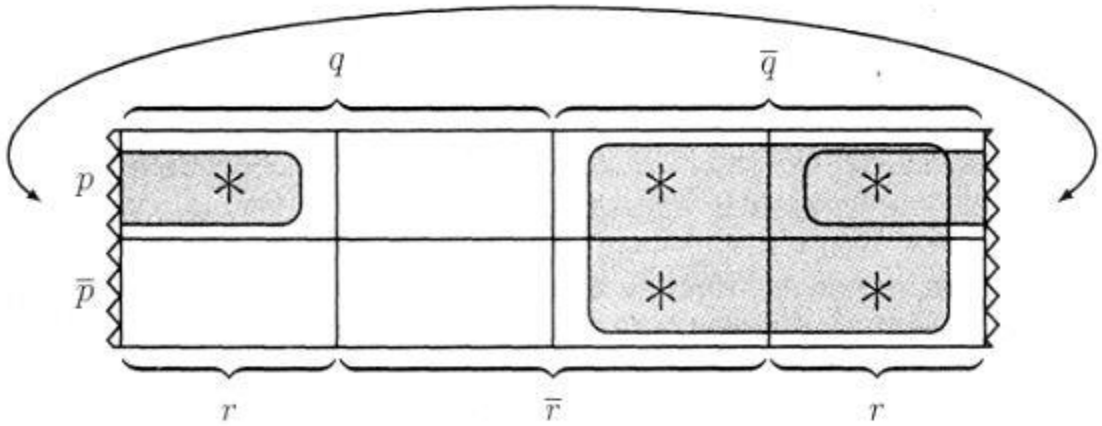
Элементарные конъюнкции, ячейки которых расположены рядом ($p\bar{q}\bar{r}$ и $\bar{p}\bar{q}\bar{r}$), могут быть сведены к одной конъюнкции, содержащей только две переменные: $\bar{q}$ и $\bar{r}$. В итоге выражение $f(p,q,r)$ можно записать в виде $f(p,q,r)$ = pqr $\vee$ $\bar{q}\bar{r}$.

Пример 3.6. Булева функция $f(p, q, r) = pq\bar{r} \vee p\bar{q}\bar{r} \vee \bar{p}q\bar{r} \vee \bar{p}\bar{q}\bar{r}$ содержит в карте Карно четыре знака «*», расположенных рядом.



Всё выражение $f(p, q, r)$ сводится к $f(p, q, r) = \bar{r}$.

Пример 3.7. Булева функция $f(p, q, r)$ = pqr $\vee\ p\bar{q}r \vee p\bar{q}\bar{r} \vee \bar{p}\bar{q}r \vee \bar{p}\bar{q}\bar{r}$.
Карта Карно:



Учтём, что конъюнкции pqr и $p\bar{q}r$ можно также считать соседними, если «склеить» боковые стороны таблицы. Поэтому $f(p, q, r)$ можно записать в виде $f(p, q, r) = \bar{q} \vee pr$.

3.3.2 Тупиковые и минимальные ДНФ

Если из дизъюнкции простых импликантов функции f нельзя отбросить ни одного слагаемого (иначе поменяется таблица истинности), то говорят, что получена *тупиковая ДНФ (ТДНФ)* функции f.

Тупиковая ДНФ функции f, содержащая минимальное число переменных или их отрицаний, называется *минимальной ДНФ (МДНФ)* функции f.

Алгоритм 7. Построение тупиковых и минимальных ДНФ:

1 По таблице истинности строится СДНФ функции f.

2 Строим сокращённую ДНФ функции f.

3 Занумеруем в любом порядке слагаемые сокращённой ДНФ функции f.

4 Составляем *таблицу покрытий.* Слагаемые СДНФ функции f пишем в первой строке, а слагаемые сокращённой ДНФ функции f вместе с номерами – в первом столбце. Если слагаемое сокращённой ДНФ функции f целиком входит в слагаемое СДНФ функции f , то на пересечении соответствующей строки и столбца пишем номер слагаемого сокращённой ДНФ функции f.

5 Составляем *решёточное выражение*. В каждом столбце числа соединяем знаком дизъюнкции и берём конъюнкцию этих дизъюнкций.

6 Раскрываем скобки в решёточном выражении и воспользуемся правилом поглощения.

7 Каждое слагаемое в полученном выражении соответствует тупиковой ДНФ функции f. Для восстановления тупиковой ДНФ функции f надо взять дизъюнкции тех слагаемых, номера которых указаны в полученном выражении.

8 Тупиковые ДНФ функции f с минимальным числом переменных или их отрицаний являются минимальными ДНФ функции f.

3.3.3 Минимизация конъюнктивных нормальных форм

Для построения *минимальной конъюнктивной нормальной формы (МКНФ)* функции f нужно построить минимальную ДНФ функции $\bar{f}$ (отрицание функции f). В полученной минимальной ДНФ заменить знак & на $\vee$, знак $\vee$ заменить на знак &, а над каждой переменной поставить знак отрицания. Это и будет минимальная КНФ функции f.

Пример 3.8. Построить МКНФ функции $f(x,y,z)$, заданной вектором её значений (0 1 1 0 0 1 0 0).

Строим таблицу истинности для функции $\bar{f}$. Для этого значения 0 заменим на 1, а значения 1 заменим на 0.

x	y	z	$\bar{f}(x,y,z)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Строим СДНФ функции $\bar{f}$.

СДНФ $\bar{f} = \bar{x}\bar{y}\bar{z} \vee \bar{x}yz \vee x\bar{y}\bar{z} \vee xy\bar{z} \vee xyz$.

Строим СКНФ функции $\bar{f}$.

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

Раскроем скобки в выражении для СКНФ функции $\bar{f}$. Получим сокращённую ДНФ $\bar{f}$.

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

= $(x\bar{x} \vee y\bar{x} \vee \bar{z}\bar{x} \vee xy \vee yy \vee \bar{z}y \vee x\bar{z} \vee y\bar{z} \vee \bar{z}\bar{z})$ $(\bar{x} \vee y \vee \bar{z})$ =

= $(0 \vee y\bar{x} \vee \bar{z}\bar{x} \vee xy \vee yy \vee \bar{z}y \vee x\bar{z} \vee y\bar{z} \vee \bar{z}\bar{z})$ $(\bar{x} \vee y \vee \bar{z})$ = $(y \vee \bar{z})$ $(\bar{x} \vee y \vee \bar{z})$ = $yx \vee \bar{z}x \vee y\bar{y} \vee \bar{z}\bar{y} \vee yz \vee \bar{z}z = yx \vee \bar{z}x \vee y\bar{y} \vee \bar{z}\bar{y} \vee yz \vee 0 = xy \vee x\bar{z} \vee \bar{y}\bar{z} \vee yz$ = сокр ДНФ $\bar{f}$.

Составим таблицу покрытий.

	$\bar{x}\bar{y}\bar{z}$	$\bar{x}yz$	$x\bar{y}\bar{z}$	$xy\bar{z}$	xyz
(1) xy				1	1
(2) $x\bar{z}$			2	2	
(3) $\bar{y}\bar{z}$	3		3		
(4) yz		4			4

Решёточное выражение равно 34(2 ∨ 3)(1 ∨ 2))(1 ∨ 4) = (342 ∨ 343)(1 ∨ 21 ∨ 14 ∨ 24) = (342 ∨ 34)(1 ∨24) = 34(1 ∨24) =341 ∨ 3424 = 134 ∨ 234.

Выражению 134 соответствует тупиковая ДНФ функции $\bar{f}$, равная $xy \vee \bar{y}\bar{z} \vee yz$, которая содержит 6 переменных и отрицаний переменных. Выражению 234 соответствует тупиковая ДНФ функции $\bar{f}$, равная $x\bar{z} \vee \bar{y}\bar{z} \vee yz$, которая содержит 6 переменных и отрицаний переменных.

Так как обе тупиковые ДНФ функции $\bar{f}$ содержат по 6 переменных или отрицаний переменных, то каждая из них является минимальной ДНФ функции $\bar{f}$. Построим по ним минимальные КНФ функции $\bar{f}$.

В формуле $xy \vee \bar{y}\bar{z} \vee yz$ заменим знак & на знак ∨, знак ∨ заменим на знак &, а над каждой переменной поставим знак отрицания. Получим форму $(\bar{x} \vee \bar{y})(y \vee z)(\bar{y} \vee \bar{z})$. Это минимальная КНФ функции f.

В формуле $x\bar{z} \vee \bar{y}\bar{z} \vee yz$ заменим знак & на знак ∨, знак ∨ заменим на знак &, а над каждой переменной поставим знак отрицания. Получим форму $(\bar{x} \vee z)(y \vee z)(\bar{y} \vee \bar{z})$. Это минимальная КНФ функции f.

4 ПОЛНОТА СИСТЕМЫ БУЛЕВЫХ ФУНКЦИЙ

4.1 Классы функций, сохраняющих константу

Класс T_0.

Функция $f(x_1, x_2,\ldots, x_n)$ принадлежит классу T_0, если $f(0, 0, \ldots,0) = 0$.

Пример 4.1. Определить, принадлежит ли к классу T_0 функция $f(x, y) = x\vee y$. Так как $f(0, 0) = 0\vee 0 = 0$, то функция $f(x, y) = x\vee y$ принадлежит классу T_0.

Классу T_0 принадлежат функции 0, x, xy, $x\vee y$, $x\oplus y$.

Классу T_0 не принадлежат функции 1, $\bar{x}$, $x \mid y$, $x \to y$, $x \downarrow y$, $x \sim y$.

Теорема. Класс T_0 замкнут относительно суперпозиции функций, то есть суперпозиция функций из класса T_0 принадлежит классу T_0.

Класс T_1.

Функция $f(x_1, x_2,\ldots, x_n)$ принадлежит классу T_0, если $f(1, 1, \ldots,1)=1$.

Пример 4.2. Определить, принадлежит ли к классу T_1 функция $f(x, y) = x\oplus y$. Так как $f(1, 1) = 1\oplus 1 = 0$, то функция $f(x, y) = x\oplus y$ не принадлежит классу T_1.

Классу T_1 принадлежат функции 1, x, xy, $x\vee y$, $x \to y$, $x \sim y$.

Классу T_1 не принадлежат функции 0, $\bar{x}$, $x\oplus y$, $x \mid y$, $x \downarrow y$.

Теорема. Класс T_1 замкнут относительно суперпозиции функций, то есть суперпозиция функций из класса T_1 принадлежит классу T_1.

4.2 Двойственные функции. Класс самодвойственных функций

Функция $f^*(x_1, x_2,\ldots, x_n) = \bar{f}(\bar{x}_1, \bar{x}_2,\ldots, \bar{x}_n)$ называется *двойственной функцией* для функции $f(x_1, x_2,\ldots, x_n)$. При построении двойственной функции на место каждой переменной ставится её отрицание и берётся отрицание от всей функции.

Пример 4.3. Определить двойственную функцию для функции $f(x, y) = x\vee y$.

Двойственная функция для функции $f(x, y) = x\vee y$ равна $f^*(x, y) = \bar{f}(\bar{x}, \bar{y}) = \overline{\bar{x} \vee \bar{y}} = \bar{\bar{x}}\&\bar{\bar{y}} = x\&y$. Очевидно, что конъюнкция является двойственной функцией для дизъюнкции: $(x\vee y)^* = x\&y$.

Двойственную функцию можно получить и по таблице истинности исходной функции f. Для этого строим функцию $\bar{f}$ и запишем её значения в обратном порядке. Это и будет таблица истинности двойственной функции.

Функция, совпадающая со своей двойственной функцией ($f = f^*$), называется *самодвойственной функцией.* Класс самодвойственных функций обозначается буквой S.

Теорема. Класс самодвойственных функций S замкнут относительно суперпозиции функций, то есть суперпозиция функций из класса S принадлежит классу S.

Критерий самодвойственности функции. Для самодвойственности функции необходимо и достаточно, чтобы на любых двух противоположных наборах значений переменных функция принимала разные значения (то есть все равноудалённые от концов строки значений функции числа противоположны).

Классу S принадлежат функции x, $\bar{x}$.

Классу S не принадлежат функции 0, $x \vee y$.

4.3 Класс линейных функций

Функция $f(x_1, x_2, \ldots, x_n)$, полином Жегалкина которой имеет вид $a_0 \oplus a_1x_1 \oplus \ldots \oplus a_nx_n$, называется *линейной функцией*. Иными словами, в полиноме линейной функции нет слагаемых, содержащих конъюнкцию.

Пример 4.4. Функция $f(x) = \bar{x} = x \oplus 1$является линейной.

Теорема. Класс линейных функций замкнут относительно суперпозиции, то есть суперпозиция линейных функций является линейной функцией.

Класс линейных функций обозначается буквой L.

Классу L принадлежат функции 0,1, x, $\bar{x}$, $x \oplus y$,, $x \sim y$.

Классу L не принадлежат функции 0, xy, $x \vee y$, $x \to y$, $x \mid y$, $x \downarrow y$.

4.4 Класс монотонных функций

Если для двух наборов длины n из нулей и единиц $a = (a_1, \ldots, a_n)$ и $b = (b_1, \ldots, b_n)$ выполнены условия $a_1 \le b_1, \ldots, a_n \le b_n$, то говорят, что $a \le b$. При невыполнении этих условий наборы a и b *несравнимы*. Про наборы разной длины также говорят, что они несравнимы.

Функция $f(x_1, x_2, \ldots, x_n)$ называется *монотонной*, если для всех наборов $a = (a_1, \ldots, a_n)$ и $b = (b_1, \ldots, b_n)$ из условия $a \le b$ следует $f(a) \le f(b)$.

Пример 4.5. Функция $f(x,y) = x \vee y$ является монотонной.

Теорема. Класс монотонных функций замкнут относительно суперпозиции, то есть суперпозиция монотонных функций является монотонной функцией.

Класс линейных функций обозначается буквой M.

Классу M принадлежат функции 0,1, x, xy, $x \vee y$.

Классу M не принадлежат функции $\bar{x}$, $x \oplus y$, $x \to y$, $x \mid y$, $x \downarrow y$, $x \sim y$.

4.5 Теорема Поста

Система функций А называется полной, если любую булеву функцию можно получить суперпозицией функций из А.

Теорема Поста. Система булевых функций полна в множестве всех булевых функций тогда и только тогда, когда эта система не содержится целиком ни в одном из следующих классов: T_0, T_1, L, S, M.

Другими словами, для полноты системы А необходимо и достаточно, чтобы в А имелись: хотя бы одна функция, не сохраняющая 0; хотя бы одна функция, не сохраняющая 1; хотя бы одна не самодвойственная функция; хотя бы одна нелинейная функция; хотя бы одна немонотонная функция.

Минимальное полное множество логических функций называется базисом. Если из базиса удалить любую функцию, то система перестанет быть полной.

Пример 4.6. Проверить систему функций $\{\&,\neg\}$ на полноту.

Функция $\bar{x}$ не принадлежит классам T_0 и T_1.

Функция xy не является линейной функцией.

Функция xy не является самодвойственной функцией.

Так как $0 < 1$, но $\overline{0} = 1 > \overline{1} = 0$, то функция $\bar{x}$ не является монотонной функцией.

Поэтому система функций $\{\&,\neg\}$ не содержится целиком ни в одном из классов T_0, T_1, L, S, M. Следовательно, $\{\&,\neg\}$ – полная система функций согласно теореме Поста.

Пример 4.7. Проверить систему функций $\{\vee, \oplus\}$ на полноту.

Так как обе функции принадлежат классу T_0 ($0\vee 0 = 0$, $0\oplus 0 = 0$), то система функций $\{\vee, \oplus\}$ не является полной по теореме Поста.

5 ЛОГИКА ПРЕДИКАТОВ

Не любые высказывания и логические рассуждения могут быть описаны на языке логики высказываний. Высказывания, описывающие некоторые свойства логики объектов, или отношения между объектами, выходят за рамки высказываний.

Например, мы не сможем судить о логической правильности такого рассуждения: «Каждое натуральное число является корнем некоторого квадратного уравнения. Число 5 – натуральное. Следовательно, 5 является корнем некоторого квадратного уравнения». Поэтому следует расширить логику высказываний и построить такую логическую систему, в рамках которой можно было бы исследовать структуру и содержание тех высказываний, которые в рамках алгебры высказываний считались бы элементарными. Такой логической системой является логика предикатов, а алгебра высказываний – ее составной частью.

Логика предикатов начинается с анализа строения высказываний, которые выражают тот факт, что объекты обладают некоторыми свойствами, или находятся между собой в некоторых отношениях. Понятие «свойства» и понятие «отношения» рассматриваются как частный случай общего понятия «предиката». Объекты, о которых говорится в высказывании, называются термами или предметными константами. Предметные константы, подобно константам в математике, определяют значения, которые могут быть приписаны в высказываниях предметным переменным. При этом каждой переменной соответствует своё множество предметных констант. Над переменными и константами определяются функции так же, как и в математике.

Понятие предиката обобщает понятие «высказывание».

Предикатом называется функция, аргументы которой принимают значения из некоторого множества, а сама функция – значение 0 («ложь») или 1 («истина»).

Неформально определение, предикат – это высказывание, содержащее неизвестную (или несколько неизвестных), т. е. в него можно подставлять аргументы. Если аргумент один, то предикат выражает свойство аргумента, если больше –то отношение между аргументами.

Пример предиката: ФАМИЛИЯ = «Петров». Здесь ФАМИЛИЯ – переменная, «Петров» – константа.

Предикаты, в которых описывается некоторое свойство объекта, называется предикат-свойство.

Если предикат определяет отношение между несколькими объектами, то такой предикат называется предикат-отношение.

В зависимости от того, между скольким числом объектов устанавливаются отношения, различают одноместные, двуместные, трёхместные и n-местные отношения.

Одноместным предикатом называется функция одной переменной, значениями которой являются высказывания об объектах, представляющих значения аргумента. Т.е. одноместный предикат $P(x)$ это произвольная функция переменной x, определённая на некотором множестве M и принимающая значения из множества $\{0,1\}$. Множество M на котором определен предикат $P(x)$, называется предметной областью или областью определения предиката. Переменная x называется предметной переменной. Таким образом, одноместный предикат $P(x)$ – это утверждение об объекте x, где x рассматривается как переменная. При фиксации значения переменной x об утверждении можно сказать, истинно оно или ложно. То есть, если в $P(x)$ вместо x подставить конкретный изучаемый объект a, то получаем высказывание, принадлежащее алгебре высказываний.

Областью истинности предиката $P(x)$ заданного на множестве M называется совокупность всех x из M, при которых данный предикат обращается в истинное высказывание: $I\,P = \{x \in M\ P(x)=1\}$.

Иными словами, область истинности предиката есть подмножество его предметной области, на котором данный предикат принимает значение 1.

Предикат $P(x)$, определенный на M называется тождественно истинным, если $IP=M$ и тождественно ложным, если $IP = \{\varnothing\}$.

Чтобы задать предикат от n аргументов (n – местный предикат), прежде всего следует указать множества $M_1, \ldots M_n$ – области изменения предметных переменных $x_1, \ldots, x_n$.

Предикат называется n-местным (n = 1,2 …), если соответствующая функция есть функция от n аргументов.

Высказывание – не что иное, как предикат без аргумента, или предикат с нулевым числом мест.

Пример 5.1. Бинарный предикат: $P(x,y)$= «y делится на x без остатка» - предикат от двух переменных, где x и y могут принимать значения или из множества натуральных чисел, или из множества целых чисел, или из множества многочленов.

Пример 5.2. Трёхместный предикат: $P(x,y,z) = (z = x + y)$.

Предикатную функцию $P(x, y)$ можно рассматривать как функцию, определённую на декартовом квадрате N2. Множество тех пар (x, y) для которых данная функция принимает значение истины, есть область истинности предиката $P(x, y)$. Табличную запись функции называют матрицей предиката.

Предикаты P и Q, определенные на множестве $M = M_1 \times\ldots\times M_n$, называются равносильными (пишут $P \equiv Q$), если $P(x_1,\ldots,x_n) = Q(x_1,\ldots,x_n)$ для любого набора предметных переменных $(x_1,\ldots,x_n)$ из M.

Применение логических связок

Расширение логики высказываний до логики предикатов получается за счет включения в формулы утверждений, являющихся предикатами. Так как предикаты – это отображения со значениями во множестве высказываний, где введены логические операции (конъюнкция, дизъюнкция, отрицание, импликация и др.), то эти операции (логические связки) естественно определяются и для предикатов.

При этом значения истинности сложных предикатов находятся в зависимости от значений связываемых предикатов по тем же правилам, что и для высказываний.

Пусть R(x) и E(x) – два одноместных предиката, определённых на некотором множестве M.

Конъюнкция. P1(x) ≡ R(x) & E(x) – это предикат, который истинен для тех и только для тех объектов из M, для которых оба предиката истинны. Таким образом, область истинности предиката P1(x) равна пересечению областей истинности предикатов R(x) и E(x).

Дизъюнкция. P2(x) ≡ R(x) ∨ E(x) – это предикат, который ложен для тех и только для тех объектов из M, для которых оба предиката ложны. Область истинности предиката P2(x) равна объединению областей истинности предикатов R(x) и E(x).

Отрицание. P3(x) ≡ ¬R(x) – это предикат, который истинен для тех и только для тех объектов из M, для которых предикат R(x) ложен. Его область истинности является дополнением области истинности предиката R(x).

Операции логики над многоместными предикатами определяются аналогично. Пример: Пусть имеется два предиката R(x, y) и Q(x, y), определённых на множестве M.

Тогда предикат P(x, y, z) ≡ R(x, y) & Q(y, z) – некоторый трёхместный предикат от x, y, z.

Чтобы определить для каких значений предикат P(x, y, z) принимает истинные значения, а для каких ложные, необходимо произвести унификацию переменных, то есть присвоить переменным некоторые конкретные значения из множества M.

Пусть $x = a$, $y = b$, $z = c$, где $a,b,c \in$ M, P(a, b, c) ≡ R(a, b) & Q(b, c).

Предикат P(a, b, c) = 1, когда R(a, b) = 1 и Q(b, c) = 1.

Пример 5.3.

(ФАМИЛИЯ = «Петров») & (ВУЗ = «РГУПС») & (1<КУРС>4).

Это сложное высказывание будет истинным для студента РГУПС 2-го или 3-го курса с фамилией Петров. Для всех остальных студентов значения предиката будет «ложь».

Введение переменных и функций позволяет более подробно описывать предметную область, чем при описании через элементарные высказывания, смысл которых нам был не важен. Но если там решение можно было найти перебором, истинностных значений элементарных высказываний, то перебор по всевозможным значениям предметных переменных становится невозможным, поэтому требуются другие методы поиска решений.

5.1 Кванторные операции

Функциональная природа предиката влечет за собой введение еще одного понятия – квантора. Квантор – это общее название для логических операций, ограничивающих область истинности какого-либо предиката. Выражения:

«каждому», «для всех» и т.п. служат Примером квантификации. Именно кванторы делают теорию предикатов гибкой.

В математической логике наиболее употребимы квантор всеобщности $\forall$ (читается «для всех») и квантор существования $\exists$ (читается «существует»).

Пусть P(x) – одноместный предикат, определенный на множестве M. Под выражением $\forall x$P(x) понимают высказывание, истинное если P(x) истинно для каждого элемента $x \in$ M и ложное в противном случае. Иными словами, истинность высказывания $\forall x$P(x) означает, что область истинности предиката P(x) совпадает с областью изменения переменной x. Читается это высказывание: «для всякого x истинно P(x)».

Под выражением $\exists x$P(x) понимают высказывание, истинное если существует $x \in$ M , для которого P(x) истинно, и ложное в противном случае. Иными словами истинность высказывания $\exists x$P(x) означает, что область истинности предиката P(x) не пуста. Читается это высказывание: «существует x, при котором P(x) истинно».

Оба квантора (всеобщности и существования) можно рассматривать как отображения множества одноместных предикатов во множество высказываний (0-местных предикатов), то есть отображения, уменьшающие число аргументов на 1.

О высказывании $\forall x$P(x) (соответственно, $\exists x$P(x)) говорят, что оно получено из предиката P навешиванием квантора всеобщности (соответственно, квантора существования) по переменной x.

Переменная, на которую навешен квантор, называется связанной переменной.

Если некоторый предикат P(x) определён на конечном множестве M = $\{a_1,\ldots,a_k\}$, то справедливы следующие тождества:

$$\forall x P(x)\ |P(a_1)\&\ldots,\&P(a_k),$$
$$\exists x P(x)\ |P(a_1)\vee\ldots\vee P(a_k).$$

Таким образом, кванторы можно рассматривать как обобщения логических связок. В случае предикатов, определенных на бесконечных множествах, квантор всеобщности обобщает конъюнкцию, а квантор существования – дизъюнкцию.

Навешивать кванторы можно и на многоместные предикаты и вообще на любые логические выражения. Выражение, на которое навешивается квантор $\forall x$ или $\exists x$, называется областью действия квантора; все вхождения переменной x в это выражение являются связанными.

Не связанные кванторами переменные называются свободными переменными. Например, к предикату от двух переменных кванторные операции можно применить к одной переменной или к двум переменным. Получаем следующие высказывания:

$$\forall x\ P(x,y); \forall y\ P(x,y); \exists x\ P(x,y); \exists y\ P(x,y);$$

$$\forall x \forall y\ P(x, y);\ \forall x \exists y\ P(x, y);\ \exists x \forall y\ P(x, y);\ \exists x \exists y\ P(x, y);$$

В общем случае изменение порядка следования кванторов изменяет смысл высказывания и его логическое значение.

Пример 5.4. Пусть $P(x,y)$ = «x является матерью y». Тогда $\forall y \exists x\ P(x,y)$ = = «у каждого человека есть мать»,

$\exists x \forall y\ P(x,y)$ = «существует мать всех людей».

Таким образом, перестановка кванторов изменяет смысл высказывания и его логическое значение (первое высказывание истинно, второе – ложно).

5.2 Формулы логики предикатов

Формула логики предикатов определяется индуктивно по следующей схеме:

1) всякая пропозициональная переменная (т.е. 0 – местный предикат) есть формула;

2) если P – n-местный предикат, то $P(x_1, \ldots, x_n)$ – формула. Все переменные $x_1, \ldots, x_n$ – свободные переменные, связанных переменных в этой формуле нет;

3) если A – формула, то $\neg A$ – формула с теми же свободными и связанными переменными, что и в формуле A;

4) если A и B – формулы, причем нет таких переменных, которые были бы связанными в одной формуле и свободными в другой, то выражения (A&B), $(A \vee B)$, $(A \rightarrow B)$, $(A \sim B)$, $(A \oplus B)$ суть формулы, в которых свободные переменные формул A и B остаются свободными, а связанные переменные формул A и B остаются связанными;

5) если A – формула, содержащая свободную предметную переменную x, то$\forall x$ A и $\exists x$ B – тоже формулы. Переменная x в них связана. Остальные же переменные, которые в формуле A были свободны, остаются свободными и в новых формулах. Переменные, которые были связаны в A, связаны и в новых формулах;

6) Других формул, кроме построенных по правилам пяти предыдущих пунктов, нет.

Из этого определения ясно, что всякая формула алгебры высказываний является формулой логики предикатов. Как обычно, часть скобок, определяющих порядок действий в формуле, можно опускать.

Пример 5.5. Схема исчисления высказываний $\overline{\overline{A}} = A$. Следовательно, справедлива формула: $\forall x\ \overline{\overline{P(x,y)}} |\ \forall x\ P(x,y)$.

Формула вида $P(x_1, \ldots, x_n)$, где P – n-местный предикат, называется атомарной (или элементарной).

Литеральной формулой (или литералом) называют атомарную формулу или отрицание атомарной формулы.

Атомарная формула называется положительным литералом, а ее отрицание – отрицательным литералом.

Дизъюнкт – это дизъюнкция конечного числа литералов.

Если дизъюнкт не содержит литералов, его называют пустым дизъюнктом. Говорят, что формула находится в конъюнктивной нормальной форме, если это конъюнкция конечного числа дизъюнктов. Имеет место теорема о том, что для любой бескванторной формулы существует формула, логически эквивалентная исходной и находящаяся в конъюнктивной нормальной форме.

Как и формулы исчисления высказываний, формулы исчисления предикатов делятся на три класса: общезначимые формулы (истинны при всех интерпретациях) невыполнимые (ложны при всех интерпретациях), выполнимые (истинны хотя бы при одной интерпретации).

Определение.

Пусть $P(x_1, x_2, \ldots, x_n)$ – некоторый n-местный предикат на некотором множестве М. Этот предикат называется:

a) общезначимым (тождественно-истинным), если для любого набора значений аргументов его значение равно истина;

b) тождественно-ложным, если для любого набора значений аргументов, его значение равно ложь;

c) выполнимым, если существует хотя бы один набор значений аргументов, для которых его значение равно истина.

Пример тождественно-ложного предиката:

Пусть $R(x_1,\ldots,x_n)$ и $Q(x_1,\ldots,x_n)$ – два n-местных предиката от одних и тех же переменных, заданных на одном и том же множестве. Предикат $Q(x_1,\ldots,x_n)$ называется следствием предиката $R(x_1,\ldots,x_n)$, если для любого набора переменных предикат $R(x_1,\ldots,x_n)$ является истинным, то предикат $Q(x_1,\ldots,x_n)$ также истинен.

Следствие обозначается как $R(x_1,\ldots,x_n) \Rightarrow Q(x_1,\ldots,x_n)$.

Предикаты $R(x_1,\ldots,x_n)$ и $Q(x_1,\ldots,x_n)$ равносильны тогда и только тогда, когда каждый из них есть следствие другого.

Формулы A и B называются равносильными на множестве M, если при любой замене имеющихся в них простых формул на предикаты на множеств M эти формулы превращаются в равносильные предикаты.

Теорема Чёрча: не существует алгоритма, который для любой формулы логики предикатов устанавливает, общезначима она или нет.

Предварённая нормальная форма

Для облегчения анализа сложных суждений формулы логики предикатов рекомендуется приводить к предваренной нормальной форме.

Говорят, что формула логики предикатов находится в предваренной (или префиксной) нормальной форме, если она имеет вид $Q_1x_1,\ldots,Q_nx_n\ F$, где каждый Q_i есть квантор всеобщности или существования (т.е. Q_ix_i обозначает $\forall x_i$ или $\exists x_i$), переменные x_i,y_j различны при $i \neq j$, а F – формула, содержащая операции $\&,\vee,\neg$ и не содержащая кванторов (причем знаки отрицания отнесены только к элементарным предикатам и высказываниям, то есть к элементарным формулам).

Выражение $Q_1x_1, \ldots, Q_nx_n$ называют префиксом (или кванторной приставкой), а формулу F – матрицей.

Если все Q_i –∀, то эта форма называется ∀- формулой. Если все Q_i – ∃, то эта форма называется ∃ - формулой.

Процедура приведения формулы логики предикатов к префиксной нормальной форме:

1) сначала избавляются от операций импликации, эквивалентности и неравнозначности, выразив их через логические связки ∨, & и ¬ по законам логики;

2) доводят символы отрицания до символов предикатов, используя правила де Моргана, и избавляются от двойных отрицаний:

$$\overline{\forall x P(x)} \equiv \exists x \overline{P(x)}$$

$$\overline{\exists x P(x)} \equiv \forall x \overline{P(x)}$$

$$\bar{\bar{A}} \equiv A;$$

3) для формул, содержащих подформулы вида $\forall x P(x) \vee \forall x Q(x)$, $\exists x P(x) \vee \exists x Q(x)$ и т.п., вводят новые переменные, позволяющие выносить знаки кванторов наружу. Например:

$$\exists x P(x) \ \& \ \exists x Q(x) \equiv \exists x \exists y (P(x) \ \& \ Q(y));$$

$$\forall x P(x) \vee \forall x Q(x) \equiv \forall x \forall y (P(x) \vee Q(y));$$

$$\forall x P(x) \vee \exists x Q(x) \equiv \forall x \exists y (P(x) \vee Q(y));$$

4) используя все известные равносильности логики предикатов, получают формулу в виде префиксной нормальной формы.

6 ИСЧИСЛЕНИЕ ВЫСКАЗЫВАНИЙ

6.1 Основные понятия исчисления высказываний

Исчисление высказываний – это логическая система, интерпретацией которой является алгебра высказываний (булевы функции).

Для задания любого исчисления нужно определить символы этого исчисления (алфавит), формулы и выводимость формул.

Алфавит исчисления высказываний состоит из следующих символов:

1) переменные $x,y,z,\ldots;$

2) логические связки &, $\vee$, $\to$, $\neg$;

3) запятая, скобки (левая и правая): , ( ).

Определим понятие *формулы исчисления высказываний.*

1. Любая переменная есть формула.

2. Если А и В есть формулы, то (А & В), (А $\vee$ В), (А $\to$ В), $\bar{A}$ есть формулы.

Пример 6.1. Выражения $((a \,\&\, b) \to c)$, $((\bar{a} \vee \bar{b}) \,\&\, c)$, $(\bar{a} \to c)$ являются формулами. Выражения $a \vee \bar{b})$, $(\bar{a} \to)$, $(a \,\&\&\, b)$ не являются формулами.

Определим понятие *подформулы.*

1 Единственной подформулой переменной является сама переменная.

2 Для формулы $\overline{\mathrm{A}}$ подформулами являются $\overline{\mathrm{A}}$ и все подформулы формулы А.

3 Для формулы вида (А * В) (где * – один из знаков &, $\vee$, $\to$) подформулами являются (А * В) и все подформулы формул А и В.

Пример 6.2. Определим все подформулы формулы $((\bar{a} \vee \bar{b}) \,\&\, \bar{c})$,

Подформулами формулы $((\bar{a} \vee \bar{b}) \,\&\, \bar{c})$ являются $((\bar{a} \vee \bar{b}) \,\&\, \bar{c})$, $((\bar{a} \vee \bar{b})$, $\bar{a}$, $\bar{b}$, $\bar{c}$, a, b ,c.

Введём в запись формул некоторые упрощения. Будем опускать итоговые скобки, а также установим приоритет связок: $\neg$, &, $\vee$, $\to$.

Пример 6.3. Упростить запись формулы $((a \,\&\, b) \vee (a \to c))$. Упрощённая запись формулы будет $a \,\&\, b \vee (a \to c)$.

6.2 Система аксиом исчисления высказываний и правила вывода

Система аксиом исчисления высказываний состоит из 11 аксиом:

1) $x \to (y \to x);$

2) $(x \to (y \to z)) \to ((x \to y) \to (x \to z));$

3) $x \,\&\, y \to x;$

4) $x \,\&\, y \to y;$

5) $(z \to x) \to ((z \to y) \,\&\, (z \to x \,\&\, y));$

6) $x \to x \vee y$;
7) $y \to x \vee y$;
8) $(x \to z) \to ((y \to z) \to (x \vee y \to z))$;
9) $(x \to y) \to (\bar{y} \to \bar{x})$;
10) $x \to \bar{\bar{x}}$;
11) $\bar{\bar{x}} \to x$.

Правила вывода.

К правилам вывода относятся правило подстановки и правило заключения.

Правило подстановки $П_x^B(A)$ заключается в замене в формуле А переменной x на формулу B.

Правило заключения ПЗ(A, B) позволяет перейти от формул A и $A \to B$ к формуле B.

Доказуемые формулы.

Доказательство в исчислении высказываний – это конечная последовательность формул, каждая из которых есть либо аксиома, либо получена из предыдущих формул с помощью правил подстановки или заключения. Формула А *доказуема*, если в исчислении высказываний существует доказательство, последней формулой которого является формула А.

Пример 6.4. Показать, что формула $x \to x$ доказуема в исчислении высказываний.

Доказательство:

1) $(x \to (y \to z)) \to ((x \to y) \to (x \to z))$ (аксиома);

2) $(x \to (y \to x)) \to ((x \to y) \to (x \to x))$ (к формуле 1 применили правило подстановки $П_z^x(1)$);

3) $x \to (y \to x)$ (аксиома);

4) $(x \to y) \to (x \to x)$ (к формулам 2 и 3 применили правило заключения ПЗ(2,3);

5) $(x \to (y \to x)) \to (x \to x)$ (к формуле 4 применили правило подстановки $П_y^{y \to x}(4)$);

6) $x \to x$ (к формулам 3 и 5 применили правило заключения ПЗ(3,5).

Производные правила вывода.

Производные правила вывода, получаемые с помощью правил подстановки и заключения, позволяют сократить доказательство в исчислении высказываний.

Правило одновременной подстановки $П_{x1,x2,\dots,xn}^{B1,B2,\dots,Bn}(A)$ даёт возможность из доказуемой формулы А с помощью замены переменных $x_1, x_2, \dots, x_n$ на формулы $B_1, B_2, \dots, B_n$ получить доказуемую формулу.

Правило сложного заключения позволяет от доказуемых формул $A_1, A_2, \dots, A_n$ и $(A_1 \to (A_2 \to (A_3 \to (\dots(A_n \to B)\dots)))$ перейти к доказуемой формуле B.

Правило силлогизма даёт возможность от доказуемых формул $A \to B$ и $B \to C$ перейти к доказуемой формуле $A \to C$.

Правило контрапозиции утверждает, что из доказуемости формулы $A \to B$ следует доказуемость формулы $\overline{B} \to \overline{A}$.

Правило снятия двойного отрицания:

а) из доказуемости формулы $A \to \overline{\overline{B}}$ следует доказуемость формулы $A \to B$.

б) из доказуемости формулы $\overline{\overline{A}} \to B$ следует доказуемость формулы $A \to B$.

Пример 6.5. Доказать правило снятия двойного отрицания (а).

Доказательство:

1) $A \to \overline{\overline{B}}$ (доказуема по условию);

2) $\overline{\overline{x}} \to x$ (аксиома);

3) $\overline{\overline{B}} \to B$ (формуле 2 применено правило подстановки $\Pi_x^B(2)$);

4) $A \to B$ (к формулам 1 и 3 применено правило силлогизма).

6.3 Вывод из совокупности формул и правила выводимости

Пусть H – конечная совокупность формул в исчислении высказываний. Конечная последовательность формул в исчислении высказываний есть *вывод из совокупности формул H,*если любая формула этой последовательности либо принадлежит *H,* либо доказуема, либо получена из предыдущих формул последовательности по правилам заключения (правило подстановки применимо только к доказуемым формулам).

Формула А *выводима из совокупности формул H* $(H \vdash A)$, если существует вывод из совокупности формул H, последней формулой которого является формула А. Доказуемая формула выводима из любой (в том числе пустой) совокупности формул.

Пример 6.6. Показать, что из H = {A} выводима формула $B \to A$.

Вывод:

1) А (формула из H);

2) $x \to (y \to x)$ (аксиома);

3) $A \to (B \to A)$ (правило одновременной подстановки $\Pi_{x,y}^{A,B}(2)$).

4) $B \to A$ (правило заключения ПЗ(1,3)).

Правила выводимости.

Пусть H и W – две совокупности формул исчисления высказываний, C – формула исчисления высказываний.

1 Если формула А выводима из совокупности формул H $(H \vdash A)$, то формула А выводима и из объединения формул H и W $(H, W \vdash A)$.

2 Если формула А выводима из совокупности формул *H* и *C(H, C⊢A)*, а формула С выводима из совокупности формул *H (H⊢C)*, то формула А выводима из совокупности формул *H (H⊢A)*.

3 Если формула А выводима из совокупности формул *H* и *C(H, C⊢A)*, а формула С выводима из совокупности формул *W (W⊢C)*, то формула А выводима из объединения формул *H* и *W(H,W⊢A)*.

4 Если формула $C \to A$ выводима из совокупности формул *H (H⊢ C→A)*, то формула А выводима из совокупности формул *H* и *C(H,C⊢A)*.

5 *Теорема дедукции*. Если формула А выводима из совокупности формул *H* и *C(H, C⊢A)*, то формула $C \to A$ выводима из совокупности формул *H (H⊢C→A)*.

6 *Обобщённая теорема дедукции*. Если формула А выводима из совокупности формул $\{C_1, \ldots, C_n\}$ ($\{C_1, \ldots, C_n\} \vdash A$), то формула $C_1 \to (C_2 \to (C_3 \to \ldots (C_n \to A) \ldots))$ доказуема ($\vdash C_1 \to (C_2 \to (C_3 \to \ldots (C_n \to A) \ldots))$).

7 *Правило введения конъюнкции.* Если формулы А и В выводимы из совокупности формул *H (H⊢A, H⊢B)*, то формула A&B выводима из совокупности формул *H (H⊢A&B)*.

8 *Правило введения дизъюнкции.* Если формула С выводима из совокупности формул *H,A (H, A⊢ C)* и выводима из совокупности формул *H,B (H,B⊢C)*, то формула *C* выводима из совокупности формул *H* и $A \vee B$ ($H, A \vee B \vdash C$).

Правила выводимости позволяют доказать ряд законов логики.

Закон перестановки посылок. Формула $(x \to (y \to z)) \to (y \to (x \to z))$ доказуема.

Правило перестановки посылок в доказуемых формулах. Из доказуемости формулы $x \to (y \to z)$ следует доказуемость формулы $y \to (x \to z)$.

Закон соединения посылок. Формула $(x \to (y \to z)) \to (x \,\&\, y \to z)$ доказуема.

Правило соединения посылок в доказуемых формулах. Из доказуемости формулы $x \to (y \to z)$ следует доказуемость формулы $x \,\&\, y \to z$

Закон разъединения посылок. Формула $(x \,\&\, y \to z) \to (x \to (y \to z))$ доказуема.

Правило разъединения посылок в доказуемых формулах. Из доказуемости формулы $x \,\&\, y \to z$ следует доказуемость формулы $x \to (y \to z)$.

Закон исключённого третьего. Формула $x \vee \bar{x}$ доказуема.

Пример 6.7. Показать, что формула $x \to (y \to x\&y)$ доказуема.

Доказательство:

1) $A \to A$ (доказуемая формула);

2) $x\&y \to x\&y$ (правило подстановки $\Pi_A^{x\&y}(1)$).

3) $x \to (y \to x\&y)$ (к формуле 2 применено правило разъединения посылок).

Формулы исчисления высказываний можно интерпретировать как булевы функции.

Теорема. Каждая доказуемая формула является тождественно истинной булевой функцией.

Теорема. Каждая тождественно истинная булева функция доказуема в исчислении высказываний.

Теорема о выводимости. Пусть А – формула исчисления высказываний; $x_1, x_2, \ldots, x_n$ – набор переменных формулы А; $(\alpha_1, \ldots, \alpha_n)$ – произвольный набор из нулей и единиц. Введём следующие обозначения:

$$x_i^{\alpha i} = \{ \begin{array}{l} x\mathrm{i} \text{ при } \alpha_{\mathrm{i}} = 1, \\ \bar{x}_{\mathrm{i}} \text{ при } \alpha_{\mathrm{i}} = 0 \end{array}$$

Рассмотрим совокупность формул Н = $\{x_1^{\alpha 1}, \ldots, x_n^{\alpha n}\}$. Тогда если формула А = 1 на наборе $(\alpha_1, \ldots, \alpha_n)$, то формула А выводима из совокупности формул Н. А если формула А = 0 на наборе $(\alpha_1, \ldots, \alpha_n)$, то из совокупности формул Н выводима формула $\overline{\mathrm{A}}$.

Пример 6.8. Дана формула А = $x_1 \vee x_2 \to x_3$ и наборы (0, 0, 1), (1, 0, 0). Вывести А или $\overline{\mathrm{A}}$ из соответствующей совокупности формул.

Значение формулы А на наборе (0, 0, 1) равно $0 \vee 0 \to 1 = 0 \to 1 = 1$, следовательно формула А выводима из совокупности формул Н = $\{x_1^0, x_2^0, x_3^1\}$ = $\{\bar{x}_1, \bar{x}_2, x_3\}$.

Соответствующий вывод:

1) $\bar{x}_1$ (формула Н);

2) $\bar{x}_2$ (формула Н);

3) $\bar{x}_1$ & $\bar{x}_2$ (введение конъюнкции);

4) $\bar{x}_1$ & $\bar{x}_2 \to \overline{x1 \vee x2}$ (закон логики);

5) $\overline{x1 \vee x2}$ (правило заключения ПЗ(3,4));

6) $x \to (y \to x)$ (аксиома);

7) $\overline{x1 \vee x2} \to (\bar{x}_3 \to \overline{x1 \vee x2})$ (к формуле 6 применено правило одновременной подстановки $\Pi_{x,y}^{\overline{x1 \vee x2}), \bar{x}3}(6)$);

8) $(x_1 \vee x_2 \to x_3) \to x_3$ (к формулам 4 и 7 применено правило заключения ПЗ(4,7));

9) $\bar{x}_3 \to \overline{x1 \vee x2 \to x3}$ (к формуле 8 применено правило контрапозиции);

10) $\bar{x}_3$ (формула Н);

11) $\overline{x1 \vee x2 \to x3}$ (к формулам 10 и 9 применено правило заключения ПЗ(10,9));

7 ЭЛЕМЕНТЫ ТЕОРИИ АЛГОРИТМОВ

Основные понятия

Дадим интуитивное определение алгоритма.

Алгоритмом называется двусортное множество $\langle M_\rho, R_2 \rangle$, где M_ρ – множество правил (процедур) решения задачи, обладающих следующими свойствами:

массовость – инвариантность относительно входной информации;

детерминированность – однозначность применения этих правил на каждом шаге;

результативность – получение после применения этих правил информации, являющейся результатом;

элементарность (прозрачность) – отсутствие необходимости дальнейшего уточнения правил.

Символ R_2 – бинарное отношение в множестве M_ρ, $R_2 \subset M_\rho^{\ 2}$, $(\rho_i, \rho_j) \in R_2$, если после процедуры ρ_i выполняется процедура ρ_j.

Алгоритм можно представить в виде графа, каждая вершина которого соответствует правилу, и бинарное отношение R_2 определяет порядок выполнения этих правил. Если при этом вершина является началом или концом только одной дуги, то правило называется *арифметическим*. Если правило является концом только одной дуги, а началом более чем одной дуги, то правило называется *логическим*. После выполнения логического правила происходит ветвление вычислительного процесса согласно полученному результату.

Чаще всего алгоритм представляется в виде схемы, состоящей из стандартных блоков, соответствующих арифметическим и логическим правилам.

Уточним неформальное определение алгоритма.

Для этого рассмотрим проблему результативности для множеств, являющихся носителями всех математических объектов.

Множество M называется *разрешимым* (рекурсивным), если существует алгоритм A_M, который по любому объекту a даёт ответ, принадлежит a множеству M или нет. Алгоритм A_M называется разрешающим алгоритмом для M.

Или более строго: множество M называется *разрешимым*, если оно обладает общерекурсивной характеристической функцией, т.е. вычислимой всюду определённой функцией χ_M:

$$\chi_M(a) = \begin{cases} 1, & a \in M, \\ 0, & a \notin M. \end{cases}$$

Множество M называется *эффективно перечислимым* (рекурсивно перечислимым), если оно является областью значений некоторой общерекурсивной функции, т.е. существует общерекурсивная функция $\psi_M(x)$, такая, что $a \in M$, если и только если для некоторого x.

$$a = \psi_M(x).$$

Функция ψ_M называется *перечисляющей* для множества M; соответственно алгоритм, вычисляющий ψ_M, называется *перечисляющим* или порождающим для M.

Поскольку множества являются базовым объектом математики и всякому утверждению можно придать вид утверждения о множествах, язык разрешимых и перечислимых множеств является универсальным языком для утверждений о существовании (или отсутствии) алгоритмов решения математических проблем.

Итак, *эффективно заданное* множество – это множество, обладающее разрешающей или перечисляющей функцией. Возникает вопрос о равносильности этих двух типов задания множеств.

Оказывается, что если непустое множество M разрешимо, то оно перечислимо, но не наоборот.

Таким образом, понятие общерекурсивной функции конкретизирует понятие алгоритма.

7.1 Рекурсивные функции

Рассмотрим теперь основные положения теории рекурсивных функций.

Пусть $f^n(x_1, \ldots, x_n)$, $n = 1, 2, \ldots,$ – частичные числовые функции, определённые на некотором подмножестве $M \subset N^n$ с натуральными значениями. Для любых $a_1, \ldots, a_n \in N$ и любых функций f^k и g^k пишем $f(a_{i_1}, \ldots, a_{i_k}) = g(a_{j_1}, \ldots, a_{j_s})$, если значения $f(a_{i_1}, \ldots, a_{i_k})$ и $g(a_{j_1}, \ldots, a_{j_s})$ не определены или эти значения определены и совпадают.

n-местная функция $f^n(x_1, \ldots, x_n)$ называется *всюду определённой*, если $Dom\ f^n = N^n$.

Назовём простейшими всюду определённые функции, приведенные в табл. 6.

Таблица 6

Простейшие функции – операторы

Наименование	Соотношение
Оператор сдвига	$S^1(x) = x + 1$
Оператор аннулирования	$O^1(x) = 0$
Оператор проектирования	$I_m^n(x_1, \ldots, x_n) = x_m,\ 1 \le m \le n$
Оператор суперпозиции	$S_m^n \equiv h^n(x_1, \ldots, x_n) = g^m(f_1^n(x_1, \ldots, x_n), \ldots, f_m^n(x_1, \ldots, x_n))$

Оператор примитивной рекурсии	$f^{n+1}(x_1,\dots,x_n,0)=g^n(x_1,\dots,x_n)$, $f^{n+1}(x_1,\dots,x_n,y+1)=h^{n+2}(x_1,\dots,x_n,y,\ f^{n+1}(x_1,\dots,x_n,y))$
Оператор рекурсии при $n=0$	$f(0)=a$ $f(y+1)=g(y,\ f(y))$
Оператор минимизации (μ – оператор)	$f^n(x_1,\dots,x_n)=\mu\, y[g^{n+1}(x_1,\dots,x_n,y)=0]$, $f^n(x_1,\dots,x_n)=y\leftrightarrow g(x_1,\dots,x_n,0)\neq 0,\dots,$ $g(x_1,\dots,x_n,y-1)\neq 0,\ g(x_1,\dots,x_n,y)=0$

Функция $f^n(x_1,\dots,x_{\text{n}})$ называется *примитивно рекурсивной*, если она может быть получена из простейших функций с помощью конечного числа применений операторов суперпозиции и примитивной рекурсии.

Функция $f^n(x_1,\dots,x_{\text{n}})$ называется *частично рекурсивной* (частично вычислимой), если она может быть получена из простейших функций с помощью конечного числа применений операторов суперпозиции, примитивной рекурсии и минимизации.

Функция $f^n(x_1,\dots,x_{\text{n}})$ называется *общерекурсивной* (вычислимой), если она частично рекурсивна и всюду определена.

Пример 7.1

Пусть функция $f(x,y)$ задана равенствами:

$$f(x,0)=x,$$

$$f(x,y+1)=f(x,y)+1.$$

Покажем, что эта рекурсия соответствует операции сложения, т.е. $f(x,y)=x+y$.

Действительно, из второго равенства рекурсии получаем

$$f(x,y+p)=f(x,y)+p.$$

Полагая $y=0$, получим

$$f(x,p)=f(x,0)+p.$$

Или, учитывая первое равенство рекурсии:

$$f(x,p)=x+p.$$

Пример 7.2.

Пусть функция $f(x,y)$ задана равенствами:

$$f(x,0)=0,$$

$$f(x,y+1)=x+f(x,y).$$

Тогда данной рекурсии соответствует операция умножения:

$$f(x,y)=xy.$$

Действительно, из второго равенства рекурсии имеем

$$f(x,y+p)=px+f(x,y).$$

Полагая $y=0$ и $f(x,0)=0$, получим

$$f(x,\, p)=px+f(x,0)=px\,.$$

Пример 7.3.

Доказать примитивную рекурсивность «арифметизированных» логических функций, т.е. числовых функций, которые на множестве {0, 1} ведут себя как логические функции.

Действительно, если $x,\, y \in \{0,1\}$, то

$$\bar{x}=1 \dot{-} x=\begin{cases} x-y, & x>y; \\ 0, & \text{иначе,} \end{cases}$$

$$x \vee y=\max(x,\, y),$$

$$x \wedge y=\min(x,\, y).$$

Далее можно доказать рекурсию для $f(x)=x \dot{-} 1$:

$$f(x,0)=0,$$

$$f(x,\, y+1)=f(x,y) \dot{-} 1.$$

Тогда

$$\min(x,y)=x \dot{-} (x \dot{-} y)=x\cdot sg(y \dot{-} x)+y\cdot \overline{sg}(y \dot{-} x),$$

$$\max(x,y)=y+(x \dot{-} y)=x\cdot sg(x \dot{-} y)+y\cdot \overline{sg}(x \dot{-} y),$$

где функция сигнум

$$sg(x)=\begin{cases} 0, & x=0, \\ 1, & x\neq 0 \end{cases}$$

определяется рекурсией

$$sg(0)=0,$$

$$sg(x+1)=1,$$

а $\overline{sg}$ – отрицание функции sg

$$\overline{sg}(0)=1,$$

$$\overline{sg}(x+1)=0.$$

Из функциональной полноты булева базиса $\bar{\ }, \vee, \wedge$ и того, что суперпозиция является примитивно рекурсивным оператором, следует примитивная рекурсивность всех логических функций.

7.2 Принцип работы машины Тьюринга

Для формализации понятия алгоритма английский математик Алан Тьюринг предложил модель вычислительной машины, названной его именем. Машина Тьюринга – абстрактное вычислительное устройство, состоящее из ленты, считывающей головки и управляющего устройства.

Рассмотрим отдельные элементы машины Тьюринга.

1 *Лента* разделена на одинаковые ячейки и не ограничена в обе стороны. В каждую ячейку может быть записан только один символ из *внешнего алфавита* – конечного множества символов $\{a_0, a_1, \ldots, a_n\}$, которыми кодируется вся информация, вводимая в машину. Символ a_0 называется пустым символом.

Внутренний алфавит состоит из символов $q_0, q_1, \ldots, q_m$ (состояния машины) и символов сдвига R (вправо), L (влево), S (на месте). Как правило, q_1 – это начальное состояние машины Тьюринга, а q_0 – это конечное состояние, после перехода в которое машина завершает работу.

2 *Считывающая головка* передвигается вдоль ленты и в каждый момент времени обозревает одну ячейку ленты. Головка считывает содержимое ячейки и записывает в неё символ из внешнего алфавита.

3 *Управляющее устройство* в каждый момент времени находится в некотором состоянии q_i. В начале работы машины Тьюринга на ленте записана некоторая последовательность символов. В каждом такте машина работает по схеме $q_s a_i \to a_j A q_t$. Управляющее устройство находится в состоянии q_s и видит в ячейке символ a_i.Вместо символа a_i пишется символ a_j. A – один из символов R, L, S. В зависимости от того, какой символ написан на месте символа A, происходит сдвиг считывающей головки на одну ячейку вправо или влево, либо считывающая головка остаётся на месте. При этом машина Тьюринга переходит в состояние q_t. Работа на следующем такте происходит аналогично.

Если после конечного числа тактов машина переходит в конечное состояние, то говорят, что машина Тьюринга применима к начальному слову. В случае, когда работа машины Тьюринга продолжается бесконечно (машина зацикливается), говорят, что машина Тьюринга не применима к начальному слову.

Описание одного такта работы машины назовём командой. Список всех команд называется программой машины Тьюринга и обозначается через П. Программу удобно представлять в виде таблицы.

Пример 7.4. Рассмотрим машину Тьюринга T, которая имеет следующие состояния: q_1 (начальное), q_2, q_3 и q_0 (конечное). Пусть T имеет внешний алфавит {0,1} и работает согласно программе П:

П	q_1	q_2	q_3
0	$q_2$0L	$q_1$0R	$q_0$1L
1	$q_2$1R	$q_3$1R	$q_1$0R

Выяснить, применима ли машина T к словам $P_1 = 11$ и $P_2 = 111$.

Решение.

Рассмотрим ленту с записанным на ней словом P_1. Так как первоначально T находится в состоянии q_1 и обозревает ячейку с символом 1, то выполнение первой команды $q_1 1 \to q_2 1\text{R}$ приведёт к записи в текущую ячейку символа «1», переходу T в состояние q_2 и сдвигу головки на одну ячейку вправо. После этого будет выполнена очередная команда $q_1 1 \to q_3$, затем команда $q_3 0 \to q_0 1\text{L}$. Запишем формируемую последовательность конфигураций машины Тьюринга:

$$q_1 11,\ 1q_2 1,\ 11q_3 0,\ 1q_0 11.$$

Поскольку состояние q_0 является конечным, то машина Т по достижении конфигурации $1q_0 11$ закончит работу. Значит, машина Тьюринга Т к слову $P_1 = 11$ применима.

Теперь рассмотрим ленту с записанным на ней словом P_2. Последовательность конфигураций в этом случае будет иметь вид:

$$q_1 111,\ 1q_2 11,\ 11q_3 1,\ 110q_1 0,\ 11\ q_2 00,\ 110q_1 0,\ 11\ q_2 00,\ \ldots$$

Ясно, что две конфигурации, а именно $110q_1 0$ и $11\ q_2 00$, переходят одна в другую, и за конечное число тактов машина не закончит свою работу. Значит, машина Тьюринга Т к слову $P_2 = 111$ не применима.

Тезис Чёрча – Тьюринга: всякий алгоритм может быть реализован с помощью некоторой машины Тьюринга. Как показывает опыт, любые вычислительные действия, которые может выполнить человек, могут быть представлены последовательностью действий некоторой машины Тьюринга. Тезис Чёрча – Тьюринга нельзя доказать, так как он устанавливает эквивалентность между строго определённым классом задач, решаемых машиной Тьюринга, и неформальным понятием алгоритма. Но он может быть опровергнут, если будет найден алгоритм, не реализуемый на машине Тьюринга.

8 МОДАЛЬНЫЕ И ТЕМПОРАЛЬНЫЕ ЛОГИКИ

8.1 Ограничения классической логики

В классической логике высказывания являются статичными, т.е. они не привязаны ко времени. Поэтому эта логика не адекватна для высказываний о последовательности событий во времени. Например, для таких высказываний наблюдается некоммутативность конъюнкции ($A \& B \neq B \& A$):

«Коля умер, и его похоронили» ≠ «Колю похоронили, и он умер».

Атомарные утверждения в общем случае истинны в один момент времени и ложны в другой. Это типично для динамических систем, таких, как программы или схемы дискретных устройств, состояния которых изменяется во времени.

Из таких атомарных утверждений строятся сложные утверждения, которые характеризуют динамические свойства процесса, развивающегося во времени.

Модальная логика строится на основе логики высказываний за счет добавления новых знаков, позволяющих выражать отношение тех или иных высказываний к окружающей действительности. Назначение различных систем модальной логики состоит в том, чтобы включить в логику так называемые модальности – прежде всего *необходимости* и *возможности*: того, что «должно быть», и того, что «может быть».

Классическая логика имеет дело с высказываниями, которые утверждают наличие или отсутствие той или иной ситуации. Однако в жизни приходится иметь дело с высказываниями, содержащими указание на необходимость или возможность чего-либо. Это связано с элементами случайного в природе, либо с констатацией того, что может произойти в будущем или имело место в прошлом, но чего нет в данный момент. Высказываниями этого рода называют *модальными.*

В модальной логике высказывание логически необходимо, если его истинность может быть установлена независимо от опыта, или чисто логическим путем. В модальной логике из необходимости высказывания вытекает его истинность, но не наоборот. Высказывание и его отрицание не могут быть вместе необходимыми.

Необходимость является более сильным видом истины, чем фактическая истинность. Существует различие между истинными высказываниями, являющимися таковыми, так сказать в силу необходимости, и высказываниями истинными случайными, возможными.

Существует несколько видов модальностей.

Алетические модальности. Эти высказывания содержат такие виды модальностей, как «необходимо», «возможно», «невозможно», «случайно».

Деонтические модальности связаны с характеристиками действий и поступками людей в обществе, например, «обязательно», «разрешено», «запрещено», «безразлично».

Эпистемические модальности характеризуют наши знания. Эти высказывания содержат такие модальности: «доказано», «опровергнуто», «не доказано», «не опровергнуто», «верит», «убежден», «сомневается».

Зарождение модальной логики произошло в античный период, а первоначальное развитие относится к средневековью. Модальности были введены Аристотелем, который считал, термин «возможность» имеет различный смысл. Аристотель ввел и исследовал модальные силлогизмы.

Дальнейшее развитие модальной логики связано с именами К. Льюиса (начало 20 века) и С. Крипке (середина 20 века)

8.2 Модальные логики

Модальная логика

Модальная (от лат. – способ, мера) логика, в которой кроме стандартных логических связок, переменных и предикатов есть *модальности* (модальные операторы). Модальности бывают разные; наиболее распространены временны́е («когда-то в будущем», «всегда в прошлом», «всегда» и т. д.) и пространственные («здесь», «где-то», «близко» и т. д.). Например, модальная логика способна оперировать утверждениями типа «Москва всегда была столицей России» или «Санкт-Петербург, когда-то в прошлом, был столицей России», которые невозможно или крайне сложно выразить в немодальном языке. Кроме временных и пространственных модальностей, есть и другие, например «известно, что» (логика знания) или «можно доказать, что» (логика доказуемости).

Обычно для обозначения модального оператора используется \Box и двойственный к нему \diamondsuit:

\diamondsuit A = \neg \Box \neg A

Это отражает то, что сказать «Москва когда-то была столицей России» то же самое, что сказать «не верно, что Москва никогда не была столицей России».

Типы модальностей

Алетические (от древнегр. alethinos – истинный) модальные понятия:

Логические

- L – необходимо;
- M – возможно;
- C – случайно.

Фактические

- \Box – необходимо;
- \diamond – возможно;
- \triangle – случайно.

Деонтические (древнегр. deon, deontos – должное, необходимое) модальные понятия:

- обязательно;
- разрешено;
- запрещено.

Аксиологические (древнегр. axios – ценность) модальные понятия:

- хорошо;
- нейтрально;
- плохо.

Эпистемические (древнегр. episteme – знание) модальные понятия:

- знание;
- полагание;
- незнание.

Временные:

- прошлое;
- настоящее;
- будущее.

Пространственные:

- там;
- здесь;
- нигде.

Семантика

В математической логике и информатике наиболее распространённой является семантика *Крипке*, также существуют алгебраическая семантика, топологическая семантика и ряд других.

Синтаксис

Модальная формула определяется рекурсивно как слово в алфавите, состоящем из счетного множества пропозициональных переменных PL, классических связок $\to$, $\bot$,, скобок (,) и модального оператора $\Box$.

А именно, формулой является

1. p для любого $p \in PL$;
2. $\bot$;
3. (A to B), если A и B – формулы;
4. $(\Box A)$, если A – формула.

Нормальной модальной логикой называется множество модальных формул, содержащее все классические тавтологии, аксиому нормальности

$$\Box(p \to q) \to (\Box p \to \Box q)$$

и замкнутое относительно правил *Modus ponens* $\frac{A, A\to B}{B}$, подстановки $\frac{A(p)}{A(B)}$и введение модальности $\frac{A}{\Box A}$.
Минимальная нормальная модальная логика обозначается K.

8.3 Темпоральные логики

Темпоральная логика (TL) – это любая логическая система, которая позволяет формализовать утверждения, истинность которых изменяется со временем, не вводя явно понятие времени.

Применения TL (используются разные TL):

ФИЛОСОФИЯ: формализм для прояснения философских вопросов о времени;

ЕСТЕСТВЕННЫЙ ЯЗЫК: формализм для определения семантики утверждений в естественных языках, включающих время, различных времен языка;

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: язык для представления знаний, связанных со временем;

ВЫЧИСЛИТЕЛЬНАЯ НАУКА: язык для выражения утверждений о временных свойствах развивающихся процессов, в т.ч. *вычислений* – выполнения программ

ТЕХНИКА: для формализации утверждений о свойствах *поведения* технических систем и свойствах процессов.

Примеры темпоральных модальных операторов:

Fp – «когда-нибудь в будущем p выполнится обязательно»;

Gp – «всегда в будущем будет выполняться p»;

Pq – «q было истинным когда-то в прошлом»;

Hq – «q было истинно всегда в прошлом».

Соотношения между модальностями:

Fp $\equiv \bar{G}\bar{p}$ – «p когда-то будет неверно, что p никогда не будет»;

Пример сложного утверждения:

Gp $\Rightarrow$Fp – «если нечто будет всегда, то оно когда-нибудь случится».

Другие темпоральные операторы: X (neXttime) и U (Until).

Рассмотрим утверждение: «Лифт никогда не пройдет мимо этажа, от которого поступил еще не удовлетворенный запрос».

Требуемая последовательность событий: в момент времени t происходит запрос, в момент времени t1 запрос обслужен, в момент времени t2 лифт проходит мимо.

Средствами логики предикатов ситуация описывается следующим образом:

($\forall$t$\geq$0)($\forall$t2 >t) [*Запрос* (t) & (t1: t$\leq$ t1< t2) $\neg$*Обслужен* (t1) $\Rightarrow$Мимо(t2).

С помощью операторов темпоральной логики та же ситуация описывается гораздо короче:

G [*Запрос* ($\neg$Мимо U *Обслужен*)].

Семантика Крипке – это наиболее распространённый общий подход к описанию значения модальных формул.

Пусть P – множество пропозициональных переменных

Тогда модель Крипке – это система (W ,R , ξ), где

I W – множество состояний (возможных миров)

I R ⊆ W ×W – отношение достижимости миров

I ξ : W → 2P – оценка переменных

Шкала Крипке (Kripke frame), на которой основывается модель (W ,R , ξ), – это пара (W ,R)

Если (*w* ,*w* ′) ∈ R , то *w* ′ – альтернативный мир для *w* (*w* -альтернатива)

Модель Крипке – это интерпретация модальной логики.

Время может истолковываться по-разному, и в зависимости от истолкования (то есть точного вида рассматриваемых шкал) могут получаться разные темпоральные логики, например:

Логика линейного времени (LTL, Linear Temporal Logic):

- время дискретно линейно течёт вперёд;
- формула – это свойство линейного развития событий.

Логика деревьев вычислений (CTL, Computation Tree Logic):

- время – это частично упорядоченное множество, которым описываются все альтернативы развития событий;
- формула – это высказывание о возможности и невозможности заданного развития событий с учётом всех альтернатив.

9 ФУНКЦИОНАЛЬНЫЕ СХЕМЫ

Методы булевой алгебры используются при создании и анализе функциональных схем, позволяющих понять структуру и логику работы цифровых устройств. Функциональные схемы состоят из электронных устройств (функциональных элементов) с конечным числом входов и выходов, причем на каждом входе и выходе могут появляться только два значения сигнала.

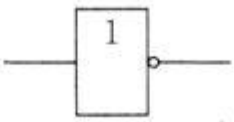
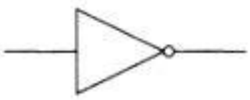
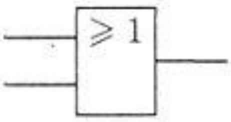
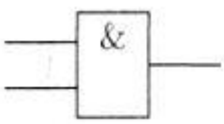
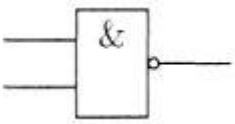
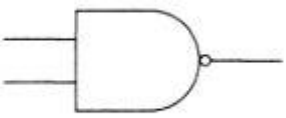
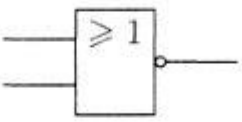
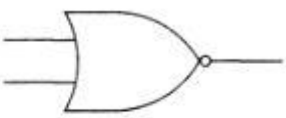
Сигналы 0 и 1 задаются разными уровнями напряжения. Сигнал логического нуля обычно представляется низким уровнем напряжения, логической единицы – высоким. Такое соглашение называется положительной логикой. В отрицательной логике принято обратное отображение напряжений на логические величины.

Изображают логические схемы с помощью условных графических обозначений, указывающих на выполняемую данным элементом функцию и не затрагивающих особенностей его физической реализации, которая может быть относительно сложной. В настоящее время существует несколько общепринятых стандартов условных обозначений. Наиболее распространённым является международный стандарт ANSI/IEEE Std 91-1984 «IEEE Standard Graphic Symbols for Logic Functions». В отечественной литературе условные обозначения элементов соответствуют ГОСТ 2.743-91 «Обозначения условные графические в схемах. Элементы цифровой техники».

Обозначения основных функциональных элементов в соответствии со стандартами IEEE и ГОСТ приведены в таблице 7

Таблица 7

Обозначения основных функциональных элементов

Название	ГОСТ 2.743.91	ANSI /IEEE Std 91-1984	Булева функция
НЕ			отрицание
ИЛИ			дизъюнкция
И			конъюнкция
И-НЕ			штрих Шеффера
ИЛИ=НЕ			стрелка Пирса

С помощью рассмотренных функциональных схем можно реализовать любую булеву функцию.

Пример 9.1. Записать булево выражение, соответствующее функциональной схеме на рис. 9.1.

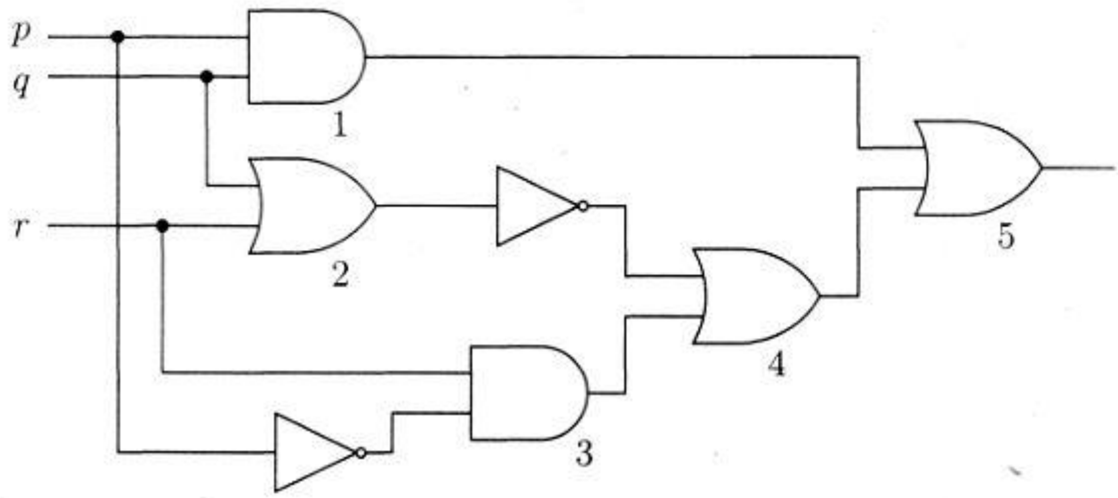


Рис. 9.1 Функциональная схема S_1 с тремя входами

Для этого выпишем в таблицу 8 входы и выходы для каждого из бинарных функциональных элементов, образующих схему.

Таблица 8

Значения на входах и выходах функциональной схемы S_1

Функциональный элемент	Входы	Выходы
1	p, q	$p \wedge q$
2	q, r	$q \vee r$
3	$r, \bar{p}$	$r \wedge \bar{p}$
4	$(\overline{q \vee r})$, $\mathrm{r} \wedge \bar{p}$	$(\overline{q \vee r}) \vee (\mathrm{r} \wedge \bar{p})$
5	$p \wedge q$, $(\overline{q \vee r}) \vee (\mathrm{r} \wedge \bar{p})$	$(p \wedge q) \vee (\bar{q} \wedge \bar{r}) \vee (\mathrm{r} \wedge \bar{p})$

Получаем булево выражение $(p \wedge q) \vee (\overline{q \vee r}) \vee (\mathrm{r} \wedge \bar{p})$, которое после применения закона де Моргана можно переписать в эквивалентном виде:

$$(p \wedge q) \vee (\bar{q} \wedge \bar{r}) \vee (\mathrm{r} \wedge \bar{p}).$$

Пример 9.2. Пусть задана функциональная схема S_2, представленная на рис. 9.2. С помощью карты Карно построить эквивалентную схему, состоящую из меньшего количества функциональных элементов по сравнению с исходной.

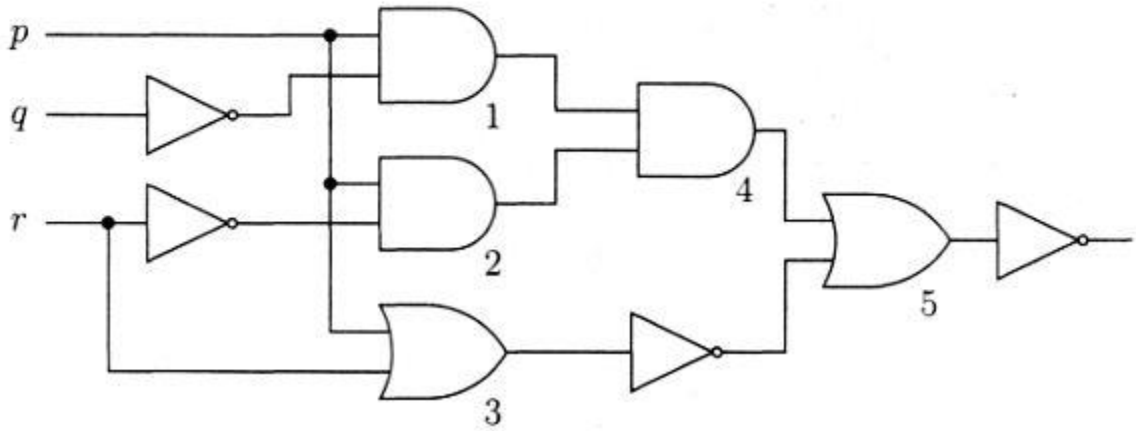


Рис. 9.2. Функциональная схема S_2

Выпишем в таблицу 9 значения сигналов на входе и выходе для каждого из бинарных функциональных элементов, образующих схему.

Таблица 9

Значения на входах и выходах функциональной схемы S_2

Функциональный элемент	Входы	Выходы
1	$p, \bar{q}$	$p \wedge \bar{q}$
2	$p, \bar{r}$	$p \wedge \bar{r}$
3	p, r	$p \vee r$
4	$p \wedge \bar{q}, p \wedge \bar{r}$	$(p \wedge \bar{q}) \wedge (p \wedge \bar{r}) = p \wedge \bar{q} \wedge \bar{r}$
5	$p \wedge \bar{q} \wedge \bar{r}, (\overline{p \vee r})$	$(p \wedge q) \vee (\bar{q} \wedge \bar{r}) \vee (\mathrm{r} \wedge \bar{p})$

Функция $f(p, q, r)$, генерируемая данной схемой, имеет вид:

$$f(p, q, r) = \overline{((p \wedge \bar{q} \wedge \bar{r}) \vee (\overline{p \vee r}))}.$$

С учётом законов де Моргана и свойств отрицания, получим:

$$f(p, q, r) = \neg(p \wedge \bar{q} \wedge \bar{r}) \wedge (p \vee r)$$

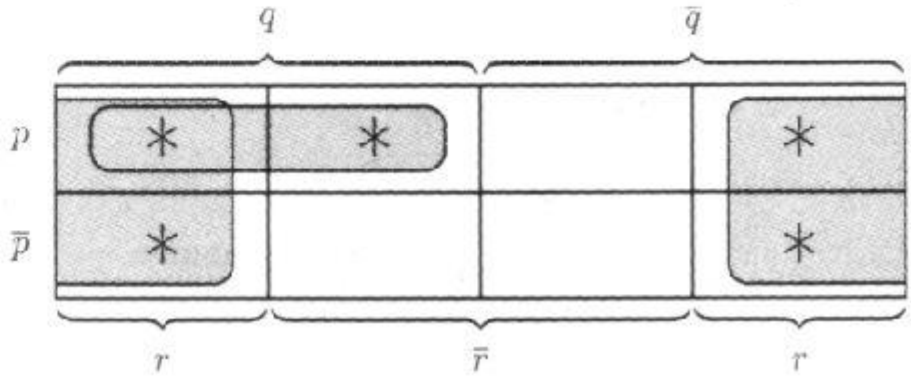
Вычислим вектор значений функции $f(p, q, r)$: α_f = (0101 0111). Далее, найдём совершенную дизъюнктивную нормальную форму f. Функция f принимает значение, равное 1, на следующих наборах аргументов (p, q, r):

(0,0,1), (0,1,1), (1,0,1), (1,1,0), (1, 1, 1).

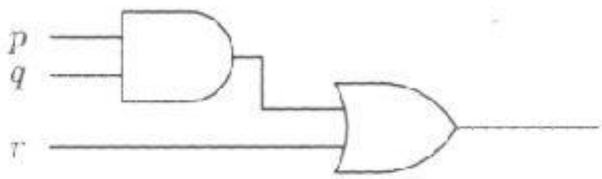
Объединяя соответствующие элементарные конъюнкции, получим СДНФ функции f.

$D_f = (\bar{p} \wedge \bar{q} \wedge \mathrm{r}) \vee (\bar{p} \wedge q \wedge \mathrm{r}) \vee (\mathrm{p} \wedge \bar{q} \wedge \mathrm{r}) \vee (\mathrm{p} \wedge q \wedge \bar{r}) \vee (\mathrm{p} \wedge q \wedge \mathrm{r})$.

Карту Карно формируем на основе полученной D_f.



Помеченные ячейки образуют два блока, и для $f(p, q, r)$ получаем следующее выражение: $f(p, q, r) = (p \wedge q) \vee r$. Следовательно, функциональная схема вида



эквивалентна S_2 и содержит меньшее количество функциональных элементов по сравнению с ней.

10 ЗАДАЧИ И УПРАЖНЕНИЯ

1 Установите, какие из следующих предложений являются высказываниями:

1) Язык Си относится к языкам программирования высокого уровня.
2) Летайте самолётами Аэрофлота!
3) Роботы – разумные существа.
4) Обитаема ли Луна?

2 Покажите, что высказывание $A \to (B \to C)$ логически эквивалентно высказыванию $(A \wedge B) \to C$.

3 Покажите, что высказывание $A \to (B \to C)$ логически эквивалентно высказыванию $\bar{C} \to (\bar{A} \vee \bar{B})$.

4 Покажите, что высказывание $A \vee (B \vee C)$ логически эквивалентно высказыванию $(A \vee B) \wedge (A \vee C)$.

5 Покажите, что высказывание $A \wedge (B \vee C)$ логически эквивалентно высказыванию $(A \wedge B) \vee (A \wedge C)$.

6 Докажите, что высказывание $((A \vee B) \wedge (\overline{A \wedge B})) \leftrightarrow ((A \wedge \bar{B}) \vee (\bar{A} \wedge B)$ является тавтологией.

7 Являются ли высказывания

1) $(A\to (B \to C)) \to ((A \wedge B) \to C)$,
2) $(A \wedge (B \vee C)) \to (A \vee (B \wedge C)$,
3) $(A \wedge (B \to C)) \wedge (B \vee (B \to C))$,
4) $(A \wedge (B \to C)) \to (\bar{C} \to A)$

тавтологиями?

8 Обозначим через X высказывание: « сейчас зима », через Y – «новый год наступил », а через Z – «надо покупать ёлку ». Запишите следующие высказывания как составные высказывания, включающие X, Y и Z.

1) Если сейчас зима и новый год не наступил, то надо покупать ёлку.
2) Если сейчас не зима, то не надо покупать ёлку.
3) Если сейчас зима и новый год наступил, то не надо покупать ёлку.

9 Обозначим через X высказывание: « пешеход законопослушен », через Y – «пешеход переходит дорогу », а через Z – «горит запрещающий знак светофора». Запишите следующие высказывания как составные высказывания, включающие X, Y и Z.

1) Если горит запрещающий знак светофора и пешеход переходит дорогу, то пешеход не законопослушен.
2) Если пешеход законопослушен и пешеход переходит дорогу, то не горит запрещающий знак светофора.
3) Если не горит запрещающий знак светофора, то пешеход переходит дорогу.

10 Сформулируйте высказывания, контрапозитивные (т.е. противоположные) данным:
1)«Если студент сдал сессию на «отлично», то он получает повышенную стипендию».
2) «Если студент сдал сессию на «хорошо и отлично» и сделал доклад на научной конференции, то он получает повышенную стипендию».

11 Установите, какие из следующих высказываний являются тождественно ложными:

1) $A \to (A \vee B)$;

2) $(A \to B) \to (\bar{B} \to \bar{A})$;

3) $(A \wedge B) \wedge (\bar{A} \vee \bar{B})$;

4) $A \wedge ((A \to \bar{A}) \wedge (\bar{A} \to A)$.

12 Построить таблицы истинности для функций:

1) $f_1 = \bar{x}_1 \vee \bar{x}_2$;

2) $f_2 = (x \vee y) \to (x \vee y) \to (x \wedge \bar{y} \vee \bar{x} \to \bar{y})$;

3) $f_3 = (x_1 \wedge x_2) \vee x_3$;

4) $f_4 = (x \wedge \bar{y}) \to (y \vee \bar{x} \to \bar{z})$;

5) $f_5 = (x_1 \to \bar{x}_2) \to (\overline{x_1 \vee x_2} \wedge \bar{x}_3)$;

6) $f_6 = (\bar{x} \vee z) \wedge (y \to (u \to x))$;

7) $f_7 = x_1 \to (x_2 \to (\ldots \to x_n)\ldots)$;

8) $f_8 = x_1 \vee x_2 \vee \ldots \vee x_n \to y_1 \wedge y_2 \wedge \ldots \wedge y_n$.

13 Какие из функций принадлежат классу Поста T_0:

1) 0; 2) 1; 3) $x_1 \vee x_2$; 4) $x_1 \wedge x_2$; 5) $x_1 \to x_2$?

14 Какие из функций принадлежат классу Поста T_1:

1) 0; 2) 1; 3) $x_1 \vee x_2$; 4) $x_1 \wedge x_2$; 5) $x_1 \to x_2$?

15 Выяснить, принадлежат ли следующие функции классу Поста L:
1) 1; 2) 0; 3) $\bar{x}$; 4) $x_1 \wedge x_2$; 5) $x_1 \vee x_2$; 6) $x_1 \to x_2$; 7) $x_1 \oplus x_2$?

16 Выяснить, принадлежат ли следующие функции классу Поста M:
1) 1; 2) 0; 3) $\bar{x}$; 4) $x_1 \wedge x_2$; 5) $x_1 \vee x_2$; 6) $x_1 \to x_2$; 7) $x_1 \oplus x_2$?

17 Выяснить, принадлежат ли следующие функции классу Поста S:
1) 1; 2) 0; 3) $\bar{x}$; 4) $x_1 \wedge x_2$; 5) $x_1 \vee x_2$; 6) $x_1 \to x_2$; 7) $x_1 \oplus x_2$?

18 Выразите основные логические операции (отрицание, дизъюнкцию, конъюнкцию, импликацию) через:

1) штрих Шеффера;
2) стрелку Пирса.

19 Выразите:

1) операцию «штрих Шеффера» через операцию «стрелка Пирса»;
2) операцию «стрелка Пирса» через операцию «штрих Шеффера»;

20 Технологический процесс предусматривает следующую схему работы четырёх станков S_1 – S_4. Если работает первый станок, то работает второй. Второй станок работает тогда и только тогда, когда работает третий, и если работает четвёртый, то третий остановлен. Определите, какие станки работают в данный момент, если известно, что сейчас работает либо первый, либо второй (но не одновременно).

21 В ограблении магазина подозреваются трое: Антонов, Богданов и Вернер. Из предварительного расследования следует:

1) если к преступлению причастен Вернер, то причастен и Богданов;
2) если виновен Антонов, то виновен и Богданов.

Первый вывод подтвердился, а второй оказался ложным. Кто совершил ограбление?

22 Для выполнения важного поручения президента группа специалистов отправляется в Китай. Известно, что поедет либо Анри, либо Виктор (причем только один из них). Если Франсуа не едет, то Виктор тоже не едет. Кроме того, если поедет Анри, то поедет Франсуа. Кто отправится выполнять поручение президента?

23 В летнем лагере отдыхают четверо студентов: Анна, Борис, Вера и Григорий. Одни отдыхающие отправились на экскурсию, другие – на пляж. Либо Борис, либо Вера (а возможно оба) отправились на экскурсию. Если Вера на пляже и Григорий на пляже, то Анна тоже отдыхает на пляже. Неверно, что если Анна на пляже, то Вера или Григорий на экскурсии. Кто отправился на экскурсию, а кто на пляж?

24 Какие из следующих предложений являются предикатами:

1) целое число i не меньше 10;
2) x – простое число;
3) произвольное число d можно представить в виде суммы двух нечётных чисел;
4) k – натуральное число?

25 Запишите отрицание предиката P, если

1) $P(n)$ = {натуральное число n – нечётное};
2) $P(x, y) = \{x > y\}$.

26 Пусть x и y – вещественные числа. Определите истинностное значение

высказываний
1) $\forall x\ (\exists y\ (x > y))$;
2) $\forall x\ (x \neq 0 \Rightarrow \exists y\ (xy = 1))$.

27 Пусть x и y – вещественные числа. Определите истинностное значение высказываний
1) $\exists x\ (\forall y\ (xy = y))$;
2) $\exists x\ (\forall y\ (x + y = 1))$.

28 Определить, являются ли формулами выражения:
1)$\forall x\ P(x)$;
2) $\exists x\ (Q(x) \to P(x,y))$;
3) $\forall x\ P(x) \wedge Q(x,y)$;
4) $\exists x\ P(x,y) \vee Q(x)$.

29 Докажите, что булевы выражения $x_1 \leftrightarrow x_2$ и $1 \oplus x_1 \oplus x_2$ определяют одну и ту же булеву функцию.

30 Докажите, что булевы выражения $x_1 \mid x_2$ и $1 \oplus (x_1 \wedge x_2)$определяют одну и ту же булеву функцию.

31 Установите, являются ли булевы функции f и g эквивалентными:
1) $f(x_1, x_2, x_3) = x_1 \to (x_2 \to x_3)$, $g(x_1, x_2, x_3) = (x_1 \wedge x_2) \to x_3)$;
2) $f(x_1, x_2) = (x_1 \to x_2\ ) \wedge (\overline{x_2} \to x_1)$, $g(x_1, x_2) = x_1$.

32 Используя законы булевой алгебры, покажите, что для переменных x_1, x_2, x_3 выполняется дистрибутивность импликации относительно
1) конъюнкции: $x_1 \to (x_2 \wedge x_3) = (x_1 \to x_2) \wedge (x_1 \to x_3)$;
2) дизъюнкции: $x_1 \to (x_2 \vee x_3) = (x_1 \to x_2) \vee (x_1 \to x_3)$;
3) импликации: $x_1 \to (x_2 \to x_3) = (x_1 \to x_2) \to (x_1 \to x_3)$;
4) эквиваленции: $x_1 \to (x_2 \leftrightarrow x_3) = (x_1 \to x_2) \leftrightarrow (x_1 \to x_3)$.

33 Постройте СДНФ и СКНФ функций, заданных векторами значений:
1) $\alpha_{f1} = (1011\ 0101)$;
2) $\alpha_{f2} = (0111\ 1000)$;
3) $\alpha_{f3} = (1010\ 0001)$;
4) $\alpha_{f4} = (0110\ 1001)$.

34 Постройте СДНФ и СКНФ для следующих функций:
1) $f(x_1, x_2, x_3) = (x_1 \oplus x_2) \vee \overline{x_3}$;
2) $f(x_1, x_2, x_3) = x_1 \to (\ x_2 \mid x_3)$;
3) $f(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee\ (x_2 \oplus x_3) \vee (x_1 \to x_3)$;
4) $f(x_1, x_2, x_3) = (x_1 \downarrow\ x_2) \leftrightarrow \overline{(\ x_2 \mid x_3}$).

35 Используя метод неопределённых коэффициентов, постройте полином Жегалкина по вектору значений функции:

1) α_{f1} = (1011 0110);
2) α_{f2} = (0110 1100);
3) α_{f3} = (1110 0001);
4) α_{f4} = (0010 1000).

36 Используя метод эквивалентных преобразований, постройте полином Жегалкина для следующих функций:
1) $f(x_1, x_2, x_3) = (x_1 \wedge x_2) \oplus (x_1 \downarrow x_3)$;
2) $g(x_1, x_2, x_3) = x_1 \to (x_2 \mid x_3)$;
3) $h(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee (x_2 \oplus x_3) \vee (x_1 \to x_3)$;
4) $s(x_1, x_2, x_3, x_4) = \overline{(x_1 \to x_2)} \to (x_3 \leftrightarrow x_4)$.

37 Выясните, является ли функция h двойственной к функции g:
1) $g(x_1, x_2) = (x_1 \vee x_2)$, $h(x_1, x_2) = (x_1 \wedge x_2)$;
2) $g(x_1, x_2) = (x_1 \wedge \bar{x}_2) \vee \overline{(x_1 \to x_2)}$, $h(x_1, x_2) = \bar{x}_1$.

38 Является ли функция, задаваемая вектором значений α, самодвойственной, если
1) α_{f1} = (0110);
2) α_{f2} = (1011 0010);
3) α_{f3} = (1000 0101 0101 1110);
4) α_{f4} = (0101 1100 0011 1010)?

39 Проверьте, являются ли следующие функции самодвойственными:
1) $f(x_1, x_2) = x_1 \oplus x_2$;
2) $f(x_1, x_2) = x_1 \leftrightarrow x_2$;
3) $f(x_1, x_2) = x_1 \mid x_2$;
4) $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$.

40 Проверьте, являются ли следующие функции монотонными:
1) $f(x_1, x_2) = x_1 \oplus x_2$;
2) $f(x_1, x_2) = x_1 \to x_2$;
3) $f(x_1, x_2) = (x_1 \downarrow x_2) \vee x_1$;
4) $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$.

41 Проверьте, являются ли следующие функции линейными:
1) $f(x_1, x_2) = x_1 \vee \bar{x}_2$;
2) $f(x_1, x_2) = x_1 \mid x_2$;
3) $f(x_1, x_2) = x_1 \to x_2$;
4) $f(x_1) = (\overline{x_1 \oplus 1})$.

42 Докажите, что система функций {¬, ∨, ∧} является полной в P_2, где P_2 – множество всех функций булевой алгебры.

43 Проверьте, является ли система функций {¬, ∧} является полной в P_2.

44 Проверьте, является ли система функций {¬, ∨} является полной в P_2.

45 Проверьте, является ли полной в P_2 каждая из систем функций:

1) $\{\downarrow\}$;
2) $\{\,|\,\}$;
3) $\{\oplus, \wedge, 1\}$.

46 Докажите, что каждая из систем функций является полной в P_2:

1) $\{0, \vee, \leftrightarrow\}$;
2) $\{0, \rightarrow\}$.

47 Проверьте, какие из следующих систем функций являются базисами в P_2:

1) $\{\oplus, \leftrightarrow, \wedge\}$;
2) $\{\oplus, \leftrightarrow, \vee\}$;
3) $\{\rightarrow, \oplus\}$;
4) $\{\leftrightarrow, \rightarrow, \neg\}$.

48 Упростить следующие выражения, применяя карты Карно:

1) $S_1 = \mathrm{pqr} \vee \mathrm{pq}\bar{r} \vee \mathrm{p}\bar{q}\bar{r} \vee \mathrm{p}\bar{q}\mathrm{r}$;
2) $S_2 = \mathrm{pqr} \vee \mathrm{pq}\bar{r} \vee \bar{p}\bar{q}\bar{r} \vee \bar{p}\bar{q}\mathrm{r}$;
3) $S_3 = \mathrm{pqr} \vee \bar{p}\mathrm{qr} \vee \mathrm{p}\bar{q}\mathrm{r} \vee \bar{p}\bar{q}\mathrm{r}$;
4) $S_4 = \mathrm{pqr} \vee \bar{p}\mathrm{qr} \vee \bar{p}\mathrm{q}\bar{r} \vee \mathrm{p}\bar{q}\mathrm{r} \vee \bar{p}\bar{q}\mathrm{r}$.

49 Построить карту Карно для булевой функции четырёх переменных $f(x_1,x_2,x_3,x_4)$ и упростить f:

$$f(x_1,x_2,x_3,x_4) = x_1x_2x_3x_4 \vee x_1x_2x_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2x_3\bar{x}_4 \vee x_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4.$$

50 Построить карту Карно для булевой функции четырёх переменных $g(x_1,x_2,x_3,x_4)$ и упростить g:

$$g(x_1,x_2,x_3,x_4) = x_1x_2x_3\bar{x}_4 \vee \bar{x}_1x_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1\bar{x}_2x_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3x_4.$$

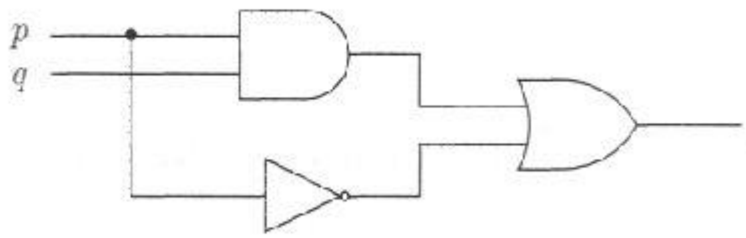
51 Построить карту Карно для булевой функции четырёх переменных $h(x_1,x_2,x_3,x_4)$ и упростить h:

$$h(x_1,x_2,x_3,x_4) = x_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4 \vee x_1x_2x_3\bar{x}_4 \vee \bar{x}_1x_2x_3\bar{x}_4 \vee x_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee$$
$$\vee \bar{x}_1x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3x_4.$$

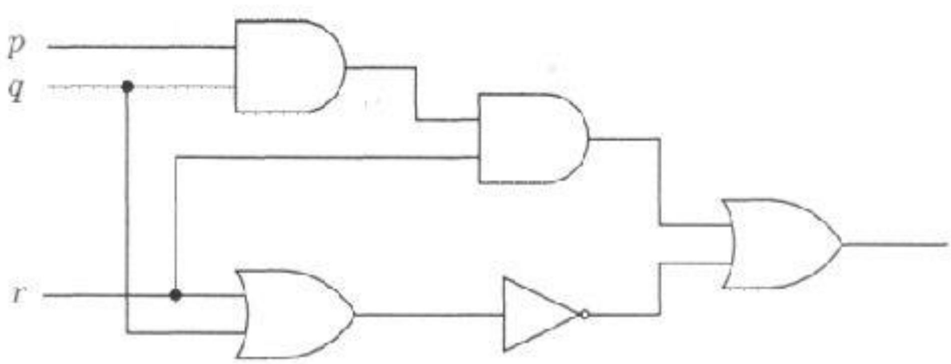
52 Вычислить вектор значений функции $f(x)$, построить карту Карно и упростить $f(x)$:

1) $f(x_1,x_2,x_3,x_4) = (x_1 \oplus x_3) \vee (\bar{x}_2 \wedge x_4)$;
2) $f(x_1,x_2,x_3,x_4) = (x_1 \wedge x_3) \vee (\bar{x}_2 \rightarrow x_4)$.

53 Запишите булеву функцию, генерируемую функциональной схемой



54 Запишите булеву функцию, генерируемую функциональной схемой



55 Постройте функциональную схему, реализующую импликацию $p \to q$, используя только функциональные элементы:

1) И-НЕ;
2) ИЛИ-НЕ.

56 Постройте функциональную схему, реализующую функцию эквивалентности $p \leftrightarrow q$, используя только функциональные элементы:

1) И-НЕ;
2) ИЛИ-НЕ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 **Акимов, О.Е.** Дискретная математика: логика, группы, графы / О.Е. Акимов. – М. : Лаборатория Базовых Знаний, 2001. – 376 с.

2 **Андерсон, Д.А.** Дискретная математика и комбинаторика / Д.А. Андерсон. – М. : Изд. дом «Вильямс», 2003. – 960 с.

3 **Белоусов, А.И.** Дискретная математика : учебник для вузов / А.И. Белоусов, С.Б. Ткачёв. – М. : Изд-во МГТУ им. Н. Баумана, 2001. – 744 с.

4 **Борзунов, С.В**. Задачи по дискретной математике / С.В. Борзунов, С.Д. Кургалин. – СПб.: БХВ-Петербург, 2016.– 528 с.

5 **Гаврилов, Г.П.** Задачи и упражнения по дискретной математике : учебное пособие / Г.П. Гаврилов, А.А. Сапоженко. – М. : ФИЗМАТЛИТ, 2004. – 416 с.

6 **Горбатов, В.А.** Фундаментальные основы дискретной математики. Информационная математика / В.А. Горбатов. – М. : Наука, 2000. – 544 с.

7 **Ерусалимский, Я.М.** Дискретная математика: теория, задачи, приложения / Я.М. Ерусалимский. – М. : Вузовская книга, 1998. – 280 с.

8 **Иванов, Б.Н.** Дискретная математика. Алгоритмы и программы : учебное пособие / Б.Н. Иванов. – М. : Лаборатория Базовых Знаний, 2001. – 288 с.

9 **Кузнецов, О.П.** Дискретная математика для инженера / О.П. Кузнецов, Г.М. Адельсон-Вельский. – М. : Энергоатомиздат, 1988. – 480 с.

10 **Нефедов, В.Н.** Курс дискретной математики : учеб. пособие / В.Н. Нефедов, В.А. Осипова. – М. : Изд-во МАИ, 1992. – 264 с.

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

12 **Самсонов, Б.Б.** Компьютерная математика (основание информатики) / Б.Б. Самсонов, Е.М. Плохов, А.И. Филоненков. – Ростов н/Д : Феникс, 2002. – 512 с.

13 **Узденов, А.М.** Основы компьютерной математики (конечные структуры алгебры, логики, дискретной математики) / А.М. Узденов, Б.Б. Самсонов, А.И. Филоненков. – Ростов н/Д : Изд-во «Булат», 2005. – 480 с.

14 **Фомичёв, В.М.** Дискретная математика и криптология: курс лекций / В.М. Фомичёв. – М. : ДИАЛОГ – МИФИ, 2003. – 400 с.

15 **Хаггарти, Р.** Дискретная математика для программистов / Р. Хаггарти. – М. : Техносфера, 2004. – 320 с.

16 **Яблонский, С.В.** Введение в дискретную математику : учеб. пособие для вузов / С.В. Яблонский. – М. : Высшая школа, 2001. – 384 с.

Учебное издание

Голубенко Евгений Владимирович

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ
ИНФОРМАЦИОННЫХ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ.
ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ**

Редактор Т.М. Чеснокова
Техническое редактирование и корректура Т.М. Чесноковой

Подписано в печать 27.12.19. Формат 60×84/16.
Бумага офсетная. Печать офсетная. Усл. печ. л. 4,08.
Тираж 500 экз. Изд. № 49. Заказ

Редакционно-издательский центр ФГБОУ ВО РГУПС.

Адрес университета: 344038, Ростов н/Д, пл. Ростовского Стрелкового Полка
Народного Ополчения, д. 2.