Р. В. Бузату

# МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Учебное пособие

$$\begin{cases} \alpha_1 \to \beta_1 \\ \alpha_2 \to \beta_2 \\ \quad \dots \\ \alpha_k \to \beta_k \end{cases}$$

МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет математики и информатики
Департамент математики

Р. В. Бузату

# Математическая логика и теория алгоритмов

Учебное пособие (курс лекций)

*Утверждено*
*Советом по качеству*
*Молдавского государственного университета*

Кишинёв 2021

510.5/.6(075.8)
Б 904

*Рекомендовано к изданию департаментом математики и советом факультета математики и информатики Молдавского государственного университета.*

Учебное пособие написано на основе курса лекций, читаемого автором в течение ряда лет студентам, обучающимся по специальности “Информационные технологии”. В нем рассматриваются понятия и вопросы, касающиеся основных разделов математической логики и теории алгоритмов: алгебра логики, логические функции, исчисление высказываний, логика предикатов, машина Тьюринга, нормальные алгоритмы Маркова. Изложение сопровождается большим количеством разобранных примеров и задач для самостоятельного решения.

Предназначено для студентов и магистров технических специальностей университетов, изучающих курсы математической логики и теории алгоритмов.

**Автор** *Бузату Р. В.*, кандидат математических наук (PhD), Молдавский государственный университет

**Рецензент** *Годоноагэ А. Ф.*, кандидат физико-математических наук, доцент, Академия экономических знаний Молдовы

**Описание CIP Национальной Книжной Палаты**

**Бузату, Р. В.**

Математическая логика и теория алгоритмов: Учебное пособие / Р. В. Бузату; Молдавский государственный университет, Факультет математики и информатики, Департамент математики. – Кишинёв: CEP USM, 2021. – 135 с.

Библиогр.: с. 134-135 (26 назв.). – 50 экз.

ISBN 978-9975-158-91-6.
510.5/.6(075.8)
Б 904

ISBN 978-9975-158-91-6

# Предисловие

Логика – это наука о законах мышления и способах построения доказательств. Логика берет своё начало в работах древнегреческого учёного Аристотеля (IV в. до нашей эры), который обстоятельно систематизировал логические формы и правила мышления.

Идеи применения в логике математической символики и построений логических исчислений были высказаны выдающимся учёным Лейбницем в XVII веке. Однако только в XIX веке математик Д. Буль реализовал эти идеи, применив алгебраическую символику для записи логических операций и логических выводов. Таким образом, Д. Буль заложил основы математической логики.

Современный этап развития математической логики характеризуется применением строгого аксиоматического метода, который предполагает выделения базовых утверждений математической теории (аксиом) и правил комбинирования утверждений (правил вывода). Существенный вклад в развитие современной математической логики внесли Д. Гильберт и К. Гёдель.

В первой половине XX века на базе математической логики была разработана теория алгоритмов как аппарат для исследования разрешимости математических проблем. В разработку этой теории внесли значительный вклад математики: А. Тьюринг, Э. Пост и А. А. Марков.

В настоящее время математическая логика представляет собой обширный и разветвлённый раздел математики, которой находит многочисленные приложения в кибернетике, вычислительной математике и лингвистике.

Данное учебное пособие “Математическая логика и теория алгоритмов” предназначено для студентов технических специальностей университетов, но также будет полезно всем тем, кто приступает к изучению основ математической логики и теории алгоритмов.

Оно состоит из пяти глав.

Первая глава посвящена алгебре высказываний. В ней изучаются основные понятия алгебры логики, обсуждаются понятия равносильности и двойственности, приводится ряд наиболее важных законов алгебры логики и вводятся нормальные формы.

Во второй главе рассматриваются функции алгебры логики. Излагаются различные представления булевых функций, важнейшим из которых является полином Жегалкина. Изучаются замкнутость и полнота. Описываются важнейшие замкнутые классы.

В третьей главе излагается исчисление высказываний. Доказываются теорема дедукции и производные правила вывода. Исследуются проблемы аксиоматического исчисления высказываний: разрешимости, непротиворечивости, полноты и независимости.

Четвёртая глава посвящена логике предикатов. В ней после введения базовых понятий обсуждаются понятия равносильности, логического следствия и приводится список основных законов. Вводятся предварённая и сколемовская нормальные формы. Приводятся теоремы распознавания общезначимости формул, и даются примеры применения логики предикатов для записи различных математических выражений. Также даны основы исчисления предикатов.

Последняя пятая глава посвящена теории алгоритмов. Здесь формализуется понятие алгоритма в виде частично рекурсивных функций. Также даются и другие варианты формализации понятия алгоритма: машина Тьюринга и нормальные алгоритмы Маркова. Для каждого из них приводятся разнообразные примеры решённых задач. В конце главы анализируются некоторые неразрешимые алгоритмические проблемы.

Каждая глава содержит значительное количество примеров и задачи для самостоятельного решения.

# Оглавление

# Глава 1
# Алгебра логики (алгебра высказываний)

## 1.1 Высказывания и логические операции

Основным понятием математической логики является понятие (простого) высказывания.

**Определение 1.1.1.** ***Высказывание*** *– это повествовательное предложение (утверждение), о содержании которого всегда можно сказать истинно оно или ложно. Истинность или ложность, приписываемые некоторому высказыванию, называются его* ***значением истинности*** *или* ***истинностным значением****.*

Приведём примеры высказываний.

1) Париж – столица Франции.
2) Щука не рыба.
3) Число 8 делится на 2 и на 3.
4) В прямоугольном треугольнике квадрат гипотенузы равен сумме квадратов катетов.
5) Если дважды два четыре, то снег чёрный.

Высказывания 1) и 4) истинны, а высказывания 2), 3) и 5) ложны. Очевидно, предложение “Сколько учится на программиста?” не является высказыванием. Высказывание, содержащее одно утверждение, называется *простым* или *элементарным*. А высказывание, полученное в результате комбинации нескольких простых высказываний, называется *сложным* или *составным.*

Из приведённых выше утверждений примерами элементарных высказываний могут служить высказывания 1), 2) и 4), а примерами сложных – высказывания 3) и 5).

В алгебре высказываний все высказывания рассматриваются только с точки зрения их логического значения. Считается, что каждое высказывание либо истинно, либо ложно, и не может быть истинным и ложным одновременно.

Алгебра высказываний изучает способы построения высказываний из уже имеющихся высказываний, а также закономерности способов сочетания высказываний. Алгебра высказываний является фундаментом математической логики.

В дальнейшем, элементарные высказывания будем обозначать большими буквами латинского алфавита ($A$, $B$, $C$, ...), а их истинные значение цифрами 1 (истина) и 0 (ложь). Если высказывание $A$ истинно, то будем писать $A = 1$, а если ложно, то $A = 0$.

Сложные высказывания могут быть построены из простых высказываний с помощью операций над высказываниями или логических операций (связок).

Вообще говоря, под *операцией* на множестве $M$ понимают любое отображение $\varphi: M^n \to M$, которое любому упорядоченному набору из $n$ элементов множества $M$ (кортежу) ставит в соответствие элемент того же множества. Натуральный показатель $n$ называют *арностью* этой операции. Ясно, что множество всех операций на заданном множестве $M$ бесконечно, даже если само множество $M$ конечно. На практике ограничиваются операциями небольшой арности: *унарными* ($n = 1$), *бинарными* ($n = 2$), *тернарными* ($n = 3$).

**Определение 1.1.2.** ***Отрицанием*** *высказывания $A$ называется высказывание $\neg A$ или $\bar{A}$ (читается: "не $A$" или "неверно, что $A$"), которое истинно тогда и только тогда, когда высказывание $A$ ложно.*

Высказывание $\bar{A}$ истинно, если высказывание $A$ ложно, и ложно, если $A$ истинно. Например, для высказывания "Щука не рыба" отрицанием будет высказывание "Неверно, что щука не рыба", которое истинно.

**Определение 1.1.3.** ***Конъюнкцией*** *высказываний $A$ и $B$ называется высказывание $A \wedge B$ или $A\&B$ (читается: "$A$ и $B$"), истинное тогда и только тогда, когда истинны оба высказывания $A$ и $B$.*

Например, для высказываний "10 делится на 2", "10 делится на 5" их конъюнкцией будет высказывание "10 делится на 2 и на 5", которое, очевидно, истинно.

Из определения операции конъюнкции видно, что союз "и" в алгебре логики употребляется в том же смысле, что и в повседневной речи. Но в обычной речи принято соединять союзом "и" два близких по смыслу высказывания, а в алгебре логики рассматривается конъюнкция двух любых высказываний.

**Определение 1.1.4.** ***Дизъюнкцией*** *высказываний $A$ и $B$ называется высказывание $A \vee B$ (читается: "$A$ или $B$"), ложное тогда и только тогда, когда ложны оба высказывания $A$ и $B$.*

Например, рассмотрим высказывание "Все люди смертны, или $5 > 8$". Несмотря на первоначально кажущуюся странность этого высказывания, нет сомнений в его истинности, так как высказывание "Все люди смертны" истинно.

Заметим, что в русском языке союз "или" понимается в двух смыслах: разделительном – или то, или другое, но не оба, и соединительном – или то, или другое, или оба. В алгебре логики союз "или" понимается в соединительном смысле.

**Определение 1.1.5.** ***Импликацией*** *высказываний $A$ и $B$ называется высказывание $A \to B$ (читается: "$A$ влечет за собой $B$", или "из $A$ следует $B$", или "если $A$, то $B$"), ложное тогда и только тогда, когда $A$ истинно, а $B$ ложно.*

В импликации $A \to B$ высказывание $A$ называется *посылкой*, а высказывание $B$ – *следствием* или *заключением*. Импликация призвана отразить процесс рассуждения, умозаключения, который характеризуется следующим образом: если мы исходим из истинной посылки и правильно (верно) рассуждаем, то в любом случае мы придём к истинному заключению (следствию, выводу). Другими словами, если мы исходили из истинной посылки и пришли к ложному выводу, значит рассуждение было построено неверно.

К примеру, высказывание "Если число 10 делится на 5, то и число $10^2$ делится на 5", очевидно, является истинным, так как здесь истинна посылка "Число 10 делится на 5" и истинно заключение "Число $10^2$ делится на 5".

В предложении вида "Если $A$, то $B$" в обыденной речи, всегда подразумеваем, что предложение $B$ вытекает из предложения $A$, а математическая логика не требует этого, поскольку в ней смысл содержания высказываний не рассматривается.

**Определение 1.1.6.** ***Эквивалентностью*** *высказываний $A$ и $B$ называется высказывание $A \sim B$ или $A \leftrightarrow B$ (читается: "$A$ эквивалентно $B$", или "$A$ необходимо и достаточно для $B$", или "$A$ тогда и только тогда, когда $B$"), истинное тогда и только тогда, когда оба высказывания $A$ и $B$ одновременно истинны или ложны.*

Например, высказывание "$\sin(30^o) = 1/2$ тогда и только тогда, когда $2 * 3 = 5$" является ложным, так как высказывание "$\sin(30^o) = 1/2$" истинно, а высказывание "$2 * 3 = 5$" ложно.

Отметим также, что эквивалентность и импликация играют большую роль в математических доказательствах, так как с их помощью формулируются значительное число теорем и лемм.

Определения 1.1.2 – 1.1.6 логических операций над высказываниями можно свести в одну таблицу (Таблица 1.1.1), которую принято называть *таблицей истинности*:

| $A$ | $B$ | $\bar{A}$ | $A\&B$ | $A \vee B$ | $A \to B$ | $A \leftrightarrow B$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Таблица 1.1.1

В алгебре логики помимо указанных операций используются и другие (производные) операции, которые получаются из основных операций. Опишем следующие три операции.

**Определение 1.1.7.** ***Неравнозначностью*** *двух высказываний $A$ и $B$ называется высказывание $A \oplus B$ (читается: "либо $A$, либо $B$"), истинное, когда истинностные значения $A$ и $B$ не совпадают, и ложное – в противном случае.*

Таким образом, по определению неравнозначность является отрицанием эквивалентности. Неравнозначность также называется *сложением по модулю 2* или *исключающим или*.

**Определение 1.1.8.** ***Штрихом Шефера*** *двух высказываний $A$ и $B$ называется высказывание $A|B$ (читается: "$A$ и-не $B$"), которое является отрицанием конъюнкции этих высказываний.*

**Определение 1.1.9.** ***Стрелкой Пирса*** *или* ***Штрихом Лукасевича*** *двух высказываний $A$ и $B$ называется высказывание $A \downarrow B$ (читается: "$A$ или-не $B$"), которое является отрицанием дизъюнкции этих высказываний.*

Таблица истинности для вышеупомянутых трёх операций представлена в Таблице 1.1.2.

| $A$ | $B$ | $A \oplus B$ | $A\|B$ | $A \downarrow B$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |

Таблица 1.1.2

## 1.2 Формулы логики высказываний

С помощью логических операций над высказываниями можно строить различные сложные высказывания. При этом порядок выполнения операций регулируется скобками. Например, из трёх высказываний $A, B, C$ можно построить высказывания $(A\&B) \vee C$ и $\overline{(A \to C)}\&(B \oplus C)$. Приоритет операций отрицания, конъюнкции, дизъюнкции, импликации и эквивалентности устанавливается в виде: ¬, &, ∨, →, ↔. То есть самой сильной операцией является отрицание, затем конъюнкция, дизъюнкция, импликация и, наконец, эквивалентность. Что касается операций ⊕, | и ↓, то приоритет для них будет устанавливаться исключительно скобками.

В первом параграфе высказывания были введены как повествовательные предложения естественного языка, т. е. как лингвистические объекты. Для изучения этих объектов используется понятие формулы логики высказываний. Дадим соответствующие определения.

**Определение 1.2.1.** *Переменные, вместо которых можно подставлять любые элементарные высказывания или их значения истинности, будем называть* ***пропозициональными*** *(****высказывательными****) переменными.*

Пропозициональные переменные также могут называться *атомарными формулами*. Для записи пропозициональных переменных будем использовать заглавные буквы латинского алфавита с индексами или без них ($A$, $B$, $C$, $X$, $Y$, $Z$, $X_1$, $Y_1$, $Y_3$, ...).

С помощью пропозициональных переменных и символов логических операций любое высказывание можно формализовать, т. е. заменить формулой, выражающей его логическую структуру.

Например, высказывание “Если в четырёхугольнике две противоположные стороны конгруэнтны и параллельны, то этот четырёхугольник параллелограмм” формализуется в виде $(X\&Y) \to Z$, где переменной $X$ соответствует высказывание “В четырёхугольнике две противоположные стороны конгруэнтны”, переменной $Y$ соответствует высказывание “В четырёхугольнике две противоположные стороны параллельны”, а переменной $Z$ – “Четырёхугольник является параллелограммом”.

Для того чтобы определить понятие формулы логики высказываний, сначала зададим *алфавит* (набор символов, которые мы будем употреблять в логике высказываний):

1) $A$, $B$, $C$, $X$, $Y$, $Z$, $X_1, Y_2$, $Y_3$, ... – символы пропозициональных переменных;
2) 0, 1 – символы, обозначающие логические константы “истина” и “ложь”;
3) ¬, &, ∨, ⊕, →, ↔, |, ↓ – символы логических операций;

4) (, ) – вспомогательные символы (скобки), служащие для указания порядка выполнения операций.

С помощью элементов алфавита можно построить разнообразные логические формулы. Дадим теперь строгое определение формулы логики высказываний.

**Определение 1.2.2.** ***Формулами логики высказываний*** *называются:*

1) *атомарные формулы;*
2) *символы истины 1 и лжи 0;*
3) *выражения вида* $\bar{A}$, $(A\&B)$, $(A \vee B)$, $(A \to B)$, $(A \leftrightarrow B)$, $(A \oplus B)$, $(A|B)$, $(A \downarrow B)$, *где A и B также являются формулами.*

*Других формул, кроме построенных по правилам трёх предыдущих пунктов, нет.*

Вышестоящее определение относится к индуктивным определениям, в которых вводятся базовые объекты (атомарные формулы и символы 0 и 1) и способы порождения новых объектов из уже существующих, либо полученных ранее.

Относительно любой последовательности знаков алфавита языка логики высказываний можно решить, является она формулой или нет. Если эта последовательность может быть построена в соответствии с Определением 1.2.2, то она является формулой, в противном случае – нет.

Приведём некоторые примеры выражений, не являющихся формулами:

$$A \oplus \big((B \leftrightarrow \bar{A}\&) \to\big) C\&B\big), \ \overline{A \vee B} \to \overline{\big((CB \leftrightarrow \bar{B})\&\bar{\oplus}A\big)}.$$

Из Определения 1.2.2 можно заметить, что формулы насыщены скобками и как результат трудночитаемы, поэтому для упрощения записи формул принят ряд соглашений:

1. Наружные скобки в записи формул можно опускать.
2. Если над формулой стоит знак отрицания, то скобки тоже опускаются.
3. Скобки можно опускать, придерживаясь приоритета выполнения логических операций: ¬, &, ∨, →, ↔ (слева – более приоритетные операции, а справа – менее приоритетные).
4. Конъюнкцию можно обозначать знаком “·” или и вовсе знак конъюнкции можно опускать.

Иногда для более лёгкого восприятия формулы опускают не все скобки. В первую очередь выполняются операции в скобках, затем все остальные логические операции в порядке приоритета.

С учётом сформулированных соглашений формулу

$$\Big(\big(X \vee (\bar{Y}\&Z)\big) \vee \Big(X \leftrightarrow \overline{\big(Y \to (X \oplus Z)\big)}\Big)\Big)$$

можно записать в виде $X \vee (\bar{Y}\&Z) \vee \big(X \leftrightarrow \overline{Y \to (X \oplus Z)}\big)$. Заметим, что длина этой формулы немного уменьшилась, где под *длиной формулы* понимаем число символов, входящих в запись формулы.

**Определение 1.2.3.** ***Подформулой*** *формулы $F$ является слитная часть формулы $F$, которая сама является формулой.*

Для каждой формулы существует конечная последовательность всех её подформул. К примеру, формула $A \to (\bar{B} \to \overline{A \vee B})$ имеет семь подформул: $A$, $B$, $\bar{B}$, $A \vee B$, $\overline{A \vee B}$, $\bar{B} \to \overline{A \vee B}$, $A \to (\bar{B} \to \overline{A \vee B})$.

Каждая формула определяет соответствие между множеством возможных значений всех пропозициональных переменных, из которых состоит формула, и логическими значениями “истина” и “ложь”. Подобное соответствие может быть представлено таблицей истинности. Легко заметить, что добавление каждой новой пропозициональной переменной увеличивает количество строк в таблице истинности вдвое. Из этого следует, что формула, состоящая из $n$ различных переменных, имеет таблицу истинности с $2^n$ строками.

**Пример 1.2.1.** Составить таблицу истинности для формулы логики высказываний $F = \big(X \vee (\bar{Y}\&Z)\big) \oplus (X \to Z)$.

*Решение*. Так как формула зависит от трёх переменных, то её таблица истинности будет содержать $2^3$ строк и 8 столбцов (количество операций плюс три столбца значений переменных).

Отметим порядок выполнения операций:

$$F = \overset{3}{\big(}X \vee \overset{1}{(}\bar{Y}\overset{2}{\&}Z\big)\big) \overset{5}{\oplus} \big(X \overset{4}{\to} Z\big).$$

Заполним таблицу истинности:

| $X$ | $Y$ | $Z$ | $\bar{Y}$ | $\bar{Y}\&Z$ | $X \vee (\bar{Y}\&Z)$ | $X \to Z$ | $F$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Таблица 1.2.1

## 1.3 Равносильность формул

**Определение 1.3.1.** *Формула называется* ***тождественно истинной*** *(или* ***тавтологией****), если она принимает значение "истина" при всех значениях входящих в неё переменных.*

Очевидно, что всегда можно установить, является ли данная формула тавтологией. Для этого достаточно составить её таблицу истинности и удостоверится в том, что последний столбец таблицы содержит только единицы. Для того чтобы убедиться, что формула не есть тавтология, достаточно найти один набор значений переменных, при которых формула принимает значение 0 ("ложь").

Например, формула $F = X\&Y \to X$ является тождественно истинной, потому что последний столбец таблицы истинности формулы $F$ состоит из одних единиц (Таблица 1.3.1).

| $X$ | $Y$ | $X\&Y$ | $F = X\&Y \to X$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Таблица 1.3.1

**Теорема 1.3.1.** *Если формулы $F$ и $F \longrightarrow G$ – тавтологии, то формула $G$ – тоже тавтология.*

*Доказательство.* Пусть $F$ и $F \longrightarrow G$ – тавтологии. Допустим, что при некотором распределении истинностных значений для пропозициональных переменных, входящих в $F$ и $G$, $G$ принимает значение 0. Поскольку $F$ есть тавтология, то при этом распределении истинностных значений $F$ принимает значение 1. Тогда по определению операции импликации формула $F \longrightarrow G$ получит значение 0. Это противоречит предположению о том, что $F \longrightarrow G$ есть тавтология. □

**Определение 1.3.2.** *Формула называется* ***тождественно ложной*** ***(невыполнимой*** *или* ***противоречивой****), если она принимает значение "ложь" при всех значениях входящих в неё переменных.*

Для примера, используя метод подстановки значений переменных, докажем тождественную ложность формулы $F = X\&\overline{Y \to X}$:

$$F(0,0) = 0\&\overline{0 \to 0} = 0,\ F(0,1) = 0\&\overline{1 \to 0} = 0,$$

$$F(1,0) = 1\&\overline{0 \to 1} = 0,\ F(1,1) = 1\&\overline{1 \to 1} = 0.$$

Как видно, при любых значениях пропозициональных переменных формула $F$ равна 0. Это означает, что формула $F$ тождественно ложна (противоречива).

**Определение 1.3.3.** *Формула называется* ***выполнимой****, если на некотором наборе значений переменных она принимает значение "истина".*

**Определение 1.3.4.** *Формула называется* ***опровержимой****, если на некотором наборе значений переменных она принимает значение "ложь".*

Например, формула $F = \bar{X} \to (Y \to X)$ выполнима, так как, например $F(0,0) = \bar{0} \to (0 \to 0) = 1 \to 1 = 1$. Также данная формула опровержима, так как $F(0,1) = \bar{0} \to (1 \to 0) = 1 \to 0 = 0$. Аналогично, можно уверенно утверждать, что данная формула не противоречива и не является тавтологией.

Пусть $F$, $G$ и $H$ – три формулы алгебры логики.

**Определение 1.3.5.** *Формулы $F$ и $G$ называются* ***равносильными*** *(или* ***эквивалентными****), если они принимают одинаковые логические значения на любом наборе значений входящих в них пропозициональных переменных. Равносильность формул в алгебре логики обозначается $F \equiv G$, где знак $\equiv$ называется* ***знаком тождественного равенства****.*

Например, равносильны формулы:

$$X \equiv \bar{\bar{X}} \text{ и } X \vee (Y \& \bar{Y}) \equiv X.$$

Важно отметить, что выражение $F \equiv G$ – не формула в языке логики высказываний. Оно является метаязыковым высказыванием о формулах $F$ и $G$, которое утверждает, что эти формулы равносильны.

Существует тесная связь между понятием равносильности формул и понятием тавтологии.

**Теорема 1.3.2.** *Две формулы алгебры высказываний $F$ и $G$ равносильны тогда и только тогда, когда формула $F \leftrightarrow G$ является тавтологией.*

*Доказательство.* Необходимость. Пусть $F \equiv G$, следовательно, при каждом наборе значений всех пропозициональных переменных, входящих в формулы $F$ и $G$, они принимают одинаковые истинностные значения. Тогда, по определению операции эквивалентности, формула $F \leftrightarrow G$ всегда принимает значение 1, т. е. является тавтологией.

Достаточность. Пусть формула $F \leftrightarrow G$ тавтология, т. е. принимает всегда значение 1. Это, в свою очередь, означает, что формулы $F$ и $G$ имеют всегда одинаковые истинностные значения и, как следствие, они равносильны. □

Легко убедиться, что отношение равносильности обладает следующими свойствами:

a) рефлективность: $F \equiv F$;

b) симметричность: если $F \equiv G$ то $G \equiv F$;

c) транзитивность: если $F \equiv G$ и $G \equiv H$, то $F \equiv H$.

Для наглядности докажем свойство транзитивности.

**Теорема 1.3.3.** *Если $F \equiv G$ и $G \equiv H$, то $F \equiv H$.*

*Доказательство.* В силу Теоремы 1.3.2, формулы $F \leftrightarrow G$ и $G \leftrightarrow H$ являются тавтологиями. По определению операции эквивалентности, формулы $F$ и $G$, а также формулы $G$ и $H$ при каждом наборе значений всех пропозициональных переменных, входящих в них, принимают одинаковые истинностные значения. Откуда следует, что формулы $F$ и $H$ также равносильны. □

В частности, из Теоремы 1.3.3 непосредственно вытекает *правило цепи равносильностей*:

$$\textit{Если } F_1 \equiv F_2, F_2 \equiv F_3, \dots, F_{n-1} \equiv F_n, \textit{ то } F_1 \equiv F_n.$$

Обычно в данном случае используют запись $F_1 \equiv F_2 \equiv \cdots \equiv F_n$ и говорят, что формула $F_n$ получена из $F_1$ путём эквивалентных (равносильных) преобразований.

Заметим, что если в формуле заменить одну из её подформул другой, то получится новая цепочка символов, которая также будет формулой. Справедлива следующая теорема.

**Теорема 1.3.4.** *Пусть формула $F$ есть тавтология, содержащая пропозициональные переменные $X_1, X_2, \dots, X_n$. Если $G$ получается из $F$ путём подстановки в $F$ формул $A_1, A_2, \dots, A_n$ вместо переменных $X_1, X_2, \dots, X_n$ соответственно, то $G$ есть тавтология.*

*Доказательство.* Пусть задано произвольное распределение истинностных значений для пропозициональных переменных, входящих в $G$. Формулы $A_1, A_2, \dots, A_n$ примут тогда некоторые значения $\alpha_1, \alpha_2, \dots, \alpha_n$, где $\alpha_i$ $(i = \overline{1,n})$ есть 0 или 1. Если мы придадим значения $\alpha_1, \alpha_2, \dots, \alpha_n$ соответственно переменным $X_1, X_2, \dots, X_n$, то так как по условию $F$ есть тавтология, $F$ будет истинна, и это же значение принимает $G$. Таким образом, при произвольных значениях пропозициональных переменных формула $G$ принимает значение 1. □

Например, если в тавтологии $\overline{X_1} \to (X_1 \to X_2)$ формулы $A \leftrightarrow B$ и $A|(C \oplus B)$ подставить вместо переменных $X_1$ и $X_2$, то получим тавтологию $\overline{A \leftrightarrow B} \to \big((A \leftrightarrow B) \to (A|(C \oplus B))\big)$.

**Пример 1.3.1.** Показать, что формулы $F = A \vee (\bar{A}\&B)$ и $G = A \vee B$ равносильны.

*Решение.* Для этого построим таблицу истинности и убедимся, что логические значения формул на любом наборе значений входящих в формулы пропозициональных переменных совпадают.

| $A$ | $B$ | $\bar{A}$ | $\bar{A}\&B$ | $F = A \vee (\bar{A}\&B)$ | $G = A \vee B$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |

Таблица 1.3.2

**Пример 1.3.2.** Доказать равносильность: $A \leftrightarrow B \equiv (A \longrightarrow B)\&(B \longrightarrow A)$.

*Решение.* Так как при одинаковых логических значениях высказываний $A$ и $B$ истинными являются формулы $A \leftrightarrow B$, $A \longrightarrow B$ и $B \longrightarrow A$, то истинной будет и конъюнкция $(A \longrightarrow B)\&(B \longrightarrow A)$. Следовательно, в этом случае обе части равносильности имеют одинаковые истинные значения.

Предположим, что $A$ и $B$ имеют различные логические значения. Тогда будут ложными эквивалентность $A \leftrightarrow B$ и одна из двух импликаций $A \longrightarrow B$ или $B \longrightarrow A$. Важно, что при этом будет ложной и конъюнкция $(A \longrightarrow B)\&(B \longrightarrow A)$.

Таким образом, и в этом случае обе части равносильности имеют одинаковые логические значения.

Ниже приведён ряд наиболее важных **законов алгебры логики**.

1. $A \equiv A$ – Закон тождества.
2. $A\&\bar{A} \equiv 0$ – Закон противоречия.
3. $A \vee \bar{A} \equiv 1$ – Закон исключённого третьего.
4. $A \equiv \bar{\bar{A}}$ – Закон двойного отрицания.
5. Свойства констант:
   a) $\bar{0} \equiv 1$;
   b) $\bar{1} \equiv 0$;
   c) $A \vee 0 \equiv A$;
   d) $A\&0 \equiv 0$;
   e) $A \vee 1 \equiv 1$;
   f) $A\&1 \equiv A$.
6. Законы идемпотентности:
   a) $A\&A \equiv A$;
   b) $A \vee A \equiv A$.

7. Законы коммутативности (переместительности):
   a) $A\&B \equiv B\&A$;
   b) $A \vee B \equiv B \vee A$;
   c) $A \oplus B \equiv B \oplus A$;
   d) $A \leftrightarrow B \equiv B \leftrightarrow A$.
8. Законы ассоциативности (сочетательности):
   a) $(A\&B)\&C \equiv A\&(B\&C)$;
   b) $(A \vee B) \vee C \equiv A \vee (B \vee C)$;
   c) $(A \oplus B) \oplus C \equiv A \oplus (B \oplus C)$.
9. Законы дистрибутивности (распределительности):
   a) $A\&(B \vee C) \equiv (A\&B) \vee (A\&C)$;
   b) $A \vee (B\&C) \equiv (A \vee B)\&(A \vee C)$;
   c) $A\&(B \oplus C) \equiv (A\&B) \oplus (A\&C)$.
10. Законы поглощения:
    a) $A \vee (A\&B) \equiv A$;
    b) $A\&(A \vee B) \equiv A$.
11. Законы де Моргана:
    a) $\overline{A\&B} \equiv \bar{A} \vee \bar{B}$;
    b) $\overline{A \vee B} \equiv \bar{A}\&\bar{B}$.
12. Законы композиции:
    a) $A \longrightarrow B \equiv \bar{B} \longrightarrow \bar{A} \equiv \bar{A} \vee B$;
    b) $A \leftrightarrow B \equiv (A \longrightarrow B)\&(B \longrightarrow A) \equiv (A\&\, B) \vee (\bar{A}\&\, \bar{B})$;
    c) $A \oplus B \equiv (A\&\, \bar{B}) \vee (\bar{A}\&\, B)$.
13. Законы Порецкого:
    a) $A \vee (\bar{A}\&B) \equiv A \vee B$;
    b) $A\&(\bar{A} \vee B) \equiv A\&B$.

Законы алгебры логики могут быть использованы для доказательства равносильностей, для приведения формул к заданному виду и для упрощения формул.

Считается, что формула $A$ проще равносильной ей формулы $B$, если она содержит меньше букв и логических операций. При этом обычно более простая формула должна содержать только операции конъюнкции, дизъюнкции и отрицания.

**Пример 1.3.3.** Используя равносильные преобразования, доказать:

$$A \leftrightarrow B \equiv (A\&\, B) \vee (\bar{A}\&\, \bar{B}).$$

*Решение*. Запишем цепочку равносильных формул:

$$A \leftrightarrow B \equiv (A \longrightarrow B)\&(B \longrightarrow A) \equiv (\bar{A} \vee B)\&(\bar{B} \vee A) \equiv$$
$$\equiv (\bar{A} \vee B)\&\bar{B} \vee (\bar{A} \vee B)\&A \equiv \bar{A}\&\bar{B} \vee B\&\bar{B} \vee \bar{A}\&A \vee B\&A \equiv$$
$$\equiv \bar{A}\&\bar{B} \vee 0 \vee 0 \vee B\&A \equiv \bar{A}\&\bar{B} \vee B\&A \equiv$$
$$\equiv (A\&\, B) \vee (\bar{A}\&\, \bar{B}).$$

**Пример 1.3.4.** Используя равносильные преобразования, доказать, что формула $A \longrightarrow (B \longrightarrow A)$ тождественно истинна.

*Решение*. Запишем цепочку равносильных формул:

$$A \longrightarrow (B \longrightarrow A) \equiv \bar{A} \vee (\bar{B} \vee A) \equiv \bar{A} \vee (A \vee \bar{B}) \equiv$$
$$\equiv (\bar{A} \vee A) \vee \bar{B} \equiv 1 \vee \bar{B} \equiv 1.$$

**Пример 1.3.5.** С помощью равносильных преобразований упростить формулу $(\bar{A} \vee B)\&(A|B)$.

*Решение*. Запишем цепочку равносильных формул:

$$(\bar{A} \vee B)\&(A|B) \equiv (\bar{A} \vee B)\&\overline{A\&B} \equiv (\bar{A} \vee B)\&(\bar{A} \vee \bar{B}) \equiv$$
$$\equiv \bar{A}\&\bar{A} \vee \bar{A}\&\bar{B} \vee B\&\bar{A} \vee B\&\bar{B} \equiv$$
$$\equiv \bar{A} \vee \bar{A}\&\bar{B} \vee B\&\bar{A} \vee 0 \equiv$$
$$\equiv \bar{A}\&(1 \vee \bar{B} \vee B) \equiv \bar{A}\&1 \equiv \bar{A}.$$

## 1.4 Закон двойственности

В этом параграфе мы рассмотрим формулы, содержащие только операции конъюнкции, дизъюнкции и отрицания.

**Теорема 1.4.1.** *Для каждой пропозициональной формулы $F$ существует равносильная ей формула, содержащая только связки &, ∨ и ¬, причём связка ¬ относится только к атомарным формулам.*

*Доказательство.* Согласно равносильностям из предыдущего параграфа известно, что связки →, ↔, ⊕, | и ↓ могут быть выражены через связки &, ∨ и ¬. Если ¬ стоит перед некоторой скобкой, то на основании законов де Моргана можно внести ¬ под скобки, при этом связка & меняется на ∨, в свою очередь, ∨ на &, а ¬ будет относится только к пропозициональным переменным. □

Будем называть операцию конъюнкции *двойственной* операции дизъюнкции, а операцию дизъюнкции *двойственной* операции конъюнкции. Введём также понятие двойственных формул.

**Определение 1.4.1.** *Формулы $F$ и $F^*$ называются* ***двойственными****, если одна получается из другой путём замены каждой операции на двойственную.*

Например, формулы $F = X\&(Y \vee \bar{Z})$ и $F^* = X \vee (Y\&\bar{Z})$ – двойственны.

**Теорема 1.4.2.** *Если формулы $F$ и $G$ равносильны, то и двойственные им формулы $F^*$ и $G^*$ также равносильны.*

*Доказательство.* Пусть формулы $F$ и $G$ равносильны. Обозначим через $X_1, X_2, \dots, X_n$ пропозициональные переменные, входящие в $F$ или $G$. Будем считать, что $X_1, X_2, \dots, X_n$ входят и в $F$, и в $G$. Если это не так, например, $G$ не содержит переменную $X_k$, $1 \le k \le n$, входящую в $F$, то $G$ можно заменить равносильной формой $G \vee (X_k\&\overline{X_k})$, которая содержит эту переменную. Таким образом, всегда можем добиться того, чтобы формулы $F$ и $G$ содержали все переменные $X_1, X_2, \dots, X_n$.

Согласно условию

$$F(X_1, X_2, \dots, X_n) \equiv G(X_1, X_2, \dots, X_n). \tag{1.1}$$

Если формулы $F$ и $G$ равносильны, то, очевидно, равносильны и их отрицания, поэтому из (1.1) получим

$$\overline{F(X_1, X_2, \dots, X_n)} \equiv \overline{G(X_1, X_2, \dots, X_n)}. \tag{1.2}$$

Путём равносильных преобразований добьёмся того, чтобы в выражение (1.2) знак отрицания относился только к пропозициональным переменным. При этом согласно законам де Моргана связки & и ∨ поменяются на двойственные. Следовательно, получим

$$F^*(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n) \equiv G^*(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n) \tag{1.3}$$

По определению равносильности формула $F^*(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n)$ и формула $G^*(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n)$ принимают одинаковые значения при любых наборах значений пропозициональных переменных $X_1, X_2, \dots, X_n$. Поэтому если вместо переменных $X_1, X_2, \dots, X_n$ подставить их отрицания $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n$, то формулы останутся равносильными. Имея в виду тот факт, что $\bar{\bar{X}}_i$ равносильна $X_i$ для всякого $i$, $1 \le i \le n$, получим

$$F^*(X_1, X_2, \dots, X_n) \equiv G^*(X_1, X_2, \dots, X_n),$$

что и требовалось доказать. □

## 1.5 Нормальные формы

**Определение 1.5.1.** *Конъюнкция некоторых пропозициональных переменных или их отрицаний называется* ***элементарной конъюнкцией (конъюнктом)****.*

**Определение 1.5.2.** *Дизъюнкция некоторых пропозициональных переменных или их отрицаний называется* ***элементарной дизъюнкцией (дизъюнктом)****.*

Например, формулы $X\&\bar{Y}$ и $\bar{X}_2\&X_1\&X_3\&\bar{X}_3$ являются элементарными конъюнкциями, а формулы $\bar{X} \vee Y \vee Z$ и $X_1 \vee X_4 \vee X_3 \vee \bar{X}_2$ – элементарными дизъюнкциями.

**Определение 1.5.3.** *Атомарная формула или её отрицание называется* ***литералом****.*

**Определение 1.5.4.** ***Дизъюнктивной нормальной формой (ДНФ)*** *называется произвольная дизъюнкция элементарных конъюнкций. Говорят, что формула находится в дизъюнктивной нормальной форме.*

**Определение 1.5.5.** ***Конъюнктивной нормальной формой (КНФ)*** *называется произвольная конъюнкция элементарных дизъюнкций. Говорят, что формула находится в конъюнктивной нормальной форме.*

Например, формулы $X_1$, $X_1\&\bar{X}_2$ и $(X_1\&X_2\&X_3) \vee (X_2\&\bar{X}_3)$ находятся в ДНФ, а формулы $\bar{X}_2$, $X_1 \vee X_2$ и $(X_1 \vee X_2 \vee \bar{X}_3)\&X_1\&(\bar{X}_1 \vee \bar{X}_3)$ находятся в КНФ.

В ДНФ нет двух одинаковых элементарных конъюнкций, а в КНФ нет двух одинаковых элементарных дизъюнкций, потому что по закону идемпотентности $A\&A \equiv A \equiv A \vee A$.

**Теорема 1.5.1.** *Для каждой формулы существует равносильная ей ДНФ и КНФ (не единственные).*

*Доказательство.* Ранее была доказана Теорема 1.4.1, которая утверждает, что для каждой формулы существует равносильная ей формула, содержащая только связки &, ∨ и ¬, причём связка ¬ относится только к атомарным высказываниям. Если полученная формула не находится в ДНФ или КНФ, то можно применять операции раскрытия скобок (законы дистрибутивности) до тех пор, пока не будет получена ДНФ или КНФ исходной формулы. □

***Алгоритм приведения к нормальной форме***

*Шаг* 1. Исключить из исходной формулы все логические связки, кроме &, ∨ *и* ¬.

*Шаг* 2. Продвинуть отрицание к атомарным формулам, использую законы де Моргана и двойного отрицания.

*Шаг* 3. Применить закон дистрибутивности конъюнкции относительно дизъюнкции (дизъюнкции относительно конъюнкции).

**Пример 1.5.1.** Привести к ДНФ и КНФ следующую формулу:

$$F = \bar{Y}\&\big(Z \to (X \leftrightarrow Y)\big).$$

*Решение*. Построим сначала ДНФ для $F$:

$$\bar{Y}\&\big(Z \to (X \leftrightarrow Y)\big) \equiv \bar{Y}\&\big(\bar{Z} \vee (X \to Y)\&(Y \to X)\big) \equiv$$

$$\equiv \bar{Y}\&\big(\bar{Z} \vee (\bar{X} \vee Y)\&(\bar{Y} \vee X)\big) \equiv \bar{Y}\&\bar{Z} \vee \bar{Y}\&(\bar{X} \vee Y)\&(\bar{Y} \vee X) \equiv$$

$$\equiv \bar{Y}\&\bar{Z} \vee \bar{Y}\&(\bar{X}\&\bar{Y} \vee \bar{X}\&X \vee Y\&\bar{Y} \vee Y\&X) \equiv$$

$$\equiv \bar{Y}\&\bar{Z} \vee \bar{Y}\&(\bar{X}\&\bar{Y} \vee 0 \vee 0 \vee Y\&X) \equiv \bar{Y}\&\bar{Z} \vee \bar{Y}\&\bar{X}\&\bar{Y} \vee \bar{Y}\&Y\&X \equiv$$

$$\equiv \bar{Y}\&\bar{Z} \vee \bar{Y}\&\bar{X} \vee 0 \equiv \bar{Y}\&\bar{Z} \vee \bar{X}\&\bar{Y}.$$

Преобразуем полученную ДНФ в КНФ:

$$\bar{Y}\&\bar{Z} \vee \bar{X}\&\bar{Y} \equiv (\bar{Y} \vee \bar{X})\&(\bar{Y} \vee \bar{Y})\&(\bar{Z} \vee \bar{X})\&(\bar{Z} \vee \bar{Y}) \equiv$$

$$\equiv (\bar{Y} \vee \bar{X})\&\bar{Y}\&(\bar{Z} \vee \bar{X})\&(\bar{Z} \vee \bar{Y}).$$

## 1.6 Проблема разрешимости

*Проблемой разрешимости* для алгебры высказываний называют следующую проблему: существует ли алгоритм, позволяющий для произвольной логической формулы в конечное число шагов выяснить, является ли она тождественно истинной, выполнимой или тождественно ложной?

Очевидно, что эта проблема разрешима для алгебры высказываний, поскольку всегда можно перебрать все возможные наборы значений переменных и вычислить на них значения заданной формулы (то есть составить таблицу истинности). Однако практическое использование таблицы истинности для больших формул весьма затруднительно. Поэтому для установления тождественной истинности или тождественной ложности формул часто используют другую процедуру распознавания, связанную с приведением формулы к КНФ или ДНФ.

Сформулируем соответствующие теоремы.

**Теорема 1.6.1.** *Для того чтобы элементарная дизъюнкция была тождественно истинной (тавтологией), необходимо и достаточно, чтобы в ней содержалась хотя бы одна пропозициональная переменная вместе со своим отрицанием.*

*Доказательство.* Необходимость. Пусть некоторая элементарная дизъюнкция – тавтология и в ней одновременно не содержится переменная вместе с её отрицанием. В таком случае каждой переменной, не стоящей под знаком отрицания, задаём значение 0, а каждой, стоящей под знаком

отрицания, задаём значение 1. После указанной подстановки каждое слагаемое примет значение 0 и, следовательно, вся элементарная дизъюнкция примет значение 0, а это в свою очередь означает, что она не является тавтологией. Полученное противоречие и доказывает необходимость.

Достаточность. Пусть некоторая элементарная дизъюнкция содержит переменную и её отрицание, то есть имеет вид $X \vee \bar{X} \vee ...$ Для нас важно, что имеются слагаемые $X$ и $\bar{X}$, остальные слагаемые могут быть, а могут и отсутствовать. Формула $X \vee \bar{X}$ всегда принимает значение 1, поэтому и вся элементарная дизъюнкция будет истиной при любых значениях пропозициональных переменных, в неё входящих. Следовательно, данная элементарная дизъюнкция – тавтология. □

Аналогично доказывается и следующая теорема.

**Теорема 1.6.2.** *Для того чтобы элементарная конъюнкция была тождественно ложной (противоречием), необходимо и достаточно, чтобы в ней содержалась хотя бы одна пропозициональная переменная вместе со своим отрицанием.*

**Теорема 1.6.3.** *Для того чтобы формула $F$ была тождественно ложной (противоречием), необходимо и достаточно, чтобы равносильная ей ДНФ содержала в каждой элементарной конъюнкции некоторую переменную вместе с её отрицанием.*

*Доказательство.* Пусть для формулы $F$ равносильной ей ДНФ является формула

$$A_1 \vee A_2 \vee ... \vee A_k, k \geq 1, \qquad (1.4)$$

где $A_i$ $(i = \overline{1,k})$ есть элементарная конъюнкция. Дизъюнкция (1.4) будет противоречием тогда и только тогда, когда будет противоречием каждая элементарная конъюнкция $A_i$. Согласно Теореме 1.6.2 фиксированная элементарная конъюнкция $A_i$ является противоречием тогда и только тогда, когда она содержит хотя бы одну пропозициональную переменную вместе со своим отрицанием. □

**Следствие 1.6.1.** *Формула $F$ является выполнимой, если равносильная ей ДНФ содержит хотя бы одну элементарную конъюнкцию, в которой нет некоторой переменной вместе с её отрицанием.*

Легко доказывается и следующая теорема.

**Теорема 1.6.4.** *Для того чтобы формула F была тождественно истинной (тавтологией), необходимо и достаточно, чтобы равносильная ей КНФ содержала в каждой элементарной дизъюнкции некоторую переменную вместе с её отрицанием.*

Приведённая теорема позволяет очень просто выяснить является ли формула тавтологией или нет. Для этого достаточно построить её КНФ и проверить если каждая элементарная дизъюнкция содержит некоторую переменную вместе с её отрицанием.

Заметим, что также просто выяснить, используя Теорему 1.6.4, выполнима ли формула $F$ или нет. Для этого достаточно найти КНФ для $\bar{F}$ и если найденная КНФ является тавтологией, то $F$ не выполнима, если же найденная КНФ не тавтология, то формула $F$ выполнима.

**Пример 1.6.1.** Определить является ли тавтологией формула:

$$F = Y \vee \bar{Y}\&X \vee \bar{Y}\&\bar{X}.$$

*Решение*. Построим цепочку равносильностей и получим КНФ:

$$Y \vee \bar{Y}\&X \vee \bar{Y}\&\bar{X} \equiv Y \vee \bar{Y}\&(X \vee \bar{X}) \equiv (Y \vee \bar{Y})\&(Y \vee X \vee \bar{X}).$$

Так как каждая элементарная дизъюнкция $Y \vee \bar{Y}$ и $Y \vee X \vee \bar{X}$, входящая в КНФ формулы $F$, содержит переменную и ее отрицание, то $F$ является тавтологией.

## 1.7 Совершенные нормальные формы

Согласно Параграфу 1.5 для одной и той же формулы ДНФ и КНФ определены неоднозначно. То есть зачастую одна и та же формула, находящаяся в ДНФ, может быть приведена к другой равносильной ей ДНФ.

Введём следующее понятие.

**Определение 1.7.1.** *Множество всех равносильных формул будем называть* ***классом равносильных формул*** *(или* ***классом эквивалентных формул****).*

В этом параграфе докажем, что для любой формулы существуют единственные дизъюнктивная и конъюнктивная нормальные формы специального вида, которые определяют класс равносильных формул. Формулы такого вида называются *совершенными нормальными формами*.

**Определение 1.7.2.** ***Совершенной дизъюнктивной нормальной формой (СДНФ)*** *формулы F называется ДНФ этой формулы, удовлетворяющая следующим условиям:*

1) *в каждую элементарную конъюнкцию входят все переменные один и только один раз (и только они) с отрицанием либо без отрицания;*
2) *в ней нет одинаковых элементарных конъюнкций.*

**Определение 1.7.3.** ***Совершенной конъюнктивной нормальной формой (СКНФ)*** *формулы $F$ называется КНФ этой формулы, удовлетворяющая следующим условиям:*

1) *в каждую элементарную дизъюнкцию входят все переменные один и только один раз (и только они) с отрицанием либо без отрицания;*
2) *в ней нет одинаковых элементарных дизъюнкций.*

Докажем две дополнительные теоремы.

**Теорема 1.7.1.** *Число всех упорядоченных наборов (кортежей) длины $n$ $(\alpha_1, \alpha_2, \dots, \alpha_n)$, где $\alpha_i$ $(i = \overline{1,n})$ может принимать значение 1 или 0, равно $2^n$.*

*Доказательство.* Докажем с помощью математической индукции по числу элементов $n$. Для $n = 1$ существуют всего два набора: 1 или 0. Следовательно, для $n = 1$ утверждение теоремы верно.

Предположим теперь, что утверждение теоремы верно для $n = k$, $k \geq 1$. Покажем, что оно справедливо и для $n = k + 1$. Для этого представим число всех кортежей длины $k + 1$ в виде:

$$|(\alpha_1, \alpha_2, \dots, \alpha_k, \alpha_{k+1})| = |(\alpha_1, \alpha_2, \dots, \alpha_k, 1)| + |(\alpha_1, \alpha_2, \dots, \alpha_k, 0)| =$$
$$= 2^k + 2^k = 2 * 2^k = 2^{k+1}.$$

Следовательно, теорема доказана. □

**Теорема 1.7.2.** *Число всех классов равносильных формул, состоящих из $n$ переменных, равно $2^{2^n}$.*

*Доказательство.* Известно, что в алгебре высказываний любая формула может быть выражена с помощью таблицы истинности. Для этого необходимо указать значение формулы для всех наборов значений переменных $X_1, X_2, \dots, X_n$, входящих в данную формулу. Согласно Теореме 1.7.1 таких наборов ровно $2^n$. Для каждого такого набора формула может принимать значение 0 или 1. Следовательно, число всех формул, состоящих из $n$ переменных, равно $2^{2^n}$. □

**Теорема 1.7.3.** *Для любой формулы $F$, не являющейся противоречием, существует единственная СДНФ.*

*Доказательство.* Для начала докажем существование СДНФ. Имея в виду Теорему 1.5.1, будем считать, что формула $F$ находится в ДНФ. Если $F$ удовлетворяет условиям 1) и 2) из Определения 1.7.2, тогда $F$ и есть СДНФ. Иначе проанализируем следующие два случая:

a) Формула $F$ не удовлетворяет условию 1) из Определения 1.7.2. В каждой элементарной конъюнкции оставляем только один из идентичных литералов. Если в $F$ входит элементарная конъюнкция (для определённости обозначим её через $A$), которая не содержит одну из пропозициональных переменных (пусть этой переменной будет $X$), то в формуле $F$ заменяем $A$ на $A\&1 \equiv A\&(X \vee \bar{X}) \equiv A\&X \vee A\&\bar{X}$.

b) Формула $F$ не удовлетворяет условию 2). В $F$ сохраняем только одну из идентичных элементарных конъюнкций.

В итоге для любой формулы легко может быть получена СДНФ.

Теперь докажем единственность СДНФ. Обозначим пропозициональные переменные, входящие в $F$, через $X_1, X_2, \dots, X_n$. Найдём количество всех СДНФ, состоящих из $n$ переменных. Для этого опишем СДНФ, содержащую все элементарные конъюнкции:

$$X_1\&X_2\& \dots \&X_n \vee \bar{X}_1\&X_2\& \dots \&X_n \vee X_1\&\bar{X}_2\& \dots \&X_n \vee \dots$$
$$\dots \vee \bar{X}_1\&\bar{X}_2\& \dots \&\bar{X}_{n-1}\&X_n \vee \bar{X}_1\&\bar{X}_2\& \dots \&\bar{X}_{n-1}\&\bar{X}_n \qquad (1.5)$$

При присвоении каждой переменной значения 0, если она входит без отрицания, и 1, если она входит с отрицанием, каждая элементарная конъюнкция может быть представлена вектором длины $n$, состоящим из значений 1 и 0. По Теореме 1.7.1, таких наборов всего $2^n$. Пронумеруем все элементарные конъюнкции номерами $1,2, \dots, 2^n$. Далее составим наборы $(\alpha_1, \alpha_2, \dots, \alpha_{2^n})$, где $\alpha_i = 0$, если конъюнкция под номером $i$ входит в СДНФ, и $\alpha_i = 1$, если не входит. Следовательно, число всех таких наборов равно $2^{2^n}$. Исключая набор $(0,0, \dots,0)$, получим, что число всех возможных СДНФ равно $2^{2^n} - 1$ (без класса тождественно ложных формул). Согласно Теореме 1.7.2 число всех классов равносильных формул, состоящих из $n$ переменных, равно $2^{2^n}$. Так как каждому классу равносильных формул соответствует хотя бы одна СДНФ, то следует, что каждому классу соответствует одна единственная СДНФ. □

**Теорема 1.7.4.** *Для любой формулы F, не являющейся тавтологией, существует единственная СКНФ.*

Доказательство Теоремы 1.7.4 производится аналогично доказательству Теоремы 1.7.3.

СДНФ и СКНФ для формулы $F$ могут быть получены методом равносильных преобразований. Для этого можно использовать следующий алгоритм.

***Алгоритм приведения к совершенной нормальной форме***

*Шаги* 1 - 3 – те же, что и в алгоритме приведения к нормальной форме, указанном в Параграфе 1.5.

*Шаг* 4. Если в конъюнкт (дизъюнкт) $B$ входит переменная вместе со своим отрицанием, то $B$ удаляется из формулы.

*Шаг* 5. Если в конъюнкт (дизъюнкт) $B$ один литерал входит несколько раз, то из $B$ удаляются все такие литералы, кроме одного.

*Шаг* 6. Если конъюнкт (дизъюнкт) $B$ не содержит ни атомарной формулы $X$, ни её отрицания, то $X$ вводится в конъюнкт (дизъюнкт) $B$ с помощью следующих правил:

Если $B$ – конъюнкт, то

$$B \equiv B\&1 \equiv B\&(X \vee \bar{X}) \equiv B\&X \vee B\&\bar{X}.$$

Если $B$ – дизъюнкт, то

$$B \equiv B \vee 0 \equiv B \vee (X\&\bar{X}) \equiv (B \vee X)\&(B \vee \bar{X}).$$

*Шаг* 7. Если формула содержит одинаковые элементарные конъюнкции (дизъюнкции), то все они удаляются, кроме одной.

Также, СДНФ и СКНФ для формулы $F$ могут быть построены по таблице истинности этой формулы.

Для того чтобы построить СДНФ по таблице истинности, выбираем строки, где $F$ принимает значение 1 (пусть это будут строки $k_1, k_2, \dots, k_m$). Для каждой отдельной выбранной строки $k_i$ $(i = \overline{1,m})$ строим элементарную конъюнкцию $K_i$ следующим образом. Если в выбранной строке $k_i$ переменная $X_j$ принимает значение 1, то в $K_i$ она входит без отрицания, если же $X_j$ принимает значение 0, то в $K_i$ она входит с отрицанием. Дизъюнкция полученных элементарных конъюнкций и будет СДНФ формулы $F$.

Аналогично для того, чтобы с помощью таблицы истинности построить СКНФ для формулы $F$, выбираем строки, где $F$ принимает значение 0. Для каждой строки, в которой $F$ принимает значение 0, строим элементарную дизъюнкцию $K$ следующим образом. Если в выбранной строке переменная $X_j$ принимает значение 1, то в $K$ она входит с отрицанием, если же $X_j$ принимает значение 0, то в $K$ она входит без отрицания. Конъюнкция полученных элементарных дизъюнкций и будет СКНФ формулы $F$.

**Пример 1.7.1.** Применяя равносильные преобразования, найти СДНФ и СКНФ для формулы

$$F = \bar{Y}\&\big(Z \to (X \leftrightarrow Y)\big).$$

*Решение*. В Примере 1.5.1 мы уже получили ДНФ и КНФ для формулы $F$. Получившиеся нормальные формы не содержат одинаковых элементарных конъюнкций или дизъюнкций. Осталось пополнить каждую элементарную конъюнкцию или дизъюнкцию относительно набора пере-

менных $\{X, Y, Z\}$, чтобы получить соответственно СДНФ и СКНФ формулы $F$.

В начале построим СДНФ из ДНФ:

$$\bar{Y}\&\bar{Z} \vee \bar{X}\&\bar{Y} \equiv \bar{Y}\&\bar{Z}\&1 \vee \bar{X}\&\bar{Y}\&1 \equiv$$

$$\equiv \bar{Y}\&\bar{Z}\&(X \vee \bar{X}) \vee \bar{X}\&\bar{Y}\&(Z \vee \bar{Z}) \equiv$$

$$\equiv \bar{Y}\&\bar{Z}\&X \vee \bar{Y}\&\bar{Z}\&\bar{X} \vee \bar{X}\&\bar{Y}\&Z \vee \bar{X}\&\bar{Y}\&\bar{Z} \equiv$$

$$\equiv \bar{Y}\&\bar{Z}\&X \vee \bar{Y}\&\bar{Z}\&\bar{X} \vee \bar{X}\&\bar{Y}\&Z.$$

Построим СКНФ из КНФ:

$$\bar{Y}\&(\bar{Y} \vee \bar{X})\&(\bar{Z} \vee \bar{X})\&(\bar{Z} \vee \bar{Y}) \equiv$$

$$\equiv (\bar{Y} \vee 0)\&(\bar{Y} \vee \bar{X} \vee 0)\&(\bar{Z} \vee \bar{X} \vee 0)\&(\bar{Z} \vee \bar{Y} \vee 0) \equiv$$

$$\equiv (\bar{Y} \vee X\&\bar{X} \vee Z\&\bar{Z})\&(\bar{Y} \vee \bar{X} \vee Z\&\bar{Z})\&$$

$$\&(\bar{Z} \vee \bar{X} \vee Y\&\bar{Y})\&(\bar{Z} \vee \bar{Y} \vee X\&\bar{X}) \equiv$$

$$\equiv \big(\bar{Y} \vee (X \vee Z)\&(X \vee \bar{Z})\&(\bar{X} \vee Z)\&(\bar{X} \vee \bar{Z})\big)\&(\bar{Y} \vee \bar{X} \vee Z)\&$$

$$\&(\bar{Y} \vee \bar{X} \vee \bar{Z})\&(\bar{Z} \vee \bar{X} \vee Y)\&(\bar{Z} \vee \bar{X} \vee \bar{Y})\&(\bar{Z} \vee \bar{Y} \vee X)\&(\bar{Z} \vee \bar{Y} \vee \bar{X}) \equiv$$

$$\equiv (\bar{Y} \vee X \vee Z)\&(\bar{Y} \vee X \vee \bar{Z})\&(\bar{Y} \vee \bar{X} \vee Z)\&(\bar{Y} \vee \bar{X} \vee \bar{Z})\&(\bar{Y} \vee \bar{X} \vee Z)\&$$

$$\&(\bar{Y} \vee \bar{X} \vee \bar{Z})\&(\bar{Z} \vee \bar{X} \vee Y)\&(\bar{Z} \vee \bar{X} \vee \bar{Y})\&(\bar{Z} \vee \bar{Y} \vee X)\&(\bar{Z} \vee \bar{Y} \vee \bar{X}) \equiv$$

$$\equiv (\bar{Y} \vee X \vee Z)\&\ (\bar{Y} \vee X \vee \bar{Z})\&(\bar{Y} \vee \bar{X} \vee Z)\&(\bar{Y} \vee \bar{X} \vee \bar{Z})\&(\bar{Z} \vee \bar{X} \vee Y).$$

**Пример 1.7.2.** Найти СДНФ и СКНФ по таблице истинности для формулы $F = \big(X \downarrow (Y \to Z)\big) \oplus (X|\bar{Z})$.

*Решение*. Сначала составим таблицу истинности для $F$:

| $X$ | $Y$ | $Z$ | $\bar{Z}$ | $Y \to Z$ | $X\|\bar{Z}$ | $X \downarrow (Y \to Z)$ | $F$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

Таблица 1.7.1

Составим СДНФ. Для этого выберем строки, где $F$ принимает значение 1, т. е. строки с номерами 1, 2, 4, 6 и 8. Для первой строки элемен-

тарная конъюнкция представляется в виде $\bar{X}\&\bar{Y}\&\bar{Z}$, для второй - $\bar{X}\&\bar{Y}\&Z$. Построив таким же образом элементарные конъюнкции для оставшихся строк, получим СДНФ формулы $F$:

$$\bar{X}\&\bar{Y}\&\bar{Z} \vee \bar{X}\&\bar{Y}\&Z \vee \bar{X}\&Y\&Z \vee X\&\bar{Y}\&Z \vee X\&Y\&Z.$$

Теперь составим СКНФ. Для этого выберем строки, в которых формула $F$ принимает значение 0, т. е. строки 3, 5 и 7. Для третьей строки элементарная дизъюнкция представляется в виде $X \vee \bar{Y} \vee Z$, для пятой строки – $\bar{X} \vee Y \vee Z$, а для седьмой – $\bar{X} \vee \bar{Y} \vee Z$. СКНФ для формулы $F$ имеет вид:

$$(X \vee \bar{Y} \vee Z)\&(\bar{X} \vee Y \vee Z)\&(\bar{X} \vee \bar{Y} \vee Z).$$

## 1.8 Задачи для самостоятельного решения

1. Составить таблицу истинности для формул:
   a) $(A \to B)\&(\bar{B} \vee A)$;
   b) $(A \leftrightarrow B) \to (A\&B)$;
   c) $A \vee \overline{B \to C} \vee (A \oplus B)$;
   d) $\big((A|\bar{B}) \downarrow B\big) \vee \big(C \oplus (B \to C)\big)$;
   e) $\big(A \leftrightarrow (B \oplus C)\big) \to \big(A\&\overline{\bar{A} \vee B \vee \bar{C}}\big)$.
2. Доказать выполнимость формул:
   a) $\overline{A \to \bar{A}}$;
   b) $(A \to B\&C)\&\overline{B \vee C \to A}$;
   c) $\big(A \oplus (B \downarrow C)\big) \to \overline{C \vee \overline{B \vee \bar{A}}}$;
   d) $(A \to B) \to \Big(B \to \big(A \to (B \leftrightarrow A)\big)\Big)$.
3. Доказать тождественную истинность формул:
   a) $A \to \big(B \to (A\&B)\big)$;
   b) $(A \to B) \to \big((A \to \bar{B}) \to \bar{A}\big)$;
   c) $A\&B \to (B\&\bar{A} \to B\&C)$;
   d) $(B \to C) \to (A \vee B \to A \vee C)$;
   e) $(A \to C) \to \big((B \to C) \to (A \vee B \to C)\big)$.
4. Доказать равносильность формул:
   a) $\overline{A \to B} \equiv A\&\bar{B}$;
   b) $A\&(A \vee C)\&(B \vee C) \equiv (A\&B) \vee (A\&C)$;
   c) $(A\&B) \vee \big((A \vee B)\&(\bar{A} \vee \bar{B})\big) \equiv A \vee B$;

d) $\overline{(A \leftrightarrow B) \to (A \to \bar{C})} \vee (A \oplus \bar{B} \& C) \equiv A \leftrightarrow (C \to B)$;

e) $(A|\bar{B}) \to \big((B \downarrow \bar{C}) \to (A \oplus C)\big) \equiv A \& (B \& C) \oplus (\bar{A} \to B)$.

5. С помощью равносильных преобразований упростить формулы:

a) $(A \leftrightarrow B) \& (A \vee B)$;

b) $(\overline{A \vee B} \to A \vee B) \& B$;

c) $\big((A \to B) \to \bar{A}\big) \to \big(A \to (A \& B)\big)$;

d) $(A \to B) \& (B \to C) \to (C \to A)$;

e) $\big((A \leftrightarrow B) \& (B \to \bar{A})\big) \to B$;

f) $\big(A \& \overline{A \& \bar{A}} \to B \& \bar{B} \to C\big) \vee A \vee (B \& C)$;

g) $\Big(\big((A \vee B) \& (A \vee \bar{C})\big) \to \big(\bar{A} \to (B \& \bar{C})\big)\Big) \& B$.

6. Привести к ДНФ и КНФ формулы:

a) $(\bar{A} \to C) \to \bar{B}$;

b) $(\bar{A} \vee \bar{B} \vee \bar{C}) \& (A \& B \vee C)$;

c) $(\bar{A} \& \bar{B}) \vee (A \leftrightarrow C)$;

d) $\big((A \to B) \oplus \bar{A}\big) \to \big(A \to (A \& C)\big)$;

e) $(A \leftrightarrow B) \downarrow \overline{C|(\bar{B} \to A \& C)}$;

f) $\Big(\big((A \to B) \to \bar{A}\big) \to \bar{B}\Big) \to \bar{C}$.

7. Найти СДНФ и СКНФ для формул:

a) $\overline{A \& B \to \bar{A}} \& \overline{A \& B \to \bar{B}}$;

b) $A \& B \vee \bar{C}$;

c) $A \vee \Big(B \to \big(C \leftrightarrow (A \& B)\big)\Big)$;

d) $\bar{B} \& \big(C \to (A \leftrightarrow B)\big)$;

e) $(A \to B) \oplus (A|B \& C)$;

f) $(A \vee \bar{B} \to A \& C) \to \overline{A \to \bar{A}} \vee B \& \bar{C}$;

g) $A_1 \vee A_2 \vee \ldots \vee A_n \to B_1 \vee B_2 \vee \ldots \vee B_n$.

# Глава 2
# Функции алгебры логики

## 2.1 Булевы функции. Способы задания

Булевы функции получили своё название по имени английского математика Джорджа Буля (1815—1864), который первым начал применять математические методы в логике.

Как известно, значение формулы логики высказываний полностью зависит от значений входящих в эту формулу высказывательных переменных. Поэтому формула логики высказываний является функцией от входящих в неё элементарных высказываний.

**Определение 2.1.1.** ***Булевой переменной*** *называется переменная, которая принимает значение из множества* $\{0,1\}$.

Булевые переменные будем обозначать малыми буквами латинского алфавита с индексами и без них: $x, y, z, u, v, \dots, x_1, x_2, \dots, x_n, \dots$ Ясно, что высказывание можно рассматривать как частный случай булевой переменной.

**Определение 2.1.2.** *Функция* $f(x_1, x_2, \dots, x_n)$*, зависящая от булевых переменных* $x_1, x_2, \dots, x_n$ *и принимающая значение из множества* $\{0,1\}$*, называется* ***функцией алгебры логики*** *(или* ***булевой функцией****).*

Другими словами, булева функция $f(x_1, x_2, \dots, x_n)$ сопоставляет всякому упорядоченному набору $(\alpha_1, \alpha_2, \dots, \alpha_n)$, состоящему из элементов 0 и 1, единственное значение, то есть

$$f: \underbrace{\{0,1\} \times \{0,1\} \times \dots \times \{0,1\}}_{n} \to \{0,1\}.$$

Вполне понятно, что существует $2^n$ различных двоичных наборов, при каждом из которых функция $f(x_1, x_2, \dots, x_n)$ принимает одно из двух значений: 0 или 1.

При $n = 0$ мы получаем две булевы константы: 0 и 1 (рассматриваемые как булевы функции от 0 переменных).

Очевидно, что тождественно истинные и тождественно ложные формулы алгебры логики представляют собой постоянные функции, а две равносильные формулы выражают одну и ту же функцию.

Обозначим через $P_2$ множество всех булевых функций, а через $P_2(n)$ – множество всех булевых функций от $n$ переменных (для фиксированного $n$). Ясно, что

$$P_2 = \bigcup_{n \geq 0} P_2(n).$$

Имея в виду тот факт, что равносильным формулам логики высказываний соответствует одна и та же булева функция, истинностные функции характеризуют классы равносильных формул. Следовательно, по Теореме 1.7.2 мощность множества $P_2(n)$ равна $2^{2^n}$.

Булеву функцию можно задать различными способами. Мы же остановимся на трёх основных способах.

***Задание функции таблицей истинности***

Для задания функции $f(x_1, x_2, \dots, x_n)$ достаточно указать какое значение функции соответствует каждому из наборов значений переменных $x_1, x_2, \dots, x_n$, то есть записать таблицу истинности, подобную указанной в Таблице 2.1.1.

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

Таблица 2.1.1

***Задание функции формулами***

Задание функции с помощью таблицы истинности зачастую неудобно, так как число строк в таблице экспоненциально зависит от числа переменных функции. Также при анализе свойств функции нельзя выполнять какие-либо алгебраические преобразования для облегчения этого процесса. В связи с этим удобно задавать функцию формулами.

Предположим, что имеется некоторое непустое (необязательно конечное) множество $F$ булевых функций. Используя индукцию, определим понятие формулы над множеством $F$:

1) каждая функция $f(x_1, x_2, \dots, x_n)$ из $F$ есть формула над $F$;
2) пусть $f(x_1, x_2, \dots, x_n)$ – функция из $F$ и $A_1, A_2, \dots, A_n$ – выражения, являющиеся либо формулами над $F$, либо переменными (необязательно различными). Тогда выражение $f(A_1, A_2, \dots, A_n)$ также является формулой над $F$.

Каждой формуле можно однозначно сопоставить функцию.

Пусть, например, $F$ есть множество булевых функций, состоящее из трёх функций: $f_1(x_1, x_2) = x_1 \& x_2$, $f_2(x_1, x_2) = x_1 \to x_2$ и $f_3(x_1, x_2) = x_1 \oplus x_2$. Тогда следующие выражения будут являться формулами над $F$:

$$x \to y, \big((x \oplus z) \to y\big) \& z, \big((x_1 \to x_3) \& (x_3 \to x_2)\big) \oplus (x_2 \& x_1).$$

***Задание функции вектором значений***

Условимся, что в случае $n$ переменных порядок перебора их значений будет совпадать с двоичным представлением чисел 0, 1, … , $n-1$. Следовательно, можно говорить о нулевом наборе значений переменных, первом, втором, третьем и т. д. К примеру, в случае трёх переменных первым набором будет $(0,0,1)$. Так как множество значений аргументов упорядочено, функцию можно задать вектором значений. Например, вектор значений $f = (01001110)$ задает функцию трех аргументов, для которой $f(0,0,0) = 0$, $f(0,0,1) = 1$, $f(0,1,0) = 0$ и т. д.

В теории булевых функций важной является не только зависимость функции от переменных, но и так называемая существенная зависимость функции от переменной.

**Определение 2.1.3.** ***Соседними наборами по i-ой переменной*** *называются наборы*

$$\tilde{\alpha} = (\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) \text{ и } \tilde{\beta} = (\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n),$$

*различающиеся только в i-ой компоненте.*

**Определение 2.1.4.** *Функция $f(x_1, x_2, \dots, x_n)$* ***существенно*** *зависит от переменной $x_i$, если существуют пара наборов $\tilde{\alpha}$ и $\tilde{\beta}$, соседних по i-ой переменной, таких, что $f(\tilde{\alpha}) \neq f(\tilde{\beta})$.*

**Определение 2.1.5.** *Если функция $f(x_1, x_2, \dots, x_n)$ существенно зависит от переменной $x_i$, то переменная $x_i$ называется* ***существенной*** *переменной функции $f(x_1, x_2, \dots, x_n)$. В противном случае переменная $x_i$ называется* ***фиктивной*** *переменной, а функция $f(x_1, x_2, \dots, x_n)$* ***не существенно*** *зависит от переменной $x_i$.*

Пусть для функции $f(x_1, x_2, \dots, x_n)$ переменная $x_i$ является фиктивной. Возьмём таблицу истинности, задающую функцию $f$ и вычеркнем из

неё все строки вида $\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n, f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n)$, а также столбец, соответствующий переменной $x_i$. Полученная таблица будет определять некоторую булеву функцию $g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ от $n-1$ переменных. Говорят, что функция $g$ получена из функции $f$ *путём удаления фиктивной переменной* $x_i$, а функция $f$ получается из $g$ *путём введения фиктивной* переменной $x_i$.

Понятие фиктивной переменной позволяет ввести понятие равенства булевых функций.

**Определение 2.1.6.** *Функции $f$ и $g$ называются* ***равными****, если они существенно зависят от одних и тех же переменных, и значения функций $f$ и $g$ совпадают на любом наборе значений существенных переменных.*

Например, функции $f(x_1, x_2)$ и $g(x_1)$, заданные Таблицами 2.1.2 и 2.1.3, являются равными, так как $f$ не существенно зависит от переменной $x_2$ и значения функций $f$ и $g$ совпадают при любом значении существенной переменной $x_1$. Путём удаления фиктивной переменной $x_2$ из функции $f$, получим функцию $g$.

| $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Таблица 2.1.2

| $x_1$ | $g(x_1)$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

Таблица 2.1.3

**Пример 2.1.1.** Перечислить все существенные и фиктивные переменные функции $f(x_1, x_2, x_3) = (01011010)$. Выразить $f(x_1, x_2, x_3)$ формулой, содержащей только существенные переменные.

*Решение*. Рассмотрим таблицу истинности функции $f$ (Таблица 2.1.4).

Видим, что переменная $x_1$ является существенной для данной функции, так как, например, для наборов $(0,0,0)$ и $(1,0,0)$, являющихся соседними по переменной $x_1$, имеем $f(0,0,0) \neq f(1,0,0)$.

Переменная $x_3$ также является существенной для функции $f$, так как для наборов $(0,0,0)$ и $(0,0,1)$, являющихся соседними по переменной $x_3$, имеем $f(0,0,0) \neq f(0,0,1)$.

В свою очередь переменная $x_2$ является фиктивной для функции $f$, потому что на всех наборах, соседних по переменной $x_2$, значения функции равны между собой, то есть выполняются равенства:

$$f(0,0,0) = f(0,1,0), f(0,0,1) = f(0,1,1),$$
$$f(1,0,0) = f(1,1,0), f(1,0,1) = f(1,1,1).$$

Выпишем таблицу функции $f$, как функцию только от существенных переменных (Таблица 2.1.5). Легко заметить, что $f(x_1, x_3) = x_1 \oplus x_3$.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Таблица 2.1.4

| $x_1$ | $x_3$ | $f(x_1, x_3)$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Таблица 2.1.5

При задании булевой функции $f(x_1, x_2, \dots, x_n)$ с помощью формулы для выяснения того, какие переменные у нее существенные (а какие фиктивные), иногда бывает удобно преобразовать исходное выражение к некоторому специальному виду, например к совершенной дизъюнктивной нормальной форме.

Заметим, что переменная $x_i$ является фиктивной тогда и только тогда, когда в СДНФ этой функции вместе с каждой элементарной конъюнкцией вида $x_i \& K$ содержится и элементарная конъюнкция $\bar{x}_i \& K$. Применим этот подход в следующем примере.

**Пример 2.1.2.** Пусть $f(x_1, x_2, x_3) = \big((x_1 \oplus x_2) \to x_3\big) \& \overline{x_3 \to x_2}$. Показать, что $x_1$ – фиктивная переменная функции $f(x_1, x_2, x_3)$. Выразить функцию $f$ формулой, не содержащей переменную $x_1$.

*Решение*. Сначала приведём функцию $f$ к ДНФ:

$$f(x_1, x_2, x_3) = \big((x_1 \oplus x_2) \to x_3\big) \& \overline{x_3 \to x_2} =$$

$$= \big(\overline{x_1 \oplus x_2} \vee x_3\big) \& \overline{\bar{x}_3 \vee x_2} = \big((x_1 \leftrightarrow x_2) \vee x_3\big) \& x_3 \& \bar{x}_2 =$$

$$= \big((x_1 \to x_2) \& (x_2 \to x_1) \vee x_3\big) \& x_3 \& \bar{x}_2 =$$

$$= \big((\bar{x}_1 \vee x_2) \& (\bar{x}_2 \vee x_1) \vee x_3\big) \& x_3 \& \bar{x}_2 =$$

$$= (\bar{x}_1 \& \bar{x}_2 \vee \bar{x}_1 \& x_1 \vee x_2 \& \bar{x}_2 \vee x_2 \& x_1 \vee x_3) \& x_3 \& \bar{x}_2 =$$

$$= \bar{x}_1 \& \bar{x}_2 \& x_3 \& \bar{x}_2 \vee x_2 \& x_1 \& x_3 \& \bar{x}_2 \vee x_3 \& x_3 \& \bar{x}_2 = \bar{x}_1 \& \bar{x}_2 \& x_3 \vee \bar{x}_2 \& x_3.$$

Приведём полученную ДНФ к СДНФ:

$$\bar{x}_1 \& \bar{x}_2 \& x_3 \vee \bar{x}_2 \& x_3 = \bar{x}_1 \& \bar{x}_2 \& x_3 \vee (x_1 \vee \bar{x}_1) \& \bar{x}_2 \& x_3 =$$

$$= \bar{x}_1 \& \bar{x}_2 \& x_3 \vee x_1 \& \bar{x}_2 \& x_3 \vee \bar{x}_1 \& \bar{x}_2 \& x_3 = \bar{x}_1 \& \bar{x}_2 \& x_3 \vee x_1 \& \bar{x}_2 \& x_3.$$

Согласно вышеуказанному замечанию $x_1$ является фиктивной переменной. Теперь выразим функцию $f$ формулой, не содержащей переменную $x_1$:

$$f(x_1, x_2, x_3) = \bar{x}_1 \& \bar{x}_2 \& x_3 \vee x_1 \& \bar{x}_2 \& x_3 =$$
$$= (x_1 \vee \bar{x}_1) \& \bar{x}_2 \& x_3 = 1 \& \bar{x}_2 \& x_3 = \bar{x}_2 \& x_3.$$

## 2.2 Разложение функций по переменным

Пусть $x$ – булевая переменная. Введём обозначение:

$$x^\sigma = \begin{cases} \bar{x}, & \text{при } \sigma = 0, \\ x, & \text{при } \sigma = 1. \end{cases}$$

Рассмотрим разложение булевой функции $f(x_1, x_2, \dots, x_n)$ по $k$ переменным, где $1 \le k \le n$, называемое разложением Шеннона.

**Теорема 2.2.1. (Первая теорема Шеннона)** *Пусть даны функция $f(x_1, x_2, \dots, x_n)$ и число $k$, $1 \le k \le n$. Тогда функцию $f$ можно представить в виде:*

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{\text{по всем наборам} \\ (\sigma_1, \dots, \sigma_k)}} x_1^{\sigma_1} \& \dots \& x_k^{\sigma_k} \& f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n). \quad (2.1)$$

*Доказательство.* Возьмём произвольный набор значений переменных $(\alpha_1, \dots, \alpha_k)$ и покажем, что при этом наборе левая и правая части формулы (2.1) принимают одинаковые значения. Левая часть равенства (2.1) равна $f(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots, x_n)$. Так как $\alpha_i^{\sigma_i} = 1$ $(i = \overline{1,k})$ тогда и только тогда, когда $\alpha_i = \sigma_i$, то в правой части формулы (2.1) останется только один член, для которого все $\sigma_i$ совпадают с $\alpha_i$ (он имеет вид $f(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots, x_n)$), а остальные дизъюнктивные члены будут равны нулю.

Таким образом, мы показали, что для произвольного набора длины $k$ функции $f$, левая и правая части формулы (2.1) равны, что и требовалось доказать. □

Аналогично доказывается и следующая теорема.

**Теорема 2.2.2. (Вторая теорема Шеннона)** *Пусть даны функция $f(x_1, x_2, \dots, x_n)$ и число $k$, $1 \le k \le n$. Тогда функцию $f$ можно представить в виде:*

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{\substack{\text{по всем наборам} \\ (\sigma_1, \dots, \sigma_k)}} x_1^{1-\sigma_1} \vee \dots \vee x_k^{1-\sigma_k} \vee f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n).$$

Из Теоремы 2.2.1 вытекает Следствие 2.2.1.

**Следствие 2.2.1.** *Всякую булеву функцию* $f(x_1, x_2, \dots, x_n)$ *можно представить в виде:*

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{\text{по всем наборам} \\ (\sigma_1,\dots,\sigma_n)}} x_1^{\sigma_1} \& \dots \& x_n^{\sigma_n} \& f(\sigma_1, \dots, \sigma_n).$$

В случае, когда $f(x_1, x_2, \dots, x_n) \neq 0$, данное представление приобретает вид:

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{\text{по всем наборам} \\ (\sigma_1,\dots,\sigma_n), \\ \text{на которых} \\ f(\sigma_1,\dots,\sigma_n)=1}} x_1^{\sigma_1} \& \dots \& x_n^{\sigma_n} - \text{СДНФ}.$$

Из Теоремы 2.2.2 вытекает Следствие 2.2.2.

**Следствие 2.2.2** *Всякую булеву функции* $f(x_1, x_2, \dots, x_n)$ *можно представить в виде:*

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{\substack{\text{по всем наборам} \\ (\sigma_1,\dots,\sigma_n)}} x_1^{1-\sigma_1} \vee \dots \vee x_n^{1-\sigma_n} \vee f(\sigma_1, \dots, \sigma_n).$$

В случае, когда $f(x_1, x_2, \dots, x_n) \neq 1$, данное представление приобретает вид:

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{\substack{\text{по всем наборам} \\ (\sigma_1,\dots,\sigma_n), \\ \text{на которых} \\ f(\sigma_1,\dots,\sigma_n)=0}} x_1^{1-\sigma_1} \vee \dots \vee x_n^{1-\sigma_n} - \text{СКНФ}.$$

**Следствие 2.2.3.** *Всякую булеву функции* $f(x_1, x_2, \dots, x_n)$ *можно представить в виде формулы над множеством функций* $\{\&, \vee, \neg\}$.

**Пример 2.2.1.** Выполнить разложение функции

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

по переменным $x_1$ и $x_3$, применив первую теорему Шеннона.

*Решение*. Запишем

$$f(x_1, x_2, x_3) = \bigvee_{(\sigma_1,\sigma_3)} x_1^{\sigma_1} \& x_3^{\sigma_3} \& f(\sigma_1, x_2, \sigma_3) =$$

$$= x_1^0 \& x_3^0 \& f(0, x_2, 0) \vee x_1^0 \& x_3^1 \& f(0, x_2, 1) \vee$$
$$\vee x_1^1 \& x_3^0 \& f(1, x_2, 0) \vee x_1^1 \& x_3^1 \& f(1, x_2, 1) =$$
$$= \bar{x}_1 \& \bar{x}_3 \& \overline{(0 \longrightarrow x_2) \oplus 0} \vee \bar{x}_1 \& x_3 \& \overline{(0 \longrightarrow x_2) \oplus 1} \vee$$
$$\vee x_1 \& \bar{x}_3 \& \overline{(1 \longrightarrow x_2) \oplus 0} \vee x_1 \& x_3 \& \overline{(1 \longrightarrow x_2) \oplus 1} =$$
$$= \bar{x}_1 \& \bar{x}_3 \& \overline{1 \oplus 0} \vee \bar{x}_1 \& x_3 \& \overline{1 \oplus 1} \vee$$
$$\vee x_1 \& \bar{x}_3 \& \overline{(1 \longrightarrow x_2) \oplus 0} \vee x_1 \& x_3 \& \overline{(1 \longrightarrow x_2) \oplus 1} =$$
$$= \bar{x}_1 \& x_3 \vee x_1 \& \bar{x}_3 \& \overline{\overline{(1 \vee x_2)} \leftrightarrow 0} \vee x_1 \& x_3 \& \overline{\overline{(1 \vee x_2)} \leftrightarrow 1} =$$
$$= \bar{x}_1 \& x_3 \vee x_1 \& \bar{x}_3 \& \big((0 \vee x_2) \leftrightarrow 0\big) \vee x_1 \& x_3 \& \big((0 \vee x_2) \leftrightarrow 1\big) =$$
$$= \bar{x}_1 \& x_3 \vee x_1 \& \bar{x}_3 \& \bar{x}_2 \vee x_1 \& x_3 \& x_2.$$

## 2.3 Полином Жегалкина

Рассмотрим ещё одно представление функции в виде формулы заданного вида.

**Определение 2.3.1.** *Формула вида:*

$$\alpha_0 \oplus \alpha_1 \& x_1 \oplus \alpha_2 \& x_2 \oplus \ldots \oplus \alpha_n \& x_n \oplus \alpha_{12} \& x_1 \& x_2 \oplus \ldots$$
$$\ldots \oplus \alpha_{12\ldots n} \& x_1 \& x_2 \& \ldots \& x_n, \qquad (2.2)$$

*где $x_1, x_2, \ldots, x_n$ – логические переменные, $\alpha_1, \alpha_2, \ldots, \alpha_{12\ldots n}$ – логические константы, называется* ***полиномом*** *(или* ***многочленом) Жегалкина****.*

Например, формулы

$$1 \oplus x_1 \oplus x_1 \& x_2 \text{ и } x_2 \oplus x_2 \& x_3 \oplus x_1 \& x_3 \oplus x_1 \& x_2 \& x_3$$

являются полиномами Жегалкина.

**Теорема 2.3.1.** *Всякая булева функция может быть представлена полиномом Жегалкина, причём единственным образом.*

*Доказательство.* Для начала докажем существование такого представления для функции. Согласно Следствию 2.2.3 произвольная булева функция $f(x_1, x_2, \ldots, x_n)$ из $P_2$ представима в виде формулы $U$ над множеством булевых функций $\{\&, \vee, \neg\}$. А именно, если $f \neq 0$, то её можно представить в виде СДНФ, а если $f = 0$, то её можно представить как $f = x \& \bar{x}$. Воспользуемся законом де Моргана $(x \vee y = \overline{\bar{x} \& \bar{y}})$ и заменим все дизъюнкции в формуле $U$ на конъюнкции и отрицание. Получим представление функции $f(x_1, x_2, \ldots, x_n)$ в виде формулы $U'$ над системой булевых функций $\{\&, \neg\}$. Теперь заменим в формуле $U'$ все отрицания на

сложение по модулю два согласно тождеству $\bar{x} = x \oplus 1$. Раскроем все скобки, пользуясь дистрибутивностью $\wedge$ относительно $\oplus$, и получим формулу вида (2.2).

Докажем теперь единственность такого представления. Так как любое слагаемое либо содержит, либо не содержит каждую из переменных $x_1, x_2, \dots, x_n$, то всего различных слагаемых из $n$ переменных будет ровно $2^n$. При этом вместо пустого полинома (не содержащего ни одной из переменных) мы берём константу 1. Исходя из того, что любой полином содержит либо не содержит каждое из $2^n$ слагаемых, то всего различных полиномов Жегалкина (включая пустой) будет $2^{2^n}$, а это как раз столько, сколько существует различных функций от $n$ переменных в $P_2$. Откуда следует, что если какая-нибудь функция представима в виде двух различных полиномов Жегалкина, то найдётся функция, которая не будет представима в виде полинома Жегалкина (так как каждый полином Жегалкина выражает ровно одну функцию), что невозможно. □

Верно следующее замечание.

**Замечание 2.3.1.** *Полином Жегалкина является формулой над* $\{0, 1, \&, \oplus\}$. *Константа* $0$ *требуется для задания тождественно нулевой функции. Для остальных случаев будет достаточно функций* $1, \&, \oplus$.

Существуют различные методы построения полинома Жегалкина для заданной логической функции. Опишем три из них.

***Метод неопределённых коэффициентов***

В общем виде полином Жегалкина для функции $f(x_1, x_2, \dots, x_n)$ задан формулой (2.2). Обозначим его через $P(\tilde{x}^n)$, где $\tilde{x}^n$ – запись $n$ переменных $x_1, x_2, \dots, x_n$.

Найдём неизвестные коэффициенты $\alpha_0, \alpha_1, \dots, \alpha_n, \alpha_{12}, \alpha_{13}, \dots, \alpha_{12\dots n}$. Для каждого набора значений булевых переменных $\tilde{\beta}$ длины $n$ составляем уравнение $P(\tilde{\beta}) = f(\tilde{\beta})$, соответствующее подстановке вместо $\tilde{x}^n$ набора $\tilde{\beta}$. В итоге получаем систему из $2^n$ линейных уравнений с $2^n$ неизвестными, которая имеет единственное решение. Решив систему, находим искомые коэффициенты.

**Пример 2.3.1.** Методом неопределённых коэффициентов построить полином Жегалкина для функции $f(\tilde{x}^3) = (01101101)$.

*Решение*. Полином Жегалкина функции от 3 переменных в общем виде задаётся формулой

$$P(\tilde{x}^3) = \alpha_0 \oplus \alpha_1 \& x_1 \oplus \alpha_2 \& x_2 \oplus \alpha_3 \& x_3 \oplus \alpha_{12} \& x_1 \& x_2 \oplus$$
$$\oplus \alpha_{13} \& x_1 \& x_3 \oplus \alpha_{23} \& x_2 \& x_3 \oplus \alpha_{123} \& x_1 \& x_2 \& x_3. \qquad (2.3)$$

По очереди подставляя все возможные наборы значений булевых переменных, составим систему уравнений для коэффициентов $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{123}$:

$$\begin{cases} f(0,0,0) = 0 = \alpha_0 \\ f(0,0,1) = 1 = \alpha_0 \oplus \alpha_3 \\ f(0,1,0) = 1 = \alpha_0 \oplus \alpha_2 \\ f(0,1,1) = 0 = \alpha_0 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_{23} \\ f(1,0,0) = 1 = \alpha_0 \oplus \alpha_1 \\ f(1,0,1) = 1 = \alpha_0 \oplus \alpha_1 \oplus \alpha_3 \oplus \alpha_{13} \\ f(1,1,0) = 0 = \alpha_0 \oplus \alpha_1 \oplus \alpha_2 \oplus \alpha_{12} \\ f(1,1,1) = 1 = \alpha_0 \oplus \alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_{12} \oplus \alpha_{13} \oplus \alpha_{23} \oplus \alpha_{123} \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} \alpha_0 = 0 \\ 0 \oplus \alpha_3 = 1 \Longrightarrow \alpha_3 = 1 \\ 0 \oplus \alpha_2 = 1 \Longrightarrow \alpha_2 = 1 \\ 0 \oplus 1 \oplus 1 \oplus \alpha_{23} = 0 \Longrightarrow \alpha_{23} = 0 \\ 0 \oplus \alpha_1 = 1 \Longrightarrow \alpha_1 = 1 \\ 0 \oplus 1 \oplus 1 \oplus \alpha_{13} = 1 \Longrightarrow \alpha_{13} = 1 \\ 0 \oplus 1 \oplus 1 \oplus \alpha_{12} = 0 \Longrightarrow \alpha_{12} = 0 \\ 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus \alpha_{123} = 1 \Longrightarrow \alpha_{123} = 1 \end{cases} \Leftrightarrow \begin{cases} \alpha_0 = 0 \\ \alpha_3 = 1 \\ \alpha_2 = 1 \\ \alpha_{23} = 0 \\ \alpha_1 = 1 \\ \alpha_{13} = 1 \\ \alpha_{12} = 0 \\ \alpha_{123} = 1 \end{cases}$$

Подставляя найденные коэффициенты в $P(\tilde{x}^3)$, получаем полином Жегалкина заданной функции

$$f(\tilde{x}^3) = x_1 \oplus x_2 \oplus x_3 \oplus x_1 \& x_3 \oplus x_1 \& x_2 \& x_3.$$

***Метод треугольника (метод треугольника Паскаля)***

Метод треугольника позволяет преобразовать таблицу истинности в полином Жегалкина путём построения вспомогательной треугольной таблицы в соответствии со следующими правилами:

1. Строится таблица истинности для заданной функции, в которой строки идут в порядке возрастания двоичных наборов от 000...00 до 111...11.
2. Строится вспомогательная треугольная таблица, первый столбец которой совпадает со столбцом значений функции в таблице истинности.
3. Каждая ячейка в каждом последующем столбце треугольной таблицы получается путём сложения по модулю два значений двух ячеек предыдущего столбца, стоящих в той же строке и в нижестоящей строке. В результате каждый последующий столбец содержит на одну ячейку меньше, чем предыдущий столбец.

4. Столбцы вспомогательной таблицы нумеруются двоичными наборами в том же порядке, что и строки таблицы истинности. Каждому двоичному набору ставится в соответствие один из коэффициентов полинома Жегалкина с теми индексами, на позиции которых стоят единицы. Например, ячейке со значением 111 ставится в соответствие коэффициент $\alpha_{123}$, ячейке со значением 011 соответствует коэффициент $\alpha_{23}$, ячейке со значением 100 – коэффициент $\alpha_1$, а ячейке со значением 000 соответствует коэффициент $\alpha_0$ (индекс 0 появляется в случае, когда в двоичном наборе не встречается ни одной единицы) и т. д.
5. Ячейки верхней строки треугольной таблицы содержат значения соответствующих коэффициентов полинома Жегалкина.

В отличие от метода неопределённых коэффициентов, метод треугольника удобнее тем, что расчёты занимают меньше места и в них сложнее ошибиться.

**Пример 2.3.2.** Методом треугольника построить полином Жегалкина для функции $f(\tilde{x}^3) = (10010100)$.

*Решение*. Запишем таблицу истинности функции $f(\tilde{x}^3)$ и на её основе построим вспомогательную треугольную таблицу (Таблица 2.3.1).

| $x_1$ | $x_2$ | $x_3$ | $f(\tilde{x}^3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|
| $\alpha_0$ | $\alpha_3$ | $\alpha_2$ | $\alpha_{23}$ | $\alpha_1$ | $\alpha_{13}$ | $\alpha_{12}$ | $\alpha_{123}$ |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 0 | 1 | | |
| 1 | 1 | 0 | 0 | 1 | | | |
| 0 | 1 | 0 | 1 | | | | |
| 1 | 1 | 1 | | | | | |
| 0 | 0 | | | | | | |
| 0 | | | | | | | |

Таблица 2.3.1

Подставляя найденные коэффициенты в (2.3), получаем полином Жегалкина заданной функции:

$$f(\tilde{x}^3) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_1 \& x_2 \oplus x_1 \& x_2 \& x_3.$$

***Метод преобразования формул***

Сначала формулу, реализующую функцию $f$, преобразуем в формулу над множеством связок $\{\&, \neg\}$. Это можно сделать путём представления

функции $f$ в виде СДНФ или СКНФ и последующей замены всех дизъюнкции на конъюнкции с помощью закона де Моргана ($x \vee y = \overline{\bar{x}\&\bar{y}}$).

Затем следует заменить всюду подформулы вида $\bar{x}$ на $x \oplus 1$, раскрыть скобки, пользуясь дистрибутивным законом $x\&(y \oplus z) = x\&y \oplus x\&z$, и применить эквивалентности $x\&x = x$, $x\&1 = x$, $x \oplus x = 0$ и $x \oplus 0 = x$.

**Пример 2.3.3.** Методом преобразования формул построить полином Жегалкина для функции $f(\tilde{x}^3) = (x_1 \downarrow x_2)|(x_2 \downarrow x_3)$.

*Решение*. Запишем:

$$f(\tilde{x}^3) = (x_1 \downarrow x_2)|(x_2 \downarrow x_3) = \overline{\overline{x_1 \vee x_2}\&\overline{x_2 \vee x_3}} = \overline{\bar{x}_1\&\bar{x}_2\&\bar{x}_3} =$$
$$= (x_1 \oplus 1)\&(x_2 \oplus 1)\&(x_3 \oplus 1) \oplus 1 =$$
$$= x_1\&x_2\&x_3 \oplus x_1\&x_2 \oplus x_1\&x_3 \oplus x_2\&x_3 \oplus x_1 \oplus x_2 \oplus x_3.$$

## 2.4 Замкнутость и полнота

Пусть $F$ – некоторая система функций из $P_2$. Нас интересуют следующие два вопроса: всякая ли логическая функция может быть представлена формулой над $F$? И если нет, то какие логические функции могут быть представлены формулами над $F$?

Вначале дадим определение замыканию.

**Определение 2.4.1.** ***Замыканием*** *множества $F$ называется множество всех функций из $P_2$, представимых в виде формул над множеством $F$. Замыкание множества $F$ обозначается через $[F]$.*

Например, для множества $F = \{x_1 \oplus x_2\}$ замыкание $[F]$ содержит суммы по модулю два любого числа переменных (в том числе константу 0) и сами переменные, так как $x \oplus x = 0$ и $x \oplus x \oplus x = x$.

Пусть $F$ и $G$ – некоторые системы функций из $P_2$. Отметим пять простых свойств замыкания:

1) $F \subseteq [F]$;
2) $\big[[F]\big] = [F]$;
3) если $F \subseteq G$, то $[F] \subseteq [G]$;
4) $[F \cap G] \subseteq [F] \cap [G]$;
5) $[F] \cup [G] \subseteq [F \cup G]$.

**Определение 2.4.2.** *Класс (система функций) $F$ называется (**функционально) замкнутым**, если $[F] = F$.*

Например, класс $F = P_2$ является функционально замкнутым, так как $[F] = P_2$. А класс $G = \{x_1 \oplus x_2, 1\}$ не замкнут, потому что $[G]$ содержит константу $0 = 1 \oplus 1$.

**Определение 2.4.3.** *Система функций $F \subseteq P_2$ называется (**функционально) полной**, если любую функцию из $P_2$ можно представить в виде формулы над $F$, то есть $[F] = P_2$.*

Например, система функций $P_2$ является функционально полной. Также согласно Следствию 2.2.3, $\{x_1 \& x_2, x_1 \vee x_2, \bar{x}\}$ – полная система функций. В свою очередь, очевидно, что система $\{1, \bar{x}\}$ не полна.

Следующая теорема позволяет доказывать полноту системы функций, показывая, что она сводится к уже известной полной системе.

**Теорема 2.4.1. (Теорема сведения)** *Пусть даны две системы функций $F = \{f_1, f_2, \dots\}$ и $G = \{g_1, g_2, \dots\}$ из $P_2$ такие, что $F$ полна и каждая её функция $f_i \in F$ представима в виде формулы над $G$. Тогда система $G$ является полной.*

*Доказательство.* По условию теоремы $[F] = P_2$ и $F \subseteq [G]$. Тогда в силу свойства 3) операции замыкания, $P_2 = [F] \subseteq \big[[G]\big]$. Теперь, применив свойство 2), получим $P_2 \subseteq [G]$. С другой стороны, понятно, что $P_2 \supseteq [G]$. Следовательно, имеет место равенство $P_2 = [G]$, а это в свою очередь означает, что система $G$ является полной. □

Опираясь на Теорему 2.4.1, установим полноту ряда систем функций.

**Теорема 2.4.2.** *Следующие системы функций являются функционально полными:*

1) $\{x_1 \& x_2, \bar{x}\}$;
2) $\{x_1 \vee x_2, \bar{x}\}$;
3) $\{x_1 | x_2\}$;
4) $\{1, x_1 \& x_2, x_1 \oplus x_2\}$.

*Доказательство.* Покажем, что каждая из систем является функционально полной.

1) Как известно, система функций $F = \{x_1 \& x_2, x_1 \vee x_2, \bar{x}\}$ является функционально полной. Пусть $G = \{x_1 \& x_2, \bar{x}\}$. Опираясь на закон де Моргана, получим тождество

$$x_1 \vee x_2 = \overline{\bar{x}_1 \& \bar{x}_2}.$$

Следовательно, $F \subseteq [G]$ и по Теореме 2.4.1 система $G$ является полной.

2) Доказывается аналогично пункту 1).

3) Пусть $H = \{x_1 | x_2\}$. Легко убедиться в том, что

$$\bar{x} = x | x,$$
$$x_1 \& x_2 = \overline{x_1 | x_2} = (x_1 | x_2) | (x_1 | x_2).$$

В результате получим $G \subseteq [H]$.

4) В соответствии с Теоремой 2.3.1, $\{1, x_1 \& x_2, x_1 \oplus x_2\}$ – полная система функций. □

**Пример 2.4.1.** Сведением к заведомо полной системе показать, что множество $F = \{x_1 \to x_2, \overline{x_1 \oplus x_2 \oplus x_3}\}$ является полной системой функций.

*Решение*. Сведём $F$ к полной системе функций $\{x_1 \vee x_2, \bar{x}\}$.

Легко видеть, что

$$\bar{x} = \overline{x \oplus x \oplus x},$$

$$x_1 \vee x_2 = \bar{x}_1 \to x_2 = \overline{x_1 \oplus x_1 \oplus x_1} \to x_2.$$

## 2.5 Важнейшие замкнутые классы

Перед тем как начать рассмотрение важнейших замкнутых классов, введём понятие суперпозиции.

**Определение 2.5.1.** *Пусть функции* $f_1(x_1, x_2, \dots, x_{n_1}), f_2(x_1, x_2, \dots, x_{n_2}), \dots, f_m(x_1, x_2, \dots, x_{n_m})$ *принадлежат* $P_2$. ***Суперпозицией*** *функций* $f_1, f_2, \dots, f_m$ *называется операция, состоящая в последовательном применении конечного количества раз следующих двух операций:*

*1) переименования некоторой переменной* $x_j$ *функции* $f_i$, $i \in \{1, \dots, m\}, j \in \{1, \dots, n_i\}$:

$$f_i(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_{n_i}),$$

*где* $y$ *может совпадать с любой переменной;*

*2) подстановки некоторой функции* $f_k$ *вместо переменной* $x_j$ *функции* $f_i$, $k, i \in \{1, \dots, m\}, j \in \{1, \dots, n_i\}$:

$$f_i(x_1, \dots, x_{j-1}, f_k(x_1, x_2, \dots, x_{n_k}), x_{j+1}, \dots, x_{n_i}).$$

Пусть $F$ – некоторая система функций из $P_2$. По сути, утверждение, что $f$ является суперпозицией функций из $F$, эквивалентно утверждению, что $f$ представима в виде формулы над $F$. Соответственно, замыканием $F$ называется множество всех функций, являющихся суперпозицией функций из $F$.

***Классы $T_0$ и $T_1$***

**Определение 2.5.2.** *Говорят, что функция* $f(x_1, x_2, \dots, x_n) \in P_2$ ***сохраняет константу*** **0**, *если* $f(0,0,\dots,0) = 0$. *Множество всех функций из* $P_2$, *сохраняющих константу* 0, *обозначается через* $T_0$.

К примеру, классу $T_0$ принадлежат функции: $0$, $x$, $x_1 \& x_2$, $x_1 \oplus x_2$ и не принадлежат функции $1$, $\bar{x}$, $x_1 \to x_2$, $x_1 | x_2$. Таблица истинности функции

класса $T_0$ характеризуется тем, что в первой строке столбца значений функции содержится 0.

**Теорема 2.5.1.** *Класс $T_0$ замкнут.*

*Доказательство.* Рассмотрим суперпозицию функций из $T_0$.

1) Пусть $f(x_1, x_2, \dots, x_n) \in T_0$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y) = f(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n).$$

Очевидно, что $g(0,0,\dots,0) = f(0,0,\dots,0) = 0$. Следовательно, $g \in T_0$.

2) Пусть функции $f(x_1, x_2, \dots, x_n)$ и $h(y_1, y_2, \dots, y_m)$ принадлежат классу $T_0$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) =$$
$$= f(x_1, \dots, x_{j-1}, h(y_1, y_2, \dots, y_m), x_{j+1}, \dots, x_n).$$

Тогда $g(0,0,\dots,0) = f(0,\dots,0,h(0,0,\dots,0),0,\dots,0) = 0$, откуда следует, что $g \in T_0$.

Таким образом, $[T_0] = T_0$. □

**Определение 2.5.3.** *Говорят, что функция $f(x_1, x_2, \dots, x_n) \in P_2$* ***сохраняет константу 1****, если $f(1,1,\dots,1) = 1$. Множество всех функций из $P_2$, сохраняющих константу 1, обозначается через $T_1$.*

Например, функции $1$, $x$, $x_1 \vee x_2$ и $x_1 \leftrightarrow x_2$ принадлежат классу $T_1$, а функции $0$, $\bar{x}$ и $x_1 \downarrow x_2$ не принадлежат классу $T_1$.

**Теорема 2.5.2.** *Класс $T_1$ замкнут.*

Доказательство Теоремы 2.5.2 производится аналогично доказательству Теоремы 2.5.1.

Обозначим через $T_0(n)$ и $T_1(n)$ множество всех функций из $T_0$ и, соответственно, из $T_1$ от $n$ переменных.

**Теорема 2.5.3.** $|T_0(n)| = |T_1(n)| = 2^{2^n-1}$.

*Доказательство.* При задании функции $f(\tilde{x}^n)$ из $T_0$ с помощью таблицы истинности значения функции $f$ можно произвольно выбирать на всех двоичных наборах, отличных от набора $(0,0,\dots,0)$, на котором $f$ принимает значение 0. Таких наборов всего $2^n - 1$, откуда следует, что выбор может быть осуществлён $2^{2^n-1}$ способами, то есть $|T_0(n)| = 2^{2^n-1}$. По аналогии, $|T_1(n)| = 2^{2^n-1}$. □

***Класс S***

**Определение 2.5.4.** *Функция $f^*(x_1, x_2, \dots, x_n) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ называется* ***двойственной*** *к функции $f(x_1, x_2, \dots, x_n) \in P_2$.*

**Определение 2.5.5.** *Функция* $f(x_1, x_2, \dots, x_n) \in P_2$ *называется* ***самодвойственной****, если* $f(x_1, x_2, \dots, x_n) = f^*(x_1, x_2, \dots, x_n)$. *Множество всех* ***самодвойственных*** *функций из* $P_2$ *обозначается через* $S$.

Из этого определения вытекает, что для самодвойственных функций $f(x_1, x_2, \dots, x_n)$ имеет место тождество $\bar{f}(x_1, x_2, \dots, x_n) = f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$, которое означает, что она принимает противоположные значения на любой паре противоположных наборов $(\alpha_1, \alpha_2, \dots, \alpha_n)$ и $(\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n)$.

Примерами самодвойственных функций могут служить $x$ и $\bar{x}$, в то время как функция $f = x_1 \& x_2$ не принадлежит классу $S$, потому что $f^* = \overline{\bar{x}_1 \& \bar{x}_2} = x_1 \vee x_2$.

**Теорема 2.5.4.** *Класс S замкнут.*

*Доказательство.* Рассмотрим суперпозицию функций из $S$.

1) Пусть $f(x_1, x_2, \dots, x_n) \in S$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y) = f(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n).$$

Тогда

$$\bar{g}(\bar{x}_1, \dots, \bar{x}_{j-1}, \bar{x}_{j+1}, \dots, \bar{x}_n, \bar{y}) = \bar{f}(\bar{x}_1, \dots, \bar{x}_{j-1}, \bar{y}, \bar{x}_{j+1}, \dots, \bar{x}_n) =$$
$$= f^*(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n) = f(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n) =$$
$$= g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y).$$

Исходя из вышесказанного, $g \in S$.

2) Пусть функции $f(x_1, x_2, \dots, x_n)$ и $h(y_1, y_2, \dots, y_m)$ принадлежат классу $S$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) =$$
$$= f(x_1, \dots, x_{j-1}, h(y_1, y_2, \dots, y_m), x_{j+1}, \dots, x_n).$$

В этом случае

$$\bar{g}(\bar{x}_1, \dots, \bar{x}_{j-1}, \bar{x}_{j+1}, \dots, \bar{x}_n, \bar{y}_1, \dots, \bar{y}_m) =$$
$$= \bar{f}(\bar{x}_1, \dots, \bar{x}_{j-1}, h(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_m), \bar{x}_{j+1}, \dots, \bar{x}_n) =$$
$$= \bar{f}(\bar{x}_1, \dots, \bar{x}_{j-1}, \bar{\bar{h}}(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_m), \bar{x}_{j+1}, \dots, \bar{x}_n) =$$
$$= \bar{f}(\bar{x}_1, \dots, \bar{x}_{j-1}, \overline{h^*}(y_1, y_2, \dots, y_m), \bar{x}_{j+1}, \dots, \bar{x}_n) =$$
$$= f^*(x_1, \dots, x_{j-1}, h^*(y_1, y_2, \dots, y_m), x_{j+1}, \dots, x_n) =$$
$$= f(x_1, \dots, x_{j-1}, h(y_1, y_2, \dots, y_m), x_{j+1}, \dots, x_n) =$$
$$= g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m).$$

В итоге $g(x_1, \dots, x_{j-1}, x_{j+1} \dots, x_n, y_1, \dots, y_m) \in S$ и $[S] = S$. □

Обозначим через $S(n)$ множество всех функций из $S$ от $n$ переменных.

**Теорема 2.5.5.** $|S(n)| = 2^{2^{n-1}}$.

*Доказательство.* Рассмотрим самодвойственную функцию от $n$ переменных, заданную таблицей истинности. Если на первом наборе, то есть на наборе $(0,0,\dots,0)$, эта функция принимает значение $\alpha$, то согласно определению самодвойственной функции, на последнем наборе (противоположном первому) она принимает значение $\bar{\alpha}$. То же самое можно сказать о втором и предпоследнем наборах и так далее. Из чего можно заключить, что каждая самодвойственная функция полностью определяется верхней половиной своего столбца значений из таблицы истинности, то есть двоичным набором длины $2^n/2 = 2^{n-1}$. Таким образом, число различных самодвойственных функций от $n$ переменных равно $2^{2^{n-1}}$. □

**Пример 2.5.1.** Определить является ли самодвойственной функция

$$f(x_1, x_2, x_3) = x_1 \& x_2 \vee x_2 \& \bar{x}_3 \vee x_1 \& \bar{x}_3.$$

*Решение*. Воспользуемся двумя способами.

Запишем таблицу истинности функции $f(x_1, x_2, x_3)$ и убедимся, что на всех противоположных наборах значений переменных функция $f$ принимает противоположные значения (Таблица 2.5.1). Откуда следует, что $f \in S$.

| $x_1$ | $x_2$ | $x_3$ | $x_1 \& x_2$ | $x_2 \& \bar{x}_3$ | $x_1 \& \bar{x}_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Таблица 2.5.1

Приведём функции $f$ и $f^*$ к СДНФ и сравним их:

$$f^*(x_1, x_2, x_3) = \bar{f}(\bar{x}_1, \bar{x}_2, \bar{x}_3) = \overline{\bar{x}_1 \& \bar{x}_2 \vee \bar{x}_2 \& \bar{\bar{x}}_3 \vee \bar{x}_1 \& \bar{\bar{x}}_3} =$$

$$= \overline{\bar{x}_1 \& \bar{x}_2} \& \overline{\bar{x}_2 \& x_3} \& \overline{\bar{x}_1 \& x_3} = (x_1 \vee x_2) \& (x_2 \vee \bar{x}_3) \& (x_1 \vee \bar{x}_3) =$$

$$= x_1 \& x_2 \vee x_1 \& x_2 \& \bar{x}_3 \vee x_1 \& \bar{x}_3 \vee x_2 \& \bar{x}_3 =$$

$$= x_1 \& x_2 \& x_3 \vee x_1 \& x_2 \& \bar{x}_3 \vee x_1 \& \bar{x}_2 \& \bar{x}_3 \vee \bar{x}_1 \& x_2 \& \bar{x}_3.$$

$$f(x_1, x_2, x_3) = x_1 \& x_2 \vee x_2 \& \bar{x}_3 \vee x_1 \& \bar{x}_3 =$$

$$= x_1 \& x_2 \& x_3 \vee x_1 \& x_2 \& \bar{x}_3 \vee \bar{x}_1 \& x_2 \& \bar{x}_3 \vee x_1 \& \bar{x}_2 \& \bar{x}_3.$$

Как мы можем видеть, $f(x_1, x_2, x_3) = f^*(x_1, x_2, x_3)$. Из чего следует, что $f \in S$.

***Класс M***

Определим правило сравнения двоичных наборов одинаковой длины.

**Определение 2.5.6.** *Говорят, что набор* $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ ***предшествует*** *набору* $\tilde{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$ *и пишут* $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, *если* $\alpha_1 \le \beta_1$, $\alpha_2 \le \beta_2, \dots, \alpha_n \le \beta_n$.

**Определение 2.5.7.** *Наборы* $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ *и* $\tilde{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$ *называются* ***сравнимыми****, если либо* $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, *либо* $\tilde{\beta} \preccurlyeq \tilde{\alpha}$. *В случае, когда ни одно из этих отношений не выполняется, наборы* $\tilde{\alpha}$ *и* $\tilde{\beta}$ *называются* ***несравнимыми***.

К примеру, набор $(1,0,0,1)$ предшествует набору $(1,0,1,1)$, а наборы $(1,0,1)$ и $(0,1,1)$ несравнимы. Теперь мы можем упорядочить некоторые наборы:

$$(0,0,\dots,0) \preccurlyeq \cdots \preccurlyeq (\alpha_1, \alpha_2, \dots, \alpha_n) \preccurlyeq \cdots \preccurlyeq (1,1,\dots,1).$$

**Определение 2.5.8.** *Функция* $f(x_1, x_2, \dots, x_n) \in P_2$ *называется* ***монотонной****, если для любых двух наборов* $\tilde{\alpha}$ *и* $\tilde{\beta}$ *таких, что* $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, *выполняется неравенство* $f(\tilde{\alpha}) \le f(\tilde{\beta})$. *Множество всех монотонных функций из* $P_2$ *обозначается через* $M$.

Приведём примеры некоторых монотонных функций. Функции $1$, $0$, $x$ и $x_1 \& x_2$ являются монотонными, а функции $\bar{x}$ и $x_1 \to x_2$ таковыми не являются.

**Теорема 2.5.6.** *Класс M замкнут.*

*Доказательство.* Рассмотрим суперпозицию от функций из $M$.

1) Пусть $f(x_1, x_2, \dots, x_n) \in M$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y) = f(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n).$$

Тогда монотонность $g$ непосредственно следует из монотонности функции $f$.

2) Пусть функции $f(x_1, x_2, \dots, x_n)$ и $h(y_1, y_2, \dots, y_m)$ принадлежат классу $M$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) =$$
$$= f(x_1, \dots, x_{j-1}, h(y_1, y_2, \dots, y_m), x_{j+1}, \dots, x_n).$$

Пусть $\tilde{\alpha}^{n+m-1} \preccurlyeq \tilde{\beta}^{n+m-1}$. Тогда, в частности,

$$(\alpha_n, \alpha_{n+1}, \dots, \alpha_{n+m-1}) \preccurlyeq (\beta_n, \beta_{n+1}, \dots, \beta_{n+m-1}).$$

Поскольку $h$ является монотонной функцией, то

$$h(\alpha_n, \alpha_{n+1}, \dots, \alpha_{n+m-1}) \leq h(\beta_n, \beta_{n+1}, \dots, \beta_{n+m-1})$$

и, как следствие,

$$(\alpha_1, \dots, \alpha_{j-1}, h(\alpha_n, \dots, \alpha_{n+m-1}), \alpha_{j+1}, \dots, \alpha_{n-1}) \preccurlyeq$$
$$\preccurlyeq (\beta_1, \dots, \beta_{j-1}, h(\beta_n, \dots, \beta_{n+m-1}), \beta_{j+1}, \dots, \beta_{n-1}).$$

Учитывая монотонность функции $f$ имеет место

$$g(\tilde{\alpha}^{n+m-1}) = f(\alpha_1, \dots, \alpha_{j-1}, h(\alpha_n, \dots, \alpha_{n+m-1}), \alpha_{j+1}, \dots, \alpha_{n-1}) \leq$$
$$\leq f(\beta_1, \dots, \beta_{j-1}, h(\beta_n, \dots, \beta_{n+m-1}), \beta_{j+1}, \dots, \beta_{n-1}) = g(\tilde{\beta}^{n+m-1}).$$

Таким образом, $g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) \in M$ и $[M] = M$. □

Через $M(n)$ обозначим множество всех функций из $M$ от $n$ переменных. В общем, задача нахождения числа монотонных функций является сложной и называется задачей Дидиченко. Точное значения числа монотонных функций известно только для $0 \leq n \leq 8$: 2, 3, 6, 20, 168, 7581, 7828354, 2414682040998, 56130437228687557907788.

Дадим нижнюю оценку числа $|M(n)|$. Обозначим через $\lfloor x \rfloor$ наибольшее целое, меньшее или равное числу $x$. Например, $\lfloor 2 \rfloor = 2$ и $\lfloor 3.7 \rfloor = 3$.

**Теорема 2.5.7.** $|M(n)| \geq 2^{C_n^{\lfloor n/2 \rfloor}}$.

*Доказательство.* Рассмотрим два случая, когда $n$ – чётное число и когда $n$ – нечётное число.

Сначала рассмотрим случай, когда $n$ – чётное число. На всех наборах, в которых количество нулей превышает количество единиц, значение функций зададим равным 0, а множество таких наборов обозначим через $A$. Если количество единиц превышает количество нулей, значение функций на таких наборах зададим равным 1, а множество всех таких наборов обозначим через $B$. На оставшихся наборах, где нулей и единиц поровну (множество этих наборов обозначим через Г), зададим значение функции произвольным образом. Пусть $f(\tilde{x}^n)$ – произвольная функция, заданная согласно вышеописанному алгоритму. Заметим, что для каждого

набора $\tilde{\alpha} \in A$, который предшествует какому-либо набору $\tilde{\beta} \in B \cup \Gamma$, имеем $f(\tilde{\alpha}) \leq f(\tilde{\beta})$, так как $f(\tilde{\alpha}) = 0$. Аналогично для каждого набора $\tilde{\alpha} \in A \cup \Gamma$, который предшествует какому-либо набору $\tilde{\beta} \in B$, имеем $f(\tilde{\alpha}) \leq f(\tilde{\beta})$, так как $f(\tilde{\beta}) = 1$. Кроме того, два различных набора $\tilde{\alpha}$ и $\tilde{\beta}$ из Γ несравнимы. Следовательно, $f(\tilde{x}^n)$ – монотонная функция, и поскольку $|\Gamma| = C_n^{\lfloor n/2 \rfloor}$, общее количество монотонных функций, заданных согласно вышеописанному алгоритму, равно $|M(n)| = 2^{C_n^{\lfloor n/2 \rfloor}}$.

Для нечётного $n$ предложим аналогичную конструкцию. На всех наборах, в которых количество единиц превышает число $\lfloor n/2 \rfloor$, значение функций зададим равным 1, а на наборах, в которых количество единиц меньше $\lfloor n/2 \rfloor$, значение функций зададим равным 0. На оставшихся наборах, где количество единиц равно $\lfloor n/2 \rfloor$, зададим значение функций произвольным образом. Как и в случае с чётным $n$, легко показать, что все такие функции будут монотонными, и их количество равно $2^{C_n^{\lfloor n/2 \rfloor}}$. □

Дадим верхнюю оценку числа $|M(n)|$.

**Теорема 2.5.8.** $|M(n)| \leq 3^{C_n^{\lfloor n/2 \rfloor}}$.

Данную теорему оставим без доказательства.

Напомним, что соседними называются наборы, различающиеся только в одной компоненте. Например, наборы

$$\tilde{\alpha} = (0,0,1,0) \text{ и } \tilde{\beta} = (0,1,1,0)$$

являются соседними.

**Теорема 2.5.9. (Условие немонотонности)** *Для любой пары наборов $\tilde{\alpha}$ и $\tilde{\beta}$ таких, что $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ и $f(\tilde{\alpha}) > f(\tilde{\beta})$, найдется пара соседних наборов $\tilde{\alpha}'$ и $\tilde{\beta}'$ с теми же свойствами: $\tilde{\alpha}' \preccurlyeq \tilde{\beta}'$ и $f(\tilde{\alpha}') > f(\tilde{\beta}')$.*

*Доказательство.* Если $\tilde{\alpha}$ и $\tilde{\beta}$ – соседние наборы, то утверждение теоремы верно. Иначе построим цепочку кратчайшей длины, состоящей из наборов $\tilde{\gamma}_0, \tilde{\gamma}_1, \dots, \tilde{\gamma}_k$, такую, что

$$\tilde{\alpha} = \tilde{\gamma}_0 \preccurlyeq \tilde{\gamma}_1 \preccurlyeq \cdots \preccurlyeq \tilde{\gamma}_k = \tilde{\beta},$$

и любые два расположенных рядом набора $\tilde{\gamma}_{i-1}, \tilde{\gamma}_i$ $(i = \overline{1,k})$ являются соседними. Очередной набор $\tilde{\gamma}_i$ получим из предыдущего набора $\tilde{\gamma}_{i-1}$ путём замены значения одной из его компонент на противоположное (это будет замена 0 на 1, так как $\tilde{\alpha} \preccurlyeq \tilde{\beta}$. Затем проверим условие немонотонности $f(\tilde{\gamma}_{i-1}) > f(\tilde{\gamma}_i)$. Если оно выполнено, то утверждение теоремы доказано ($\tilde{\alpha}' = \tilde{\gamma}_{i-1}$ и $\tilde{\beta}' = \tilde{\gamma}_i$). В противном случае получим и исследуем

очередной набор $\tilde{\gamma}_{i+1}$. В случае, когда постоянно выполняется условие монотонности, имеем

$$f(\tilde{\alpha}) = f(\tilde{\gamma}_1) = f(\tilde{\gamma}_2) = \cdots = f(\tilde{\gamma}_{k-1}),$$

но тогда получим $f(\tilde{\gamma}_{k-1}) = 1$ и $f(\tilde{\beta}) = 0$, а это значит, что условие немонотонности выполнится для последней пары наборов $\tilde{\alpha}' = \tilde{\gamma}_{k-1}$ и $\tilde{\beta}' = \tilde{\gamma}_k = \tilde{\beta}$. □

Основываясь на условии немонотонности, для того чтобы определить является ли булева функция $f$ монотонной, достаточно сравнить значения функции $f$ на всех соседних наборах. Если на каких-то двух соседних наборов $\tilde{\alpha}$ и $\tilde{\beta}$ таких, что $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, получим $f(\tilde{\alpha}) > f(\tilde{\beta})$, то функция $f$ не принадлежит классу $M$. Иначе $f \in M$.

**Пример 2.5.2.** Проверить, является ли монотонной функция

$$f(x_1, x_2, x_2) = \big(x_1 \downarrow (x_2 \longrightarrow x_3)\big) \vee x_1.$$

*Решение*. Запишем таблицу истинности функции $f$:

| $x_1$ | $x_2$ | $x_3$ | $x_2 \longrightarrow x_3$ | $x_1 \downarrow (x_2 \longrightarrow x_3)$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |

Таблица 2.5.2

Функция $f(x_1, x_2, x_2)$ не является монотонной, так как верны следующие отношения:

$$(0,1,0) \preccurlyeq (0,1,1) \text{ и } f(0,1,0) > f(0,1,1).$$

***Класс L***

**Определение 2.5.9.** *Функция $f(x_1, x_2, \dots, x_n) \in P_2$ называется **линейной**, если её полином Жегалкина имеет вид*

$$f(x_1, x_2, \dots, x_n) = \alpha_0 \oplus \alpha_1 \& x_1 \oplus \alpha_2 \& x_2 \oplus \dots \oplus \alpha_n \& x_n. \qquad (2.4)$$

*Множество всех линейных функций из $P_2$ обозначается через L.*

**Теорема 2.5.10.** *Класс L замкнут.*

*Доказательство.* Рассмотрим суперпозицию функций из $L$.

1) Пусть $f(x_1, x_2, \dots, x_n) \in L$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y) = f(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n).$$

Если функция $f$ имеет вид (2.4), то

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y) =$$

$$= \alpha_0 \oplus \alpha_1 \& x_1 \oplus \dots \oplus \alpha_{j-1} \& x_{j-1} \oplus \alpha_j \& y \oplus \alpha_{j+1} \& x_{j+1} \oplus \dots \oplus \alpha_n \& x_n.$$

Очевидно, что $g \in L$.

2) Пусть функции $f(x_1, x_2, \dots, x_n)$ и $h(y_1, y_2, \dots, y_m)$ принадлежат классу $L$ и

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) =$$

$$= f(x_1, \dots, x_{j-1}, h(y_1, y_2, \dots, y_m), x_{j+1}, \dots, x_n).$$

Пусть функция $f$ имеет вид (2.4) и

$$h(y_1, y_2, \dots, y_m) = \beta_0 \oplus \beta_1 \& y_1 \oplus \beta_2 \& y_2 \oplus \dots \oplus \beta_m \& y_m.$$

Тогда

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) =$$

$$= \alpha_0 \oplus \alpha_1 \& x_1 \oplus \dots \oplus \alpha_{j-1} \& x_{j-1} \oplus$$

$$\oplus \alpha_j \& (\beta_0 \oplus \beta_1 \& y_1 \oplus \beta_2 \& y_2 \oplus \dots \oplus \beta_m \& y_m) \oplus$$

$$\oplus \alpha_{j+1} \& x_{j+1} \oplus \dots \oplus \alpha_n \& x_n =$$

$$= \alpha_0 \oplus \alpha_j \& \beta_0 \oplus \alpha_1 \& x_1 \oplus \dots \oplus \alpha_{j-1} \& x_{j-1} \oplus$$

$$\oplus \alpha_{j+1} \& x_{j+1} \oplus \dots \oplus \alpha_n \& x_n \oplus$$

$$\oplus \alpha_j \& \beta_1 \& y_1 \oplus \alpha_j \& \beta_2 \& y_2 \oplus \dots \oplus \alpha_j \& \beta_m \& y_m.$$

Полученный полином Жегалкина линеен, то есть

$$g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y_1, \dots, y_m) \in L.$$

В конечном итоге $[L] = L$. □

Обозначим через $L(n)$ множество всех функций из $L$ от $n$ переменных.

**Теорема 2.5.11.** $|L(n)| = 2^{n+1}$.

*Доказательство.* По определению, линейная функция $f$ от $n$ имеет вид:

$$f(x_1, x_2, \dots, x_n) = \alpha_0 \oplus \alpha_1 \& x_1 \oplus \alpha_2 \& x_2 \oplus \dots \oplus \alpha_n \& x_n,$$

где $\alpha_0, \alpha_1, \dots, \alpha_n$ – логические константы. Из чего заключаем, что каждый линейный полином определяется двоичным набором $(\alpha_0, \alpha_1, \dots, \alpha_n)$ длины

$n+1$, и наоборот, каждый двоичный набор длины $n+1$ задаёт линейный полином Жегалкина некоторой функции от $n$ переменных. Следовательно, число линейных полиномов (число различных линейных функций от $n$ переменных) равно числу различных двоичным набором длины $n+1$, то есть равно $2^{n+1}$. □

Для того чтобы проверить функцию на линейность, необходимо построить для неё полином Жегалкина, и если он не содержит произведения переменных, то функция линейна.

**Пример 2.5.3.** Определить, является ли линейной функция

$$f(\tilde{x}^3) = (10100101).$$

*Решение*. Воспользуемся методом треугольника для построения полинома Жегалкина функции $f(\tilde{x}^3)$. Запишем таблицу истинности функции $f(\tilde{x}^3)$ и на её основе построим вспомогательную треугольную таблицу (Таблица 2.5.3).

| $x_1$ | $x_2$ | $x_3$ | $f(\tilde{x}^3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|
| $\alpha_0$ | $\alpha_3$ | $\alpha_2$ | $\alpha_{23}$ | $\alpha_1$ | $\alpha_{13}$ | $\alpha_{12}$ | $\alpha_{123}$ |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | |
| 1 | 1 | 1 | 0 | 1 | 0 | | |
| 0 | 0 | 1 | 1 | 1 | | | |
| 0 | 1 | 0 | 0 | | | | |
| 1 | 1 | 0 | | | | | |
| 0 | 1 | | | | | | |
| 1 | | | | | | | |

Таблица 2.5.3

Получим полином Жегалкина заданной функции:

$$f(\tilde{x}^3) = 1 \oplus x_1 \oplus x_3.$$

Данный полином не содержит произведения переменных. Следовательно, функция $f(\tilde{x}^3)$ является линейной.

**Пример 2.5.4.** Сколько функций от $n$ переменных содержит множество $(L \cup T_1) \cap S$?

*Решение*. Нужно определить мощность множества $(L \cup T_1) \cap S$. Упростим выражение:

$$(L \cup T_1) \cap S = (L \cap S) \cup (T_1 \cap S).$$

Используя формулу включений-исключений

$$|A \cup B| = |A| + |B| - |A \cap B|,$$

получаем

$$|(L \cap S) \cup (T_1 \cap S)| = |L \cap S| + |T_1 \cap S| - |L \cap T_1 \cap S|.$$

Общей вид линейной функции $f(x_1, x_2, \dots, x_n)$ указан в (2.4). Для того чтобы линейная функция была самодвойственной, должно выполнятся равенство $f(x_1, x_2, \dots, x_n) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$. Получим:

$$\alpha_0 \oplus \alpha_1 \& x_1 \oplus \alpha_2 \& x_2 \oplus \dots \oplus \alpha_n \& x_n =$$
$$= \alpha_0 \oplus \alpha_1 \& (x_1 \oplus 1) \oplus \alpha_2 \& (x_2 \oplus 1) \oplus \dots \oplus \alpha_n \& (x_n \oplus 1) \oplus 1 \Longrightarrow$$
$$\Longrightarrow \alpha_1 \oplus \alpha_2 \oplus \dots \oplus \alpha_n \oplus 1 = 0 \Longrightarrow \alpha_1 \oplus \alpha_2 \oplus \dots \oplus \alpha_n = 1.$$

При $n \geq 1$ ровно в половине случаев значение этой суммы равно 1 (когда по модулю два складывается нечётное количество единиц), а в другой половине оно равно 0. Отметим, что при любом значении $\alpha_0 \in \{0,1\}$ функция $f$, для которой $\alpha_1 \oplus \alpha_2 \oplus \dots \oplus \alpha_n = 1$, остаётся самодвойственной. Исходя из того, что $|L| = 2^{n+1}$, получим $|L \cap S| = |L \cap \bar{S}| = 2^{n+1}/2 = 2^n$ для $n \geq 1$. Отметим также, что половина только что учтённых функций, когда $\alpha_0 = 0$, принадлежит классу $T_1$, а другая половина нет, откуда следует $|L \cap T_1 \cap S| = 2^{n-1}$.

Осталось найти число функций из $T_1$, являющихся самодвойственными. Общее число функций из $T_1$ равно $2^{2^n-1}$. При $n \geq 1$ все самодвойственные функции могут быть заданы произвольным образом на всех наборах из нижней половины таблицы истинности, кроме набора из единиц. Это значит, что они определяются двоичными наборами длины $2^{n-1} - 1$ и, как следствие, получим $|T_1 \cap S| = 2^{2^{n-1}-1}$.

В результате имеем

$$|(L \cup T_1) \cap S| = 2^n + 2^{2^{n-1}-1} + 2^{n-1} = 3 \cdot 2^{n-1} + 2^{2^{n-1}-1}.$$

## 2.6 Критерий полноты

Перед тем как перейти к рассмотрению вопроса о необходимых и достаточных условиях полноты, докажем несколько лемм.

**Лемма 2.6.1. (О несамодвойственной функции)** *Пусть функция $f(x_1, x_2, \dots, x_n)$ не принадлежит классу $S$. Тогда, подставляя в $f$ вместо переменных $x_1, x_2, \dots, x_n$ переменную $x$ или ее отрицание $\bar{x}$, можно получить константу.*

*Доказательство.* Поскольку $f(x_1, x_2, \dots, x_n) \notin S$, то существует пара противоположных наборов $(\alpha_1, \alpha_2, \dots, \alpha_n)$ и $(\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n)$ таких, что $f(\alpha_1, \alpha_2, \dots, \alpha_n) = f(\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n)$. Подставим в $f$ вместо переменных

$x_1, x_2, \dots, x_n$ функции $x \oplus \alpha_1, x \oplus \alpha_2, \dots, x \oplus \alpha_n$, то есть вместо каждой переменной $x_i$ подставляется либо $x$, либо $\bar{x}$. В результате этой подстановки получим некоторую функцию

$$h(x) = f(x \oplus \alpha_1, x \oplus \alpha_2, \dots, x \oplus \alpha_n).$$

Имея в виду тот факт, что $h(0) = f(\alpha_1, \alpha_2, \dots, \alpha_n)$, $h(1) = f(\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n)$ и $f(\alpha_1, \alpha_2, \dots, \alpha_n) = f(\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n)$, выполняется равенство $h(0) = h(1)$, откуда следует, что $h(x)$ является константой. □

**Следствие 2.6.1.** *Если* $f \notin S$, *то* $0,1 \in [\{f, \bar{x}\}]$.

*Доказательство.* По Лемме 2.6.1 одна из констант содержится в $[\{f, \bar{x}\}]$. Вторая константа получается при помощи отрицания $\bar{x}$. □

**Лемма 2.6.2. (О немонотонной функции)** *Пусть функция* $f(x_1, x_2, \dots, x_n)$ *не принадлежит классу* $M$. *Тогда, подставляя в* $f$ *вместо некоторых переменных константы* 0 *и* 1, *можно получить отрицание.*

*Доказательство.* Поскольку $f \notin M$, то по условию немонотонности (Теорема 2.5.9 из предыдущего параграфа), существует пара соседних наборов $\tilde{\alpha}$ и $\tilde{\beta}$ таких, что $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ и $f(\tilde{\alpha}) > f(\tilde{\beta})$. Пусть $\tilde{\alpha}$ и $\tilde{\beta}$ различаются $i$-ой координатой:

$$\tilde{\alpha} = (\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n),\ \tilde{\beta} = (\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n).$$

Определим новую функцию $h(x) = f(\alpha_1, \dots, \alpha_{i-1}, x, \alpha_{i+1}, \dots, \alpha_n)$. Легко проверить, что $h(0) = f(\tilde{\alpha}) = 1$ и $h(1) = f(\tilde{\beta}) = 0$. То есть $h(x) = \bar{x}$, что и требовалось доказать □

Из Леммы 2.6.2 вытекает Следствие 2.6.2.

**Следствие 2.6.2.** *Если* $f \notin M$, *то* $\bar{x} \in [\{f, 0,1\}]$.

**Лемма 2.6.3. (О нелинейной функции)** *Пусть* $f(x_1, x_2, \dots, x_n)$ *не принадлежит классу* $L$. *Тогда, подставляя в* $f$ *вместо некоторых переменных константы* 0, 1, *отрицание* $\bar{x}$ *и, возможно, навешивая отрицание над* $f$, *можно получить конъюнкцию.*

*Доказательство.* Поскольку функция $f(x_1, x_2, \dots, x_n) \notin L$, то в её полиноме Жегалкина найдётся слагаемое, содержащее не менее двух переменных (нелинейное произведение). Выберем самое короткое из подобных произведений. Пусть оно имеет вид $x_1 \& x_2\& \dots \& x_p,\ p \geq 2$. Тогда

$$f(x_1, x_2, \dots, x_n) = x_1 \& x_2 \& \dots \& x_p \oplus A_1 \oplus \dots \oplus A_q,$$

где $A_i\ (i = \overline{1, q})$ – остальные слагаемые. Каждое другое нелинейное произведение содержит хотя бы одну переменную, отличную от переменных $x_1, x_2, \dots, x_p$. Подставим константу 0 вместо всех переменных

$x_{p+1}, x_{p+2}, \dots, x_n$. Тогда все остальные слагаемые из $A_1, A_2, \dots, A_q$, содержащие не менее двух переменных, обратятся в ноль. Поэтому

$$f(x_1, x_2, \dots, x_p, 0, \dots, 0) = x_1 \& x_2 \& \dots \& x_p \oplus h(x_1, x_2, \dots, x_p),$$

где $h(x_1, x_2, \dots, x_p)$ – некоторая линейная функция от переменных $x_1, x_2, \dots, x_p$.

Далее оставим переменные $x_1$ и $x_2$ без изменения, а вместо переменных $x_3, \dots, x_p$ (если они есть) подставим 1. Получим функцию

$$f(x_1, x_2, 1, \dots 1, 0, \dots, 0) = x_1 \& x_2 \oplus h(x_1, x_2, 1, \dots, 1) =$$
$$= x_1 \& x_2 \oplus \alpha \& x_1 \oplus \beta \& x_2 \oplus \gamma,$$

где $\alpha, \beta, \gamma \in \{0,1\}$. Обозначим получившуюся функцию через $g(x_1, x_2)$. Теперь подставим в функцию $g(x_1, x_2)$ вместо переменных $x_1$ и $x_2$ соответственно функции $x_1 \oplus \beta$ и $x_2 \oplus \alpha$ и прибавим ко всей функции константу $\alpha \& \beta \oplus \gamma$ (т. е. либо ничего не изменим, либо навесим отрицание). В результате получим

$$g(x_1 \oplus \beta, x_2 \oplus \alpha) \oplus (\alpha \& \beta \oplus \gamma) =$$
$$= (x_1 \oplus \beta) \& (x_2 \oplus \alpha) \oplus \alpha \& (x_1 \oplus \beta) \oplus$$
$$\oplus \beta \& (x_2 \oplus \alpha) \oplus \gamma \oplus (\alpha \& \beta \oplus \gamma) =$$
$$= x_1 \& x_2 \oplus x_1 \& \alpha \oplus \beta \& x_2 \oplus \beta \& \alpha \oplus \alpha \& x_1 \oplus \alpha \& \beta \oplus$$
$$\oplus \beta \& x_2 \oplus \beta \& \alpha \oplus \gamma \oplus \alpha \& \beta \oplus \gamma = x_1 \& x_2.$$

Лемма доказана. □

Из Леммы 2.6.3 вытекает Следствие 2.6.3.

**Следствие 2.6.3.** *Если* $f \notin L$, *то* $x_1 \& x_2 \in [\{f, 0, 1, \bar{x}\}]$.

Теперь мы можем перейти к доказательству одной из основных теорем алгебры логики, которая в литературе называется теоремой Поста о полноте.

**Теорема 2.6.1. (О функциональной полноте)** *Система функций* $F$ *функционально полна тогда и только тогда, когда она целиком не содержится ни в одном из пяти замкнутых классов:* $T_0, T_1, S, M, L$.

*Доказательство.* Докажем необходимость и достаточность.

Необходимость. Если система $F$ содержится в каком-либо из указанных замкнутых классов, например в $T_0$, то в силу свойств замыкания $[F] \subseteq [T_0]$, а это значит, что система $F$ неполная.

Достаточность. Так как $F$ целиком не содержится ни в одном из пяти указанных замкнутых классов, то в $F$ найдутся функции $f_0, f_1, f_S, f_M, f_L$, которые не принадлежат соответственно классам $T_0, T_1, S, M, L$.

Разобьём дальнейшее доказательство на три этапа.

1) Получение констант. Возьмём функцию $f_0 \notin T_0$. Рассмотрим функцию $g(x) = f_0(x, \dots, x)$. Тогда $g(0) = f_0(0, \dots, 0) = 1$. Если $g(1) = f_0(1, \dots, 1) = 1$, то функция $g$ – константа 1, а константу 0 можно получить следующим образом:

$$h(x) = f_1\big(g(x), \dots, g(x)\big) = f_1(1, \dots, 1) = 0,$$

где $f_1 \notin T_1$. Если же $g(1) = 0$, то $g(x) = \bar{x}$. Тогда по Лемме о несамодвойственной функции при помощи отрицания из $f_S \notin S$ можно получить обе константы.

2) Получение отрицания. По Лемме о немонотонной функции при помощи констант и функции $f_M \notin M$ можно получить отрицание.

3) Получение конъюнкции. По Лемме о нелинейной функции при помощи констант, отрицания и функции $f_L \notin L$ можно легко получить конъюнкцию.

Таким образом, полная система $\{\bar{x}, x_1 \& x_2\}$ содержится в $[F]$, что доказывает полноту системы $F$. □

**Следствие 2.6.4.** *Всякая неполная система функций содержится по крайней мере в одном из пяти замкнутых классов:* $T_0, T_1, S, M, L$.

При исследовании полноты систем функций $F$ удобно пользоваться таблицей, которую будем называть *критериальной таблицей*. Эта таблица имеет пять столбцов, каждый из которых соответствует одному классу из семейства $H = \{T_0, T_1, S, M, L\}$, а строки таблицы соответствуют функциям системы $F$. На пересечении строки, соответствующей функции $f \in F$, и столбца, соответствующего классу $K \in H$, ставится знак плюс, если $f \in K$, и минус, если $f \notin K$. Система функций полна тогда и только тогда, когда в каждом столбце содержится хотя бы один знак минус.

**Пример 2.6.1.** Исследовать полноту системы

$$F = \{x_1 \oplus x_2 \oplus x_3, x_1 \& x_2, 1, 0\}.$$

*Решение*. Построим критериальную таблицу для системы $F$ (Таблица 2.6.1). Для этого проанализируем каждую функцию из $F$ на принадлежность классам $T_0, T_1, S, M, L$.

Начнём с функции $f_1(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$:

1) $f_1(0,0,0) = 0 \Longrightarrow f_1 \in T_0$.
2) $f_1(1,1,1) = 1 \Longrightarrow f_1 \in T_1$.
3) $f_1^* = \bar{f}_1(\bar{x}_1, \bar{x}_2, \bar{x}_3) = \overline{\bar{x}_1 \oplus \bar{x}_2 \oplus \bar{x}_3} =$
$= (x_1 \oplus 1) \oplus (x_2 \oplus 1) \oplus (x_3 \oplus 1) \oplus 1 =$
$= x_1 \oplus x_2 \oplus x_3 \Longrightarrow f_1 \in S$.

4) Для наборов $(0,1,0) \preccurlyeq (0,1,1)$ имеем
$$f_1(0,1,0) = 1 > f_1(0,1,1) = 0,$$
откуда $f_1 \notin M$.
5) Очевидно, что $f_1 \in L$.

Теперь проанализируем функцию $f_2(x_1, x_2) = x_1 \& x_2$:

1) $f_2(0,0) = 0 \Longrightarrow f_2 \in T_0$.
2) $f_2(1,1) = 1 \Longrightarrow f_2 \in T_1$.
3) $f_2^* = \overline{\bar{x}_1 \& \bar{x}_2} = x_1 \vee x_2 \Longrightarrow f_2 \notin S$.
4) $f_2 \in M$, так как не существуют такие два набора $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, что $f_2(\tilde{\alpha}) > f_2(\tilde{\beta})$.
5) Видно, что $f_2 \notin L$.

Оставшиеся функции $f_3 = 1$ и $f_4 = 0$ проверяются элементарно.

| | $T_0$ | $T_1$ | $S$ | $M$ | $L$ |
|---|---|---|---|---|---|
| $x_1 \oplus x_2 \oplus x_3$ | + | + | + | − | + |
| $x_1 \& x_2$ | + | + | − | + | − |
| 1 | − | + | − | + | + |
| 0 | + | − | − | + | + |

Таблица 2.6.1

Заполнив критериальную таблицу, можно убедится в том, что система $F$ полна, так как она не содержится целиком ни в одном из пяти замкнутых классов (каждый столбец Таблицы 2.6.1 содержит не менее одного минуса).

## 2.7 Базисы. Предполные классы

**Определение 2.7.1.** *Полная система функций $F$ называется **базисом** в $P_2$, если никакая её подсистема не является полной.*

Например, система $\{x_1 | x_2\}$ является базисом, а $\{\bar{x}, x_1 \& x_2, x_1 \vee x_2\}$ базисом не является, потому что её подсистема $\{\bar{x}, x_1 \& x_2\}$ – базис.

**Теорема 2.7.1.** *Любой базис в $P_2$ состоит не более чем из четырёх функций.*

*Доказательство.* Покажем, что из любой полной системы можно выделить полную подсистему, содержащую не более четырёх функций. Действительно, если система функций $F$ полна, то согласно Теореме о функциональной полноте в ней существует пять функций $f_0 \notin T_0$, $f_1 \notin T_1, f_S \notin S, f_M \notin M, f_L \notin L$, которые образуют полную систему.

Обратимся к первому этапу Теоремы о функциональной полноте (получение констант). В первом случае, когда $g(x) = 1$, функция $g(x)$ является несамодвойственной и, следовательно, можно обойтись без функции $f_S$. Во втором случае, когда $g(x) = \bar{x}$, функция $g(x)$ является немонотонной и поэтому можно обойтись без функции $f_M$. Как следствие, из любой полной системы можно выделить полную подсистему, состоящую не более чем из четырёх функций.

Отметим также, что в общем случае это число уменьшить нельзя. Это следует из системы $F$, которая исследовалась в Примере 2.6.1. Ни одну функцию из системы $F$ удалить нельзя, так как все функции, кроме константы 1, принадлежат классу $T_0$, все функции, кроме константы 0, принадлежат классу $T_1$, все функции, кроме $x_1 \oplus x_2 \oplus x_3$, принадлежат классу $M$ и все функции, кроме $x_1 \& x_2$, принадлежат классу $L$. □

**Определение 2.7.2.** *Множество булевых функций $F$ называется* ***предполным классом****, если выполняются следующие условия:*

1) $F \neq P_2$;
2) $F = [F]$;
3) *для любой функции $f \in P_2 \backslash F$ класс $F \cup \{f\}$ полный.*

**Теорема 2.7.2.** *В $P_2$ существует только пять предполных классов: $T_0, T_1, S, M, L$.*

*Доказательство.* Покажем сначала, что ни один из пяти классов $T_0, T_1, S, M, L$ не содержится в другом. Для этого укажем для каждого из пяти классов функции, принадлежащие данному классу, но не принадлежащие остальным четырём. Составим таблицу из пяти строк и пяти столбцов. Каждой строке и каждому столбцу соответствуют классы $T_0, T_1, S, M, L$. В ячейку таблицы, стоящую па пересечении $i$-й строки и $j$-го столбца, $i \neq j$, поместим функцию, которая принадлежит классу, соответствующему $i$-й строке, и не принадлежащую классу, который соответствует $j$-му столбцу. Ячейки таблицы, стоящие па главной диагонали, оставим пустыми (Таблица 2.7.1).

| | $T_0$ | $T_1$ | $S$ | $M$ | $L$ |
|---|---|---|---|---|---|
| $T_0$ | | 0 | 0 | $x_1 \oplus x_2$ | $x_1 \& x_2$ |
| $T_1$ | 1 | | 1 | $x_1 \leftrightarrow x_2$ | $x_1 \& x_2$ |
| $S$ | $\bar{x}$ | $\bar{x}$ | | $\bar{x}$ | $x_1 \& (x_2 \vee x_3) \vee x_2 \& x_3$ |
| $M$ | 1 | 0 | 0 | | $x_1 \& x_2$ |
| $L$ | 1 | 0 | 0 | $x_1 \oplus x_2$ | |

Таблица 2.7.1

Покажем теперь, что классы $T_0, T_1, S, M$ и $L$ являются предполными. Пусть $K \in \{T_0, T_1, S, M, L\}$ и $f \notin K$. Тогда система $K \cup \{f\}$ не содержится ни в одном из пяти классов, так как $K$ не содержится в четырёх из них и $f \notin K$. Согласно Теореме 2.6.1 (о функциональной полноте) система $K \cup \{f\}$ функционально полна, а $K$ является предполным классом.

Остаётся доказать, что в $P_2$ не существует других предполных классов. Пусть $A$ − предполный класс. Тогда $[A] \neq P_2$ и по Теореме 2.6.1 существует такой класс $K \in \{T_0, T_1, S, M, L\}$, что $A \subseteq K$. Если $A \neq K$, то найдётся функция $f$ такая, что $f \in K$ и $f \notin A$, тогда $A \cup \{f\} \subseteq K$ и, следовательно, $[A \cup \{f\}] \subseteq [K] \neq P_2$. Полученное противоречие завершает доказательство теоремы. □

На следующем примере покажем, как с помощью критериальной таблицы найти все базисы, содержащиеся в системе функций.

**Пример 2.7.1.** Исследовать полноту системы

$$F = \{f_1 = (00110111), x_1 \& x_2 \oplus x_1 \& x_3 \oplus x_2 \& x_3, \bar{x}, 1\}$$

и выделить всевозможные базисы.

*Решение*. Построим критериальную таблицу для системы $F$.

Сначала проанализируем функцию $f_1 = (00110111)$:

1) Очевидно, что $f_1 \in T_0$ и $f_1 \in T_1$.
2) На противоположных наборах $(0,1,0)$ и $(1,0,1)$ функция $f_1$ принимает одинаковые значения $f_1(0,1,0) = f_1(1,0,1) = 1$. Следовательно, $f_1 \notin S$.
3) $f_1 \in M$, так как не существуют таких двух наборов $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, что $f_1(\tilde{\alpha}) > f_1(\tilde{\beta})$.
4) Воспользуемся методом треугольника для построения полинома Жегалкина функции $f_1$ (Таблица 2.7.2).

| $x_1$ | $x_2$ | $x_3$ | $f(\tilde{x}^3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|
| $\alpha_0$ | $\alpha_3$ | $\alpha_2$ | $\alpha_{23}$ | $\alpha_1$ | $\alpha_{13}$ | $\alpha_{12}$ | $\alpha_{123}$ |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | |
| 1 | 0 | 1 | 1 | 0 | 0 | | |
| 1 | 1 | 0 | 1 | 0 | | | |
| 0 | 1 | 1 | 1 | | | | |
| 1 | 0 | 0 | | | | | |
| 1 | 0 | | | | | | |
| 1 | 0 | | | | | | |
| 1 | | | | | | | |

Таблица 2.7.2

Таким образом, полином Жегалкина для функции $f_1$ выглядит так:

$$f_1 = x_2 \oplus x_1 \& x_3 \oplus x_1 \& x_2 \& x_3.$$

Понятно, что $f_1 \notin L$.

Оставшиеся функции $f_2 = x_1 \& x_2 \oplus x_1 \& x_3 \oplus x_2 \& x_3$, $f_3 = \bar{x}$ и $f_4 = 1$ проверяются аналогично.

| | $T_0$ | $T_1$ | $S$ | $M$ | $L$ |
|---|---|---|---|---|---|
| $f_1$ | + | + | − | + | − |
| $f_2$ | + | + | + | + | − |
| $f_3$ | − | − | + | − | + |
| $f_4$ | − | + | − | + | + |

Таблица 2.7.3

Каждый столбец критериальной таблицы (Таблица 2.3.7) содержит, по крайней мере, один минус. Вследствие этого система $F$ является полной.

Нам остаётся выделить из $F$ всевозможные базисы. По критериальной таблице составим выражение, представляющее собой КНФ, в котором элементарные дизъюнкции соответствуют столбцам таблицы и включают в качестве слагаемых символы тех функций, которые не входят в класс, соответствующий столбцу. Для системы функций $F$ имеем

$$K = (f_3 \vee f_4) \& f_3 \& (f_1 \vee f_4) \& f_3 \& (f_1 \vee f_2).$$

Раскрывая скобки и используя для упрощения эквивалентности вида $A \& A = A$, $A \& (A \vee B) = A$ и $A \vee A \& B = A$, приведем КНФ $K$ к ДНФ $D$, в которой закон поглощения уже неприменим. Таким образом, имеем

$$K = (f_3 \vee f_4) \& f_3 \& (f_1 \vee f_4) \& f_3 \& (f_1 \vee f_2) =$$
$$= f_3 \& (f_1 \vee f_4) \& (f_1 \vee f_2) =$$
$$= f_3 \& (f_1 \vee f_1 \& f_2 \vee f_4 \& f_1 \vee f_4 \& f_2) =$$
$$= f_3 \& (f_1 \vee f_4 \& f_2) = f_1 \& f_3 \vee f_2 \& f_3 \& f_4 = D.$$

По полученной ДНФ $D$ выпишем подмножества функций, соответствующие слагаемым выражения $D$. Это и будут искомые базисы. В системе $F$ имеются два базиса: $B_1 = \{f_1, f_3\}$ и $B_2 = \{f_2, f_3, f_4\}$.

## 2.8 Задачи для самостоятельного решения

1. Перечислить все существенные и фиктивные переменные функций:
   a) $f(\tilde{x}^3) = (11110011)$;
   b) $f(\tilde{x}^4) = (0101111101011111)$;

c) $f(\tilde{x}^4) = (1011010110110101)$;
d) $f(\tilde{x}^2) = (x_1 \leftrightarrow x_2) \vee (x_1|x_2)$;
e) $f(\tilde{x}^3) = \big((x_1 \oplus x_2) \to x_3\big)\&\overline{x_3 \to x_2}$;
f) $f(\tilde{x}^4) = \Big(x_1 \to \big((x_2 \to x_3) \to x_4\big)\Big) \leftrightarrow \bar{x}_1\&(x_2 \to x_3)\&\bar{x}_4$.

2. Построить полином Жегалкина для функций:
a) $f(\tilde{x}^3) = (10101110)$;
b) $f(\tilde{x}^3) = (01110011)$;
c) $f(\tilde{x}^4) = (0100000000010001)$;
d) $f(x,y,z) = \bar{x}\&y\&z \vee x\&\bar{z}$;
e) $f(\tilde{x}^3) = (x_1 \oplus x_2) \vee (x_2|x_3)$;
f) $f(\tilde{x}^4) = (x_1 \to x_2) \to (x_3 \to x_1\&x_4)$.

3. Выяснить, являются ли самодвойственными функции:
a) $f(\tilde{x}^4) = (1100100101101100)$;
b) $f(x,y,z) = x\&y \vee z$;
c) $f(x,y,z) = (x \vee \bar{y} \vee z)\&y \vee x\&\bar{y}\&z$;
d) $f(x,y,z) = x\&(z \to y) \vee \overline{y \to z}$;
e) $f(\tilde{x}^3) = (x_1 \to x_2) \oplus (x_2 \to x_3) \oplus (x_2 \to x_1)$.

4. Определить, являются ли монотонными функции:
a) $f(\tilde{x}^3) = (00010111)$;
b) $f(\tilde{x}^4) = (0010001101111111)$;
c) $f(x,y) = x \oplus y\&(x \leftrightarrow y)$;
d) $f(x,y,z) = (x \vee y \vee z)\&(\bar{x} \vee y \vee z)\&(x \vee y \vee \bar{z})$;
e) $f(\tilde{x}^3) = x_1\&x_2 \oplus x_2\&x_3 \oplus x_3\&x_1 \oplus x_1$.

5. Определить, являются ли линейными функции:
a) $f(\tilde{x}^3) = (10100110)$;
b) $f(\tilde{x}^4) = (0011110011000011)$;
c) $f(\tilde{x}^3) = \big((x_1 \to x_2)(x_2 \to x_1)\big) \leftrightarrow x_3$;
d) $f(x,y,z) = x\&y\&\bar{z} \oplus y\&(x \leftrightarrow y)$;
e) $f(x,y,z) = \bar{x}\&y\&z \vee x\&\bar{z}$.

6. Заменить прочерки символами 0 и 1 так, чтобы получился вектор значений некоторых линейных функций. Выразить полученные функции полиномом:
a) $f(\tilde{x}^3) = (-001--10)$;
b) $f(\tilde{x}^3) = (11-0---1)$;

c) $f(\tilde{x}^4) = (--10----0--1-110)$;
d) $f(\tilde{x}^4) = (-11-1---1------0)$;
e) $f(\tilde{x}^4) = (---0-00-1-0-----)$.

7. Подсчитать число функций, зависящих от $n$ переменных и принадлежащих классу $A$:
   a) $A = T_0 \cap T_1$;
   b) $A = T_1 \cup L$;
   c) $A = T_0 \cup S$;
   d) $A = (T_0 \backslash T_1) \cap L$;
   e) $A = (S \cup L) \backslash T_1$;
   f) $A = (S \cap L) \backslash (T_0 \cup T_1)$;
   g) $A = (L \backslash S) \cup (T_0 \backslash T_1)$.

8. Сведением к заведомо полной системе показать, что множество функций $F$ является полной системой:
   a) $F = \{x\&y \oplus z, (x \leftrightarrow y) \oplus z\}$;
   b) $F = \{x \to y, \overline{x \oplus y \oplus z}\}$;
   c) $F = \{\bar{x}\&\bar{y} \vee z, x \oplus y\}$;
   d) $F = \{x\&y \vee \bar{x}\&\bar{z}, f = (01111110)\}$.

9. Исследовать полноту системы функций и выделить всевозможные базисы, если системы функционально полны:
   a) $F = \{(y \to x)\&(\bar{y} \to z), 0, 1\}$;
   b) $F = \{x\&y \oplus z, x \oplus y \oplus z, x \oplus y \oplus 1\}$;
   c) $F = \{x \oplus y, x\&y \oplus z, x \oplus y \oplus z \oplus 1, x\&y \oplus y\&z \oplus z\&x\}$;
   d) $F = \{1, \bar{x}, x \leftrightarrow y, x\&(y \leftrightarrow z) \oplus \bar{x}\&(y \oplus z)\}$;
   e) $F = \{0, x \oplus y, x \to y, x\&y \leftrightarrow x\&z\}$;
   f) $F = \{x\&y, x\&y \vee z, x \oplus y, x \to y, \bar{x}\}$.

# Глава 3
# Исчисление высказываний

## 3.1 Вывод формул в исчислении высказываний

**Определение 3.1.1.** ***Аксиоматическая система в исчислении высказываний*** *– это формальная система, позволяющая формулировать высказывания путём указания исходных высказываний, имеющих значение истина, каждое из которых описывает как формулировать новое высказывание из уже построенных высказываний.*

Изучение исчисления высказываний начнём с определения аксиом и правил вывода, обеспечивающих доказательство истинности заключения.

**Определение 3.1.2.** ***Аксиома*** *– это высказывание, имеющее значение истины при любых значениях пропозициональных переменных, входящих в это высказывание.*

**Определение 3.1.3.** ***Правилом вывода*** *называют такое отношение между высказываниями, которое позволяет из множества посылок и аксиом делать выводы об истинности заключения.*

**Определение 3.1.4.** ***Доказательством формулы (выводом формулы)*** *$B$ называют конечный список формул $F_1, F_2, \dots, F_n$, где $F_n = B$ и каждая формула $F_i$, $1 \le i \le n$, есть аксиома или получена из предыдущих формул по одному из правил вывода. В этом случае формулу $B$ называют* ***выводимой***.

Говорят, что формула $B$ выводима из $\Gamma = \{A_1, A_2, \dots, A_n\}$, если существует вывод формулы $B$ из списка Γ. В данном случае для выводимости используется обозначение $\Gamma \vdash B$. Формулы $A_i$ $(i = \overline{1,n})$ из списка Γ называются *гипотезами*, а выражение $\Gamma \vdash B$ называется *секвенцией*. Последовательность гипотез в списке не играет роли, поэтому будем

обращаться с Γ как со множеством. Когда $\Gamma = \emptyset$, пишем $\emptyset \vdash B$ или $\vdash B$.

**Определение 3.1.5.** ***Полная система аксиом*** *– это такая система аксиом, исходя из которой можно либо доказать, либо опровергнуть любое утверждение.*

Исчисление высказываний опирается на четыре составляющие: алфавит, формулы, системы аксиом и правила вывода.

*Алфавит состоит из:*

1) символов пропозициональных переменных: $A, B, \dots, Z$ (возможно с индексами);
2) символов логических связок: $\neg, \rightarrow, \&, \vee$;
3) вспомогательных символов: круглые скобки, запятые.

*Формулы определяются индуктивно:*

1) всякая пропозициональная переменная есть формула;
2) если $A$ и $B$ формулы, тогда $\bar{A}$, $(A \rightarrow B)$, $(A \& B)$, $(A \vee B)$ – тоже формулы;
3) выражение исчисления высказываний является формулой тогда и только тогда, когда это следует из 1) и 2).

Заметим, что из определения формул следует, что всякая формула исчисления высказываний есть пропозициональная формула, построенная из пропозициональных букв с помощью связок. Так что будем придерживаться тех же правил опускания скобок в формулах, что и раньше для алгебры высказываний.

*Системы аксиом*

В качестве основной полной системы аксиом, опирающейся на две логические связки → и ¬, используются следующие формулы:

$A_1. A \rightarrow (B \rightarrow A)$;

$A_2. \big(A \rightarrow (B \rightarrow C)\big) \rightarrow \big((A \rightarrow B) \rightarrow (A \rightarrow C)\big)$;

$A_3. (\bar{B} \rightarrow \bar{A}) \rightarrow \big((\bar{B} \rightarrow A) \rightarrow B\big)$.

В зависимости от используемых логических связок, выделяются дополнительные полные системы аксиом. Укажем несколько из них.

I $(\rightarrow)$

1. $A \rightarrow (B \rightarrow A)$;
2. $\big(A \rightarrow (B \rightarrow C)\big) \rightarrow \big((A \rightarrow B) \rightarrow (A \rightarrow C)\big)$.

II $(\rightarrow, \&)$

1. $A\&B \rightarrow A$;
2. $A\&B \rightarrow B$;
3. $(A \rightarrow B) \rightarrow \Big((A \rightarrow C) \rightarrow \big(A \rightarrow (B\&C)\big)\Big)$.

III $(\rightarrow, \vee)$

1. $A \rightarrow A \vee B$;
2. $B \rightarrow A \vee B$;
3. $(A \rightarrow C) \rightarrow \Big((B \rightarrow C) \rightarrow \big((A \vee B) \rightarrow C\big)\Big)$.

IV $(\rightarrow, \neg)$

1. $(A \rightarrow B) \rightarrow (\bar{B} \rightarrow \bar{A})$;
2. $A \rightarrow \bar{\bar{A}}$;
3. $\bar{\bar{A}} \rightarrow A$.

*Правила вывода включают в себя два правила:*

1. *Правило подстановки*. Пусть $A$ – формула, содержащая переменную $X$. Тогда, если $A$ является выводимой формулой исчисления высказываний, то заменив в ней $X$ всюду, куда она входит, произвольной формулой $B$, получим выводимую формулу. Записывается так:
$$\int_X^B (A).$$
2. *Правило заключения* или *Modus Ponens (MP)*. Если $A$ и $A \rightarrow B$ выводимые формулы исчисления высказываний, то $B$ – также выводимая формула. Схематическая запись этого правила имеет вид:
$$\frac{A, A \rightarrow B}{B}.$$

Приведём несколько примеров доказательств формул.

**Пример 3.1.1.** Доказать выводимость формулы $(A \rightarrow B) \rightarrow (A \rightarrow A)$.

*Решение*. Построим вывод формулы $(A \rightarrow B) \rightarrow (A \rightarrow A)$.

1. $\big(A \rightarrow (B \rightarrow C)\big) \rightarrow \big((A \rightarrow B) \rightarrow (A \rightarrow C)\big)$ (аксиома $A_2$).
2. $\int_C^A (1) \vdash \big(A \rightarrow (B \rightarrow A)\big) \rightarrow \big((A \rightarrow B) \rightarrow (A \rightarrow A)\big)$ (подставляем $A$ вместо $C$ в формулу из пункта 1).
3. $A \rightarrow (B \rightarrow A)$ (аксиома $A_1$).

4. $MP(3,2) \vdash (A \to B) \to (A \to A)$ (по Modus Ponens для пунктов вывода 3 и 2).

**Пример 3.1.2.** Доказать, что $\vdash \bar{\bar{\bar{A}}} \to \bar{A}$.

*Решение*. Построим вывод:

1. $(A \to B) \to (\bar{B} \to \bar{A})$ (аксиома IV. 1).
2. $\int_B^{\bar{\bar{A}}}(1) \vdash \left(A \to \bar{\bar{A}}\right) \to \left(\bar{\bar{\bar{A}}} \to \bar{A}\right)$.
3. $A \to \bar{\bar{A}}$ (аксиома IV. 2).
4. $MP(3,2) \vdash \bar{\bar{\bar{A}}} \to \bar{A}$.

**Пример 3.1.3.** Доказать выводимость формулы $A\&B \to B\&A$.

*Решение*. Построим цепочку вывода:

1. $(A \to B) \to \Big((A \to C) \to \big(A \to (B\&C)\big)\Big)$ (аксиома II. 3).
2. $\int_{A,C}^{A\&B,A}(1) \vdash (A\&B \to B) \to \Big((A\&B \to A) \to \big((A\&B) \to (B\&A)\big)\Big)$ (подставляем $A\&B$ вместо $A$ в формулу из пункта 1 и после подставляем $A$ вместо $C$ в получившуюся формулу).
3. $A\&B \to B$ (аксиома II. 2).
4. $MP(3,2) \vdash (A\&B \to A) \to \big((A\&B) \to (B\&A)\big)$.
5. $A\&B \to A$ (аксиома II. 1).
6. $MP(5,4) \vdash (A\&B) \to (B\&A)$.

Докажем две теоремы, которые нам потребуются в дальнейшем.

**Теорема 3.1.1.** $\vdash A \to A$.

*Доказательство.* Построим вывод:

1. $\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$ (аксиома $A_2$).
2. $\int_{C,B}^{A,A\to A}(1) \vdash \Big(A \to \big((A \to A) \to A\big)\Big) \to \Big(\big(A \to (A \to A)\big) \to (A \to A)\Big)$.
3. $A \to (B \to A)$ (аксиома $A_1$).
4. $\int_B^{A\to A}(3) \vdash A \to \big((A \to A) \to A\big)$.
5. $MP(4,2) \vdash \big(A \to (A \to A)\big) \to (A \to A)$.
6. $\int_B^{A}(3) \vdash A \to (A \to A)$.
7. $MP(6,5) \vdash A \to A$.

Теорема доказана. □

**Теорема 3.1.2. (Свойства выводимости)** *Пусть $A$ и $B$ – произвольные формулы, а $\Gamma$ и $\Delta$ – множества формул. Тогда:*

1) *если $A \in \Gamma$, то $\Gamma \vdash A$ (рефлексивность);*
2) *если $\Gamma \subset \Delta$ и $\Gamma \vdash A$, то $\Delta \vdash A$ (уточнение);*
3) *если $\Gamma \vdash A_i$ для всех $A_i \in \Delta$ и $\Delta \vdash B$, то $\Gamma \vdash B$ (транзитивность).*

*Доказательство.* Свойство 1) очевидно. Свойство 2) означает, что к списку гипотез можно добавить совершенно произвольные формулы, при этом вывод останется прежним, просто новые гипотезы в нём не будут участвовать. Свойство 3) доказывается путём реконструкции выводов (если все гипотезы $A_i \in \Delta$ заменить на их выводы из $\Gamma$, то в результате получится вывод формулы $B$ из $\Gamma$). □

## 3.2 Теорема дедукции

Доказательства выводимости сильно упрощает следующая теорема.

**Теорема 3.2.1. (Теорема дедукции)** *Если $\Gamma, A \vdash B$, то $\Gamma \vdash A \to B$.*

*Доказательство.* Пусть $B_1, B_2, \dots, B_n$ $(B_n = B)$ – вывод $B$ из $\Gamma \cup \{A\}$. Построим вывод формулы $A \to B$ из $\Gamma$. Таким выводом могла бы быть последовательность $A \to B_1, A \to B_2, \dots, A \to B_n$, но для этого каждую формулу $A \to B_i$, $1 \le i \le n$, ещё нужно вывести. Построим вывод формулы $A \to B_i$ индукцией по $i$.

I. Для $i = 1$ формула $B_1$ – либо аксиома, либо принадлежит $\Gamma$, либо совпадает с $A$.

a) Если $B_1$ – аксиома, то строим цепочку вывода:
   1. $A \to (B \to A)$ (аксиома $A_1$).
   2. $\int_{A,B}^{B_1,A}(1) \vdash B_1 \to (A \to B_1)$.
   3. $B_1$ (аксиома).
   4. $MP(3,2) \vdash A \to B_1$.
b) Если $B_1 \in \Gamma$, то вывод такой же, как и в предыдущем случае.
c) Если $B_1 = A$, тогда согласно Теореме 3.1.1 получим $\vdash A \to A$.

II. Пусть построены цепочки выводов $A \to B_1, \dots, A \to B_{i-1}, i > 1$. Построим вывод формулы $A \to B_i$. Возможны четыре случая: $B_i$ – аксиома, либо $B_i \in \Gamma$, либо $B_i = A$, либо $B_i$ непосредственно выводится по правилу $MP$ из формул, предшествующих ей в исходном выводе. Первые три случая аналогичны случаям, разобранным выше. В продолжении рассмотрим четвёртый случай. Пусть $B_i$ выводится путём применения пра-

вила $MP$ к формулам $B_j$ и $B_k$, $(j,k<i)$, причём $B_k$ имеет вид $B_j \to B_i$. По предположению индукции уже имеются выводы из Γ формул $A \to B_j$ $(*)$ и $A \to B_k$, т. е. $A \to (B_j \to B_i)$ $(**)$. Искомая цепочка теперь строится следующим образом:

1. $\big(A \to (B \to C)\big) \to \big((A \to B) \to\ (A \to C)\big)$ (аксиома $A_2$).
2. $\int_{B,C}^{B_j,B_i}(1) \vdash (A \to (B_j \to B_i\ )) \to \Big(\big(A \to B_j\big) \to (A \to B_i\ )\Big)$.
3. $MP(**,2) \vdash (A \to B_j) \to (A \to B_i)$.
4. $MP(*,3) \vdash A \to B_i$.

Таким образом, существует вывод:

$$A \to B_1, \dots, A \to B_i, \dots, A \to B.$$

Теорема доказана. □

**Следствие 3.2.1.** *Если* $A \vdash B$, *то* $\vdash A \to B$.

Следствие 3.2.1 следует из Теоремы 3.2.1 при $\Gamma = \emptyset$.

**Теорема 3.2.2. (Обратная теорема дедукции)** *Если* $\Gamma \vdash A \to B$, *то* $\Gamma, A \vdash B$.

*Доказательство.* Пусть $C_1, C_2, \dots, C_m$ $(C_m = A \to B)$ – вывод $A \to B$ из Γ. Добавим к нему ещё два шага вывода:

1. $C_1$.
2. $C_2$.
⋮ ⋮
$m$. $A \to B$.
$m+1$. $A$ (гипотеза).
$m+2$. $MP(m+1, m) \vdash B$.

В результате получим требуемый вывод. □

Приведём пример применения Теоремы дедукции.

**Пример 3.2.1.** Пусть $\Gamma = \{A, A \to B, B \to C\}$. Показать, что

$$\Gamma \vdash (A \to B) \to \big((B \to C) \to\ (A \to C)\big).$$

*Решение*. Построим требуемый вывод:

1. $A$ (гипотеза).
2. $A \to B$ (гипотеза).
3. $B \to C$ (гипотеза).
4. $MP(1,2) \vdash B$.

5. $MP(4,3) \vdash C$.
6. $\{A, A \to B, B \to C\} \vdash C$.
7. Упорядочим гипотезы из пункта 6 $A \to B, B \to C, A \vdash C$.
8. По теореме дедукции $A \to B, B \to C \vdash A \to C$.
9. По теореме дедукции $A \to B \vdash (B \to C) \to (A \to C)$.
10. По теореме дедукции $\vdash (A \to B) \to \big((B \to C) \to (A \to C)\big)$.

## 3.3 Производные правила вывода

Производные правила вывода, как и рассмотренные в Параграфе 3.1 правила подстановки и заключения, позволяют получать новые выводимые формулы. Они получаются с помощью правил подстановки и заключения, и поэтому являются производными от них.

Ниже даны наиболее важные производные правила вывода:

1) $\dfrac{A \to B, B \to C}{A \to C}$ – правило силлогизма (ПСИ);
2) $\dfrac{A \to (B \to C), B}{A \to C}$ – правило сечения (ПСЕ);
3) $\dfrac{A}{\int_{X_1, X_2, \ldots, X_n}^{B_1, B_2, \ldots, B_n}(A)}$ – правило одновременной подстановки;
4) $\dfrac{A_1, A_2, \ldots, A_n, A_1 \to (A_2 \to (\ldots (A_n \to B))}{B}$ – правило сложного заключения (ПСЗ);
5) $\dfrac{B_1, B_2, \ldots, B_n \vdash A}{B_1 \to (B_2 \to (\ldots (B_n \to A))}$ – обобщение теоремы дедукции;
6) $\dfrac{A \to B, \bar{B}}{\bar{A}}$ – Modus Tollens (MT), рассуждение от противного;
7) $\dfrac{A \to B}{\bar{B} \to \bar{A}}, \dfrac{\bar{A} \to \bar{B}}{B \to A}$ – правило контрапозиции (ПК);
8) $\dfrac{A \to \bar{\bar{B}}}{A \to B}, \dfrac{\bar{\bar{A}} \to B}{A \to B}$ – правило снятия двойного отрицания (ПСДО);
9) $\dfrac{A \to (B \to C)}{B \to (A \to C)}$ – правило перестановки посылок (ППП);
10) $\dfrac{A \to (B \to C)}{A \& B \to C}$ – правило объединения посылок (ПОП);

11) $\dfrac{A\&B\to C}{A\to(B\to C)}$ – правило разъединения посылок (ПРП);

12) $\dfrac{H\vdash A;\ H\vdash B}{H\vdash A\&B}$ – правило введения конъюнкции (ПВК);

13) $\dfrac{H,A\vdash C;\ H,B\vdash C}{H,A\vee B\vdash C}$ – правило введения дизъюнкции (ПВД);

14) $\dfrac{H\vdash A}{H,W\vdash A}$;

15) $\dfrac{H,C\vdash A;\ H\vdash C}{H\vdash A}$;

16) $\dfrac{H,C\vdash A;\ W\vdash C}{H,W\vdash A}$.

Докажем первые три производные правил вывода. Остальные правила доказываются аналогичным образом.

**Теорема 3.3.1. (Правило силлогизма)** $A \to B, B \to C \vdash A \to C$.

*Доказательство.* Построим вывод:

1. $A \to B$ (гипотеза).
2. $B \to C$ (гипотеза).
3. $A$ (гипотеза).
4. $MP(3,1) \vdash B$.
5. $MP(4,2) \vdash C$.
6. $A \to B, B \to C, A \vdash C$.
7. По теореме дедукции $A \to B, B \to C \vdash A \to C$.

Теорема доказана. □

**Теорема 3.3.2. (Правило сечения)** $A \to (B \to C), B \vdash A \to C$.

*Доказательство.* Построим вывод:

1. $A \to (B \to C)$ (гипотеза).
2. $B$ (гипотеза).
3. $A$ (гипотеза).
4. $MP(3,1) \vdash B \to C$.
5. $MP(2,4) \vdash C$.
6. $A \to (B \to C), B, A \vdash C$.
7. По теореме дедукции $A \to (B \to C), B \vdash A \to C$.

Теорема доказана. □

**Теорема 3.3.3. (Правило одновременной подстановки)** *Пусть $A$ есть выводимая формула, $X_1, X_2, \dots, X_n$ – переменные, а $B_1, B_2, \dots, B_n$ – любые формулы исчисления высказываний. Тогда результат одновременной подстановки в формулу $A$ вместо $X_1, X_2, \dots, X_n$ соответственно формул $B_1, B_2, \dots, B_n$ является выводимой формулой.*

*Доказательство.* Если формулы $B_1, B_2, \dots, B_n$ не содержат переменных $X_1, X_2, \dots, X_n$, то порядок, в котором будут происходить подстановки, не имеет значения. В противном случае введём вспомогательные высказывательные переменные $C_1, C_2, \dots, C_n$, обладающие следующими свойствами:

1) эти переменные попарно отличны друг от друга;
2) они отличаются от всех переменных, входящих в формулу $A$;
3) они отличаются от всех переменных, входящих в $B_1, B_2, \dots, B_n$;
4) они отличаются от всех переменных $X_1, X_2, \dots, X_n$.

По условию теоремы $\vdash A$, т. е. $A$ – выводимая формула. По правилу подстановки выводимой будет и формула $\int_{X_1}^{C_1}(A)$. Так как $\vdash \int_{X_1}^{C_1}(A)$, выводимой будет формула $\int_{X_2}^{C_2}\left(\int_{X_1}^{C_1}(A)\right) = \int_{X_1,X_2}^{C_1,C_2}(A)$ и т. д. В результате $\vdash \int_{X_1,X_2,\dots,X_n}^{C_1,C_2,\dots,C_n}(A)$ $(*)$. Имея в виду выводимость формулы $(*)$, выводимой будет $\int_{C_1}^{B_1}\left(\int_{X_1,X_2,\dots,X_n}^{C_1,C_2,\dots,C_n}(A)\right)$, т. е. $\vdash \int_{X_1,X_2,\dots,X_n}^{B_1,C_2,\dots,C_n}(A)$. Продолжая вышеуказанный процесс, в итоге выводимой будет и формула $\int_{X_1,X_2,\dots,X_n}^{B_1,B_2,\dots,B_n}(A)$. □

**Пример 3.3.1.** Показать выводимость формулы

$$(A \vee \bar{A}) \longrightarrow \big((A \to \bar{A}) \longrightarrow \bar{A}\big).$$

*Решение*. Построим вывод для формулы.

1. $(A \to C) \to \Big((B \to C) \to \big((A \vee B) \to C\big)\Big)$ (аксиома III. 3).
2. $\int_{B,C}^{\bar{A},\bar{A}}(1) \vdash (A \to \bar{A}) \to \Big((\bar{A} \to \bar{A}) \to \big((A \vee \bar{A}) \to \bar{A}\big)\Big)$.
3. ППП$(2) \vdash (\bar{A} \to \bar{A}) \to \Big((A \to \bar{A}) \to \big((A \vee \bar{A}) \to \bar{A}\big)\Big)$.
4. $A \to A$ (Теорема 3.1.1).
5. $\int_{A}^{\bar{A}}(4) \vdash \bar{A} \to \bar{A}$.
6. $MP(5,3) \vdash (A \to \bar{A}) \to \big((A \vee \bar{A}) \to \bar{A}\big)$.
7. ППП$(6) \vdash (A \vee \bar{A}) \longrightarrow \big((A \to \bar{A}) \longrightarrow \bar{A}\big)$.

Ниже приведём пример доказательства эквивалентности формул в исчислении высказываний. Формулу $A \leftrightarrow B$ можно представить как:

$$(A \to B)\&(B \to A).$$

Согласно правилу введения конъюнкции:

$$A \to B, A \to B \vdash (A \to B)\&(B \to A).$$

Следовательно, для доказательства эквивалентности $A \leftrightarrow B$ нужно доказать выводимость формул $A \to B$ и $B \to A$.

**Пример 3.3.2.** Показать, что $\vdash (A\&A) \leftrightarrow A$.

*Решение*. Построим вывод для формул $(A\&A) \to A$ и $A \to (A\&A)$. Таким образом докажем, что выводима формула $(A\&A) \leftrightarrow A$.

a) $(A\&A) \to A$

1. $(A\&B) \to A$ (аксиома II. 1).
2. $\int_B^A (1) \vdash (A\&A) \to A$.

b) $A \to (A\&A)$

1. $(A \to B) \to \Big((A \to C) \to \big(A \to (B\&C)\big)\Big)$ (аксиома II. 3).
2. $\int_{B,C}^{A,A} (1) \vdash (A \to A) \to \Big((A \to A) \to \big(A \to (A\&A)\big)\Big)$.
3. $MP(A \to A, 2) \vdash (A \to A) \to \big(A \to (A\&A)\big)$.
4. $MP(A \to A, 3) \vdash A \to (A\&A)$.

c) $(A\&A) \to A, A \to (A\&A) \vdash (A\&A) \leftrightarrow A$

## 3.4 Связь между алгеброй высказываний и исчислением высказываний

Запишем две теоремы, которые устанавливают связь между алгеброй высказываний и исчислением высказываний.

**Теорема 3.4.1.** *Каждая формула, выводимая в исчислении высказываний, является тождественно истинной в алгебре высказываний.*

*Доказательство.* Докажем три положения:

1) Каждая аксиома исчисления высказываний является тождественно истинной формулой в алгебре высказываний. Это можно проверить используя таблицу истинности, перебрав все возможные значения входящих в неё переменных.

2) Правило подстановки, применённое к тождественно истинной формуле, приводит к тождественно истинной формуле.

Пусть $X_1, X_2, \dots, X_n, X$ – все пропозициональные переменные, входящие в формулы $A$ и $B$, и пусть $\alpha_1, \alpha_2, \dots, \alpha_n, \alpha$ – произвольный фикси-

рованный набор значений переменных, состоящий из 0 и 1. Тогда если $B(\alpha_1, \alpha_2, \dots, \alpha_n, \alpha) = \beta$, то

$$\int_X^B A(\alpha_1, \alpha_2, \dots, \alpha_n, \alpha) = A(\alpha_1, \alpha_2, \dots, \alpha_n, \beta). \qquad (*)$$

Покажем, что если $A$ тождественно истинная формула, то и $\int_X^B A$ также тождественно истинная формула. Предположим, что существует такой набор $(\alpha_1^0, \alpha_2^0, \dots, \alpha_n^0, \alpha^0) = \tilde{\alpha}$, что $\int_X^B A(\tilde{\alpha}) = 0$ и $B(\tilde{\alpha}) = \beta^0$. Тогда, согласно $(*)$ получим $A(\alpha_1^0, \alpha_2^0, \dots, \alpha_n^0, \beta^0) = 0$, а это противоречит тому, что $A$ является тождественно истинной формулой.

3) Правило заключения, применённое к тождественно истинным формулам, приводит к тождественно истинной формуле. То есть, если формулы $C$ и $C \to A$ являются тождественно истинными, то и формула $A$ – тождественно истинна.

Предположим, что $A$ не является тождественно истинной формулой. Тогда существует такой набор $(\alpha_1^0, \alpha_2^0, \dots, \alpha_n^0) = \tilde{\alpha}$, что $A(\tilde{\alpha}) = 0$. Но при этом

$$(C \to A)(\tilde{\alpha}) = C(\tilde{\alpha}) \to A(\tilde{\alpha}) = 1 \to 0 = 0,$$

что противоречит тождественной истинности формулы $C \to A$. □

Для доказательства обратного утверждения, сначала требуется произвести доказательство следующих пяти лемм.

**Лемма 3.4.1.** $\vdash \overline{A \vee B} \to \bar{A}\,\&\bar{B}$.

*Доказательство.* Построим вывод формулы $\overline{A \vee B} \to \bar{A}\,\&\bar{B}$.

1. $(A \to B) \to \Big((A \to C) \to \big(A \to (B\&C)\big)\Big)$ (аксиома II. 3).
2. $\int_{A,B,C}^{\overline{A\vee B},\bar{A},\bar{B}}(1) \vdash (\overline{A \vee B} \to \bar{A}) \to \Big((\overline{A \vee B} \to \bar{B}) \to \big(\overline{A \vee B} \to (\bar{A}\&\bar{B})\big)\Big)$.
3. $(A \to B) \to (\bar{B} \to \bar{A})$ (аксиома IV. 1).
4. $\int_B^{A\vee B}(3) \vdash (A \to A \vee B) \to (\overline{A \vee B} \to \bar{A})$.
5. $A \to A \vee B$ (аксиома III. 1).
6. $MP(5,4) \vdash \overline{A \vee B} \to \bar{A}$.
7. $MP(6,2) \vdash (\overline{A \vee B} \to \bar{B}) \to (\overline{A \vee B} \to \bar{A}\&\bar{B})$.
8. $\int_{A,B}^{B,A\vee B}(3) \vdash (B \to A \vee B) \to (\overline{A \vee B} \to \bar{B})$.
9. $B \to A \vee B$ (аксиома III. 2).
10. $MP(9,8) \vdash \overline{A \vee B} \to \bar{B}$.
11. $MP(10,7) \vdash \overline{A \vee B} \to \bar{A}\&\bar{B}$.

Лемма доказана. □

**Лемма 3.4.2.** $\vdash A \to (\bar{A} \to B)$.

*Доказательство.* Построим вывод:

1. $A \to (B \to A)$ (аксиома I. 1).
2. $\int_B^{\bar{B}}(1) \vdash A \to (\bar{B} \to A)$.
3. $(A \to B) \to (\bar{B} \to \bar{A})$ (аксиома IV. 1).
4. $\int_{A,B}^{\bar{B},A}(3) \vdash (\bar{B} \to A) \to \left(\bar{A} \to \bar{\bar{B}}\right)$.
5. ПСИ(2,4) $\vdash A \to \left(\bar{A} \to \bar{\bar{B}}\right)$.
6. ПОП(5) $\vdash A\&\bar{A} \to \bar{\bar{B}}$.
7. ПСДО(6) $\vdash A\&\bar{A} \to B$.
8. ПРП(7) $\vdash A \to (\bar{A} \to B)$.

Лемма доказана. □

**Лемма 3.4.3.** $\vdash A \vee \bar{A}$.

*Доказательство.* Построим вывод:

1. $\overline{\overline{A \vee B}} \to \bar{A}\,\&\bar{B}$ (Лемма 3.4.1).
2. $\int_B^{\bar{A}}(1) \vdash \overline{\overline{A \vee \bar{A}}} \to \bar{A}\,\&\bar{\bar{A}}$.
3. $A \to (\bar{A} \to B)$ (Лемма 3.4.2).
4. $\int_{A,B}^{\bar{A},\bar{B}}(3) \vdash \bar{A} \to \left(\bar{\bar{A}} \to \bar{B}\right)$.
5. ПОП(4) $\vdash \bar{A}\&\bar{\bar{A}} \to \bar{B}$.
6. ПСИ(2,5) $\vdash \overline{\overline{A \vee \bar{A}}} \to \bar{B}$.
7. ПК(6) $\vdash \bar{\bar{B}} \to \overline{\overline{\overline{A \vee \bar{A}}}}$.
8. ПСДО(7) $\vdash B \to A \vee \bar{A}$.
9. $\int_B^{B \to A \vee \bar{A}}(8) \vdash (B \to A \vee \bar{A}) \to A \vee \bar{A}$.
10. $MP(8,9) \vdash A \vee \bar{A}$.

Лемма доказана. □

**Лемма 3.4.4.** $\bar{A} \vdash A \to B$.

*Доказательство.* Построим вывод:

1. $A \to (B \to A)$ (аксиома $A_1$).
2. $\int_{A,B}^{\bar{A},\bar{B}}(1) \vdash \bar{A} \to (\bar{B} \to \bar{A})$.
3. $\bar{A}$ (гипотеза).

4. $MP(3,2) \vdash \bar{B} \to \bar{A}$.
5. ПК$(4) \vdash \bar{\bar{A}} \to \bar{\bar{B}}$.
6. ПСДО$(5) \vdash A \to B$.

Лемма доказана. □

**Лемма 3.4.5.** *Пусть $X_1, X_2, \dots, X_n$ – все пропозициональные переменные, которые входят в формулу $A$ и $\alpha_1, \alpha_2, \dots, \alpha_n$ – произвольный фиксированный набор значений этих переменных. Тогда*

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash A^{\beta},$$

*где* $X_i^{\alpha_i} = \begin{cases} X_i, & \alpha_i = 1, \\ \bar{X}_i, & \alpha_i = 0, \end{cases} i = \overline{1,n}$ *и* $A^{\beta} = \begin{cases} A, & \beta = 1, \\ \bar{A}, & \beta = 0, \end{cases} \beta = A(\alpha_1, \alpha_2, \dots, \alpha_n).$

*Доказательство.* Доказательство проведём индукцией по числу вхождений логических связок в формулу $A$. Число логических связок обозначим через $k$.

Если $k = 0$, то $A$ представляет собой пропозициональную переменную $X_i$, где $1 \le i \le n$, и утверждение леммы сводится к $X_i \vdash X_i$ при $\alpha_i = 1$ и $\bar{X}_i \vdash \bar{X}_i$ при $\alpha_i = 0$.

Допустим, что утверждение леммы верно для любой формулы с числом связок $k$, докажем его для $k + 1$. Рассмотрим три случая.

I. Формула $A$ имеет вид $\bar{B}$. Положим $\beta_1 = B(\alpha_1, \alpha_2, \dots, \alpha_n)$. Тогда $\beta = \bar{\beta}_1$. Так как $B$ содержит на одну связку меньше, чем $A$, по предположению индукции справедливо

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B^{\beta_1}.$$

Если $\beta = 1$, то $\beta_1 = 0$ и

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash \bar{B}.$$

Если $\beta = 0$, то $\beta_1 = 1$ и

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B.$$

В силу аксиомы IV. 2 $(A \to \bar{\bar{A}})$ и обратной теоремы дедукции при подстановке $B$ вместо $A$ получим $B \vdash \bar{\bar{B}}$. Поэтому исходя из транзитивности выводимости

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash \bar{\bar{B}}.$$

Так как $A^1 = \bar{B}$ и $A^0 = \bar{\bar{B}}$, то

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash A^{\beta}.$$

Таким образом, в этом случае утверждение леммы доказано.

II. Формула $A$ имеет вид $B \to C$. Положим $\beta_1 = B(\alpha_1, \alpha_2, \dots, \alpha_n)$ и $\beta_2 = C(\alpha_1, \alpha_2, \dots, \alpha_n)$. Тогда $\beta = \beta_1 \to \beta_2$. Формулы $B$ и $C$ содержат меньше связок, чем формула $A$, и в силу индуктивного предположения справедливо

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B^{\beta_1} \text{ и } X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash C^{\beta_2}$$

(в случае, когда формулы $B$ или $C$ содержат не все переменные $X_1, X_2, \dots, X_n$, в левые части этих выражений добавляются лишние гипотезы).

Если $\beta = 0$, то $\beta_1 = 1$, $\beta_2 = 0$ и

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B \text{ и } X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash \bar{C}.$$

Очевидно, что $B, B \to C \vdash C$. В силу теоремы дедукции $B \vdash (B \to C) \to C$. Отсюда по правилу контрапозиции вытекает $B \vdash \bar{C} \to \overline{B \to C}$. Согласно обратной теореме дедукции получим $B, \bar{C} \vdash \overline{B \to C}$. Поэтому

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash \overline{B \to C}.$$

Если $\beta = 1$ и $\beta_1 = 0$, то

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash \bar{B}.$$

Принимая во внимание Лемму 3.4.4 ($\bar{A} \vdash A \to B$), а также подставляя $C$ вместо $B$ и $B$ вместо $A$, получим

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B \to C.$$

Если $\beta = 1$ и $\beta_1 = 1$, то $\beta_2 = 1$ и

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B \text{ и } X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash C.$$

Исходя из аксиомы $A_1$ $\big(A \to (B \to A)\big)$ при подстановке $C$ вместо $A$ получим $C \to (B \to C)$. По правилу $MP(C, C \to (B \to C))$ получим $B \to C$. Поэтому

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash B \to C.$$

Так как $A^1 = B \to C$ и $A^0 = \overline{B \to C}$, то

$$X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n} \vdash A^{\beta}.$$

III. В случае, когда формула $A$ имеет вид $B\&C$ или $B \vee C$, доказательство производится аналогично первым двум случаям.

Таким образом, лемма полностью доказана. □

Теперь можем перейти к доказательству утверждения обратного Теореме 3.4.1.

**Теорема 3.4.2.** *Каждая тождественно истинная формула алгебры высказываний выводима в исчислении высказываний.*

*Доказательство.* Пусть $A$ – тождественно истинная формула в алгебре высказываний, а $X_1, X_2, \dots, X_n$ – все переменные, входящие в $A$. Для любого набора $(\alpha_1, \alpha_2, \dots, \alpha_n)$, где $\alpha_i \in \{0,1\}$ и $X_i^{\alpha_i} = \begin{cases} X_i, & \alpha_i = 1, \\ \bar{X}_i & \alpha_i = 0, \end{cases} i = \overline{1,n}$, получим $A\left(X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n}\right) = 1$. Пусть $H_n = \left\{X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_n^{\alpha_n}\right\}$, тогда, согласно Лемме 3.4.5, можем записать $H_n \vdash A$. В продолжении, при $\alpha_n = 1$ и $\alpha_n = 0$ имеем соответственно $H_{n-1}, X_n \vdash A$ и $H_{n-1}, \bar{X}_n \vdash A$, где $H_{n-1} = \left\{X_1^{\alpha_1}, X_2^{\alpha_2}, \dots, X_{n-1}^{\alpha_{n-1}}\right\}$. По правилу введения дизъюнкции получим $H_{n-1}, X_n \vee \bar{X}_n \vdash A$. Согласно Лемме 3.4.3 формула $X_n \vee \bar{X}_n$ выводима в исчислении высказываний и её можно опустить из совокупности формул $H_{n-1}, X_n \vee \bar{X}_n$. Значит, $H_{n-1} \vdash A$. Таким образом, мы исключили $X_n$ из списка гипотез. Повторяя эту процедуру, мы можем показать, что $H_{n-2} \vdash A$ и так далее, пока не получим $\vdash A$. □

## 3.5 Проблемы аксиоматического исчисления высказываний

Всякая аксиоматическая теория для её обоснования требует рассмотрения четырёх проблем: разрешимости, непротиворечивости, полноты и независимости. Рассмотрим исчисление высказываний как формальную аксиоматическую теорию с точки зрения этих проблем.

Начнём с определений.

**Определение 3.5.1.** *Логическое исчисление называется* ***разрешимым****, если существует эффективный метод (алгоритм), позволяющий для любой формулы исчисления за конечное число шагов определить, является ли она выводимой (тождественно истинной) или нет.*

**Определение 3.5.2.** *Логическое исчисление называется* ***непротиворечивым****, если в нём не выводимы никакие две формулы, из которых одна является отрицанием другой.*

**Определение 3.5.3.** *Логическое исчисление называется* ***полным в узком смысле****, если добавление к списку её аксиом любой не выводимой в исчислении формулы в качестве новой аксиомы приводит к противоречивому исчислению.*

**Определение 3.5.4.** *Логическое исчисление называется* ***полным в широком смысле****, если всякая тождественно истинная формула в ней выводима.*

**Определение 3.5.5.** *Аксиома называется* ***независимой*** *от всех остальных аксиом исчисления, если она не может быть выведена из остальных аксиом. Система аксиом исчисления называется* ***независимой****, если каждая аксиома системы независима.*

Считается, что аксиоматическая теория удовлетворительна с точки зрения общих требований, предъявляемых к аксиоматическим теориям, если она разрешима, непротиворечива, полна (в узком и широком смыслах) и обладает независимой системой аксиом.

**Теорема 3.5.1.** *Исчисление высказываний разрешимо.*

*Доказательство.* Пусть $A$ – произвольная формула исчисления высказываний, зависящая от $n$ переменных. Вычислим значение формулы $A$ на всех наборах значений, входящих в неё переменных. Заметим, что каждая переменная принимает значение 0 или 1 и, следовательно, имеются $2^n$ всевозможных комбинаций значений этих переменных. Если на всех наборах $A = 1$, то формула $A$ общезначима. Иначе формула $A$ не общезначима. □

Легко понять, что на практике проверить общезначимость формулы бывает очень сложно в случае, если формула зависит от большого числа переменных. Для определения истинности значений формулы может понадобиться слишком много машинного времени.

**Теорема 3.5.2.** *Исчисление высказываний непротиворечиво.*

*Доказательство.* Докажем, что в исчислении высказываний нет такой формулы $A$, для которой выводимыми являются формулы $A$ и $\bar{A}$.

Пусть $A$ – произвольная формула исчисления высказываний. Если $A$ выводима, то по Теореме 3.4.1 она тождественно истинна, и, значит, что $\bar{A}$ – тождественно ложная формула, и поэтому не выводима. □

**Теорема 3.5.3.** *Исчисление высказываний полно в узком смысле.*

*Доказательство.* Пусть $A$ – произвольная невыводимая формула исчисления высказываний, а $X_1, X_2, \dots, X_n$ – полный перечень входящих в неё переменных. Так как формула $A$ не выводима, то она не является тождественно истинной. Следовательно, существует набор значений переменных $\alpha_1, \alpha_2, \dots, \alpha_n$ такой, что $A(\alpha_1, \alpha_2, \dots, \alpha_n) = 0$.

Пусть $B_1, B_2, \dots, B_n$ – любой тождественно истинный набор формул, зависящих от тех же переменных $X_1, X_2, \dots, X_n$. Рассмотрим набор $B_1^{\alpha_1}, B_2^{\alpha_2}, \dots, B_n^{\alpha_n}$, где $B_i^{\alpha_i} = \begin{cases} B_i, & \alpha_i = 1, \\ \bar{B}_i, & \alpha_i = 0, \end{cases}$ $i = \overline{1,n}$. Осуществим подстановку $\int_{X_1,\dots,X_n}^{B_1^{\alpha_1},\dots,B_n^{\alpha_n}}(A)$. В результате получим формулу $A(B_1^{\alpha_1}, B_2^{\alpha_2}, \dots, B_n^{\alpha_n})$,

которая является тождественно ложной. Из этого всего следует, что $\overline{A\left(B_1^{\alpha_1}, B_2^{\alpha_2}, \dots, B_n^{\alpha_n}\right)}$ – это тождественно истинная формула и, значит, она выводима.

Если к списку аксиом исчисления высказываний присоединить в качестве новой аксиомы формулу $A$, то в новом исчислении высказываний оказываются выводимыми формулы $A(B_1^{\alpha_1}, B_2^{\alpha_2}, \dots, B_n^{\alpha_n})$ и $\overline{A(B_1^{\alpha_1}, B_2^{\alpha_2}, \dots, B_n^{\alpha_n})}$, что приводит к противоречивому исчислению. □

**Теорема 3.5.4.** *Исчисление высказываний полно в широком смысле.*

Справедливость этой теоремы следует из Теоремы 3.4.2.

Для доказательства независимости системы аксиом исчисления высказываний воспользуемся следующим утверждением.

**Утверждение 3.5.1.** *Аксиома $A$ данной формальной аксиоматической теории независима от остальных, если все остальные аксиомы обладают некоторым свойством $P$, которое сохраняется правилами выводимости, а аксиома $A$ не обладает этим свойством $P$.*

**Теорема 3.5.5.** *Система аксиом исчисления высказываний независима.*

*Доказательство.* В качестве свойства $P$ возьмем свойство формулы быть тождественно истинной.

1) Возьмём, например, аксиому $A \to A \vee B$ (аксиома III. 1) и докажем её независимость от остальных аксиом исчисления высказываний. Для этого по-новому определим дизъюнкцию:

| $A$ | $B$ | $A \vee B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

При таком определении $A \vee B = B$ $(*)$. Нетрудно видеть, что все аксиомы исчисления высказываний, не содержащие связку $\vee$, не изменят своих истинных значений, т. е. останутся тавтологиями. Лишь аксиомы III. 1, III. 2 и III. 3 содержат знак дизъюнкции $\vee$, поэтому проверим их на свойство $P$, т. е. являются ли они тавтологиями.

Аксиома III. 3 перепишется с учётом $(*)$ так:

$$(A \to C) \to ((B \to C) \to (B \to C)),$$

откуда сразу следует, что III. 3 является тавтологией.

Аналогично, III. 2 перепишется с учётом $(*)$ как $B \to B$, а значит, очевидно, является тавтологией. С другой стороны, III. 1 с учётом $(*)$

переписывается как $A \to B$, а значит, не является тавтологией, так как принимает значение 0, когда $A = 1$ и $B = 0$. Следовательно, III. 1 независима от остальных аксиом.

2) Проверим независимость аксиомы $A\&B \to A$ (аксиома II. 1). Для этого по-новому определим конъюнкцию:

| $A$ | $B$ | $A\&B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Тогда $A\&B = B$ и прямая проверка показывает, что все аксиомы, кроме II. 1, в новом определении являются тавтологиями. Следовательно, II. 1 независима.

3) Подобным образом доказывается независимость всех остальных аксиом. Например, независимость аксиомы IV. 1 доказывается посредством нового определения отрицания:

| $x$ | $\bar{x}$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

Теорема доказана. □

## 3.6 Задачи для самостоятельного решения

1. Применяя только правило подстановки, доказать, что выводимы следующие формулы:
   a) $A\&B \to A\&B \vee C$;
   b) $(A \to A) \to \big((A \to B\&C) \to (A \to A\&B\&C)\big)$;
   c) $\big(A\&B \to (C \to B\&C)\big) \to \big((A\&B \to C) \to (A\&B \to B\&C)\big)$;
   d) $\big((A \vee \bar{B}) \to C\big) \to \big(\bar{C} \to \overline{A \vee \bar{B}}\big)$.
2. Доказать выводимость формул в исчислении высказываний:
   a) $A \vee A \to A$;
   b) $(A \to \bar{A}) \to \bar{A}$;
   c) $\bar{A} \vee \bar{B} \to \overline{A\&B}$;
   d) $\bar{A}\&\bar{B} \to \overline{A \vee B}$;
   e) $(A \to \bar{B}) \to (B \to \bar{A})$;
   f) $\overline{A \to C} \to \overline{\big(A \to (B \to C)\big)\&(A \to B)}$;

g) $A \to \big((B \to B) \to (C \to A)\big)$;

h) $A \to \big(B \to (A \to B)\big)$;

i) $(A \to B)\&\bar{B} \to \bar{A}$;

j) $A \to \big((A \to B) \to B\big)$;

k) $(A \to B) \to \big((B \to C) \to (A \to C)\big)$;

l) $\big(A \to (A \to B)\big) \to (A \to B)$;

m) $\overline{\overline{(A \to B) \to A}} \to \bar{A}$;

n) $\overline{\bar{C} \to \bar{\bar{B}}} \to \bar{C}$.

3. Доказать, что имеют место следующие выводимости, построив соответствующие выводы из гипотез:

a) $\bar{A} \vdash A \to \bar{\bar{\bar{B}}}$;

b) $A \to \bar{\bar{\bar{B}}} \vdash A \to B$;

c) $\bar{A} \to \bar{B} \vdash B \to A$;

d) $A \to B, \bar{A} \to B \vdash B$;

e) $A \to (B \to C) \vdash B \to (A \to C)$;

f) $A \to (A \to B) \vdash A \to B$;

g) $\bar{A} \to B, \bar{A} \to \bar{B} \vdash B$;

h) $\bar{A} \to \bar{B} \vdash B$.

4. Используя теорему дедукции, доказать выводимость формул:

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

b) $A \to (\bar{B} \to \overline{A \to B})$;

c) $\bar{A} \to \big((\bar{B} \to A) \to B\big)$;

d) $(A \to B) \to \big((\bar{A} \to B) \to B\big)$;

e) $(A \to B) \to \big((A \vee C) \to (B \vee C)\big)$.

5. Доказать эквивалентность формул:

a) $A \vee B \leftrightarrow A$;

b) $A\&(A \vee B) \leftrightarrow A$;

c) $A \vee (A\&B) \leftrightarrow A$;

d) $A \vee B \leftrightarrow B \vee A$;

e) $(A \to B) \leftrightarrow (\bar{A} \vee B)$;

f) $(A \to \bar{A}) \leftrightarrow \bar{A}$;

g) $A \leftrightarrow B \vdash (A \vee C) \leftrightarrow (B \vee C)$;

h) $\{A \leftrightarrow B, B \leftrightarrow C\} \vdash \bar{A} \leftrightarrow \bar{C}$.

# Глава 4
# Логика предикатов

## 4.1 Предикаты и операции над ними

Ранее была рассмотрена логика высказываний, которая является самым простым и вместе с тем очень важным разделом математической логики. В рамках логики высказываний можно описывать и анализировать правильность очень многих рассуждений, но недостаточность средств логики высказываний состоит в том, что элементарные высказывания в логике высказываний рассматриваются как целые, нерасчленимые величины, без внутренней структуры. Для более глубокого исследования рассуждений используется логика предикатов.

Как было сказано ранее, под высказыванием мы понимаем повествовательное предложение, о котором можно утверждать, что оно истинно или ложно. В свою очередь, под предикатом мы понимаем повествовательное предложение с одной или несколькими переменными, которое превращается в высказывание при подстановке вместо переменных их значений.

Приведём примеры предикатов.

1) Число $x$ больше 5.
2) Студент $x$ сдал экзамен по дискретной математике на отлично.
3) $x + y \geq 3$.
4) Планета $z$ меньше планеты Земля.
5) Прямая $x$ параллельна прямой $y$.

Каждое из предложений 1 – 5 является предикатом, поскольку подставив вместо переменных их значения, получатся истинные либо ложные высказывания. Так, например, при $x = 9$ первое предложение превращается в истинное высказывание, а при $x = 2$ – в ложное.

Фактически, предикат представляет собой функцию, которая каждому набору значений переменных ставит в соответствие истинное либо ложное высказывание. Напомним, что истинностное высказывание отождествляется с 1, а ложное высказывание – с 0.

**Определение 4.1.1.** ***Предикатом*** $P(x_1, x_2, \dots, x_n)$ *называется функция, определённая на некотором множестве* $M$ *и принимающая истинностные значения* 0 *и* 1. *По числу переменных предикаты называются* ***одноместными***, ***двухместными***, *...* , ***n-местными***.

Множество $M$, на котором определён $P(x_1, x_2, \dots, x_n)$, называется *областью определения* предиката, а множество

$$I_P = \{(x_1, x_2, \dots, x_n) \in M : P(x_1, x_2, \dots, x_n) = 1\}$$

называется *областью истинности* предиката.

Вообще говоря, $n$-местный предикат – это утверждение об объектах $x_1, x_2, \dots, x_n$, рассматриваемых как переменные. В результате замены переменных $x_1, x_2, \dots, x_n$ некоторыми значениями из области определения предикат $P(x_1, x_2, \dots, x_n)$ превращается в высказывание. Важно заметить, что любое высказывание можно рассматривать как 0-местный предикат.

**Определение 4.1.2.** *Предикат* $P(x_1, x_2, \dots, x_n)$, *определённый на множестве* $M$, *называется* ***общезначимым*** *или* ***тождественно истинным*** (***тождественно ложным***), *если* $I_P = M$ ($I_P = \emptyset$).

**Определение 4.1.3.** *Предикат* $P(x_1, x_2, \dots, x_n)$, *определённый на множестве* $M$, *называется* ***выполнимым*** (***опровержимым***), *если* $I_P \neq \emptyset$ ($I_P \neq M$).

Рассмотрим примеры предикатов, определённых на множестве действительных чисел $\mathbb{R}$:

1) $P(x) = (x^2 < 0)$. Легко убедиться в том, что $I_P = \emptyset$, так как для любого $a \in \mathbb{R}$ будет $P(a) = (a^2 < 0) = 0$ (ложь).
2) $Q(x, y) = (x^3 + y + 2 - y = x^3 + 2)$. Очевидно, что $I_Q = \mathbb{R} \times \mathbb{R}$, так как для любого набор $(a_1, a_2)$, для которого $a_1, a_2 \in \mathbb{R}$, будет $Q(a_1, a_2) = ({a_1}^3 + 2 = {a_1}^3 + 2) = 1$ (истина).

**Определение 4.1.4.** *Предикат* $P$ *называется* ***следствием*** *предиката* $Q$, *если* $I_Q \subseteq I_P$.

Пусть следующие три предиката определены на множестве натуральных чисел $\mathbb{N}$: $P(x) = (x$ – чётное число$)$, $Q(x) = (x$ кратно 2$)$, $R(x) = (x$ кратно 4$)$. Тогда:

1) $P(x)$ является следствием $Q(x)$ и наоборот. Кроме того, $P(x)$ и $Q(x)$ принимают одинаковые логические значения на области $\mathbb{N}$.

2) $Q(x)$ является следствием $R(x)$, но не наоборот. Кроме того, $Q(x)$ и $R(x)$ не принимают одинаковые значения на области $\mathbb{N}$.

Операции над предикатами определяются так же, как и над высказываниями. Дадим определение этих операций на примере $n$-местных предикатов.

**Определение 4.1.5.** *Пусть даны n-местные предикаты* $P(x_1, x_2, \dots, x_n)$ *и* $Q(x_1, x_2, \dots, x_n)$ *с областью определения* $M$. *Тогда:*

1) ***конъюнкция*** $R(x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n) \& Q(x_1, x_2, \dots, x_n)$ *есть n-местный предикат с областью определения* $M$ *и областью истинности* $I_R = I_P \cap I_Q$;
2) ***дизъюнкция*** $G(x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n) \vee Q(x_1, x_2, \dots, x_n)$ *есть n-местный предикат с областью определения* $M$ *и областью истинности* $I_G = I_P \cup I_Q$;
3) ***отрицание*** $\overline{P(x_1, x_2, \dots, x_n)}$ *есть n-местный предикат с областью определения* $M$ *и областью истинности* $M \backslash I_P$;
4) ***импликация*** $R(x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n) \longrightarrow Q(x_1, x_2, \dots, x_n)$ *есть n-местный предикат с областью определения* $M$ *и областью истинности* $(M \backslash I_P) \cup I_Q$, *что следует из* $P(x_1, x_2, \dots, x_n) \longrightarrow Q(x_1, x_2, \dots, x_n) \equiv \overline{P(x_1, x_2, \dots, x_n)} \vee Q(x_1, x_2, \dots, x_n)$.

В логике предикатов рассматриваются две дополнительные операции, которые превращают одноместный предикат в высказывание.

**Определение 4.1.6.** *Пусть* $P(x)$ *– предикат, определённый на некотором множестве* $M$. *Тогда под выражением* $\forall x P(x)$ *понимают высказывание, которое истинно тогда, когда* $P(x)$ *истинно для каждого элемента* $x \in M$ *и ложно в противном случае. Запись* $\forall x P(x)$ *читается: "для всякого* $x$ $P(x)$ *истинно". Символ* $\forall$ *называется* ***квантором всеобщности***.

**Определение 4.1.7.** *Пусть* $P(x)$ *– предикат, определённый на некотором множестве* $M$. *Тогда под выражением* $\exists x P(x)$ *понимают высказывание, которое истинно тогда, когда существует такой элемент* $x \in M$, *для которого* $P(x)$ *истинно, и ложно в противном случае. Запись* $\exists x P(x)$ *читается: "существует* $x$, *для которого* $P(x)$ *истинно". Символ* $\exists$ *называется* ***квантором существования***.

Ясно, что высказывание $\forall x P(x)$ истинно только в том случае, когда $P(x)$ является тождественно истинным предикатом, а высказывание $\exists x P(x)$ ложно только в том единственном случае, когда $P(x)$ есть тождественно ложный предикат.

Переменная $x$ в предикате $P(x)$ называют *свободной* (она может принимать различные значения из $M$), а в высказываниях $\forall x P(x)$ и $\exists x P(x)$ переменная $x$ называется *связанной* квантором $\forall$ и, соответственно, квантором $\exists$.

Кванторы $\forall$ и $\exists$ называются двойственными друг другу. Приписывание квантора слева к предикату называется *навешиванием* квантора на предикат. Кванторы можно навешивать и на многоместные предикаты. Так, например, навесив квантор $\forall x_i$, $1 \le i \le n$, на предикат $P(x_1, x_2, \dots, x_n)$, получаем $(n-1)$-местный предикат $\forall x_i P(x_1, x_2, \dots, x_n)$, зависящий от переменных $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$.

Приведём примеры навешивания кванторов на предикаты, определённые на множества натуральных чисел $\mathbb{N}$:

1) Пусть $P(x) = (x > 3)$. Тогда $\forall x P(x) = \forall x (x > 3)$ – ложное высказывание, а $\exists x P(x) = \exists x (x > 3)$ – истинное высказывание.
2) Пусть $Q(x, y) = (x + y = 3)$ – двухместный предикат. Тогда:
   a) $\forall x Q(x, y) = \forall x (x + y = 3)$ – одноместный предикат от переменной $y$. Найдем некоторые его значения:

$$\forall x Q(x, 2) = \forall x (x + 2 = 3) = 0, \ \forall x Q(x, 8) = \forall x (x + 8 = 3) = 0.$$

   b) $\exists y Q(x, y) = \exists y (x + y = 3)$ – одноместный предикат от переменной $x$. Найдем некоторые его значения:

$$\exists y Q(2, y) = \exists y (2 + y = 3) = 1, \ \exists y Q(8, y) = \exists y (8 + y = 3) = 0.$$

Если $P(x_1, x_2, \dots, x_n)$ – некоторый $n$-местный предикат, определённый на множестве $M = M_1 \times M_2 \times \dots \times M_n$, где $M_1$, $M_2$, … , $M_n$ – области определения переменных $x_1, x_2, \dots, x_n$, и $M_i = \{a_i^1, a_i^2, \dots, a_i^m\}$ – некоторое конечное множество, то справедливы следующие выражения:

$$\forall x_i P(x_1, x_2, \dots, x_n) = P(x_1, \dots, x_{i-1}, a_i^1, x_{i+1}, \dots, x_n) \wedge \dots$$
$$\dots \wedge P(x_1, \dots, x_{i-1}, a_i^m, x_{i+1}, \dots, x_n),$$
$$\exists x_i P(x_1, x_2, \dots, x_n) = P(x_1, \dots, x_{i-1}, a_i^1, x_{i+1}, \dots, x_n) \vee \dots$$
$$\dots \vee P(x_1, \dots, x_{i-1}, a_i^m, x_{i+1}, \dots, x_n).$$

Следовательно, кванторные операции можно рассматривать как обобщение операций конъюнкции и дизъюнкции.

## 4.2 Формулы логики предикатов

Аналогично логике высказываний, определим понятие формулы логики предикатов. Для этого сначала определим *алфавит логики предикатов*:

1) предметные переменные: $x, y, z, x_1, y_1, z_1, \dots, x_i, y_i, z_i, \dots$;
2) предметные константы: $a, b, c, a_1, b_1, c_1, \dots, a_i, b_i, c_i, \dots$;
3) высказывательные переменные: $A, B, C, A_1, B_1, \dots, A_i, B_i, \dots$;
4) предикатные символы: $P, Q, R, P_1, Q_1, R_1, \dots, P_i, Q_i, R_i, \dots$;
5) функциональные символы: $f, g, f_1, g_1, \dots, f_i, g_i, \dots$;
6) логические связки: &, $\vee$, $\longrightarrow$, $\neg$;
7) кванторные операции: $\forall$, $\exists$;
8) вспомогательные символы: круглые скобки, запятые.

*Формула логики предикатов* определяется индуктивно по следующей схеме:

1) Всякая высказывательная переменная является формулой (элементарной).
2) Если $P(\cdot,\cdot,\dots,\cdot)$ $n$-местный предикат, а $\alpha_1, \alpha_2, \dots, \alpha_n$ – предметные переменные, предметные константы или функциональные символы, то $P(\alpha_1, \alpha_2, \dots, \alpha_n)$ есть формула.
3) Если $P$ и $Q$ формулы, причём такие, что одна и та же предметная переменная не является в одной из них связанной, а в другой – свободной, то $P \vee Q$, $P \& Q$ и $P \longrightarrow Q$ есть формулы.
4) Если $P$ формула, то и $\bar{P}$ тоже формула.
5) Если $P$ формула, в которую предметная переменная $x$ входит свободно, то $\forall x P$ и $\exists x P$ являются формулами, причём предметная переменная входит в них связанно.
6) Всякое слово, отличное от названных в пунктах 1) – 5), не является формулой.

Например, если $P(x)$ и $Q(x,y)$ – одноместный и двухместный предикаты, а $A$ и $B$ – высказывательные переменные, то формулами будут: $A$, $P(x)$, $P(x) \vee \exists y Q(x,y)$, $\forall x P(x) \vee \exists x \forall y Q(x,y)$, $(\overline{Q(x,y)} \& A) \longrightarrow B$.

Заметим, что всякая формула логики высказываний является формулой логики предикатов, то есть, в общим и целом, язык логики предикатов является расширением языка логики высказываний.

**Определение 4.2.1.** *Две формулы логики предикатов $P$ и $Q$ называются* ***равносильными на области*** *$M$, если они принимают одинаковые логические значения при всех значениях входящих в них переменных, отнесенных к области $M$.*

**Определение 4.2.2.** *Две формулы логики предикатов $P$ и $Q$ называются* ***равносильными****, если они равносильны на всякой области.*

Как и в алгебре высказываний, для равносильных формул $P$ и $Q$ принято использовать обозначение $P \equiv Q$. Ниже рассмотрим основные равносильности логики предикатов.

Пусть $P$ и $Q$ – $n$-местные предикаты, а $A$ – высказывательная переменная. Приведём равносильности, связанные с применением кванторов.

Законы де Моргана для кванторов:

1) $\overline{\forall x_k P(x_1, x_2, \dots, x_n)} \equiv \exists x_k \overline{P(x_1, x_2, \dots, x_n)}$;
2) $\overline{\exists x_k P(x_1, x_2, \dots, x_n)} \equiv \forall x_k \overline{P(x_1, x_2, \dots, x_n)}$;
3) $\forall x_k P(x_1, x_2, \dots, x_n) \equiv \overline{\exists x_k \overline{P(x_1, x_2, \dots, x_n)}}$;
4) $\exists x_k P(x_1, x_2, \dots, x_n) \equiv \overline{\forall x_k \overline{P(x_1, x_2, \dots, x_n)}}$.

Законы пронесения кванторов через конъюнкцию и дизъюнкцию:

1) $\forall x_i P(x_1, x_2, \dots, x_n) \& \forall x_i Q(x_1, x_2, \dots, x_n) \equiv$
   $\equiv \forall x_i \big(P(x_1, x_2, \dots, x_n) \& Q(x_1, x_2, \dots, x_n)\big)$;
2) $\exists x_i P(x_1, x_2, \dots, x_n) \vee \exists x_i Q(x_1, x_2, \dots, x_n) \equiv$
   $\equiv \exists x_i \big(P(x_1, x_2, \dots, x_n) \vee Q(x_1, x_2, \dots, x_n)\big)$;
3) $A \& \forall x_i P(x_1, x_2, \dots, x_n) \equiv \forall x_i \big(A \& P(x_1, x_2, \dots, x_n)\big)$;
4) $A \vee \forall x_i P(x_1, x_2, \dots, x_n) \equiv \forall x_i \big(A \vee P(x_1, x_2, \dots, x_n)\big)$;
5) $A \& \exists x_i P(x_1, x_2, \dots, x_n) \equiv \exists x_i \big(A \& P(x_1, x_2, \dots, x_n)\big)$;
6) $A \vee \exists x_i P(x_1, x_2, \dots, x_n) \equiv \exists x_i \big(A \vee P(x_1, x_2, \dots, x_n)\big)$.

Законы пронесения кванторов через импликацию:

1) $\forall x_i \big(A \longrightarrow P(x_1, x_2, \dots, x_n)\big) \equiv A \longrightarrow \forall x_i P(x_1, x_2, \dots, x_n)$;
2) $\forall x_i (P(x_1, x_2, \dots, x_n) \longrightarrow A) \equiv \exists x_i P(x_1, x_2, \dots, x_n) \longrightarrow A$;
3) $\exists x_i (A \longrightarrow P(x_1, x_2, \dots, x_n)) \equiv A \longrightarrow \exists x_i P(x_1, x_2, \dots, x_n)$;
4) $\exists x_i (P(x_1, x_2, \dots, x_n) \longrightarrow A) \equiv \forall x_i P(x_1, x_2, \dots, x_n) \longrightarrow A$.

Кванторные законы:

1) $\forall x_i \forall x_j P(x_1, x_2, \dots, x_n) \equiv \forall x_j \forall x_i P(x_1, x_2, \dots, x_n)$;
2) $\exists x_i \exists x_j P(x_1, x_2, \dots, x_n) \equiv \exists x_j \exists x_i P(x_1, x_2, \dots, x_n)$;
3) $\forall y P(x_1, x_2, \dots, x_n) \equiv P(x_1, x_2, \dots, x_n)$;
4) $\exists y P(x_1, x_2, \dots, x_n) \equiv P(x_1, x_2, \dots, x_n)$;
5) $\forall x_i P(x_1, x_2, \dots, x_n) \equiv \forall y P(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)$;
6) $\exists x_i P(x_1, x_2, \dots, x_n) \equiv \exists y P(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)$.

Кроме того, все равносильности логики высказываний являются равносильностями логики предикатов.

Отметим, что формула $\forall x_i P(x_1, x_2, \dots, x_n) \vee \forall x_i Q(x_1, x_2, \dots, x_n)$ не равносильна формуле $\forall x_i\big(P(x_1, x_2, \dots, x_n) \vee Q(x_1, x_2, \dots, x_n)\big)$ и также формула $\exists x_i P(x_1, x_2, \dots, x_n) \& \exists x_i Q(x_1, x_2, \dots, x_n)$ не равносильна формуле $\exists x_i (P(x_1, x_2, \dots, x_n) \& Q(x_1, x_2, \dots, x_n))$. Однако справедливы следующие равносильности, в которых по факту происходит переименование переменной:

$$\forall x_i P(x_1, x_2, \dots, x_n) \vee \forall x_i Q(x_1, x_2, \dots, x_n) \equiv$$

$$\equiv \forall x_i P(x_1, x_2, \dots, x_n) \vee \forall y Q(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n) \equiv$$

$$\equiv \forall x_i\big(P(x_1, x_2, \dots, x_n) \vee \forall y Q(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)\big) \equiv$$

$$\equiv \forall x_i \forall y\big(P(x_1, x_2, \dots, x_n) \vee Q(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)\big),$$

$$\exists x_i P(x_1, x_2, \dots, x_n) \& \exists x_i Q(x_1, x_2, \dots, x_n) \equiv$$

$$\equiv \exists x_i P(x_1, x_2, \dots, x_n) \& \exists y Q(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n) \equiv$$

$$\equiv \exists x_i\big(P(x_1, x_2, \dots, x_n) \& \exists y Q(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)\big) \equiv$$

$$\equiv \exists x_i \exists y\big(P(x_1, x_2, \dots, x_n) \& Q(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)\big).$$

Вдобавок не равносильны формулы $\forall x_i \exists x_j P(x_1, x_2, \dots, x_n)$ и $\exists x_j \forall x_i P(x_1, x_2, \dots, x_n)$. К примеру, определим предикат

$$P(x, y) = (x \text{ является матерью } y).$$

Тогда получим высказывания:

$$\forall y \exists x P(x, y) = (\text{у каждого человека есть мать}),$$

$$\exists x \forall y P(x, y) = (\text{существует мать всех людей}).$$

Из чего можно заключить, что перестановка кванторов всеобщности и существования изменяет смысл высказывания и его логическое значение (первое высказывание истинно, а второе высказывание ложно).

**Пример 4.2.1.** Доказать методом равносильных преобразований тождественную ложность формулы

$$\exists x \exists y \Big(\big(P(x) \to P(y)\big) \& \big(P(x) \to \overline{P(y)}\big) \& P(x)\Big).$$

*Решение*. Воспользуемся равносильными преобразованиями.

$$\exists x \exists y \Big(\big(P(x) \to P(y)\big) \& \big(P(x) \to \overline{P(y)}\big) \& P(x)\Big) \equiv$$

$$\equiv \exists x \exists y \Big(\big(\overline{P(x)} \vee P(y)\big) \& \big(\overline{P(x)} \vee \overline{P(y)}\big) \& P(x)\Big) \equiv$$

$$\equiv \exists x \exists y \left( \left( \overline{P(x)} \vee P(y) \right) \& \left( \left( \overline{P(x)} \& P(x) \right) \vee \left( \overline{P(y)} \& P(x) \right) \right) \right) \equiv$$

$$\equiv \exists x \exists y \left( \left( \overline{P(x)} \vee P(y) \right) \& \left( \overline{P(y)} \& P(x) \right) \right) \equiv$$

$$\equiv \exists x \exists y \left( \left( \overline{P(x)} \& \overline{P(y)} \& P(x) \right) \vee \left( P(y) \& \overline{P(y)} \& P(x) \right) \right) \equiv$$

$$\equiv \exists x \exists y (0 \vee 0) \equiv 0.$$

**Пример 4.2.2.** Доказать тождественную истинность следующей формулы:

$$\forall x (P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x)).$$

*Решение*. Докажем тождественную истинность с помощью равносильных преобразований.

$$\forall x (P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x)) \equiv$$

$$\equiv \overline{\forall x \left( \overline{P(x)} \vee Q(x) \right)} \vee \overline{\forall x P(x)} \vee \forall x Q(x) \equiv$$

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

$$\equiv \exists x \left( P(x) \& \overline{Q(x)} \right) \vee \exists x \overline{P(x)} \vee \forall x Q(x) \equiv$$

$$\equiv \exists x \left( P(x) \& \overline{Q(x)} \vee \overline{P(x)} \right) \vee \forall x Q(x) \equiv$$

$$\equiv \exists x \left( \left( P(x) \vee \overline{P(x)} \right) \& \left( \overline{Q(x)} \vee \overline{P(x)} \right) \right) \vee \forall x Q(x) \equiv$$

$$\equiv \exists x \left( \overline{Q(x)} \vee \overline{P(x)} \right) \vee \forall x Q(x) \equiv$$

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

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

$$\equiv \exists x \overline{P(x)} \vee 1 \equiv 1.$$

**Пример 4.2.3.** Доказать (аргументировать) общезначимость формулы

$$\exists x \left( P(x) \& (A \to Q(x)) \right) \to \left( \forall x \left( P(x) \to \overline{Q(x)} \right) \to \bar{A} \right).$$

*Решение*. Докажем методом от противного. Обозначим данную формулу через $C$. Пусть $C = 0$. Тогда:

$$\exists x \left( P(x) \& \left( A \to Q(x) \right) \right) = 1, \qquad (*)$$

$$\forall x \left( P(x) \to \overline{Q(x)} \right) \to \bar{A} = 0. \qquad (**)$$

Из $(**)$ получаем $\forall x \left( P(x) \to \overline{Q(x)} \right) = 1$ $(***)$ и $\bar{A} = 0$. Откуда следует, что $A = 1$. Рассмотрим запись $(*)$, из которой следует, что существует $x = a$ такой, что

$$P(a)\&\big(A \to Q(a)\big) = 1 \Longrightarrow$$
$$\Longrightarrow P(a) = 1 \text{ и } A \to Q(a) = 1 \Longrightarrow$$
$$\Longrightarrow 1 \to Q(a) = 1 \Longrightarrow Q(a) = 1.$$

В результате мы получили $A = 1$, $P(a) = 1$, $Q(a) = 1$. При этом $P(a) \to \overline{Q(a)} = 0$, что противоречит записи $(***)$. Из чего следует, что формула $C$ общезначима.

## 4.3 Нормальные формы

**Определение 4.3.1.** *Формула логики предикатов имеет **нормальную форму**, если она содержит только операции конъюнкции, дизъюнкции и кванторные операции, а операция отрицания отнесена к простым высказывательным переменным и предикатам.*

Очевидно, что, используя равносильности алгебры высказываний и логики предикатов, каждую формулу логики предикатов можно привести к нормальной форме.

**Пример 4.3.1.** Привести к нормальной форме следующую формулу:

$$\big(\exists x P(x) \to \forall y Q(y)\big) \to R(z).$$

*Решение*. Произведём соответствующие преобразования:

$$\big(\exists x P(x) \to \forall y Q(y)\big) \to R(z) \equiv \overline{\exists x P(x) \to \forall y Q(y)} \vee R(z) \equiv$$
$$\equiv \overline{\overline{\exists x P(x)} \vee \forall y Q(y)} \vee R(z) \equiv \overline{\overline{\exists x P(x)}}\&\overline{\forall y Q(y)} \vee R(z) \equiv$$
$$\equiv \exists x P(x)\&\exists y \overline{Q(y)} \vee R(z).$$

**Определение 4.3.2.** *Формула логики предикатов имеет **предварённую нормальную форму** (ПНФ), если в ней кванторные операции либо полностью отсутствуют, либо они используются после всех операций алгебры логики, то есть предварённая нормальная форма формулы логики предикатов имеет вид:*

$$(\alpha x_1)(\alpha x_2) \dots (\alpha x_m) F(x_1, x_2, \dots, x_n), \qquad m \le n,$$

*где под $(\alpha x_i)$, $i = \overline{1,m}$, понимается один из кванторов $\forall x_i$ или $\exists x_i$, а формула $F$ кванторов не содержит.*

**Теорема 4.3.1.** *Всякая формула логики предикатов может быть приведена к предварённой нормальной форме.*

*Доказательство.* Будем считать, что формула уже приведена к нормальной форме и покажем, что её можно привести к ПНФ. Докажем теорему с помощью индукции по числу логических связок в формуле.

Если формула является элементарной, то она кванторов не содержит, и, очевидно, уже является ПНФ.

Предположим, что теорема справедлива для формул, содержащих не более $k$ логических связок, и докажем, что при этом предположении она будет справедлива и для формул, содержащих ровно $k+1$ знаков.

Пусть имеется формула $R$, которая содержит $k+1$ знаков операций. Проанализируем следующие случаи:

1) $R = (\alpha x)P$, где $(\alpha x)$ обозначает один из кванторов. Так как $P$ содержит $k$ операций, её можно считать приведённой к ПНФ и, следовательно, $(\alpha x)P$ представляет собой ПНФ. Аналогично если $R = \bar{P}$, тогда, используя законы де Моргана для кванторов, с лёгкостью можно получить ПНФ формулы $R$.

2) Пусть $R = P \vee Q$. По предположению индукции формулы $P$ и $Q$ приведены к предварённой нормальной форме. Переименуем в формуле $Q$ связанные предметные переменные так, чтобы в формулах $P$ и $Q$ все связанные предметные переменные были различными. При этом формулы $P$ и $Q$ могут быть записаны в виде:

$$P = (\alpha x_1)(\alpha x_2)\dots(\alpha x_m)P'(x_1, x_2, \dots, x_n), \qquad m \le n,$$
$$Q = (\alpha y_1)(\alpha y_2)\dots(\alpha y_k)Q'(y_1, y_2, \dots, y_l), \qquad k \le l.$$

Используя законы пронесения кванторов через дизъюнкцию и учитывая что в текущей записи формула $Q$ может считаться высказыванием по отношению к переменным $x_1, x_2, \dots, x_n$, запишем формулу $R$, вводя формулу $Q$ под знаки кванторов $(\alpha x_1)(\alpha x_2)\dots(\alpha x_m)$:

$$R = (\alpha x_1)(\alpha x_2)\dots(\alpha x_m)(P'(x_1, x_2, \dots, x_n) \vee$$
$$\vee (\alpha y_1)(\alpha y_2)\dots(\alpha y_k)Q'(y_1, y_2, \dots, y_l)).$$

Затем введём под знаки кванторов $(\alpha y_1)(\alpha y_2)\dots(\alpha y_k)$ формулу $P'(x_1, x_2, \dots, x_n)$. В результате для формулы $R$ получим предварённую нормальную форму:

$$R = (\alpha x_1)(\alpha x_2)\dots(\alpha x_m)(\alpha y_1)(\alpha y_2)\dots$$
$$\dots(\alpha y_k)\big(P'(x_1, x_2, \dots, x_n) \vee Q'(y_1, y_2, \dots, y_l)\big).$$

В случае, когда формула $R$ имеет вид $P\&Q$, доказательство проводится аналогично. □

**Пример 4.3.2.** Привести к ПНФ формулу $\forall x\exists y P(x,y)\&\overline{\exists x\forall y Q(x,y)}$.

*Решение*. Произведём эквивалентные преобразования:

$$\forall x\exists y P(x,y)\&\overline{\exists x\forall y Q(x,y)} \equiv \forall x\exists y P(x,y)\&\forall x\overline{\forall y Q(x,y)} \equiv$$
$$\equiv \forall x\exists y P(x,y)\&\forall x\exists y\overline{Q(x,y)} \equiv \forall x\big(\exists y P(x,y)\&\exists y\overline{Q(x,y)}\big) \equiv$$

$$\equiv \forall x\left(\exists y P(x,y) \& \exists z \overline{Q(x,z)}\right) \equiv \forall x \exists y \exists z \left(P(x,y) \& \overline{Q(x,z)}\right).$$

**Определение 4.3.3.** ***Сколемовская нормальная форма*** *(СНФ) – это предварённая нормальная форма, в которой используются только кванторы всеобщности и не содержатся кванторы существования.*

Сколемовская нормальная форма строится следующим образом:

1) Если квантору существования не предшествует ни один квантор всеобщности, то квантор существования вычёркивается, а связанная с ним предметная переменная заменяется на предметную постоянную, отсутствующую в исходной формуле.

2) Если квантору существования предшествует один или несколько кванторов всеобщности, то квантор существования вычёркивается, а предметная переменная, связанная этим квантором, заменяется на функцию (отсутствующую в исходной формуле), зависящую от предметных переменных, связанных кванторами всеобщности, которые предшествуют квантору существования.

Следующую теорему приведём без доказательств.

**Теорема 4.3.2.** *Всякая формула логики предикатов может быть приведена к сколемовской нормальной форме.*

**Замечание 4.3.1.** *Сколемовская нормальная форма, вообще говоря, не равносильна исходной формуле.*

**Пример 4.3.3.** Привести к СНФ формулу

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \left(P(x_1,x_2) \to Q(x_3,x_4,x_5)\right).$$

*Решение.*

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \left(P(x_1,x_2) \to Q(x_3,x_4,x_5)\right) \equiv$$

$$\equiv \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \left(\overline{P(x_1,x_2)} \vee Q(x_3,x_4,x_5)\right) \equiv$$

$$\equiv \forall x_2 \exists x_3 \forall x_4 \exists x_5 \left(\overline{P(a,x_2)} \vee Q(x_3,x_4,x_5)\right) \equiv$$

$$\equiv \forall x_2 \forall x_4 \exists x_5 \left(\overline{P(a,x_2)} \vee Q(f(x_2),x_4,x_5)\right) \equiv$$

$$\equiv \forall x_2 \forall x_4 \left(\overline{P(a,x_2)} \vee Q\left(f(x_2),x_4,g(x_2,x_4)\right)\right).$$

## 4.4 Распознавание общезначимости формул

Проблема разрешимости в логике предикатов ставится так же, как и в алгебре логики: существует ли алгоритм, позволяющий для любой форму-

лы логики предикатов установить, является ли она общезначимой, выполнимой, или тождественно ложной?

В отличие от алгебры логики, в логике предикатов не применим метод перебора всех вариантов значений переменных, входящих в формулу, так как таких вариантов может быть бесконечное множество. Приведём без доказательства следующую теорему.

**Теорема 4.4.1.** *Проблема разрешимости логики предикатов в общем виде алгоритмически не разрешима.*

Очевидно, что проблема разрешимости в случае конечной области определения предиката разрешима. В этом случае, как было указано выше, кванторные операции могут быть заменены операциями конъюнкции и дизъюнкции, и таким образом формула логики предикатов сводится к формуле алгебры логики, для которой, в свою очередь, проблема разрешимости разрешима.

Например, пусть дана формула $\forall x \exists y \big(P(x,y) \vee \overline{P(x,x)}\big)$, определённая на области $M = M_1 \times M_2$, где $M_1 = M_2 = \{a, b\}$. Тогда она может быть приведена к виду:

$$\forall x \exists y \big(P(x,y) \vee \overline{P(x,x)}\big) \equiv \forall x \big(P(x,a) \vee \overline{P(x,x)} \vee P(x,b) \vee \overline{P(x,x)}\big) \equiv$$

$$\equiv \forall x \big(P(x,a) \vee P(x,b) \vee \overline{P(x,x)}\big) \equiv$$

$$\equiv \big(P(a,a) \vee P(a,b) \vee \overline{P(a,a)}\big) \& \big(P(b,a) \vee P(b,b) \vee \overline{P(b,b)}\big).$$

Представим некоторые результаты для формул, содержащих в предварённой нормальной форме только кванторы всеобщности или только кванторы существования.

**Определение 4.4.1.** *Формула логики предикатов называется* ***замкнутой****, если она не содержит свободных переменных.*

**Определение 4.4.2.** *Если формула логики предикатов $P$ содержит свободные переменные $x_1, x_2, \dots, x_n$, то формула $\forall x_1 \dots \forall x_n P(x_1, x_2, \dots, x_n)$ называется* ***замыканием всеобщности*** *формулы $P$, а формула $\exists x_1 \dots \exists x_n P(x_1, x_2, \dots, x_n)$ называется* ***замыканием существования*** *формулы $P$.*

**Теорема 4.4.2.** *Если замкнутая формула логики предикатов в предварённой нормальной форме содержит только кванторы существования и тождественно истинна на любой области, состоящей из одного элемента, то она общезначима.*

*Доказательство.* Пусть формула логики предикатов в предварённой нормальной форме имеет вид:

$$B = \exists x_1 \exists x_2 \dots \exists x_n C(A_1, A_2, \dots, P_1, P_2, \dots, Q_1, Q_2, \dots), \qquad (4.1)$$

где формула $C$ не содержит кванторов, $A_i$ – высказывательные переменные, $P_i$ – одноместные предикаты, $Q_i$ – двухместные предикаты и т. д.

По условию теоремы на любой области $M = \{a\}$, содержащей только один элемент $a$, данная формула тождественно истинна, то есть

$$C(A_1, A_2, \dots, P_1(a), P_2(a), \dots, Q_1(a,a), Q_2(a,a), \dots) = 1. \qquad (4.2)$$

Предположим, что формула (4.1) не является общезначимой. Тогда существует такая область $M'$ и такой набор значений переменных $A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots$, на котором формула (4.1) принимает значение ложь, то есть

$$\exists x_1 \exists x_2 \dots \exists x_n C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots) = 0. \qquad (4.3)$$

Рассмотрим отрицание формулы (4.3).

$$\overline{\exists x_1 \exists x_2 \dots \exists x_n C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)} \equiv$$

$$\equiv \forall x_1 \forall x_2 \dots \forall x_n \overline{C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)} = 1.$$

Отсюда следует, что формула

$$\overline{C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)} \qquad (4.4)$$

тождественно истинна, независимо от выбора предметных переменных из области $M'$. Возьмём из области $M'$ какой-нибудь элемент $b$ и подставим его в формулу (4.4) вместо всех предметных переменных. Тогда

$$\overline{C(A_1^0, A_2^0, \dots, P_1^0(b), P_2^0(b), \dots, Q_1^0(b,b), Q_2^0(b,b), \dots)} = 1 \Longrightarrow$$

$$\Longrightarrow C(A_1^0, A_2^0, \dots, P_1^0(b), P_2^0(b), \dots, Q_1^0(b,b), Q_2^0(b,b), \dots) = 0. \qquad (4.5)$$

Заметим, что формула (4.5) противоречит формуле (4.2). Теорема доказана. □

**Теорема 4.4.3.** *Если замкнутая формула логики предикатов в предварённой нормальной форме содержит только кванторы всеобщности, число которых ровно $n$, и тождественно истинна на любой области, содержащей не более чем $n$ элементов, то она общезначима.*

*Доказательство.* Пусть формула логики предикатов в предварённой нормальной форме имеет вид:

$$B = \forall x_1 \forall x_2 \dots \forall x_n C(A_1, A_2, \dots, P_1, P_2, \dots, Q_1, Q_2, \dots), \qquad (4.6)$$

где формула $C$ кванторов не содержит, $A_i$ – высказывательные переменные, $P_i$ – одноместные предикаты, $Q_i$ – двухместные предикаты и т. д.

Предположим, что формула (4.6) не является общезначимой. Тогда существует область $M$ с числом элементов, большим чем $n$, и такой набор значений переменных $A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots$, на котором формула (4.6) принимает значение ложь, то есть

$$\forall x_1 \forall x_2 \dots \forall x_n C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots) = 0. \qquad (4.7)$$

Рассмотрим отрицание формулы (4.7).

$$\overline{\forall x_1 \forall x_2 \dots \forall x_n C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)} \equiv$$

$$\equiv \exists x_1 \exists x_2 \dots \exists x_n \overline{C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)} = 1.$$

Из чего можно сделать вывод, что существует набор предметных переменных $x_1, x_2, \dots, x_n$ из области $M$, при котором формула $\overline{C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)}$ принимает значение 1, а формула $C(A_1^0, A_2^0, \dots, P_1^0, P_2^0, \dots, Q_1^0, Q_2^0, \dots)$ принимает значение 0. Значит, из области $M$ можно выделить область $M'$, содержащую не более $n$ элементов, на которой данная формула не является тождественно истинной, а это, в свою очередь, противоречит условию теоремы. □

## 4.5 Применение языка логики предикатов для записи математических предложений

Язык логики предикатов удобен для записи математических предложений. Он даёт возможность выражать логические связи между понятиями, записывать определения, теоремы, доказательства.

Приведём пару примеров таких записей.

1) Определение строго возрастающей функции. Функция $f(x)$, определённая на множестве $M$, строго возрастает на этом множестве, если

$$\forall x \in M \, \forall y \in M \, \big(x < y \Longrightarrow f(x) < f(y)\big).$$

Здесь $P(x, y) = (x < y \Longrightarrow f(x) < f(y))$ – двухместный предикат.

2) Определение предела числовой последовательности.

$$\lim_{n \to \infty} a_n = a \Leftrightarrow \forall \varepsilon > 0 \; \exists n_0 \in \mathbb{N} \; \forall n \in \mathbb{N} \; (n \geq n_0 \Longrightarrow |a_n - a| < \varepsilon).$$

Здесь $Q(\varepsilon, n, n_0) = (n \geq n_0 \Longrightarrow |a_n - a| < \varepsilon)$ – трёхместный предикат, а $\mathbb{N}$ – множество натуральных чисел.

Математические теоремы допускают формулировку в виде условных предложений. Говоря о строении теорем, можно выделить в них три части: 1) условие теоремы; 2) заключение теоремы; 3) разъяснительная часть, описывающая множество объектов, о которых идёт речь в теореме.

Например, в теореме, записанной в виде $\exists x \in M\,\big(P(x) \longrightarrow Q(x)\big)$, $P(x)$ – условие, $Q(x)$ –заключение, а $\exists x \in M$ – разъяснительная часть.

***Построение противоположных утверждений и доказательство методом от противного***

Пусть дано некоторое математическое утверждение $A$. Ему противоположным будет утверждение $\bar{A}$, которое можно построить с помощью равносильных преобразований.

Например, определение ограниченной сверху на множестве $M$ функции $f(x)$ задаётся формулой

$$\exists K > 0\ \forall x \in M\ (f(x) \le K).$$

Определение неограниченной функции может быть получено, беря отрицание этой формулы и проводя равносильные преобразования:

$$\overline{\exists K > 0\ \forall x \in M\ (f(x) \le K)} \equiv \forall K > 0\ \overline{\forall x \in M\ (f(x) \le K)} \equiv$$

$$\equiv \forall K > 0\ \exists x \in M\ \overline{(f(x) \le K)} \equiv \forall K > 0\ \exists x \in M\ (f(x) > K).$$

Последняя формула даёт не негативное, а положительное определение неограниченной функции.

Особый интерес представляет построение утверждения, отрицающего справедливость некоторой теоремы: $\forall x \in M\,\big(P(x) \longrightarrow Q(x)\big)$ $(*)$. Получим следующее утверждение:

$$\overline{\forall x \in M\,\big(P(x) \longrightarrow Q(x)\big)} \equiv \exists x \in M\,\overline{\Big(\overline{P(x)} \vee Q(x)\Big)} \equiv \exists x \in M\,\big(P(x)\&\overline{Q(x)}\big).$$

Следовательно, чтобы доказать, что теорема $\forall x \in M\big(P(x) \longrightarrow Q(x)\big)$ неверна, достаточно указать такой элемент $x \in M$, для которого $P(x)$ истинна, а $Q(x)$ ложна, то есть привести *контрпример*.

В свою очередь, для доказательства теоремы $(*)$ методом от противного предполагается, что теорема $(*)$ неверна, то есть существует $x \in M$, для которого $P(x)$ истинна, а $Q(x)$ ложна. Если из этих предположений путём логических рассуждений получится прийти к противоречивому утверждению, то делается вывод о том, что исходное предположение неверно, и верна теорема $(*)$.

***Формулировка обратных и противоположных теорем***

Рассмотрим четыре теоремы:

1) $\forall x \in M\,\big(P(x) \longrightarrow Q(x)\big)$;
2) $\forall x \in M\,\big(Q(x) \longrightarrow P(x)\big)$;
3) $\forall x \in M\,\big(\overline{P(x)} \longrightarrow \overline{Q(x)}\big)$;
4) $\forall x \in M\,\big(\overline{Q(x)} \longrightarrow \overline{P(x)}\big)$.

Пара теорем, у которых условие одной является заключением второй, а условие второй является заключением первой, называются *взаимно обратными* друг другу. Так, теоремы 1) и 2), а также 3) и 4) – взаимно обратны. Одна из этих теорем обычно называется *прямой*, а вторая называется *обратной*. Пара теорем, у которых условие и заключение одной является отрицанием соответственно условия и заключения другой, называются *взаимно противоположными*. Так, теоремы 1) и 3), а также 2) и 4) являются взаимно противоположными теоремами.

Например, для теоремы "Если последняя цифра числа $k$ – ноль или делится на 2, то число $k$ чётно" $(**)$ обратной теоремой будет "Если число $k$ является чётным числом, то последняя цифра числа $k$ – ноль или делится на 2". Для теоремы $(**)$ противоположной будет теорема "Если последняя цифра числа $k$ не является нулём и не делится на 2, то число $k$ нечётно".

Прямая и обратная теоремы, вообще говоря, не равносильны, то есть одна из них может быть истинной, а другая ложной. Однако теоремы 1) и 4), а также 2) и 3) всегда равносильны. Это легко доказывается:

$$\forall x \in M \left(P(x) \longrightarrow Q(x)\right) \equiv \forall x \in M \left(\overline{P(x)} \vee Q(x)\right) \equiv$$

$$\equiv \forall x \in M \left(Q(x) \vee \overline{P(x)}\right) \equiv \forall x \in M \left(\overline{Q(x)} \longrightarrow \overline{P(x)}\right).$$

Аналогично доказывается и равносильность

$$\forall x \in M \left(Q(x) \longrightarrow P(x)\right) \equiv \forall x \in M \left(\overline{P(x)} \longrightarrow \overline{Q(x)}\right).$$

***Формулировка необходимых и достаточных условий***

Рассмотрим теорему 1), описанную выше. В данном случае предикат $Q(x)$ логически следует из предиката $P(x)$, поэтому $P(x)$ называется *достаточным условием* для $Q(x)$, а $Q(x)$ называется *необходимым условием* для $P(x)$. Если истинны взаимно обратные теоремы 1) и 2), то истинна и теорема

$$\forall x \in M \left(\left(P(x) \longrightarrow Q(x)\right)\&\left(Q(x) \longrightarrow P(x)\right)\right) \equiv$$

$$\equiv \forall x \in M\left(P(x) \longrightarrow Q(x)\right)\& \forall x \in M\left(Q(x) \longrightarrow P(x)\right).$$

В таком случае говорят, что $P(x)$ является необходимым и достаточным условием для $Q(x)$, а $Q(x)$ – необходимым и достаточным условием для $P(x)$.

В математике подобные теоремы формулируются в следующем виде: "необходимо и достаточно, чтобы" или "тогда и только тогда, когда".

Приведём пример такой теоремы: "Окружность можно описать около параллелограмма тогда и только тогда, когда параллелограмм является прямоугольником".

## 4.6 Исчисление предикатов

Исчисление предикатов является расширением исчисления высказываний. Языком исчисления предикатов является язык логики предикатов.

I. Понятие формулы исчисления предикатов совпадает с понятием формулы логики предикатов.

II. Система аксиом исчисления предикатов состоит из четырёх групп аксиом, используемых в исчислении высказываний, и ещё одной пятой группы (V), состоящей из двух дополнительных аксиом:

1. $\forall x F(x) \longrightarrow F(y)$;
2. $F(y) \longrightarrow \exists x F(x)$.

III. К правилам вывода, которые используются в исчислении высказываний (правило подстановки и $MP$), добавляются еще два правила:

1. Правило обобщения ($\forall$ −правило). Если формула $F \to G(x)$ выводима в исчислении предикатов и $F$ не зависит от $x$, то и формула $F \to \forall x G(x)$ также выводима. Записывается так:

$$\frac{F \to G(x)}{F \to \forall x G(x)}.$$

2. Правило конкретизации ($\exists$ −правило). Если формула $G(x) \to F$ выводима в исчислении предикатов и $F$ не зависит от $x$, то и формула $\exists x G(x) \to F$ также выводима. Записывается так:

$$\frac{G(x) \to F}{\exists x G(x) \to F}.$$

Дадим некоторые уточнения о применении правила подстановки в исчислении предикатов.

*Замена высказывательной переменной.* Пусть формула $A$ содержит высказывательную переменную $B$. Тогда $B$ можно заменить любой формулой $G$ при условии, что если $B$ находится в области действия квантора, который связывает какую-то переменную, то эта переменная не входит в $G$.

Например, пусть $A = \forall x \forall y\left(\bar{C} \vee \forall z W(z,x) \vee F(x,y)\right)$. Здесь $C$ нельзя заменять на $\forall x G(x)$, так как не будет соблюдаться условие коллизии переменных. Возможна следующая замена:

$$\int_{C}^{\exists z(C\&\forall t W(z,t)\&B)} (A) \vdash \forall x \forall y \left(\overline{\exists z(C\&\forall t W(z,t)\&B)} \vee \forall z W(z,x) \vee F(x,y)\right).$$

*Замена предиката.* Заменяемый предикат $P$ в формуле $A$ должен заменятся другим предикатом $Q$, который содержит как минимум все переменные, входящие в $P$. Также переменные в $Q$ могут быть связаны кванторами такими, что эти кванторы не создают коллизии с кванторами, в область действия которых попадает $P$ в $A$.

Например, если $A = \forall x \forall y\big(\exists z P(x,z) \vee \overline{P(y,x)}\big)$, то возможна следующая замена:

$$\int_{P(x,z)}^{\forall u H(x,z,u)} (A) \vdash \forall x \forall y\big(\exists z \forall u H(x,z,u) \vee \overline{\forall u H(y,x,u)}\big).$$

*Замена предметной переменной.* Заменяемая связанная предметная переменная должна заменятся всюду в области действия квантора, связывающего данную переменную и в самом кванторе. Заменяемая свободная переменная должна заменятся всюду только на свободную переменную.

Например, если $A = \forall x F(x) \longrightarrow \exists y\big(F(y) \vee P(y,t)\big)$, то возможна следующая замена:

$$\int_{y,t,x}^{z,e,y} (A) \vdash \forall y F(y) \longrightarrow \exists z\big(F(z) \vee P(z,e)\big).$$

Также определены дополнительные правила вывода исчисления предикатов (для кванторов):

1. $\dfrac{F(x)}{\forall x F(x)}$ – правило введения квантора всеобщности;
2. $\dfrac{\forall x F(x)}{F(y)}$ – правило удаления квантора всеобщности;
3. $\dfrac{F(y)}{\exists x F(x)}$ – правило введения квантора существования;
4. $\dfrac{\exists x F(x)}{F(y)}$ – правило удаления квантора существования.

И даны производные аксиомы:

1. $\forall x F(x) \longrightarrow \exists x F(x)$;
2. $\forall x \forall y F(x,y) \leftrightarrow \forall y \forall x F(x,y)$;
3. $\exists x \forall y F(x,y) \longrightarrow \forall y \exists x F(x,y)$;
4. $\forall x\big(F(x) \longrightarrow G(x)\big) \longrightarrow \big(\forall x F(x) \longrightarrow \forall x G(x)\big)$;
5. $\forall x\big(F(x) \longrightarrow G(x)\big) \longrightarrow \big(\exists x F(x) \longrightarrow \exists x G(x)\big)$;
6. $\forall x\big(F(x) \leftrightarrow G(x)\big) \longrightarrow \big(\exists x F(x) \leftrightarrow \exists x G(x)\big)$.

IV. Понятия вывода и выводимой формулы определяются аналогично этим понятиям в исчислении высказываний.

V. Как и во всякой аксиоматической теории, рассматриваются следующие проблемы: разрешимости, непротиворечивости, полноты, независимости.

**Теорема 4.6.1.** *Всякая выводимая формула исчисления предикатов общезначима.*

Эту теорему оставим без доказательства.

Напомним, что противоречивым называется такое исчисление, в котором какая-либо формула выводима вместе со своим отрицанием. Исчисление называется полным в узком смысле, если добавление к его аксиомам любой не выводимой формулы приводит к противоречивому исчислению. Исчисление называется полным в широком смысле, если любая тождественно истинная формула в ней выводима.

Известно, что исчисление предикатов непротиворечиво, неполно в узком смысле, полно в широком смысле и неразрешимо.

Также верна и следующая теорема.

**Теорема 4.6.2.** *Система аксиом исчисления предикатов независима.*

**Пример 4.6.1.** Показать, что $\forall x F(x) \vdash \forall y F(y)$.

*Решение*. Построим вывод:

1. $\forall x F(x)$ (гипотеза).
2. $\forall x F(x) \longrightarrow F(y)$ (аксиома V. 1).
3. $\forall -$правило$(2) \vdash \forall x F(x) \longrightarrow \forall y F(y)$.
4. $MP(1,3) \vdash \forall y F(y)$.

**Пример 4.6.2.** Доказать выводимость формулы $\exists y \forall x F(x) \longrightarrow \forall y \exists x F(x)$.

*Решение*. Построим цепочку вывода:

1. $\forall x F(x) \longrightarrow F(y)$ (аксиома V. 1).
2. $F(y) \longrightarrow \exists x F(x)$ (аксиома V. 2).
3. $\text{ПС}(1,2) \vdash \forall x F(x) \longrightarrow \exists x F(x)$.
4. $\exists -$правило$(3) \vdash \exists y \forall x F(x) \longrightarrow \exists x F(x)$.
5. $\forall -$правило$(4) \vdash \exists y \forall x F(x) \longrightarrow \forall y \exists x F(x)$.

**Пример 4.6.3.** Доказать выводимость формулы $F(x) \longrightarrow \big(F(x) \vee \forall y G(y)\big)$.

*Решение*. Построим вывод:

1. $A \longrightarrow (A \vee B)$ (аксиома III. 1).
2. $\int_{A,B}^{F(x),\forall y G(y)}(1) \vdash F(x) \longrightarrow \big(F(x) \vee \forall y G(y)\big)$.

**Пример 4.6.4.** Используя аксиому $(A \longrightarrow B) \longrightarrow \big((B \longrightarrow C) \longrightarrow (A \longrightarrow C)\big)$, доказать: $\forall x\big(F(x) \longrightarrow G(x)\big) \vdash \forall x\big(F(x) \longrightarrow G(x)\big) \rightarrow \big(\forall xF(x) \longrightarrow \forall xG(x)\big)$.

*Решение*. Построим вывод:

1. $\forall xF(x) \longrightarrow F(y)$ (аксиома V. 1).
2. $\int_{F(x)}^{F(x)\longrightarrow G(x)}(1) \vdash \forall x\big(F(x) \longrightarrow G(x)\big) \longrightarrow \big(F(y) \longrightarrow G(y)\big)$.
3. $\forall x\big(F(x) \longrightarrow G(x)\big)$ (гипотеза).
4. $MP(3,2) \vdash\ F(y) \longrightarrow G(y)$.
5. $(A \longrightarrow B) \longrightarrow \big((B \longrightarrow C) \longrightarrow (A \longrightarrow C)\big)$ (аксиома).
6. $\int_{A,B,C}^{\forall xF(x),F(y),G(y)}(5) \vdash \big(\forall xF(x) \longrightarrow F(y)\big) \longrightarrow \Big(\big(F(y) \longrightarrow G(y)\big) \longrightarrow \big(\forall xF(x) \longrightarrow G(y)\big)\Big)$.
7. $MP(1,6) \vdash\ \big(F(y) \longrightarrow G(y)\big) \longrightarrow \big(\forall xF(x) \longrightarrow G(y)\big)$.
8. $MP(4,7) \vdash \forall xF(x) \longrightarrow G(y)$.
9. $\forall$ −правило$(8) \vdash \forall xF(x) \longrightarrow \forall yG(y)$.
10. $\int_{y}^{x}(9) \vdash \forall xF(x) \longrightarrow \forall xG(x)$.
11. $\forall x\big(F(x) \longrightarrow G(x)\big) \vdash \forall xF(x) \longrightarrow \forall xG(x)$.
12. По теореме дедукции $\vdash \forall x\big(F(x) \longrightarrow G(x)\big) \rightarrow \big(\forall xF(x) \longrightarrow \forall xG(x)\big)$.

## 4.7 Задачи для самостоятельного решения

1. Найти отрицание формул и упростить их так, чтобы знаки отрицания относились только к предикатным переменным:
   a) $\forall x\big(P(x)\&Q(x)\big)$;
   b) $\exists x\Big(\overline{P(x) \vee Q(x)}\&\forall yL(x,y)\Big)$;
   c) $F(x) \longrightarrow \forall x\overline{\big(\exists yP(x,y) \vee \overline{F(x)}\big)\&\forall zF(z)}$;
   d) $\forall x\exists y\forall z\big(P(x,y) \longrightarrow \exists u\overline{Q(x,y,z,u)}\big)\&\exists y\overline{\forall xP(x,y)}$;
   e) $\forall x\exists y\Big(\big(R(x,y) \vee \overline{Q(x,y,z)}\big) \longrightarrow L(x,y)\Big)$.
2. Определить тождественную истинность, ложность либо выполнимость в области натуральных чисел $\mathbb{N}$ следующих формул:
   a) $\big(P(x,y,z)\&P(y,x,u)\big) \longrightarrow Q(x,y)$;

b) $\forall x \forall y \forall z \forall u \left( \left( P(x,y,z) \& P(x,y,u) \right) \longrightarrow Q(x,y) \right)$;

c) $\exists y \left( F(x,x,y) \longrightarrow R(x,x,y) \right)$;

d) $Q(x,y) \longrightarrow \left( \exists y R(x,y,z) \vee P(x,z) \right)$;

e) $\forall x \forall y \forall z \left( \left( P(x,y,z) \& \overline{Q(x,y)} \right) \longrightarrow R(x,y,z) \right)$.

3. Доказать методом равносильных преобразований общезначимость (тождественную истинность) формул:

a) $\exists x \left( P(x) \vee Q(x) \right) \to \left( \exists x P(x) \vee \exists x Q(x) \right)$;

b) $\exists x \left( P(x) \to Q(x) \right) \to \left( \forall x P(x) \to \exists x Q(x) \right)$;

c) $\forall x \left( P(x) \to \overline{Q(x)} \right) \to \overline{\forall x P(x) \& \exists x Q(x)}$;

d) $\forall x P(x) \vee \forall x Q(x) \to \overline{\overline{\exists x P(x)} \& \overline{\forall x P(x)}}$;

e) $\forall x P(x) \& \forall x Q(x) \to \exists x \left( P(x) \vee Q(x) \right)$.

4. Доказать методом от противного общезначимость формул:

a) $\forall x \left( P(x) \to Q(x) \right) \vee \left( Q(x) \to P(x) \right)$;

b) $\forall x \left( P(x) \to \left( Q(x) \to \left( P(x) \& Q(x) \right) \right) \right)$;

c) $\forall x \left( \overline{P(x)} \to \left( P(x) \to Q(x) \right) \right)$;

d) $\forall x \left( \left( Q(x) \to R(x) \right) \to \left( \left( P(x) \vee Q(x) \right) \to \left( P(x) \to R(x) \right) \right) \right)$;

e) $\forall x \left( \left( P(x) \to Q(x) \right) \to \left( \left( P(x) \to \overline{Q(x)} \right) \to \overline{P(x)} \right) \right)$.

5. Привести к ПНФ и СНФ формулы:

a) $\overline{S \to \forall x P(x)}$;

b) $\overline{\exists x \forall y \exists z \forall u P(x,y,z,u)}$;

c) $\exists x \forall y P(x,y) \vee \overline{\forall y \exists x Q(x,y)}$;

d) $\forall x P(x,y) \vee \left( \exists x P(x,x) \to \forall z \left( \overline{Q(y,z)} \to \exists x P(x,z) \right) \right)$;

e) $\forall x \left( A(x) \to \forall y \left( B(x,y) \to \overline{\forall z C(y,z)} \right) \right)$;

f) $\forall x \left( \overline{P(x)} \to \exists y \overline{Q(y)} \right) \to \left( Q(z) \to P(z) \right)$;

g) $\overline{\forall x \forall y \forall z \overline{P(x,y,z)} \to \exists y \exists z Q(y,z)} \& \forall x \forall z R(x,z)$;

h) $\exists x \forall y \left( P(x,y) \to \exists z \left( Q(x,z) \& R(y) \right) \right)$.

6. Показать выводимость формул в исчисление предикатов:
   a) $\forall x \forall y \Big(P(x) \longrightarrow \big(Q(y) \longrightarrow P(x)\big)\Big)$;
   b) $P(x) \longrightarrow P(x) \vee Q(x)$;
   c) $\forall x\big(A \longrightarrow F(x)\big) \longrightarrow \big(A \longrightarrow \forall x F(x)\big)$;
   d) $\exists x F(x) \longrightarrow \overline{\forall y \overline{F(y)}}$;
   e) $\exists x\big(Q(y) \& P(x)\big) \longrightarrow \exists x P(x)$;
   f) $\forall x \exists y \forall z \Big(\big(P(x) \& \overline{P(y)}\big) \longrightarrow Q(z)\Big)$.

7. Доказать, что имеют место следующие выводимости, построив соответствующие выводы из гипотез:
   a) $P(x) \longrightarrow \forall y Q(y) \vdash \forall x\big(P(x) \longrightarrow \forall y Q(y)\big)$;
   b) $\exists x\big(G(y) \longrightarrow F(x)\big) \vdash G(y) \longrightarrow \exists x F(x)$;
   c) $\exists x F(x) \longrightarrow G(y) \vdash \forall x\big(F(x) \longrightarrow G(y)\big)$;
   d) $\forall x\big(P(x) \longrightarrow Q(x)\big) \vdash \exists x P(x) \longrightarrow \exists x Q(x)$;
   e) $\exists x P(x) \longrightarrow \forall x Q(x) \vdash \forall x\big(P(x) \longrightarrow Q(x)\big)$;
   f) $P(x), P(x) \longrightarrow \forall y Q(y) \vdash \forall x Q(x)$;
   g) $\forall x\big(F(x) \longrightarrow G(x)\big) \vdash \forall x\big(F(x) \longrightarrow G(x)\big) \longrightarrow \big(\exists x F(x) \longrightarrow \exists x G(x)\big)$.

8. Доказать выводимость формул в исчисление предикатов:
   a) $\exists x\big(P(x) \longrightarrow Q(y)\big) \leftrightarrow \big(\forall x P(x) \longrightarrow Q(y)\big)$;
   b) $\forall x \forall y F(x, y) \leftrightarrow \forall y \forall x F(x, y)$;
   c) $\exists x F(x) \leftrightarrow \overline{\forall x \overline{F(x)}}$;
   d) $\Big(\exists z R(z) \vee \big(\exists z R(z) \& \forall x \forall y \overline{Q(x, y)}\big)\Big) \leftrightarrow \exists z R(z)$.

# Глава 5
# Теория алгоритмов

Теория алгоритмов – раздел математики, изучающий общие свойства алгоритмов. Эта теория тесно связана с математической логикой, поскольку на понятие алгоритма опирается одно из центральных понятий математической логики – понятие исчисления, о котором говорилось в предыдущих параграфах. В сущности, вся математика в той или иной мере связана с алгоритмами.

Точное описание последовательных действий, исполнение которых приводит к решению поставленных задач, называется *алгоритмом*. Примеры алгоритмов: выполнение арифметических действий над числами, решение квадратного уравнения, нахождение производной и т. п.

Алгоритм на основе входных данных выдаёт выходные данные и обладает рядом свойств:

1. *Дискретность*. Алгоритм должен осуществляться путём выполнения последовательности элементарных действий.
2. *Детерминированность* (*определённость*). В каждый момент времени алгоритм должен точно знать, какое действие выполнить следующим.
3. *Результативность.* Алгоритм должен выдавать результат за конечное число шагов.
4. *Массовость.* Алгоритм должен быть пригоден для решения всех задач данного типа.
5. *Правильность.* Результат работы алгоритма должен соответствовать условию задачи.

Существуют различные формализации понятия алгоритма. Мы более детально остановимся на следующих трёх: частично рекурсивные функции, машины Тьюринга, нормальные алгоритмы Маркова.

## 5.1 Частично рекурсивные функции

Этот подход к формализации алгоритмов принадлежит Курту Гёделю и Стивену Клину (1936). Основная идея состоит в том, чтобы свести все известные алгоритмы к вопросу вычисления значений подходящей функции.

В дальнейшем под обозначением $\mathbb{N}$ будем понимать множество натуральных чисел, включающее ноль.

Пусть $f(x_1, x_2, \dots, x_n)$ – функция от $n$ переменных. Обозначим через $D(f)$ – область определения функции $f$, а через $E(f)$ – область значения функции $f$.

**Определение 5.1.1.** *Функция $f(x_1, x_2, \dots, x_n)$ называется **частично числовой функцией**, если:*

1) $D(f) \subseteq \mathbb{N}^n = \underbrace{\mathbb{N} \times \mathbb{N} \times \dots \times \mathbb{N}}_{n}$;
2) $E(f) \subseteq \mathbb{N}$.

Значения частично числовой функции $f$ на наборах $\mathbb{N}^n \backslash D(f)$ считаются неопределёнными. Множество всех частичных числовых функций обозначим через $P_{\mathbb{N}}$.

**Определение 5.1.2.** *Функция $f(x_1, x_2, \dots, x_n)$ называется **всюду определённой числовой функцией**, если:*

1) $D(f) = \mathbb{N}^n$;
2) $E(f) \subseteq \mathbb{N}$.

**Определение 5.1.3.** ***Рекурсивной** называется числовая функция, которая определяется через себя.*

Задание рекурсивной функции $f$ от аргумента $n \in \mathbb{N}$ аналогично доказательству методом математической индукции:

1) сначала функция задаётся для некоторых начальных значений $n_0$ (например, для $n_0 = 0$ или 1);
2) значение функции $f(n+1)$ выражается через значение $f(n)$, $n \geq n_0$.

Приведём пример рекурсивной функции. Факториалом числа $n$ называется функция, определённая на множестве $\mathbb{N}$, равная произведению натуральных чисел от 1 до $n$ включительно:

$$f_!(n) = 1 * 2 * 3 * \dots * (n-1) * n = n!$$

Факториал нуля полагают равным единице.

Так как $f_!(n-1) = 1 * 2 * 3 * \ldots * (n-2) * (n-1) = (n-1)!$, то $n! = (n-1)! * n$. Следовательно, при вычислении этой функции можно пользоваться рекурсивной формой её записи:

$$f_!(n) = \begin{cases} f_!(0) = 1, \\ f_!(n) = f_!(n-1) * n, \ \ \forall n \geq 1. \end{cases}$$

**Определение 5.1.4.** *Функция называется* ***вычислимой****, если существует алгоритм, позволяющий вычислить её значения.*

Введём простейшие числовые функции:

1) $\lambda(x) = x + 1$ – оператор сдвига;
2) $O(x) = 0$ – оператор аннулирования;
3) $I_n^m(x_1, x_2, \ldots, x_n) = x_m, \ 1 \leq m \leq n, \ n = 1,2,3, \ldots$ – оператор проецирования.

Ясно, что все три простейшие функции всюду определены и вычислимы. Далее определим на множестве $P_{\mathbb{N}}$ операции суперпозиции, примитивной рекурсии и минимизации.

***Суперпозиция функций***

Пусть даны некоторые частично числовые функции $g(x_1, x_2, \ldots, x_m)$, $f_1(x_1, x_2, \ldots, x_n)$, … , $f_m(x_1, x_2, \ldots, x_n)$, а $D(g)$, $D(f_1)$, … , $D(f_m)$ – области определения функций $g$, $f_1$, … , $f_m$ соответственно. Будем говорить, что функция

$$h(x_1, x_2, \ldots, x_n) = g\big(f_1(x_1, x_2, \ldots, x_n), \ldots, f_m(x_1, x_2, \ldots, x_n)\big)$$

получается из функций $g$, $f_1$, … , $f_m$ при помощи операции *суперпозиции*. Значение функции $h$ на наборе $\tilde{\alpha} \in \mathbb{N}^n$ определяются следующим образом:

1) если $\tilde{\alpha} \in D(f_i)$ для всех $i = \overline{1,m}$ и набор $\big(f_1(\tilde{\alpha}), f_2(\tilde{\alpha}), \ldots, f_m(\tilde{\alpha})\big)$ принадлежит множеству $D(g)$, то $h(\tilde{\alpha}) = g\big(f_1(\tilde{\alpha}), \ldots, f_m(\tilde{\alpha})\big)$;
2) в противном случае значение $h(\tilde{\alpha})$ считается неопределённым.

Очевидно, что если функции $g, f_1, \ldots, f_m$ всюду определены, то и функция $h$ тоже всюду определена.

***Примитивная рекурсия***

Пусть $g(x_1, x_2, \ldots, x_{n-1})$, $h(x_1, x_2, \ldots, x_{n+1})$ – некоторые частично числовые функции и $n \geq 2$. Рассмотрим новую функцию $f(x_1, x_2, \ldots, x_n)$, которая удовлетворяет следующим равенствам:

$$\begin{cases} f(x_1, x_2, \ldots, x_{n-1}, 0) = g(x_1, x_2, \ldots, x_{n-1}), \\ f(x_1, x_2, \ldots, x_{n-1}, x_n + 1) = h\big(x_1, x_2, \ldots, x_{n-1}, x_n, f(x_1, x_2, \ldots, x_n)\big). \end{cases} \quad (5.1)$$

Отметим, что функция $g$ зависит от $n-1$ аргументов, функция $h$ – от $n+1$ аргументов, а функция $f$ от $n$ аргументов.

Равенства (5.1) называются *схемой примитивной рекурсии* (по переменной $x_n$). Будем говорить, что функция $f(x_1, x_2, \dots, x_n)$ получена из функций $g$ и $h$ с помощью операции *примитивной рекурсии* (по переменной $x_n$).

В случае $n = 1$, схема примитивной рекурсии имеет следующий вид:

$$\begin{cases} f(0) = a, \\ f(x+1) = h\big(x, f(x)\big), \end{cases}$$

где $a$ – фиксированное число из множества $\mathbb{N}$.

Нетрудно показать, что для любых частичных числовых функций $g(x_1, x_2, \dots, x_{n-1})$, $h(x_1, x_2, \dots, x_{n+1})$ существует в точности одна частичная числовая функция $f(x_1, x_2, \dots, x_n)$, получающаяся из функций $g$ и $h$ с помощью операции примитивной рекурсии (по переменной $x_n$). При этом область определения функции $f$ удовлетворяет следующим условиям:

1) набор $(\alpha_1, \alpha_2, \dots, \alpha_{n-1}, 0)$ принадлежит $D(f)$ тогда и только тогда, когда $(\alpha_1, \alpha_2, \dots, \alpha_{n-1}) \in D(g)$;
2) набор $(\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n + 1)$ принадлежит $D(f)$ тогда и только тогда, когда набор $(\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n)$ принадлежит $D(f)$ и одновременно $\big(\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n, f(\alpha_1, \alpha_2, \dots, \alpha_n)\big) \in D(h)$.

В частности, если для некоторых $\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n$ значение $f(\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n)$ не определено, то и для всех $\beta \geq \alpha_n$ значения $f(\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \beta)$ также будут не определены.

Операцию примитивной рекурсии можно применять по любой переменной.

**Определение 5.1.5.** *Функция $f(x_1, x_2, \dots, x_n)$ называется **примитивно рекурсивной**, если она может быть получена из простейших функций $\lambda$, $O$ и $I_n^m$ с помощью конечного числа применений операций суперпозиции и примитивной рекурсии.*

Очевидно, что если функции $g$ и $h$ всюду определены, то будет всюду определена и функция $f$.

**Пример 5.1.1.** Доказать, что функция $x + y$ примитивно рекурсивна.

*Решение.* Обозначим функцию $x + y$ через $f(x, y)$. Функция $f$ удовлетворяет соотношениям:

$$\begin{cases} f(x, 0) = x = I_1^1(x), \\ f(x, y+1) = f(x, y) + 1 = \lambda\big(f(x, y)\big) = h\big(x, y, f(x, y)\big). \end{cases}$$

Функция $x + y$ получается из функций $I_1^1$ и

$$h(x, y, z) = \lambda\big(I_3^3(x, y, z)\big) = z + 1$$

с помощью операции примитивной рекурсии. Поскольку функции $I_1^1$, $\lambda$ и $I_3^3$ примитивно рекурсивны, а функция $h$ получена из функций $\lambda$ и $I_3^3$ с помощью операции суперпозиции, то функция $x + y$ примитивно рекурсивна.

**Пример 5.1.2.** Доказать, что функция $x * y$ примитивно рекурсивна.

*Решение*. Обозначим функцию $x * y$ через $g(x, y)$. Функция $g$ удовлетворяет соотношениям:

$$\begin{cases} g(x, 0) = x * 0 = 0 = O(x), \\ g(x, y + 1) = x * (y + 1) = x * y + x = g(x, y) + x \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} g(x, 0) = O(x), \\ g(x, y + 1) = g(x, y) + x = f(g(x, y), x) = h\big(x, y, g(x, y)\big). \end{cases}$$

В этих соотношениях функция $f$ является функцией суммирования из Примера 5.1.1, а

$$h(x, y, z) = f\big(I_3^3(x, y, z), I_3^1(x, y, z)\big) = z + x.$$

Итак, функция $x * y$ получается рекурсией из примитивно рекурсивной функции $O(x)$ и функции $h(x, y, z)$, полученной из функций $f$, $I_3^3$ и $I_3^1$ с помощью операции суперпозиции. Следовательно, функция $x * y$ примитивно рекурсивна.

**Пример 5.1.3.** Доказать, что функция $x^y$ примитивно рекурсивна.

*Решение*. Обозначим функцию $x^y$ через $u(x, y)$. Функция $u$ удовлетворяет следующим соотношениям:

$$\begin{cases} u(x, 0) = x^0 = 1 = \lambda\big(O(x)\big), \\ u(x, y + 1) = x^{y+1} = x^y * x = u(x, y) * x \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} u(x, 0) = \lambda\big(O(x)\big), \\ u(x, y + 1) = u(x, y) * x = g(u(x, y), x) = h\big(x, y, u(x, y)\big). \end{cases}$$

В этих соотношениях функция $g$ является функцией произведения из Примера 5.1.2, а

$$h(x, y, z) = g(I_3^3(x, y, z), I_3^1(x, y, z)) = z * x.$$

Функция $x^y$ получается рекурсией из примитивно рекурсивных функций $\lambda\big(O(x)\big)$ и $h(x, y, z)$, следовательно, функция $x^y$ примитивно рекурсивна.

***Операция минимизации** ($\mu$-**оператор**)*

Пусть $f(x_1, x_2, \dots, x_n, x_{n+1})$ – частичная числовая функция, $n \geq 1$, и пусть $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ – произвольный набор из $\mathbb{N}^n$. Введём новую функцию

$$g(x_1, x_2, \dots, x_n) = \mu_y(f(x_1, x_2, \dots, x_n, y) = 0).$$

Данная функция отыскивает для функции $f(x_1, x_2, \dots, x_n, x_{n+1})$ и фиксированного набора $\tilde{\alpha}$ значений переменных $x_1, x_2, \dots, x_n$ наименьшее значение $y$, при котором $f(\alpha_1, \alpha_2, \dots, \alpha_n, y) = 0$. В формальном виде, если существует $y_0 \in \mathbb{N}$, такой, что $f(\alpha_1, \alpha_2, \dots, \alpha_n, y_0) = 0$, а значения $f(\tilde{\alpha}, 0)$, $f(\tilde{\alpha}, 1)$, … , $f(\tilde{\alpha}, y_0 - 1)$ определены и отличны от 0, то $g(\tilde{\alpha}) = y_0$. В противном случае значение $g(\tilde{\alpha})$ считается неопределённым. Будем говорить, что функция $g(x_1, x_2, \dots, x_n)$ получена из функции $f(x_1, x_2, \dots, x_n, x_{n+1})$ при помощи операции *минимизации* (по переменной $x_{n+1}$).

Вообще функция $g(x_1, x_2, \dots, x_n)$ вычисляется методом подбора, подставляя вместо $y$ последовательно числа 0, 1, 2, …

Отметим, что значение $g(x_1, x_2, \dots, x_n)$ будет неопределённым в следующих случаях:

1) значение $f(x_1, x_2, \dots, x_n, 0)$ не определено;
2) значения $f(\tilde{\alpha}, 0), f(\tilde{\alpha}, 1), \dots, f(\tilde{\alpha}, y_0 - 1)$ определены, но отличны от 0, а значение $f(\tilde{\alpha}, y_0)$ не определено;
3) значения функции $f(\tilde{\alpha}, y)$ определены для всех $y = 0,1,2, \dots$ и отличны от 0.

Операцию минимизации можно применять по любой переменной.

**Определение 5.1.6.** *Функция $f(x_1, x_2, \dots, x_n)$ называется **частично рекурсивной**, если она может быть получена из простейших функций $\lambda$, $O$ и $I_n^m$ с помощью конечного числа применений операций суперпозиции, примитивной рекурсии и минимизации.*

Отметим, что всюду определённая частично рекурсивная функция называется *общерекурсивной*.

**Пример 5.1.4.** Доказать, что функция $x - y$ частично рекурсивна.

*Решение.* Обозначим функцию $x - y$ через $f(x, y)$. Выразим функцию $f$ при помощи оператора минимизации:

$$f(x, y) = \mu_z(y + z = x) = \mu_z\big(I_3^2(x, y, z) + I_3^3(x, y, z) = I_3^1(x, y, z)\big).$$

Функция $x - y$ получается в результате применения операции минимизации, частично рекурсивной функции сложения, и простейших функций $I_3^1$, $I_3^2$ и $I_3^3$. Следовательно, функция $f$ частично рекурсивна.

Для любой частично рекурсивной функции существует алгоритм вычисления её значений, т. е. все частично рекурсивные функции являются вычислимыми. Также справедлив и тезис Чёрча.

**Тезис Чёрча.** *Класс вычислимых частично числовых функций совпадает с классом всех частично рекурсивных функций.*

Тезис Чёрча не может быть строго доказан, но считается справедливым, поскольку он подтверждается опытом, накопленным в математике за всю её историю. Какие бы классы алгоритмов ни строились, вычисляемые ими числовые функции оказывались частично рекурсивными.

## 5.2 Машина Тьюринга

Ещё одной важной формализацией понятия алгоритма является машина Тьюринга, разработанная английским математиком Аланом Тьюрингом в 1936 г.

*Машина Тьюринга* представляет собой абстрактное устройство, состоящее из бесконечной в обе стороны ленты, считывающей и печатающей головки (автомата), способной перемещаться вправо и влево вдоль ленты, и управляющего устройства. Лента разбита на ячейки (клетки) и она используется для хранения информации. Считывающая и печатающая головка перемещается вдоль ленты так, что в каждый дискретный момент времени она обозревает ровно одну ячейку ленты. В каждой ячейке может быть записан только один символ из некоторого конечного алфавита

$$A = \{a_0, a_1, a_2, \dots, a_n\}.$$

Алфавит $A$ называется *внешним алфавитом* машины Тьюринга. Считается, что любая ячейка, которая не содержит символы алфавита $A$, содержит символ, называемый *пустым.* Ячейка, содержащая в данный момент пустой символ, называется *пустой ячейкой*. В качестве пустого символа используется символ $\Lambda$ (“лямбда”).

В каждый момент времени машина Тьюринга находится в некотором состоянии из *множества состояний*

$$Q = \{q_0, q_1, q_2, \dots, q_m\}.$$

Отметим, что множество состояний $Q$ также называется *внутренним алфавитом* машины. Изначально машина находится в состоянии $q_1$, а заключительным считается состояние $q_0$ (в этом состоянии машина останавливает своё выполнение).

Работа машины Тьюринга складывается из тактов (шагов), по ходу выполнения которых происходит преобразование информации, записанной на ленте. Каждый шаг машины осуществляется в соответствии с командой вида:

$$q, a \to q', a', D,$$

где, $a, a' \in A$, $q, q' \in Q$, $D \in \{R, L, N\}$, $R, L, N$ – вправо, влево, не двигаться.

Смысл команды таков: если машина находится в состоянии $q$ и считывает с ленты символ $a$, то машина переходит в состояние $q'$, печатает в текущей ячейке символ $a'$ и затем выполняет действие $D$. При этом, если $D = R$, то машина смещается на одну ячейку вправо, если $D = L$, – на одну ячейку влево, а если $D = N$, то машина остается на месте.

Конечный набор команд образует программу. Необходимо, чтобы в программе не было разных команд с одинаковыми входами, то есть команд вида:

$$q, a \to q', a', D' \text{ и } q, a \to q'', a'', D'',$$

так как это противоречит детерминированности алгоритма.

Обычно программа для машины Тьюринга записывается в виде следующей двумерной таблицы:

| | $a_1$ | … | $a_j$ | … | $a_n$ | $\Lambda$ |
|---|---|---|---|---|---|---|
| $q_1$ | | | | | | |
| … | | | | | | |
| $q_i$ | | | $q_{ij}, a_{ij}, D_{ij}$ | | | |
| … | | | | | | |
| $q_m$ | | | | | | |

Таблица 5.2.1

Слева перечисляются все состояния, в которых может находиться машина, сверху – все символы внешнего алфавита, в том числе и $\Lambda$ (какие именно символы и состояния указывать в таблице – определяет автор программы). На пересечениях линий и колонок (в ячейках таблицы) указываются команды для машины Тьюринга, которые должна выполнять машина, когда она находится в соответствующем состоянии и видит на ленте соответствующий символ. В целом данная таблица определяет действия машины Тьюринга и тем самым полностью задаёт её поведение.

Касательно выполнения программ на машине Тьюринга, будем придерживаться следующих двух соглашений:

1) Во-первых, входное слово, к которому будет применена программа, должно быть записано на ленту. *Входное слово* – это

конечная последовательность символов, записанных в соседних ячейках ленты. Внутри входного слова пустых ячеек быть не должно, а слева и справа от него должны быть только пустые ячейки. Пустое входное слово означает, что все ячейки ленты пусты.

2) Во-вторых, вначале выполнения программы машина находится в состояние $q_1$ и размещена она под первым левым символом входного слова (см. Рисунок 5.2.1). Если входное слово пустое, то машина может быть размещена под любой ячейкой.

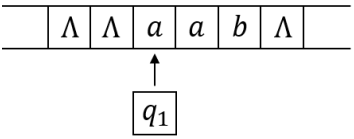


Рисунок 5.2.1

На каждом шаге выполнения программы машины Тьюринга, в таблице отыскивается ячейка на пересечении той строки, которая соответствует текущему состоянию машины и того столбца, который соответствует обозреваемому символу, и выполняется команда, указанная в этой ячейке. В результате автомат оказывается в новой конфигурации, для которой повторяются такие же действия. И так далее. (Под *конфигурацией* машины Тьюринга понимается совокупность состояния машины, состояния ленты и положения автомата на ленте)

Шаги работы машины повторяются до тех пор, пока не произойдёт одно из двух событий:

1) Для текущего состояния и обозреваемого символа нет команды.
2) Текущая команда привела к заключительному состоянию $q_0$. В этом случае слово, оказавшееся на ленте, считается результатом работы машины.

Машина называется *применимой* к некоторому входному слову, если при работе над этим словом после конечного числа шагов она останавливается. Если она никогда не остановится, то машина называется *неприменимой* к этому слову. Важно отметить, что в момент остановки внутри выходного слова не должно быть пустых ячеек, а машина должна остановиться под одним из символов выходного слова (если слово пустое – под любой ячейкой ленты).

Функция называется *вычислимой по Тьюрингу*, если существует машина Тьюринга, которая вычисляет эту функцию, т. е. такая машина Тьюринга, которая вычисляет её значение для тех наборов значений аргументов, на которых функция определена, и работающая вечно для наборов значений аргументов, на которых функция не определена.

**Теорема 5.2.1** *Функция вычислима по Тьюрингу тогда и только тогда, когда она частично рекурсивна.*

***Примеры программ для машины Тьюринга***

Рассмотрим примеры на составление программ для машины Тьюринга, чтобы продемонстрировать некоторые типичные приёмы программирования на машине Тьюринга.

Договоримся о некоторых сокращениях записи программ для машины Тьюринга. Если в такте (шаге) на ленте не меняется видимый символ, или машина не смещается, или не меняется состояние машины, то в соответствующей позиции такта мы не будем ничего писать.

Например, следующие записи команд эквивалентны:

$$q, a \to q, a, R \text{ и } q, a \to , , R$$

$$q, a \to q', a, L \text{ и } q, a \to q', , L$$

$$q, a \to q, a', N \text{ и } q, a \to , a',$$

Если заранее известно, что в процессе выполнения программы не может появиться некоторая конфигурация, тогда, чтобы подчеркнуть это явно, будем в соответствующей ячейке таблицы записывать прочерк.

В формулировке задач буквой $P$ будем обозначать входное слово.

**Пример 5.2.1.** Выяснить, применима ли машина Тьюринга, задаваемая программой из Таблицы 5.2.2, к входному слову $P$:

a) $P_1 = 11101$;

b) $P_2 = 110$.

Если применима, то установить результат применения машины к слову $P$.

| | 0 | 1 | Λ |
|---|---|---|---|
| $q_1$ | $, , R$ | $q_2, 0, R$ | $, , L$ |
| $q_2$ | $q_0, 1, N$ | $q_1, 0, R$ | $q_0, 1, R$ |

Таблица 5.2.2

*Решение.* Пошагово опишем результат применения машины Тьюринга к словам $P_1$ и $P_2$:

$$P_1{:}\,\Lambda\,1\,1\,1\,0\,1\,\Lambda \to \Lambda\,0\,1\,1\,0\,1\,\Lambda \to \Lambda\,0\,0\,1\,0\,1\,\Lambda \to$$
$$\uparrow q_1 \qquad \uparrow q_2 \qquad \uparrow q_1$$
$$\to \Lambda\,0\,0\,0\,0\,1\,\Lambda \to \Lambda\,0\,0\,0\,1\,1\,\Lambda.$$
$$\uparrow q_2 \qquad \uparrow q_0$$

$$P_2{:}\,\Lambda\,1\,1\,0\,\Lambda \to \Lambda\,0\,1\,0\,\Lambda \to \Lambda\,0\,0\,0\,\Lambda \to \Lambda\,0\,0\,0\,\Lambda\,\Lambda \to \Lambda\,0\,0\,0\,\Lambda \to \ldots$$
$$\uparrow q_1 \qquad \uparrow q_2 \qquad \uparrow q_1 \qquad \uparrow q_1 \qquad \uparrow q_1$$

В первом случае машина останавливается и образует выходное слово 00011. Во втором случае машина никогда не остановится и будет работать вечно. Таким образом, она неприменима к слову $P_2$, но применима к слову $P_1$.

**Пример 5.2.2.** Пусть $A = \{0,1\}$ и $P$ – непустое слово. Требуется приписать слева к слову $P$ символ 1, а справа символы 10.

Например: $10101 \rightarrow 11010110$.

*Решение.* Для решения этой задачи нужно выполнить два действия:

1. Переместить машину на одну ячейку левее крайнего левого символа слова $P$ и записать в ней 1.
2. Перегнать машину под первую ячейку после крайнего правого символа слова $P$, записать в ней 1, затем переместить машину на одну ячейку правее и записать 0, после чего остановить машину.

На основе этих действий запишем программу для машины Тьюринга (Таблица 5.2.3).

| | 0 | 1 | Λ | *комментарий* |
|---|---|---|---|---|
| $q_1$ | $,,L$ | $,,L$ | $q_2,1,R$ | перемещение на одну ячейку левее и запись 1 |
| $q_2$ | $,,R$ | $,,R$ | $q_3,1,R$ | перемещение к концу слова $P$ и запись 1 справа |
| $q_3$ | – | – | $q_0,0,$ | запись 0 справа |

Таблица 5.2.3

Например, результатом применения этой программы для слова $P = 10101$ будет:

$$P{:}\,\Lambda\,1\,0\,1\,0\,1\,\Lambda \rightarrow \Lambda\,\Lambda\,1\,0\,1\,0\,1\,\Lambda \rightarrow \Lambda\,1\,1\,0\,1\,0\,1\,\Lambda \rightarrow \ldots \rightarrow$$
$$\uparrow q_1 \qquad \uparrow q_1 \qquad \uparrow q_2$$

$$\rightarrow \Lambda\,1\,1\,0\,1\,0\,1\,\Lambda\,\Lambda \rightarrow \Lambda\,1\,1\,0\,1\,0\,1\,1\,\Lambda\,\Lambda \rightarrow \Lambda\,1\,1\,0\,1\,0\,1\,1\,0\,\Lambda.$$
$$\uparrow q_2 \qquad \uparrow q_3 \qquad \uparrow q_0$$

**Пример 5.2.3.** Пусть $A = \{0,1,2,3,4,5,6,7\}$ и $P$ – непустое слово. То есть, $P$ – это неотрицательное целое число в восьмеричной системе исчисления. Требуется получить на ленте запись числа, которое на 1 больше числа $P$.

*Решение.* Для решения этой задачи нужно выполнить следующие действия:

1. Перегнать машину под последнюю цифру числа.

2. Если это цифра от 0 до 6, то заменить её цифрой на 1 больше и остановиться. Например:

$$\begin{array}{ccccccc} \Lambda & 3 & 1 & 4 & \Lambda & \rightarrow \Lambda & 3 & 1 & 4 & \Lambda & \rightarrow & \Lambda & 3 & 1 & 5 & \Lambda . \\ & \uparrow & & & & & \uparrow & & & & & & \uparrow & & & & \end{array}$$

3. Если последней цифрой является 7, то заменить её на 0 и сдвинуть машину к предыдущей цифре, после чего аналогичным образом увеличить предпоследнюю цифру на 1. Например:

$$\begin{array}{ccccccc} \Lambda & 5 & 0 & 2 & 7 & \Lambda \rightarrow \Lambda & 5 & 0 & 2 & 7 & \Lambda \rightarrow \Lambda & 5 & 0 & 2 & 0 & \Lambda \rightarrow \Lambda & 5 & 0 & 3 & 0 & \Lambda . \\ & \uparrow & & & & & \uparrow & & & & & \uparrow & & & & & \uparrow & & & & & \end{array}$$

4. В случае, когда в слове $P$ встречаются только семёрки, машина будет сдвигаться влево, заменяя семёрки на нули, и в конце концов окажется под пустой ячейкой. В эту пустую ячейку надо записать 1 и остановить машину. Например:

$$\begin{array}{ccccccc} \Lambda & 7 & 7 & 7 & \Lambda \rightarrow \Lambda & 7 & 7 & 7 & \Lambda \rightarrow \ldots \rightarrow \Lambda & \Lambda & 0 & 0 & 0 & \Lambda \rightarrow \Lambda & 1 & 0 & 0 & 0 & \Lambda . \\ & \uparrow & & & & & \uparrow & & & & & \uparrow & & & & & \uparrow & & & & \end{array}$$

Эти действия могут быть записаны в виде программы следующим образом:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\Lambda$ |
|---|---|---|---|---|---|---|---|---|---|
| $q_1$ | $,,R$ | $,,R$ | $,,R$ | $,,R$ | $,,R$ | $,,R$ | $,,R$ | $,,R$ | $q_2,,L$ |
| $q_2$ | $q_0,1,$ | $q_0,2,$ | $q_0,3,$ | $q_0,4,$ | $q_0,5,$ | $q_0,6,$ | $q_0,7,$ | $,0,L$ | $q_0,1,$ |

Таблица 5.2.4

В данной программе состояние $q_1$ используется для перемещения машины под последнюю цифру числа. Не меняя видимые цифры и дойдя до первой пустой ячейки, машина возвращается назад под последнюю цифру и переходит в состояние $q_2$. В свою очередь, состояние $q_2$ используется машиной для прибавления 1, согласно вышеуказанным действиям 2, 3 и 4.

**Пример 5.2.4.** Пусть $A = \{a, b, c\}$. Если первый и последний символы слова $P$ одинаковы, тогда это слово не менять, а иначе заменить его на пустое слово.

*Решение.* Для решения этой задачи нужно выполнить следующие шаги:

1. Запомнить первый символ входного слова.
2. Переместить машину под последний символ и сравнить его с запомненным. Если они равны, то больше ничего не делать.
3. В противном случае, если запомненный и последний символы не равны, то стереть всё входное слово.

Для запоминания первого символа можно использовать разные состояния машины. Мы будем использовать состояния $q_2$, $q_4$ и $q_6$ для

сохранения символов $a$, $b$ и $c$ соответственно. Если после перегона машины под последний символ определяется, что запомненный и последний символы не равны, то входное слово стирается путём замены всех его символов на символ $\Lambda$. При этом машина будет перемещаться справа налево до первой пустой ячейки. Для стирания слова будем использовать отдельное состояние $q_8$.

С учётом вышесказанного программа для машины Тьюринга записывается так:

| | $a$ | $b$ | $c$ | $\Lambda$ | *комментарий* |
|---|---|---|---|---|---|
| $q_1$ | $q_2,,$ | $q_4,,$ | $q_6,,$ | $q_0,,$ | анализ первого символа, разветвление |
| $q_2$ | $,,R$ | $,,R$ | $,,R$ | $q_3,,L$ | перемещение к последнему символу при первом символе $a$ |
| $q_3$ | $q_0,,$ | $q_8,,$ | $q_8,,$ | $-$ | если последний символ отличен от $a$, то переход в состояние $q_8$ (стереть $P$) |
| $q_4$ | $,,R$ | $,,R$ | $,,R$ | $q_5,,L$ | перемещение к последнему символу при первом символе $b$ |
| $q_5$ | $q_8,,$ | $q_0,,$ | $q_8,,$ | $-$ | если последний символ отличен от $b$, то переход в состояние $q_8$ (стереть $P$) |
| $q_6$ | $,,R$ | $,,R$ | $,,R$ | $q_7,,L$ | перемещение к последнему символу при первом символе $c$ |
| $q_7$ | $q_8,,$ | $q_8,,$ | $q_0,,$ | $-$ | если последний символ отличен от $c$, то переход в состояние $q_8$ (стереть $P$) |
| $q_8$ | $,\Lambda,L$ | $,\Lambda,L$ | $,\Lambda,L$ | $q_0,,$ | стереть слово, двигаясь справа на лево |

Таблица 5.2.5

**Пример 5.2.5.** Пусть $A = \{a, b, c\}$. Вставить в слово $P$ символ $b$ за первым вхождением символа $c$, если такое есть.

Например: $abca \to abcba$.

*Решение.* Для решения этой задачи нужно выполнить следующие шаги:

1. Просматривать входное слово слева направо до тех пор, пока не встретится пустая ячейка или символ $c$:

$$\begin{matrix}\Lambda\, a\, b\, c\, a\, \Lambda \;\to\; \ldots \;\to\; \Lambda\, a\, b\, c\, a\, \Lambda\,.\\ \uparrow \qquad\qquad\qquad\qquad \uparrow\end{matrix}$$

2. Если встретилась пустая ячейка, то больше ничего не делать.
3. В противном случае, если встретился символ $c$, то освободить место для вставляемого символа $b$, для чего сдвигается начало слова $P$ (от первого символа до найденного символа $c$) на одну позицию влево:

$$\Lambda\, a\, b\, c\, a\, \Lambda \;\rightarrow\; \Lambda\, a\, b\, b\, a\, \Lambda \rightarrow\; \Lambda\, a\, c\, b\, a\, \Lambda \;\rightarrow\; \Lambda\, \Lambda\, b\, c\, b\, a\, \Lambda \;\rightarrow\; \Lambda\, a\, b\, c\, b\, a\, \Lambda\,.$$
$$\uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow$$

Введём некоторые разъяснения по поводу сдвига, указанного в третьем шаге. Этот сдвиг осуществляем справа налево – от символа $c$ к началу слова. Начинаем этот сдвиг с записи символа $b$ вместо найденного символа $c$, перевода машины в состояние, соответствующее сохранению символа $c$ и перемещения машины на одну ячейку влево. Затем записываем в обозреваемую ячейку символ, ассоциированный текущему состоянию машины, а обозреваемый символ сохраняем путём перевода состояния машины в новое состояние, соответствующее данному символу. Повторение таких тактов приведёт к сдвигу влево на одну позицию всех символов начиная с первого вхождения символа $c$. Процесс сдвига будет происходить до тех пор, пока машина не достигнет пустой ячейки. В пустую ячейку машина запишет символ, сохранённый из предыдущей ячейки, и остановит своё выполнение.

В итоге получается следующая программа для машины Тьюринга:

| | $a$ | $b$ | $c$ | $\Lambda$ | *комментарий* |
|---|---|---|---|---|---|
| $q_1$ | $,,R$ | $,,R$ | $q_4,b,L$ | $q_0,,L$ | перемещение вправо до $c$, вставка $b$, перенос $c$ влево |
| $q_2$ | $,,L$ | $q_3,a,L$ | $q_4,a,L$ | $q_0,a,$ | перенос $a$ справа |
| $q_3$ | $q_2,b,L$ | $,,L$ | $q_4,b,L$ | $q_0,b,$ | перенос $b$ справа |
| $q_4$ | $q_2,c,L$ | $q_3,c,L$ | $,,L$ | $q_0,c,$ | перенос $c$ справа |

Таблица 5.2.6

**Пример 5.2.6.** Пусть $A = \{a, b\}$. Удвоить слово $P$, поставив между ним и его копией знак $=$.

Например: $abb \rightarrow abb = abb$.

*Решение.* Для того чтобы зафиксировать на ленте некоторые позиции символов $a$ и $b$, к которым при выполнении программы машины Тьюринга нужно будет возвратиться, будем использовать дополнительные символы $A$ и $B$ (двойники символов $a$ и $b$).

Предполагается выполнить следующие действия:

1. Вначале записать знак $=$ за входным словом.
2. Затем возвратить машину под первый символ входного слова:

$$\Lambda\, a\, b\, b\, \Lambda \;\rightarrow\; \ldots \;\rightarrow\; \Lambda\, a\, b\, b \;=\; \Lambda \;\rightarrow\; \ldots \;\rightarrow\; \Lambda\, a\, b\, b \;=\; \Lambda\,.$$
$$\uparrow \qquad\qquad\qquad\qquad \uparrow \qquad\qquad\qquad\qquad \uparrow$$

3. Если обозреваемым является символ $a$ (если обозреваемым является символ $b$), то заменить его на символ $A$ ($B$), переместить машину вправо до первой свободной ячейки и записать в неё символ $a$ ($b$). После этого возвратить машину влево к ячейке с символом $A$ ($B$), восстановить прежний символ $a$ ($b$) и сдвинуть машину вправо к следующему символу. И аналогичным образом копировать все последующие символы входного слова.
4. После копирования последнего символа входного слова и возврата к его двойнику, машина сдвигается на одну позицию вправо и попадёт под знак =. Это означает, что входное слово полностью скопировано и машина Тьюринга завершает свою работу.

$$\Lambda\, a\, b\, b = \Lambda \rightarrow \ldots \rightarrow \Lambda\, A\, b\, b = a\, \Lambda \rightarrow \ldots \rightarrow \Lambda\, A\, b\, b = a\, \Lambda \rightarrow$$
$$\rightarrow \Lambda\, a\, b\, b = a\, \Lambda \rightarrow \ldots \rightarrow \Lambda\, a\, B\, b = a\, b\, \Lambda \rightarrow \ldots \rightarrow \Lambda\, a\, B\, b = a\, b\, \Lambda \rightarrow$$
$$\rightarrow \Lambda\, a\, b\, b = a\, b\, \Lambda \rightarrow \ldots \rightarrow \Lambda\, a\, b\, B = a\, b\, b\, \Lambda \rightarrow \ldots \rightarrow$$
$$\rightarrow \Lambda\, a\, b\, B = a\, b\, b\, \Lambda \rightarrow \Lambda\, a\, b\, b = a\, b\, b\, \Lambda\,.$$

С учётом сказанного получаем следующую программу для машины Тьюринга:

| | $a$ | $b$ | $=$ | $A$ | $B$ | $\Lambda$ | *комментарий* |
|---|---|---|---|---|---|---|---|
| $q_1$ | $,,R$ | $,,R$ | $-$ | $-$ | $-$ | $q_2,=,L$ | поставить = справа от слова |
| $q_2$ | $,,L$ | $,,L$ | $-$ | $-$ | $-$ | $q_3,,R$ | перемещение к первому символу |
| $q_3$ | $q_4,A,R$ | $q_5,B,R$ | $q_0,,$ | $-$ | $-$ | $-$ | анализ и замена символа на двойника |
| $q_4$ | $,,R$ | $,,R$ | $,,R$ | $-$ | $-$ | $q_6,a,$ | запись $a$ справа |
| $q_5$ | $,,R$ | $,,R$ | $,,R$ | $-$ | $-$ | $q_6,b,$ | запись $b$ справа |
| $q_6$ | $,,L$ | $,,L$ | $,,L$ | $q_3,a,R$ | $q_3,b,R$ | $-$ | возврат, восстановление, переход к следующему |

Таблица 5.2.7

## 5.3 Нормальные алгоритмы Маркова

Понятие нормального алгоритма Маркова (НАМ) было введено в 40-х годах XX века советским математиком А. А. Марковым с целью формализации понятия алгоритма. НАМ примечательны тем, что в них используется лишь одна элементарная операция – так называемая подстановка.

Пусть дан алфавит $A$ и слова в этом алфавите. *Формулой подстановки* называется запись вида $\alpha \to \beta$ (читается: "$\alpha$ заменить на $\beta$"), где $\alpha$ и $\beta$ – любые слова (возможно, и пустые). Слова $\alpha$ и $\beta$ называются соответственно *левой* и *правой частью подстановки*.

Формула подстановки применяется для преобразования некоторого слова $P$ из алфавита $A$. Сама операция подстановки сводится к тому, что в слове $P$ отыскивается часть, совпадающая с левой частью подстановки (т. е. с $\alpha$), и она заменяется на правую часть подстановки (т. е. на $\beta$). При этом остальные части слова $P$ (слева и справа от $\alpha$) не меняются. Получившееся слово $R$ называют *результатом подстановки*. Условно это можно изобразить так:

$$P: x\alpha y \to R: x\beta y.$$

Внесём некоторые уточнения:

1. Если левая часть формулы подстановки входит в слово $P$, то говорят, что эта формула *применима* к $P$. В противном случае формула считается *неприменимой* к $P$, и подстановка не выполняется.
2. Если левая часть $\alpha$ входит в слово $P$ несколько раз, то на правую часть $\beta$ заменяется только первое (самое левое) вхождение $\alpha$ в $P$:

$$P: x\alpha y\alpha z \to R: x\beta y\alpha z.$$

3. Если правая часть $\beta$ – пустое слово, то подстановка $\alpha \to$ сводится к вычёркиванию первого вхождения $\alpha$ из $P$:

$$P: x\alpha y\alpha z \to R: xy\alpha z.$$

4. Если левая часть $\alpha$ – пустое слово, то подстановка $\to \beta$ сводится к приписыванию $\beta$ слева к слову $P$:

$$P: x \to R: \beta x.$$

Из этого правила вытекает очень важный факт: формула с пустой левой частью применима к любому слову. Отметим также, что формула с пустыми левой и правой частями не меняет слово.

*Нормальным алгоритмом Маркова* (НАМ) называется непустой конечный упорядоченный набор формул подстановки:

$$\begin{cases} \alpha_1 \to \beta_1 \\ \alpha_2 \to \beta_2 \\ \quad \dots \\ \alpha_k \to \beta_k \end{cases} \quad (k \geq 1)$$

В этих формулах могут использоваться два вида стрелок: обычная стрелка (→) и стрелка "с хвостиком" (↦). Формула с обычной стрелкой называется *обычной формулой*, а формула, записанная с использованием стрелки "с хвостиком", *заключительной формулой*. Заключительные формулы используются для остановки выполнения НАМ.

Перед выполнением НАМ, задаётся некоторое входное слово $P$. Работа НАМ сводится к выполнению последовательности шагов. На каждом шаге формулы подстановки, входящие в НАМ, просматриваются сверху вниз и выбирается первая (самая верхняя) из формул, применимых к входному слову $P$. Далее выполняется подстановка согласно найденной формуле. В итоге получается новое слово $P'$. На следующем шаге это слово $P'$ берется за исходное и к нему применяется та же самая процедура, в результате которой получается новое слово $P''$. И так далее:

$$P \to P' \to P'' \to \dots$$

Работа НАМ заканчивается в двух случаях:

1. На очередном шаге к текущему слову неприменима ни одна формула.
2. Была применена заключительная формула.

То и другое считается "хорошим" окончанием работы НАМ. В обоих случаях говорят, что НАМ *применим* к входному слову. Слово, полученное в результате окончания работы НАМ, есть *выходное слово*.

Впрочем, может случиться и так, что НАМ никогда не остановится. Это происходит, если на каждом шаге существует применимая формула и эта формула с обычной стрелкой. В этом случае говорят, что НАМ *неприменим* к входному слову.

Заметим, что в процессе выполнения НАМ в обрабатываемых словах могут появляться символы, которые не входят в $A$.

Функция $f$, заданная на некотором множестве слов алфавита $A$, называется *нормально вычислимой*, если найдётся такое расширение $B$ данного алфавита ($A \subseteq B$) и такой нормальный алгоритм Маркова в $B$, что каждое слово $P$ (в алфавите $A$) из области определения функции $f$ этот алгоритм перерабатывает в слово $f(P)$.

Имеет место следующая теорема.

**Теорема 5.3.1** *Функция нормально вычислима тогда и только тогда, когда она вычислима по Тьюрингу.*

***Примеры НАМ***

Рассмотрим примеры на составление НАМ. В НАМ справа от формул подстановки будем указывать их номера. Эти номера нужны для ссылок на формулы при показе пошагового выполнения НАМ.

**Пример 5.3.1.** $A = \{a, b, c, d\}$. В слове $P$ требуется заменить первое вхождение подслова $abb$ на $cc$ и удалить все вхождения символа $d$.

Например: $cdabbdcabb \to ccccabb$.

*Решение.* Поставленную задачу решает следующий НАМ:

$$\begin{cases} d \to & (1) \\ abb \mapsto cc & (2) \end{cases}$$

Проверим этот НАМ на входном слове $cdabbdcabb$ (над стрелками указаны номера применённых формул, а для наглядности в словах слева от стрелок подчёркнуты те части, к которым были применены эти формулы):

$$c\underline{d}abbdcabb \overset{1}{\to} cabb\underline{d}cabb \overset{1}{\to} c\underline{abb}cabb \overset{2}{\mapsto} ccccabb.$$

**Пример 5.3.2.** $A = \{a, b, c\}$. Преобразовать слово $P$ так, чтобы в его начале оказались все символы $a$, потом шли все символы $b$, а в конце – все символы $c$.

Например: $cbabba \to aabbbc$.

*Решение.* По факту, требуется отсортировать в алфавитном порядке символы, входящие во входное слово $P$. Для этого составим следующий НАМ:

$$\begin{cases} ba \to ab & (1) \\ ca \to ac & (2) \\ cb \to bc & (3) \end{cases}$$

Как только слово будет отсортировано, НАМ остановится. Проверим этот НАМ на входном слове $cbabba$:

$$c\underline{ba}bba \overset{1}{\to} cabb\underline{ba} \overset{1}{\to} cab\underline{ba}b \overset{1}{\to} ca\underline{ba}bb \overset{1}{\to} \underline{ca}abbb \overset{2}{\to} a\underline{ca}bbb \overset{2}{\to}$$
$$\overset{2}{\to} aa\underline{cb}bb \overset{3}{\to} aab\underline{cb}b \overset{3}{\to} aabb\underline{cb} \overset{3}{\to} aabbbc.$$

**Пример 5.3.3.** $A = \{a, b\}$. Удалить первые два символа слова $P$. Пустое слово или слово, состоящее из одного символа, не менять.

*Решение.* В начале нужно пометить первый символ слова $P$. Для этого вставим перед ним какой-либо знак, скажем $*$, отличный от символов

алфавита $A$. Вставку знака $*$ перед первым символом реализуем, воспользовавшись формулой $\to *$ с пустой левой частью, которая приписывает свою правую часть слева к слову. После этого можно с помощью формул вида $*\,xy \mapsto$ , где $x, y \in A$, заменить знак $*$ и первые два символа слова на пустоту и остановить алгоритм:

$$babba \to *\,babba \mapsto bba.$$

В случае, когда слово $P$ является пустым или состоит из одного символа, нужно стереть знак $*$ и остановить алгоритм. Это делает формула вида $* \mapsto$.

Получаем следующий НАМ:

$$\begin{cases} *\,aa \mapsto & (1) \\ *\,ab \mapsto & (2) \\ *\,ba \mapsto & (3) \\ *\,bb \mapsto & (4) \\ * \mapsto & (5) \\ \to * & (6) \end{cases}$$

Легко можно убедится в том, что НАМ выполняется корректно в случае, когда $P$ – пустое слово, $P$ состоит из одного символа, либо $P$ содержит не менее двух символов:

$$P_1 : \overset{6}{\to} \underline{*} \overset{5}{\mapsto}.$$

$$P_2 : a \overset{6}{\to} \underline{*}\, a \overset{5}{\mapsto} a.$$

$$P_3 : bbab \overset{6}{\to} \underline{* bb}ab \overset{4}{\mapsto} ab.$$

**Пример 5.3.4.** $A = \{a, b\}$. Приписать $abab$ справа к слову $P$.

Например: $abb \to abbabab$.

*Решение.* Чтобы приписать $abab$ справа, нужно пометить конец слова $P$. Для этого нужно сначала приписать специальный знак, к примеру $*$, слева к слову $P$, а затем перегнать его в конец слова. Такой перегон реализуется с помощью формул вида $*\,\xi \to \xi\,*$, где $\xi \in A$. После чего остаётся заменить $*$ на $abab$ и остановится.

Получим следующий НАМ:

$$\begin{cases} *\,a \to a\,* & (1) \\ *\,b \to b\,* & (2) \\ * \mapsto abab & (3) \\ \to * & (4) \end{cases}$$

Проверим этот НАМ на входном слове $abb$:

$$abb \overset{4}{\to} \underline{*\,a}bb \overset{1}{\to} a\,\underline{*\,b}b \overset{2}{\to} ab\,\underline{*\,b} \overset{2}{\to} abb\,\underline{*} \overset{3}{\mapsto} abbabab.$$

**Пример 5.3.5.** $A = \{a, b\}$. В слове $P$ заменить на $aaa$ последнее вхождение символа $b$, если такое есть.

Например: $ababaa \to abaaaaaa$.

*Решение.* Сначала нужно пометить конец слова $P$. Для этого пометим начало слова $P$ знаком $*$ и перегоним его в конец слова, воспользовавшись процедурой, описанной в Примере 5.3.4. Затем заменим $*$ на какой-либо другой знак, скажем #, который будем перемещать влево, таким образом, маркируя текущую позицию в слове $P$. Если по ходу движения влево встречаем символ $b$, то производим его замену на $aaa$ и останавливаем алгоритм. В противном случае останавливаем алгоритм по достижению начала слова $P$.

Реализуем эту идею в виде следующего НАМ:

$$\begin{cases} *\,a \to a\,* & (1) \\ *\,b \to b\,* & (2) \\ * \to \# & (3) \\ a\# \to \#a & (4) \\ b\# \mapsto aaa & (5) \\ \# \mapsto & (6) \\ \to * & (7) \end{cases}$$

Проверим этот алгоритм на входном слове $ababaa$:

$$ababaa \overset{7}{\to} \underline{*\,a}ababaa \overset{1}{\to} \ldots \overset{1}{\to} ababaa\,\underline{*} \overset{3}{\to} ababaa\underline{\#} \overset{4}{\to} abab\underline{a\#}a \overset{4}{\to}$$
$$\overset{4}{\to} aba\underline{b\#}aa \overset{5}{\mapsto} abaaaaaa.$$

Если же во входное слово не входит символ $b$ (к примеру, возьмём слово $aa$), тогда имеем:

$$aa \overset{7}{\to} \underline{*\,a}a \overset{1}{\to} a\,\underline{*\,a} \overset{1}{\to} aa\,\underline{*} \overset{3}{\to} a\underline{a\#} \overset{4}{\to} \underline{a\#}a \overset{4}{\to} \underline{\#}aa \overset{6}{\mapsto} aa.$$

**Пример 5.3.6.** $A = \{a, b\}$. Приписать в конец слова $P$ символ, идентичный первому символу слова $P$. Если $P$ – пустое слово, записать символ $a$.

*Решение.* Для решения этой задачи предлагается выполнить следующие действия:

1. Помечаем начало слова $P$ знаком $*$.
2. Заменяем $*$ на новым знак: на $A$ если $a$ – первый символ из $P$, на $B$ если первым символом в $P$ является символ $b$. Этим мы вводим два

новых знака $A$ и $B$, нужных для того, чтобы сохранить первый символ и перенести его в конец слова.

3. Перегоняем новый символ $A$ или $B$ через все символы слова $P$ в его конец.
4. Наконец, заменяем $A$ или $B$ в конце слова на прежний символ ($A$ на $a$, $B$ на $b$) и останавливаем алгоритм.

Все описанные действия реализуются в виде следующего НАМ:

$$\begin{cases} *\,a \to Aa & (1) \\ *\,b \to Bb & (2) \\ Aa \to aA & (3) \\ Ab \to bA & (4) \\ Ba \to aB & (5) \\ Bb \to bB & (6) \\ A \mapsto a & (7) \\ B \mapsto b & (8) \\ * \mapsto a & (9) \\ \to * & (10) \end{cases}$$

Здесь формулы (1) и (2) заменяют первый символ (вместе с $*$) на $Aa$ или $Bb$. Формулы (3) – (6) перегоняют $A$ и $B$ в конец слова. Когда $A$ и $B$ окажутся в конце слова, формулы (7) и (8) вставляют символ идентичный первому символу слова $P$. Формула (9) нужна на случай пустого входного слова, а формула (10) вставляет $*$ перед первым символом.

Проверим полученный НАМ на входном слове $baab$:

$$baab \overset{10}{\to} \underline{*\,b}aab \overset{2}{\to} \underline{Bb}aab \overset{6}{\to} b\underline{Ba}ab \overset{5}{\to} ba\underline{Ba}b \overset{5}{\to} baa\underline{Bb} \overset{6}{\to} baab\underline{B} \overset{8}{\mapsto} baabb.$$

Если же входное слово является пустым, тогда:

$$\overset{10}{\to}\ \underline{*}\ \overset{9}{\mapsto} a.$$

Представим и другое решение для данной задачи:

$$\begin{cases} *\,a \to \#aa & (1) \\ *\,b \to \#bb & (2) \\ \#aa \to a\#a & (3) \\ \#ab \to b\#a & (4) \\ \#ba \to a\#b & (5) \\ \#bb \to b\#b & (6) \\ \# \mapsto & (7) \\ * \mapsto a & (8) \\ \to * & (9) \end{cases}$$

Здесь кроме символа $*$ используется лишь один дополнительный знак – #. Вместо символов $A$ и $B$ используются пары $\#a$ и $\#b$. Что же касается перемещения этих пар через символы слова, то оно реализуется формулами вида: $\#a\xi \to \xi\#a$ и $\#b\xi \to \xi\#b$, где $\xi \in A$.

Проверим этот алгоритм на прежнем входном слове $baab$:

$$baab \overset{9}{\to} \underline{*\,b}aab \overset{2}{\to} \underline{\#bb}aab \overset{6}{\to} b\underline{\#ba}ab \overset{5}{\to} ba\underline{\#ba}b \overset{5}{\to} baa\underline{\#bb} \overset{6}{\to} \overset{6}{\to} baab\underline{\#b} \overset{7}{\mapsto} baabb.$$

Легко можно убедится и в том, что для пустого слова этот НАМ также выполняется верно.

**Пример 5.3.7.** $A = \{a, b\}$. Удвоить слово $P$, приписав справа от $P$ его копию.

Например: $baa \to baabaa$.

*Решение.* Нужно выполнить следующие действия:

1. Приписываем справа от слова $P$ знак $=$, который будет использован для отделения входного слова $P$ от его копии.
2. Просматривая по очереди все символы слова $P$ и, не уничтожая их, переносим копию каждого символа в конец. Для запоминания позиции скопированного символа воспользуемся символом #, а переносить в конец будем соответствующею ему копию ($A$ соответствует символу $a$ и $B$ соответствует символу $b$).
3. Удаляем знак $=$ и останавливаем алгоритм.

Получим следующий НАМ:

$$\begin{cases} *\,a \to a\,* & (1) \\ *\,b \to b\,* & (2) \\ * \to = & (3) \\ Aa \to aA & (4) \\ Ab \to bA & (5) \\ A = \to = A & (6) \\ A \to a & (7) \\ Ba \to aB & (8) \\ Bb \to bB & (9) \\ B = \to = B & (10) \\ B \to b & (11) \\ \#a \to a\#A & (12) \\ \#b \to b\#B & (13) \\ \# = \mapsto & (14) \\ \to\ \#\,* & (15) \end{cases}$$

Здесь формулы (1) – (3) перегоняют символ $*$ в конец входного слова и заменяют его на символ $=$. Формулы (4) – (11) перегоняют символы $A$ и $B$ в конец слова, после чего заменяют их на $a$ и, соответственно, на $b$. Формулы (12) и (13) вводят символы $A$ и $B$, сообразные тому символу входного слова, который должен быть скопирован следующим. Формула (14) применяется только тогда, когда полностью просмотрено всё входное слово. И, наконец, формула (15) вводит сразу два специальных символа # и *.

Проверим данный алгоритм на входном слове $baa$:

$$baa \overset{15}{\to} \# * \underline{b}aa \overset{2}{\to} \#b * \underline{a}a \overset{1}{\to} \#ba * \underline{a} \overset{1}{\to} \#baa * \overset{3}{\to} \underline{\#b}aa = \overset{13}{\to}$$

$$\overset{13}{\to} b\#\underline{Ba}a = \overset{8}{\to} b\#a\underline{Ba} = \overset{8}{\to} b\#aa\underline{B} = \overset{10}{\to} b\#aa = \underline{B} \overset{11}{\to} b\underline{\#a}a = b \overset{12}{\to}$$

$$\overset{12}{\to} ba\#\underline{Aa} = b \overset{4}{\to} \ldots \overset{7}{\to} baa\# = baa \overset{14}{\mapsto} baabaa.$$

Представим ещё одно решение для задачи удвоения слова:

$$\begin{cases} * a \to aA * & (1) \\ * b \to bB * & (2) \\ * \to \ \# & (3) \\ Aa \to aA & (4) \\ Ab \to bA & (5) \\ Ba \to aB & (6) \\ Bb \to bB & (7) \\ A\# \to \#a & (8) \\ B\# \to \#b & (9) \\ \# \mapsto & (10) \\ \to * & (11) \end{cases}$$

В этом НАМ выполняются следующие действия:

1. Сначала за каждой буквой входного слова вставляем её двойник (соответствующую большую букву). Для этого приписываем слева к входному слову знак $*$, а затем переносим его через буквы так, чтобы к каждой букве слова приписывался её двойник. В конце слова заменяем $*$ на новый знак #, который понадобится для третьего действия:

$$baa \to * baa \to bB * aa \to bBA * a \to bBaAaA * \to bBaAaA\#.$$

2. В получившемся слове переставляем малые и большие буквы так, чтобы слева оказались все малые буквы, а справа – все большие, сохраняя при этом исходный взаимный порядок как среди малых, так и среди больших букв:

$$bBaAaA\# \rightarrow \ \ldots \ \rightarrow baaBAA\#.$$

3. Осталось только заменить большие буквы на малые. Для этого будем перемещать знак # влево через каждую большую букву с заменой её на малую. Как только слева от # не окажется большой буквы, удалим знак # и остановим алгоритм:

$$baaBAA\# \rightarrow \ baaBA\#a \rightarrow baaB\#aa \rightarrow baa\#baa \mapsto baabaa.$$

## 5.4 Неразрешимые алгоритмические проблемы

В теории алгоритмов рассматривается вопрос об алгоритмической разрешимости изучаемых проблем. Этот вопрос можно сформулировать так: существует ли машина Тьюринга, решающая данную проблему или же такой машины не существует?

На этот вопрос теория алгоритмов в ряде случаев даёт отрицательный ответ. Приведём несколько примеров подобных алгоритмически неразрешимых проблем.

***Проблема распознавания выводимости***

Аксиоматический метод в математике заключается в том, что все теоремы данной теории получаются посредством вывода из нескольких аксиом, принимаемых в данной теории без доказательств. Например, в математической логике описывается язык формул, позволяющий любое предложение математической теории записать в виде вполне определённой формулы, а процесс логического вывода следствия $B$ из посылки $A$ представляется в виде цепочки формальных преобразований исходной формулы.

Вопрос о логической выводимости следствия $B$ из посылки $A$ является вопросом о существовании дедуктивной цепочки, ведущей от формулы $A$ к формуле $B$. В связи с этим возникает *проблема распознавания выводимости*: существует ли для двух формул $A$ и $B$ дедуктивная цепочка, ведущая от $A$ к $B$ или нет? Решение этой проблемы понимается в смысле вопроса о существовании алгоритма, дающего ответ при любых $A$ и $B$. Чёрчем эта проблема была решена отрицательно.

**Теорема 5.4.1.** *Проблема распознавания выводимости алгоритмически неразрешима, то есть не существует машины Тьюринга, решающей эту проблему.*

***Проблема распознавания самоприменимости и применимости***

Программу машины Тьюринга можно закодировать каким-либо определённым шифром. На ленте машины можно изобразить её же

собственный шифр, записанный в алфавите машины. Здесь, как и в случае обычной программы, возможны два случая:

1) машина применима к своему шифру, т. е. она перерабатывает этот шифр и после конечного числа тактов останавливается (в этом случае говорят, что машина относится к классу *самоприменимых* Тьюринговых машин);
2) машина неприменима к своему шифру, т. е. машина никогда не переходит в заключительное состояние (в этом случае говорят, что машина относится к классу *несамоприменимых* Тьюринговых машин).

Суть *проблемы самоприменимости* состоит в следующем: по любому заданному шифру требуется установить, к какому классу относится машина, зашифрованная им, к классу самоприменимых или несамоприменимых машин.

**Теорема 5.4.2.** *Проблема распознавания самоприменимости алгоритмически неразрешима.*

*Доказательство*. Предположим обратное. Пусть существует машина $M$, которая решает проблему самоприменимости. Тогда в $M$ всякий самоприменимый шифр перерабатывается в какой-то символ $\sigma$, имеющий смысл утвердительного ответа на поставленный вопрос о самоприменимости, а всякий несамоприменимый шифр перерабатывается в другой символ $\tau$, имеющий смысл отрицательного ответа на поставленный вопрос. В таком случае можно построить и машину $M'$, которая по-прежнему перерабатывает несамоприменимые шифры в $\tau$, в то время как к самоприменимым шифрам машина $M'$ уже не применима. Этого можно добиться путём такого изменения машины $M$, что после появления символа $\sigma$ вместо перевода в заключительное состояние, машина начнёт неограниченно повторять этот же символ. Таким образом, $M'$ применима ко всякому несамоприменимому шифру и не применима к самоприменимым шифрам. Это, в свою очередь, приводит к следующим противоречиям:

1) пусть машина $M'$ самоприменима, тогда она применима к своему шифру $M'$ и перерабатывает его в символ $\tau$, но появление этого символа как раз и должно означать то, что $M'$ несамоприменима;
2) пусть $M'$ несамоприменима, тогда она не применима к своему шифру $M'$, что должно означать как раз то, что $M'$ самоприменима.

Полученные противоречия доказывают данную теорему. □

Из Теоремы 5.4.2 немедленно вытекает и алгоритмическая неразрешимость более общей *проблемы применимости*, которая сводится к тому,

чтобы узнать, применима ли машина к любой заданной конфигурации. Напомним, что машина называется применимой, если после конечного числа шагов она останавливается.

**Следствие 5.4.1.** *Проблема распознавания применимости алгоритмически неразрешима.*

***Проблема распознавания переводимости***

*Проблема распознавания переводимости* заключается в следующем. Для любой заданной машины Тьюринга и любых двух конфигураций в ней $K$ и $K'$ требуется выяснить: если в качестве начальной конфигурации взята $K$, то перейдёт ли машина (после конечного числа шагов) в конфигурацию $K'$ или нет?

Для доказательства неразрешимости проблемы распознавания переводимости применим нижеописанный метод.

Пусть для каждой единичной задачи $a_i$, входящей в серию задач $\{a_i\}$, указана единичная задача $f(a_i)$, входящая в серию задач $\{b_i\}$ и такая, что если известно решение задачи $f(a_i)$, то из него следует и решение задачи $a_i$. В таком случае говорят, что проблема $\{a_i\}$ *сведена* к проблеме $\{b_i\}$. Ясно также, что если имеется алгоритм для решения задач $\{b_i\}$, то его можно перестроить в алгоритм, решающий серию задач $\{a_i\}$. Если же ранее было установлено, что серия задач $\{a_i\}$ алгоритмически неразрешима, то отсюда вытекает и алгоритмическая неразрешимость серии $\{b_i\}$. Такая ситуация используется для установления алгоритмической неразрешимости исследуемой проблемы. Для исследуемой проблемы $G$ отыскивают проблему, которая заведомо алгоритмически неразрешима и вместе с тем сводится к проблеме $G$.

Приведём пример сводимости проблем из теории машин Тьюринга.

Для произвольной машины Тьюринга $M$ можно построить другую машину $M^*$, которая будет связана с ней, как указано ниже. Пусть $K$ – произвольная конфигурация в $M$, тогда и в $M^*$ ей сопоставлена конфигурация $K^*$ с соблюдением двух условий:

1) если $M$ неприменима к $K$, то и $M^*$ не применима к $K^*$;
2) если $M$ применима к $K$, то $M^*$ тоже применима к $K^*$.

Пользуясь введёнными выше терминами, можно сказать, что проблема применимости сводится к проблеме переводимости. Действительно, для распознавания применимости машины $M$ к конфигурации $K$ достаточно выяснить, переводима ли в машине $M^*$ конфигурация $K^*$ в конфигурацию, в которой $M^*$ окажется в заключительном состоянии. Отсюда вытекает, что проблема переводимости тоже алгоритмически неразрешима.

**Теорема 5.4.3.** *Проблема переводимости алгоритмически неразрешима.*

***Проблема эквивалентности слов для ассоциативных исчислений***

Вышеуказанные результаты об алгоритмической неразрешимости установлены для проблем, касающихся математической логики и теории алгоритмов. Но также выяснилось, что аналогичные проблемы возникают и в других разделах математики. Сюда относятся, в первую очередь, алгебраические проблемы, приводящие к различным вариантам проблемы слов.

Рассмотрим некоторый конечный алфавит $A = \{a, b, c\}$ и множество слов в этом алфавите. Будем рассматривать преобразования одних слов в другие с помощью некоторых подстановок $\alpha \to \beta$, где $\alpha$ и $\beta$ – слова в алфавите $A$. Если слово $P$ содержит $\alpha$ как своё подслово, например $bc\alpha b\alpha$, то возможны три следующие подстановки: $bc\beta b\alpha$, $bc\alpha b\beta$ или $bc\beta b\beta$. Заметим, что эти подстановки похожи на те, которые изучались в нормальных алгоритмах Маркова (разве что в НАМ подстановки однозначно применялись слева направо по отношению к входному слову).

*Ассоциативным исчислением* называется совокупность всех слов в некотором алфавите вместе с какой-нибудь конечной системой допустимых подстановок. Для задания ассоциативного исчисления достаточно задать соответствующий алфавит и систему подстановок.

Если слово $P$ может быть преобразовано в слово $R$ путем однократного применения определённой подстановки, то $P$ и $R$ называются *смежными* словами. Последовательность слов $P_1$, $P_2$, … , $P_{n-1}$, $P_n$ таких, что все пары слов $P_i$ и $P_{i+1}$ являются смежными для всякого $i \in \{1,2,\dots,n-1\}$, называется *дедуктивной цепочкой*, ведущей от слова $P_1$ к слову $P_n$. Если существует цепочка, ведущая от слова $P$ к слову $R$, то слова $P$ и $R$ называются *эквивалентными*.

Для каждого ассоциативного исчисления возникает своя специальная *проблема эквивалентности слов:* для любых двух слов в данном исчислении требуется узнать, эквивалентны они или нет.

**Теорема 5.4.4.** *Проблема эквивалентности слов в любом ассоциативном исчислении алгоритмически неразрешима.*

## 5.5 Задачи для самостоятельного решения

1. Доказать, что функции примитивно рекурсивны:
   a) $f(x,y) = |x-y| = \begin{cases} x-y, & x \geq y, \\ y-x, & x < y; \end{cases}$
   b) $f(x) = x!$;

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

d) $f(x, y) = x^{x^y}$;

e) $f(x, y) = rm(x, y) =$ остатку от деления $y$ на $x$;

f) $f(x, y) = qt(x, y) =$ частному от деления $y$ на $x$;

g) $f(x, y) = min\{x, y\}$;

h) $f(x_1, x_2, \dots, x_n) = max\{x_1, x_2, \dots, x_n\}$.

2. Составить машину Тьюринга.

a) $A = \{a, b, c\}$. Приписать справа к слову $P$ символы $bc$ (например: $P \longrightarrow Pbc$).

b) $A = \{a, b, c\}$. Заменить на $a$ каждый второй символ в слове $P$.

c) $A = \{a, b, c\}$. Оставить в слове $P$ только последний символ (пустое слово не менять).

d) $A = \{a, b, c\}$. Определить, входит ли в слово $P$ символ $a$. Ответ: слово из одного символа $a$ (да, входит) или пустое слово (нет).

e) $A = \{a, b, c\}$. Если в слово $P$ не входит символ $a$, то заменить в $P$ все символы $b$ на $c$, иначе в качестве ответа выдать слово из одного символа $a$.

f) $A = \{a, b, c\}$. Если $P$ – слово чётной длины (0, 2, 4, …), то выдать ответ $a$, иначе – пустое слово.

g) $A = \{a, b, c\}$. Пусть $P$ имеет нечётную длину. Оставить в $P$ только средний символ.

h) $A = \{a, b\}$. В непустом слове $P$ поменять местами его первый и последний символы.

i) $A = \{a, b\}$. Определить, является ли слово $P$ палиндромом (перевёртышем, симметричным словом) или нет. Ответ: $a$ (да) или пустое слово.

j) $A = \{a, b\}$. Удвоить каждый символ слова $P$ (например: $bab \longrightarrow bbabb$).

k) $A = \{0,1,2,3\}$. Считая непустое слово $P$ записью положительного числа в троичной системе счисления, уменьшить это число на 1.

l) $A = \{(,)\}$. Определить сбалансировано ли слово $P$ по круглым скобкам. Ответ: $D$ (Да) или $N$ (нет).

m) $A = \{0,1\}$. Пусть $P$ имеет вид $Q > R$, где $Q$ и $R$ – двоичные числа (возможно, с незначащими нулями), выдать в качестве ответа слово 1, если число $Q$ больше числа $R$, и слово 0 иначе.

3. Составить нормальный алгоритм Маркова.
   a) $A = \{a, b, c\}$. Заменить любое входное слово на слово $a$.
   b) $A = \{\,|\,\}$. Считая слово $P$ записью числа в единичной системе счисления, получить остаток от деления этого числа на 2, т. е. получить слово из одной палочки, если число нечётно, или пустое слово, если число чётно.

      Под единичной системой счисления понимается запись неотрицательного целого числа с помощью палочек – должно быть записано столько палочек, какова величина числа (например: $2 \to ||$ , $7 \to |||||||$, $0 \to$ ).
   c) $A = \{a, b, c\}$. Определить, входит ли символ $a$ в слово $P$. Ответ (выходное слово): слово $a$, если входит, или пустое слово, если не входит.
   d) $A = \{a, b, c\}$. Удалить из слова $P$ третье вхождение символа $a$, если такое есть.
   e) $A = \{a, b\}$. Если в слово $P$ входит больше символов $a$, чем символов $b$, то в качестве ответа выдать слово из одного символа $a$, если в $P$ равное количество $a$ и $b$, то в качестве ответа выдать пустое слово, а иначе выдать ответ $b$.
   f) $A = \{a, b, c\}$. Определить, из скольких различных символов составлено слово $P$. Ответ получить в единичной системе счисления (например: $bcbbc \to ||$).
   g) $A = \{a, b, c\}$. Если буквы в непустом слове $P$ не упорядочены по алфавиту, то заменить $P$ на пустое слово, а иначе $P$ не менять.
   h) $A = \{a, b\}$. Приписать справа к слову $P$ столько палочек, сколько всего символов входит в $P$ (например: $babab \to babab|||||$).
   i) $A = \{a, b\}$. Пусть слово $P$ имеет чётную длину (0, 2, 4, …). Удалить правую половину этого слова.
   j) $A = \{a, b\}$. Если в непустом слове $P$ совпадают первый и последний символы, то удалить оба этих символа, а иначе слово не менять.
   k) $A = \{\,|\,\}$. Считая слово $P$ записью числа $n$ в единичной системе, получить в этой же системе число $2^n$.
   l) $A = \{a, b\}$. Перевернуть слово $P$ (например: $abbb \to bbba$).

# Литература


1. Унучек С. А. *Математическая логика: Учебное пособие.* Саратов: Ай Пи Эр Медиа, 2018, 239 с.
2. Зюзьков В. М. *Введение в математическую логику: Учебное пособие.* Томск: ТГУ, 2017, 258 с.
3. Макоха А. Н., Шапошников А. В., Бережной В. В. *Математическая логика и теория алгоритмов.* Ставрополь: СКФУ, 2017, 418 с.
4. Пильщиков В. Н., Абрамов В. Г., Вылиток А. А., Горячая И. В. *Машина Тьюринга и алгоритмы Маркова*. Москва, МГУ, 2016, 72 с.
5. Гуц А. К. *Математическая логика и теория алгоритмов*. Москва: Либроком, 2014, 120 с.
6. Агарева О. Ю. *Математическая логика и теория алгоритмов: Учебное пособие*. Москва: МАТИ, 2011, 79 с.
7. Герасимов А. С. *Курс математической логики и теории вычислимости: Учебное пособие*. СПб.: ЛЕМА, 2011, 284 с.
8. Назиев А. Х., Моисеев С. А. *Математическая логика: Задачник-практикум.* Рязань: РГУ им. С. А. Есенина, 2011, 80 с.
9. Гаврилов Г. П., Сапоженко А. А. *Задачи и упражнения по дискретной математике.* Москва: Физматлит, 2009, 416 с.
10. Новиков Ф. А. *Дискретная математика для программистов: Учебник для вузов.* СПб.: Питер, 2009, 384 с.
11. Замятин А. П. *Математическая логика и теория алгоритмов: Учебное пособие.* Екатеринбург: УрГУ, 2008, 273 с.
12. Тишин В. В. *Дискретная математика в примерах и задачах*. СПб.: БХВ-Петербург, 2008, 352 с.
13. Игошин В. И. *Математическая логика и теория алгоритмов*. Москва: Академия, 2008, 448 с.
14. Игошин В. И. *Задачи и упражнения по математической логике и теории алгоритмов.* Москва: Академия, 2007, 304 с.
15. Лупанов О. Б. *Введение в математическую логику*. Москва: МГУ, 2007, 192 с.

16. Шапорев С. Д. *Математическая логика. Курс лекций и практических занятий.* СПб.: БХВ-Петербург, 2005, 416 с.
17. Галиев Ш. И. *Математическая логика и теория алгоритмов*. Казань: КГТУ им. А. Н. Туполева, 2002, 270 с.
18. Судоплатов С. В., Овчинникова Е. В. *Элементы дискретной математики. Учебник.* Москва: Инфра-М, Новосибирск, Изд. НГТУ., 2002, 280 с.
19. Фролов И. С. *Элементы математической логики: Учебное пособие для студентов математических специальностей*. Самара: Самарский университет, 2001, 80 с.
20. Москинова Г. И. *Дискретная математика. Математика для менеджера в примерах и упражнениях: Учебное пособие*. Москва: Логос, 2000, 240 с.
21. Лихтарников Л. М., Сукачева Т. Г. *Математическая логика. Курс лекций. Задачник-практикум и решения.* Серия “Учебники для вузов. Специальная литература”, СПб.: Лань, 1999, 288 с.
22. Никольская И. Л. *Математическая логика*. Москва: Высшая школа, 1981, 127 с.
23. Клини С. *Математическая логика*. Москва: Мир, 1973, 480 с.
24. Новиков П. С. *Элементы математической логики*. Москва: Наука, 1973, 400 с.
25. Мендельсон Э. *Введение в математическую логику*. Москва: Наука, 1971, 320 с.
26. Трахтенброт Б. А. *Алгоритмы и машинное решение задач*. Москва: Физматлит, 1960, 119 с.

Бузату Раду Валерьевич

# Математическая логика и теория алгоритмов

Учебное пособие (курс лекций)

Buzatu Radu Valeriu

# Logica matematică şi teoria algoritmilor

(în limba rusă)

Note de curs

Redactare: *Valentina Mladina*
Machetare computerizată: *Radu Buzatu*

Bun de tipar 27.12.2021. Formatul 70 × 100 1/12.
Coli de tipar 8,4. Coli editoriale 5,0.
Comanda 98. Tirajul 50 ex.

Centrul Editorial-Poligrafic al USM,
str. Al. Mateevici, 60, Chişinău, MD-2009