Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«МАТИ — Российский государственный технологический университет
имени К. Э. Циолковского»

О. Ю. Агарева, Ю. В. Селиванов

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Учебное пособие

*Рекомендовано УМС МАТИ в качестве учебного пособия
для студентов и аспирантов МАТИ всех форм обучения,
изучающих дисциплины: «Математическая логика и теория алгоритмов»
и «Дискретная математика»*

Москва 2011

УДК 510
ББК 22.1
А23

Авторы:

Агарева О. Ю., к.ф.-м.н., профессор кафедры «Высшая математика» МАТИ имени К. Э. Циолковского

Селиванов Ю. В., д.ф.-м.н., профессор кафедры «Высшая математика» МАТИ имени К. Э. Циолковского

Рецензенты:

Лукацкий А. М., д.ф.-м.н., ведущий научный сотрудник ИНЭИ РАН

Щетинин А. Н., к.ф.-м.н., доцент кафедры вычислительной математики и математической физики МГТУ имени Н. Э. Баумана

Агарева, О. Ю.

А23 Математическая логика и теория алгоритмов [Текст] : учеб. пособие / О. Ю. Агарева, Ю. В. Селиванов. — М. : МАТИ, 2011. — 80 с.

ISBN 978-5-93271-611-3

Учебное пособие предназначено для студентов МАТИ, изучающих дисциплины «Математическая логика и теория алгоритмов» и «Дискретная математика», обучающихся по специальностям «Информатика и вычислительная техника» и «Системы автоматизированного проектирования». Оно ставит своей целью помочь студентам лучше усвоить теоретический и практический материал. Пособие посвящено изучению важных разделов математической логики (алгебры высказываний, логики предикатов) и теории алгоритмов. Его основу составляют конспекты лекций, которые читались студентам. Данное пособие содержит большое количество примеров, иллюстрирующих основные понятия указанных разделов математической логики и теории алгоритмов и утверждения, касающиеся этих понятий.

Издание также может быть полезно для студентов других специальностей и преподавателей.

УДК 510
ББК 22.1

ISBN 978-5-93271-611-3

Введение

Математическая логика является важнейшим элементом математического образования. Данное пособие содержит изложение основ математической логики и теории алгоритмов, соответствующее первой части курса лекций по дисциплине «Математическая логика и теория алгоритмов», читаемых в МАТИ студентам, обучающимся по специальностям «Информатика и вычислительная техника» и «Системы автоматизированного проектирования». Пособие предназначено в помощь как студентам МАТИ, так и студентам других технических университетов. Оно может оказаться полезным студентам и преподавателям также и при изучении дисциплины «Дискретная математика».

В пособии рассматриваются следующие темы: алгебра (логика) высказываний, логика предикатов, теория алгоритмов. Оно содержит большое количество примеров, демонстрирующих использование изложенной теории для решения конкретных задач.

Формальная логика — одна из древнейших наук. Отдельные фрагменты ее начали разрабатываться в VI в. до нашей эры в Древней Греции и Индии. Основателем этой науки считается гениальный древнегреческий ученый Аристотель, который обстоятельно систематизировал логические формы и правила мышления, и тем самым заложил начала логики. Логика, основанная на учении Аристотеля, существовала до начала XX в., после чего в ней произошла своеобразная научная революция, связанная с широким применением методов так называемой символической, или математической логики. Идеи последней — о возможности сведения рассуждений к вычислениям — были высказаны еще немецким ученым Г. В. Лейбницем в XVII в. Однако только к началу XX столетия математическая логика (то есть логика, развиваемая математическим методом) оформилась в качестве самостоятельной дисциплины. Характерным для этой дисциплины является использование формальных языков с точным синтаксисом и четкой семантикой, однозначно определяющими понимание формул. Современная математическая логика — это та же самая логика Аристотеля, но только громоздкие словесные выводы заменены в ней математической символикой. Эта дисциплина изучает вопросы применения математических методов для решения логических задач и построения логических схем, которые лежат в основе работы любого компьютера.

Для построения основных разделов современной математической логики существуют два подхода (языка), образующих два варианта формальной логики: *алгебру логики* и *логические исчисления*. В данном пособии рассматривается только первый из этих подходов.

Суждения в математической логике называют высказываниями или логическими выражениями. Математическая логика включает два основных раздела: *логику высказываний* (или *пропозициональную логику*) и охватывающую ее *логику предикатов*.

1. АЛГЕБРА (ЛОГИКА) ВЫСКАЗЫВАНИЙ

Основными понятиями логики высказываний являются высказывания и логические связки (операции над высказываниями). В логике предикатов используются еще предикаты и кванторы.

1.1. Высказывания и операции над ними

Логическими *высказываниями* являются утвердительные предложения, о которых можно судить, истинны они или ложны. Причем они не могут быть истинными и ложными одновременно. Логика высказываний рассматривает эти предложения не с точки зрения их смысла, содержания, а только с точки зрения их истинности или ложности. Для понятия «высказывание» иногда используют термин «пропозиция», а говоря «пропозициональный», подразумевают относящийся к логике высказываний.

Классический пример утверждения, не являющегося высказыванием, таков:

Все, что написано в этой рамке, есть ложь.

Действительно, попытка определить истинностное значение этого «высказывания» приводит к противоречию: если то, что написано, истинно, то это противоречит смыслу слов в рамке. То же противоречие возникает, если предположить, что оно ложно.

Вопросительные, повелительные и бессмысленные предложения не являются логическими высказываниями.

Говорят, что если предложение истинно, то его значение истинности равно 1, если ложно — то 0. По аналогии с элементарной алгеброй, где любое число является константой, высказывание является логической константой, величина которой равна 1 или 0.

В качестве примера отметим, что предложение «$x^2 = 4$», вообще говоря, не является высказыванием. Для того чтобы имело смысл говорить об его истинности или ложности, необходимы некоторые дополнительные сведения. Конечно, достаточно знать, какое именно число обозначено буквой x.

Каждому значению переменной x будет соответствовать либо истинное, либо ложное высказывание; например, высказывания $(-2)^2 = 4$, $2^2 = 4$ истинны, остальные ложны.

Будем называть высказывание *простым (элементарным, атомарным),* если оно рассматривается нами как некое неделимое целое. Обычно к ним относят высказывания, не содержащие логических связок.

Сложным (составным) называется высказывание, составленное из простых с помощью логических связок.

В логике над высказываниями производятся следующие основные операции (логические связки): отрицание, конъюнкция, дизъюнкция, импликация, эквиваленция, неравнозначность. Они рассматриваются как средство вычисления логического значения сложного высказывания по логическим значениям составляющих его простых высказываний.

Отрицание (логическая связка «не»)

Отрицанием (инверсией) высказывания A называется высказывание, которое истинно, если высказывание A ложно, и ложно, когда A истинно. Записывается: $\overline{A}$ или $\neg A$. Читается: «не A» («не верно, что A»). Отметим, что отрицание является логической операцией, выполняемой над одним аргументом.

Эта логическая связка может быть проиллюстрирована следующей таблицей (*таблицей истинности*):

A	$\neg A$
0	1
1	0

Логическое умножение (конъюнкция)

Конъюнкция двух высказываний A и B — это сложное логическое высказывание, которое истинно только в случае истинности всех составляющих высказываний, в противном случае оно ложно. Обозначения: $A \& B$, $A \wedge B$. Читается: «A и B».

Эта логическая связка может быть также проиллюстрирована таблицей истинности, в которой показаны значения истинности сложного высказывания в зависимости от значений истинности составляющих его простых высказываний A и B.

A	B	$A \& B$
0	0	0
0	1	0
1	0	0
1	1	1

Логическое сложение (дизъюнкция)

Дизъюнкция двух высказываний A и B — это сложное логическое высказывание, которое ложно только в случае ложности всех составляющих высказываний, в противном случае оно истинно. Таким образом, это высказывание считается истинным, когда истинно хотя бы одно из составляющих высказываний. Обозначается: $A \vee B$. Иногда встречается обозначение $A + B$. Читается: «A или B».

Дизъюнкция иллюстрируется следующей таблицей истинности:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Логическое следование (импликация)

В математических доказательствах часто пользуются сложными высказываниями, образованными с помощью слов «если…, то…». Здесь высказывание, расположенное после слова «если», называется основанием или посылкой, а высказывание, расположенное после слова «то», называется следствием или заключением. *Импликацией* двух высказываний A и B называется высказывание, обозначаемое символом $A \to B$, которое ложно тогда и только тогда, когда A истинно, а B ложно. Иногда встречается обозначение $A \supset B$. Читается: «если A, то B» («A влечет B», «из A следует B»).

Импликация проиллюстрирована таблицей истинности:

A	B	$A \to B$
0	0	1
0	1	1
1	0	0
1	1	1

Определение импликации вынуждает считать истинными такие предложения, как: «если $2 \times 2 = 4$, то Москва столица России»; «если $2 \times 2 = 5$, то $3 \times 3 = 6$». Это связано с тем, что определениями логических операций смысл составляющих высказываний не учитывается, они рассматриваются как объекты, обладающие единственным свойством — быть истинными, либо ложными.

Истинность высказывания «если $2 \times 2 = 5$, то $3 \times 3 = 6$» кажется парадоксальной. Но объяснение этому, во-первых, следует искать в том, что сами высказывания $2 \times 2 = 5$ и $3 \times 3 = 6$ мало связаны между собой, а во-вторых, в том, что использование сослагательного наклонения несколько точнее отражало бы смысл указанной импликации. В самом деле, утверждение «если бы $2 \times 2 = 5$, то $3 \times 3 = 6$» не кажется противоречивым, то есть истинность нашей импликации означает, что «$3 \times 3 = 6$ не менее истинно, чем $2 \times 2 = 5$».

Логическое тождество (эквиваленция)

Эквиваленцией (эквивалентностью, равнозначностью) двух высказываний A и B называется высказывание, обозначаемое символом $A \sim B$ (или $A \leftrightarrow B$), которое истинно когда истинностные значения высказываний A и B совпадают, и ложно — в противном случае.

Таблица истинности для эквивалентности имеет вид:

A	B	$A \sim B$
0	0	1
0	1	0
1	0	0
1	1	1

Логическая операция $A \sim B$ соответствует союзу «тогда и только тогда, когда» и читается: «A эквивалентно B» («A равнозначно B», «для того, чтобы A необходимо и достаточно, чтобы B»).

Когда мы говорим «A тогда и только тогда, когда B», то имеем в виду, что оба предложения A и B одновременно истинны, либо одновременно ложны. Например, говоря: «Я поеду в Ленинград тогда и только тогда, когда ты поедешь в Киев», мы утверждаем, что: либо произойдет и то и другое, либо ни того, ни другого.

Исключающее «или» (неравнозначность)

Неравнозначностью двух высказываний A и B называется высказывание, истинное, когда истинностные значения A и B не совпадают, и ложное — в противном случае. Обозначается: $A \oplus B$. Читается: «либо A, либо B» (понимается — в разделительном смысле). Таблица истинности для неравнозначности имеет вид:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Итак, в математической логике для записи сложных высказываний используются следующие логические операции над простыми высказываниями:

$\neg$ — не;

$\&, \wedge$ — и;

$\vee$ — или;

$\rightarrow$ — влечет;

$\sim$ — эквивалентно;

$\oplus$ — либо, либо.

Каждую из этих операций можно рассматривать как операцию над символами 0 и 1.

1.2. Формулы алгебры высказываний

Пропозициональными (*высказывательными*) *переменными* называются такие переменные, вместо которых можно подставлять конкретные высказывания. Пропозициональные переменные (то есть буквы, обозначающие высказывания), логические связки и скобки составляют *алфавит* языка алгебры высказываний. С помощью элементов алфавита можно построить разнообразные *логические формулы*. Под формулами алгебры высказываний понимаются осмысленные выражения, полученные из символов пропозициональных переменных, знаков операций и скобок, определяющих порядок действий. Дадим более четкое определение логической формулы.

Логическая *формула* определяется индуктивно по следующей схеме:

1) Всякая пропозициональная переменная есть формула.

2) Если A — формула, то и $\neg A$ является формулой.

3) Если A и B — формулы, то выражения $(A \& B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \sim B)$, $(A \oplus B)$ также являются формулами.

4) Других формул, кроме построенных по правилам трех предыдущих пунктов, нет.

Определение формулы таково, что формулы насыщены скобками и трудночитаемы, поэтому обычно принимают соглашение об упрощении записи формул:

1) Наружные скобки в записи формул можно опускать.

2) Считается, что конъюнкция «сильнее» дизъюнкции, а обе они «сильнее» неравнозначности, импликации и эквиваленции. Отрицание «сильнее» всех других операций. Поэтому часть скобок, определяющих порядок действий, можно опускать.

3) Скобки, определяющие порядок действий, в ассоциативном случае можно опускать.

4) Конъюнкцию можно обозначать знаком «·» или знак конъюнкции опускать.

Иногда опускают не все скобки, которые можно опустить, чтобы формула легче воспринималась. В первую очередь выполняются операции в скобках, затем все остальные логические операции в порядке старшинства. Порядок старшинства логических операций следующий:

$$\neg,\ \&,\ \vee,\ \oplus,\ \rightarrow,\ \sim.$$

Примеры

1. Представить логическими формулами следующие высказывания:

а) «Сегодня суббота или воскресенье».

Решение. Пусть A — «сегодня суббота», а B — «сегодня воскресенье». Тогда «сегодня суббота или воскресенье» представимо формулой: $A \oplus B$. (Это сложное высказывание состоит из двух простых высказываний A и B, соединенных связкой «или» в разделительном смысле.)

б) «Идет снег или дождь».

Решение. Пусть A — «идет снег», а B — «идет дождь». Тогда логическая формула для высказывания «идет снег или дождь» имеет вид: $A \vee B$.

в) «Если идет дождь, то крыши мокрые».

Решение. Пусть A — «идет дождь», а B — «крыши мокрые». Тогда «если идет дождь, то крыши мокрые» представимо формулой: $A \rightarrow B$.

г) «Что в лоб, что по лбу».

Решение. Пусть A — «в лоб», а B — «по лбу». Тогда «что в лоб, что по лбу» может иметь вид: $A \sim B$.

д) «В квартире грязно и холодно».

Решение. Пусть A — «в квартире грязно», а B — «в квартире холодно». Тогда «в квартире грязно и холодно» представимо логической формулой: $A \& B$.

е) «Если допоздна работаешь с компьютером и при этом пьешь много кофе, то утром просыпаешься в дурном настроении или с головной болью».

Решение. Пусть:

A — «допоздна работаешь с компьютером»,
B — «пьешь много кофе»,
C — «утром просыпаешься в дурном настроении»,
E — «утром просыпаешься с головной болью».

Тогда сложное высказывание «если допоздна работаешь с компьютером и при этом пьешь много кофе, то утром просыпаешься в дурном настроении или с головной болью» представимо формулой: $(A \& B) \rightarrow (C \vee E)$.

2. Пусть даны высказывания:

A — «число 9 делится на 3»,
B — «число 10 делится на 3».

Требуется определить значения истинности следующих высказываний:

1) $B \to A$;   2) $\neg A \to B$;   3) $\neg B \to \neg A$.

Решение. Имеем: A истинно, $\neg A$ ложно, B ложно, $\neg B$ истинно. Следовательно, из определения импликации получаем:

1) $0 \to 1$ истинно;
2) $0 \to 0$ истинно;
3) $1 \to 0$ ложно.

1.3. Логические функции высказываний

Теория, которая изучает логические формулы, определенные выше, называется алгеброй высказываний. Она изучает строение сложных логических высказываний и способы установления их истинности с помощью алгебраических методов. В алгебре высказываний каждая пропозициональная переменная, каждая формула принимает одно из двух значений: 1 (истина) или 0 (ложь). Таким образом, каждая логическая формула задает *логическую функцию* — функцию от логических переменных, которая сама может принимать только два логических значения.

Любую логическую функцию $f\left(x_1, \ldots, x_n\right)$ можно задать *таблицей истинности*, в левой части которой перечислены все возможные наборы значений ее аргументов (то есть двоичных векторов длины n), а в правой части — значения заданной функции на этих наборах. Будем обозначать через P_2 множество всех логических функций (для всех возможных значений n числа переменных), а через $P_2(n)$ — множество всех логических функций от n переменных (для фиксированного n).

В таблице истинности наборы значений аргументов расположены в определенном порядке — *лексикографическом*, который совпадает с порядком возрастания наборов, рассматриваемых как двоичные числа.

Число всех возможных двоичных векторов длины n равно 2^n. Поэтому число всех различных логических функций от n переменных равно числу возможных расстановок нулей и единиц в столбце длины 2^n, то есть это число равно 2^{2^n}.

Полезно иметь в виду, что, записывая таблицу истинности логической функции от n переменных, нет необходимости каждый раз перечислять все наборы длины n — достаточно записать *вектор значений* логической функции, понимая, что i-я компонента этого вектора есть значение функции на i-м наборе (двоичном коде числа i).

Можно также перечислить номера тех наборов, на которых функция принимает значение 1. Наборы значений аргументов x_1, x_2, ..., x_n, на которых $f(x_1, \ldots, x_n) = 1$, часто называют *единичными наборами* функции f, а множество всех единичных наборов — *единичным множеством* функции f. Подобным образом, наборы значений, на которых $f = 0$, называют *нулевыми наборами* функции f, а множество нулевых наборов — *нулевым множеством.*

Известно, что всякую логическую функцию можно задать и с помощью формулы. Особую роль в алгебре высказываний играют логические функции одной и двух переменных — *унарные* и *бинарные* логические операции. Среди них: отрицание, конъюнкция, дизъюнкция, импликация, эквиваленция, неравнозначность (сумма по модулю 2) и др. Эти функции очевидным образом интерпретируются естественными логическими связками «не», «и», «или» и т. д., широко используемыми при описании систем, явлений, формализации рассуждений и пр.

Таким образом, формула наряду с таблицей истинности служит способом задания и вычисления функции. В общем случае формула описывает логическую функцию как суперпозицию других более простых функций. Имея функции $f\left(x_1,\ \ldots,\ x_m\right)$, $g_k\left(x_1,\ \ldots,\ x_n\right)$, где $k = 1,\ \ldots,\ m$, образуем функцию

$$h\left(x_1,\ \ldots,\ x_n\right) = f\left(g_1\left(x_1,\ \ldots,\ x_n\right),\ \ldots,\ g_m\left(x_1,\ \ldots,\ x_n\right)\right).$$

Эту операцию и называют *суперпозицией*.

Пример

Требуется построить таблицу истинности для формулы:

$$F = \left(\left(A \wedge \left(B \to \overline{C}\right)\right) \wedge \left(\overline{B} \to A\right) \vee B\right).$$

Решение. Определим порядок действий и заполним последовательно таблицу истинности. Имеем:

$$F = \left(\underbrace{\underbrace{\left(\underbrace{A \wedge \left(\underbrace{B \to \underbrace{\overline{C}}_{1}}_{2} \right)}_{3} \right) \wedge \left(\underbrace{\underbrace{\overline{B}}_{4} \to A}_{5} \right)}_{6} \vee B}_{7} \right),$$

A	B	C	1	2	3	4	5	6	$7=F$
0	0	0	1	1	0	1	0	0	0
0	0	1	0	1	0	1	0	0	0
0	1	0	1	1	0	0	1	0	1
0	1	1	0	0	0	0	1	0	1
1	0	0	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	1	0	0	0	0	1	0	1

1.4. Равносильность формул

Назовем *эквивалентными* (или *равносильными*) формулы, которые представляют равные функции. Равносильность формул в алгебре логики обозначается знаком тождественного равенства $\equiv$. Стандартный метод установления равносильности двух формул:

1) по каждой формуле восстанавливается таблица истинности;

2) полученные таблицы сравниваются по каждому набору значений переменных.

Нужно различать символы $\sim$ и $\equiv$. Знак $\sim$ является символом формального языка, с помощью которого строятся формулы, а знак $\equiv$ обозначает отношение на множестве формул.

В логике выделяют следующие равносильные формулы — основные *эквивалентные соотношения* (*законы*):

1. $A \equiv A$ (*закон тождества*);

2. $A \& 0 \equiv 0$;

3. $A \vee 0 \equiv A$;

4. $A \& 1 \equiv A$;

5. $A \vee 1 \equiv 1$;

6. $\neg(\neg A) \equiv A$ (*закон двойного отрицания*);

7. $A \,\&\, (\neg A) \equiv 0$ (*закон логического противоречия*);

8. $A \vee (\neg A) \equiv 1$ (*закон исключенного третьего*);

9. $A \,\&\, A \equiv A$ (*идемпотентность конъюнкции*);

10. $A \vee A \equiv A$ (*идемпотентность дизъюнкции*);

11. $A \,\&\, B \equiv B \,\&\, A$ (*коммутативность конъюнкции*);

12. $A \vee B \equiv B \vee A$ (*коммутативность дизъюнкции*);

13. $A \,\&\, (B \,\&\, C) \equiv (A \,\&\, B) \,\&\, C$ (*ассоциативность конъюнкции*);

14. $A \vee (B \vee C) \equiv (A \vee B) \vee C$ (*ассоциативность дизъюнкции*);

15. $A \,\&\, (B \vee C) \equiv (A \,\&\, B) \vee (A \,\&\, C)$ (*дистрибутивность конъюнкции относительно дизъюнкции*);

16. $A \vee (B \,\&\, C) \equiv (A \vee B) \,\&\, (A \vee C)$ (*дистрибутивность дизъюнкции относительно конъюнкции*);

17. $A \,\&\, (A \vee B) \equiv A$ (*первый закон поглощения*);

18. $A \vee (A \,\&\, B) \equiv A$ (*второй закон поглощения*);

19. $\neg(A \,\&\, B) \equiv \neg A \vee \neg B$ (*первый закон де Моргана*);

20. $\neg(A \vee B) \equiv \neg A \,\&\, \neg B$ (*второй закон де Моргана*);

21. $A \equiv (A \& B) \vee (A \& \neg B)$ (*первый закон расщепления*);

22. $A \equiv (A \vee B) \& (A \vee \neg B)$ (*второй закон расщепления*);

23. $A \to B \equiv \neg B \to \neg A$ (*закон контрапозиции*);

24. $A \to B \equiv \neg A \vee B \equiv \neg (A \& \neg B)$;

25. $A \sim B \equiv (\neg A \vee B) \& (\neg B \vee A) \equiv (A \& B) \vee (\neg A \& \neg B)$;

26. $A \oplus B \equiv (A \& \neg B) \vee (\neg A \& B)$;

27. $A \vee B \equiv \neg A \to B \equiv \neg (\neg A \& \neg B)$;

28. $A \& B \equiv \neg (A \to \neg B) \equiv \neg (\neg A \vee \neg B)$.

Все эти равносильности легко доказываются с помощью таблиц истинности. Равносильности 24—28 показывают, что одни логические операции могут быть выражены через другие.

Эквивалентным (или *тождественным*) *преобразованием* некоторой формулы называют переход (по определенным правилам) к любой формуле, эквивалентной данной. Например, часто применяется *правило подстановки* формулы F вместо переменной A. При подстановке формулы F вместо переменной A все вхождения переменной A в исходное соотношение должны быть одновременно заменены формулой F. Это правило применяется к эквивалентным соотношениям для получения новых эквивалентных соотношений.

Правило замены подформул позволяет, используя известные эквивалентные соотношения, получать формулы, эквивалентные данной (в частности, упрощать формулы). Подформула — это часть формулы, сама являющаяся формулой. Если некоторая формула F содержит F_1 в качестве подформулы, то можно заменить F_1 на эквивалентную ей F_2. Полученная с помощью такой замены новая формула G эквивалентна исходной F.

Пример

Выписать все подформулы следующей формулы:

$$A \to \left(\overline{B} \to \overline{(A \to B)}\right).$$

Решение. Нужно помнить, что сама формула также является подформулой. Имеем:

$$A,\ \ B,\ \ \overline{B},\ \ A \to B,\ \ \overline{(A \to B)},\ \ \overline{B} \to \overline{(A \to B)},$$

$$A \to \left(\overline{B} \to \overline{(A \to B)}\right).$$

1.5. Полные системы логических функций

Как уже отмечалось, с каждой формулой алгебры высказываний связана функция, сопоставляющая каждому фиксированному набору высказывательных переменных значение истинности формулы $(\in\{0,1\})$. В то же время существуют наборы логических функций, с помощью которых можно выразить любые другие логические функции.

Система $S = \left\{ f_1, f_2, \ldots, f_k \right\}$ логических функций называется *полной* (или *функционально полной*), если любая логическая функция является суперпозицией функций из этой системы.

Пусть $S_1 = \{ \&, \vee, \neg \}$, где $\&\left(x, y\right) = x \,\&\, y$, $\vee\left(x, y\right) = x \vee y$ и $\neg\,(x) = \neg\, x$. Это множество, состоящее из функций конъюнкции, дизъюнкции и отрицания, часто называют *стандартным базисом*. Оно является полной системой логических функций, поскольку любая логическая функция высказываний может быть представлена некоторой формулой, использующей только операции конъюнкции, дизъюнкции и отрицания. Это следует из того, что, как отмечалось ранее, всякую логическую функцию можно задать с помощью формулы, а также из уже упоминавшихся выше законов:

$$A \to B \equiv \neg A \vee B,$$

$$A \sim B \equiv (A \,\&\, B) \vee (\neg A \,\&\, \neg B),$$

$$A \oplus B \equiv (A \,\&\, \neg B) \vee (\neg A \,\&\, B).$$

Пусть теперь $S_2 = \{\vee, \neg\}$. Это множество, состоящее из функций дизъюнкции и отрицания, также является полной системой логических функций. Для доказательства достаточно заметить, что операция конъюнкции выражается через отрицание и дизъюнкцию. В самом деле, напомним, что

$$A \,\&\, B \equiv \neg(\neg A \vee \neg B).$$

Аналогично, если $S_3 = \{\&, \neg\}$, то мы снова получаем полную систему, поскольку

$$A \vee B \equiv \neg(\neg A \,\&\, \neg B).$$

Таким образом, полная система функций S_1 является в некотором смысле «избыточной» — при удалении из нее функции $\&$ или $\vee$ полнота полученных систем S_2 и S_3 сохраняется. Однако система функций S_1 позволяет использовать для логических функций более простые формулы. В то же время каждая замена одной из функций $\&$ или $\vee$ из системы S_1 на остальные вносит в формулу много лишних отрицаний.

Заметим, что если из системы функций S_1 удалить отрицание, то получаемая в результате «усеченная» система $S_4 = \{\&, \vee\}$ неполна. Дело в том, что каждая из функций $\&$ и $\vee$ сохраняет нуль. (Логическая функция $f\left(x_1, \ldots, x_n\right)$ называется *сохраняющей нуль*, если $f\left(0, \ldots, 0\right) = 0$.) А суперпозиция логических функций, сохраняющих нуль, есть снова логическая функция, сохраняющая нуль. В то же время далеко не все логические функции сохраняют нуль (например, отрицание не обладает этим свойством), и потому далеко не все они выражаются через $\&$ и $\vee$.

Положим теперь $S_5 = \{\to, \neg\}$. Для доказательства полноты этой системы логических функций достаточно заметить, что функция $\vee$ из полной системы $S_2 = \{\vee, \neg\}$ представима через отрицание и импликацию:

$$A \vee B \equiv \neg A \to B.$$

Рассмотрим теперь множество логических функций

$$S_6 = \{\oplus, \&, 1\},$$

которое называют *базисом Жегалкина*. Будем вместо & писать знак умножения. Полнота системы S_6 следует из следующих тождеств

$$\neg A \equiv A \oplus 1,$$

$$A \vee B \equiv A \cdot B \oplus A \oplus B.$$

Здесь, как обычно, приоритет операции конъюнкции считается выше приоритета операции неравнозначности.

Поскольку операция неравнозначности ассоциативна, то при записи формул с этой операцией опускают скобки, определяющие порядок действий. Так, формулами над базисом Жегалкина будут:

$$X \oplus 1,$$

$$X \cdot Y \oplus X \oplus Y,$$

$$X \cdot Y \cdot Z \oplus X \cdot Y \oplus X \cdot Z \oplus Y \cdot Z \oplus X \oplus Y \oplus Z \oplus 1.$$

1.6. Тавтологии. Выполнимые формулы

Формулу, значения которой для любого набора переменных есть 1, будем называть *тождественно истинной формулой* (или *тавтологией*). Формулу, значения которой для любого набора переменных есть 0, будем называть *тождественно ложной формулой* (или *противоречием*).

Тавтологии играют в логике особо важную роль как формулы, отражающие логическую структуру предложений, истинных в силу одной только этой структуры. Для доказательства того, что формула является тавтологией достаточно построить таблицу истинности для нее. В этой таблице столбец под самой формулой должен состоять только из единиц.

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

Перечислим важные тавтологии (A, B, C — произвольные формулы):

1. $A \vee (\neg A)$ (*закон исключенного третьего*);
2. $\neg (A \,\&\, \neg A)$ (*закон отрицания противоречия*);
3. $\neg\neg A \sim A$ (*закон двойного отрицания*);
4. $A \to A$ (*закон тождества*);
5. $(A \to B) \sim (\neg B \to \neg A)$ (*закон контрапозиции*);
6. $((A \to B) \,\&\, (B \to C)) \to (A \to C)$ (*закон транзитивности импликации*);
7. $(A \sim B) \sim (\neg A \sim \neg B)$ (*закон противоположности*);

8. $((A \sim B)\,\&\,(B \sim C)) \to (A \sim C)$ (*закон транзитивности эквивалентности*);

9. $(A \to B) \sim (\neg A \lor B)$;

10. $(A \sim B) \sim ((A \to B)\,\&\,(B \to A))$;

11. $(A \oplus B) \sim ((A\,\&\,\neg B) \lor (\neg A\,\&\,B))$;

12. $(A \lor B) \sim (\neg A \to B)$;

13. $(A \to B) \lor (B \to A)$;

14. $(A\,\&\,(A \lor B)) \sim A$ (*первый закон поглощения*);

15. $(A \lor (A\,\&\,B)) \sim A$ (*второй закон поглощения*);

16. $\neg(A\,\&\,B) \sim (\neg A \lor \neg B)$ (*первый закон де Моргана*);

17. $\neg(A \lor B) \sim (\neg A\,\&\,\neg B)$ (*второй закон де Моргана*);

18. $(\neg A \to A) \to A$;

19. $(A\,\&\,B) \to A$; $(A\,\&\,B) \to B$;

20. $A \to (B \to (A\,\&\,B))$;

21. $A \to (B \to A)$;

22. $A \to (A \lor B)$; $B \to (A \lor B)$;

23. $((A \to B) \to A) \to A$ (*закон Пирса*);

24. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$.

Примеры

1. Выяснить, является ли следующая формула тождественно истинной:

$$F = \big((A \to B) \wedge \neg B\big) \to \neg A\big).$$

Решение. Построим таблицу истинности заданной формулы, используя определения логических операций:

A	B	$A \to B$	$\neg A$	$\neg B$	$(A \to B) \wedge \neg B$	F
0	0	1	1	1	1	1
0	1	1	1	0	0	1
1	0	0	0	1	0	1
1	1	1	0	0	0	1

Так как последний столбец состоит только из 1, то формула тождественно истинна.

2. Выяснить, является ли следующая формула выполнимой:

$$F = (\neg A \vee B) \to (A \wedge C).$$

Решение. Построим таблицу истинности заданной формулы, используя определения логических операций. Имеем:

A	B	C	$\neg A$	$\neg A \vee B$	$A \wedge C$	F
0	0	0	1	1	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	1
1	1	0	0	1	0	0
1	1	1	0	1	1	1

Поскольку на трех наборах (достаточно хотя бы на одном) функция принимает значение 1, то формула выполнима.

3. Выяснить, выполнима ли следующая формула:

$$F = \left(\left(B \to (A \wedge B)\right) \wedge \neg\left((A \wedge C) \to B\right)\right).$$

Решение. Определим порядок действий и заполним последовательно таблицу истинности заданной формулы. Имеем:

$$F = \underbrace{\left(\underbrace{\left(B \to \underbrace{(A \wedge B)}_{1}\right)}_{2} \wedge \underbrace{\neg\underbrace{\left(\underbrace{(A \wedge C)}_{3} \to B\right)}_{4}}_{5}\right)}_{6}.$$

Таблица истинности принимает вид:

A	B	C	1	2	3	4	5	$6=F$
0	0	0	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0
0	1	0	0	0	0	1	0	0
0	1	1	0	0	0	1	0	0
1	0	0	0	1	0	1	0	0
1	0	1	0	1	1	0	1	1
1	1	0	1	1	0	1	0	0
1	1	1	1	1	1	1	0	0

Так как справедливо равенство $F(1, 0, 1) \equiv 1$, то формула выполнима.

1.7. Нормальные формы для формул

Пусть $A_1, \ldots, A_n$ — формулы. Тогда, в силу ассоциативности операций & и $\vee$, как бы мы не расставляли скобки в выражениях $A_1 \& A_2 \& \ldots \& A_n$ и $A_1 \vee A_2 \vee \ldots \vee A_n$, всегда будем приходить к равносильным формулам. Поэтому каждое из этих выражений будем считать формулами (опуская скобки, определяющие порядок действий).

Формулу $A_1 \& A_2 \& \ldots \& A_n$ мы будем называть *конъюнкцией* формул $A_1, \ldots, A_n$, а формулу $A_1 \vee A_2 \vee \ldots \vee A_n$ — *дизъюнкцией* формул $A_1, \ldots, A_n$. Отметим, что имеют место *обобщенные законы де Моргана*:

$$\neg (A_1 \& A_2 \& \ldots \& A_n) \equiv \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n,$$

$$\neg (A_1 \vee A_2 \vee \ldots \vee A_n) \equiv \neg A_1 \& \neg A_2 \& \ldots \& \neg A_n.$$

Определим теперь некоторые канонические виды формул. Формула, которая есть пропозициональная переменная или отрицание переменной, называется *литералом*. Некоторая формула называется *элементарной конъюнкцией* (или *конъюнктом*), если она является конъюнкцией литералов, то есть конъюнкцией переменных и отрицаний переменных. Например, формулы

$$\neg X_1,\ X_2,\ X_1 \& \neg X_2,\ \neg X_1 \& X_2 \& X_1 \& X_3 \text{ и } \neg X_4 \& \neg X_2$$

являются элементарными конъюнкциями. Элементарная конъюнкция называется *совершенной*, если каждая из пропозициональных переменных входит в нее ровно один раз, со знаком отрицания или без него.

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

$$\neg X_1,\ X_2,\ X_1 \& \neg X_2,\ \neg X_1 \vee X_2,\ (\neg X_1 \& X_2) \vee (X_1 \& \neg X_2),$$

$$(\neg X_1 \& X_2 \& X_1 \& X_3) \vee X_1 \vee (\neg X_4 \& \neg X_2)$$

находятся в ДНФ.

Каждую логическую формулу можно привести эквивалентными преобразованиями к ДНФ (т. е. для любой формулы A можно найти такую формулу B, находящуюся в ДНФ, что $A \equiv B$). Для этого:

1) Сначала избавляются от операций импликации, эквивалентности и неравнозначности, выразив их через логические связки $\neg$, $\&$ и $\vee$ по законам:

$$A \to B \equiv \neg A \vee B,$$

$$A \sim B \equiv (A \& B) \vee (\neg A \& \neg B),$$

$$A \oplus B \equiv (A \& \neg B) \vee (\neg A \& B).$$

2) Доводят знаки отрицания до независимых переменных, используя законы де Моргана:

$$\neg (A \& B) \equiv \neg A \vee \neg B,$$

$$\neg (A \vee B) \equiv \neg A \& \neg B.$$

3) Применяя закон дистрибутивности

$$(A \vee B) \& C \equiv (A \& C) \vee (B \& C),$$

преобразуют формулу к дизъюнкции элементарных конъюнкций.

4) Постоянно избавляются от двойных отрицаний:

$$\neg\neg A \equiv A.$$

ДНФ A называется *совершенной* и обозначается *СДНФ*, если каждая переменная формулы A входит с отрицанием или без отрицания в каждый конъюнкт точно один раз. Например,

$$\neg X_1 \vee (X_2 \,\&\, X_3) \vee (\neg X_1 \,\&\, X_2 \,\&\, X_3)$$

— дизъюнктивная (но не совершенная) нормальная форма от трех переменных X_1, X_2, X_3, а

$$(\neg X_1 \,\&\, X_2) \vee (X_1 \,\&\, \neg X_2)$$

— СДНФ от двух переменных X_1 и X_2.

Каждую выполнимую логическую формулу можно привести эквивалентными преобразованиями к СДНФ. Для этого формулу сначала приводят к какой-нибудь ДНФ, а затем:

1) Удаляют повторения переменных в конъюнкциях, используя закон идемпотентности:

$$A \,\&\, A \,\&\, \ldots \,\&\, A \equiv A.$$

2) Убирают члены дизъюнкции, содержащие переменную вместе с ее отрицанием, а из одинаковых членов дизъюнкции удаляют все, кроме одного.

3) Если какая-либо элементарная конъюнкция в ДНФ содержит не все переменные из числа входящих в исходную формулу, то ее умножают на единицы, представляемые в виде дизъюнкций $X_j \vee \neg X_j$ каждой недостающей переменной X_j с ее отрицанием $\neg X_j$ (закон исключенного третьего). После этого раскрывают

скобки внутри каждой такой элементарной конъюнкции, применяя закон дистрибутивности.

4) Если среди членов полученной дизъюнкции окажутся одинаковые элементарные конъюнкции, то из каждой серии таковых оставляют по одной.

Например, ясно, что тождество

$$X_1 \to X_2 \equiv \neg X_1 \vee X_2$$

служит представлением логической операции импликации в виде ДНФ. Однако СДНФ для этой операции имеет вид:

$$X_1 \to X_2 \equiv (\neg X_1 \& \neg X_2) \vee (\neg X_1 \& X_2) \vee (X_1 \& X_2).$$

В то же время СДНФ операций эквивалентности и неравнозначности совпадают с формулами

$$X_1 \sim X_2 \equiv (X_1 \& X_2) \vee (\neg X_1 \& \neg X_2),$$

$$X_1 \oplus X_2 \equiv (X_1 \& \neg X_2) \vee (\neg X_1 \& X_2).$$

Возможность приведения выполнимой логической формулы к СДНФ лежит в основе алгоритма, устанавливающего равносильность или неравносильность двух заданных формул. Этот алгоритм состоит в следующем: исследуемые формулы приводят к совершенным ДНФ, содержащим все переменные, которые есть в обеих формулах, и смотрят, совпадают полученные выражения или нет; если совпадают, то формулы равносильны, если нет — они неравносильны.

Кроме ДНФ, употребляются также конъюнктивные нормальные формы. Произвольная дизъюнкция литералов называется *элементарной дизъюнкцией* (или *дизъюнктом*). *Конъюнктивной* *нормальной формой (КНФ)* называется произвольная конъюнкция дизъюнктов.

Конъюнктивные нормальные формы можно получить из ДНФ путем замены в них знаков $\vee$ на $\&$, а $\&$ на $\vee$. Символы $\&$ и $\vee$ называются двойственными друг другу.

Известно, что каждую логическую формулу можно привести эквивалентными преобразованиями к КНФ.

КНФ A называется *совершенной* и обозначается *СКНФ*, если каждая переменная формулы A входит с отрицанием или без отрицания в каждый дизъюнкт точно один раз. Например,

$$(\neg X_1 \vee X_2) \,\&\, (X_1 \vee \neg X_2) \,\&\, (\neg X_1 \vee \neg X_2)$$

— совершенная конъюнктивная нормальная форма от двух переменных X_1 и X_2.

Известно, что каждую опровержимую логическую формулу можно привести эквивалентными преобразованиями к СКНФ.

Примеры

1. Привести к ДНФ формулу $\neg(X_1 \vee X_3) \,\&\, (X_1 \to X_2)$.

Решение. Имеем

$$\neg(X_1 \vee X_3) \,\&\, (X_1 \to X_2) \equiv (\neg X_1 \,\&\, \neg X_3) \,\&\, (\neg X_1 \vee X_2) \equiv$$

$$\equiv \neg X_1 \,\&\, (\neg X_3 \,\&\, (\neg X_1 \vee X_2)) \equiv$$

$$\equiv \neg X_1 \,\&\, ((\neg X_3 \,\&\, \neg X_1) \vee (\neg X_3 \,\&\, X_2)) \equiv$$

$$\equiv (\neg X_1 \,\&\, \neg X_3 \,\&\, \neg X_1) \vee (\neg X_1 \,\&\, X_2 \,\&\, \neg X_3) \equiv$$

$$\equiv (\neg X_1 \,\&\, \neg X_3) \vee (\neg X_1 \,\&\, X_2 \,\&\, \neg X_3) \equiv (\neg X_1 \,\&\, \neg X_3).$$

2. Привести ту же формулу к СДНФ.

Решение. Начав преобразования с ДНФ, получаем

$$\neg(X_1 \vee X_3) \,\&\, (X_1 \to X_2) \equiv (\neg X_1 \,\&\, \neg X_3) \equiv$$

$$\equiv (\neg X_1 \,\&\, \neg X_3) \,\&\, (X_2 \vee \neg X_2) \equiv$$

$$\equiv (\neg X_1 \,\&\, \neg X_3 \,\&\, X_2) \vee (\neg X_1 \,\&\, \neg X_3 \,\&\, \neg X_2) \equiv$$

$$\equiv (\neg X_1 \,\&\, X_2 \,\&\, \neg X_3) \vee (\neg X_1 \,\&\, \neg X_2 \,\&\, \neg X_3).$$

1.8. Проблема разрешения и методы ее решения

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

Ясно, что эта проблема имеет положительное решение, поскольку всегда можно перебрать все возможные наборы значений аргументов и вычислить на них значения заданной формулы (то есть составить таблицу истинности). Но для больших формул эти таблицы громоздки и их использование затруднительно.

Поэтому для установления тождественной истинности или ложности формул часто используют другую процедуру распознавания, связанную с приведением формулы к КНФ или ДНФ. Сформулируем соответствующие теоремы:

1. Критерий тождественной истинности формулы. *Для того чтобы формула алгебры высказываний была тождественно истинной, необходимо и достаточно, чтобы в равносильной ей КНФ были тождественно истинны все элементарные дизъюнкции.*

2. Критерий тождественной истинности элементарной дизъюнкции. *Для того чтобы элементарная дизъюнкция была тождественно истинной, необходимо и достаточно, чтобы в ней существовала хотя бы для одной переменной пара — переменная и ее отрицание.*

3. Критерий тождественной ложности формулы. *Для того чтобы формула алгебры высказываний была тождественно ложной, необходимо и достаточно, чтобы в равносильной ей ДНФ все элементарные конъюнкции были тождественно ложны.*

4. Критерий тождественной ложности элементарной конъюнкции. *Для того чтобы элементарная конъюнкция была тождественно ложной, необходимо и достаточно, чтобы в ней существовала хотя бы для одной переменной пара — переменная и ее отрицание.*

Примеры

1. Дана формула

$$((X_1 \vee X_3) \vee \neg X_3) \rightarrow (\neg (X_1 \vee X_3) \,\&\, X_1 \,\&\, X_2).$$

Проверить, является ли эта формула тождественно ложной.

Решение. Приведем данную формулу эквивалентными преобразованиями к ДНФ:

$$((X_1 \vee X_3) \vee \neg X_3) \rightarrow (\neg (X_1 \vee X_3) \,\&\, X_1 \,\&\, X_2) \equiv$$

$$\equiv \neg((X_1 \vee X_3) \vee \neg X_3) \vee (\neg (X_1 \vee X_3) \,\&\, X_1 \,\&\, X_2) \equiv$$

$$\equiv (\neg (X_1 \vee X_3) \,\&\, X_3) \vee (\neg X_1 \,\&\, \neg X_3 \,\&\, X_1 \,\&\, X_2) \equiv$$

$$\equiv (\neg X_1 \,\&\, \neg X_3 \,\&\, X_3) \vee (\neg X_1 \,\&\, \neg X_3 \,\&\, X_1 \,\&\, X_2).$$

Каждая элементарная конъюнкция полученной ДНФ содержит хотя бы одну переменную вместе с ее отрицанием (первая содержит X_3, вторая — X_1), а потому является тождественно ложной. Следовательно, заданная формула тождественно ложна.

2. Дана формула $X_1 \& \neg X_2 \& (X_1 \to X_3) \sim \neg X_3$. Проверить, является ли эта формула тождественно истинной.

Решение. Приведем данную формулу эквивалентными преобразованиями к КНФ:

$$X_1 \& \neg X_2 \& (X_1 \to X_3) \sim \neg X_3 \equiv$$

$$\equiv X_1 \& \neg X_2 \& (X_1 \to X_3) \& \neg X_3 \vee$$
$$\vee \neg(X_1 \& \neg X_2 \& (X_1 \to X_3)) \& X_3 \equiv$$

$$\equiv X_1 \& \neg X_2 \& (\neg X_1 \vee X_3) \& \neg X_3 \vee$$
$$\vee (\neg X_1 \vee X_2 \vee \neg(\neg X_1 \vee X_3)) \& X_3 \equiv$$

$$\equiv (X_1 \& \neg X_2 \& \neg X_1 \& \neg X_3) \vee (X_1 \& \neg X_2 \& X_3 \& \neg X_3) \vee$$
$$\vee (\neg X_1 \& X_3) \vee (X_2 \& X_3) \vee (X_1 \& \neg X_3 \& X_3) \equiv$$

$$\equiv (\neg X_1 \& X_3) \vee (X_2 \& X_3) \equiv$$

$$\equiv (\neg X_1 \vee X_2) \& X_3.$$

Используя сформулированные выше критерии тождественной истинности, получаем, что заданная формула не является тождественно истинной.

1.9. Гипотезы и следствия в алгебре высказываний

Под *гипотезой* формулы A понимается такая формула B, что

$$(B \to A) \equiv 1.$$

Гипотеза формулы A называется *простой*, если она есть конъюнкция переменных или их отрицаний и после отбрасывания любого из ее сомножителей перестает быть гипотезой формулы A.

Под *следствием* формулы A понимается такая формула B, что

$$(A \to B) \equiv 1.$$

Следствие формулы A называется *простым*, если оно есть дизъюнкция переменных или их отрицаний и после отбрасывания любого из ее слагаемых перестает быть следствием формулы A.

Всякое слагаемое ДНФ является гипотезой этой ДНФ, а сомножитель КНФ — следствием этой КНФ.

Важно отметить, что:

1) Если A — гипотеза формулы B, то $A \& C$ есть тоже гипотеза формулы B.

2) Если A — следствие формулы B, то $A \vee C$ есть тоже следствие из B.

3) Если A и B — гипотезы формулы C, то $A \vee B$ — тоже гипотеза для C.

4) Если A и B — следствия из C, то $A \& B$ — тоже следствие из C.

Совершенные ДНФ и КНФ часто используются для решения задачи обзора всех гипотез и всех следствий заданной формулы. (Ведь если $A \equiv B$, то A имеет точно те же гипотезы и следствия, что и B.) У совершенной ДНФ нет других гипотез (не содержащих букв, не входящих в эту ДНФ), кроме дизъюнкций некоторых ее слагаемых или равносильных им выражений. В то же время у совершенной КНФ нет других следствий (не содержащих букв, не входящих в эту ДНФ), кроме конъюнкций некоторых ее сомножителей или равносильных им выражений.

Примеры

1. На вопрос, кто из трех студентов изучал логику, был получен ответ: если изучал первый, то изучал и третий, но неверно, что если изучал второй, то изучал и третий. Кто изучал логику?

Решение. Пусть: A — «логику изучал первый», B — «логику изучал второй», C — «логику изучал третий».

Тогда условия задачи дают:

$$1)\ A \to C \equiv 1;$$

$$2)\ \neg(B \to C) \equiv 1.$$

Составим конъюнкцию этих высказываний и упростим ее.

$$1 \equiv (A \to C) \& \neg(B \to C) \equiv (\neg A \vee C) \& \neg(B \to C) \equiv$$

$$\equiv (\neg A \vee C) \& \neg(\neg B \vee C) \equiv (\neg A \vee C) \& (\neg\neg B \& \neg C) \equiv$$

$$\equiv (\neg A \vee C) \& (B \& \neg C) \equiv (\neg A \& B \& \neg C) \vee (C \& B \& \neg C) \equiv$$

$$\equiv (\neg A \& B \& \neg C) \vee 0 \equiv \neg A \& B \& \neg C.$$

Таким образом,

$$\neg A \& B \& \neg C \equiv 1.$$

Так как здесь отрицания нет только над B, то логику изучал второй студент.

2. Определить, кто из четырех студентов сдал экзамен, если известно, что:

1) если первый сдал, то и второй сдал;

2) если второй сдал, то третий сдал или первый не сдал;

3) если четвертый не сдал, то первый сдал, а третий не сдал;

4) если четвертый сдал, то и первый сдал.

Решение. Пусть A, B, C, E — следующие высказывания:
A — «первый студент сдал экзамен»;
B — «второй студент сдал экзамен»;
C — «третий студент сдал экзамен»;
E — «четвертый студент сдал экзамен».

Тогда:

$$1)\ A \to B \equiv 1;$$

$$2)\ B \to \left(C \vee \overline{A}\right) \equiv 1;$$

$$3)\ \overline{E} \to \left(A \wedge \overline{C}\right) \equiv 1;$$

$$4)\ E \to A \equiv 1.$$

Конъюнкция этих высказываний:

$$(A \to B) \wedge \left(B \to \left(C \vee \overline{A}\right)\right) \wedge \left(\overline{E} \to \left(A \wedge \overline{C}\right)\right) \wedge (E \to A) \equiv 1.$$

Упростим эту формулу, используя свойства логических операций:

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

$$\equiv [\text{раскроем скобки, опуская знак } \wedge] \equiv$$

$$\equiv\left(\overline{A}\cdot\overline{B}\vee B\overline{B}\vee\overline{A}C\vee BC\vee\overline{A}\cdot\overline{A}\vee B\overline{A}\right)\wedge\left(E\overline{E}\vee A\overline{C}\cdot\overline{E}\vee EA\vee A\overline{C}A\right)\equiv$$

$$\equiv\left(\overline{A}C\vee BC\vee\overline{A}\right)\wedge\left(A\overline{C}\cdot\overline{E}\vee EA\vee A\overline{C}\right)\equiv\left(BC\vee\overline{A}\right)\wedge\left(EA\vee A\overline{C}\right)\equiv$$

$$\equiv BCEA\vee\overline{A}EA\vee BCA\overline{C}\vee\overline{A}A\overline{C}\equiv BCEA\equiv 1.$$

Так как конъюнкция A, B, C и E равна единице, то экзамен сдали все четыре студента.

1.10. Основные схемы логически правильных умозаключений

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

В логике умозаключения делятся на *дедуктивные* и *индуктивные*. В дедуктивных умозаключениях связи между посылками и заключением представляют собой формально-логические законы, в силу чего при истинных посылках заключение всегда оказывается истинным. В индуктивных умозаключениях между посылками и заключением имеют место такие связи, которые обеспечивают получение только правдоподобного заключения при истинных посылках. В этих рассуждениях посылки лишь подтверждают заключение.

В процессе рассуждения иногда за дедуктивные принимают умозаключения, которые таковыми не являются. Последние называют неправильными, а (собственно) дедуктивные — правильными.

Приведем примеры наиболее употребимых схем логически правильных умозаключений:

1. Утверждающий модус (modus ponens):
«Если из высказывания A следует высказывание B и справедливо (истинно) высказывание A, то справедливо B».

Логическая форма этого умозаключения такова:

$$\frac{A \to B,\ A}{B}.$$

2. Отрицающий модус (modus tollens):
«Если из A следует B, но высказывание B неверно, то неверно A».

Обозначается:

$$\frac{A \to B,\ \neg B}{\neg A}.$$

3. Утверждающе-отрицающий модус (modus ponendo-tollens):
«Если справедливо либо высказывание A, либо высказывание B (в разделительном смысле) и истинно одно из них, то другое ложно»:

$$\frac{A \oplus B,\ A}{\neg B}; \qquad \frac{A \oplus B,\ B}{\neg A}.$$

4. Отрицающе-утверждающий модус (modus tollendo-ponens):
а) *«Если истинно либо A, либо B (в разделительном смысле) и неверно одно из них, то истинно другое»:*

$$\frac{A \oplus B,\ \neg A}{B}; \quad \frac{A \oplus B,\ \neg B}{A}.$$

б) *«Если истинно A или B (в неразделительном смысле) и неверно одно из них, то истинно другое»:*

$$\frac{A \vee B,\ \neg A}{B}; \quad \frac{A \vee B,\ \neg B}{A}.$$

5. Правило транзитивности:
«Если из A *следует* B, *а из* B *следует* C, *то из* A *следует* C*»:*

$$\frac{A \to B, B \to C}{A \to C}.$$

6. Закон противоречия:
«Если из A *следует* B, *а также из* A *следует* $\neg B$, *то неверно* A*»:*

$$\frac{A \to B, A \to \neg B}{\neg A}.$$

7. Правило контрапозиции:
«Если из A *следует* B, *то из того, что неверно* B, *следует, что неверно* A*»:*

$$\frac{A \to B}{\neg B \to \neg A}.$$

8. Сложная контрапозиция:
«Если из A *и* B *следует* C, *то из* A *и* $\neg C$ *следует* $\neg B$*»:*

$$\frac{(A \wedge B) \to C}{(A \wedge \neg C) \to \neg B}.$$

9. Правило сечения:
«Если из A *следует* B, *а из* B *и* C *следует* D, *то из* A *и* C *следует* D*»:*

$$\frac{A \to B, (B \wedge C) \to D}{(A \wedge C) \to D}.$$

10. Правило импортации (объединения посылок):

$$\frac{A \to (B \to C)}{(A \wedge B) \to C}.$$

11. Правило экспортации (разъединения посылок):

$$\frac{(A \wedge B) \to C}{A \to (B \to C)}.$$

12. Простые дилеммы:

а) $\dfrac{A \to C,\, B \to C,\, A \vee B}{C}$; б) $\dfrac{A \to B,\, A \to C,\, \neg B \vee \neg C}{\neg A}$.

13. Сложные дилеммы:

а) $\dfrac{A \to B,\, C \to D,\, A \vee C}{B \vee D}$; б) $\dfrac{A \to B,\, C \to D,\, \neg B \vee \neg D}{\neg A \vee \neg C}$.

Примерами неправильных умозаключений могут служить:

а) $\dfrac{A \to B,\, B}{A}$; б) $\dfrac{A \to B,\, \neg A}{\neg B}$; в) $\dfrac{A \vee B,\, B}{\neg A}$ и др.

Алгебра высказываний дает эффективный метод проверки правильности рассуждений. Рассуждение считается правильным, если между его посылками и заключением имеет место *отношение логического следования*. Определяем последнее: *из посылок* $A_1, \ldots, A_n$ *следует заключение* B, *если импликация*

$$A_1 \,\&\, \ldots \,\&\, A_n \to B$$

является тождественно истинной.

Если формула, являющаяся переводом рассуждения на язык символов, оказывается тождественно истинной, то можно сделать вывод о том, что рассуждение правильное. Если эта формула является тождественно ложной, то рассуждение неправильное. Может оказаться, что формула является выполнимой, но не тождественно истинной. В этом случае нет оснований считать рассуждение правильным. Необходимо продолжить анализ рассуждения, но уже средствами более богатого раздела логики — средствами логики предикатов.

2. ЛОГИКА ПРЕДИКАТОВ

Напомним, что *математическая логика* — это логика, развиваемая с помощью математических методов. В то же время этот термин имеет и другой смысл: математическая логика — это логика, используемая в математике. Центральная идея математической логики состоит в том, чтобы записывать математические утверждения в виде последовательностей символов и оперировать с ними по формальным правилам. При этом правильность рассуждений можно проверять механически, не вникая в их смысл.

Не всякие высказывания и не любые логические рассуждения могут быть описаны на языке логики высказываний. Иногда высказывания касаются свойств объектов или отношений между объектами. Кроме того, необходимо иметь возможность утверждать, что любые или какие-то объекты обладают определенными свойствами или находятся в некоторых отношениях.

Поэтому следует расширить логику высказываний и построить такую логическую систему, в рамках которой можно было бы исследовать структуру и содержание тех высказываний, которые в рамках алгебры высказываний считались бы элементарными.

Такой логической системой является логика предикатов, а алгебра высказываний — ее составной частью.

2.1. Предикаты

Понятие *предиката* восходит к Аристотелю. Это понятие обобщает понятие «высказывание». Неформально говоря, предикат — это высказывание, содержащее неизвестную (или несколько неизвестных), т. е. в него можно подставлять аргументы. Если аргумент один — то предикат выражает свойство аргумента, если больше — то отношение между аргументами. Сам Аристотель ограничился в своей логике рассмотрением предикатов только от одной переменной (одноместных предикатов). Но позднее (после работ Дж. Буля) в рассмотрение вошли и предикаты от нескольких переменных.

Одноместным предикатом называется функция одной переменной, значениями которой являются высказывания об объектах, представляющих значения аргумента. Следовательно, одноместный предикат $P(x)$ — это произвольная функция переменной x, определенная на некотором множестве M и принимающая (логические) значения из множества $\{0, 1\}$. Множество M, на котором определен предикат $P(x)$, называется *предметной областью* или *областью определения предиката*, а сама переменная x — *предметной переменной*.

Таким образом, одноместный предикат $P(x)$ — это утверждение об объекте x, где x рассматривается как переменная. При фиксации значения переменной x об утверждении $P(x)$ можно сказать, истинно оно или ложно. То есть если в $P(x)$ вместо x подставить конкретный изучаемый объект a, то получаем высказывание, принадлежащее алгебре высказываний.

Областью истинности предиката $P(x)$, заданного на множестве M, называется совокупность всех x из M, при которых данный предикат обращается в истинное высказывание:

$$I_P = \{x \in M \mid P(x) \equiv 1\}.$$

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

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

Чтобы задать предикат от n аргументов (n-местный предикат), прежде всего следует указать множества $M_1, \ldots, M_n$ — области изменения *предметных переменных* $x_1, \ldots, x_n$. При этом n-*местным предикатом* называется произвольная функция n переменных $P\left(x_1, \ldots, x_n\right)$, определенная на множестве

$$M = M_1 \times \ldots \times M_n$$

и принимающая (логические) значения из множества $\{0, 1\}$.

Например, $P(x, y)$ = «y делится на x без остатка» — предикат от двух переменных, где x и y могут принимать значения или из множества натуральных чисел, или из множества целых чисел, или возможно из множества многочленов. При этом, вообще говоря, получаются различные предикаты.

Таким образом, n-местный предикат $P\left(x_1, \ldots, x_n\right)$ — это утверждение об объектах $x_1, \ldots, x_n$, где $x_1, \ldots, x_n$ рассматриваются как переменные.

Важно отметить, что при замене переменных некоторыми значениями из области определения предикат превращается в высказывание, являющееся истинным или ложным. Значение истинности этого высказывания называется значением предиката от данных переменных. Например, при подстановке в предикат $P\left(x_1, \ldots, x_n\right)$ значений переменных $x_1 = a_1, \ldots, x_n = a_n$ мы получим высказывание $P\left(a_1, \ldots, a_n\right)$. В то же время любое высказывание можно рассматривать как 0-местный предикат.

Предикаты P и Q, определенные на множестве

$$M = M_1 \times \ldots \times M_n,$$

называются *равносильными* (пишут $P \equiv Q$), если

$$P\left(x_1, \ldots, x_n\right) \equiv Q\left(x_1, \ldots, x_n\right)$$

для любого набора $x_1, \ldots, x_n$ предметных переменных из M.

Расширение логики высказываний до логики предикатов получается за счет включения в формулы утверждений, являющихся предикатами. Так как предикаты — это отображения со значениями во множестве высказываний, где введены логические операции (конъюнкция, дизъюнкция, отрицание, импликация и др.), то эти операции (логические связки) естественно определяются и для предикатов. При этом значения истинности сложных предикатов находятся в зависимости от значений связываемых предикатов по тем же правилам, что и для высказываний. Например, *конъюнкцией* n-местных предикатов P и Q, определенных на множестве $M = M_1 \times \ldots \times M_n$, называют новый n-местный предикат, определенный на этом же множестве, обозначаемый $P \wedge Q$ (или $P \& Q$), который обращается в истинное высказывание на тех и только тех значениях переменных из множеств $M_1, \ldots, M_n$, на которых в истинное высказывание обращаются оба данных предиката. Таким образом,

$$\left(P \wedge Q\right)\left(x_1, \ldots, x_n\right) \equiv P\left(x_1, \ldots, x_n\right) \wedge Q\left(x_1, \ldots, x_n\right).$$

Аналогично определяются $P \vee Q$, $\neg P$, $P \to Q$, $P \sim Q$, $P \oplus Q$ и др.

Примеры

1. $x^2 - x - 2 = 0$ — предикат. Здесь $x \in \mathbb{R}$, $I_P = \left\{-1, 2\right\}$.

2. Пусть $M = \left\{3, 4, 5, 6, 7, 8\right\}$. Предикаты $P(x)$ = «x — простое число» и $Q(x)$ = «x — четное число» заданы на множестве M. Требуется найти области истинности этих предикатов.

Решение. Очевидно, мы имеем $I_P = \left\{3, 5, 7\right\}$ и $I_Q = \left\{4, 6, 8\right\}$.

3. Найти область истинности предиката

$$P(x, y) = «y - x^2 \geq 0»$$

и изобразить его на координатной плоскости.

Решение. Область истинности данного предиката — часть плоскости Oxy, заключенная между ветвями параболы $y = x^2$.

4. Найти область истинности предиката

$$B(x, y) = «x^2 + y^2 < 5»,$$

заданного на множестве $\mathbb{Z}_+ \times \mathbb{Z}_+$ $(\mathbb{Z}_+ = \{0, 1, 2, \dots\})$.

Ответ: $\{(0;\ 0),\ (0;\ 1),\ (0;\ 2),\ (1;\ 0),\ (1;\ 1),\ (2;\ 0)\}$.

2.2. Кванторы

Функциональная природа предиката влечет за собой введение еще одного понятия — квантора. *Квантор* — это общее название для логических операций, ограничивающих область истинности какого-либо предиката. В математической логике наиболее употребительны *квантор всеобщности* $\forall$ и *квантор существования* $\exists$.

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

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

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

О высказывании $\forall x\, P(x)$ (соответственно, $\exists\, x\, P(x)$) говорят, что оно получено из предиката P навешиванием квантора всеобщности (соответственно, квантора существования) по переменной x. Переменная, на которую навешен квантор, называется *связанной переменной.*

Полезно отметить, что если некоторый предикат $P(x)$ определен на конечном множестве $M = \{a_1, \ldots, a_k\}$, то справедливы следующие тождества:

$$\forall x\, P(x) \equiv P(a_1) \wedge \ldots \wedge P(a_k); \quad \exists\, x\, P(x) \equiv P(a_1) \vee \ldots \vee P(a_k).$$

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

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

Например, к предикату от двух переменных $P(x, y)$ кванторные операции можно применить к одной переменной или к двум переменным. Получаем следующие высказывания:

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

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

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

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

Примеры

1. Пусть $P(x, y)$ = «x является матерью y». Тогда

$$\forall y \, \exists x \, P(x, y) = \text{«у каждого человека есть мать»},$$

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

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

2. Установить истинность или ложность высказывания:

$$\exists x \left(x \in \{2,5\} \to \left(x^2 - 6x + 8 = 0 \right) \right).$$

Решение. Уравнение $x^2 - 6x + 8 = 0$ имеет корни $x_1 = 2$; $x_2 = 4$. Используя тождество $A \to B \equiv \neg(A \wedge \neg B)$, исходное высказывание преобразуем к виду:

$$\exists x \left(x \in \{2,5\} \to \left(x^2 - 6x + 8 = 0 \right) \right) \equiv$$

$$\equiv \exists x \overline{\left(x \in \{2,5\} \wedge \overline{\left(x^2 - 6x + 8 = 0 \right)} \right)} \equiv$$

$$\equiv \exists x \overline{\left(x \in \{2,5\} \wedge x \notin \{2,4\} \right)} \equiv \exists x \overline{\left(x = 5 \right)} \equiv \exists x \left(x \neq 5 \right).$$

Следовательно, исходное высказывание истинно.

3. Пусть на множестве $M = \{1, 2, 3, 4\}$ задан предикат

$$P(x) = \text{«}x\text{ — четное число»}.$$

С помощью логических связок $\wedge$ и $\vee$ записать высказывания $\forall x\, P(x)$ и $\exists\, x\, P(x)$. Каковы значения истинности этих высказываний?

Решение. Так как множество M — конечное, кванторы $\forall$ и $\exists$ сводятся к повторному применению логических связок $\wedge$ или $\vee$. Имеем:

$$\forall x\, P(x) \equiv P(1)\wedge P(2)\wedge P(3)\wedge P(4) \equiv 0;$$

$$\exists\, x\, P(x) \equiv P(1)\vee P(2)\vee P(3)\vee P(4) \equiv 1.$$

2.3. Формулы логики предикатов

Как уже отмечалось, расширение логики высказываний до логики предикатов получается за счет включения в формулы утверждений, являющихся предикатами.

Формула логики предикатов определяется индуктивно по следующей схеме:

1) Всякая пропозициональная переменная (т. е. 0-местный предикат) есть формула.

2) Если P — n-местный предикат, то $P\left(x_1, \ldots, x_n\right)$ — формула. Все переменные $x_1, \ldots, x_n$ — свободные переменные, связанных переменных в этой формуле нет.

3) Если A — формула, то $\neg A$ — формула с теми же свободными и связанными переменными, что и в формуле A.

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

5) Если A — формула, содержащая свободную предметную переменную x, то $\forall x\, A$ и $\exists\, x\, A$ — тоже формулы. Переменная x

в них связана. Остальные же переменные, которые в формуле A были свободны, остаются свободными и в новых формулах. Переменные, которые были связаны в A, связаны и в новых формулах.

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

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

Формула вида $P\left(x_1, \ldots, x_n\right)$, где P — n-местный предикат, называется а*томарной* (или *элементарной*). *Литеральной формулой* (или *литералом*) называют атомарную формулу или отрицание атомарной формулы. Атомарная формула называется *положительным литералом*, а ее отрицание — *отрицательным литералом*. *Дизъюнкт* — это дизъюнкция конечного числа литералов. Если дизъюнкт не содержит литералов, его называют *пустым* дизъюнктом.

Говорят, что формула находится в *конъюнктивной нормальной форме*, если это конъюнкция конечного числа дизъюнктов. Имеет место теорема о том, что для любой бескванторной формулы существует формула, логически эквивалентная исходной и находящаяся в конъюнктивной нормальной форме.

Примеры

1. Следующее выражение является формулой логики предикатов:

$$(\forall x \, \exists \, y \, P(x,y,u)) \;\to\; \exists \, x \, Q(x,w);$$

здесь P — трехместный предикат, а Q — двухместный предикат. В этой формуле переменные x, y связаны, а переменные u, w свободны.

2. Выражение

$$(\forall x\, P(x,y)) \to Q(x,z)$$

не является формулой логики предикатов.

3. Записать, введя предикаты, в виде формулы рассуждение:

«Каждый человек любит себя. Значит, кто-то кого-нибудь любит».

Решение. Введем двухместный предикат $P(x,y)$ = «x любит y». Тогда первая часть предложения выражается высказыванием $\forall x\, P(x,x)$, а вторая — высказыванием $\exists\, x\, \exists\, y\, P(x,y)$. Искомая общая формула имеет вид:

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

4. Даны утверждения:

$A(k)$ = «число k делится на 3»;

$B(k)$ = «число k делится на 2»;

$C(k)$ = «число k делится на 12».

Выяснить, истинно или ложно следующее высказывание:

$$\forall\, k\, (A(k) \wedge B(k) \to C(k)).$$

Решение. Рассмотрим составляющие части этой формулы:

$A(k) \wedge B(k)$ = «число k делится на 6»,

$(A(k) \wedge B(k) \to C(k))$ =

= «если число k делится на 6, то оно делится на 12».

При $k=6$ данная импликация ложна. Следовательно, и формула

$$\forall k\,(A(k) \wedge B(k) \to C(k))$$

ложна.

2.4. Основные равносильности, содержащие кванторы

Пусть $P(x)$, $Q(x)$ и $R(x, y)$ — произвольные два одноместных предиката и двухместный предикат, а S — произвольное переменное высказывание (или формула, не содержащая x). Тогда имеют место следующие равносильности:

1. $\overline{\forall x\, P(x)} \equiv \exists x\, \overline{P(x)}$ (*первый закон де Моргана для кванторов*);

2. $\overline{\exists x\, P(x)} \equiv \forall x\, \overline{P(x)}$ (*второй закон де Моргана для кванторов*);

(Используя эти соотношения, можно выразить один квантор через другой.)

3. $\forall x\, P(x) \equiv \overline{\exists x\, \overline{P(x)}}$;

4. $\exists x\, P(x) \equiv \overline{\forall x\, \overline{P(x)}}$;

5. $\forall x\,(P(x) \wedge S) \equiv \forall x\, P(x) \wedge S$;

6. $\forall x\,(S \wedge P(x)) \equiv S \wedge \forall x\, P(x)$;

7. $\forall x\,(P(x) \vee S) \equiv \forall x\, P(x) \vee S$;

8. $\forall x\,(S \vee P(x)) \equiv S \vee \forall x\, P(x)$;

(Соотношения 5—8 показывают, что произвольное высказывание или формулу, не содержащую x, можно вносить под знак квантора всеобщности и выносить из-под знака этого квантора в конъюнкции и дизъюнкции.)

9. $\exists x\,(P(x)\wedge S) \equiv \exists x\,P(x)\wedge S$;

10. $\exists x\,(S\wedge P(x)) \equiv S\wedge\exists x\,P(x)$;

11. $\exists x\,(P(x)\vee S) \equiv \exists x\,P(x)\vee S$;

12. $\exists x\,(S\vee P(x)) \equiv S\vee\exists x\,P(x)$;

(Соотношения 9—12 показывают, что произвольное высказывание или формулу, не содержащую x, можно вносить под знак квантора существования и выносить из-под знака этого квантора в конъюнкции и дизъюнкции.)

13. $\forall x\,(P(x)\to S) \equiv \exists x\,P(x)\to S$;

14. $\forall x\,(S\to P(x)) \equiv S\to\forall x\,P(x)$;

15. $\exists x\,(P(x)\to S) \equiv \forall x\,P(x)\to S$;

16. $\exists x\,(S\to P(x)) \equiv S\to\exists x\,P(x)$;

17. $\forall x\,S \equiv S$;

18. $\exists x\,S \equiv S$;

19. $\forall x\,(P(x)\wedge Q(x)) \equiv \forall x\,P(x)\wedge\forall x\,Q(x)$ (*дистрибутивность квантора всеобщности относительно конъюнкции*);

20. $\exists x\,(P(x)\vee Q(x)) \equiv \exists x\,P(x)\vee\exists x\,Q(x)$ (*дистрибутивность квантора существования относительно дизъюнкции*);

21. $\exists x\,(P(x)\wedge Q(x)) \to (\exists x\,P(x)\wedge \exists x\,Q(x)) \equiv 1;$

22. $(\forall x\,P(x)\vee \forall x\,Q(x)) \to \forall x\,(P(x)\vee Q(x)) \equiv 1;$

23. $\forall x\,P(x) \equiv \forall y\,P(y);$

24. $\exists x\,P(x) \equiv \exists y\,P(y);$

25. $\forall x\,\forall y\,R(x,y) \equiv \forall y\,\forall x\,R(x,y)$ (*коммутация одноименных кванторов*);

26. $\exists x\,\exists y\,R(x,y) \equiv \exists y\,\exists x\,R(x,y)$ (*коммутация одноименных кванторов*);

27. $\exists y\,\forall x\,R(x,y) \to \forall x\,\exists y\,R(x,y) \equiv 1.$

(Соотношения 21, 22 и 27 не будут верны, если поменять направления стрелок на противоположное.)

2.5. Предваренная нормальная форма

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

Говорят, что формула логики предикатов находится в *предваренной* (или *префиксной*) *нормальной форме*, если она имеет вид

$$Q_1 x_1 \ldots Q_n x_n F,$$

где каждый Q_i есть квантор всеобщности или существования (т. е. $Q_i x_i$ обозначает $\forall x_i$ или $\exists x_i$), переменные x_i, x_j различны при $i \neq j$, а F — формула, содержащая операции $\wedge$, $\vee$, $\neg$ и не содержащая кванторов (причем знаки отрицания отнесены только к элементарным предикатам и высказываниям, то есть к элементарным формулам). Выражение $Q_1 x_1 \ldots Q_n x_n$ называют *префиксом* (или *кванторной приставкой*), а формулу F — *матрицей*.

Если все Q_i — $\forall$, то эта форма называется $\forall$-*формулой*. Если все Q_i — $\exists$, то эта форма называется $\exists$-*формулой*. Если существует i такое, что $Q_1, \ldots, Q_i$ суть $\exists$, а $Q_{i+1}, \ldots, Q_n$ суть $\forall$, то эта форма называется *скулемовской нормальной формой* (или $\exists\forall$-*формулой*).

Формулы, в которых из логических символов имеются только символы $\wedge$, $\vee$, $\neg$, причем символ $\neg$ встречается лишь перед символами предикатов, называют *приведенными* формулами. Таким образом, формула логики предикатов, находящаяся в префиксной нормальной форме, является, в частности, приведенной формулой. Для любой формулы существует равносильная ей приведенная формула, которая имеет те же множества свободных и связанных переменных. Более того, всякая формула логики предикатов с помощью равносильных преобразований может быть приведена к равносильной ей префиксной нормальной форме.

Например, следующие формулы находятся в префиксной нормальной форме:

$$\forall x\, \forall y \left(\neg P(x,y) \wedge Q(y)\right);$$

$$\forall x\, \exists y\, \forall z \left(A(x,y) \vee \neg B(y,z) \vee C(z)\right).$$

Процедура приведения формулы логики предикатов к префиксной нормальной форме:

1) Сначала избавляются от операций импликации, эквивалентности и неравнозначности, выразив их через логические связки $\neg$, & и $\vee$ по законам:

$$A \to B \equiv \neg A \vee B,$$

$$A \sim B \equiv (A \,\&\, B) \vee (\neg A \,\&\, \neg B),$$

$$A \oplus B \equiv (A \& \neg B) \vee (\neg A \& B).$$

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

$$\neg (A \& B) \equiv \neg A \vee \neg B,$$

$$\neg (A \vee B) \equiv \neg A \& \neg B,$$

$$\neg \left(\forall x\, P(x)\right) \equiv \exists\, x \left(\neg P(x)\right),$$

$$\neg \left(\exists\, x\, P(x)\right) \equiv \forall x \left(\neg P(x)\right),$$

$$\neg\neg A \equiv A.$$

3) Для формул, содержащих подформулы вида

$$\forall x\, P(x) \vee \forall x\, Q(x), \qquad \exists\, x\, P(x) \wedge \exists\, x\, Q(x)$$

и т. п., вводят новые переменные, позволяющие выносить знаки кванторов наружу. Например:

$$\exists\, x\, P(x) \wedge \exists\, x\, Q(x) \equiv \exists\, x\, \exists\, y\, (P(x) \wedge Q(y));$$

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

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

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

Примеры

1. Привести равносильными преобразованиями к префиксной нормальной форме следующую формулу логики предикатов:

$$\overline{S \to \exists\, x\, A(x)}.$$

Решение. Имеем

$$\overline{S\to\exists\, x\, A(x)} \equiv \left[A\to B \equiv \overline{A}\vee B\right] \equiv \overline{\overline{S}\vee\exists\, x\, A(x)} \equiv$$

$$\equiv \left[\overline{A\vee B} \equiv \overline{A}\wedge\overline{B}\right] \equiv S\wedge\overline{\exists\, x\, A(x)} \equiv \left[\overline{\exists\, x\, A(x)} \equiv \forall x\ \overline{A(x)}\right] \equiv$$

$$\equiv S\wedge\forall x\ \overline{A(x)} \equiv \forall x\left(S\wedge\overline{A(x)}\right).$$

2. Привести следующую формулу к предваренной нормальной форме, считая A и B бескванторными формулами:

$$\left(\exists\, x\ \forall y\ A(x,y)\right)\wedge\left(\exists\, x\ \forall y\ B(x,y)\right).$$

Решение. Имеем

$$\left(\exists\, x\ \forall y\ A(x,y)\right)\wedge\left(\exists\, x\ \forall y\ B(x,y)\right) \equiv$$

$$\equiv \left(\exists\, x\ \forall y\ A(x,y)\right)\wedge\left(\exists\, z\ \forall y\ B(z,y)\right) \equiv$$

$$\equiv \exists\, x\,\exists\, z\left(\left(\forall y\ A(x,y)\right)\wedge\left(\forall y\ B(z,y)\right)\right) \equiv$$

$$\equiv \exists\, x\,\exists\, z\ \forall y\left(A(x,y)\wedge B(z,y)\right).$$

3. Привести к предваренной нормальной форме следующую формулу:

$$\forall x\left(\neg A(x) \to \exists\, y\left(\neg B(y)\right)\right)\to\left(B(z)\to A(z)\right).$$

Решение. Имеем

$$\forall x \left(\neg A(x) \to \exists y \left(\neg B(y)\right)\right) \to \left(B(z) \to A(z)\right) \equiv$$

$$\equiv \forall x \left(A(x) \vee \exists y \left(\neg B(y)\right)\right) \to \left(\neg B(z) \vee A(z)\right) \equiv$$

$$\equiv \neg \forall x \left(A(x) \vee \exists y \left(\neg B(y)\right)\right) \vee \left(\neg B(z) \vee A(z)\right) \equiv$$

$$\equiv \exists x \left(\neg A(x) \wedge \forall y\, B(y)\right) \vee \left(\neg B(z) \vee A(z)\right) \equiv$$

$$\equiv \exists x\, \forall y \left[\left(\neg A(x) \wedge B(y)\right) \vee \left(\neg B(z) \vee A(z)\right)\right] \equiv$$

$$\equiv \exists x\, \forall y \left[\left(\neg A(x) \vee \neg B(z) \vee A(z)\right) \wedge \left(B(y) \vee \neg B(z) \vee A(z)\right)\right].$$

2.6. Тавтологии логики предикатов

Напомним, что формула алгебры высказываний, принимающая значение 1 (истина) при любом наборе значений входящих в нее переменных, называется тождественно истинной или тавтологией.

Формулу логики предикатов называют *общезначимой* (или *тождественно истинной*, или *тавтологией*), если при всякой подстановке вместо предикатных переменных любых конкретных предикатов, заданных на каких угодно множествах, она превращается в тождественно истинный предикат. Общезначимую формулу называют также *логическим законом*.

Формулу A назовем *выполнимой*, если формула $\neg A$ не является тождественно истинной.

Формулу логики предикатов называют *тождественно ложной* (или *противоречием*), если при всякой подстановке вместо предикатных переменных любых конкретных предикатов, заданных на каких угодно множествах, она превращается в тождественно ложный предикат.

Напомним, что проблемой разрешения для алгебры высказываний называется следующий вопрос: *существует ли алгоритм, позволяющий для произвольной логической формулы в конечное число шагов выяснить, является ли она тождественно истинной?* Как отмечалось, этот вопрос имеет положительное решение (подробно о понятии алгоритма будет рассказано ниже, на с. 63).

Проблема распознавания общезначимости формул логики предикатов существенно сложнее. Она также называется *проблемой разрешения*. Метод перебора всех вариантов здесь не применим, так как вариантов может быть бесконечно много. В 1936 году А. Чёрч доказал, что в общем виде проблема разрешения для логики предикатов не имеет алгоритмического решения. Сформулируем соответствующую теорему:

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

Примеры

1. Доказать общезначимость следующей формулы логики предикатов:

$$\forall x\Big(P(x) \to Q(x)\Big)\to\Big(\forall x\, P(x) \to \forall x\, Q(x)\Big).$$

Решение. Упростим формулу, используя равносильные преобразования:

$$\forall x\Big(P(x) \to Q(x)\Big)\to\Big(\forall x\, P(x) \to \forall x\, Q(x)\Big) \equiv$$

$$\equiv \overline{\forall x \left(\overline{P(x)} \vee Q(x)\right)}\vee\left(\overline{\forall\, x\, P(x)} \vee \forall x\, Q(x)\right) \equiv$$

$$\equiv \exists\, x \left(\overline{\overline{P(x)} \vee Q(x)}\right)\vee\left(\exists\, x\, \overline{P(x)} \vee \forall x\, Q(x)\right) \equiv$$

$$\equiv \exists x \left(P(x) \wedge \overline{Q(x)}\right) \vee \left(\exists x \overline{P(x)} \vee \forall x \, Q(x)\right) \equiv$$

$$\equiv \exists x \left(P(x) \wedge \overline{Q(x)}\right) \vee \left(\exists x \overline{P(x)}\right) \vee \left(\forall x \, Q(x)\right) \equiv$$

$$\equiv \exists x \left(\left(P(x) \wedge \overline{Q(x)}\right) \vee \overline{P(x)}\ \right) \vee \left(\forall x \, Q(x)\right) \equiv$$

$$\equiv \exists x \left(\overline{Q(x)} \vee \overline{P(x)}\right) \vee \left(\forall x \, Q(x)\right) \equiv$$

$$\equiv \exists x \left(\overline{P(x)} \vee \overline{Q(x)}\right) \vee \left(\forall x \, Q(x)\right) \equiv$$

$$\equiv \left(\exists x \overline{P(x)}\right) \vee \left(\exists x \overline{Q(x)}\right) \vee \left(\forall x \, Q(x)\right) \equiv$$

$$\equiv \left(\exists x \overline{P(x)}\right) \vee \left(\overline{\forall x \, Q(x)}\right) \vee \left(\forall x \, Q(x)\right) \equiv$$

$$\equiv \left(\exists x \overline{P(x)}\right) \vee 1 \equiv 1.$$

Таким образом, общезначимость формулы доказана.

2. Доказать тождественную ложность следующей формулы:

$$\exists x \, \exists y \left(\left(F(x) \rightarrow F(y)\right) \wedge \left(F(x) \rightarrow \overline{F(y)}\right) \wedge F(x)\right).$$

Решение. Упростим формулу, используя равносильные преобразования:

$$\exists x \, \exists y \left(\left(F(x) \rightarrow F(y)\right) \wedge \left(F(x) \rightarrow \overline{F(y)}\right) \wedge F(x)\right) \equiv$$

$$\equiv \left[A \to B \equiv \overline{A} \vee B \right] \equiv$$

$$\equiv \exists x \, \exists y \left(\left(\overline{F(x)} \vee F(y) \right) \wedge \left(\overline{F(x)} \vee \overline{F(y)} \right) \wedge F(x) \right) \equiv$$

$$\equiv \left[(A \vee B) \wedge C \equiv (A \wedge C) \vee (B \wedge C) \right] \equiv$$

$$\equiv \exists x \, \exists y \left(\left(\overline{F(x)} \vee F(y) \right) \wedge \left(\left(\overline{F(x)} \wedge F(x) \right) \vee \left(\overline{F(y)} \wedge F(x) \right) \right) \right) \equiv$$

$$\equiv \exists x \, \exists y \left(\left(\overline{F(x)} \vee F(y) \right) \wedge \left(0 \vee \left(\overline{F(y)} \wedge F(x) \right) \right) \right) \equiv$$

$$\equiv \left[\overline{A} \wedge A \equiv 0 \right] \equiv$$

$$\equiv \exists x \, \exists y \left(\left(\overline{F(x)} \vee F(y) \right) \wedge \left(\overline{F(y)} \wedge F(x) \right) \right) \equiv$$

$$\equiv \exists x \, \exists y \left(\overline{F(x)} \wedge \overline{F(y)} \wedge F(x) \right) \vee \left(F(y) \wedge \overline{F(y)} \wedge F(x) \right) \equiv$$

$$\equiv \exists x \, \exists y \left(0 \vee 0 \right) \equiv 0.$$

Таким образом, формула тождественно ложна, то есть является противоречием.

3. Доказать общезначимость следующей формулы:

$$\exists x \left(A(x) \wedge \left(S \to B(x) \right) \right) \to \left(\forall x \left(A(x) \to \neg B(x) \right) \to \neg S \right).$$

Решение. Проведем доказательство методом от противного. Предположим, что

$$\exists x\,\bigl(A(x)\wedge(S\to B(x))\bigr)\to\bigl(\forall x\,(A(x)\to\neg B(x))\to\neg S\bigr)\equiv 0.$$

Но произвольная импликация $P\to Q$ ложна в том и только в том случае, когда P истинно, а Q ложно. Следовательно, мы имеем

$$\exists x\,\bigl(A(x)\wedge(S\to B(x))\bigr)\equiv 1$$

и

$$\forall x\,\bigl(A(x)\to\neg B(x)\bigr)\to\neg S\equiv 0.$$

Рассуждая аналогично, из последнего равенства получаем, что $\neg S\equiv 0$ (т. е. $S\equiv 1$) и $\forall x\,\bigl(A(x)\to\neg B(x)\bigr)\equiv 1$.

Из того, что $\exists x\,\bigl(A(x)\wedge(S\to B(x))\bigr)\equiv 1$, следует, что в области определения предметной переменной x найдется константа a, для которой

$$A(a)\wedge\bigl(S\to B(a)\bigr)\equiv 1.$$

Но тогда $A(a)\equiv 1$ и $\bigl(S\to B(a)\bigr)\equiv 1$, а поскольку $S\equiv 1$, мы видим, что $B(a)\equiv 1$. Следовательно,

$$\bigl(A(a)\to\neg B(a)\bigr)\equiv 0,$$

что противоречит равенству $\forall x\,\bigl(A(x)\to\neg B(x)\bigr)\equiv 1$.

3. ТЕОРИЯ АЛГОРИТМОВ

Во всех сферах своей деятельности, и в частности, в сфере обработки информации, человек сталкивается с различными способами или методиками решения задач. Они определяют порядок выполнения действий для получения желаемого результата — мы можем трактовать это как первоначальное или интуитивное определение алгоритма.

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*. По существу вся математика связана с теми или иными алгоритмами. Одним из наиболее замечательных достижений математической логики и теории алгоритмов явилась разработка понятия рекурсивной функции и формулировка тезиса Чёрча, утверждающего, что понятие рекурсивной функции является уточнением интуитивного понятия алгоритма.

Само понятие «алгоритм» сформировалось лишь в первой половине XX века. Началом систематической разработки теории алгоритмов можно считать 1936 году, когда А. Чёрч опубликовал первое уточнение понятия вычислимой функции и привёл первый пример функции, не являющейся вычислимой. Приблизительно в это же время появились первые уточнения понятия алгоритма (в терминах идеализированных вычислительных машин). В дальнейшем теория алгоритмов получила развитие в трудах многих математиков (А. М. Тьюринг, Э. Л. Пост, С. К. Клини, А. А. Марков, А. Н. Колмогоров и др.).

Алгоритмом называется общий единообразный, точно определенный способ решения любой задачи из данной массовой проблемы.

Алгоритм — это процесс последовательного построения величин таким образом, что в начальный момент задаётся исходная конечная система величин, а в каждый следующий момент система величин получается по определенному закону из системы величин, имевшихся в предыдущий момент. Последовательный процесс построения величин должен быть конечным и давать результат, то есть решение задачи.

Областью применимости алгоритма называется совокупность тех объектов, к которым он применим. Про алгоритм говорят, что он «вычисляет функцию f», коль скоро его область применимости совпадает с областью определения f, и он перерабатывает всякий x из своей области применимости в $f(x)$.

Проблема построения алгоритма, обладающего теми или иными свойствами, называется *алгоритмической проблемой*. Важный пример алгоритмической проблемы — проблема вычисления данной функции (требуется построить алгоритм, вычисляющий эту функцию). Функция называется *вычислимой*, если существует вычисляющий ее алгоритм.

Основными математическими моделями понятия алгоритма являются машины Тьюринга, частично рекурсивные функции и др.

3.1. Машина Тьюринга

Попытки формализовать понятие алгоритма привели к созданию машины Тьюринга, как некоторого воображаемого устройства, реализующего алгоритм.

Машина Тьюринга — абстрактное устройство, состоящее из бесконечной в обе стороны ленты, считывающей и печатающей головки, способной перемещаться вправо и влево, и управляющего устройства. Лента разбита на ячейки (клетки). Считывающая и печатающая головка перемещается вдоль ленты так, что в каждый момент времени она обозревает ровно одну ячейку ленты. В ячейках могут быть записаны символы некоторого конечного алфавита

$$A=\{a_0, a_1, \ldots, a_k\}.$$

Устройство обладает некоторым конечным набором состояний

$$Q=\{q_0, q_1, \ldots, q_n\}.$$

Программа для машины Тьюринга представляет собой конечный список команд вида:

$$qa \to q'a'D,$$

где $a,\ a' \in A,\quad q,\ q' \in Q,\quad D \in \{R,\ L,\ S\};\quad R,\ L,\ S$ — вправо, влево, стоп.

Команда расшифровывается так: если машина находится в состоянии q и считанный с ленты символ равен a, то машина переходит в состояние q', печатает в текущей клетке символ a' и затем выполняет одно из трех действий D. Если $D = R$, то машина смещается на одну клетку вправо, если $D = L$, то на одну клетку влево, а если $D = S$, то машина никуда не смещается.

Необходимо, чтобы в программе не было разных команд с одинаковыми входами вида

$$qa \to q'a'D' \quad \text{и} \quad qa \to q''a''D''$$

— это противоречит однозначности алгоритма. Порядок расположения команд программы, в отличие от привычных языков программирования, может быть произвольным.

Изначально машина находится в состоянии q_1. Если машина пришла в состояние q_0, то она останавливается.

Машина называется *применимой* к некоторой записи на ленте, если при работе над этой записью после конечного числа шагов она останавливается.

Если она никогда не остановится, то машина называется *неприменимой* к этой записи.

Пример

Найти результат применения машины Тьюринга, заданной программой, к записям на ленте

$$P_1 = 0\,1\,1\,1\,0\,1\,0 \qquad \text{и} \qquad P_2 = 0\,1\,1\,1\,1\,0:$$

$$
\begin{aligned}
q_1 0 &\to q_1 0R,\\
q_1 1 &\to q_2 0R,\\
q_2 0 &\to q_0 1S,\\
q_2 1 &\to q_1 0R.
\end{aligned}
$$

Решение. Имеем

$$P_1:\ 0\ \underset{q_1}{1}\ 1 1 0 1 0 \to 0\ 0\ \underset{q_2}{1}\ 1 0 1 0 \to 0\ 0\ 0\ \underset{q_1}{1}\ 0 1 0 \to$$

$$\to 0\ 0\ 0\ 0\ \underset{q_2}{0}\ 1\ 0 \to 0\ 0\ 0\ 0\ \underset{q_0}{1}\ 1\ 0,$$

$$P_2:\ 0\ \underset{q_1}{1}\ 1 1 1 0 \to 0\ 0\ \underset{q_2}{1}\ 1 1 0 \to 0\ 0\ 0\ \underset{q_1}{1}\ 1\ 0 \to$$

$$\to 0\ 0\ 0\ 0\ \underset{q_2}{1}\ 0 \to 0\ 0\ 0\ 0\ 0\ \underset{q_1}{0} \to 0\ 0\ 0\ 0\ 0\ 0\ \underset{q_1}{0} \to \ldots.$$

В последнем случае машина никогда не остановится, и будет работать вечно. Таким образом, она неприменима ко второй записи на ленте.

Машину Тьюринга удобно применять при вычислении функций вида

$$f:\ \mathbb{Z}_+^k \to \mathbb{Z}_+^m,$$

где $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ — множество всех целых неотрицательных чисел. Введем правила записи чисел из множества $\mathbb{Z}_+$ на ленте. Пусть алфавит A состоит из символов 0 и 1. Тогда число n будем записывать с помощью последовательности из $n+1$ единицы, а набор $(n_1, n_2, \ldots, n_k)$ в виде

$$\ldots 0\ \underbrace{11\ldots 1}_{n_1+1}\ 0\ \underbrace{11\ldots 1}_{n_2+1}\ 0\ldots 0\ \underbrace{11\ldots 1}_{n_k+1}\ 0\ \ldots.$$

Таким образом, на ленте:

$$\begin{gathered} 0 \to 0\,1\,0; \\ 1 \to 0\,1\,1\,0; \\ 2 \to 0\,1\,1\,1\,0; \\ \vdots \end{gathered}$$

Примеры

1. Построим машину Тьюринга, которая к числу на ленте будет прибавлять 1. Она дойдет до конца массива из единиц, поставит туда 1 и вернется назад, т. е.

$$0\ \underset{q_1}{1}\underbrace{\ 1\ldots 1}_{n+1}\ 0 \to 0\ \underset{q_0}{1}\underbrace{\ 1\ldots 1}_{n+2}\ 0.$$

Вот её программа:

$$\begin{gathered} q_1 1 \to q_1 1R, \\ q_1 0 \to q_2 1L, \\ q_2 1 \to q_2 1L, \\ q_2 0 \to q_0 0R. \end{gathered}$$

2. Построим машину Тьюринга, которая, имея на ленте два массива из единиц, разделенные нулями, заполняет эти нули единицами и останавливается у последней единицы второго массива, т. е.

$$0\ \underset{q_1}{1}\ 1\,1\,0\,0\,0\,1\,1\,1\,1\,0 \to 0\,1\,1\,1\,1\,1\,1\,1\,1\,1\ \underset{q_0}{1}\ 0.$$

Алгоритм действий можно записать словами:

1-й шаг: пройти первый массив единиц, найти массив нулей, идущий после него и заменить первый нуль единицей;

2-й шаг: идти через массив нулей, заменяя их единицами, до тех пор, пока не появится второй массив единиц;

3-й шаг: пройти второй массив единиц и остановиться в его конце.

Машина Тьюринга, реализующая этот алгоритм, имеет следующую программу:

$q_1 1 \to q_1 1R,$ $q_1 0 \to q_2 1R,$	1-й шаг
$q_2 0 \to q_2 1R,$ $q_2 1 \to q_3 1R,$	2-й шаг
$q_3 1 \to q_3 1R,$ $q_3 0 \to q_0 0L.$	3-й шаг

Говорят, что машина Тьюринга *вычисляет* функцию

$$f\left(x_1, \ldots, x_k\right):\ \mathbb{Z}_+^k \to \mathbb{Z}_+^m,$$

если на любом наборе $\left(a_1, \ldots, a_k\right) \in D(f)$ машина останавливается и на ленте остается результат $\left(b_1, \ldots, b_m\right) = f\left(a_1, \ldots, a_k\right)$, а в случае $\left(a_1, \ldots, a_k\right) \notin D(f)$ она работает вечно, т. е. неприменима к таким входным данным.

Универсальная кодировка машины Тьюринга

Построим универсальную кодировку программы машины Тьюринга M. Для этого занумеруем алфавит $\{a_0, a_1, \ldots, a_k\}$, состояния $\{q_0, q_1, \ldots, q_n\}$ и сдвиги натуральными числами:

R	L	S	a_0	a_1	$\ldots$	a_k	q_0	q_1	$\ldots$	q_n
1	3	5	7	9	$\ldots$	$2k+1$	0	2	$\ldots$	$2n$

Тогда любую команду можно закодировать набором из 5 чисел, а этот набор представить на ленте с помощью алфавита $\{1, *\}$ стандартным образом, помещая вместо разделяющих нулей символ $*$. При этом мы получим код команды, который обозначим через K.

Теперь можно составить код всей программы

$$K(M) = K_1 * K_2 * \cdots * K_p,$$

где K_i — коды всех команд программы. Отметим, что по коду всегда можно восстановить программу.

Пример

Построить код машины Тьюринга с программой:

$$q_1 1 \to q_1 1R,$$
$$q_1 0 \to q_2 1L,$$
$$q_2 1 \to q_2 1L,$$
$$q_2 0 \to q_0 0R.$$

Решение. Закодируем набором из 5 чисел каждую команду, используя таблицу кодов:

$$2*9*2*9*1, \quad 2*7*4*9*3, \quad 4*9*4*9*3, \quad 4*7*0*7*1.$$

Теперь представим коды команд с помощью алфавита $\{1,\ *\}$:

$$1^3 * 1^{10} * 1^3 * 1^{10} * 1^2 * 1^3 * 1^8 * 1^5 * 1^{10} * 1^4 * 1^5 * 1^{10} * 1^5 * 1^{10} * 1^4 * 1^5 * 1^8 * 1 * 1^8 * 1^2,$$

где 1^k есть единица, повторенная k раз.

3.2. Алгоритмически неразрешимые проблемы

Пусть алфавит машины Тьюринга содержит символы 1 и $*$. В этом случае можно предложить машине в качестве исходной записи на ленте ее код. Если при работе над собственным кодом машина Тьюринга M останавливается, то она называется *самоприменимой*. Если она не останавливается, она называется *несамоприменимой*.

Возникает вопрос: *существует ли машина M_S, которая по коду любой машины M определяет, самоприменима ли она?*

Если она существует, она должна быть применима к коду любой машины M и должна останавливаться на клетке с 1 в случае самоприменимости M, и на клетке с 0 в противном случае. Ответ на поставленный вопрос дает следующая теорема.

Теорема. M_S *не существует, то есть проблема самоприменимости алгоритмически неразрешима.*

Доказательство. Пусть машина Тьюринга S решает проблему самоприменимости, т. е., начав работу с кода машины T, приходит в состояние

$$\ldots\, q_0 1 \,\ldots, \qquad (*)$$

если машина T самоприменима, и в состояние

$$\ldots\, q_0 0 \,\ldots, \qquad (**)$$

если T несамоприменима.

Рассмотрим машину R, программа которой состоит из всех команд машины S и еще двух команд $q_0 1 \to q_0 1$ и $q_0 0 \to q_0^{/} 0$. Здесь q_0 — не заключительное, а $q_0^{/}$ — заключительное состояние. Если машина R самоприменима, то, начав работу со своего кода, она, в силу команд машины S придет в состояние (*). Затем в силу команды $q_0 1 \to q_0 1$ она будет работать бесконечно. Это значит, что R несамоприменима. Противоречие. Точно так же, если R несамоприменима, она придет сначала в состояние (**), а затем остановится в силу команды $q_0 0 \to q_0^{/} 0$. Значит, R самоприменима. Полученное противоречие и доказывает теорему.

3.3. Рекурсивные функции. Тезис Чёрча

Все известные примеры алгоритмов можно свести к вопросу вычисления значений подходящей функции. Естественно назвать функцию, значения которой могут находиться с помощью некоторого алгоритма, вычислимой функцией. Таким образом, *вычислимая* функция — это такая функция, для которой существует алгоритм, вычисляющий ее значения (вычисление функции происходит последовательно по определенным, заранее заданным, правилам и инструкциям).

Рассмотрим функции $f : \mathbb{Z}_+^n \to \mathbb{Z}_+$, где $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$. *Частичными* числовыми функциями

$$f(x_1, \ldots, x_n), \ x_i \in \mathbb{Z}_+ \quad (1 \le i \le n),$$

называют функции, определенные не на всех наборах

$$(x_1, \ldots, x_n) \in \mathbb{Z}_+ \times \ldots \times \mathbb{Z}_+ = \mathbb{Z}_+^n.$$

Назовем *простейшими* следующие всюду определенные функции:

$o(x)=0$ — *нулевая функция*;

$s(x)=x+1$ — *функция следования*;

$I_n^m\left(x_1, \ldots, x_n\right)=x_m$, $1 \le m \le n$, — *функция выбора аргумента*.

Все простейшие функции могут быть вычислены на машине Тьюринга:

1. $o\left(x\right)=0$:

$$\begin{aligned} &q_1 1 \to q_1 1R, \\ &q_1 0 \to q_2 0L, \\ &q_2 1 \to q_2 0L, \\ &q_2 0 \to q_0 1S. \end{aligned}$$

2. $s\left(x\right)=x+1$:

$$\begin{aligned} &q_1 1 \to q_1 1R, \\ &q_1 0 \to q_2 1L, \\ &q_2 1 \to q_2 1L, \\ &q_2 0 \to q_0 0R. \end{aligned}$$

3. $f\left(x, y\right)=y$, $\quad 0 \underset{q_1}{}\underbrace{1\ 1 \ldots 1}_{x+1}\ 0\ \underbrace{1\ 1 \ldots 1}_{y+1}\ 0 \ \to\ 0 \underset{q_0}{}\underbrace{1\ 1 \ldots 1}_{y+1}\ 0$:

$$\begin{aligned} &q_1 1 \to q_1 0R, \\ &q_1 0 \to q_0 0R. \end{aligned}$$

Определим операции, которые из простейших строят новые функции.

Операция суперпозиции

Пусть даны функция $f\left(y_1, \ldots, y_m\right)$ от m аргументов и функции $g_1\left(x_1, \ldots, x_n\right), \ldots, g_m\left(x_1, \ldots, x_n\right)$ от n аргументов. Функция

$$F\left(x_1, \ldots, x_n\right) = f\left(g_1\left(x_1, \ldots, x_n\right), \ldots, g_m\left(x_1, \ldots, x_n\right)\right)$$

называется *суперпозицией функций* f и $g_1, \ldots, g_m$.

Примеры

1. $s(o(x)) = 1$;

2. $s(s(x)) = (x+1)+1 = x+2$.

Операция примитивной рекурсии

Пусть при $n \geq 1$ даны какие-либо две функции $g\left(x_1, \ldots, x_n\right)$ и $h\left(x_1, \ldots, x_n, x_{n+1}, x_{n+2}\right)$. Определим третью функцию

$$f\left(x_1, \ldots, x_n, x_{n+1}\right)$$

по следующей схеме (*схеме примитивной рекурсии*):

$$\begin{cases} f\left(x_1, \ldots, x_n, 0\right) = g\left(x_1, \ldots, x_n\right), \\ f\left(x_1, \ldots, x_n, y+1\right) = h\left(x_1, \ldots, x_n, y, f\left(x_1, \ldots, x_n, y\right)\right). \end{cases}$$

Эти равенства однозначно определяют функцию f. Говорят, что функция f получена из функций g и h с помощью *операции примитивной рекурсии* по переменной x_{n+1}.

При $n = 0$ схема примитивной рекурсии имеет следующий вид:

$$\begin{cases} f(0) = a, \\ f(y+1) = h(y, f(y)), \end{cases}$$

где a — константа (число из множества $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$).

Функции, которые могут быть получены из простейших функций o, s, I_n^m с помощью конечного числа применений операций суперпозиции и примитивной рекурсии, называются *примитивно рекурсивными*.

Операция минимизации

Пусть $f(x_1, x_2, \ldots, x_n)$ $(x_i \in \mathbb{Z}_+)$ — некоторая частичная числовая функция.

Операция минимизации по i-й переменной функции $f(x_1, x_2, \ldots, x_n)$ обозначается

$$g(x_1, x_2, \ldots, x_n) = \mu_y\Big(f(x_1, x_2, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n) = x_i \Big)$$

и определяется следующим образом.

Пусть $(x_1, x_2, \ldots, x_n)$ — произвольный набор чисел из множества $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$. Рассмотрим соотношение

$$f(x_1, x_2, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n) = x_i, \qquad (*)$$

которое будем рассматривать, как уравнение относительно y. Это уравнение будем решать подбором, подставляя вместо y последовательно числа $0, 1, 2, \ldots$. Возможны случаи:

1) На некотором шаге левая часть соотношения (*) не определена. В этом случае считаем, что на наборе $(x_1, x_2, \ldots, x_n)$ операция минимизации не определена.

2) На каждом шаге левая часть соотношения (*) определена, но ни при каких y равенство не выполняется. В этом случае также считаем, что на наборе $\left(x_1, x_2, \ldots, x_n\right)$ операция минимизации не определена.

3) Левая часть соотношения (*) определена при $y=0$, $y=1, \ldots, y=y_0-1$, $y=y_0$, но при $y<y_0$ равенство (*) не выполнялось, а при $y=y_0$ оно выполняется. В этом случае полагаем $g\left(x_1, x_2, \ldots, x_n\right)=y_0$, т. е. число y_0 считается значением операции минимизации на наборе $\left(x_1, x_2, \ldots, x_n\right)$.

Функции, которые могут быть получены из простейших функций $o,\ s,\ I_n^m$ с помощью конечного числа применений операций суперпозиции, примитивной рекурсии и минимизации называются *частично рекурсивными.*

Общерекурсивные функции — это подмножество частично рекурсивных функций, определенных для всех значений аргументов.

Легко понять, что множество общерекурсивных функций включает в себя множество примитивно рекурсивных функций, а частично рекурсивные функции включают в себя общерекурсивные функции.

Известно, что не всякая общерекурсивная функция является примитивно рекурсивной. В то же время существуют частично рекурсивные функции, которые не могут быть продолжены до общерекурсивных.

Заметим, что частично рекурсивные функции иногда называют просто *рекурсивными* функциями.

Для любой частично рекурсивной функции можно указать алгоритм вычисления ее значений, т. е. все частично рекурсивные функции суть вычислимые функции. Обращение этого высказывания носит название *тезиса Чёрча*.

Тезис Чёрча. *Всякая вычислимая частичная числовая функция является частично рекурсивной функцией.*

Таким образом, согласно этому принципу класс функций, вычислимых с помощью *алгоритмов* в широком интуитивном смысле, совпадает с классом частично рекурсивных функций.

Тезис Чёрча не может быть строго доказан, но считается справедливым, поскольку он подтверждается опытом, накопленным в математике за всю ее историю. Какие бы классы алгоритмов ни строились, вычисляемые ими числовые функции оказывались частично рекурсивными.

В теории алгоритмов также известно следующее соглашение.

Тезис Тьюринга. *Всякий алгоритм представим в форме машины Тьюринга.*

Согласно этому тезису, всякая вычислимая в интуитивном смысле функция вычислима с помощью некоторой машины Тьюринга. Принятие тезиса Тьюринга равносильно принятию тезиса Чёрча для частично рекурсивных функций.

Пример

Функция

$$g(x) = \begin{cases} x-1, & x > 0, \\ \text{не определено}, & x = 0 \end{cases}$$

частично рекурсивна.

Действительно,

$$g(x) = \mu_y\{s(y) = y+1 = x\}.$$

Следовательно, она получена из простейшей функции с помощью оператора минимизации.

Рекомендуемая литература

Обязательная

1. *Гаврилов* Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2009.
2. *Гуц* А. К. Математическая логика и теория алгоритмов. — М.: Либроком, 2009.
3. *Зюзьков* В. М., Шелупанов А. А. Математическая логика и теория алгоритмов. — М.: Горячая линия — Телеком, 2007.
4. *Игошин* В. И. Задачи и упражнения по математической логике и теории алгоритмов. — М.: Академия, 2008.
5. *Колмогоров* А. Н., Драгалин А. Г. Математическая логика. — М.: Физматлит, 2006.
6. *Лавров* И. А., Максимова Л. Л. Задачи по теории множеств, математической логике и теории алгоритмов. — М.: Физматлит, 2009.
7. *Тишин* В. В. Дискретная математика в примерах и задачах. — СПб.: БХВ-Петербург, 2008.
8. *Шапорев* С. Д. Математическая логика. Курс лекций и практических занятий. — СПб.: БХВ-Петербург, 2009.

Дополнительная

1. *Акимов* О. Е. Дискретная математика. Логика, группы, графы. — М.: Лаборатория Базовых Знаний, 2003.
2. *Гиндикин* С. Г. Алгебра логики в задачах. — М.: Наука, 1972.
3. *Горбатов* В. А. Основы дискретной математики. — М.: Высшая школа, 1986.
4. *Кулабухов* С. Ю. Дискретная математика (конспект лекций). — Таганрог: ТРУ, 2001.
5. *Лупанов* О. Б. Курс лекций по математической логике. — М.: МГУ, 2004.
6. *Москинова* Г. И. Дискретная математика. Математика для менеджера в примерах и упражнениях. — М.: Логос, 2007.
7. *Нефедов* В. Н., Осипова В. А. Курс дискретной математики. — М.: Изд-во МАИ, 1993.
8. *Новиков* П. С. Элементы математической логики. — М.: Наука, 1973.

Оглавление

Введение 3

1. АЛГЕБРА (ЛОГИКА) ВЫСКАЗЫВАНИЙ 5

1.1. Высказывания и операции над ними 5
Отрицание (логическая связка «не») 6
Логическое умножение (конъюнкция) 6
Логическое сложение (дизъюнкция) 7
Логическое следование (импликация) 7
Логическое тождество (эквиваленция) 8
Исключающее «или» (неравнозначность) 9

1.2. Формулы алгебры высказываний 10

1.3. Логические функции высказываний 13

1.4. Равносильность формул 16

1.5. Полные системы логических функций 19

1.6. Тавтологии. Выполнимые формулы 22

1.7. Нормальные формы для формул 26

1.8. Проблема разрешения и методы ее решения 31

1.9. Гипотезы и следствия в алгебре высказываний 34

1.10. Основные схемы логически правильных умозаключений ... 37

2. ЛОГИКА ПРЕДИКАТОВ 42

2.1. Предикаты 42

2.2. Кванторы 46

2.3. Формулы логики предикатов 49

2.4. Основные равносильности, содержащие кванторы 52

2.5. Предваренная нормальная форма 54

2.6. Тавтологии логики предикатов 58

3. ТЕОРИЯ АЛГОРИТМОВ 63

3.1. Машина Тьюринга 64
Универсальная кодировка машины Тьюринга 69

3.2. Алгоритмически неразрешимые проблемы 70

3.3. Рекурсивные функции. Тезис Чёрча 71
Операция суперпозиции 73
Операция примитивной рекурсии 73
Операция минимизации 74

Рекомендуемая литература 77

Учебное издание

Агарева Ольга Юрьевна
Селиванов Юрий Васильевич

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Под редакцией авторов

Технический редактор *Ю. В. Чуфистова*

Компьютерная верстка выполнена *Ю. В. Селивановым*

Подписано в печать 14.10.11 г. Формат 60 x 84 $^1/_{16}$
Печать на ризографе. Усл. п. л. 4,65. Уч.-изд. л. 3,65
Тираж 120 экз. Заказ № 124

Издательско-типографский центр МАТИ
109240, Москва, Берниковская наб., 14