Лекция 1

Алгебра логики: введение.

План:

1. Высказывания и логические связки.

2. Булевы функции и способы их задания: таблицы истинности, вектор значений, формулы.

3. Законы коммутативности, ассоциативности и дистрибутивности, приоритет операций.

4. Законы поглощения.

5. Равенство булевых функций (и булевых формул). Существенные и фиктивные переменные.

Ключевые слова: *высказывание, булева функция, конъюнкция, дизъюнкция, импликация, отрицание (инверсия), эквивалентность, XOR (исключающее или), коммутативность, ассоциативность, дистрибутивность, законы поглощения, существенная переменная, фиктивная переменная.*

Мы начинаем наш курс с основ логики, поскольку логика — это цемент для построения математических утверждений и доказательств. Хотя мы и начинаем фактически с изучения правил переписывания логических формул, которые можно выполнять не вдаваясь в суть самих высказываний, дальше эти правила будут использоваться для построения утверждений и доказательств.

Одной из основных целей нашего курса является прививание первокурсникам математической культуры. Математическая культура начинается с работы с формальными определениями, и это влечёт трудности. Формальные определения требуют язык теории множеств и часто при первом изучении формализм мешает содержательному пониманию вещей. Мы решаем эту проблему следующим образом. До введения теории множеств все определения будут неформальными, а после её изучения мы формализуем уже введённые определения на языке теории множеств. Однако после изучения теории множеств мы не откажемся от неформальных определений перед формальными. Определений в курсе достаточно много, поэтому они часто не выделяются в отдельные блоки, а просто по ходу текста выделяется ***определяемое понятие***. Можно было бы начать изложение с теории множеств, а не логики, но тогда у нас была бы другая проблема — пришлось бы вести изложение без объяснения, что считается доказательством. Надеюсь, мы выбрали меньшее из зол.

Итак, перейдём к изучении основ логики. Алгебра логики оперирует с высказываниями. ***Высказывание*** — это утверждение, которое либо истино, либо ложно. Истинность высказывания A будем обозначать «1», а ложность — «0». В роли A может выступать высказывание «за окном идёт дождь» или «число 7 делится на 6». Заметьте, что утверждение «число x делится на 6» не является высказыванием, в случае если число x не зафиксировано; тогда утверждение зависит от параметра и в зависимости от значения x может меняться истинность этого утверждения. Оговоримся, что утверждение, зависящее от параметра, не считается высказываением даже в случае, когда оно истинно при любом значении параметра: например, «число x делится на 1» — не высказывание.

С помощью логических связок из высказываний можно получать более сложные высказывания, такие как «A и B», «A или B», «не A». Вы знакомы с этими связками со школьной скамьи и знаете, что их называют «конъюнкция», «дизъюнкция» и «отрицание» («инверсия») и обозначают

$$A \wedge B, \qquad A \vee B, \qquad \neg A \text{ или } \bar{A}.$$

Что же такое логические связки? Это функции, которые зависят от набора переменных, принимающих значения 0 или 1 (от набора высказываний). Такие переменные называют ***булевыми переменными***, а функции — ***булевыми функциями***. Есть несколько стандартных способов задания булевой функции. Мы начнём с таблицы истинности.

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$A \oplus B$
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

Рис. 1.1. задание логических связок таблицами истинности.

Если функция зависит от k переменных, то первые k столбцов соответствуют переменным, а $k+1$-ый столбец — значению функции на соответсвующем наборе значений переменных (все наборы значений перечисляются в строках таблицы). В таблице на рис. 1.1 мы скомпоновали таблицы истинности для нескольких функций в одну таблицу.

Пожалуй, все логические связки имеют естественную интерпретацию, кроме импликации (почему её определяют именно так, мы обсудим позже):

Обозначение	Смысл	Название
$A \wedge B$	«A и B»	конъюнкция
$A \vee B$	«A или B»	дизъюнкция
$\neg A$	«не A»	отрицание
$A \to B$	«из A следует B»	импликация
$A \leftrightarrow B$	«A равносильно B»	эквивалентность
$A \oplus B$	«либо A, либо B»	XOR (исключающее или)

Наборы значений переменных принято перечислять в следующем порядке. Первым идёт набор из одних нулей, а дальше i-ый набор является двоичной записью числа $i-1$. Таким образом, всего в таблице истинности 2^k строк (именно столько чисел имеют двоичную запись длины k).

Благодаря стандартному порядку можно просто задать булеву функцию столбцом её значений:

$$f(x_1) = 10 = \neg x_1, \quad g(x_1, x_2) = 0001 = x_1 \wedge x_2, \quad h(x_1, x_2) = 0110 = x_1 \oplus x_2.$$

Говорят, что функция задана ***вектором значений***.

1.1 Задание булевых функций с помощью логических связок (формулами)

Следующий способ задания булевой функции — использовать уже заданные булевы функции для определения более сложных функций. С точки зрения логики, мы используем логические связки для построения более сложных высказываний, например

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

Формулу можно изобразить с помощью дерева, вычисления по которому (после присваивания значений переменным) идут снизу вверх; вообще говоря вычислять значение каждого узла дерева можно параллельно.

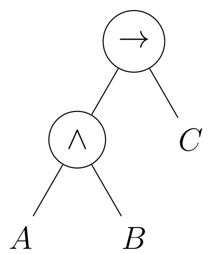

Рис. 1.2. Дерево для формулы $(A \wedge B) \to C$

Скобки используются для указания приоритета операций: в формуле выше следует сначала выполнить конъюнкцию, а затем импликацию. Для удобства записи используют следующие договорённости о приоритете операций:

1) отрицание $\neg$;

2) конъюнкция $\wedge$;

3) дизъюнкция $\vee$ и XOR $\oplus$;

4) импликация $\rightarrow$;

5) эквивалентность $\leftrightarrow$.

Согласно этим договорённостям можно восстановить скобки: самый высокий приоритет у отрицания, самый низкий — у импликации. В случае, если две операции имеют одинаковый приоритет, то считается, что сперва выполняется та, которая стоит левее. Но лучше поставить лишнюю пару скобок, дабы не было непонимания, что мы и будем делать.

Пример 1. Расставив скобки для формулы $x_1 \rightarrow \neg x_2 \wedge x_1 \vee x_2$ получим

$$(x_1 \rightarrow (((\neg x_2) \wedge x_1) \vee x_2)).$$

Расставив скобки, согласно приоритету можно преобразовать формулу в дерево, руководствуясь следующими правилами. Если формула состоит только из переменной, то дерево состоит только из одного узла — самой переменной. Если в формуле есть операции, то нужно ззять операцию из внешних скобках и сделать её вершиной дерева; построить деревья для левого и правого операндов и сделать их левым и правым детьми внешней операции соответственно.

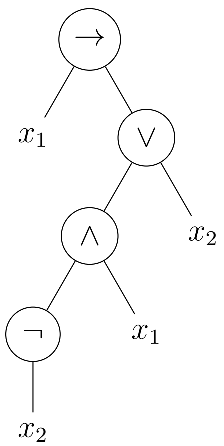

Рис. 1.3. Дерево для формулы из примера 1

1.2 Равенство булевых функций и формул. Фиктивные переменные

Один из самых коварных знаков в математике — это знак равенства. Справедливость формулы $x_1 \wedge x_2 = x_2 \wedge x_1$ не вызвает сомнений, ведь можно подставить в неё всевозможные наборы значений переменных и проверить, что значение левой части всегда равняется значению правой. Разумно было бы сказать, что левая и правая части равенства задают булевы функции, и равенство справедливо, если булевы функции совпадают, но тут есть одна загвоздка. Рассмотрим теперь равенство

$$x_1 \wedge x_2 = x_2 \wedge x_1 \wedge (x_2 \vee \neg x_3). \tag{1}$$

Это равенство также легко проверить подстановкой, но в его левой части записана булева функция от двух аргументов, а в правой — от трёх. От этого можно было бы отмахнуться, сказав, что можно считать, что левая часть равенства задаёт функцию от трёх аргументов, но глядя только на левую часть невозможно предсказать сколько переменных понадобится. Любой, кто хоть раз пробовал писать на C++ понимает, что эта проблема довольно деликатная.

Решение этой проблемы приводит к определению равенства булевых функций, но начнём мы с другого определения. ***Тавтология*** — это булева функция, которая возвращает 1 на любом наборе переменных. Булевы функции f и g ***равны***, если функция $h = (f \leftrightarrow g)$ — тавтология. Определение равенства булевых функций переносится на равенство формул. Если не оговорено противного, то мы считаем, что формула задаёт булеву функцию, зависящую только от переменных, встречающихся в этой формуле.

Обозначим через $g(x_1, x_2, x_3)$ булеву функцию, заданную правой частью формулы (1). Ясно, что значение g не зависит от переменной x_3. Благодаря приведённому определению равенства, мы получаем, что $g(x_1, x_2, x_3) = g(x_1, x_2, 0) = g(x_1, x_2, 1)$. В случае, если для булевой функции $f(x_1, \dots, x_n)$ справедливо равенство

$$f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) = f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n), \tag{2}$$

переменная x_i называется ***фиктивной*** (для функции f); в случае, если равенство (2) не выполняется для переменной x_i, то она называется ***существенной***.

Заметим, что для доказателсьства существенности переменной x_i достаточно найти в таблице истинности f два набора значений переменных, отличающиеся только значениями x_i, на которых f принимает разные значения. Для доказательства же фиктивности, необходимо проверить совпадание значений f на всех таких наборов: равенство (2) выполняется если и только если оно выполняется для всех наборов значений переменных $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$.

1.3 Логические тождества

Для логических связок выполняется довольно много законов (тождеств). Их справедливость можно проверить с помощью таблиц истинности.

Законы коммутативности:

$$x_1 \wedge x_2 = x_2 \wedge x_1, \qquad x_1 \vee x_2 = x_2 \vee x_1, \qquad x_1 \oplus x_2 = x_2 \oplus x_1.$$

Законы ассоциативности:

$$x_1 \wedge (x_2 \wedge x_3) = (x_1 \wedge x_2) \wedge x_3.$$

Законы дистрибутивности:

- $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
- $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
- $A \vee (B \to C) = (A \vee B) \to (A \vee C)$
- $A \to (B \wedge C) = (A \to B) \wedge (A \to C)$
- $A \to (B \vee C) = (A \to B) \vee (A \to C)$
- $A \to (B \to C) = (A \to B) \to (A \to C)$

Правила поглощения:

$$A \vee (A \wedge B) = A, \quad A \wedge (A \vee B) = A.$$

Другие важные свойства:

- $A \wedge A = A, \quad A \vee A = A$ (идемпотентность)
- $A \wedge \neg A = 0, \quad A \vee \neg A = 1$ (дополнение)
- $A \wedge 0 = 0, \quad A \vee 0 = A$ (универсальные границы)
- $A \wedge 1 = A, \quad A \vee 1 = 1$ (универсальные границы)
- $\neg(\neg A) = A$ (инволютивность)
- $\neg(A \wedge B) = \neg A \vee \neg B, \neg(A \vee B) = \neg A \wedge \neg B$ (законы Моргана)
- $A \to B = \neg A \vee B, \quad A \vee B = \neg A \to B$

Просто чертить таблицы истинности для проверки данных тождеств довольно времязатратно. Приведём пример доказательства тождества через таблицу истинности, не рисуя саму таблицу.

Пример 2. Докажем тождество $A \to B = \neg A \vee B$. И импликация и дизъюнкция имеют ровно один ноль среди значений: на наборах $(1, 0)$ и $(0, 0)$ соответственно. Таким образом, заменив в дизъюнкции $A \vee B$ переменную A на её отрицание, мы получим таблицу истинности для импликации.

Многие тождества среди других важных свойств получаются путём подстановки вместо переменной констант.

Тождества, которые мы рассматривали выше, иллюстрируют свойства логических связок. Логических законов за ними сходу не видно. Не такие, например, «закон двойного отрицания» $\neg\neg A = B$ и закон контрапозиции $A \to B = \neg B \to \neg A$.

С осмысленными логическими законами мы познакомимся чуть позже, а пока учимся утилитарно работать с алгеброй логики.