

Лекция 3. Алгебра логики

План лекции

1. Алгебра логики
2. Логические операции. Дизъюнкция, конъюнкция и отрицание
3. Логические основы компьютера
4. Законы алгебры логики
5. Логические элементы. Вентили

Алгебра логики (алгебра высказываний) — раздел математической логики, в котором изучаются логические операции над высказываниями.

Алгебра логики – это раздел математики, возникший в XIX веке благодаря усилиям английского математика Дж. Буля.

Законы и аппарат алгебры логики используются при проектировании различных частей компьютеров (память, процессор).

Логическое высказывание - любое повествовательное предложение, в отношении которого однозначно можно сказать, истинно оно или ложно.

Пример 1: предложение «6 - четное число» является высказыванием, т.к. оно истинное.

Существуют предложения, в которых для выяснения истинности или ложности требуются дополнительные сведения. Такие предложения являются высказывательными формами.

Высказывательная форма - повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную и становится высказыванием, когда все переменные замещаются своими значениями.

Пример 2: предложение «площадь поверхности Индийского океана равна 75 млн.км²» - и истинно (значение приближенное, приемлемо на практике) и ложно (указанное значение неточное)

Из логических высказываний составляются логические формулы.

Например, $X \rightarrow Y = X \vee Y$.

Логические операции. Дизъюнкция, конъюнкция и отрицание

Так как же связываются между собой простые логические высказывания, образуя сложные? В естественном языке мы используем различные союзы и другие части речи. Например, «и», «или», «либо», «не», «если», «то», «тогда». Пример сложных высказываний: «у него есть знания **и** навыки», «она приедет во вторник, **либо** в среду», «я буду играть **тогда, когда** сделаю уроки», «5 **не** равно 6». Как мы решаем, что нам сказали правду или нет? Как-то логически, даже где-то неосознанно, исходя из предыдущего жизненного опыта, мы понимаем, что правда при союзе «и» наступает в случае правдивости обоих простых высказываний. Стоит одному стать ложью и все сложное высказывание будет лживо. А вот, при связке «либо» должно быть правдой только одно простое высказывание, и тогда все выражение станет истинным.

Булева алгебра переложила этот жизненный опыт на аппарат математики, формализовала его, ввела жесткие правила получения однозначного результата. Союзы стали называться здесь логическими операторами.

Алгебра логики предусматривает множество логических операций. Однако три из них заслуживают особого внимания, т.к. с их помощью можно описать все остальные, и, следовательно, использовать меньше разнообразных устройств при конструировании схем. Такими операциями являются **конъюнкция (И)**, **дизъюнкция (ИЛИ)** и **отрицание (НЕ)**. Часто конъюнкцию обозначают &, дизъюнкцию - ||, а отрицание - чертой над переменной, обозначающей высказывание.

КОНЪЮНКЦИЯ

Обозначение Конъюнкция – логическое умножение (И) – and, &, ^.

Конъюнкция: соответствует союзу: «и», обозначается знаком ^, обозначает логическое умножение.

Конъюнкция двух логических высказываний истинна тогда и только тогда, когда оба высказывания истинны. Можно обобщить для любого количества переменных $A \wedge B \wedge C = 1$ если $A=1, B=1, C=1$.

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

ДИЗЪЮНКЦИЯ

Обозначение Дизъюнкция – логическое сложение (ИЛИ) – or, |, v.

Логическая операция соответствует союзу ИЛИ, обозначается знаком v, иначе называется **ЛОГИЧЕСКОЕ СЛОЖЕНИЕ**.

Дизъюнкция двух логических переменных ложна тогда и только тогда, когда оба высказывания ложны.

Это определение можно обобщить для любого количества логических переменных, объединенных дизъюнкцией.

$A \vee B \vee C = 0$, только если $A = 0, B = 0, C = 0$.

Таблица истинности дизъюнкции имеет следующий вид:

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

ИНВЕРСИЯ (Отрицание)

Обозначение Логическое отрицание (НЕ) – not, ¬.

Логическая операция соответствует частице не, обозначается ¬ или ¯ и является логическим отрицанием.

Инверсия логической переменной истинна, если переменная ложна и наоборот: инверсия ложна, если переменная истинна.

A	$\neg A$
1	0
0	1

высказывания у которых таблицы истинности совпадают называются равносильными.

Логические основы компьютера

В ЭВМ используются различные устройства, работу которых прекрасно описывает алгебра логики. К таким устройствам относятся группы переключателей, триггеры, сумматоры.

Кроме того, связь между булевой алгеброй и компьютерами лежит и в используемой в ЭВМ системе счисления. Как известно она двоичная. Поэтому в устройствах компьютера можно хранить и преобразовывать как числа, так и значения логических переменных.

Переключательные схемы

В ЭВМ применяются электрические схемы, состоящие из множества переключателей. Переключатель может находиться только в двух состояниях: замкнутым и разомкнутым. В первом случае – ток проходит, во втором – нет. Описывать работу таких схем очень удобно с помощью алгебры логики. В зависимости от положения переключателей можно получить или не получить сигналы на выходах.

Вентили, триггеры и сумматоры

Вентиль представляет собой логический элемент, который принимает одни двоичные значения и выдает другие в зависимости от своей реализации. Так, например, есть вентили, реализующие логическое умножение (конъюнкцию), сложение (дизъюнкцию) и отрицание.

Триггеры и сумматоры – это относительно сложные устройства, состоящие из более простых элементов – вентилях.

Триггер способен хранить один двоичный разряд, за счет того, что может находиться в двух устойчивых состояниях. В основном триггеры используются в регистрах процессора.

Сумматоры широко используются в арифметико-логических устройствах (АЛУ) процессора и выполняют суммирование двоичных разрядов.

Законы алгебры логики

Законы	для конъюнкции	для дизъюнкции
Идемпо- тентность (рефлексивнос- ть)	$X \wedge X = X$	$X \vee X = X$
Коммутатив- ность	$X \wedge Y = Y \wedge X$	$X \vee Y = Y \vee X$

<i>(переместительный)</i>		
<i>Ассоциативность (сочетательный)</i>	$(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$	$(X \vee Y) \vee Z = X \vee (Y \vee Z)$
<i>Дистрибутивность (распределительный)</i>	$X \wedge (Y \vee Z) =$ $= (X \wedge Y) \vee (X \wedge Z)$	$X \vee (Y \wedge Z) =$ $= (X \vee Y) \wedge (X \vee Z)$
<i>Закон (Правила) де Моргана</i>	$\overline{X \wedge Y} = \overline{X} \vee \overline{Y}$	$\overline{X \vee Y} = \overline{X} \wedge \overline{Y}$
<i>Закон поглощения</i>	$X \wedge (X \vee Y) = X$	$X \vee (X \wedge Y) = X$
<i>Операции с константами</i>	$X \wedge 1 = X$ $X \wedge 0 = 0$	$X \vee 0 = X$ $X \vee 1 = 1$
<i>Операции с инверсией</i>	$X \wedge \overline{X} = 0$	$X \vee \overline{X} = 1$
<i>Закон двойного отрицания</i>	$\overline{\overline{X}} = X$	

Логические элементы. Вентили

В основе построения компьютеров, а точнее аппаратного обеспечения, лежат так называемые **вентили**. Они представляют собой достаточно простые элементы, которые можно комбинировать между собой, создавая тем самым различные схемы. Одни схемы подходят для осуществления **арифметических операций**, а на основе других строят различную **память ЭВМ**.

Простейший вентиль представляет собой транзисторный инвертор, который преобразует низкое напряжение в высокое или наоборот (высокое в низкое). Это можно представить как преобразование логического нуля в логическую единицу или наоборот. Т.е. получаем вентиль **НЕ**.

Соединив пару транзисторов различным способом, получают вентили **ИЛИ-НЕ** и **И-НЕ**. Эти вентили принимают уже не один, а два и более входных сигнала. Выходной сигнал всегда один и зависит (выдает высокое или низкое напряжение) от входных сигналов. В случае вентиля **ИЛИ-НЕ** получить высокое напряжение (логическую единицу) можно только при условии низкого напряжения на всех входах. В случае вентиля **И-НЕ** все наоборот: логическая единица получается, если все входные сигналы будут нулевыми. Как видно, это обратно таким привычным логическим операциям как **И** и **ИЛИ**. Однако обычно используются вентили **И-НЕ** и **ИЛИ-НЕ**, т.к. их реализация проще: **И-НЕ** и **ИЛИ-НЕ** реализуются двумя транзисторами, тогда как логические **И** и **ИЛИ** тремя.

Выходной сигнал вентиля можно выражать как функцию от входных.

Транзистору требуется очень мало времени для переключения из одного состояния в другое (время переключения оценивается в наносекундах). И в этом одно из существенных преимуществ схем, построенных на их основе.

Арифметико-логическое устройство процессора (АЛУ) обязательно содержит в своем составе такие элементы как **сумматоры**. Эти схемы позволяют складывать двоичные числа.

Как происходит сложение? Допустим, требуется сложить двоичные числа 1001 и 0011. Сначала складываем младшие разряды (последние цифры): $1+1=10$. Т.е. в младшем разряде будет 0, а единица – это перенос в старший разряд. Далее: $0 + 1 + 1$ (от переноса) = 10, т.е. в данном разряде снова запишется 0, а единица уйдет в старший разряд. На третьем шаге: $0 + 0 + 1$ (от переноса) = 1. В итоге сумма равна 1100.

Полусумматор

Теперь не будем обращать внимание на перенос из предыдущего разряда и рассмотрим только, как формируется сумма текущего разряда. Если были даны две единицы или два нуля, то сумма текущего разряда равна 0. Если одно из двух слагаемых равно единице, то сумма равна единицы. Получить такие результаты можно при использовании вентиля **ИСКЛЮЧАЮЩЕГО ИЛИ**.

Перенос единицы в следующий разряд происходит, если два слагаемых равны единице. И это реализуемо вентилем **И**.

Тогда сложение в пределах одного разряда (без учета возможной пришедшей единицы из младшего разряда) можно реализовать изображенной ниже схемой, которая называется **полусумматором**. У полусумматора два входа (для слагаемых) и два выхода (для суммы и переноса). На схеме изображен полусумматор, состоящий из вентилях **ИСКЛЮЧАЮЩЕЕ ИЛИ** и **И**.

Память (устройство, предназначенное для хранения данных и команд) является важной частью компьютера. Можно сказать, что она его и определяет: если вычислительное устройство не имеет памяти, то оно уже не компьютер.

Элементарной единицей компьютерной памяти является бит. Поэтому требуется устройство, способное находиться в двух состояниях, т.е. хранить единицу или ноль. Также это устройство должно уметь быстро переключаться из одного состояния в другое под внешним воздействием, что дает возможность изменять информацию. Ну и наконец, устройство должно позволять определять его состояние, т.е. предоставлять во вне информацию о своем состоянии.

Устройством, способным запоминать, хранить и позволяющим считывать информацию, является **триггер**. Он был изобретен в начале XX века Бонч-Бруевичем.

Разнообразие триггеров весьма велико. Наиболее простой из них так называемый RS-триггер, который собирается из двух вентилях. Обычно используют вентили ИЛИ-НЕ или И-НЕ.

RS-триггер на вентилях ИЛИ-НЕ

RS-триггер «запоминает», на какой его вход подавался сигнал, соответствующий единице, в последний раз. Если сигнал был подан на S-вход, то триггер на выходе постоянно «сообщает», что хранит единицу. Если сигнал, соответствующий единице, подан на R-вход, то триггер на выходе имеет 0. Не смотря на то, что триггер имеет два выхода, имеется в виду выход Q. (Q с чертой всегда имеет противоположное Q значение.)

Другими словами, вход S (set) отвечает за установку триггера в 1, а вход R (reset) – за установку триггера в 0. Установка производится сигналом, с высоким напряжением (соответствует единице). Просто все зависит от того, на какой вход он подается.

Большую часть времени на входы подается сигнал равный 0 (низкое напряжение). При этом триггер сохраняет свое прежнее состояние.

Возможны следующие ситуации:

$Q = 1$, сигнал подан на S, следовательно, Q не меняется.

$Q = 0$, сигнал подан на S, следовательно, $Q = 1$.

$Q = 1$, сигнал подан на R, следовательно, $Q = 0$.

$Q = 0$, сигнал подан на R, следовательно, Q не меняется.

Ситуация, при которой на оба входа подаются единичные сигналы, недопустима.

Как триггер сохраняет состояние? Допустим, триггер выдает на выходе Q логический 0. Тогда судя по схеме, этот 0 возвращается также и в верхний вентиль, где инвертируется (получается 1) и уже в этом виде передается нижнему вентилю. Тот в свою очередь снова инвертирует сигнал (получается 0), который и имеется на выходе Q. Состояние триггера сохраняется, он хранит 0.

Теперь, допустим, был подан единичный сигнал на вход S. Теперь в верхний вентиль входят два сигнала: 1 от S и 0 от Q. Поскольку вентиль вида ИЛИ-НЕ, то на выходе из него получается 0. Ноль идет на нижний вентиль, там инвертируется (получается 1). Сигнал на выходе Q становится соответствующим 1.