

# Лекция 2

## Основы формальной логики

# Логика как наука

- Логика - это наука, которая устанавливает правила для упорядоченного мышления, в то время как критическое мышление – это мышление с применением этих правил.
- В процессе развития науки о мышлении был предложен, проверен и найден ряд стандартов или критериев, которые оказались надежными руководствами для обоснованного суждения о достижении истины. Пример: статистический анализ, т.е. обобщенные выводы по исследованию некоей выборки объектов.

# Критическое мышление

- это более, чем просто мышление какими-то субъективными критериями. Скорее, это мышление на основе критериев, которые были проверены и признаны надежными руководящими принципами для правильного суждения и достижения правды по рассматриваемому вопросу.

Принимать решения на основе «бросания костей» - пример критериального, но не критического мышления.

# Машина Тьюринга

- Простейшая модель машинного «мышления». Все вычислительные проблемы можно разбить на очень простой язык, цифровой язык: 0 и 1.
- Под эту концепцию попадает любое действие, описываемое двумя состояниями: «включено» или «выключено», «истинно» или «ложно», «внутри» или «снаружи»; «да» и «нет».

Пример: выключатель света.

# СВЯЗЬ ЛОГИКИ И ИНФОРМАТИКИ

- Бинарная модель, является основным языком, который понимают микропроцессоры.
- Люди пишут код на гораздо более сложных языках программирования, которые компьютеры компилируют в то, что они могут обрабатывать.
- Компьютер работает с набором команд или инструкций, обрабатывая задачи предсказуемым образом.
- Алгоритмы - не конкретный расчет, а метод, который используется при выполнении расчета (рецепт)

# Программа - рецепт

- Программы пишутся в виде набора процедур, которые определяют, что компьютер должен делать в различных обстоятельствах.

Пример: сложение 2х чисел может быть написан на нескольких языках программирования, но складывать различные числа (те, которые введет пользователь)

Логика как программа - предоставляет набор рецептов, которые мы можем использовать для организации мышления.

# Пример (псевдокод)

```
for (x=0; x < 7; x++){  
    If x is Even Then  
        print "Eat pasta for dinner"  
    Else  
        print "Eat salad for dinner"  
}
```

# Построение логического выражения: «что есть на ужин»

- Предпосылка 1: День может быть четным ИЛИ нечетным;
- Предпосылка 2: Сегодня день, который не является четным;
- Вывод: Сегодня – нечетный день.

Недостаточное выражение

# Дополнение - условие

Силлогизм для четных дней:

Предпосылка 1: **ЕСЛИ** день четный, **ТО** мы едим макароны

Предпосылка 2: день четный;

Вывод: поэтому мы едим макароны.

Силлогизм для нечетных дней:

Предпосылка 1: Если день нечетный, то мы едим салат;

Предпосылка 2: день нечетный;

Вывод: поэтому мы едим салат.

# Полное логическое выражение

Дано: текущий день недели

Предпосылка 1: либо день нечетный, либо день четный

Предпосылка 2: если день четный, то мы едим макароны

Предпосылка 3: если день нечетный, то мы едим салат

Предпосылка 4: Это не тот случай, когда день [нечетный / четный]

Предпосылка 5: день [четный / нечетный]

Вывод: поэтому мы едим [макароны / салат]

# Логика - информатика

Если вы сможете научиться применять логические рамки к вашему мышлению в повседневной жизни, это поможет вам продумать, как писать лучшие компьютерные программы - и наоборот!

Основатель Apple Стив Джобс сказал:

**«Я думаю, что все должны научиться программировать компьютер, потому что он учит вас думать».**

Теперь вы можете начать понимать, почему!

# Логика – система правил

- Когда мы изучаем логику, мы изучаем правила упорядоченного мышления. Эти правила называются **силлогизмы**.
- Другое понятие, которое изучает логика, это понятие «**истина**».
- **Силлогизмы** – это правила, которые мы используем для построения умозаключений, определяющих какие из высказанных идей являются истинными, а какие ложными.

# Истинное и ложное

- Не существует правильных или неправильных идей!
- Существуют истинные и ложные утверждения. У каждого утверждения значение истинности не зависит от субъективной оценки.

Пример:  $2 * 2 = 5$  ВСЕГДА!!!!

# Умозаключение

- Логика выстраивает упорядоченное мышление, базируясь на умозаключениях
- Умозаключение имеет конкретную цель и формальную структуру.
- Умозаключение - это структурированный набор предложений, предназначенных для убеждения кого-либо в чем-либо. Структура очень важна.

# Структура умозаключения

Любое умозаключение состоит из трех частей:

- Предпосылки: причины или аргументы;
- Вывод: утверждение, подтвержденное предпосылкой;
- Логическое отношение: логическая связь между предпосылкой и выводом.

Пример не умозаключения:

- Предпосылка 1: Иван – студент первого курса;
- Предпосылка 2: трава зеленого цвета;
- Вывод: поэтому солнце встает утром.

# Пример умозаключения

Пример:

Предпосылка 1: Все овощи - растения;

Предпосылка 2: Помидор - это овощ;

Вывод: поэтому этот помидор является растением.

Если вы хотя бы предполагаете, что эти два условия верны, то вы не можете не согласиться с выводом.

Вывод верный, основанный на правде предпосылок.

# Высказывания - предпосылки

- Предпосылками могут быть только утвердительные предложения
- Высказывания, которые декларируют только одну истину - **простыми высказываниями**.

Пример: «Петя – студент.»

- Высказывания, которые утверждают несколько истин – **составные высказывания**

Пример: «Петя – студент, и Маша тоже студентка.»

«Не все студенты потока присутствуют на лекции  
**или** мне так показалось.»

# Сложность определения ИСТИННОСТИ

- неоднозначностью слов в высказывании
- некорректное построением высказывания
- сложность структуры самого высказывания
- невозможность определения ложности либо истинности в данный момент в силу неизвестности или недоступности необходимой информации.
- высказывания могут быть верными в одно время и ложными в другое, а некоторые всегда верны и всегда ложны.

# Примеры истинности

- «Квадрат – это круг» – всегда ложное высказывание;
- «Гора Эльбрус расположена недалеко от поселка Терскол» - всегда истинное высказывание;
- «В Москве идет снег» - в какие-то моменты истинное, а в какие-то нет.

# Символы и логические операции

- В математике мы можем использовать символы в уравнениях и формулах.  $x+1=y$
- В логике можно символизировать суждения в дедуктивных умозаключениях и только с помощью символической формы мы можем лучше увидеть структуру этого доказательства

# Пример записи символами

Логическое выражение

Все люди рациональны

Все рациональные вещи сознательны

Поэтому все люди сознательны

Символьная запись

Все  $H$  являются  $R$

Все  $R$  являются  $C$

Поэтому все  $H$  являются  $C$

Можно использовать для любых  $H, R, C$

Все студенты являются образованными

Все образованные являются вежливыми

Поэтому все студенты являются вежливыми

## Символьная запись

Все Н являются R

Все R являются С

Поэтому все Н являются С

Н - «яблоко», R - «спелый» и С - «сумасшедший»

Все яблоки спелые

Все спелые вещи сумасшедшие

Поэтому все яблоки сумасшедшие

Умозаключение бессмысленно, но он сохраняет ту же структуру.

Используя символы, мы можем отделить структуру умозаключения от его содержания

# Пример сложного высказывания

« Современные компьютерные системы - это машины Тьюринга, которые могут обрабатывать триллионы операций в секунду, хотя основные операции чрезвычайно просты: они, по сути, переходят из состояний от 1 до 4 или включаются, полу- включаются, частично включаются и выключаются.»

Сколько утверждений правды декларируется в этом предложении?

# Сложное состоит из простого

- Современные компьютерные системы - это машины Тьюринга. - Истина
- И современные компьютеры могут обрабатывать триллионы операций в секунду. - Истина
- НО основные операции компьютерных систем предельно просты. - Истина
- И современные компьютерные системы переходят из состояний от 1 до 4: включен, полу-включен, частично включен и выключен. – Ложь

# Составные операторы

- Все составные операторы, оперирующие с двумя простыми высказываниями, называются «функционально-истинными» составными операторами.
- значение истинности всего утверждения определяется значением истинности отдельных простых утверждений, которые его составляют, и функцией оператора в этих утверждениях

Аналог функции:  $y = F(x, y)$

# Составные операторы - программы

- простые выражения - входные данные
- оператор - функция

Выход всей функции определяется значением входов и конкретной операцией функции.

Пример:

```
Function PrintCaps (string A)
{
    A = AllCaps(A);
    Print A}
```

# Основные понятия логики

- **Высказывание** - это повествовательное предложение, про которое можно определенно сказать истинно оно или ложно (истина (логическая 1), ложь (логический 0)).
- **Логические операции (операторы)** - мыслительные действия, результатом которых является изменение содержания или объема понятий, а также образование новых понятий.
- **Логическое выражение** - устное утверждение или запись, в которое, наряду с постоянными величинами, обязательно входят переменные величины (объекты). В зависимости от значений этих переменных величин (объектов) логическое выражение может принимать одно из двух возможных значений: истина (логическая 1) или ложь (логический 0).
- **Сложное логическое выражение** - логическое выражение, состоящее из одного или нескольких простых логических выражений (или сложных логических выражений), соединенных с помощью логических операций.

# Основные логические операторы

# Оператор отрицания

Если простое утверждение истинно, размещение перед ним (или над ним) символа отрицания делает составное утверждение ложным. Если простое утверждение ложно, но перед ним ставится отрицание, что делает сложное утверждение истинным.

Символ:  $\sim$  «высказывание» (тильда) или (верхнее подчеркивание)

Пример:

Высказывание  $A$  – «Петр на лекции»

Отрицание -  $\sim A$  – «Это не тот случай, когда Петр на лекции»

Неверно: «Петр не на лекции»

# Конъюнкция (логическое умножение)

это сложное логическое выражение, которое считается истинным в том и только том случае, когда оба простых выражения являются истинными, во всех остальных случаях данное сложное выражение ложно.

Символ: «высказывание» & «высказывание»  
«высказывание» AND «высказывание»  
«высказывание» И «высказывание»

Пример:

Высказывание 1: **A** –«Петр на лекции»

Высказывание 2: **B** –«Маша в кафе»

Конъюнкция:  $A \& B = A \cdot B =$  Петр на лекции и Маша в кафе.

# Дизъюнкция (логическое сложение)

это сложное логическое выражение, которое истинно, если хотя бы одно из простых логических выражений истинно и ложно тогда и только тогда, когда оба простых логических выражения ложны.

Символ: «высказывание»  $\vee$  «высказывание»  
«высказывание» **OR** «высказывание»  
«высказывание» **ИЛИ** «высказывание»

Пример:

Высказывание 1: **A** –«Петр на лекции»

Высказывание 2: **B** –«Маша в кафе»

Дизъюнкция:  $A \vee B = A + B =$  Петр на лекции или Маша в кафе.

НЕ ПУТАТЬ с «исключающим ИЛИ» **XOR**, которое принимает истинное значение только если одно высказывание имеет истинность противоположную второму (т.е. если первое истинно, то второе ложное и наоборот).

# Импликация (логическое условие или следование)

это сложное логическое выражение, которое истинно во всех случаях, кроме как из истины следует ложь. То есть данная логическая операция связывает два простых логических выражения, из которых первое является условием (А), а второе (В) является следствием. Другими словами - это составной оператор, который устанавливает условие для чего-то, что будет истинным. Если условие выполнено, утверждение верно.

Символ:  $A \rightarrow B$ .

Пример:

Высказывание<sub>1</sub>: А – «**Четырехугольник - квадрат**»

Высказывание<sub>2</sub>: В – «**Четырехугольник - ромб**»

Импликация:  $A \rightarrow B =$  «**ЕСЛИ** четырехугольник – квадрат, **ТО** четырехугольник - ромб.»

# Необходимые и достаточные условия

- **Необходимое условие** – это то условие, которое обязательно должно быть правдой для того, что бы все выражение следования оказалось правдой. Без выполнения этого условия истинность высказывания невозможна. Т.е. если  $A \rightarrow B$  истинно, то  $B$  – необходимое условие. Быть ромбом – необходимое условие, что бы быть квадратом, без этого не получится квадрат.
- Однако далеко не каждый ромб обязан быть квадратом, поэтому обратная импликация может оказаться неверной.  $B \rightarrow A =$  «ЕСЛИ четырехугольник – ромб, ТО четырехугольник - квадрат.» - ложно!  $A$  является **достаточным условием** для  $B$ . Быть квадратом – достаточное основание для того, что бы быть ромбом.

# Ошибка(ложь) в работе условного оператора

Ложь = возникновению ошибки.

- Пример: система безопасности, которая должна срабатывать в случае, если кто-то вносит в экзаменационную аудиторию включенный мобильник. В этом случае наша система безопасности озвучивает случайным образом пункт из реально существующих (истинных) предупреждений Разработчик системы должен заботиться не только об истинности произошедшего события (пронесен или не пронесен мобильник), но и об истинности озвученного правила (существует такое или нет). Но что происходит во всех других случаях, разработчик не продумывает, потому что его задача выглядит как условие:

**«ЕСЛИ на экзамен пронесли мобильник, ТО озвучить существующее правило».**

- **Ситуация 1** (условие ложное, следствие истинно): Студент, не идущий на экзамен, прошел мимо системы безопасности, и система озвучила «использование включенных мобильных телефонов на экзамене запрещено». Система озвучила настоящее правило, но это не имеет значения, поскольку студент и не собирался на экзамен. Т.е. на самом деле система может озвучивать все что угодно, в случае, если условие не выполнено, при этом корректность ее работы не противоречит условию, заданному в ТЗ.
- **Ситуация 2** (условие ложное, следствие ложное): Студент вошел на экзамен с бананом в руке, а система безопасности выдала «с бананами на экзамене присутствовать запрещено». Система выдала ложный пункт правил, но это не имеет значения, поскольку студент не принес мобильного телефона. Звучит забавно, но не противоречит ТЗ. Система работает именно так, как и ожидал разработчик.
- **Ситуация 3** (Условие верно, следствие верно): Студент входит в аудиторию с включенным мобильным телефоном и слышит предупреждение «выключите мобильный телефон». Наше условие выполнено, система сработала как надо. Значит система работает?
- **Ситуация 4** (условие верно, следствие неверно): Студент входит в аудиторию с включенным мобильным телефоном и слышит предупреждение «с бананами на экзамене присутствовать запрещено». Очевидно, что именно эта ситуация иллюстрирует, что система не работает. Условие ЕСЛИ..., ТО... привело к ошибочному результату.

# Пример (псевдокод)

Function WhatsTheTruthValue (a,b)

```
{  
  if (a > 0){  
    var t = 3*b;  
    if (t === 0){  
      print "function crashed";  
      return;  
    }  
  }  
  print "function works"; }  
}
```

Вводимые переменные	Результат
a=1; b=1	«функция работает»
a=1; b=0	«сбой функции»
a=0; b=1	«функция работает»
a=0; b=0	«функция работает»

# Сложные составные операторы

Пример:  $p \rightarrow q \ \& \ r \vee s \ \& \ t$  что это за утверждение?

$$p \rightarrow (q \ \& \ ((r \vee s) \ \& \ t))$$
$$p \rightarrow (q \ \& \ [(r \vee s) \ \& \ t])$$

Пусть  $p = \text{true}$ ;  $q = \text{false}$ ;  $r = \text{false}$ ;  $s = \text{true}$  and  $t = \text{true}$

- $r = \text{false}$ ,  $s = \text{true}$ , значит  $(r \vee s) = \text{true}$ ;
- $(r \vee s) = \text{true}$ ,  $t = \text{true}$  значит  $((r \vee s) \ \& \ t) = \text{true}$ ;
- $q = \text{false}$ ,  $((r \vee s) \ \& \ t) = \text{true}$ , значит  $(q \ \& \ [(r \vee s) \ \& \ t]) = \text{false}$ ;
- $p = \text{true}$ ,  $(q \ \& \ [(r \vee s) \ \& \ t]) = \text{false}$ , значит все выражение будет иметь ложное значение.

# Таблицы истинности

таблицы, которые позволяют определить значение истинности, в случае применения одного и того же оператора к выражениям, обладающим разными значениями истинности.

Оператор отрицания:

A	$\bar{A}$
0	1
1	0

Конъюнкция (И):

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция (ИЛИ):

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

# Импликация (условие)

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

# Таблица истинности для СЛОЖНОГО УСЛОВИЯ

●  $p \rightarrow (q \& [(r \vee s) \& t])$

p	q	r	s	t	p	□	(q	&	[(r	∨	s)	&	t])
1	0	0	1	1	1		0		0		1		1
					1		0			1			1
					1		0					1	
					1			0					
							0						