Математическая логика

Михайлов Максим

15 февраля 2021 г.

Оглавление

Лекция 1

12 февраля

0. Мотивация

0.1. Математикам

Аксиома 1 (Архимеда). Для любого $k > 0$ найдётся n, такое что $kn > 1$.

Под эту аксиому не подходят бесконечно малые числа и это является проблемой. Например, $\lim\limits_{x \to +\infty} \frac{1}{x} = 0 = \lim\limits_{x \to +\infty} \frac{1}{x^2}$, но мы хотим уметь различать эти два числа. Ньютон предложил идею бесконечно малых чисел, откуда пошли последовательности. Возникает вопрос — что такое последовательность и что такое число?

Общепринятое определение целых чисел $\mathbb{N}$ происходит из теории множеств. Однако эта теория содержит в себе множество фундаментальных парадоксов, от которых нельзя избавиться.

Возникает вопрос — а что такое множество? Посмотрим на некоторое множество $A = \{x \mid x \notin x\}$. Содержит ли оно себя, $A \in A$? На этот вопрос нельзя ответить, это называется парадокс Рассела. Есть простой способ его разрешить — запретить ставить такой вопрос. Нет вопроса — нет парадокса. Существование такого парадокса ставит под вопрос существование любого множества — а существует ли $\mathbb{N}$? Может быть его существование парадоксально, просто мы не нашли этот парадокс. Пришло чуть более умное решение парадокса — запретим множества, содержащие себя. Таким образом вывели аксиоматику теории множеств *(Цермело — Френкеля)*.

Пример. Рассмотрим множество всех чисел, которые можно задать в ≤ 1000 слов русского языка. Фраза “наименьшее число, которое нельзя задать в ≤ 1000 слов” содержит ≤ 1000 слов, т.е. такое число принадлежит искомому множеству — парадокс.

Возникает идея — человеческий язык порождает парадоксы, поэтому нужно задать новый язык, который их не порождает. Этот язык и является математической логикой.

0.2. Программистам

Математическая логика применяется в двух областях *(для программистов)*:

1. Языки программирования
2. Формальные доказательства

Для языков программирования матлогика применима как теория типов *(переменных)*.

Формальные доказательства нужны например для smart-контрактов, где корректность программы критически важна, т.к. если в нём есть ошибка, у вас злоумышленник заберет все деньги, а вы не сможете этот контракт откатить.

1. Исчисление высказываний

1.1. Язык

Определение. Язык содержит в себе:

1. **Пропозициональные переменные**

   A_i' — большая буква начала латинского алфавита, возможно с индексом и/или штрихом.

2. **Связки**

   Пусть α, β — высказывания. Тогда $(\alpha \rightarrow \beta), (\alpha \mathbin{\&} \beta), (\alpha \vee \beta), (\neg\alpha)$ — высказывания.

   α, β называются **метапеременными**.

Примечание. Математическая логика алгеброподобна *(а не анализоподобна)*, т.к. в ней много определений и мало доказательств.

1.2. Метаязык и предметный язык

У нас есть два различных языка — **предметный язык** и **метаязык**. Метаязык — русский, предметный язык мы определили выше.

Пример. $\alpha \rightarrow \beta$ — метавыражение; $A \rightarrow (A \rightarrow A)$ — предметное выражение.

Обозначение. Метапеременные обозначаются различными способами в зависимости от того, что они обозначают:

- Буквы греческого алфавита $(\alpha, \beta, \gamma, \ldots, \varphi, \psi)$ — выражения
- Заглавные буквы конца латинского алфавита (X, Y, Z) — произвольные переменные

Пример. $X \to Y \Rightarrow A \to B$ — подстановка переменных. Этот синтаксис не формален, мы будем записывать так:

$$(X \to Y)[X := A, Y := B] \equiv A \to B$$

Соглашение. символы логических операций не пишутся в метаязыке.

Пример.

$$(\alpha \to (A \to X))[\alpha := A, X := B] \equiv A \to (A \to B)$$
$$(\alpha \to (A \to X))[\alpha := (A \to P), X := B] \equiv (A \to P) \to (A \to B)$$

1.3. Сокращения записи

- $\vee, \&, \neg$ — скобки слева направо *(лево-ассоциативные операции) (не коммутативные)*
- $\to$ — правоассоциативная.

Примечание. Здесь операторы записаны в порядке их приоритета

Пример. Расставим скобки в следующем выражении:

$$A \to B \,\&\, C \to D$$

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

1.4. Теория моделей

Модель состоит из:

Обозначение.

- P — некоторое множество предметных переменных
- τ — множество высказываний предметного языка
- V — множество истинных значений. Классическое — $\{\text{И}, \text{Л}\}$
- $[\![\ ]\!] : \tau \to V$ — оценка высказывания *(высказывание ставится в скобки).*

1. $[\![x]\!] : P \to V$ — задается при оценке.
2. $[\![\alpha \star \beta]\!] = [\![\alpha]\!] \star [\![\beta]\!]$, где $\star$ есть логическая операция ($\vee, \&, \neg, \to$), а $\star$ определено естественным образом как элемент метаязыка.

1.5. Теория доказательств

Определение. **Схема высказывания** — строка, соответствующая определению высказывания + метапеременные.

Пример.

$$(\alpha \to (\beta \to (A \to \alpha)))$$

10 схем аксиом:

1. $\alpha \to \beta \to \alpha$
2. $(\alpha \to \beta) \to (\alpha \to \beta \to \gamma) \to (\alpha \to \gamma)$
3. $\alpha \to \beta \to \alpha \,\&\, \beta$
4. $\alpha \,\&\, \beta \to \alpha$
5. $\alpha \,\&\, \beta \to \beta$
6. $\alpha \to \alpha \vee \beta$
7. $\beta \to \alpha \vee \beta$
8. $(\alpha \to \gamma) \to (\beta \to \gamma) \to (\alpha \vee \beta \to \gamma)$
9. $(\alpha \to \beta) \to (\alpha \to \neg\beta) \to \neg\alpha$
10. $\neg\neg\alpha \to \alpha$

1.6. Правило Modus Ponens и доказательство

Определение. **Доказательство *(вывод)*** есть конечная последовательность высказываний $\alpha_1 \dots \alpha_n$, где α_i — либо аксиома, либо $\exists k, l < i : \alpha_k \equiv \alpha_l \to \alpha_i$ *(правило Modus Ponens)*

Пример. $\vdash A \to A$

1.	$A \to A \to A$	сх. акс. 1
2.	$A \to (A \to A) \to A$	сх. акс. 1
3.	$(A \to (A \to A)) \to (A \to (A \to A) \to A) \to (A \to A)$	сх. акс. 2
4.	$(A \to (A \to A) \to A) \to (A \to A)$	M.P. 1, 3
5.	$A \to A$	M.P. 2, 4

Определение. Доказательство $\alpha_1 \dots \alpha_n$ доказывает выражение β, если $\alpha_n \equiv \beta$