

# Математическая логика

mk.cs.msu.ru → Лекционные курсы → Математическая логика (318, 319/2, 241, 242)

## Блок 40

Формальная арифметика  
Теорема Гёделя о неполноте

Лектор:  
**Подымов Владислав Васильевич**

E-mail:  
**valdus@yandex.ru**

# Вступление

Основные понятия и свойства которые обсудили (на данный момент) для аксиоматических теорий:

- ▶ *непротиворечивость*, *полнота* и *разрешимость* теорий и взаимосвязь этих свойств
- ▶ *выразимость* понятий в интерпретации (а значит, и в полной теории этой интерпретации)

В примерах возникало немало определений понятий в арифметических интерпретациях, но не говорилось ничего про границу между тем, что можно выразить и что нельзя, и про конкретные “хорошие” арифметические теории и их свойства

# Формальная арифметика

Напоминание: **формальная арифметика** — это теория интерпретации  $Ar[\mathbb{N}_0; 0; +, \cdot, s; =]$

Самая простая формальная арифметика, какую только можно придумать:  $\emptyset$

В примерах возникала формальная арифметика и получше:

$$\left\{ \begin{array}{l} \forall x (x + 0 = x) \quad \forall x \forall y (x + s(y) = s(x + y)) \\ \forall x (x = x) \quad \forall x \forall y ((x = y) \rightarrow (y = x)) \\ \forall x \forall y \forall z ((x = y) \& (y = z) \rightarrow (x = z)) \\ \forall x \forall y ((x = y) \rightarrow (s(x) = s(y))) \\ \forall x \forall y \forall u \forall v ((x = u) \& (y = v) \rightarrow (x + y = u + v)) \end{array} \right\}$$

В аксиомах этой арифметики не содержится умножение (что разумно: в соответствующем примере использовалась сигнатура без умножения)

Кроме того, эта теория неполна (даже если удалить умножение из сигнатуры)

# Формальная арифметика

Более нетривиальный пример: арифметика Пеано

$$\left\{ \begin{array}{l} \forall x (x + 0 = x) \quad \forall x \forall y (x + s(y) = s(x + y)) \\ \forall x (x \cdot 0 = 0) \quad \forall x \forall y (x \cdot s(y) = x \cdot y + x) \\ \forall x (x = x) \quad \forall x \forall y ((x = y) \rightarrow (y = x)) \\ \quad \forall x \forall y \forall z ((x = y) \& (y = z) \rightarrow (x = z)) \\ \forall x \forall y ((x = y) \rightarrow (s(x) = s(y))) \quad \forall x \forall y ((s(x) = s(y)) \rightarrow (x = y)) \\ \forall x \neg(0 = s(x)) \\ \varphi \{x/0\} \& \forall x (\varphi \rightarrow \varphi \{x/s(x)\}) \rightarrow \forall x \varphi \end{array} \right.$$

В последней строке записана *схема аксиом* индукции с параметром  $\varphi$  — произвольной формулой с единственной свободной переменной  $x$

А насколько хороша арифметика Пеано,  
и можно ли придумать арифметику ещё лучше?

# Вычислимость и перечислимость

Ненадолго отвлечёмся от аксиоматических теорий и обратим внимание на множество  $\mathbb{N}_0$

Множество  $X$ ,  $X \subseteq \mathbb{N}_0$ , называется (рекурсивно) **перечислимым**, если существует алгоритм без входных данных, не завершающий своё выполнение и **перечисляющий** элементы множества  $X$ :

- ▶ В качестве отдельного действия алгоритмом может быть **выдано** (перечислено) вычисленное ранее число
- ▶ Алгоритмом выдаются только элементы множества  $X$
- ▶ Каждый элемент  $X$  хотя бы раз выдаётся алгоритмом

Чтобы можно было оценить, насколько строгим является ограничение перечислимости множества чисел, свяжем его с другими аналогичными алгоритмическими понятиями

# Вычислимость и перечислимость

Частично определённая функция  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$  называется **вычислимой**, если существует алгоритм, реализующий эту функцию

$f(x) = *$  — так будем обозначать тот факт, что значение  $f(x)$  не определено

Множество  $X$ ,  $X \subseteq \mathbb{N}_0$ , называется **разрешимым**, если вычислима характеристическая функция  $\chi_X$  этого множества:

$$\chi_X(x) = \begin{cases} 1, & \text{если } x \in X; \\ 0 & \text{иначе} \end{cases}$$

**Утверждение.** Любое разрешимое множество перечислимо

**Доказательство.** Если множество  $X$  разрешимо, то существует алгоритм  $\mathcal{A}$ , реализующий  $\chi_X$ , и перечислить элементы  $X$  можно так:

- ▶ По очереди вычисляются значения  $\overline{\mathcal{A}}(0), \overline{\mathcal{A}}(1), \overline{\mathcal{A}}(2), \dots$
- ▶ Если вычислено значение  $\overline{\mathcal{A}}(x) = 1$ , то выдаётся число  $x$  ▼

# Вычислимость и перечислимость

Множество  $X$ ,  $X \subseteq \mathbb{N}_0$ , называется **полуразрешимым**, если вычислима **полухарактеристическая функция**  $\chi_X^*$  этого множества:

$$\chi_X^*(x) = \begin{cases} 1, & \text{если } x \in X; \\ * & \text{иначе} \end{cases}$$

**Утверждение.** Множество перечислимо  $\Leftrightarrow$  оно полуразрешимо

**Доказательство.**

( $\Rightarrow$ ): Если алгоритм  $\mathcal{A}$  перечисляет множество  $X$ , то функцию  $\chi_X^*$  для входа  $x$  вычисляет такой алгоритм:

- ▶ Запустим алгоритм  $\mathcal{A}$
- ▶ Если на очередном шаге выполнения  $\mathcal{A}$  выдано число  $x$ , то немедленно завершим выполнение с результатом 1

( $\Leftarrow$ ): Если функцию  $\chi_X^*$  вычисляет алгоритм  $\mathcal{B}$ , то множество  $X$  перечисляет такой алгоритм:

- ▶ Выполнение алгоритма разделено на этапы  $(0, 1, \dots)$
- ▶ На  $i$ -м этапе выполняется один шаг вычисления каждого из значений  $\bar{B}(0), \dots, \bar{B}(i)$
- ▶ Для каждого вычисленного значения  $\bar{B}(x)$  выдаётся  $x$  ▼

# Вычислимость и перечислимость

Перечислимость множества чисел из  $\mathbb{N}_0$  можно расценивать как *очень слабое* требование, обеспечивающее *хоть какое-то* удобство работы с этим множеством:

- ▶ если множество перечислимо, то соответствующий перечисляющий алгоритм можно использовать в других алгоритмах
  - ▶ если элементы множества нельзя даже перечислить, то *удобно* использовать его в других алгоритмах не выйдет
- .....

В дальнейшем рассказе про формальную арифметику потребуется такое требование удобства не только для  $\mathbb{N}_0$ , но и для других множеств

Для этого введём особенный вид **кодирования** элементов заданного множества натуральными числами — **гёделеву нумерацию**

# Гёделева нумерация

Пусть задано множество  $S$  объектов, представляющих собой слова в заданном алфавите

Тогда (гёделевой) нумерацией называется

всюду определённое отображение  $g : S \rightarrow \mathbb{N}_0$ , такое что существуют

- ▶ *обратное* частично определённое отображение  $g^{-1} : \mathbb{N}_0 \rightarrow S$  и
- ▶ алгоритмы, реализующие  $g$  и  $g^{-1}$

**Пример:** нумерация  $g$  формул сигнатуры формальной арифметики

Для каждого символа  $s$  зададим число  $g(s)$ :

$$\begin{array}{llll} g(0) = 1 & g(+) = 2 & g(\cdot) = 3 & g(=) = 4 \\ g(( ) = 5 & g(,) = 6 & g()) = 7 & g(\forall) = 8 & g(\exists) = 9 \\ g(\&) = 10 & g(\vee) = 11 & g(\rightarrow) = 12 & g(\neg) = 13 \\ g(x_1) = 14 & g(x_2) = 15 & g(x_3) = 16 & \dots \end{array}$$

Гёделева нумерацию формул можно задать так:

$$g(\varphi) = p_1^{g(\varphi[1])} \cdot p_2^{g(\varphi[2])} \cdot \dots \cdot p_{|\varphi|}^{g(\varphi[|\varphi|])}, \text{ где}$$

$p_i$  —  $i$ -е простое число,  $\varphi[i]$  —  $i$ -й символ в записи формулы  $\varphi$  и

$|\varphi|$  — размер записи формулы  $\varphi$

# Теорема Гёделя о неполноте (формулировка)

Если для множества  $S$  задана гёделева нумерация  $g$ , то  $S$  назовём (рекурсивно) **перечислимым**, если перечислимо множество  $\{g(s) \mid s \in S\}$

## Теорема (Гёделя о неполноте)

Не существует перечислимой полной формальной арифметики

**Следствие.** Арифметика Пеано неполна

*(Так как арифметика Пеано перечислима)*

**Следствие.** Элементарная теория интерпретации  $Ar[\mathbb{N}_0; 0; +, \cdot, s; =]$  не является перечислимой

*(Так как любая элементарная теория полна)*

**Следствие.** Любая полная формальная арифметика неразрешима

*(Так как теоремы полной формальной арифметики — это аксиомы элементарной теории интерпретации  $Ar[\mathbb{N}_0; 0; +, \cdot, s; =]$ , и это множество не является даже рекурсивно перечислимым)*

# Теорема Гёделя ... (эскиз доказательства)

Для технической простоты докажем более простое утверждение:

Не существует **конечной** полной формальной арифметики

*От противного* предположим, что такая теория  $\mathcal{T}$  существует

Докажем, что в  $\mathcal{T}$  необходимо присутствует **парадокс лжеца**:

существует предложение,  
утверждающее, что это предложение ложно

.....

$Ar$  — так в доказательстве будем обозначать интерпретацию  $Ar[\mathbb{N}_0; 0; +, \cdot, s; =]$

Отношение  $R \subseteq \mathbb{N}_0^n$  называется **арифметизуемым**, если оно выражимо в  $Ar$

**Лемма(об арифметизации)**

График любой вычислимой функции арифметизуем

Эта лемма — ключевая и самая нетривиальная часть доказательства, но доказываться она не будет

## Теорема Гёделя ... (эскиз доказательства)

Рассмотрим какую-нибудь гёделеву нумерацию  $g$  формул (например, *описанную ранее в примере*)

**Утверждение.** Существует алгоритм, проверяющий, является ли число  $i$ ,  $i \in \mathbb{N}_0$ , номером формулы

**Утверждение.** Существует алгоритм, вычисляющий формулу по её номеру

**Утверждение.** Существует алгоритм, вычисляющий номер формулы, поданной на вход

Используя приём со степенями простых чисел из примера для построения нумерации, можно аналогично определить гёделеву нумерацию для

- ▶ конечных последовательностей формул
- ▶ конечных семантических таблиц
- ▶ конечных табличных выводов

## Теорема Гёделя ... (эскиз доказательства)

В теореме о полноте табличного вывода был описан алгоритм вычисления успешного вывода  $Tab(\varphi)$  для таблицы  $\langle \mathcal{T} \mid \varphi \rangle$ , где  $\varphi$  — произвольная  $\mathcal{T}$ -общезначимая формула

**Утверждение.** Существует алгоритм, завершающийся тогда и только тогда, когда на вход подан номер общезначимой формулы  $\varphi$ , и выдающий в ответ номер вывода  $Tab(\varphi)$

Иными словами, следующая функция вычислима:

$$f_t(i) = \begin{cases} g(Tab(g^{-1}(i))), & \text{если } i \text{ — номер } \mathcal{T}\text{-общезначимой формулы} \\ *, & \text{иначе} \end{cases}$$

**Значит**, график этой функции арифметизуем, то есть существует формула  $Proof(x, y)$ , такая что

$$Ar \models Proof(x, y)[d_1, d_2]$$

$\Leftrightarrow$

$d_2$  — номер табличного вывода  $Tab(\varphi)$ ,  
где  $\varphi$  — формула, номером которой является число  $d_1$

## Теорема Гёделя ... (эскиз доказательства)

$$Ar \models Proof(x, y)[d_1, d_2]$$

$\Leftrightarrow$

$d_2$  — номер табличного вывода  $Tab(\varphi)$ ,

где  $\varphi$  — формула, номером которой является число  $d_1$

.....

Рассмотрим формулу  $Valid(x) = \exists y Proof(x, y)$

Для неё верно:

$$Ar \models Valid[d] \Leftrightarrow d \text{ — номер формулы, истинной в } Ar$$

Тогда для формулы  $Invalid(x) = \neg Valid(x)$  будет верно:

$$Ar \models Invalid[d] \Leftrightarrow d \text{ не номер формулы, истинной в } Ar$$

Чтобы получить искомый **парадокс лжеца**, хотелось бы теперь сказать “рассмотрим  $d = g(Invalid)$ ” и получить противоречие

Увы, незамкнутость  $Invalid$  мешает получить противоречие так просто

## Теорема Гёделя ... (эскиз доказательства)

$Ar \models Invalid[d] \Leftrightarrow d$  не номер формулы, истинной в  $Ar$

.....

**Лемма(о диагонали).** Для любой формулы  $\varphi(x)$  в сигнатуре формальной арифметики существует предложение  $\psi$ , такое что

$$Ar \models (\psi \leftrightarrow \varphi(x))[g(\psi)]$$

Это вторая нетривиальная лемма, оставленная здесь без доказательства

Применим лемму о диагонали к формуле  $Invalid$ :

Существует предложение  $\psi$ , такое что  $Ar \models (\psi \leftrightarrow Invalid(x))[g(\psi)]$

Иными словами:

Существует предложение  $\psi$ , истинное в  $Ar$  тогда и только тогда, когда оно (согласно записанному в  $Invalid$ ) не является истинным в  $Ar$

Это соотношение представляет собой *противоречие*, выраженное в виде искомого *парадокса лжеца* ▼

# Заключение

Негативный результат теоремы Гёделя может показаться странным в свете того, что выбран довольно узкий фрагмент арифметики: “всего лишь” число 0, операции  $+$ ,  $\cdot$  и  $s$  и отношение  $=$

*Лемма об арифметизации* подсказывает, что эта кажущаяся узость обманчива: если разрешить использовать всё это и логические операции, то можно в арифметическом виде записать функцию, реализуемую любым алгоритмом