

Математическая логика

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

Блок 41

Формальная арифметика
Теорема Гёделя о неполноте

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

Вступление

Основные понятия и свойства, введённые (на данный момент) для аксиоматических теорий в этих лекциях:

- ▶ **непротиворечивость**, **полнота** и **разрешимость** теорий и взаимосвязь этих свойств
- ▶ **выразимость** понятий в 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 $Ar[\mathbb{N}_0; 0; +, \cdot, \mathbf{s}; =]$

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

В примерах возникала формальная арифметика и получше:

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

В аксиомах этой арифметики не содержится умножение
(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

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

В последней строке записана **схема аксиом** индукции с параметром φ — 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 обсудим несколько определений, связывающих арифметику, теорию алгоритмов и логику

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 :

- ▶ Одно из действий алгоритма — **выдать** (перечислить) указанное 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** χ_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 χ_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** χ_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 χ_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 χ_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 полностью 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 понадобится такое требование *удобства* не только для \mathbb{N}_0 , но и для других множеств

Для этого введём особенный вид **кодирования** элементов заданного множества числами из \mathbb{N}_0 — **гёделеву нумерацию**

Гёделева нумерация

Пусть задано множество S объектов,

представляющих собой слова в заданном алфавите

Тогда (гёделева) 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: нумерацию g формул сигнатуры формальной арифметики можно устроить так

Каждой букве ℓ алфавита формул сопоставим число $g(\ell)$:

$$\begin{array}{cccccc} g(0) = 1 & g(+) = 2 & g(\cdot) = 3 & g(\mathbf{s}) = 4 & g(=) = 5 & \\ g(() = 6 & g(,) = 7 & g(() = 8 & g(\forall) = 9 & g(\exists) = 10 & \\ g(\&) = 11 & g(\vee) = 12 & g(\rightarrow) = 13 & g(\neg) = 14 & & \\ g(\mathbf{x}_1) = 15 & g(\mathbf{x}_2) = 16 & g(\mathbf{x}_3) = 17 & & \dots & \end{array}$$

Каждой формуле φ сопоставим число

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

p_i — i -е простое число, $\varphi[i]$ — i -я буква в записи формулы φ и

$|\varphi|$ — размер записи формулы φ

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 $Ar[\mathbb{N}_0; 0; +, \cdot, s; =]$ **неперечислима**

(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 о неполноте (эскиз доказательства)

Для простоты обозначим схему доказательства более простого утверждения:

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 $Ar[\mathbb{N}_0; 0; +, \cdot, \mathbf{s}; =]$

Отношение $R \subseteq \mathbb{N}_0^n$ называется **арифметизуемым**, если оно выразимо 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 о неполноте (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 о неполноте (эскиз доказательства)

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

Утверждение. Существует алгоритм, завершающийся тогда и только тогда, когда на вход подан номер общезначимой формулы φ , 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 φ — формула, номером которой является число d_1

Теорема Гёделя о неполноте (эскиз доказательства)

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

\Leftrightarrow

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

Но формула $Invalid$ незамкнута,

а в **парадоксе лжеца** используется замкнутая формула

Теорема Гёделя о неполноте (эскиз доказательства)

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

.....

Лемма (о диагонали). Для любой формулы $\varphi(x)$ в сигнатуре формальной арифметики существует предложение ψ , 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 ψ , такое что $Ar \models (\psi \leftrightarrow Invalid(x))[g(\psi)]$

Иными словами:

Существует предложение ψ , истинное в Ar тогда и только тогда, когда оно (согласно устройству $Invalid$) 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 \mathbf{s} и отношение $=$

Лемма об арифметизации подсказывает, что эта кажущаяся узость обманчива:

в $Ar[\mathbb{N}_0; 0; +, \cdot, \mathbf{s}; =]$ выразима **любая** вычислимая функция

Иными словами, формулы сигнатуры формальной арифметики хотя и используют «скудный» набор понятий, но настолько же выразительны, как и машины Тьюринга