ФИВТ МФТИ, весна 2013.

Краткие заметки по курсу *математическая логика*. Часть четвертая: арифметическая иерархия и теорема Гёделя о неполноте (лекции 10–12).
А.Е. Ромащенко.

Заметки написаны для студентов, слушавших лекции курса и посещавших семинары на факультете ИВТ Физтеха. Текст непригоден для использования в качестве самостоятельного учебного пособия, независимого от занятий.

1 Арифметическая иерархия

Определение 1 *Говорят, что множество $A \subset \mathbb{N}^k$ принадлежит классу Σ_n, если существует такое разрешимое множество $R \in \mathbb{N}^{k+n}$, что*

$$(x_1, \ldots, x_k) \in A \leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n \ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R].$$

(кванторы по переменным y_i чередуются, начиная с квантора существования; квантор $\mathcal{Q}$ по переменной y_n будет квантором существования или всеобщности, в зависимости от чётности n).

Аналогично, говорят, что множество $A \subset \mathbb{N}^k$ принадлежит классу Π_n, если существует такое разрешимое множество $R \in \mathbb{N}^{k+n}$, что

$$(x_1, \ldots, x_k) \in A \leftrightarrow \forall y_1 \exists y_2 \forall y_3 \ldots \mathcal{Q} y_n \ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R].$$

Согласно этому определению $\Sigma_0 = \Pi_0$ (классы Σ_0 и Π_0 совпадают с классом всех разрешимых множеств). Заметим также, что класс Σ_1 состоит в точности из всех перечислимых множеств.

Утверждение 1 *Множество A принадлежит классу Σ_n, если и только если его дополнение $\bar{A}$ принадлежит классу Π_n.*

Доказательство: Если A принадлежит классу Σ_n, то существует такое разрешимое множество R, что

$$(x_1, \ldots, x_k) \in A \leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n \ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R].$$

Следовательно, для дополнения этого множества

$$\begin{array}{rcl} (x_1, \ldots, x_k) \in \bar{A} & \leftrightarrow & \neg \ \exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n \ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R] \\ & \leftrightarrow & \forall y_1 \exists y_2 \forall y_3 \ldots \bar{\mathcal{Q}} y_n \ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \notin R] \end{array}$$

(где $\bar{\mathcal{Q}}$ обозначает квантор противоположный к $\mathcal{Q}$). Утверждение доказано.

В частности, из Утверждения 1 следует, что класс Π_1 состоит из всех коперечислимых множеств (множеств, являющихся дополнениями к перечислимым).

Семейство всех классов Σ_n и Π_n называют *арифметической иерархией*. Смысл слова «иерархия» становится ясным из следующего утверждения (в котором мы докажем, что классы Σ_n и Π_n для меньших n содержатся в соответствующих классах с большими номерами n).

Утверждение 2 *Для каждого* $n \geq 0$

$$\Sigma_n \subset \Sigma_{n+1},\ \Pi_n \subset \Pi_{n+1},\ \Sigma_n \subset \Pi_{n+1},\ \Pi_n \subset \Sigma_{n+1}.$$

Доказательство: Множество A из класса Σ_n можно задать формулой вида

$$(x_1, \ldots, x_k) \in A \leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n\ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R].$$

Добавляя фиктивный квантор по переменной y_{n+1}, мы можем написать

$$(x_1, \ldots, x_k) \in A \leftrightarrow \forall y_{n+1} \exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n\ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R]$$

или

$$(x_1, \ldots, x_k) \in A \leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n \bar{\mathcal{Q}} y_{n+1}\ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R],$$

откуда видно, что A принадлежит классам Π_{n+1} и Σ_{n+1}. Вложения $\Pi_n \subset \Pi_{n+1}$ и $\Pi_n \subset \Sigma_{n+1}$ доказываются аналогично.

Как мы увидим ниже, все вложения в Утверждении 2 являются строгими.

Утверждение 3 *Определение классов* Σ_n *и* Π_n *не изменится, если в них разрешить формулы с произвольным числом кванторов и ограничить только число* чередований *кванторов. Более точно, множество* $A \subset \mathbb{N}^k$ *принадлежит* Σ_n*, если его можно задать формулой вида*

$$(x_1, \ldots, x_k) \in A \leftrightarrow \underbrace{\underbrace{\exists y_1 \ldots \exists y_{k_1}}_{\exists\text{-кванторы}}\ \underbrace{\forall y_{k_1+1} \ldots \forall y_{k_1+k_2}}_{\forall\text{-кванторы}}\ \underbrace{\exists y_{k_1+k_2+1} \ldots \exists y_{k_1+k_2+k_3}}_{\exists\text{-кванторы}} \cdots\cdots\ [(x_1, \ldots, y_1, \ldots) \in R]}_{n \text{ групп одноименных кванторов}}$$

для некоторого разрешимого множества R*. Аналогично, множество* A *принадлежит классу* Π_n*, если его можно задать формулой вида*

$$(x_1, \ldots, x_k) \in A \leftrightarrow \underbrace{\underbrace{\forall y_1 \ldots \forall y_{k_1}}_{\forall\text{-кванторы}}\ \underbrace{\exists y_{k_1+1} \ldots \exists y_{k_1+k_2}}_{\exists\text{-кванторы}}\ \underbrace{\forall y_{k_1+k_2+1} \ldots \forall y_{k_1+k_2+k_3}}_{\forall\text{-кванторы}} \cdots\cdots\ [(x_1, \ldots, y_1, \ldots) \in R]}_{n \text{ групп одноименных кванторов}}$$

для некоторого разрешимого R*.*

Для доказательства этого утверждения нужно воспользоваться вычислимым *кодированием* кортежей натуральных чисел: каждую группу идущих подряд одноименных кванторов кванторов $\exists y_1 \ldots \exists y_{k_1}$ можно заменить на квантор по одной единственной переменной $\exists z$, где z будет интерпретируется как *код* кортежа $\langle y_1 \ldots \exists y_{k_1} \rangle$. При этом потребуется соответствующим образом поменять и множество R. Попробуйте восстановить полное доказательство этого утверждения самостоятельно, не заглядывая в конспект лекций.

2 Вычисления с оракулом

На лекциях мы дали определение.вычисления с *оракулом*. Напомним, что машина Поста с оракулом $\mathcal{O}$ работает как обычная машина Поста, но имеет дополнительную возможность: можно время от времени обращаться с запросом к оракулу и (за один шаг) узнавать, принадлежит ли некоторое число z множеству $\mathcal{O}$. При этом множество $\mathcal{O} \subset \mathbb{N}$ может быть невычислимым.

Таким образом, для каждого множества $\mathcal{O} \subset \mathbb{N}$ у нас возникает новое (как говорят, «релятивизованное») определение алгоритма. Мы можем говорить о функциях, *вычислимых с оракулом* $\mathcal{O}$ (функция вычислима с оракулом $\mathcal{O}$, если существует машина с этим оракулом, которая для каждого входа из области определения функции находит её значение и останавливается, а для входов, не принадлежащих области значения, не останавливается), о множествах, *разрешимых с оракулом* $\mathcal{O}$ (множество разрешимо с оракулом $\mathcal{O}$, если её характеристическая функция вычислима с этим оракулом), о множествах, *перечислимых с оракулом* $\mathcal{O}$ (множество перечислимо с оракулом $\mathcal{O}$, если её полухарактеристическая функция вычислима с оракулом $\mathcal{O}$), и т.д.

Отметим, что все теоремы о вычислимости из нашего курса выдерживают «релятивизацию»: теоремы о существовании невычислимых функций и о перечислимых неразрешимых множествах, теорема Райса–Успенского, теорема Клини, и т.д. останутся верными, если в их формулировках заменить «алгоритмы» и «вычисления» на алгоритмы и вычисления с некоторым оракулом $\mathcal{O}$. При этом не нужно изобретать новые доказательства для «релятивизованных» теорем — все известные нам доказательства почти дословно переносятся на вычисления с оракулом.

Определение 2 *Говорят, что множество A сводится по Тьюрингу к множеству B, если A разрешимо с оракулом B. Обозначение: $A \leq_T B$.*

Утверждение 4 *Для сводимости по Тьюрингу выполнены следующие свойства:*

- *любое множество A сводится по Тьюрингу к самому себе: $A \leq_T A$,*
- *любое множество A сводится по Тьюрингу к своему дополнению, т.е., $A \leq_T \bar{A}$,*
- *если A разрешимо, то для любого B выполнено $A \leq_T B$,*
- *если $A \leq_T B$ и B разрешимо, то A тоже разрешимо,*
- *если $A \leq_T B$ и $B \leq_T C$, то $A \leq_T C$,*
- *если $A \leq_m B$, то $A \leq_T B$.*

Попробуйте воспроизвести доказательства всех этих свойств сводимости по Тьюрингу самостоятельно, не обращаясь к конспекту. В случае затруднений можно обратиться к главе 7 [1].

Как обычно, мы обозначаем $p_0, p_1, \ldots, p_n, \ldots$ стандартную нумерацию программ для машины Поста с оракулом. Используя вычислимость с оракулом, мы можем по индукции определить следующую бесконечную последовательность множеств, возрастающей «неразрешимости»:

0. обозначим $\mathbf{0}$ пустое множество,
1. обозначим $\mathbf{0}'$ множество $\{n \mid$ машина p_n останавливается на входе $n\}$,
2. обозначим $\mathbf{0}''$ множество $\{n \mid$ машина p_n с оракулом $\mathbf{0}'$ останавливается на входе $n\}$,

   ...

k. обозначим $\mathbf{0}^{(k)}$ множество $\{n \mid$ машина p_n с оракулом $\mathbf{0}^{(k-1)}$ останавливается на входе $n\}$,

   ...

где p_n есть n-ая программа для машины Поста в стандартной нумерации.

Теорема 1 (теорема об арифметической иерархии) *Для каждого натурального числа $k > 0$ множество $\mathbf{0}^{(k)}$ принадлежит классу Σ_k, но не принадлежит Σ_{k-1}.*

На лекциях мы не доказывали теорему об арифметической иерархии, и в программу экзамена доказательство не входит. Однако вы можете прочитать это доказательство в [1].

Из теоремы об арифметической иерархии немедленно вытекает, что для каждого n класс Σ_n является собственным подмножеством Σ_{n+1} (поскольку существует множество, лежащее в Σ_{n+1}, но не лежащее в Σ_{n+1}).

Выведите из теорема об арифметической иерархии, что (а) для каждого n класс Π_n является собственным подмножеством Π_{n+1}, и (б) для каждого $n > 0$ $\Sigma_n \neq \Pi_n$.

3 Арифметические множества

Напомним, что язык первого порядка формальной арифметики содержит константу 0, один одноместный функциональный символ S и два двухместных функциональных символа $+$ и $\cdot$. Единственным предикатным символом языка является равенство. Носителем стандартной модели этого языка являются натуральные числа; при этом 0 интерпретируется как ноль, S как прибавление единицы, $+$ и $\cdot$ как обычное сложение и умножение соответственно.

С одной стороны, этот язык довольно прост. С другой стороны, он позволяет выразить многие важные понятия «финитной» математики.

Утверждение 5 *Существует арифметическая формула $Seq(x, y, z, w)$, которая представляет в языке арифметики все конечные последовательности натуральных чисел в следующем смысле: для любой последовательности натуральных чисел $m_1, m_2, \ldots, m_n$ существуют такие натуральные числа a, b, что для каждого $i = 1, \ldots, n$ формула Seq истинна на оценках, сопоставляющих свободным переменным x, y, z, w значения a, b, i, m_i соответственно, и ложна на оценках, сопоставляющих переменным x, y, z, w значения a, b, i, m' для любого $m' \neq m$.*

Прежде чем приступать к доказательству, отметим, что с помощью формулы Seq в языке арифметики можно «кодировать» все конечные последовательности натуральных чисел, При этом «кодом» последовательности $m_1, m_2, \ldots, m_n$ будет тройка чисел (a, b, n) (где n есть длина последовательности, а числа a и b описывают члены последовательности в указанном выше смысле). Отметим так же, что для $i > n$ Утверждение 5 ничего не гарантирует относительно истинности Seq на наборах a, b, i, x.

Для доказательства Утверждения 5 нам понадобится следующая лемма из элементарной теории чисел.

Лемма 1 *Для любого n существует бесконечно много натуральных чисел b таких, что $b + 1, 2b + 1, \ldots, nb + 1$ попарно взаимно просты.*

Доказательство леммы: Достаточно взять b кратным числу $n!$. В самом деле, предположим, что у чисел $ib + 1$ и $jb + 1$ (для $0 \leq i < j \leq n$) есть общий простой делитель p. Тогда разность этих двух чисел $(j - i)b$ тоже делится на p. Это значит, что p делит b или $j - i$. Далее, заметим, что если p делит разность $j - i$, то (поскольку $0 < j - i < n$) p должен делить и $n!$. Следовательно, число p в любом случае должно делить b. Но простое число p не может делить одновременно ib (кратное $n!$) и число $ib + 1$. Полученное противоречие завершает доказательство леммы.

Доказательство утверждения 5: Пусть задана последовательность натуральных чисел $m_1, \ldots, m_n$. Согласно Лемма 1, можно найти такое натуральное число b, которое больше всех n_i из заданной последовательности, и при этом числа $b + 1, 2b + 1, \ldots, nb + 1$ попарно взаимно просты.

Далее, по *китайской теореме об остатках* найдётся такое число a, что для всех $i = 1 \ldots n$ при делении a на число $ib + 1$ получается в остаток n_i. Вспомните формулировку и доказательство китайской теоремы об остатках!

Теперь мы можем «закодировать» последовательность $m_1, \ldots, m_n$ тройкой чисел a, b, n: каждое m_i определяется по правилу «взять остаток от деления a на $ib + 1$». Остаётся записать формулу в языке арифметики, которая выражает это свойство. Это несложно. Мы хотим выразить свойство

$$a = (i \cdot b + 1) \cdot q + m_i,$$

где q есть неполное частное, а m_i остаток. Это значит, что

$$\exists q((a = (i \cdot b + 1) \cdot q + m_i) \ \& \ (m_i < (i \cdot b + 1)).$$

Поскольку в нашем языке формальной арифметики нет константы «один» и символа «меньше», мы должны прибегнуть к несложному трюку и переписать данное свойство в виде

$$\exists q((a = (i \cdot b + S0) \cdot q + m_i) \ \& \ (\exists t \ m_i + t = i \cdot b)).$$

Таким образом, формулу $Seq(x, y, z, w)$ можно записать в виде

$$\exists q((x = (z \cdot y + S0) \cdot q + w) \ \& \ (\exists t \ w + t = z \cdot y)).$$

Утверждение 5 доказано.

Теорема 2 *Для любой машины Поста M можно построить такую арифметическую формулу $\varphi_M(x)$ (с одной свободной переменной x), что машина M останавливается на входе $1^{n+1} = \underbrace{11\ldots1}_{n+1}$, если и только если $\varphi_M(x)$ истинна на оценках, сопоставляющих значение n переменной x.*

Замечание: аналогичное утверждение верно для машин с несколькими входами, а также для машин, получающий вход не в унарной, а в двоичной записи.

Набросок доказательства Теоремы 2: Чтобы описать текущую конфигурацию машины M в процессе её работы, нужно указать следующую информацию:

- содержимое каждой булевой переменной, используемой программой,
- указатель на текущую строку программы,
- набор битов на входе, ещё не прочитанных программой, (в нашем случае это просто число единиц, ещё не прочитанных из входа),
- запись на рабочей ленте левее положения каретки (включая ячейку, в которой находится каретка в данный момент),
- запись на рабочей ленте правее положения картеки (хотя лента бесконечна вправо, почти все её ячейки заполнены нулями; поэтому для описание состояния всей ленты справа от каретки нужно лишь конечное число битов).

Заметим, что всякую такую конфигруацию можно описать небольшим набором натуральных чисел. В самом деле, содержимое каждой булевой переменной есть ноль или единица; указатель на текущую строку программы и число единиц, ещё не прочитанных из входа, есть неотрицательные целые числа. Записи на рабочей ленте левее и правее положения каретки тоже можно закодировать натуральными числами. Припишем к записи на ленте левее каретки единицу и будем рассматривать полученную последовательность битов как двоичную запись некоторого натурального числа L (удобно считать, что читая биты на ленте слева направо мы идём от старших разрядов к младшим). (Единицу мы приписали для того, чтобы 0, 00, 000 и

другие записи из одних нулей соответствовали двоичным представлениям *разных* чисел $10_2 = 2$, $100_2 = 4$, $1000_2 = 8$ и т.д.) Аналогично, биты на ленте правее каретки будем рассматривать как двоичную запись некоторого наутрального числа R (удобно считать, что читая биты на ленте направо от каретки мы идём от младших разрядов двоичного представления R к старшим).

Утверждение о том, что машина M останавливается на входе 1^{n+1}, можно записать таким образом: существует натуральное число t и существует последовательность конфигураций машины M, которые мы обозначим $\pi_1, \ldots, \pi_t$, со следующими свойствами:

- конфигурация π_1 соответствует начальному состоянию машины, т.е.,
  - содержимое каждой булевой переменной равно нулю,
  - указатель текущей строки показывает на самую первую строку программы,
  - набор битов на входе, ещё не прочитанных программой, состоит из $(n+1)$ единиц,
  - запись на рабочей ленте левее положения каретки состоит из единственной клетки с записанным в ней нулём,
  - запись на рабочей ленте правее положения картеки состоит из одних нулей,
- состояние π_t соответствует состоянию остановки, т.е., указатель текущей строки показывает на одну из строк программы, в которой записан оператор `stop`,
- для каждого $i = 1, \ldots, (t-1)$ конфигурация π_{i+1} получается из конфигурации π_i за один шаг работы машины M.

Таким образом, утверждение о том, что M на входе n останавливается, можно переписать как утверждение о существовании числа t и набора последовательностей чисел (длины t каждая), которые кодируют всю информацию о мгновенных состояниях $\pi_1, \ldots, \pi_t$ данной машины, от начальной конфигурации до состояния остановки.

Утверждение 5 показывает, что каждую последовательность из t натуральных чисел можно так закодировать парой натуральных чисел a, b, чтобы утверждения о значении i-го члена этой последовательности можно было выразить несложной арифметической формулой. Используя эту идею, можно переписать утверждение «машина M останавливается на входе 1^{n+1}» с помощью формулы в языке формальной арфиметики. (Попробуйте закончить доказательство теоремы, не заглядывая в конспект лекций!)

Определение 3 *Множество $A \subset \mathbb{N}^k$ называется* арифметическим, *если существует такая формула языка формальной арифметики $\varphi(x_1, \ldots, x_k)$ с k свободными переменными, что $(n_1, \ldots, n_k) \in A$, если и только если $\varphi(x_1, \ldots, x_k)$ истинна на оценках, сопоставляющих числа $n_1, \ldots, n_k$ параметрам $x_1, \ldots, x_k$.*

Теорема 3 *Множество A является арифметическим, если и только если A лежит в одном из классов арифметической иерархии, т.е., в $\bigcup_{i=n}^{\infty} \Sigma_n$.*

Доказательство: Импликацию в одну сторону доказать совсем просто: если множество арифметическое, то оно представляется некоторой формулой $\varphi(x_1, \ldots, x_k)$ в языке формальной арифметики. Приведем эту формулу к предваренной нормальной форме (вытащим наружу все кванторы). Тогда внутри останется бесквантороная часть – булева комбинация равенств арифметических термов. Эти термы кроме свободных переменных $x_1, \ldots, x_k$ могут включать и некоторое количество связанных кванторами переменных. Нетрудно видеть, что данная бесквантороная часть является разрешимым свойством значений параметров: для каждого набора значений всех переменных мы можем вычислить значение каждого из термов, входящих в формулу, и выяснить, является ли вся эта (бесквантороная) формула истинной или ложной. Таким образом, приведенная к предваренной нормальной форме $\varphi(x_1, \ldots, x_k)$ имеет вид

$$[\text{кванторная приставка}]\ (\text{разрешимый предикат}).$$

Понятно, что такая формула задаёт свойство класса Σ_n для некоторого (достаточно большого) n.

Для доказательства импликации в другую сторону нам нужна Теорема 2. Пусть некоторое множество A лежит в Σ_n. Это значит, что принадлежность к A можно выразить формулой

$$\exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n\ [(x_1, \ldots, x_k, y_1, \ldots, y_n) \in R],$$

для некоторого разрешимого множества R. Поскольку R разрешима, существует машина Поста M, которая получает на вход набор из $k+n$ натуральных чисел и останавливается в том и только том случае, когда этот набор принадлежит R. По существу, мы здесь пользуемся даже не разрешимостью, а перечислимостью множества R. По Теореме 2 существует арифметическая формула $\varphi_M(x_1, \ldots, x_k, y_1, \ldots, y_k)$, которая истинна ровно на тех наборах натуральных чисел, на которых машина M останавливается (т.е., на тех наборах, которые принадлежат множеству R). Остаётся приписать к этой формуле кванторную приставку: принадлежность к A выражается арифметической формулой

$$\exists y_1 \forall y_2 \exists y_3 \ldots \mathcal{Q} y_n\ \varphi_M(x_1, \ldots, x_k, y_1, \ldots, y_n).$$

Теорема доказана.

4 Системы аксиом для языков первого порядка

Определение 4 Системой аксиом *в языке первого порядка L называется произвольное разрешимое множество формул данного языка.*

Важно, что система аксиом может состоять из бесконечного множества формул; мы требуем лишь разрешимости этого множества. Отметим, что данное определение является чисто синтаксическим, в нём ничего не говорится об интерпретации языка и тем более об истинности аксиом.

Говорят, что формула φ языка L выводится из системы аксиом $\mathcal{A}$ (*доказывается* в данной аксиоматической системе), если φ выводится из $\mathcal{A}$ в исчислении предикатов. Другими словами, существует такая последовательность формул $\psi_1, \ldots, \psi_n$, в которой каждая из формул либо является аксиомой исчисления предикатов, либо одной из формул $\mathcal{A}$, либо получается из предыдущих формул по одному из правил вывода исчисления предикатов (modus ponens или одному из правил Бернайса); при этом последняя формула ψ_n должна совпадать с формулой φ.

В дальнейшем мы будем рассматривать только языки первого порядка с равенством и будем считать, что аксиомы равенства включены в исчисление предикатов.

Утверждение 6 *Для любой системы аксиом $\mathcal{A}$ множество выводимых из неё формул перечислимо.*

Доказательство: Мы можем алгоритмически перечислять конечные последовательности формул (например, в порядке возрастания суммы длин формул в такой последовательности). Для каждой последовательности формул можно проверить, является ли она выводом из $\mathcal{A}$ (тут мы пользуемся тем, что множество $\mathcal{A}$, а также множество аксиом исчисления предикатов разрешимы). Если очередная последовательность формул оказывается выводом из $\mathcal{A}$, мы добавляем последнюю формулу из этой последовательности к списку выводимых формул. Так мы рано или поздно перечислим каждую формулу, выводимую из данной системы аксиом.

Определение 5 *Система аксиом $\mathcal{A}$ называется* противоречивой*, если существует такая формула φ, что из $\mathcal{A}$ выводятся одновременно φ и $\neg\varphi$. Система аксиом, не являющаяся противоречивой, называется* непротиворечивой*.*

Утверждение 7 *Система аксиом $\mathcal{A}$ противоречива, если и только если из $\mathcal{A}$ можно вывести все формулы языка.*

Доказательство: Импликация в одну сторону: если из $\mathcal{A}$ выводятся все формулы языка, то можно вывести и некоторые φ и $\neg\varphi$.

Импликация в другую сторону: пусть из $\mathcal{A}$ выводятся некоторые φ и $\neg\varphi$. Напомним, что в исчислении предикатов для любых φ и ψ выводится формула $\varphi \to (\neg\varphi \to \psi)$. Применяя дважды modus ponens, получаем ψ. Таким образом, из $\mathcal{A}$ оказывается возможным вывести любую формулу ψ.

Определение 6 *Непротиворечивая система аксиом $\mathcal{A}$ называется* полной*, если для любой замкнутой формулы ψ из $\mathcal{A}$ выводятся либо ψ, либо $\neg\psi$.*

4.1 Пример непротиворечивой полной системы аксиом

Рассмотрим язык линейного порядка (язык содержит предикатные символы $\leq$ и $=$). Обозначим T множество замкнутых формул данного языка, истинных в интерпретации $(\mathbb{Q}, \leq)$ (теория линейного порядка на множестве рациональных числе). В первом семестре мы доказывали, что T разрешимо: существует алгоритм, проверяющий, истинна ли заданная формула в данной интерпретации. Вспомните алгоритм элиминации кванторов, который позволял выяснить истинность или ложность замкнутой формулы в данной интерпретации. Следовательно, можно рассмотреть систему аксиом, состоящую из всех формул T. Данная система аксиом полна и непротиворечива. (Объясните, почему!)

Интересно, что данную теорию можно описать и конечным множеством аксиом. А именно, рассмотрим аксиомы плотного линейного порядка без максимального и минимального элементов. (Запишите эти аксиомы в виде формул языка первого порядка!) Из этих аксиом будут вводиться все замкнутые формулы, истинные в $(\mathbb{Q}, \leq)$, и не будет выводиться ни одна замкнутая формула, ложная в данной интерпретации. В самом деле, если некоторые формулы φ и $\neg\varphi$ обе не могут быть выведены из этих аксиом, то каждую из формул φ и $\neg\varphi$ (но не обе сразу) можно добавить к данной системе аксиом. В результате получатся две разные непротиворечивые системы аксиом. (Объясните, почему эти системы аксиом будут непротиворечивы!) У каждой из этих систем аксиом будет (хотя бы одна) счетная модель, и эти модели не будут элементарно эквивалентны. Но это противоречит тому, что все счетные модели плотного линейного порядка без максимального и минимального элементов изоморфны. (Проведите это рассуждение подробно.)

4.2 Аксиомы Пеано

Следующий бесконечный набор аксиом называют системой аксиом Пеано для формальной арифметики:

1. $\forall x(\neg sx = 0)$
2. $\forall x(\neg x = 0 \to \exists y(sy = x))$
3. $\forall x \forall y(sx = sy \to x = y)$
4. $\forall x \forall y(x + 0 = x)$
5. $\forall x \forall y(x + sy = s(x + y))$
6. $\forall x \forall y(x \cdot 0 = 0)$
7. $\forall x \forall y(x \cdot sy = x \cdot y + x)$
8. $(\varphi(0)\ \&\ \forall x(\varphi(x) \to (\varphi(sx)))) \to (\forall x \varphi(x))$

(последняя строчка является бесконечной серией аксиом — мы включаем в список аксиом такие формулы для всех подформул $\varphi(x)$).

Данный набор аксиом кажется очень бедным (в нём отсутствуют даже такие привычные нам свойства натуральных чисел, как коммутативность и ассоциативность сложения и умножения). Однако этого набора аксиом оказывается достаточно, чтобы доказывать очень сложные свойства натуральных чисел. Иногда говорят, что в системе аксиом Пеано можно формализовать любое «финитное» рассуждение. (Данное утверждение, подобно тезису Чёрча, не является математической теоремой, поскольку у нас нет формального определения *финитного* рассуждения.)

Отметим, что система аксиом Пеано непротиворечива (поскольку имеет модель – стандартную модель арифметики).

Необязательные упражнения: докажите с помощью аксиом Пеано формулы $S(S0) + S(S0) = S(S(S(S0)))$ и $\forall x \forall y (x + y = y + x)$.

5 Неразрешимость формальной арифметики и первая теорема Гёделя о неполноте

Теорема 4 *Множество замкнутых формул языка формальной арифметики, истинных в стандартной интерпретации, неразрешимо.*

Доказательство: Пусть $A \subset \mathbb{N}$ какое-нибудь перечислимое неразрешимое множество, и M машина поста, вычисляющая его полухарактеристическую функцию (останавливающаяся на входе 1^{n+1} в том и только том случае, когда $n \in A$). По Теореме 2 найдется такая арифметическая формула $\varphi_M(x)$ которая истинна на оценках, сопоставляющих параметру x одно из чисел множества A (число, на котором машина M останавливается), и ложна на оценках, сопоставляющих параметру x число из дополнения A (число, на котором машина M не останавливается). Таким образом, формула $\varphi_M(\underbrace{S(S\ldots S}_{n}(0)))$ истинна, если $n \in A$ и ложна, если $n \notin A$.

Теперь предположим, что множество истинных (в стандартной интерпретации) формул языка формальной арифметики разрешимо. Тогда существует алгоритм, который по формуле вида $\varphi(S(S\ldots S(0)))$ определяет, истинна она или ложна. С помощью этого алгоритма можно определить, принадлежит ли заданное число n множеству A. Но это противоречит неразрешимости множества A. Теорема доказана.

Теорема 5 (Первая теорема Гёделя о неполноте) *Не существует системы аксиом, из которой были бы выводимы все замкнутые формулы языка формальной арифметики, истинные в стандартной интерпретации, и не выводимо ни одной ложной (в стандартной интерпретации) замкнутой формулы.*

Доказательство: Пусть существует такая система аксиом $\mathcal{A}$, из которой выводимы все замкнутые формулы языка формальной арифметики, истин-

ные в стандартной интерпретации, и не выводимо ни одной ложной. Это значит, что для истинных замкнутых формул φ из $\mathcal{A}$ можно вывести саму φ, а для ложных φ из $\mathcal{A}$ выводится $\neg\varphi$ (поскольку отрицание ложной формулы будет истинным).

Для любой системы аксиом множество выводимых формул перечислимо. Но это значит, что множество истинных замкнутых формул формальной арифметики должно быть разрешимым. В самом деле, чтобы узнать, истинна ли некоторая формула φ или ложна, мы запустим перечисление всех выводимых из $\mathcal{A}$ формул и будем ждать, пока в этом списке не появится φ или $\neg\varphi$.

Таким образом, мы пришли к выводу о разрешимости множества истинных замкнутых формул формальной арифметики. Но это противоречит Теореме 4. Теорема Гёделя доказана.

6 Прямое доказательство первой теоремы Гёделя о неполноте

В этом разделе мы приведем прямое доказательство первой теоремы Гёделя о неполноте, явно указав истинную, но недоказуемую формулу. Пусть $\mathcal{A}$ некоторая система аксиом языка формальной арифметики, из которой выводимы только истинные (в стандартной модели) замкнутые формулы.

Рассмотрим следующее вычислимое преобразование F на множестве программ для машин Поста. Чтобы определить значение $F(\pi)$ для некоторой программы π мы прежде всего с помощью Теоремы 2 строим арифметическую формулу $\varphi_\pi(x)$ с одной свободной переменной, которая истинна на тех и только тех n, на которых программа π определена (программа завершает работу, получив n на вход). Затем строим следующую программу π'.

На входе x выполняем следующие действия:

1. Перечисляем все выводы из $\mathcal{A}$, пока не найдём вывод формулы $\neg\varphi_M(\underbrace{S(S\ldots S}_{x\text{ раз}}(0)))$
2. Печатаем 1 и прекращаем работу.

Данную программу π' мы и будем считать значением $F(\pi)$.

Построенная программа $\pi' = F(\pi)$ останавливается на входе n, если и только если про исходную программу π система аксиом $\mathcal{A}$ может доказать, что та *не останавливается* на входе n.

По теорема Клини о неподвижной точке существует такая программа π_0, что π_0 и $F(\pi_0)$ эквивалентны друг другу. Несколько неформально можно сказать, что программа π_0 останавливается на входе n, если и только если в $\mathcal{A}$ можно доказать, что π_0 на входе n не останавливается.

Таким образом, если в $\mathcal{A}$ можно доказать, что π_0 на некотором входе n не останавливается, то это утверждение оказывается ложным — программа на данном входе на самом деле останавливается. Напомним, что из $\mathcal{A}$

нельзя вывести ложных формул. Следовательно, π_0 не останавливается ни на одном входе; однако в системе аксиом $\mathcal{A}$ невозможно доказать истинную формулу $\neg\varphi_M(\underbrace{S(S\ldots S}_{x \text{ раз}}(0)))$, выражающую этот факт.

7 Синтаксическая формулировка первой теоремы Гёделя о неполноте

Этот раздел не входит в программу экзамена.

Теорема 6 (Синтаксический вариант первой теоремы Гёделя о неполноте) *Систему аксиом Пеано невозможно расширить до непротиворечивой и полной системы аксиом для языка формальной арифметики.*

Эквивалентная переформулировка теоремы: *Если из некоторой системе аксиом $\mathcal{A}$ можно вывести все аксиомы Пеано, то $\mathcal{A}$ не может быть непротиворечивой и полной.*

Замечание 1. В данной формулировке ничего не говорится об истинности аксиом. Теорема распространяется и на системы аксиом, содержащие ложные (в стандартной интерпретации) арифметические формулы.

Замечание 2. Требование включения аксиом Пеано в систему аксиом $\mathcal{A}$ можно немного ослабить (см. [6]), но некоторое аналогичное условие в теореме обязательно должно присутствовать. Такое условие гарантирует, что мы рассматриваем систему аксиом, корректно описывающую хотя бы простейшие свойства натуральных чисел. В самом деле, можно искусственно построить для языка арифметики некоторую полную и непротиворечивую систему аксиом; однако всякая модель такой теории будет совершенно не похожа на натуральные числа с обычными сложением и умножением.

Упражнение: Докажите, что теорема Гёделя о неполноте в «семантической» формулировке является следствие теоремы Гёделя о неполноте в приведенной выше «синтаксической» формулировке.

8 Вторая теорема Гёделя о неполноте

Пусть $\mathcal{A}$ некоторая система аксиом формальной арифметики, содержащая аксиомы Пеано (и, быть может, некоторые другие аксиомы). Заметим, что из аксиом Пеано можно вывести формулу $\neg(0 = S0)$ (ноль не равен единице). (Объясните, как эта формула выводится из аксиом Пеано!) Таким образом, $\mathcal{A}$ противоречива, если и только если из $\mathcal{A}$ можно вывести формулу $0 = S0$.

Рассмотрим машину Поста M, которая перечисляет все выводы в $\mathcal{A}$ и останавливается, когда найдет вывод $0 = S0$. Таким образом, данная машина останавливается, если $\mathcal{A}$ противоречива, и никогда не останавливается, если система аксиом непротиворечива.

С помощью конструкции из Теоремы 2 можно построить замкнутую арифметическую формулу φ_M, которая истинна, если M останавливается, и ложна в противном случае. Другими словами, данная формула выражает свойство непротиворечивости $\mathcal{A}$. Эту формулу традиционно обозначают $\mathrm{Consis}_{\mathcal{A}}$ (от слова *consistency*).

Теорема 7 (Вторая теорема Гёделя о неполноте) *Если система аксиом $\mathcal{A}$ непротиворечива и содержит аксиомы Пеано, то из неё нельзя вывести формулу* $\mathrm{Consis}_{\mathcal{A}}$.

Вторую теорему Гёделя о неполноте часто интерпретируют так: непротиворечивая система аксиом достаточно богатой теории не может доказать свою собственную непротиворечивость.

Список литературы

[1] Верещагин Н., Шень А. Вычислимые функции - М.: МЦНМО, 1999.

[2] Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов - М.: Физико-математическая литература, 1995.

[3] Успенский В.А. Машина Поста. - М.: Наука, 1988.

[4] Верещагин Н.К, Плиско, В.Е., Успенский В.А. Вводный курс математической логики. М.: 1997.

[5] Колмогоров А.Н., Драгалин А.Г. Математическая логика. - М.: КомКнига, 2006.

[6] Булос Дж., Джеффри Р. Вычилимость и логика. - М.: Мир, 1994.