Теорема Гёделя–Россера

Мех-мат МГУ, 1-й курс, весна 2008 г.

Л.Д. Беклемишев

1 Теорема Гёделя о неполноте

1.1 Арифметика Пеано и арифметика Робинсона

Мы докажем несколько упрощённый вариант теоремы Гёделя о неполноте. Основное упрощение касается выбора языка формальной арифметики: мы расширим сигнатуру арифметики символами $\leq$ (порядок) и exp (экспонента), где exp есть функция $\exp(x) = 2^x$. Таким образом, сигнатура арифметики содержит символы 0, S, $+$, $\cdot$, exp, $\leq$, $=$.

Наличие экспоненты в языке арифметики позволяет очень просто формализовать кодирование слов в данном алфавите. С другой стороны, Гёдель показал, что функция 2^x является определимой в арифметике Пеано, поэтому явное добавление символа exp в сигнатуру (вместе с соответствующими аксиомами) не меняет, по существу, саму теорию. Поэтому мы сохраним за теорией в расширенном языке название *арифметика Пеано*.

Определение 1.1. *Арифметика Пеано* PA задаётся следующими нелогическими аксиомами:

1. аксиомы равенства для сигнатуры 0, S, $+$, $\cdot$, exp, $\leq$, $=$;
2. $\neg S(a) = 0, \quad S(a) = S(b) \to a = b$,
3. $a + 0 = a, \quad a + S(b) = S(a + b)$,
4. $a \cdot 0 = 0, \quad a \cdot S(b) = a \cdot b + a$,
5. $\exp(0) = S(0), \quad \exp(S(a)) = \exp(a) + \exp(a)$
6. $a \leq 0 \leftrightarrow a = 0$
7. $a \leq S(b) \leftrightarrow (a \leq b \lor a = S(b))$

8. (Схема аксиом индукции)
   $A[a/0] \land \forall x\,(A[a/x] \to A[a/S(x)]) \to \forall x\,A[a/x]$,
   для любой формулы A.

Стандартной моделью арифметики Пеано называем модель

$$(\mathbb{N}; 0, S, +, \cdot, \exp, \leq, =).$$

Следующие лемма и следствие очевидны.

Лемма 1.2. $\mathbb{N} \models \mathsf{PA}$.

Следствие 1.3. PA *непротиворечива.*

Определение 1.4. *Арифметика Робинсона* Q получается из PA заменой схемы индукции единственной аксиомой:

$$a \leq b \lor b \leq a.$$

Замечание 1.5. Заметим, что из этой аксиомы следует $a \leq a$ (положим $b = a$) и $a \leq b \lor b < a$ (поскольку $\neg\, a \leq b \to \neg\, a = b$ в силу предыдущего).

Замечание 1.6. Теория Q задаётся конечным числом аксиом.

Упражнение 1.7. *Показать, что* $\mathsf{PA} \vdash \mathsf{Q}$.

Решение. Последовательно докажем индукцией по x:

(i) $\forall x\ (a \leq x \leftrightarrow a = x \lor S(a) \leq x)$;

(ii) $\forall x\ (a \leq x \lor x \leq a)$.

Заметим, что из (i) следует $a \leq a$ и $a \leq S(a)$.

Вывод утверждения (i):

Базис индукции: $a \leq 0 \leftrightarrow a = 0 \lor S(a) \leq 0$.
Импликации $a \leq 0 \to a = 0$ и $a = 0 \to a \leq 0$ получаем по аксиоме 6. Поэтому достаточно вывести $\neg S(a) \leq 0$. По аксиоме 6 формула $S(a) \leq 0$ влечет $S(a) = 0$, что противоречит аксиоме 2.

Шаг индукции: надо показать $a \leq S(x) \leftrightarrow S(a) \leq S(x) \lor a = S(x)$.
Пользуясь предположением индукции строим следующую цепочку формул, каждая из которых эквивалентна предыдущей:

1. $a \leq S(x)$
2. $a \leq x \vee a = S(x)$ (по аксиоме 7)
3. $(a = x \vee S(a) \leq x) \vee a = S(x)$ (по предположению индукции)
4. $(S(a) = S(x) \vee S(a) \leq x) \vee a = S(x)$ (по аксиоме 2)
5. $S(a) \leq S(x) \vee a = S(x)$ (по аксиоме 7).

Вывод утверждения (ii):

Базис индукции: $a \leq 0 \vee 0 \leq a$. Мы получаем $0 \leq a$ очевидной индукцией по a.

Шаг индукции:

1. $a \leq x \vee x \leq a$ (предположение индукции)
2. $a \leq S(x) \vee x \leq a$ (по аксиоме 7)
3. $a \leq S(x) \vee (S(x) \leq a \vee x = a)$ (по утверждению (i))
4. $a \leq S(x) \vee (S(x) \leq a \vee a \leq S(x))$ (из $a \leq S(a)$)
5. $a \leq S(x) \vee S(x) \leq a$.

Таким образом, теория Q представляет собой конечную подтеорию арифметики PA.

Замечание 1.8. В теории Q не возможны доказательства по индукции, поэтому она не позволяет вывести сколько-нибудь содержательные свойства арифметических операций (см. упражнение ниже). Другими словами, Q является очень слабой подтеорией арифметики PA. Она играет роль минимально достаточной теории, для которой справедливы теоремы Гёделя о неполноте. Выбор такой теории, в отличие от PA, в значительной степени произволен. В частности, сам Р. Робинсон обозначал через Q несколько иную теорию (отличия, в основном, связаны с выбранным здесь вариантом языка арифметики).

Упражнение 1.9. *Докажите, что в теории* Q *не выводимы следующие формулы:* $\neg\, a = S(a)$, $a \cdot b = b \cdot a$.

1.2 Формулировки теорем Гёделя о неполноте

Теперь мы можем дать формулировки теорем Гёделя о неполноте.

Теорема 1.10 (первая теорема Гёделя о неполноте). *Если теория T*

- *в арифметическом языке,*
- *эффективно аксиоматизируема,*
- $\mathbb{N} \models T$*,*

то T неполна, то есть существует арифметическое предложение A такое, что $T \nvdash A$ и $T \nvdash \neg A$.

Следствие 1.11. PA *неполна.*

Замечание 1.12. Условие корректности $\mathbb{N} \models T$ в данном варианте теоремы Гёделя не является оптимальным. Сам Гёдель установил свою теорему при более слабом, но менее естественном, предположении ω-непротиворечивости T. В дальнейшем Дж.Б. Россер усилил теорему Гёделя, доказав неполноту теории T всего лишь при условии её обычной непротиворечивости. Однако при этом возникает дополнительное (но не очень ограничительное) требование $T \vdash \mathsf{Q}$.

Теорема 1.13 (теорема Гёделя–Россера). *Если*

- *теория T содержит Q,*
- *T эффективно аксиоматизируема,*
- *T непротиворечива,*

то T неполна.

Замечание 1.14. Теорема Гёделя–Россера также применима к теориям в произвольном языке, в которых интерпретируема Q. К таким теориям относится, в частности, теория множеств ZFC.

Следствие 1.15. *ZFC неполна (при условии своей непротиворечивости).*

Теорема 1.16 (вторая теорема Гёделя о неполноте). *Если*

- PA *интерпретируема в T,*

- *T эффективно аксиоматизируема,*
- *T непротиворечива,*

то $T \nvdash \mathsf{Con}(T)$, где $\mathsf{Con}(T)$ – арифметическая формула, выражающая непротиворечивость T.

Замечание 1.17. Условие интерпретируемости PA в T во второй теореме Гёделя о неполноте было в дальнейшем ослаблено до знакомого нам условия интерпретируемости теории Q в T. Такое ослабление, однако, требует привлечения существенных новых идей в первоначальное доказательство Гёделя. В окончательном виде этот результат был получен чешским математиком П. Пудлаком (в 1985 году).

Замечание 1.18. Одним из следствий второй теоремы Гёделя о неполноте является то, что непротиворечивость PA нельзя доказать средствами самой теории PA. Подчеркнём, что речь в этой теореме не идёт о том, что непротиворечивость PA может вызывать сомнения, а лишь о том, что обоснование (очевидным образом) верного факта непротиворечивости PA требует допущений, выходящих за рамки этой теории.

Ситуация менее очевидная с теорией ZFC: мы также верим в непротиворечивость ZFC, но предположения, на основании которых мы могли бы обосновать этот факт, не могут быть формализованы внутри самой ZFC, то есть должны выходить за рамки «обычной», общепринятой математики! Поэтому, в частности, в формулировке следствия 1.15 мы сделали оговорку относительно условия о непротиворечивости ZFC.[1]

Полное доказательство второй теоремы Гёделя о неполноте выходит за рамки данного курса.

2 Вычислимость и определимость

Первая теорема Гёделя о неполноте и теорема Гёделя–Россера будут выведены нами из одного результата, указывающего на фундаментальную связь между понятиями вычислимости и определимости в арифметике. Мы называем этот результат теоремой о Σ_1-определимости. Мы введём два класса арифметических формул: ограниченные формулы и Σ_1-формулы.

[1]В теории множеств рассматриваются дополнительные аксиомы, так называемые аксиомы больших кардиналов, из которых следует непротиворечивость ZFC. Однако эти аксиомы все-таки нельзя считать общепринятыми.

Определение 2.1. *Ограниченными* называются формулы, все вхождения кванторов в которые имеют вид

- $\forall x\,(x \leq t \to A(x))$ (сокращённо $\forall x{\leq}t\, A(x)$), или
- $\exists x\,(x \leq t \wedge A(x))$ (сокращённо $\exists x{\leq}t\, A(x)$),

где t — произвольный терм арифметического языка.

Множество всех ограниченных формул обозначаем Δ_0.

Определение 2.2. Σ_1-*формулами* называются формулы вида $\exists\vec{x}\, A(\vec{x},\vec{a})$, где $A \in \Delta_0$. Множество всех Σ_1-формул обозначаем Σ_1.

Нетрудно видеть, что всякая Δ_0-формула определяет в стандартной модели $\mathbb{N}$ некоторый разрешимый предикат, а Σ_1-формула — перечислимый предикат, то есть имеет место следующая лемма.

Лемма 2.3. *Пусть список $\vec{a} = (a_1,\ldots,a_k)$ содержит все свободные переменные формулы $A(\vec{a})$. Тогда*

(i) *если $A(\vec{a}) \in \Delta_0$, то множество $\{\vec{n} \in \mathbb{N}^k : \mathbb{N} \models A[\vec{n}]\}$ разрешимо;*

(ii) *если $A(\vec{a}) \in \Sigma_1$, то множество $\{\vec{n} \in \mathbb{N}^k : \mathbb{N} \models A[\vec{n}]\}$ перечислимо.*

Доказательство. Утверждение (ii) следует из (i) по теореме о проекции разрешимого множества. Утверждение (i) доказывается индукцией по построению A. Атомарные формулы арифметического языка очевидным образом определяют разрешимые множества, и разрешимые множества замкнуты относительно булевых операций.

Рассмотрим формулу $A(\vec{a}) = \forall x{\leq}t(\vec{a})\; B(x,\vec{a})$, где мы считаем, что список переменных $\vec{a}$ содержит все свободные переменные формулы B и терма t. Тогда истинностное значение формулы $A[\vec{n}]$ можно узнать, вычислив значение $m = t(\vec{n})$ и проверив полным перебором, что $\mathbb{N} \models B[i,\vec{n}]$ для каждого $i \leq m$. Аналогично рассматривается ограниченный квантор существования. ⊠

2.1 Теорема о Σ_1-определимости и вывод из неё первой теоремы Гёделя

Определение 2.4. Множество $P \subseteq \mathbb{N}^k$ Σ_1-*определимо в* $\mathbb{N}$, если существует $A(a_1,\ldots,a_k) \in \Sigma_1$ такая, что для всех $n_1,\ldots,n_k \in \mathbb{N}$

$$\langle n_1,\ldots,n_k\rangle \in P \iff \mathbb{N} \models A[n_1,\ldots,n_k].$$

Теорема 2.5 (о Σ_1-определимости). *$P \subseteq \mathbb{N}^k$ перечислимо $\iff$ P Σ_1-определимо в $\mathbb{N}$.*

Из этой теоремы мы получаем ряд важных следствий, включающих первую теорему Гёделя о неполноте.

Теорема 2.6. *Множество $Th(\mathbb{N})$ всех предложений A таких, что $\mathbb{N} \models A$, неперечислимо.*

Доказательство. Пусть $K \subseteq \mathbb{N}$ перечислимо и неразрешимо. По теореме о Σ_1-определимости найдётся формула $K(a)$ такая, что

$$n \in K \iff \mathbb{N} \models K[n] \iff \mathbb{N} \models K(\overline{n}).$$

Отсюда получаем

$$n \notin K \iff \mathbb{N} \not\models K(\overline{n}) \iff \mathbb{N} \models \neg K(\overline{n}).$$

Если $Th(\mathbb{N})$ перечислимо, то таково и $\{n \in \mathbb{N} : \mathbb{N} \models \neg K(\overline{n})\}$, так как по n эффективно восстанавливается формула $\neg K(\overline{n})$ (подстановка нумерала в фиксированную формулу является вычислимой операцией). Таким образом, будет перечислимым также и дополнение множества K, что противоречит теореме Чёрча–Поста. ⊠

Теорема 2.7. *Если T эффективно аксиоматизируема и $\mathbb{N} \models T$, то найдётся предложение A такое, что $T \nvdash A$ и $T \nvdash \neg A$.*

Доказательство. Поскольку $\mathbb{N} \models T$ имеем $T \subseteq Th(\mathbb{N})$, значит по теореме 2.6 найдётся $A \in Th(\mathbb{N})$ такое, что $T \nvdash A$. Так как $\mathbb{N} \not\models \neg A$, имеем $T \nvdash \neg A$. ⊠

2.2 Доказательство теоремы о Σ_1-определимости

Идея доказательства состоит в том, чтобы для каждой машины Тьюринга M выписать Σ_1-формулу $T_M(\vec{x})$, выражающую тот факт, что на входе, кодирующем $\vec{x}$, машина M завершает работу. Это достигается путём кодирования машин Тьюринга и описания их вычислений на арифметическом языке.

2.2.1 Обогащение модели с помощью Δ_0-определений

Искомую формулу удобно строить, обогащая сигнатуру арифметики новыми предикатными и функциональными символами с помощью Δ_0-определений.

Пусть Σ — сигнатура, содержащая арифметическую, и $\mathbb{N}_\Sigma$ — обогащение стандартной модели арифметики до некоторой модели сигнатуры Σ. Говорим, что модель $\mathbb{N}_\Sigma$ обладает *свойством ограниченности*, если для любого терма $t(\vec{a})$ сигнатуры Σ найдётся арифметический терм $t'(\vec{a})$ такой, что $\mathbb{N}_\Sigma \vDash \forall \vec{x}\, (t(\vec{x}) \leq t'(\vec{x}))$. *Ограниченными формулами* сигнатуры Σ называем формулы сигнатуры Σ, все вхождения кванторов в которые ограничены термами Σ. Множество всех таких формул обозначаем $\Delta_0(\Sigma)$.

Мы рассматриваем два типа определений:

- Определение предиката P формулой $A \in \Delta_0(\Sigma)$, обозначаемое

  $$P(\vec{a}) :\leftrightarrow A(\vec{a}).$$

  Сигнатура Σ расширяется новым предикатным символом P. В стандартной модели $\mathbb{N}$ символу P соответствует предикат

  $$P_\mathbb{N} \rightleftharpoons \{\vec{n} \in \mathbb{N}^k : \mathbb{N}_\Sigma \vDash A[\vec{n}]\}.$$

- Определение функции f формулой $F \in \Delta_0(\Sigma)$, обозначаемое

  $$f(\vec{a}) = b :\leftrightarrow F(\vec{a}, b).$$

  Сигнатура Σ расширяется новым функциональным символом f. В стандартной модели $\mathbb{N}$ символу f соответствует функция $f_\mathbb{N}$ с графиком

  $$F_\mathbb{N} \rightleftharpoons \{\langle \vec{n}, m\rangle : \mathbb{N}_\Sigma \vDash F[\vec{n}, m]\}.$$

  Такое определение считается корректным, если

  – $F_\mathbb{N}$ действительно задаёт график функции, то есть

  $$\mathbb{N}_\Sigma \vDash \forall \vec{x} \exists! y\, F(\vec{x}, y);$$

  – функция $f_\mathbb{N}$ ограничена некоторым термом $t(\vec{a})$ сигнатуры Σ, то есть

  $$\mathbb{N}_\Sigma \vDash \forall \vec{x}, y\, (F(\vec{x}, y) \rightarrow y \leq t(\vec{x})).$$

Следующие простейшие примеры показывают, как строить одни Δ_0-определения на основе других.

$$\begin{aligned} x \neq y \quad &:\leftrightarrow \quad \neg\, x = y \\ x < y \quad &:\leftrightarrow \quad x \leq y \wedge x \neq y \\ x \dot{-} y = z \quad &:\leftrightarrow \quad (y \leq x \wedge x = z + y) \vee (\neg\, y \leq x \wedge z = 0) \end{aligned}$$

Перевод $f \mapsto F$, $P \mapsto A$ задает интерпретацию модели $(\mathbb{N}_\Sigma; P_\mathbb{N}, f_\mathbb{N})$ в $\mathbb{N}_\Sigma$. Такие интерпретации I называем *ограниченными.* Как обычно, всякой формуле A в расширенной сигнатуре соответствует её перевод A^I в сигнатуру Σ.

Лемма 2.8. *Пусть $\mathbb{N}_\Sigma$ обладает свойством ограниченности. Тогда*

(i) $(\mathbb{N}_\Sigma; P_\mathbb{N}, f_\mathbb{N})$ *обладает тем же свойством;*

(ii) *если A — ограниченная формула расширенного языка, то перевод A^I эквивалентен $\Delta_0(\Sigma)$-формуле в модели $\mathbb{N}_\Sigma$.*

Доказательство. Утверждение (i) получается простой индукцией по построению терма t, с учётом монотонности всех функций сигнатуры арифметики.

Утверждение (ii) очевидно для случая определения предиката P, поскольку формула A^I получается заменой в A всех вхождений вида $P(t_1, \ldots, t_k)$ на $A(t_1, \ldots, t_k)$.

Для случая определения функции f рассуждаем индукцией по построению формулы A.

Сначала докажем утверждение для атомарных формул A. Такие формулы имеют вид $Q(t_1, \ldots, t_k)$, для некоторого предикатного символа Q сигнатуры Σ и некоторых термов $t_1, \ldots, t_k$ расширенной сигнатуры. Применяем индукцию по общему количеству вхождений символа f в термы $t_1, \ldots, t_k$.

Допустим, например, что f входит в t_1. Рассмотрим самое внутреннее такое вхождение; тогда t_1 имеет вид $t'_1(f(s_1, \ldots, s_n))$, где термы s_i не содержат символа f, и t'_1 имеет на одно вхождение f меньше, чем t_1. Поскольку функция f ограничена некоторым Σ-термом t, перевод $Q(t_1, \ldots, t_k)^I$ равносилен в $\mathbb{N}_\Sigma$ формуле

$$\exists x \leq t(s_1, \ldots, s_n)\, [(Q(t'_1(x), t_2, \ldots, t_k))^I \wedge F(s_1, \ldots, s_n, x)],$$

где $(Q(t'_1(x), t_2, \ldots, t_k))^I$ эквивалентна ограниченной формуле по предположению индукции.

Если формула A имеет вид $(A_1 \wedge A_2)$, $(A_1 \vee A_2)$, $\neg A_1$ или $(A_1 \to A_2)$, утверждение легко следует из предположения индукции.

Пусть A имеет вид $\forall x{\leq}s\, B(x)$. Воспользуемся частью (i) и рассмотрим арифметический терм s' такой, что

$$(\mathbb{N}_\Sigma; P_\mathbb{N}, f_\mathbb{N}) \vDash s \leq s'.$$

Тогда перевод A^I равносилен формуле

$$\forall x{\leq}s'\, ((x \leq s)^I \to B(x)^I).$$

Заметим, что формула $(x \leq s)^I$ ограничена как перевод атомарной формулы, а ограниченность $B(x)^I$ следует из предположения индукции. Случай ограниченного квантора существования рассматривается аналогично. $\boxtimes$

Следствие 2.9. *Композиция ограниченных интерпретаций ограничена.*

Теперь мы применим технику Δ_0-определений к формализации в арифметике вычислений машин Тьюринга.

2.2.2 Кодирование двоичных слов

Любое $x > 0$ однозначно представляется в виде

$$x = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2 + a_0,$$

где $a_0, \ldots, a_n \in \{0, 1\}$ и $a_n \neq 0$.

Слово $a_{n-1} \ldots a_0$ кодируем числом x, представимым как $1a_{n-1} \ldots a_0$ в двоичной записи. Таким образом, пустое слово Λ кодируется числом 1, а 0 не является кодом никакого двоичного слова.

Обозначим $|x| \rightleftharpoons n$. Заметим, что $|x|$ есть длина слова, кодируемого числом x.

Предикат $String(x)$ «быть двоичным словом» и функции $|x|$ (длина слова) и $x * y$ (конкатенация слов) определяются следующим образом.[2]

$$\begin{aligned} String(x) \quad &:\leftrightarrow \quad x \neq 0 \\ |x| = y \quad &:\leftrightarrow \quad (x = 0 \wedge y = 0) \vee (2^y \leq x \wedge x < 2^{y+1}) \\ x * y = z \quad &:\leftrightarrow \quad z = x \cdot 2^{|y|} + (y \dot{-} 2^{|y|}) \end{aligned}$$

[2]Для наглядности, ниже мы игнорируем разницу между алфавитами связанных и свободных переменных и пишем 2^x вместо $\exp(x)$.

Заметим, что значения функций $|x|$ и $x * y$ определены произвольно на аргументах, не являющихся словами (то есть, при $x = 0$ или $y = 0$).

2.2.3 Кодирование алфавита Σ

Пусть $\Sigma = \{C_0, \ldots, C_n\}$ — конечный алфавит. Зафиксируем некоторую константу c такую, что $2^c \geq n + 2$. Символы алфавита Σ и дополнительный символ разделителя «;» кодируем *байтами*, то есть двоичными словами длины c. Код (или *гёделев номер*) объекта O будем обозначать по традиции через $\ulcorner O \urcorner$. Положим, например, $\ulcorner C_i \urcorner \rightleftharpoons 2^c + i$ для $0 \leq i \leq n$ и $\ulcorner ; \urcorner \rightleftharpoons 2^c + n + 1$. Множество Σ и множество всех байтов определим следующим образом:

$$\begin{array}{rcl} \Sigma(x) & :\leftrightarrow & x = \ulcorner C_0 \urcorner \vee \cdots \vee x = \ulcorner C_n \urcorner \\ Byte(x) & :\leftrightarrow & String(x) \wedge |x| = \overline{c} \end{array}$$

2.2.4 Слова в алфавите Σ

В рамках выбранной кодировки будем называть *словом* конечную последовательность байтов. Σ-*словом* называем слово в данном нам алфавите Σ. Определяем последовательно следующие предикаты и функции: $Word(x)$ «x есть слово», $\|x\|$ длина слова x, $x \subseteq_w y$ «x есть подслово y», $x \in_w y$ «x есть элемент слова y», $Word_\Sigma(x)$ «x есть Σ-слово».

$$\begin{array}{rcl} Word(x) & :\leftrightarrow & String(x) \wedge \exists k \leq x \, |x| = \overline{c} \cdot k \\ \|x\| = y & :\leftrightarrow & (Word(x) \wedge \overline{c} \cdot y = |x|) \vee (\neg Word(x) \wedge y = 0) \\ x \subseteq_w y & :\leftrightarrow & Word(x) \wedge Word(y) \wedge \\ & & \exists v, w \leq y \, (Word(v) \wedge y = v * x * w) \\ x \in_w y & \leftrightarrow & Byte(x) \wedge x \subseteq_w y \\ Word_\Sigma(x) & :\leftrightarrow & Word(x) \wedge \forall y \leq x \, (y \in_w x \to \Sigma(y)) \end{array}$$

Заметим, что при выбранном кодировании конкатенация слов совпадает с конкатенацией соответствующих двоичных последовательностей.

2.2.5 Последовательности слов в алфавите Σ

Последовательность $\langle w_1, \ldots, w_s \rangle$ Σ-слов кодируем словом $w_1; w_2; \ldots; w_s$, где «;» — разделитель. Код пустой последовательности $\langle\ \rangle$ положим равным 0.

Заметим, что для любого слова $w \in \Sigma^*$, $\ulcorner \langle w \rangle \urcorner = \ulcorner w \urcorner$, в частности, $\ulcorner \langle \Lambda \rangle \urcorner = 1$.

Определяем следующие предикаты и функции: $Seq_\Sigma(x)$ «x есть последовательность Σ-слов», $x;y$ конкатенация последовательностей x и y, $x \subseteq_s y$ «x есть подпоследовательность y», $x \in_s y$ «x есть элемент последовательности y».

$$\begin{aligned}
Seq_\Sigma(x) \quad &:\leftrightarrow \quad Word(x) \wedge \forall y \in_w x\, (\Sigma(y) \vee y = \ulcorner ; \urcorner) \vee x = 0 \\
x;y = z \quad &:\leftrightarrow \quad (x = 0 \wedge z = y) \vee (y = 0 \wedge z = x) \vee \\
&\qquad\ (x \neq 0 \wedge y \neq 0 \wedge z = x * \ulcorner ; \urcorner * y) \\
x \subseteq_s y \quad &:\leftrightarrow \quad Seq_\Sigma(x) \wedge Seq_\Sigma(y) \wedge \\
&\qquad\ \exists u, v \leq y\, (Seq_\Sigma(u) \wedge Seq_\Sigma(v) \wedge y = u;x;v) \\
x \in_s y \quad &:\leftrightarrow \quad Word_\Sigma(x) \wedge x \subseteq_s y
\end{aligned}$$

2.2.6 Кодирование Машин Тьюринга

Зафиксируем произвольную машину Тьюринга M с рабочим алфавитом Σ и алфавитом состояний Q. Мы будем кодировать слова и последовательности слов в алфавите $\Sigma \cup Q \cup \{L, N, R\}$ и соответствующим образом фиксируем константу c (см. выше). Конечные множества символов и команд для данной машины легко определить формулами, перечисляющими их поэлементно. Пусть формула $\Sigma(x)$ определяет рабочий алфавит и $Q(x)$ — алфавит состояний. Формула $\Gamma(x) \rightleftharpoons Q(x) \vee \Sigma(x)$ задаёт их объединение.

Команда $q_i S_j \to q_k S_l \nu$, где $\nu \in \{L, N, R\}$, кодируется как

$$\ulcorner q_i \urcorner * \ulcorner S_j \urcorner * \ulcorner q_k \urcorner * \ulcorner S_l \urcorner * \ulcorner \nu \urcorner.$$

Формула $P(x)$ определяет множество команд M.

2.2.7 Конфигурации

Конфигурация машины M кодируется словом вида uqv, где u, v — слова в рабочем алфавите, слово v непусто, головка находится в состоянии $q \in Q$ и обозревает первый символ слова v. Таким образом, множество конфигураций определяется как

$$\begin{aligned}
Config(z) \quad :\leftrightarrow \quad & Word_\Gamma(z) \wedge \exists u, v, q \leq z \\
& (Word_\Sigma(u) \wedge Word_\Sigma(v) \wedge Q(q) \wedge \\
& v \neq 1 \wedge z = u * q * v)
\end{aligned}$$

2.2.8 Переходы

Следующая формула $Step_M(x, y)$ определяет отношение «машина M переходит за один шаг из конфигурации x в конфигурацию y». Тем самым, эта формула описывает применение одной команды из программы P заданной машины Тьюринга.

Пусть некоторая команда имеет вид $pa \to qb\nu$. В зависимости от направления движения головки разбираются один или два случая: если $\nu = N$, то конфигурация $upav$ переходит в $uqbv$. Если $\nu = L$ и слово слева от головки непусто (имеет вид uc), то конфигурация $upav$ переходит в $uqcbv$, иначе конфигурация имеет вид pav и переходит в $q\#bv$ (слева лента заполнена пробелами). Аналогично описывается движение головки направо, то есть случай $\nu = R$.

$$
\begin{aligned}
&Step_M(x, y) :\leftrightarrow \\
&Config(x) \wedge Config(y) \wedge \exists u, v, p, q, a, b, c \subseteq_w x * y \\
&\quad [Word_\Sigma(u) \wedge Word_\Sigma(v) \wedge Q(p) \wedge Q(q) \wedge \Sigma(a) \wedge \Sigma(b) \wedge \Sigma(c) \wedge \\
&\qquad [(x = u * p * a * v \wedge y = u * q * b * v \wedge P(p * a * q * b * \ulcorner N \urcorner)) \\
&\qquad \vee (x = u * c * p * a * v \wedge y = u * q * c * b * v \wedge P(p * a * q * b * \ulcorner L \urcorner)) \\
&\qquad \vee (x = p * a * v \wedge y = q * \ulcorner \# \urcorner * b * v \wedge P(p * a * q * b * \ulcorner L \urcorner)) \\
&\qquad \vee (x = u * p * a * v \wedge v \neq 1 \wedge y = u * b * q * v \wedge P(p * a * q * b * \ulcorner R \urcorner)) \\
&\qquad \vee (x = u * p * a \wedge y = u * b * q * \ulcorner \# \urcorner \wedge P(p * a * q * b * \ulcorner R \urcorner)) \\
&\qquad] \\
&\quad]
\end{aligned}
$$

Теперь мы можем определить понятие (протокола) вычисления машины M.

2.2.9 Вычисления

Определим отношения $Init_M(x, z)$ «z есть начальная конфигурация с входом x», $Stop_M(z)$ «z есть заключительная конфигурация», и $Comp_M(x, z)$ «z есть протокол завершающегося вычисления машины M на входе x».

Определения, приводимые ниже, говорят сами за себя.

$$\begin{aligned}
Init_M(x,z) \quad &:\leftrightarrow \quad Config(z) \wedge z = \ulcorner q_1 \urcorner * \ulcorner \# \urcorner * x \\
Stop_M(z) \quad &:\leftrightarrow \quad Config(z) \wedge \exists u, v \subseteq_w z\, (z = u * q_0 * v) \\
Comp_M(x,z) \quad &:\leftrightarrow \quad Seq_\Gamma(z) \wedge \exists v \in_s z\; Stop_M(v) \wedge \forall u, v, w \le z \\
&\qquad\quad (z = u; v; w \wedge Word_\Gamma(v) \to \\
&\qquad\qquad (Init_M(x,v) \vee \exists y \in_s u\; Step_M(y,v)))
\end{aligned}$$

2.2.10 Кодирование входа и предикат остановки

Наконец, мы должны вспомнить, что для машины Тьюринга, вычисляющей функцию натуральных аргументов, вместо последовательности чисел $\langle n_1, \dots, n_k \rangle$ мы подаём на вход слово $1^{n_1}\$\dots\$1^{n_k}$ в алфавите $\{1, \$\}$.

Пусть Σ содержит $1, \$$. Положим для любого $n \in \mathbb{N}$

$$code(n) \rightleftharpoons 1^n \rightleftharpoons 1\dots1 \qquad (n \text{ раз}).$$

Функция *code* определяется как

$$code(x) = y \quad :\leftrightarrow \quad Word(y) \wedge \|y\| = x \wedge \forall y {\in_w} x\; y = \ulcorner 1 \urcorner$$

Теперь мы можем выразить тот факт, что машина M на входе, кодирующем $\langle x_1, \dots, x_k \rangle$, завершает работу:

$$T_M(x_1, \dots, x_k) :\leftrightarrow \exists z\, Comp_M(code(x_1) * \ulcorner \$ \urcorner * \dots * \ulcorner \$ \urcorner * code(x_k), z)$$

Имеем:

$$\mathbb{N} \vDash T_M[n_1, \dots, n_k] \iff !M(n_1, \dots, n_k).$$

Тем самым доказательство теоремы о Σ_1-определимости завершено. Заметим, что построенная нами формула содержит один единственный неограниченный квантор существоваения. ⊠

3 Теорема Гёделя–Россера

Теорема Гёделя–Россера базируется на одном принципиальном факте, касающемся теории Q и содержащих её теорий. Как было отмечено выше, теория Q очень слаба для доказательства утверждений с неограниченными кванторами общности. С другой стороны, следующая теорема показывает, что Q достаточно сильна для доказательства всех истинных Σ_1-утверждений.

3.1 Σ_1-полнота

Определение 3.1. Теория T в арифметическом языке называется Σ_1-*полной*, если для любого предложения $A \in \Sigma_1$

$$\mathbb{N} \vDash A \ \Rightarrow\ T \vdash A.$$

Теорема 3.2. *Теория* Q Σ_1*-полна.*

Доказательство. Идея доказательства Σ_1-полноты проста: истинность любого Σ_1-предложения A может быть эффективно установлена с помощью процедуры, описанной в лемме 2.3. Это вычисление, по существу, представляет собой доказательство A в Q.

Более аккуратное доказательство получается из последовательности простых лемм, приводимой ниже.

Лемма 3.3. *Для любых* $m, n \in \mathbb{N}$*, в* Q *доказуемо*

(i) $\overline{m} + \overline{n} = \overline{m+n}$

(ii) $\overline{m} \cdot \overline{n} = \overline{m \cdot n}$

(iii) $\exp(\overline{n}) = \overline{\exp(n)}$

Доказательство. Каждое из утверждений доказывается «внешней» индукцией по n. То есть, мы используем индукцию для обоснования выводимости в Q, а не в рамках самой теории Q (где индукция не постулируется в качестве аксиомы). Напомним, что $\overline{0}$ есть 0 и $\overline{n+1}$ есть $S(\overline{n})$.

(i) Базис: $\overline{m} + 0 = \overline{m}$, по аксиоме 3.

Шаг индукции. Допустим, что в Q доказуемо $\overline{m} + \overline{n} = \overline{m+n}$. Достроим этот вывод до вывода формулы $\overline{m} + S(\overline{n}) = S(\overline{m+n})$:

1. $\overline{m} + \overline{n} = \overline{m+n}$ (гипотеза)
2. $S(\overline{m} + \overline{n}) = S(\overline{m+n})$ (по аксиоме равенства)
3. $\overline{m} + S(\overline{n}) = S(\overline{m} + \overline{n})$ (по аксиоме 3)
4. $\overline{m} + S(\overline{n}) = S(\overline{m+n})$ (из 2, 3)

Доказательства (ii) и (iii) аналогичны. ⊠

Лемма 3.4. *Для любого арифметического терма* $t(b_1, \ldots, b_m)$ *и любых* $k_1, \ldots, k_m, l \in \mathbb{N}$*,*

$$\mathbb{N} \vDash t(k_1, \ldots, k_m) = l \quad \Longrightarrow \quad \mathsf{Q} \vdash t(\overline{k}_1, \ldots, \overline{k}_m) = \overline{l}.$$

Доказательство. Внешняя индукция по построению t. Если t — переменная или константа 0, утверждение очевидно. Для составных термов утверждение получается из леммы 3.3 по предположению индукции. Например, если t имеет вид $t_1 + t_2$, то для некоторых $l_1, l_2 \in \mathbb{N}$ формулы $t_1(\overline{k}_1, \ldots, \overline{k}_m) = \bar{l}_1$ и $t_2(\overline{k}_1, \ldots, \overline{k}_m) = \bar{l}_2$ доказуемы. Мы достраиваем эти выводы следующей последовательностью формул:

1. $t_1(\overline{k}_1, \ldots, \overline{k}_m) = \bar{l}_1$ (гипотеза)
2. $t_2(\overline{k}_1, \ldots, \overline{k}_m) = \bar{l}_2$ (гипотеза)
3. $t_1(\overline{k}_1, \ldots, \overline{k}_m) + t_2(\overline{k}_1, \ldots, \overline{k}_m) = \bar{l}_1 + \bar{l}_2$ (по аксиоме равенства)
4. $\bar{l}_1 + \bar{l}_2 = \overline{l_1 + l_2}$ (лемма 3.3)
5. $t_1(\overline{k}_1, \ldots, \overline{k}_m) + t_2(\overline{k}_1, \ldots, \overline{k}_m) = \overline{l_1 + l_2}$ (по аксиоме равенства)

Функции последователя, умножения и экспоненты рассматриваются аналогично. ⊠

Лемма 3.5. *Для любых $m, n \in \mathbb{N}$,*

(i) *если $m \leq n$, то $\mathsf{Q} \vdash \overline{m} \leq \overline{n}$;*

(ii) *если $m \neq n$, то $\mathsf{Q} \vdash \neg\, \overline{m} = \overline{n}$;*

(iii) *если $m < n$, то $\mathsf{Q} \vdash \neg\, \overline{n} \leq \overline{m}$.*

Доказательство. (i) Внешняя индукция по n.

Базис. Для $n = m$ утверждение сводится к $\mathsf{Q} \vdash \overline{m} \leq \overline{m}$. Последнее легко доказать внешней индукцией по m на основе аксиом 6 и 7.

Шаг индукции. Пусть $m \leq n + 1$, тогда $m \leq n$ или $m = n + 1$. Если $m \leq n$, то $\mathsf{Q} \vdash \overline{m} \leq \overline{n}$ по предположению индукции. Если $m = n + 1$, то $\overline{m}$ совпадает графически с $S(\overline{n})$ и тем самым $\mathsf{Q} \vdash \overline{m} = S(\overline{n})$ по аксиоме равенства. Отсюда мы получаем

$$\mathsf{Q} \vdash \overline{m} \leq \overline{n} \vee \overline{m} = S(\overline{n}),$$

откуда следует $\mathsf{Q} \vdash \overline{m} \leq S(\overline{n})$ по аксиоме 7.

(ii) Считаем без ограничения общности, что $m < n$. Рассуждаем индукцией по m. Если $m = 0$, то $\overline{n}$ совпадает с $S(\overline{n-1})$, и результат следует из аксиом равенства. Если же $m > 0$, то по предположению индукции найдётся вывод в Q формулы $\neg\, \overline{m-1} = \overline{n-1}$. Продолжим этот вывод следующим образом:

1. $\neg\, \overline{m-1} = \overline{n-1}$ (гипотеза)
2. $\overline{m} = \overline{n} \to \overline{m-1} = \overline{n-1}$ (аксиома 2)
3. $\neg\, \overline{m} = \overline{n}$ (из 1, 2)

и получаем вывод в Q формулы $\neg\, \overline{m} = \overline{n}$.

(iii) Рассуждаем внешней индукцией по m.

Базис. Допустим $0 = m < n$. Тогда $\overline{n} \leq 0$ влечёт $\overline{n} = 0$ по аксиоме 6, откуда следует противоречие по утверждению (ii). Значит, $\mathsf{Q} \vdash \neg\, \overline{n} \leq 0$.

Шаг индукции. Допустим $m + 1 < n$. Тогда $\overline{n} \leq S(\overline{m})$ влечёт $\overline{n} \leq \overline{m} \vee \overline{n} = S(\overline{m})$ по аксиоме 7. Однако, $\overline{n} \leq \overline{m}$ влечёт противоречие по предположению индукции, а $\overline{n} = S(\overline{m})$ влечёт противоречие по утверждению (ii). Значит, $\mathsf{Q} \vdash \neg\, \overline{n} \leq S(\overline{m})$. $\boxtimes$

Лемма 3.6. *Для любого $m \in \mathbb{N}$, в Q доказуемо*

$$a \leq \overline{m} \leftrightarrow (a = 0 \vee \cdots \vee a = \overline{m}).$$

Доказательство получается из аксиом 6 и 7 внешней индукцией по m. $\boxtimes$

Лемма 3.7. *Для любой ограниченной формулы $A(b_1, \ldots, b_m)$ и любых $k_1, \ldots, k_m \in \mathbb{N}$,*

(i) $\mathbb{N} \vDash A(k_1, \ldots, k_m) \;\Rightarrow\; \mathsf{Q} \vdash A(\overline{k_1}, \ldots, \overline{k_m})$;

(ii) $\mathbb{N} \nvDash A(k_1, \ldots, k_m) \;\Rightarrow\; \mathsf{Q} \vdash \neg A(\overline{k_1}, \ldots, \overline{k_m})$.

Доказательство. Утверждения (i) и (ii) доказываем одновременно индукцией по построению формулы A. Рассмотрим следующие случаи.

1. A — атомарная формула вида $t_1(b_1, \ldots, b_m) = t_2(b_1, \ldots, b_m)$.

Если $\mathbb{N} \vDash A(k_1, \ldots, k_m)$, то для некоторого $l \in \mathbb{N}$, по лемме 3.4 мы имеем выводы формул $t_1(\overline{k_1}, \ldots, \overline{k_m}) = \overline{l}$ и $t_2(\overline{k_1}, \ldots, \overline{k_m}) = \overline{l}$ в Q. Отсюда получаем вывод $t_1(\overline{k_1}, \ldots, \overline{k_m}) = t_2(\overline{k_1}, \ldots, \overline{k_m})$, пользуясь аксиомами равенства.

Если $\mathbb{N} \nvDash A(k_1, \ldots, k_m)$, то для некоторых $l_1 \neq l_2$ имеем выводы формул $t_1(\overline{k_1}, \ldots, \overline{k_m}) = \overline{l_1}$ и $t_2(\overline{k_1}, \ldots, \overline{k_m}) = \overline{l_2}$ в Q по лемме 3.4. Лемма 3.5(ii) даёт вывод $\neg\overline{l_1} = \overline{l_2}$, откуда мы получаем вывод

$$\neg\, t_1(\overline{k_1}, \ldots, \overline{k_m}) = t_2(\overline{k_1}, \ldots, \overline{k_m})$$

, пользуясь аксиомами равенства.

2. A — атомарная формула вида $t_1(b_1, \dots, b_m) \le t_2(b_1, \dots, b_m)$.

Этот случай рассматривается аналогично, на основе леммы 3.5 (i) и (iii).

3. A имеет вид $B \to C$ или $\neg B$.

В этом случае утверждение получается непосредственно из предположения индукции для формул B и C.

4. A имеет вид $\forall v \le t\, B(v, b_1, \dots, b_m)$.

(i) Допустим $\mathbb{N} \vDash A(k_1, \dots, k_m)$. По лемме 3.4 найдётся $l \in \mathbb{N}$ такое, что в Q доказуемо $t(\overline{k_1}, \dots, \overline{k_m}) = \bar{l}$. Значит, для всех $k \le l$ имеем $\mathbb{N} \vDash B(k, k_1, \dots, k_m)$, и по предположению индукции получаем выводы формул $B(\overline{k}, \overline{k_1}, \dots, \overline{k_m})$ для каждого $k \le l$. Достроим их до вывода формулы $A(\overline{k_1}, \dots, \overline{k_m})$ следующим образом:

1.	$(a = 0 \vee \dots \vee a = \bar{l}) \to B(a, \overline{k_1}, \dots, \overline{k_m})$	(предп. индукции)
2.	$a \le \bar{l} \to B(a, \overline{k_1}, \dots, \overline{k_m})$	(1, лемма 3.6)
3.	$a \le t(\overline{k_1}, \dots, \overline{k_m}) \to B(a, \overline{k_1}, \dots, \overline{k_m})$	(2, равенство)
4.	$\forall v\, (v \le t(\overline{k_1}, \dots, \overline{k_m}) \to B(v, \overline{k_1}, \dots, \overline{k_m}))$	(3)

(ii) Допустим $\mathbb{N} \nvDash A(k_1, \dots, k_m)$. Тогда для некоторого

$$k \le l = t(k_1, \dots, k_m)$$

имеем $\mathbb{N} \nvDash B(k, k_1, \dots, k_m)$, а значит

$$\mathsf{Q} \vdash \neg B(\overline{k}, \overline{k_1}, \dots, \overline{k_m})$$

по предположению индукции. Достраиваем этот вывод до вывода формулы $\neg A(\overline{k_1}, \dots, \overline{k_m})$:

1.	$\neg B(\overline{k}, \overline{k_1}, \dots, \overline{k_m})$	(гипотеза)
2.	$\overline{k} \le \bar{l} \wedge \neg B(\overline{k}, \overline{k_1}, \dots, \overline{k_m})$	(1, лемма 3.5(i))
3.	$\overline{k} \le t(\overline{k_1}, \dots, \overline{k_m}) \wedge \neg B(\overline{k}, \overline{k_1}, \dots, \overline{k_m})$	(2, лемма 3.4)
4.	$\exists v\, (v \le t(\overline{k_1}, \dots, \overline{k_m}) \wedge \neg B(v, \overline{k_1}, \dots, \overline{k_m}))$	(3)

Завершим доказательство теоремы 3.2. Рассуждаем индукцией по построению Σ_1-формулы $A(b_1, \dots, b_m)$. Если $A \in \Delta_0$, воспользуемся леммой 3.7 (i). Если A имеет вид $\exists v A_0(v, b_1, \dots, b_m)$ и $\mathbb{N} \vDash A(k_1, \dots, k_m)$, то для некоторого k имеем $\mathbb{N} \vDash A_0(k, k_1, \dots, k_m)$. По предположению индукции в Q доказуемо $A_0(\overline{k}, \overline{k_1}, \dots, \overline{k_m})$. Отсюда логически следует $\exists v A_0(v, \overline{k_1}, \dots, \overline{k_m})$. ⊠

Следствие 3.8.

1. *Любая арифметическая теория T, содержащая Q, Σ_1-полна.*

2. *Арифметика PA Σ_1-полна.*

3.2 Доказательство теоремы Гёделя–Россера

Доказательству этой теоремы предпошлём следующую лемму.

Лемма 3.9. *Пусть $A, B \subseteq \mathbb{N}$ перечислимы и $A \cap B = \varnothing$. Тогда найдётся Σ_1-формула $\varphi(a)$ такая, что для любого $n \in \mathbb{N}$*

(i) $n \in A \;\Rightarrow\; \mathsf{Q} \vdash \varphi(\overline{n})$,

(ii) $n \in B \;\Rightarrow\; \mathsf{Q} \vdash \neg\varphi(\overline{n})$.

Доказательство. По теореме о Σ_1-определимости найдутся Δ_0-формулы A_0 и B_0 такие, что

$$\begin{array}{lcl} n \in A & \iff & \mathbb{N} \vDash \exists x\, A_0(\overline{n}, x), \\ n \in B & \iff & \mathbb{N} \vDash \exists y\, B_0(\overline{n}, y). \end{array}$$

Для любой формулы C и терма t обозначим

$$\forall x < t\, C(x) \;\stackrel{\text{def}}{\iff}\; \forall x \le t\, (x = t \vee C(x)).$$

Положим теперь

$$\varphi(a) \;\stackrel{\text{def}}{\iff}\; \exists x\, (A_0(a, x) \wedge \forall y < x\, \neg B_0(a, y)).$$

Неформально, $\varphi(a)$ утверждает, что работа алгоритма, принимающего множество A, на входе a заканчивается раньше работы алгоритма, принимающего B («Россеровское сравнение свидетелей»).

Если $n \in A$, то для некоторого m истинна формула

$$A_0(\overline{n}, \overline{m}) \wedge \forall y < \overline{m}\, \neg B_0(\overline{n}, y).$$

По теореме о Σ_1-полноте арифметики Q получаем, что эта формула доказуема в Q, откуда $\mathsf{Q} \vdash \varphi(\overline{n})$.

Если $n \in B$, то для некоторого m истинна формула

$$B_0(\overline{n}, \overline{m}) \wedge \forall y \le \overline{m}\, \neg A_0(\overline{n}, y). \qquad (*)$$

По теореме о Σ_1-полноте арифметики Q получаем, что эта формула доказуема в Q. Отсюда следует, что $\mathsf{Q} \vdash \neg\varphi(\overline{n})$. Поясним это следующим рассуждением, которое легко преобразовать в формальный вывод противоречия из гипотезы $\varphi(\overline{n})$ в Q:

Допустим $\varphi(\overline{n})$. Тогда для некоторого x

$$A_0(\overline{n}, x) \wedge \forall y < x\, \neg B_0(\overline{n}, y).$$

Если $x \leq \overline{m}$, то имеем $\neg A_0(\overline{n}, x)$ в силу $(*)$, что противоречит $A_0(\overline{n}, x)$. Если же $\overline{m} < x$, то имеем $\neg B_0(\overline{n}, \overline{m})$, что противоречит $B_0(\overline{n}, \overline{m})$ из $(*)$. Осталось заметить, что в Q выводимо (в силу аксиомы Q и очевидного $\overline{m} \leq \overline{m}$)

$$\forall x\, (x \leq \overline{m} \vee \overline{m} < x),$$

откуда следует требуемое противоречие. ⊠

Доказательство теоремы Гёделя–Россера. Пусть A, B — неотделимая пара перечислимых подмножеств $\mathbb{N}$. Воспользуемся леммой и рассмотрим соответствующую формулу φ. Для данной теории T рассмотрим множества

$$\begin{array}{rcl} A' & \rightleftharpoons & \{n \in \mathbb{N} : T \vdash \varphi(\overline{n})\}, \\ B' & \rightleftharpoons & \{n \in \mathbb{N} : T \vdash \neg\varphi(\overline{n})\}. \end{array}$$

Поскольку T эффективно аксиоматизируема, оба эти множества перечислимы. Так как T непротиворечива, $A' \cap B' = \varnothing$. По лемме мы также имеем $A \subset A'$ и $B \subset B'$. Докажем, что найдётся $n \notin A' \cup B'$. Действительно, в противном случае A' и B' разбивают $\mathbb{N}$ (взаимно дополнительны) и по теореме Чёрча–Поста должны быть разрешимыми. Но это невозможно, так как в этом случае они отделяли бы A от B.

Если $n \notin A' \cup B'$, то очевидно $T \nvdash \varphi(\overline{n})$ и $T \nvdash \neg\varphi(\overline{n})$, то есть T неполна. Заметим, что построенное нами независимое утверждение принадлежит классу Σ_1 (а его отрицание классу Π_1). ⊠

3.3 Неразрешимость арифметических теорий и исчисления предикатов

Теорема 3.10. *Пусть теория T удовлетворяет условиям теоремы Гёделя–Россера. Тогда множество доказуемых и множество опровержимых в T предложений неотделимы.*

Доказательство. Обозначим

$$\begin{aligned} P_T &\rightleftharpoons \{\varphi : T \vdash \varphi\}, \\ R_T &\rightleftharpoons \{\varphi : T \vdash \neg\varphi\}. \end{aligned}$$

В силу непротиворечивости T эти множества не пересекаются. Допустим, что некоторое разрешимое множество C отделяет P_T от R_T, то есть $P_T \subseteq C$ и $C \cap R_T = \varnothing$.

Как и в теореме Гёделя–Россера, рассмотрим неотделимую пару перечислимых множеств A, B, воспользуемся леммой и рассмотрим соответствующую формулу φ. Если $n \in A$, то $T \vdash \varphi(\overline{n})$, то есть $\varphi(\overline{n}) \in P_T$ и $\varphi(\overline{n}) \in C$. Если же $n \in B$, то $T \vdash \neg\varphi(\overline{n})$, то есть $\varphi(\overline{n}) \in R_T$ и $\varphi(\overline{n}) \notin C$. Значит, множество $\{n \in \mathbb{N} : \varphi(\overline{n}) \in C\}$ отделяет A от B. Это множество разрешимо, поскольку по n эффективно восстанавливается формула $\varphi(\overline{n})$ (для фиксированной φ). $\boxtimes$

Следствие 3.11. *Всякая теория T, удовлетворяющая условиям теоремы Гёделя–Россера, неразрешима.*

Следствие 3.12. *Неразрешимы следующие теории:* Q, PA, ZFC.

Замечание 3.13. Заметим, что из неразрешимости теории T, удовлетворяющей условиям теоремы Гёделя–Россера, следует её неполнота (поскольку полные эффективно аксиоматизированные теории разрешимы).

Следствие 3.14. *Исчисление предикатов в арифметическом языке неразрешимо.*

Доказательство. Пусть $\tilde{\mathsf{Q}}$ означает конъюнкцию всех нелогических аксиом теории Q (включая аксиомы равенства). Для любого арифметического предложения A, по теореме о дедукции, $\mathsf{Q} \vdash A \iff \vdash \tilde{\mathsf{Q}} \to A$. Таким образом, для проверки выводимости A в Q было бы достаточно проверить выводимость формулы $\tilde{\mathsf{Q}} \to A$ в чистом исчислении предикатов, но первое невозможно. $\boxtimes$

Замечание 3.15. Последнее следствие, полученное американским логиком А. Чёрчем, показывает неразрешимость проблемы, которую Д. Гильберт считал одной из центральных проблем в математической логике (так называемая «Entscheidungsproblem»): не существует алгоритма, проверяющего данную формулу логики первого порядка на общезначимость.

Упражнение 3.16. *Докажите, что существует конечная сигнатура без функциональных символов и констант, для которой исчисление предикатов неразрешимо.*