Теорема Гёделя о неполноте лекция 13

Лев Дмитриевич Беклемишев
http://lpcs.math.msu.su/vml2008

lbekl@yandex.ru

24.04.2008

Вычислимость и определимость

Опр.

Ограниченные формулы — все вхождения кванторов имеют вид
$\forall x{\le}t\, A(x) \rightleftharpoons \forall x\, (x \le t \to A(x))$ или
$\exists x{\le}t\, A(x) \rightleftharpoons \exists x\, (x \le t \wedge A(x))$.
$\Delta_0 =$ множество ограниченных формул.

Опр.

Σ_1-*формулы* имеют вид $\exists\vec{x}\, A(\vec{x}, \vec{a})$, где $A \in \Delta_0$.

Теорема об определимости

Опр.

Множество $P \subseteq \mathbb{N}^k$ *Σ_1-определимо в $\mathbb{N}$*, если для некоторой $A(a_1, \ldots, a_k) \in \Sigma_1$

$$\langle x_1, \ldots, x_k \rangle \in P \iff \mathbb{N} \models A[x_1, \ldots, x_n].$$

Теорема.

$P \subseteq \mathbb{N}^k$ *перечислимо* $\iff$ P *Σ_1-определимо в* $\mathbb{N}$.

Теорема.

Множество TA *всех предложений* A *таких, что* $\mathbb{N} \models A$, *неперечислимо.*

Доказательство.

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

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

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

Если TA перечислимо, то таково и $\{n \in \mathbb{N} : \mathbb{N} \models \neg K(\overline{n})\}$, т.к. по n эффективно восстанавливается формула $\neg K(\overline{n})$. Т.о. перечислимо дополнение K, что противоречит теореме Поста.

Теорема Гёделя

Теорема.

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

Доказательство.

В силу корректности $T \subseteq \mathrm{TA}$, значит найдётся $A \in \mathrm{TA}$ такое, что $T \nvdash A$. Т.к. $\mathbb{N} \not\models \neg A$ имеем $T \nvdash \neg A$.

Следствие.
PA неполна.

Следствие.
ZFC неполна, если она корректна.

Доказательство теоремы о Σ_1-определимости

Идея: для каждой машины Тьюринга M надо 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.

Обогащение модели с помощью Δ_0-определений

Два типа определений:

- Определение предиката P:

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

  где $A \in \Delta_0$.

- Определение функции f

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

  где $F \in \Delta_0$ и для некоторого терма $t(\vec{a})$

$$F(\vec{a}, b) \to b \leq t(\vec{a}).$$

Отображение $f \mapsto F$, $P \mapsto A$ задает интерпретацию модели $(\mathbb{N}; P, f)$ в $\mathbb{N}$. Такие интерпретации I называем *ограниченными*.

Всякой формуле A в расширенной сигнатуре соответствует её перевод A^I в язык арифметики.

Теорема.

Если A — ограниченная формула расширенного языка, а I — ограниченная интерпретация, то перевод A^I эквивалентен ограниченной формуле.

Доказательство.

$s(f(x)) = y \leftrightarrow \exists z \leq t(x)\,(f(x) = z \wedge s(z) = y).$

Следствие.

Композиция ограниченных интерпретаций ограничена.

Примеры

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

Двоичное разложение

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 $a_n \neq 0$.

Обозначения:

- $bit(x, i) \rightleftharpoons a_i$ есть $i + 1$-й бит x;
- $|x| \rightleftharpoons n$.

Кодирование двоичных слов

Слово $a_{n-1} \dots a_0$ кодируем числом $1a_{n-1} \dots a_0$ в двоичной записи. Пустое слово Λ кодируется числом 1.

Замечание.
$|x|$ есть длина двоичной записи слова, кодируемого числом x.

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

Кодирование алфавита Σ

Пусть $\Sigma = \{C_0, \dots, C_n\}$.
Возьмём c: $2^c \geq n+2$.
Положим $\ulcorner C_i \urcorner \rightleftharpoons 2^c + i$ для $0 \leq i \leq n$ и
$\ulcorner ; \urcorner \rightleftharpoons 2^c + n + 1$ (разделитель).

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

Слова в алфавите Σ

Слово = последовательность байтов

Σ-слово = последовательность байтов из Σ

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

Последовательности слов

Посл-ть $\langle w_1, \ldots, w_s \rangle$ Σ-слов кодируем словом $w_1; w_2; \ldots; w_s$, где ; — разделитель. Код пустой посл-ти $\langle\ \rangle$ есть 0.

Замечание.
Для любого $w \in \Sigma^*$, $\ulcorner \langle w \rangle \urcorner = \ulcorner w \urcorner$, in particular, $\ulcorner \langle \Lambda \rangle \urcorner = 1$.

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

Кодирование Машин Тьюринга

$\Sigma(x)$ рабочий алфавит
$Q(x)$ алфавит состояний
$\Gamma(x) \rightleftharpoons Q(x) \vee \Sigma(x)$
$P(x)$ множество команд
$Word_{\Sigma}(x)$ слово в рабочем алфавите

Конфигурации

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

(1 есть код пустого слова)

Переходы

$$
\begin{array}{l}
\mathit{Step}_M(x, y) :\leftrightarrow \\
\mathit{Config}(x) \wedge \mathit{Config}(y) \wedge \\
\exists u, v, p, q, a, b, c \subseteq_w x * y \\
\quad [\mathit{Word}_\Sigma(u) \wedge \mathit{Word}_\Sigma(v) \wedge Q(p) \wedge Q(q) \wedge A(a) \wedge A(b) \wedge A(c) \wedge \\
\qquad [(x = u * p * a * v \wedge y = u * q * b * v \wedge P(p * a * q * b * \ulcorner N \urcorner)) \\
\qquad \vee (x = u * c * p * a * v \wedge y = u * q * c * b * v \wedge P(p * a * q * b * \ulcorner L \urcorner)) \\
\qquad \vee (x = p * a * v \wedge y = q * \ulcorner \# \urcorner * b * v \wedge P(p * a * q * b * \ulcorner L \urcorner)) \\
\qquad \vee (x = u * p * a * v \wedge v \neq 1 \wedge y = u * b * q * v \wedge P(p * a * q * b * \ulcorner R \urcorner)) \\
\qquad \vee (x = u * p * a \wedge y = u * b * q * \ulcorner \# \urcorner \wedge P(p * a * q * b * \ulcorner R \urcorner)) \\
\qquad] \\
\quad]
\end{array}
$$

Вычисления

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

Кодирование входа

Пусть Σ содержит $1, \$$.

$code(n) \rightleftharpoons 1 \ldots 1$ (n раз)

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

Предикат остановки

$T_M(a_1, \ldots, a_n) :\leftrightarrow$

$\exists z\ Comp_M(code(a_1)\$\ldots\$code(a_n), z)$

Имеем:

$$\mathbb{N} \vDash T_M[x_1, \ldots, x_n] \iff !M(x_1, \ldots, x_n).$$