Mathematical Logic (XII)

Yijia Chen

1. The Undecidability of Arithmetic

For the alphabet $\mathcal{A} = \{|\}$ we consider the halting problem

$$\Pi_{\text{halt}} := \{ w_{\mathbb{P}} \mid \mathbb{P} \text{ a program over } \mathcal{A} \text{ and } \mathbb{P} : \square \to \text{halt} \}.$$

Let $\mathbb{P}$ be a program over $\mathcal{A}$. Assume that $\mathbb{P}$ consists of instructions $\alpha_0, \dots, \alpha_k$. Let n be the maximum index i such that R_i is used by $\mathbb{P}$. Then a configuration of $\mathbb{P}$ is an $(n+2)$-tuple

$$(L, m_0, \dots, m_n),$$

where $L \leqslant k$ and $m_0, \dots, m_n \in \mathbb{N}$, meaning that α_L is the instruction to be executed next and every register R_i contains m_i, i.e., the word $\underbrace{||\cdots|}_{m_i \text{ times}}$.

We have shown:

Lemma 1.1. *From the above program $\mathbb{P}$ we can compute an S_{ar}-formula*

$$\chi_{\mathbb{P}}(x_0, \dots, x_n, z, y_0, \dots, y_n)$$

such that for all $\ell_0, \dots, \ell_n, L, m_0, \dots, m_n \in \mathbb{N}$

$$\mathfrak{N} \models \chi_{\mathbb{P}}[\ell_0, \dots, \ell_n, L, m_0, \dots, m_n]$$

if and only if $\mathbb{P}$, beginning with the configuration $(0, \ell_0, \dots, \ell_n)$, after finitely many steps, reaches the configuration $(L, m_0, \dots, m_n)$. ⊣

Theorem 1.2. $\text{Th}(\mathfrak{N})$ *is not R-decidable.*

Proof: Let $\mathbb{P}$ be a program over $\mathbb{A} = \{|\}$. Using the formula $\chi_{\mathbb{P}}$ in Lemma 1.1, we define

$$\varphi_{\mathbb{P}} := \exists y_0 \cdots \exists y_n \exists \chi_{\mathbb{P}}(0, \dots, 0, \bar{k}, y_0, \dots, y_n),$$

where $\bar{k} := \underbrace{1 + \cdots + 1}_{k \text{ times}}$. Then By Lemma 1.1, we conclude that $\mathfrak{N} \models \varphi_{\mathbb{P}}$ if and only if $\mathbb{P}$, beginning with the initial configuration $(0, 0, \dots, 0)$, after finitely many steps, reaches the configuration $(k, m_0, \dots, m_n)$, i.e., $\mathbb{P} : \square \to \text{halt}$. Thus, if $\text{Th}(\mathfrak{N})$ is R-decidable, so is Π_{halt}. □

Proof of Lemma 1.1. Recall that $\chi_{\mathbb{P}}$ expresses in $\mathfrak{N}$ that there is an $s \in \mathbb{N}$ and a sequence of configurations $C_0, \dots, C_s$ such that

- $C_0 = (0, x_0, \dots, x_n)$,
- $C_s = (z, y_0, \dots, y_n)$,
- for all $i < s$ we have $C_i \xrightarrow{\mathbb{P}} C_{i+1}$, i.e., from the configuration C_i the program $\mathbb{P}$ will reach C_{i+1} in *one step*.

We slightly rewrite the above formulation as that there is an $s \in \mathbb{N}$ and a sequence of natural numbers

$$\underbrace{a_0, \ldots, a_{n+1}}_{C_0} \underbrace{a_{n+2}, \ldots, a_{(n+2)+(n+1)}}_{C_1} \cdots \underbrace{a_{s\cdot(n+2)}, \ldots, a_{s\cdot(n+2)+(n+1)}}_{C_s} \tag{1}$$

such that

- $a_0 = 0$, $a_1 = x_0, \ldots, a_{n+1} = x_n$,
- $a_{s\cdot(n+2)} = z$, $a_{s\cdot(n+2)+1} = y_0, \ldots, a_{s\cdot(n+2)+(n+1)} = y_n$,
- for all $i < s$ we have

$$\Big(a_{i\cdot(n+2)}, \ldots, a_{i\cdot(n+2)+(n+1)}\Big) \xrightarrow{\mathbb{P}} \Big(a_{(i+1)\cdot(n+2)}, \ldots, a_{(i+1)\cdot(n+2)+(n+1)}\Big).$$

Observe that the length of the sequence (1) is unbounded, so we cannot quantify it directly in $\mathfrak{N}$. So we need the following beautiful (elementary) number-theoretic tool.

Lemma 1.3 (Gödel's β-function)**.** *There is a function* $\beta : \mathbb{N}^s \to \mathbb{N}$ *with the following properties.*

(i) *For every* $r \in \mathbb{N}$ *and every sequence* $(a_0, \ldots, a_r)$ *in* $\mathbb{N}$ *there exist* $t, p \in \mathbb{N}$ *such that for all* $i \leqslant r$

$$\beta(t, p, i) = a_i.$$

(ii) β *is definable in* $L^{S_{ar}}$. *That is, there is an* S_{ar}*-formula* $\varphi_\beta(x, y, z, w)$ *such that for all* $t, q, i, a \in \mathbb{N}$

$$\mathfrak{N} \models \varphi_\beta[t, q, i, a] \quad\Longleftrightarrow\quad \beta(t, q, i) = a.$$

$\dashv$

Equipped with the above β function and the formula φ_β, we define the desired $\chi_{\mathbb{P}}$ as follows.

$$\begin{aligned}
\exists p \exists t \exists s \Big(& \varphi_\beta(t, p, 0, 0) \wedge \varphi_\beta(t, p, 1, x_0) \wedge \cdots \wedge \varphi_\beta(t, p, \overline{n+1}, x_n) \\
& \wedge \varphi_\beta(t, p, s \cdot \overline{n+2}, z) \wedge \varphi_\beta(t, p, s \cdot \overline{n+2} + 1, y_0) \\
& \qquad \wedge \cdots \wedge \varphi_\beta(t, p, s \cdot \overline{n+2} + \overline{n+1}, y_n) \\
& \wedge \forall i \Big(i < s \to \forall u \forall u_0 \cdots \forall u_n \forall u' \forall u'_0 \cdots \forall u'_n \\
& \qquad (\varphi_\beta(t, p, i \cdot \overline{n+2}, u) \wedge \varphi_\beta(t, p, i \cdot \overline{n+2} + 1, u_0) \\
& \qquad\quad \wedge \cdots \wedge \varphi_\beta(t, p, i \cdot \overline{n+2} + \overline{n+1}, u_n) \\
& \qquad \wedge \varphi_\beta(t, p, (i+1) \cdot \overline{n+2}, u') \wedge \varphi_\beta(t, p, (i+1) \cdot \overline{n+2} + 1, u'_0) \\
& \qquad\quad \wedge \cdots \wedge \varphi_\beta(t, p, (i+1) \cdot \overline{n+2} + \overline{n+1}, u'_n) \\
& \qquad \to \text{“}(u, u_0, \ldots, u_n) \xrightarrow{\mathbb{P}} (u', u'_0, \ldots, u'_n)\text{”} \Big).
\end{aligned}$$

Here,

$$\text{“}(u, u_0, \ldots, u_n) \xrightarrow{\mathbb{P}} (u', u'_0, \ldots, u'_n)\text{”}$$

stands for a formula describing one-step computation of $\mathbb{P}$ from configuration $(u, u_0, \ldots, u_n)$ to configuration $(u', u'_0, \ldots, u'_n)$. Such a formula can be defined as a conjunction

$$\psi_0 \wedge \cdots \wedge \psi_{k-1}.$$

Recall that the program $\mathbb{P}$ consists of instructions $\alpha_0, \dots, \alpha_k$ where the last α_k is the halt instruction. Thus, say α_j is

$$j \textbf{ LET } R_1 = R_1 + |,$$

then we let

$$\psi_j := u \equiv \bar{j} \to \Big(u' \equiv u + 1 \wedge u_0' \equiv u_0 \wedge u_1' \equiv u_1 + 1 \wedge u_2' \equiv u_2 \wedge \cdots \wedge u_n' \equiv u_n\Big).$$

The remaining details are left to the reader. □

Using Lemma 1.1 we can prove similarly:

Theorem 1.4. *Let* $r \geqslant 1$.

(i) *Let* $\mathscr{R} \subseteq \mathbb{N}^r$ *be an R-decidable relation. Then there is an* $L^{S_{ar}}$*-formula* $\varphi(v_0, \dots, v_{r-1}) \in \mathbb{N}$ *such that for all* $\ell_0, \dots, \ell_{r-1} \in \mathbb{N}$

$$\big(\ell_0, \dots, \ell_{r-1}\big) \in \mathscr{R} \iff \mathfrak{N} \models \varphi(\bar{\ell}_0, \dots, \bar{\ell}_{r-1}).$$

(ii) *Let* $f : \mathbb{N}^r \to \mathbb{N}$ *be an R-computable function. Then there is an* $L^{S_{ar}}$*-formula* $\varphi(v_0, \dots, v_{r-1}, v_r)$ *such that for all* $\ell_0, \dots, \ell_{r-1}, \ell_r \in \mathbb{N}$

$$f\big(\ell_0, \dots, \ell_{r-1}\big) = \ell_r \iff \mathfrak{N} \models \varphi(\bar{\ell}_0, \dots, \bar{\ell}_{r-1}, \bar{\ell}_r).$$

Therefore,

$$\mathfrak{N} \models \exists^{=1} v_r \ \varphi(\bar{\ell}_0, \dots, \bar{\ell}_{r-1}, v_r),$$

where $\exists^{=1} x \ \theta(x)$ *denotes the formula*

$$\exists x \Big(\theta(x) \wedge \forall y \big(\varphi(y) \to y \equiv x\big)\Big).$$ ⊣

2. Gödel's Incompleteness Theorems

Let $\Phi \subseteq L_0^{S_{ar}}$.

Definition 2.1. Let $r \geqslant 1$.

(i) A relation $\mathscr{R} \subseteq \mathbb{N}^r$ is *representable in* Φ if there is an $L^{S_{ar}}$-formula $\varphi(v_0, \dots, v_{r-1})$ such that for all $n_0, \dots, n_{r-1} \in \mathbb{N}$

$$\big(n_0, \dots, n_{r-1}\big) \in \mathscr{R} \implies \Phi \vdash \varphi(\bar{n}_0, \dots, \bar{n}_{r-1}),$$
$$\big(n_0, \dots, n_{r-1}\big) \notin \mathscr{R} \implies \Phi \vdash \neg\varphi(\bar{n}_0, \dots, \bar{n}_{r-1}).$$

(ii) A function $F : \mathbb{N}^r \to \mathbb{N}$ is *representable in* Φ if there is an $L^{S_{ar}}$-formula $\varphi(v_0, \dots, v_{r-1}, v_r)$ such that for all $n_0, \dots, n_{r-1}, n_r \in \mathbb{N}$

$$f(n_0, \dots, n_{r-1}) = n_r \implies \Phi \vdash \varphi(\bar{n}_0, \dots, \bar{n}_{r-1}, \bar{n}_r),$$
$$f(n_0, \dots, n_{r-1}) \neq n_r \implies \Phi \vdash \neg\varphi(\bar{n}_0, \dots, \bar{n}_{r-1}, \bar{n}_r).$$

Moreover,

$$\Phi \vdash \exists^{=1} v_r \ \varphi(\bar{n}_0, \dots, \bar{n}_{r-1}, v_r).$$ ⊣

Lemma 2.2. (i) *If* Φ *is inconsistent, then every relation over* $\mathbb{N}$ *and every function over* $\mathbb{N}$ *is representable in* Φ.

(ii) *Let* $\Phi \subseteq \Phi' \subseteq L_0^{S_{ar}}$. *Then every relation representable in* Φ *is also representable in* Φ'. *Similarly, every function representable in* Φ *is representable in* Φ' *as well.*

(iii) Let Φ be consistent. If Φ is R-decidable, then every relation representable in Φ is R-decidable, and every function representable in Φ is R-computable.

Proof: Routine. □

Definition 2.3. Φ *allows representations* if all R-decidable relations and all R-computable functions over $\mathbb{N}$ are representable in Φ.

By Theorem 1.4:

Theorem 2.4. Th($\mathfrak{N}$) *allows representations.* ⊣

A standard but tedious analysis shows that the proof of Theorem 1.4 can be "carried" out in Φ_{PA}.

Theorem 2.5. Φ_{PA} *allows representations.* ⊣

Recall that we have exhibited the so-called Gödel numbering of register programs. For later purposes, we do the same for $L^{S_{\mathrm{ar}}}$-formulas. Let

$$\varphi_0, \varphi_1, \ldots, \tag{2}$$

be an *effective* enumeration of all $L^{S_{\mathrm{ar}}}$-formulas without repetition. That is, there is a program that prints out the sequence (2). Then for every $\varphi \in L^{S_{\mathrm{ar}}}$ we let

$$[\varphi] := n \quad \text{where } \varphi = \varphi_n.$$

Observe that both

$$n \mapsto \varphi_n \text{ and } \varphi \mapsto [\varphi]$$

are R-computable.

Next time we will show:

Theorem 2.6 (Fixed Point Theorem)**.** *Assume that Φ allows representations. Then for every $\psi \in L_1^{S_{\mathrm{ar}}}$, there is an S_{ar}-sentence φ such that*

$$\Phi \vdash \varphi \leftrightarrow \psi(\overline{[\varphi]}).$$

View $\psi(x)$ is a property. Then Theorem 2.6 intuitively says

I, i.e., φ, satisfies the property ψ.

3. Exercises

Exercise 3.1. Prove Theorem 2.5. ⊣