

Mathematical Logic (XIII)

Yijia Chen

1. The Undecidability of Arithmetic

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

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

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 $L \leq k$ 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 $\underbrace{|\dots|}_{m_i \text{ times}}$.

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) . \dashv

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

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

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

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., $\mathbb{P} : \square \rightarrow \text{halt}$. Thus, if $\text{Th}(\mathfrak{N})$ is R-decidable, so is Π_{halt} . \square

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

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

such that

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

$$\left(a_{i \cdot (n+2)}, \dots, a_{i \cdot (n+2)+(n+1)} \right) \xrightarrow{\mathbb{P}} \left(a_{(i+1) \cdot (n+2)}, \dots, a_{(i+1) \cdot (n+2)+(n+1)} \right).$$

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 $\beta : \mathbb{N}^s \rightarrow \mathbb{N}$ with the following properties.*

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

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

- (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 $t, q, i, a \in \mathbb{N}$

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

+

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

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

Here,

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

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

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

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

$$j \text{ LET } R_1 = R_1 + 1,$$

then we let

$$\psi_j := u \equiv \bar{j} \rightarrow (u' \equiv u + 1 \wedge u'_0 \equiv u_0 \wedge u'_1 \equiv u_1 + 1 \wedge u'_2 \equiv u_2 \wedge \dots \wedge u'_n \equiv u_n).$$

The remaining details are left to the reader. \square

Using Lemma 1.1 we can prove similarly:

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

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

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

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

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

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

$$\exists x (\theta(x) \wedge \forall y (\varphi(y) \rightarrow y \equiv x)). \quad \dashv$$

2. Gödel's Incompleteness Theorems

Let $\Phi \subseteq L_0^{\text{Sar}}$.

Definition 2.1. Let $r \geq 1$.

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

$$\begin{aligned} (n_0, \dots, n_{r-1}) \in \mathcal{R} &\implies \Phi \vdash \varphi(\bar{n}_0, \dots, \bar{n}_{r-1}), \\ (n_0, \dots, n_{r-1}) \notin \mathcal{R} &\implies \Phi \vdash \neg \varphi(\bar{n}_0, \dots, \bar{n}_{r-1}). \end{aligned}$$

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

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

Moreover,

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

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 $\Phi \subseteq \Phi' \subseteq L_0^{\text{Sar}}$. 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. $\text{Th}(\mathfrak{N})$ 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^{Sar} -formulas. Let

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

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

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

Observe that both

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

are R-computable.

Next time we will show:

Theorem 2.6 (Fixed Point Theorem). *Assume that Φ allows representations. Then for every $\psi \in L_1^{\text{Sar}}$, 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. ⊢