Mathematical Logic (X)

Yijia Chen

1. Decidability and Enumerability

Definition 1.1. A program $\mathbb{P}$ *starts* with $w \in \mathcal{A}^*$ if in the beginning of the execution of $\mathbb{P}$ we have $\mathtt{R}_0 = w$ and all other $\mathtt{R}_i = \square$.

If $\mathbb{P}$ starts with w and eventually reaches the last halt instruction, then we write

$$\mathbb{P} : w \to \text{halt}.$$

Otherwise,

$$\mathbb{P} : w \to \infty.$$

The notation

$$\mathbb{P} : w \to w'$$

means that if $\mathbb{P}$ starts with w, then it eventually halts, and during the course of computation, has printed exactly one string w'. ⊣

Definition 1.2. Let $W \subseteq \mathcal{A}^*$.

(i) A program $\mathbb{P}$ *decides* W if for all $w \in \mathcal{A}^*$

$$\begin{array}{ll} \mathbb{P} : w \to \square & \text{if } w \in W, \\ \mathbb{P} : w \to w' \text{ with } w' \neq \square & \text{if } w \notin W. \end{array}$$

(ii) W is *register-decidable*, or *R-decidable* for short, if there is program which decides W. ⊣

1.1. The undecidability of first-order logic.

We will make use of the following theorem.

Theorem 1.3. *Let $\mathcal{A}$ be a fixed alphabet.*

(i) The set

$$\Pi'_{\text{halt}} := \{w_{\mathbb{P}} \mid \mathbb{P} \text{ a program over } \mathcal{A} \text{ and } \mathbb{P} : w_{\mathbb{P}} \to \text{halt}\}$$

is not R-decidable.

(ii) The set

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

is not R-decidable. ⊣

Proof: (i) Assume that there is a program $\mathbb{P}_0$ which decides Π'_{halt}. That is, for every program $\mathbb{P}$

$$\begin{array}{ll} \mathbb{P}_0 : w_{\mathbb{P}} \to \square & \text{if } \mathbb{P} : w_{\mathbb{P}} \to \text{halt}, \\ \mathbb{P}_0 : w_{\mathbb{P}} \to w' \text{ with } w' \neq \square & \text{if } \mathbb{P} : w_{\mathbb{P}} \to \infty. \end{array}$$

Assume furthermore that $\mathbb{P}_0$ has the form

0 .....

1 ……

⋮

10 **PRINT**

⋮

k **HALT**

We change $\mathbb{P}_0$ in such a way that if $\mathbb{P}_0$ prints out $\square$, then the modified program will never halt. To that end, we replace the last k-th halt instruction by two instruction that "reverse the halting behavior", and replace every print instruction by a "jump" instruction that directly goes to the end:

0 ……

1 ……

⋮

10 **IF** $\mathsf{R}_0 = \square$ **THEN** k **ELSE** k **OR** k **OR** $\cdots$ **OR** k

i.e, goto the k-th instruction no matter what is in R_0

⋮

k **IF** $\mathsf{R}_0 = \square$ **THEN** k **ELSE** $k+1$ **OR** $k+1$ **OR** $\cdots$ **OR** $k+1$

$k+1$ **HALT**

Let $\mathbb{P}_1$ be the resulting program. It is then easy to see that for any program $\mathbb{P}$

$$\begin{aligned} &\mathbb{P}_0 : w_{\mathbb{P}} \to \infty \quad && \text{if } \mathbb{P} : w_{\mathbb{P}} \to \text{halt}, \\ &\mathbb{P}_0 : w_{\mathbb{P}} \to \text{halt} && \text{if } \mathbb{P} : w_{\mathbb{P}} \to \infty. \end{aligned}$$

As a result,

$$\begin{aligned} &\mathbb{P}_0 : w_{\mathbb{P}_0} \to \infty \quad && \text{if } \mathbb{P}_0 : w_{\mathbb{P}_0} \to \text{halt}, \\ &\mathbb{P}_0 : w_{\mathbb{P}_0} \to \text{halt} && \text{if } \mathbb{P}_0 : w_{\mathbb{P}_0} \to \infty, \end{aligned}$$

which is certainly a contradiction.

(ii) Towards a contradiction, assume that $\mathbb{P}_0$ decides Π_{halt}. That is, for every program $\mathbb{P}$

$$\begin{aligned} &\mathbb{P}_0 : w_{\mathbb{P}} \to \square && \text{if } \mathbb{P} : \square \to \text{halt}, \\ &\mathbb{P}_0 : w_{\mathbb{P}} \to w' \text{ with } w' \neq \square && \text{if } \mathbb{P} : \square \to \infty. \end{aligned} \tag{1}$$

Now for every program $\mathbb{P}$ we assign in an *effective* way a program $\mathbb{P}^+$ such that

$$\mathbb{P} : w_{\mathbb{P}} \to \text{halt} \quad \Longleftrightarrow \quad \mathbb{P}^+ : \square \to \text{halt}. \tag{2}$$

Being effective means that there is a further program $\mathbb{T}$ that computes the mapping

$$w_{\mathbb{P}} \to w_{\mathbb{P}^+}.$$

The construction of $\mathbb{T}$ is tedious but not difficult.

With $\mathbb{P}_0$ and $\mathbb{T}$ we design a program which decide Π'_{halt} as follows. On any input $w \in \mathcal{A}^*$, the program first test whether $w = w_{\mathbb{P}}$ for some $\mathbb{P}$. If not, it rejects immediately[1]. Otherwise, it uses

[1] i.e., prints out some $w' \neq \square$ and halts.

$\mathbb{T}$ to computes $w_{\mathbb{P}^+}$. Then the program calls $\mathbb{P}_0$ on input $w_{\mathbb{P}^+}$. By (2) and (1), it correctly decides whether

$$\mathbb{P} : w_{\mathbb{P}} \to \text{halt}.$$

This gives us the desired contradiction to (i).

It remains to show the construction of $\mathbb{P}^+$ from any given $\mathbb{P}$ that fulfills (2). Assume that

$$\mathbb{P} = \underbrace{a_0 a_0 \dots a_0}_{n \text{ times}}.$$

Let $\mathbb{P}^+$ begin with

0 **LET** $R_0 = R_0 + a_0$

1 **LET** $R_0 = R_0 + a_0$

$\vdots$

n-1 **LET** $R_0 = R_0 + a_0$

and followed by the instructions of $\mathbb{P}$ with all labels increased by n. □

Theorem 1.4. *The set*

$$\left\{ \varphi \in L_0^{S_\infty} \;\middle|\; \models \varphi \right\} \tag{3}$$

is not R-decidable.

Proof: By Theorem 1.3 (ii) for the alphabet $\mathcal{A} = \{|\}$ the problem Π_{halt} is not R-decidable. Our goal is to show that the assumed R-decidability of (3) would contradict this result. To that end, for every program $\mathbb{P}$ we will construct in an *effective* way a $\varphi_{\mathbb{P}} \in L_0^{S_\infty}$ such that

$$\mathbb{P} : \square \to \text{halt} \quad \iff \quad \models \varphi_{\mathbb{P}}.$$

Here, the effectivity means that there is a program $\mathbb{P}_1$ which computes the mapping

$$\mathbb{P} \mapsto \varphi_{\mathbb{P}}.$$

Once this is done, given an input $w \in \mathcal{A}^*$, we can first check whether $w = w_{\mathbb{P}}$, if so, extract the program $\mathbb{P}$ and compute $\varphi_{\mathbb{P}}$ using $\mathbb{P}_1$. Thus if (3) is decidable, we can apply the corresponding decision program on input $\varphi_{\mathbb{P}}$ to decide whether $\mathbb{P} : \square \to \text{halt}$. Hence, we could decide Π_{halt}.

Let $\mathbb{P}$ consist of instructions $\alpha_0, \dots, \alpha_k$, in particular every α_i has its label i. Furthermore, assume that the maximum index of the registers which $\mathbb{P}$ uses is n. Hence, the registers referred by all α_i's are among $R_0, \dots, R_n$.

Key to our construction of $\varphi_{\mathbb{P}}$ is the notion of configurations of $\mathbb{P}$. An $(n+2)$-tuple

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

is a *configuration of* $\mathbb{P}$ *(on input* $\square$*) after* s *steps* if

- starting with input $\square$ the program $\mathbb{P}$ runs at least s steps,
- after s steps, the instruction α_L is to be executed next,
- and for every $0 \leqslant i \leqslant n$ the register R_i contains the word

$$\underbrace{| \, | \cdots |}_{m_i \text{ times}}$$

at that moment. To ease presentation, in the following we will simply say that R_i contains the number m_i.

Observe that then the execution of $\mathbb{P}$ on the $s+1$-th step is completely determined by the configuration $(L, m_0, \ldots, m_n)$.

The *initial configuration*, i.e., the configuration of $\mathbb{P}$ after 0 step is

$$(0, 0, \ldots, 0).$$

Recall that α_k is the last instruction of $\mathbb{P}$, i.e., the only halt instruction. Therefore

$$\begin{aligned}\mathbb{P} : \square \to \text{halt} \iff &\text{for some } s, m_0, \ldots, m_n \in \mathbb{N} \\ &\text{the tuple } (k, m_0, \ldots, m_n) \text{ is the configuration of } \mathbb{P} \text{ after } s \text{ steps.}\end{aligned} \tag{4}$$

In case $\mathbb{P} : \square \to \text{halt}$, we define $s_{\mathbb{P}} \in \mathbb{N}$ to be the number of steps which $\mathbb{P}$ carries out until it reaches the last halt instruction.

We choose four symbols from S^∞: $R := R_0^{n+3}$, $<:= R_0^2$, $f := f_0^1$, and $c := c_0$, and set

$$S := \{R, <, f, c\}.$$

Then we associate with $\mathbb{P}$ an S-structure $\mathfrak{A}_{\mathbb{P}}$ which "describes" the execution (i.e., the behaviour) of $\mathbb{P}$ on input $\square$. There are two cases.

Case 1. $\mathbb{P} : \square \to \infty$. We set $A_{\mathbb{P}} := \mathbb{N}$, $<^{\mathfrak{A}_{\mathbb{P}}} := \big\{(i,j) \;\big|\; i, j \in \mathbb{N} \text{ and } i < j\big\}$, $f^{\mathfrak{A}_{\mathbb{P}}}(i) := i+1$ for every $i \in \mathbb{N}$, $c^{\mathfrak{A}_{\mathbb{P}}} := 0$, and

$$R^{\mathfrak{A}_{\mathbb{P}}} := \big\{(s, L, m_0, \ldots, m_n) \;\big|\; (L, m_0, \ldots, m_n) \text{ is the configuration of } \mathbb{P} \text{ after } s \text{ steps}\big\}.$$

Case 2. $\mathbb{P} : \square \to \text{halt}$. Let $e := \max\{k, s_{\mathbb{P}}\}$. Then we set $A_{\mathbb{P}} := \{0, \ldots, e\}$, $<^{\mathfrak{A}_{\mathbb{P}}} := \big\{(i,j) \;\big|\; 0 \leqslant i < j \leqslant e\big\}$, $f^{\mathfrak{A}_{\mathbb{P}}}(i) := \min\{i+1, e\}$ for every $i \in A_{\mathbb{P}}$, $c^{\mathfrak{A}_{\mathbb{P}}} := 0$, and

$$R^{\mathfrak{A}_{\mathbb{P}}} := \big\{(s, L, m_0, \ldots, m_n) \;\big|\; (L, m_0, \ldots, m_n) \text{ is the configuration of } \mathbb{P} \text{ after } s \text{ steps}\big\}.$$

Note that, since every register R_i starts with 0, and can increase its value (i.e, the length of $|\,|\cdots|$) by at most 1 in each step, thus $m_i \leqslant s_{\mathbb{P}} \leqslant e$. So $R^{\mathfrak{A}_{\mathbb{P}}}$ is well defined.

Towards the definition of $\varphi_{\mathbb{P}}$ in (3), we first construct a sentence $\psi_{\mathbb{P}}$ which expresses the execution of $\mathbb{P}$ on $\square$. We abbreviate c, fc, ffc, ... by $\bar{0}$, $\bar{1}$, $\bar{2}$, ..., respectively. The desired $\psi_{\mathbb{P}}$ should satisfy the following two properties:

(P1) $\mathfrak{A}_{\mathbb{P}} \models \psi_{\mathbb{P}}$.

(P2) Let $\mathfrak{A}$ be an S-structure with $\mathfrak{A} \models \psi_{\mathbb{P}}$. Furthermore, $(L, m_0, \ldots, m_n)$ is the configuration of $\mathbb{P}$ after s steps. Then
$$\mathfrak{A} \models R\bar{s}\bar{L}\bar{m}_0 \cdots \bar{m}_n.$$

We set
$$\psi_{\mathbb{P}} := \psi_0 \wedge R\bar{0}\bar{0}\cdots\bar{0} \wedge \psi_{\alpha_0} \wedge \cdots \wedge \psi_{\alpha_{k-1}},$$
where each conjunct is defined as follows. The first

$$\begin{aligned}\psi_0 := \text{"}< \text{ is an ordering"} \wedge \forall x(c < x \vee x \equiv c) \wedge \forall x(x < fx \vee x \equiv fx) \\ \wedge \forall x\big(\exists y\; x < y \to (x < fx \wedge \forall z(x < z \to (fx < z \vee fx \equiv z)))\big),\end{aligned}$$

i.e., $<$ is an ordering, c is the minimum element, fx is the successor of x except that $x = fx$ for the maximum x.

For $\alpha \in \{\alpha_0, \ldots, \alpha_{k-1}\}$ we define φ_α by a case analysis.

– $\alpha = L$ **LET** $R_i = R_i+|$. Then let

$$\psi_\alpha := \forall x \forall y_0 \cdots \forall y_n \big(Rx\bar{L}y_0 \cdots y_n \to (x < fx \wedge Rfx\overline{L+1}y_0 \cdots y_{i-1} f y_i y_{i+1} \cdots y_n)\big).$$

– $\alpha = L$ **LET** $R_i = R_i - |$. Then let

$$\psi_\alpha := \forall x \forall y_0 \cdots \forall y_n \big(Rx\bar{L}y_0 \cdots y_n \to (x < fx \wedge ((y_i \equiv \bar{0} \wedge Rfx\overline{L+1}y_0 \cdots y_n) \vee (\neg y_i \equiv \bar{0} \wedge \exists u (fu \equiv y_i \wedge Rfx\overline{L+1}y_0 \cdots y_{i-1} u y_{i+1} \cdots y_n))))\big).$$

– $\alpha = L$ **IF** $R_i = \square$ **THEN** L' **ELSE** L_0. Then let

$$\psi_\alpha := \forall x \forall y_0 \cdots \forall y_n \big(Rx\bar{L}y_0 \cdots y_n \to (x < fx \wedge ((y_i \equiv \bar{0} \wedge Rfx\overline{L'}y_0 \cdots y_n) \vee (\neg y_i \equiv \bar{0} \wedge Rfx\overline{L_0}y_0 \cdots y_n)))\big).$$

– $\alpha = L$ **PRINT**. Then let

$$\psi_\alpha := \forall x \forall y_0 \cdots \forall y_n \big(Rx\bar{L}y_0 \cdots y_n \to (x < fx \wedge Rfx\overline{L+1}y_0 \cdots y_n)\big).$$

The verification of (P1) and (P2) are left as an exercise.

Finally let

$$\varphi_{\mathbb{P}} := \psi_{\mathbb{P}} \to \exists x \exists y_0 \cdots \exists y_n Rx\bar{k}y_0 \cdots y_n.$$

Now we verify that $\mathbb{P} : \square \to$ halt if and only if $\models \varphi_{\mathbb{P}}$. First, assume $\models \varphi_{\mathbb{P}}$, in particular

$$\mathfrak{A}_{\mathbb{P}} \models \varphi_{\mathbb{P}}.$$

By (P1) we conclude

$$\mathfrak{A}_{\mathbb{P}} \models \exists x \exists y_0 \cdots \exists y_n Rx\bar{k}y_0 \cdots y_n.$$

Then there are some $s, m_0, \ldots, m_n \in A_{\mathbb{P}} \subseteq \mathbb{N}$ such that $(k, m_0, \ldots, m_n)$ is the configuration of $\mathbb{P}$ after s steps. Therefore, $\mathbb{P}$ reaches the last halt instruction after s steps, hence $\mathbb{P} : \square \to$ halt.

Conversely, assume $\mathbb{P} : \square \to$ halt. Let $\mathfrak{A}$ be an S-structure. We need to show that $\mathfrak{A} \models \varphi_{\mathbb{P}}$. Clearly, if $\mathfrak{A} \not\models \psi_{\mathbb{P}}$, then we are already done. Thus, assume $\mathfrak{A} \models \psi_{\mathbb{P}}$. Recall that $s_{\mathbb{P}} \in \mathbb{N}$ is the number of steps which $\mathbb{P}$ carries out until it reaches the last halt instruction α_k. Hence, for some $m_0, \ldots, m_n \leqslant s_{\mathbb{P}}$ the tuple

$$(k, m_0, \ldots, m_n)$$

is the configuration of $\mathbb{P}$ after $s_{\mathbb{P}}$ steps. Now (P2) implies that

$$\mathfrak{A} \models R\overline{s_{\mathbb{P}}}\bar{k}\bar{m}_0 \cdots \bar{m}_n.$$

Therefore

$$\mathfrak{A} \models \varphi_{\mathbb{P}}.$$

This finishes the proof. □

2. Exercises

Exercise 2.1. Prove (P1) and (P2) in the proof of Theorem 1.4. ⊣

Exercise 2.2. Assume $\mathbb{P} : \square \to$ halt. Construct an *infinite* S-structure with $\mathfrak{A} \models \psi_{\mathbb{P}}$.

Exercise 2.3. Show that

$$\{\varphi \in L_0^{S_\infty} \mid \varphi \text{ is satisfiable}\}$$

is not R-enumerable. ⊣