Mathematical Logic (X)

Yijia Chen

1. Decidability and Enumerability

1.1. Register Machines. We fix an alphabet

$$\mathcal{A} := \{a_0, \dots, a_r\}.$$

Every *register machine* (or simply, machine) has a fixed number of registers, i.e.,

$$R_0, \dots, R_m$$

for some fixed $m \in \mathbb{N}$, where any register R_i can contain any word in $\mathcal{A}^*$. A *program* consists of a finite number of *instructions*, each starting with a *label* $L \in \mathbb{N}$.

There are 5 types of instructions.

–
$$L \textbf{ LET } R_i = R_i + a_j,$$

where $L, i, j \in \mathbb{N}$ with $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant r$. That is, add the letter a_j at the end of the word in R_i.

–
$$L \textbf{ LET } R_i = R_i - a_j,$$

where $L, i, j \in \mathbb{N}$ with $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant r$. That is, if the word in R_i ends with e_j, then delete this a_j; otherwise leave the word unchanged.

–
$$L \textbf{ IF } R_i = \square \textbf{ THEN } L' \textbf{ ELSE } L_0 \textbf{ OR } L_1 \textbf{OR } \cdots \textbf{OR } L_r,$$

where $L, L', L_0, \dots, L_r \in \mathbb{N}$. That is, if R_i contains $\square$, then go the instruction labelled L'. Otherwise, if R_i contains a word ending with the letter a_j, then go to the instruction labelled L_j.

–
$$L \textbf{ PRINT},$$

where $L \in \mathbb{N}$. That is, output the word in R_0.

–
$$L \textbf{ HALT},$$

with $L \in \mathbb{N}$. That is, the program halts.

Definition 1.1. A *register program* (or simply *program*) is a finite sequence α_0, α_k of instructions with the following properties.

(i) Every α_i has label $L = i$.

(ii) Every jump operation refers to a label $\leqslant k$.

(iii) Only the last instruction α_k is a halt instruction. ⊣

Definition 1.2. A program $\mathbb{P}$ *starts* with $w \in \mathcal{A}^*$ if in the beginning of the execution of $\mathbb{P}$ we have $R_0 = w$ and all other $R_i = \Box$.

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 with $R_0 = w'$. ⊣

Definition 1.3. Let $W \subseteq \mathcal{A}^*$.

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

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

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

Definition 1.4. Let $W \subseteq \mathcal{A}^*$.

(i) A program $\mathbb{P}$ *enumerates* W if started with $\Box$, $\mathbb{P}$ prints out exactly the words in W (in some order with possible repetitions).

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

Proposition 1.5. *Let $W \subseteq \mathcal{A}^*$. Then W is R-decidable if and only if both W and $\mathcal{A}^* \setminus W$ are R-enumerable.*

Definition 1.6. Let $F \subseteq \mathcal{A}^* \to \mathcal{B}^*$, where $\mathcal{A}$ and $\mathcal{B}$ are two alphabets.

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

$$\mathbb{P} : w \to F(w).$$

(ii) F is *register-computable*, or *R-computable* for short, if there is program which computes F. ⊣

1.2. The halting problem for the register machines. Again let $\mathcal{A} := \{a_0, \dots, a_r\}$ be a fixed alphabet. Our goal is to define for every program $\mathbb{P}$ over $\mathcal{A}$ a word $w_{\mathbb{P}} \in \mathcal{A}^*$. To that end, we first introduce an auxiliary alphabet

$$\mathcal{B} := \mathcal{A} \cup \{A, B, C, \dots, Z\} \cup \{0, 1, \dots, 9\} \cup \{=, +, -, \Box, |\}.$$

As usual, we understand that the words in $\mathcal{B}^*$ are ordered *lexicographically*. Then every program can be naturally encoded as a word in $\mathcal{B}^*$. For instance

0 **LET** $R_1 = R_1 - a_0$

2 **PRINT**

3 **HALT**

is identified with the word

$$0\mathsf{LETR}1 = \mathsf{R}1 - a_0 \mid |1\mathsf{PRINT} \mid 3\mathsf{HALT}.$$

Note that a_0 is single letter from the alphabet $\mathcal{A} \subseteq \mathcal{B}$. Assume that this word is the n-th word in the lexicographical ordering of $\mathcal{B}^*$. Then we set

$$w_{\mathbb{P}} := \underbrace{11\cdots1}_{n \text{ times}}.$$

Finally let

$$\Pi := \{w_{\mathbb{P}} \mid \mathbb{P} \text{ a program over } \mathcal{A}\}.$$

The mapping

$$\mathbb{P} \mapsto w_{\mathbb{P}}$$

is often called the *Gödel numbering*, and $w_{\mathbb{P}}$ is the *Gödel number* of $\mathbb{P}$.

Lemma 1.7. Π *is* R*-decidable.* ⊣

Theorem 1.8. *(i) The set*

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

is not R-decidable.

(ii) The set

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

is not R-decidable. ⊣

2. Exercises

Exercise 2.1. Let $W \subseteq \mathcal{A}^*$. A program $\mathbb{P}$ *strictly enumerates* W if started with $\square$, $\mathbb{P}$ prints out all the words in W

$$w_0, w_1, \ldots$$

without repetitions such that $|w_i| \leqslant |w_{i+1}|$ for all $i \in \mathbb{N}$. Recall $|w|$ denotes the length of the word w.

W is strictly R-enumerable if there is a program which strictly enumerates W. Are the following statements correct?

- W is R-enumerable if and only W is strictly R-enumerable.
- W is R-decidable if and only W is strictly R-enumerable. ⊣

Exercise 2.2. Prove that the set

$$\{w_{\mathbb{P}} \mid \mathbb{P} \text{ a program over } \mathcal{A} \text{ and } \mathbb{P} : w \to \text{halt} \textit{ for some } w \in \mathcal{A}^*\}$$

is not R-decidable. ⊣