Mathematical Logic (X)

Yijia Chen

1. The Löwenheim-Skolem Theorem and the Compactness Theorem

Using the term-interpretation, it is routine to verify:

Theorem 1.1 (Löwenheim-Skolem). *Let $\Phi \subseteq L^S$ be at most countable and satisfiable. Then there is an S-interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ such that*

– *the universe A of $\mathfrak{A}$ is at most countable,*

– *and $\mathfrak{I} \models \Phi$.* $\dashv$

The following is a more general version.

Theorem 1.2 (Downward Löwenheim-Skolem). *Let $\Phi \subseteq L^S$ be satisfiable. Then there is an S-interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ such that*

– $|A| \leqslant |T^S| = |L^S|$,

– *and $\mathfrak{I} \models \Phi$.* $\dashv$

Corollary 1.3. *Let $S := \{+, \times, <, 0, 1\}$ with the usual meaning and*

$$\Phi_{\mathbb{R}} := \left\{\varphi \in L_0^S \;\middle|\; (\mathbb{R}, +, \cdot, <, 0, 1) \models \varphi\right\}.$$

Then there is a countable *S-structure $\mathfrak{A}$ with $\mathfrak{A} \models \Phi_{\mathbb{R}}$.* $\dashv$

By the Completeness Theorem:

Theorem 1.4 (Compactness). *(a) $\Phi \models \varphi$ if and only if there is a* finite *$\Phi_0 \subseteq \Phi$ with $\Phi_0 \models \varphi$.*

(b) Φ is satisfiable if and only if every finite $\Phi_0 \subseteq \Phi$ is satisfiable. $\dashv$

In fact, the "compactness" is a notion from topology. We can explain the topological perspective of Theorem 1.4 using *finite covers* from analysis. For every $\varphi \in L^S$ we define

$$\mathrm{Mod}(\varphi) := \left\{\mathfrak{I} \;\middle|\; \mathfrak{I} \models \varphi\right\},$$

and

$$\mathrm{Mod}(\Phi) := \left\{\mathfrak{I} \;\middle|\; \mathfrak{I} \models \Phi\right\} = \bigcap_{\psi \in \Phi} \mathrm{Mod}(\psi).$$

We show that Theorem 1.4 is equivalent to the following *finite cover property*.

Proposition 1.5. $\mathrm{Mod}(\varphi) \subseteq \bigcup_{\psi \in \Phi} \mathrm{Mod}(\psi)$ *if and only if for some finite $\Phi_0 \subseteq \Phi$ we have*

$$\mathrm{Mod}(\varphi) \subseteq \bigcup_{\psi \in \Phi_0} \mathrm{Mod}(\psi).$$ $\dashv$

Proof of Theorem 1.4 using Proposition 1.5:

$$
\begin{aligned}
\Phi \models \varphi &\iff \mathrm{Mod}(\Phi) \subseteq \mathrm{Mod}(\varphi) \\
&\iff \overline{\mathrm{Mod}(\varphi)} \subseteq \overline{\mathrm{Mod}(\Phi)} \\
&\iff \overline{\mathrm{Mod}(\varphi)} \subseteq \overline{\bigcap_{\psi \in \Phi} \mathrm{Mod}(\psi)} \\
&\iff \overline{\mathrm{Mod}(\varphi)} \subseteq \bigcup_{\psi \in \Phi} \overline{\mathrm{Mod}(\psi)} \\
&\iff \mathrm{Mod}(\neg\varphi) \subseteq \bigcup_{\psi \in \Phi} \mathrm{Mod}(\neg\psi) \\
&\iff \mathrm{Mod}(\neg\varphi) \subseteq \bigcup_{\psi \in \Phi_0} \mathrm{Mod}(\neg\psi) \text{ for some finite } \Phi_0 \subseteq \Phi \qquad \text{(by Proposition 1.5)} \\
&\iff \overline{\mathrm{Mod}(\varphi)} \subseteq \bigcup_{\psi \in \Phi_0} \overline{\mathrm{Mod}(\psi)} \text{ for some finite } \Phi_0 \subseteq \Phi \\
&\iff \overline{\mathrm{Mod}(\varphi)} \subseteq \overline{\bigcap_{\psi \in \Phi_0} \mathrm{Mod}(\psi)} \text{ for some finite } \Phi_0 \subseteq \Phi \\
&\iff \bigcap_{\psi \in \Phi_0} \mathrm{Mod}(\psi) \subseteq \mathrm{Mod}(\varphi) \text{ for some finite } \Phi_0 \subseteq \Phi \\
&\iff \mathrm{Mod}(\Phi_0) \subseteq \mathrm{Mod}(\varphi) \text{ for some finite } \Phi_0 \subseteq \Phi \\
&\iff \Phi_0 \models \varphi \text{ for some finite } \Phi_0 \subseteq \Phi.
\end{aligned}
$$

□

Proof of Proposition 1.5 by Theorem 1.4: The direction from right to left is trivial. So we assume that

$$\mathrm{Mod}(\varphi) \subseteq \bigcup_{\psi \in \Phi} \mathrm{Mod}(\psi).$$

Claim. $\{\neg\psi \mid \psi \in \Phi\} \models \neg\varphi$.

Proof of the claim. Let $\mathfrak{I}$ be an interpretation with

$$\mathfrak{I} \models \{\neg\psi \mid \psi \in \Phi\}.$$

That is, $\mathfrak{I} \models \neg\psi$ for every $\psi \in \Phi$. We can deduce that

$$
\begin{aligned}
\mathfrak{I} \in \bigcap_{\psi \in \Phi} \mathrm{Mod}(\neg\psi) &\iff \mathfrak{I} \in \bigcap_{\psi \in \Phi} \overline{\mathrm{Mod}(\psi)} \\
&\iff \mathfrak{I} \in \overline{\bigcup_{\psi \in \Phi} \mathrm{Mod}(\psi)} \\
&\iff \mathfrak{I} \notin \bigcup_{\psi \in \Phi} \mathrm{Mod}(\psi) \\
&\implies \mathfrak{I} \notin \mathrm{Mod}(\varphi) \qquad \left(\text{by } \mathrm{Mod}(\varphi) \subseteq \bigcup_{\psi \in \Phi} \mathrm{Mod}(\psi)\right) \\
&\iff \mathfrak{I} \models \neg\varphi.
\end{aligned}
$$

This finishes the proof of the claim. ⊣

Now we apply Theorem 1.4 to the above claim. In particular, there is a finite $\Phi_0 \subseteq \Phi$ such that

$$\{\neg\psi \mid \psi \in \Phi_0\} \models \neg\varphi$$

Then arguing similarly as above, we obtain

$$\operatorname{Mod}(\varphi) \subseteq \bigcup_{\psi \in \Phi_0} \operatorname{Mod}(\psi). \qquad \square$$

Theorem 1.6. *Let $\Phi \subseteq L^S$ such that for every $n \in \mathbb{N}$ there exists an S-interpretation $\mathfrak{I}_n = (\mathfrak{A}_n, \beta_n)$ with $|A_n| \geqslant n$ and $\mathfrak{I}_n \models \Phi$. Then there is an S-interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ with* infinite *A and $\mathfrak{I} \models \Phi$.*

Proof: For every $n \geqslant 2$ we define a sentence

$$\varphi_{\geqslant n} := \exists v_0 \cdots \exists v_{n-1} \bigwedge_{0 \leqslant i < j \leqslant n} \neg v_i \equiv v_j.$$

Clearly for any structure $\mathfrak{A}$ (regardless of the symbol set S)

$$\mathfrak{A} \models \varphi_{\geqslant n} \quad \Longleftrightarrow \quad |A| \geqslant n.$$

Now consider

$$\Psi := \Phi \cup \{\varphi_{\geqslant n} \mid n \geqslant 2\}.$$

Of course every finite subset of Ψ is contained in

$$\Psi_{n_0} := \Phi \cup \{\varphi_{\geqslant n} \mid 2 \leqslant n \leqslant n_0\}$$

for a *sufficiently large* $n_0 \in \mathbb{N}$. By assumption, the interpretation $\mathfrak{I}_{n_0}$ witnesses that Ψ_{n_0} is satisfiable. Therefore, by the Compactness Theorem, Ψ itself is satisfiable. The result follows immediately. $\square$

Theorem 1.7 (Upward Löwenheim-Skolem)**.** *Let $\Phi \subseteq L^S$ and assume that there is an S-interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ such that A is infinite and $\mathfrak{I} \models \Phi$. Then, for any set B there is an S-interpretation $\mathfrak{I}' = (\mathfrak{A}', \beta')$ with $|A'| \geqslant |B|$ and $\mathfrak{I}' \models \Phi$.*

Proof: For any $b \in B$ we introduce a new constant $c_b \notin S$. In particular, $c_b \neq c_{b'}$ for any $b, b' \in B$ with $b \neq b$. Then consider

$$\Psi := \Phi \cup \{\neg c_b \equiv c_{b'} \mid b, b' \in B \text{ with } b \neq b'\}.$$

Since Φ has an infinite interpretation, every finite subset of Ψ is satisfiable. By the Compactness Theorem, we conclude that Φ is satisfiable. Clearly the structure in any interpretation which satisfies Ψ must have size as large as $|B|$. $\square$

Corollary 1.8. *Let $S = \{+, \times, <, 0, 1\}$ and*

$$\Phi_{\mathbb{N}} := \{\varphi \in L_0^S \mid (\mathbb{N}, +, \cdot, <, 0, 1) \models \varphi\}.$$

Then there is a uncountable *S-structure $\mathfrak{A}$ with $\mathfrak{A} \models \Phi_{\mathbb{N}}$.* $\dashv$

2. Decidability and Enumerability

A. Procedure and Decidability.

Definition 2.1. Let $\mathcal{A}$ be an alphabet (which we always assume to be finite) and $W \subseteq \mathcal{A}^*$.

(i) Let $\mathbb{P}$ be a procedure/program (which we will make precise shortly afterwards). $\mathbb{P}$ is a *decision procedure for W* if on every input $w \in \mathcal{A}^*$ the procedure $\mathbb{P}$ will eventually halt and output some $w' \in \mathcal{A}^*$ such that

- if $w \in W$, then $w' = \Box$, where $\Box$ is the empty string,
- if $w \notin W$, then $w' \neq \Box$.

(ii) W is *decidable* if there is a decision procedure for W. ⊣

B. Enumerability.

Definition 2.2. Let $\mathcal{A}$ be an alphabet and $W \subseteq \mathcal{A}^*$.

(i) A procedure $\mathbb{P}$ is an *enumeration procedure for* W if $\mathbb{P}$ (without any input) outputs all the words in W (in some order and possibly with repetitions).

(ii) W is *enumerable* if there is an enumeration procedure for W. ⊣

Lemma 2.3. *If there is an enumeration procedure for W, then there is an enumeration procedure for W* without repetitions. ⊣

Lemma 2.4. *Let $\mathcal{A}$ be finite. Then $\mathcal{A}^*$ is enumerable.* ⊣

Let

$$\begin{aligned} S_\infty := & \left\{c_0, c_1, \dots\right\} && \text{(every } c_i \text{ is a constant)} \\ & \cup \bigcup_{n \geqslant 1} \left\{R^n_0, R^n_1, \dots\right\} && \text{(every } R^n_i \text{ is an } n\text{-ary relation symbol)} \\ & \cup \bigcup_{n \geqslant 1} \left\{f^n_0, f^n_1, \dots\right\} && \text{(every } f^n_i \text{ is an } n\text{-ary function symbol).} \end{aligned}$$

Lemma 2.5.

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

is enumerable.

Proof: [sketch] By the Completeness Theorem

$$\left\{\varphi \in L_0^{S_\infty} \;\middle|\; \models \varphi\right\} = \left\{\varphi \in L_0^{S_\infty} \;\middle|\; \vdash \varphi\right\}.$$

Thus, we can enumerate all possible proofs/derivations of symbol set S^∞, thus obtain all those $\varphi \in L_0^{S_\infty}$ with $\vdash \varphi$. □

C. The Relationship between Decidability and Enumerability.

Theorem 2.6. *Every decidable set is enumerable.*

Proof: Assume that the procedure $\mathbb{P}$ decides $W \subseteq \mathcal{A}^*$. By Lemma 2.4 we can enumerate all $w \in \mathcal{A}^*$. For each w we can decide whether $w \in W$ by calling $\mathbb{P}$. If so, we output w and proceed to the next string. Otherwise, we move to the next string without outputting w. □

We will see later that the converse of Theorem 2.6 does not hold, i.e., there are enumerable sets which are not decidable. Nevertheless, we can show:

Theorem 2.7. *Let $W \subseteq \mathcal{A}^*$. Then W is decidable if and only if both W and $\mathcal{A}^* \setminus W$ are enumerable.*

Proof: The direction from left to right is straightforward by Theorem 2.6 and by observing that $\mathcal{A}^* \setminus W$ is decidable as well. For the converse, we have two procedures, $\mathbb{P}_1$ which enumerates W, and $\mathbb{P}_2$ which enumerates $\mathcal{A}^* \setminus W$.

Then given an input $w \in \mathcal{A}^*$, we simulate two procedures $\mathbb{P}_1$ and $\mathbb{P}_2$ simultaneously[1], eventually w will appear in exactly one of the outputs of $\mathbb{P}_1$ and $\mathbb{P}_2$. Then we can answer whether $w \in W$ accordingly. □

D. Computable Functions.

Definition 2.8. Let $\mathcal{A}$ and $\mathcal{B}$ be two alphabets. A procedure that for each input $w \in \mathcal{A}^*$ outputs a $w' \in \mathcal{B}^*$ determines a function $f : \mathcal{A}^* \to \mathcal{B}^*$ defined by

$$w \overset{f}{\mapsto} w'.$$

f is said to be *computable*. ⊣

2.1. 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 a_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.

[1]More precisely, we simulate the steps of $\mathbb{P}_1$ and $\mathbb{P}_2$ alternatively, i.e., the first step of $\mathbb{P}_1$, the first step of $\mathbb{P}_2$, the second step of $\mathbb{P}_1$, the second step of $\mathbb{P}_2$, ...

Definition 2.9. A *register program* (or simply *program*) is a finite sequence $\alpha_0, \dots, \alpha_k$ 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. $\dashv$

Definition 2.10. 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 $R_i = \square$.

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'. $\dashv$