Gödel’s Incompleteness Theorem

CSE 205A Lecture Notes

Suppose we have the vocabulary $V = \{\sigma, +, \times, \uparrow, =, <\}$ for $\mathbb{N}$, the natural numbers. We will show

- $\{\sigma \mid \mathbb{N} \models \sigma\}$, that is, the set of statements true about the natural numbers, is not recursive, and not even recursively enumerable (r.e.);

- there is no r.e. set of sentences $\Sigma$ over vocabulary $V$ such that $\Sigma \vdash \sigma \iff \mathbb{N} \models \sigma$, that is, no r.e., sound and complete set of axioms for arithmetic; and

- The set VALID of valid sentences in first-order logic (FO) is not recursive.

Note: The results hold even without $\uparrow$. For example, see Sipser for an alternative proof of the first two facts, using just the vocabulary $\{\sigma, +, \times, =, <\}$.

In the proof presented in class (which is from Papadimitriou), we use the following.

**Tool: Recursive inseparability of languages**

Note: In the following, all Turing Machines considered are deterministic deciders: they halt on every input and their output is ”yes” (accept the input) or ”no” (reject the input).

Recall that two languages $L_1$ and $L_2$ are recursively inseparable iff there is no recursive language $R$ such that $L_1 \subseteq \overline{R}$ and $L_2 \subseteq R$. The following is well known:

**Theorem 1.** Let $L_1 = \{M \mid M(M) \text{ accepts} \}$ and $L_2 = \{M \mid M(M) \text{ rejects} \}$, where $M$ denotes a Turing machine (TM) (and, by abuse of notation, its string encoding). Then $L_1$ and $L_2$ are recursively inseparable.
Proof. Suppose \( L_1 \) and \( L_2 \) are separated by some recursive language \( R : L_1 \subseteq \overline{R}, L_2 \subseteq R \).

Consider \( M_R \), the TM deciding \( R \).

Assume \( M_R(M_R) \) accepts. Then \( M_R \in L_1 \subseteq \overline{R} \), so \( M_R \notin R \), and therefore \( M_R(M_R) \) must reject, which contradicts our assumption.

Now assume \( M_R(M_R) \) rejects. Then, likewise, \( M_R \in L_2 \subseteq R \), so \( M_R \in R \), and therefore \( M_R(M_R) \) must accept, again contradicting the assumption. \( \square \)

**Corollary 1.** Let \( L'_1 = \{ M' | M'(\epsilon) \text{ accepts} \} \) and \( L'_2 = \{ M' | M'(\epsilon) \text{ rejects} \} \). \( L'_1 \) and \( L'_2 \) are recursively inseparable.

*Proof.* Suppose \( L'_1 \) and \( L'_2 \) are separated by some recursive language \( R' : L'_1 \subseteq \overline{R'}, L'_2 \subseteq R' \).

If this were so, we could use \( R' \) to separate \( L_1 \) from \( L_2 \) as follows: given a machine \( M \in L_1 \cup L_2 \), construct a machine \( M' \) that, on input \( \epsilon \), runs \( M(M) \). Then if \( M' \in R' \), then \( M \in L_2 \); otherwise \( M \in L_1 \). \( \square \)

**Turing machines and \( \mathbb{N} \)**

What is the connection between Turing machines and integer arithmetic? The following are not hard to see:

- Each TM configuration can be represented as an integer.
- Each TM finite computation can be represented as an integer.
- The existence of an accepting computation of \( M \) on input \( \epsilon \) (or on any input for that matter) can be stated as an FO sentence over the above vocabulary.

We are now ready to outline the main stages in the proof of the incompleteness theorem.

**Plan:** We will come up with a set of sound axioms for \( \mathbb{N} \), denoted \( \mathcal{NT} \), such that \( \{ \sigma | \mathcal{NT} \vdash \sigma \} \) and \( \text{UNSAT} \) are recursively inseparable. Suppose we have such an \( \mathcal{NT} \). Figure 1 shows the connection of \( \mathcal{NT} \) with some relevant classes of sentences.

The existence of \( \mathcal{NT} \) as above immediately proves our claims. Indeed, suppose such \( \mathcal{NT} \) exists. Then the following would hold:
Figure 1: Connection between **NT** and some classes of sentences.

VALID implies provable from NT implies true in N
UNSAT implies negation provable from NT implies false in N
• \{\sigma \mid N \models \sigma \} is not recursive (otherwise, in view of the figure, this language would recursively separate \{\sigma \mid NT \vdash \sigma \} from UNSAT).

• VALID is not recursive.

Proof. Suppose it is. Then we can separate \{\sigma \mid NT \vdash \sigma \} from UNSAT as follows: On input \sigma, accept if \(NT \rightarrow \sigma \in VALID\), and reject otherwise. \(\square\)

• There is no r.e. set \(\Sigma\) of sentences over vocabulary \(V\) such that for every sentence \(\sigma\),

\[ \Sigma \vdash \sigma \iff N \models \sigma \]

Proof. Suppose there is such \(\Sigma\). Then \(\{\sigma \mid N \models \sigma \}\) is r.e., so \(\{\neg \sigma \mid N \models \sigma \}\) is also r.e., and it follows that \(\{\sigma \mid N \models \sigma \}\) is recursive, which we showed above is not the case. \(\square\)

To show that \(\{\sigma \mid NT \vdash \sigma \}\) and UNSAT are recursively inseparable, we use the fact that the languages \(L'_1 = \{M \mid M(\epsilon) \text{ accepts}\}\) and \(L'_2 = \{M \mid M(\epsilon) \text{ rejects}\}\) are recursively inseparable (by Corollary 1). For this reduction, we need NT to be powerful enough to prove certain facts about TM computations.

First recall, as noted earlier, there exist FO sentences defining the following:

\(\varphi_M(x) : "x\text{ is an accepting computation of } M\text{ on input } \epsilon"\)

\(\varphi'_M(x) : "x\text{ is a rejecting computation of } M\text{ on input } \epsilon"\)

Suppose the following hold:

(i) \(N \models \exists x \varphi_M(x) \iff NT \vdash \exists x \varphi_M(x)\)

(ii) \(N \models \exists x \varphi'_M(x) \iff NT \vdash \exists x \varphi'_M(x)\),

(iii) \(NT \land \exists x \varphi'_M(x) \land \exists x \varphi_M(x)\) is inconsistent.

We can reduce the recursive separation of \(L'_1\) and \(L'_2\) to the recursive separation of \(\{\sigma \mid NT \vdash \sigma\}\) and UNSAT. Indeed, by (i) \(M\) accepts \(\epsilon\) iff \(NT \vdash \exists x \varphi_M(x)\) iff \(NT \vdash NT \land \exists x \varphi_M(x)\). By (ii), \(M\) rejects \(\epsilon\) iff \(NT \vdash \exists x \varphi'_M(x)\). By (iii), \(NT \land \exists x \varphi'_M(x) \land \exists x \varphi_M(x)\) is inconsistent, and since \(NT \vdash \exists x \varphi'_M(x)\), \(NT \land \exists x \varphi_M(x)\) is inconsistent so unsatisfiable. In summary, if \(M\) accepts \(\epsilon\) then \(NT \vdash NT \land \exists x \varphi_M(x)\) and if \(M\) rejects \(\epsilon\) then \(NT \land \exists x \varphi_M(x)\) is in UNSAT. This reduces the separation of \(L'_1\) and \(L'_2\) to the separation of \(\{\sigma \mid NT \vdash \sigma\}\) and UNSAT. Thus, \(\{\sigma \mid NT \vdash \sigma\}\) and UNSAT are recursively inseparable.
Facts about NT

Consider the set NT of axioms shown in class (see *Computational Complexity* by C. H. Papadimitriou). It turns out that NT is powerful enough to have the properties (i) - (iii). To see this, we need to understand which properties of $\mathbb{N}$ can be proven from NT. The following facts can be shown about NT (each requires quite a bit of work!):

- NT can prove every quantifier-free property of $\mathbb{N}$.
- NT can also prove every existentially quantified property of $\mathbb{N}$.
- NT can prove every prenex-bounded quantified property of $\mathbb{N}$.

These are sentences where universally quantified variables are bounded by terms using constants or previously quantified variables. For example, this sentence is prenex-bounded: $(\forall x < 9)\exists y(\forall z < 2 \times y)(x + z + 10 < 4 \times y)$

but this sentence is not prenex-bounded: $\forall x\forall y(x + y = y + x)$

- Finally, it turns out that the sentences $\varphi'_M$ and $\varphi_M$ needed to express acceptance or rejection of $\epsilon$ by M can be written using bounded universal quantification in prenex form, so are prenex bounded.

It follows from the above that (i) - (ii) hold. Property (iii) is shown directly from the construction of $\varphi_M(x)$ and $\varphi'_M(x)$ (omitted, see Papadimitriou).