Lecture #17: Proofs of Undecidability — Examples II
Lecture Presentation

Review of Preparatory Material

The Problem To Be Solved

In the preparatory material, a language "All_{TM}" was proved to be undecidable. This was the language of Yes-instances of the following decision problem:

The "Everything Accepted" Problem
Instance: A Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$.
Question: Does M accept every input string? That is, is $L(M) = \Sigma^\star$?

So one could also say that the "Everything Accepted" Problem was proved to be undecidable.

Consider the following the following decision problem:

The "Same Language" Problem
Instance: A pair of of Turing machines, M_1 and M_2, with the same input alphabet.
Question: Does M_1 and M_2 have the same language? This is, is $L(M_1) = L(M_2)$?

The goal of this lecture is to prove that the "Same Language" Problem is undecidable.

Encodings and Languages

Let $\Sigma_{\text{2TM}} = \Sigma_{\text{TM}} \cup \{\#\}$.

- A *pair* of Turing machines M_1 and M_2 can be encoded as a string $\alpha\#\beta \in \Sigma^\star_{\text{2TM}}$ where $\alpha \in \text{TM} \subseteq \Sigma^\star_{\text{TM}}$ is the encoding for M_1 and $\beta \in \text{TM} \subseteq \Sigma^\star_{\text{TM}}$ is the encoding for M_2.
- Let $\text{Pair}_{\text{TM}} \subseteq \Sigma^\star_{\text{2TM}}$ be the language of encodings of pairs of Turing machines

  $$M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_{0,1}, q_{A,1}, q_{R,1})$$

  and

  $$M_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_{0,2}, q_{A,2}, q_{R,2})$$

  with the same input alphabet Σ.
- Now let

  $$\text{E}_{\text{TM}} \subseteq \text{Pair}_{\text{TM}} \subseteq \Sigma^\star_{\text{2TM}}$$

  be the language including encodings of pairs of Turing machines M_1 and M_2, with the same input alphabet Σ, such that $L(M_1) = L(M_2)$.

Then Pair_{TM} is the ***language of instances*** and E_{TM} is the ***language of Yes-instances*** of the "Same Language" Problem that is defined above. Thus we must prove that the language E_{TM}, is undecidable to show that the "Same Language" Problem is undecidable.

What Kind of Mapping Should We Start With?

A Useful Turing Machine

A Mapping That Can Be Used

Proof That This Mapping Would Work

Adding Detail: Considering the Language Pair$_{TM}$ in More Detail

Adding Detail: What Would the Encoding of the "Useful Turing Machine" Look Like?

Confirming that the Usual Situation Arises — and Handling Other Input Strings

What Is The Function Being Used?

Two Claims, and Their Proofs

Proving That f is Computable

One More Claim To State and Prove

A (Somewhat) "High-Level Algorithm" To Compute This Function

Additional Information To Provide

Finishing Things