Lecture #16: Proofs of Undecidability — Examples I

Lecture Presentation

Review of Preparatory Material

The Problem To Be Solved

The previous lecture considered two decision problems:

Acceptance Problem
Instance: A Turing machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

and an input string $\omega \in \Sigma^\star$
Question: Does M accept ω?

and

Halting Problem
Instance: A Turing machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

and an input string $\omega \in \Sigma^\star$
Question: Does M halt, when executed on input ω?

with languages of Yes-instances A_{TM} and $\mathit{Halt}_{\text{TM}}$, respectively. The goal, for this lecture presentation, is to show that the “Acceptance Problem’ is reducible to the “Halting Problem Problem”, that is,

$$A_{\text{TM}} \preceq_{\text{M}} \mathit{Halt}_{\text{TM}}.$$

What Kind of Mapping Should We Start With?

A Mapping That Can Be Used

Proof That This Mapping Would Work

Adding Detail: Considering Encodings

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