Lecture #18: Proofs of Undecidability — Examples

What To Do Before the Lecture

1. Complete the preparatory viewing or reading for Lecture #18 — nothing that each video will probably be understandable if you play it at double speed.
2. ***Print*** and read through the rest of this document and — if you have time — try to solve the problems!
3. The supplemental documents are for interest only. They provide additional details about the proof of undecidability included in the second set of lecture slides for this lecture as well as another example of a proof of undecidability using a man-one reduction.

Problems To Be Solved

The following decision problems have now been introduced:

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?

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?

These have 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