Lecture #16: Proofs of Undecidability — Examples I
What Will Happen During the Lecture

Goals of this lecture presentation will be to help students understand ***many-one reductions***, how to show that these exist, and how to use these reductions to prove that languages are undecidable, or or unrecognizable — by considering a problem that might be suitable for an assignment in this course.

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 previous lecture included a proof that the "Halting Problem" was reducible to the "Acceptance Problem", that is,

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

Unfortunately, this many-one reduction cannot be used to prove that the language $\mathit{Halt}_{\mathsf{M}}$ is undecidable.

The lecture presentation for this lecture will be used to establish a more useful many-one reduction, to show that the "Acceptance Problem" is reducible to the "Halting Problem" as well, that is,

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