Lecture #17: Proofs of Undecidability — Examples II
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.

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 $L(M_1) = L(M_2)$?

During the lecture presentation an alphabet and encoding scheme, that can be used to define both the language of instances and the language of Yes-instances for this decision problem, will be described. After explaining why the language of instances is decidable, a many-one reduction will be used to show that the language of Yes-instances of this decision problem is ***undecidable***. That is, it will be shown that the "Same Language" problem is undecidable.