Universal Turing Machines and First Hard Problem
Supplement for Preparatory Viewing

An Undecidable Language

A ***proof by contradiction*** is used to prove that the language A_{TM} is undecidable — so the proof begins with the assumption that this language ***is*** decidable. It follows that there exists a Turing machine M_{ATM}, with input alphabet Σ_{TM}, such that M_{ATM} decides A_{TM}.

One should then consider a Turing machine, M_D, which implements the following algorithm.

On input $\mu \in \Sigma_{\mathsf{TM}}^{\star}$:

1. `if` ($\mu \in L_{\mathsf{TM}}$ and encodes a Turing machine M_μ with input alphabet Σ_{TM}) {
2.   `if` (M_μ accepts μ) {
3.   `reject` μ
     } `else` {
4.   `accept` μ
     }
   } `else` {
5.   `reject` μ
   }

This Turing machine must be encoded by some string $\mu_D \in \Sigma_{\mathsf{TM}}^{\star}$. A ***contradiction*** is then obtained by considering the following question: Does M_D accept its encoding, μ_D?

An Unrecognizable Language

It has been proved, at this point, that the language A_{TM} is recognizable, but not decidable. A ***proof by contradiction*** is used to prove that the language

$$A_{\mathsf{TM}}^{C} = \{\mu \in \Sigma_{\mathsf{TM}}^{\star} \mid \mu \notin A_{\mathsf{TM}}\}$$

is unrecognizable — so the proof begins with the assumption that this language *is* recognizable. Since A_{TM} is also recognizable it follows that there exist Turing machines M_Y and M_Y — both with input alphabets Σ_{TM} — such that $L(M_Y) = A_{\mathsf{TM}}$ and $L(M_N) = A_{\mathsf{TM}}^{C}$.

The proof, described in the lecture video, refers to an algorithm that decides membership in A_{TM} by carrying out executions of M_Y and M_N on an input string $\mu \in \Sigma_{\mathsf{TM}}^{\star}$ ***in parallel***. This corresponds to the following algorithm — which can be implemented using a multi-tape Turing machine with two tapes:

On input $\mu \in \Sigma^{\star}$ {
1. Write a copy of μ on the second tape, restoring the copy of μ on the first tape afterwards (so that both store a copy of μ) with both tape heads at the left end of their tapes.
2. Use the finite control to remember that both M_Y and M_N are in their start states.
3. `while` (`true`) {
4.   Use Tape #1 and the finite control to carry out the next step in the execution of M_Y on input μ.
5.   `if` (M_Y accepted, at this point) {
6.   `accept`
7.   } `else if` (M_Y rejected at this point) {
8.   `reject`
    }
9.   Use Tape #2 and the finite control to carry out the next step in the execution of M_N on input μ.
10.   `if` (M_N accepted at this point) {
11.   `reject`
12.   } `else if` (M_N rejected at this point) {
13.   `accept`
    }
  } // *End of Loop*
}

It can be argued that a Turing machine implementing this algorithm would ***decide*** the language A_{TM} — giving us a contradiction, since it has already been proved that the language, A_{TM}, is undecidable.