Lecture notes for undecidability class

John MacCormick, Dickinson College

Lemma 1. Suppose we are given a Turing machine T that always halts, with output either "y" or "n". Then we can produce a new Turing machine T' that enters an infinite loop whenever T would output "y", and outputs "n" whenever T would output "n".

Proof. Edit each final state so that the machine goes into a loop if the tape contains a "y". □

Lemma 2. Suppose we have a universal Turing machine T, that takes as input strings of the form $w_M w$, where w_M is a description of the Turing machine M, and w is a string on M's alphabet. (By the definition of a universal Turing machine, the output of T when given input $w_M w$ is the same as the output of M when given input w. That is, T simulates M with input w.) Then we can produce a new Turing machine T', that takes as input *only* w_M, and simulates M with input w_M. (That is, T' simulates M with input w_M.)

Proof. Just copy the input string and run T. □

Definition of the Halting Problem: Given a Turing machine T and a string w, determine whether T halts on input w.

Theorem. The halting problem is undecidable.

Proof. Suppose not. Let H be a Turing machine that decides the Halting Problem. So H takes as input $w_M w$, and outputs "y" if M halts on input w, and "n" if it doesn't. Use Lemma 1 to transform H into I, which enters an infinite loop if M halts on input w, and halts with output "n" if it doesn't. Use the same trick as in the proof of Lemma 2 to transform I into J, which enters an infinite loop if M halts on input w_M, and halts with output "n" if it doesn't.

Now consider: what does J do on input w_J? It

- enters an infinite loop iff J halts on input w_J, and
- halts with output "n" iff J doesn't halt on input w_J.

Every possible behavior of J produces a contradiction. Therefore, J cannot exist. Thus I and H cannot exist either, contradicting our assumption that H—a Turing machine that decides the Halting Problem—exists. □

Note 1. The Turing machine J described above is normally given the symbol D, in recognition of the fact that this type of proof is often called a *diagonalization* argument.

Note 2. For the EZquiz, you can use the following abbreviated form of the proof of the undecidability of the Halting Problem: Suppose not. Then there exists a Turing machine H that solves the halting problem. Specifically, H outputs "y" on input $w_M w$ if M halts on input w, and otherwise outputs "n". By making some simple changes to H, we can produce D, which enters an infinite loop on input w_M if M halts on input w_M, and otherwise halts. Consider the behavior of D on input w_D. By the definition of D, this computation halts if and only if it does not halt. This contradiction tells us that neither D nor H can exist, proving the desired result.