COMP-170: Homework #2

Ben Tanen - February 5, 2017

Problem 4

A Turing machine is *Left Out* if it does not have the ability to move the tape head left during its computation. Prove that this model is not equivalent to the standard Turing machine model.

* * *

Given a *Left Out* machine, since our tape is of infinite length, we logically know there is no combination of right tape movements that will equate to the computation we need from a left tape movement. However, to further prove this, consider the following.

Proof by Contradiction: We would like to show that no number of finite right movements on a *Left Out* machine can decide all the languages a traditional Turing Machine can. For example, we know that without left tape movements, a finite machine could not decide the L from problem 1. However, for this proof, let's assume we are given such a *Left Out* machine w that can decide $L = \{x\#y \mid x, y \in \{0,1\}^* \text{ and } |x| < |y|\}$.

We are given w and are told that it can decide L, which means given any $s \in L$, simulating w on input s causes w to accept s, and given any $s \notin L$, simulating w on input s causes w to reject s. Without left movements, we can see that w must be able to hold enough states to remember the number of cells in our input (in order to compare $|x|$ to $|y|$). To do this (without left movements), given some x (where x is part of our input string s) of length n, w would need at least n states.

Let n be equal to the size of Q in w, or the number of states in w. Now consider the string $s \in L$ where $s = x\#y$ such that $x = \{0,1\}^{n+1}$ and $y = \{0,1\}^{n+2}$. This means s has a length of $2n + 4$. We know s is a finite string in L that should be accepted when simulated on w. However, because w only has n states, we know that w would run out of countable states before it could even begin to compare $|x|$ against $|y|$. As a result, we can see that w could not successfully simulate on s.

Since $s \in L$ and w cannot successfully decide s, we can see that w cannot actually decide L, which contradicts the claim that w does indeed decide L. As a result, because we can always find some $s \in L$ that would not work on any finite *Left Out* machine w, we can see that *Left Out* machines cannot possibly decide all of the same languages traditional Turing machines can, namely L from problem 1. Thus we can see that *Left Out* machines and Turing machines are not equivalent. ⊠