CS 4820, Spring 2014	**Homework 9, Problem 2**
Name: Piyush Maheshwari	
NetID: pm489	
Collaborators: None	

2. *(10 points)* Recall the halting problem: An instance of the halting problem is a string of the form $x\#y$ and the goal is to decide if the Turing machine M_x encoded by the string x halts on input y. (If M_x halts on y, then $x\#y$ is a YES instance. Otherwise, $x\#y$ is a NO instance.)

Show that every Turing machine fails to solve the halting problem on an infinite number of instances. (A Turing machine M *fails to solve* the halting problem for instance $x\#y$ if it loops on input $x\#y$ or if it produces the wrong answer, i.e., it rejects in case that $x\#y$ is a YES instance or it accepts in case that $x\#y$ is a NO instance.)

Solution

Proof. We will sketch this proof by contradiction. Suppose that there exists some turing machine M which fails to solve the Halting Problem on a finite number of instances. Let the set of values on which M fails be S. Let the set $B =$ U - {S} which is basically the set of all problems not in B (U stands for universe of all problems)

Consider another TM M' which basically checks if a given input is present in the S or not. Using M' and M we can solve all problems not in S. Now consider the problems in S. We know that M fails on each of the problem in this set, but we don't know if M loops or gives an incorrect answer. But since S is a finite set, we know that there exists a mapping from each element s $\in$ S to {YES, NO} where YES means that s is a YES instance of the Halting problem and NO means that s is a NO instance of the Halting problem. This means that if the cardinality of S is n, there exists 2^n such mappings, each of which can be represented by a turing machine. We also know that since each problem either halts or not, one of these 2^n mappings (turing machines) would correctly classify each instance s $\in$ S. Let that turing machine be M''. We don't know how to find such a turing machine, but we definitely know that one such machine would exist since there are only a finite number of mappings.

Now build another turing machine H which given an input first checks if that input belongs to S or not by running machine M'. Then if that input does not belong to S, it runs that input of machine M to find the correct answer (M will definitely halt). Otherwise if the input does not belong to the set S, run machine M'' and output the answer (M'' will definitely halt as well).

Now using machine H we have solved the Halting problem. But we know Halting problem is un-decidable (proved in class) and there does not exist a turing machine which can solve the halting problem for all input instances. This means that we have arrived at a contradiction and our initial assumption about the existence of M was false. Hence every Turing Machine fails to solve the halting problem on infinite number of instances. $\square$