CS 4820, Spring 2017 Homework 10, Problem 1

Name: Yuxiang Peng

NetId: yp344

Collaborator: jl3455, zl542

Solution:

Suppose that M is a Turing machine and that the set F, consisting of all halting problem instances which M fails to solve, is finite. Define a new Turing machine M' which operates as follows. On input x:y, M' first checks whether x:y belongs to F. If so, M' outputs the correct halting problem solution for x:y; in other words, it accepts or rejects x:y according to whether or not Mx halts on input y. To achieve this, both the contents of the set F and the list of correct halting problem solutions for every input $x:y \in F$ are hard-coded into M'(i.e. into its state set and transition rules), which is possible because F is a finite set. In case x:y does not belong to F, then M' simulates M on input x:y and accepts or rejects x:y if and only if M does so.

We claim that M' accepts all strings x:y that are yes instances of the halting problem, and rejects all others; in other words, we claim that M' decides the halting problem. There are two cases: for strings x:y that belong to F, M' correctly decides the halting problem on x:y because the correct answer is hard-coded into M'; for strings x:y that do not belong to F, M' correctly decides the halting problem on x:y because M does so; this follows from the definition of the set F.

Suppose the inputs allows M don't have "yes" transitions for every state but still output "yes" on all inputs. The algorithm is basically helping in simulating M on x;y and watch it giving the wrong output. Those inputs which have the same yes output with the given state are not in F set. M halts and continue to operate for the next input until one different no output appears, which guarantee the finite steps. In this case, M steps into infinite loops and M' operates to halts and return this input with the different yes output. Obviously, we at least can find one instance which is a halting problem instance x; y such that M fails to solve x; y. It does not disprove the halting problem is undecidable. In other words, all machines fail to decide the halting problem, which means there exists an instance of the halting problem, x;y, such that M fails to solve x;y. But You can say that a machine fails to solve one specific instance of the halting problem.