CS 4814: Homework 1

due 09/02, 11:59pm

Each problem is worth 20 points. The first two problems are about Turing machines as a computational model. Here your solutions should give details about how the machine operates on the tape and what kind of states and transitions it has. The third problem is about computation in general. Here you can describe your algorithms and reductions at a higher level, in particular you don't have to compile them down to a Turing machine or pseudo-code level.

Problem 1

Exercise 1.10 in the textbook.

(You may ignore the remark about time-constructability of $T(n)$.)

Problem 2

Let `LOOKUP`: $\{0,1,\#\}^* \to \{0,1,\#\}$ be the function that given $x\#i$ (where x is a bit string and i is a natural number in binary representation) outputs the i-th bit of x if $i \leq |x|$ and # if $|x| < i$. Show that the function `LOOKUP` is computable in polynomial time, that is, give a Turing machine with polynomial running time that computes `LOOKUP`.

Your solution should include a high level description of how the Turing machine operates on the tape in order to compute the desired function and an analysis of its running time. You may use a multi-tape Turing machine if you prefer (see the textbook for the definition).

Your solution should also explain what kind of states the machine has in order to achieve the desired behavior. For the latter part, it is enough to implement the transition function in a higher-level programming language, print the full transition function table, and simulate the machine on some inputs, similar to the examples in the IPython notebook at `https://github.com/dsteurer/cs4814fa15/tree/master/turing`. (You can just start from the code in the repository.)

Problem 3

For each of the following decision problems, determine if it is decidable or not. Prove your answers correct. You may use Rice's theorem. (But don't forget to check that the conditions of the theorem hold.)

(a) Given a (single-tape) Turing machine M and a bit string x, does M's head ever move to the left during the computation on input x?

(b) Given a Turing machine M and an input x, does M ever re-enter its start state in the course of its computation on input x?

(c) Given a Turing machine M, is the language $L(M)$ finite?