CSE 431 Spring 2017
Assignment #3

Due: Friday, April 21, 2017

Reading assignment: Read Chapter 5 of Sipser's text. We will cover section 5.3 before we cover computation histories in section 5.1.

Problems:

1. Define $INFINITE_{CFG} = \{\langle G \rangle \mid$ the language that context-free grammar G generates is infinite$\}$. Prove that $INFINITE_{CFG}$ is decidable.

2. A *useless state* in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Formulate this problem as a language and show that it is undecidable.

3. Let $ODD_{TM} = \{\langle M \rangle \mid M$ is a TM that accepts an odd number of strings$\}$. Show that ODD_{TM} is undecidable.

4. Suppose that $A \subseteq \{\langle M \rangle \mid M$ is a decider TM$\}$ and that A is Turing-recognizable. (That is, A only contains descriptions of TMs that are deciders but it might not contain all such descriptions.)
   Prove that there is a decidable language D such that $L(M) \neq D$ for any M with $\langle M \rangle \in A$. (Intuitively, this means that one couldn't come up with some restricted easy-to-recognize format for deciders that captured all decidable languages.)
   (Hint: You may find it helpful to consider an enumerator for A.)

5. (Bonus) Let $\Gamma = \{0, 1, blank\}$ be the tape alphabet for all TMs in this problem. Define the *busy beaver function* $BB : \mathbb{N} \to \mathbb{N}$ as follows: For each value of k, consider all k-state TMs that halt when started with a blank tape. Let $BB(k)$ be the maximum number of 1s that remain on the tape among all of these machines. Show that BB is not a computable function.

6. (Bonus) For a PDA $M = (Q, \Sigma, \delta, q_0, F)$ we say that a string $\alpha \in \Gamma^*$ is a *possible stack of* M if there is some input and some choice of moves of M such that α appears as M's stack contents during its computation. Prove that the languge $L \subseteq \Gamma^*$ of possible stacks is regular. (This fact is actually important for certain software verification systems since it allows one to consider the set of possible call stacks using only a finite state machine.)