CSCE 355, Assignment 6

The two exercises marked "optional" mostly build on one another and on previous exercises. The exercises should be considered in order. The first two exercises have solutions on the textbook's website.

1. Textbook Exercise 8.1.1(a): Give reductions from the hello-world problem to each of the problems below. Use the informal style of this section for describing plausible program transformations, and do not worry about the real limits such as maximum file size or memory size that real computers impose.

   *! a) Given a program and an input, does the program eventually halt; i.e., does the program not loop forever on the input?

2. Textbook Exercise 8.2.4(a,b), slightly modified with original wording in brackets: In this exercise we explore the equivalence between function computation and language recognition for Turing machines. For simplicity, we shall consider only functions from nonnegative integers to nonnegative integers, but the ideas of this problem apply to any computable functions. Here are the two central definitions:

   - Define the *graph* of a function f to be the set of all strings of the form $[x, f(x)]$, where x is a nonnegative integer in binary, and $f(x)$ is the value of function f with argument x, also written in binary.
   - A Turing machine is said to *compute* function f if, started with any nonnegative integer x on its tape, in binary, it halts (in any state) with $f(x)$, in binary, on its tape.

   Answer the following, with informal, but clear constructions.

   a) Show how, given a TM that computes f, you can construct a TM that recognizes [orig: accepts] the graph of f as a language.

   b) Show how, given a TM that recognizes [orig: accepts] the graph of f, you can construct a TM that computes f.

3. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a Turing machine. As usual, we assume that $Q \cap \Gamma = \emptyset$. Recall that an *instantaneous description* (ID) of M is any string over the alphabet $Q \cup \Gamma$ containing exactly one symbol from Q. Give a DFA that recognizes the language of all IDs of M.

4. Let M be as in the last problem. Recall that if ID_1 and ID_2 are IDs of M, then $\text{ID}_1 \vdash \text{ID}_2$ means that ID_2 results from ID_1 by a single step of M. Let \$ be some symbol not in $Q \cup \Gamma$.

The languages

$$L_1 := \{w\$x^R \mid w \text{ and } x \text{ are IDs of } M \text{ and } w \vdash x\}$$
$$L_2 := \{w^R\$x \mid w \text{ and } x \text{ are IDs of } M \text{ and } w \vdash x\}$$

are both context-free. (Recall that x^R and w^R are the reversals of strings x and w, respectively.) Describe how to build CFGs for L_1 and L_2, given a complete description of M.

5. (Optional) Let M and \$ be as in the last problem. Describe a PDA P that, given as input some string of the form $w\$x$, where w and x are IDs of M, accepts if and only if $w \not\vdash x$. In other words, P accepts iff it finds a "mistake" in M's transition from w to x. If you want, you may assume that the IDs w and x cover the exact same portion of M's tape, but this assumption is not necessary. You can stick to a high-level description of P instead of a formal one.

6. (Optional) Prove that there is no algorithm that decides, given a PDA P, whether P accepts all strings over its input alphabet. Hint: Given a TM M as in the last problem and an input string x, design a PDA P that accepts a string w if and only if w is *not* of the form $\mathrm{ID}_0\$\mathrm{ID}_1\$\cdots\$\mathrm{ID}_n$, where $\mathrm{ID}_0 \vdash \mathrm{ID}_1 \vdash \cdots \mathrm{ID}_n$ is a complete trace of a halting computation of M on input x. (P can be constructed using ideas from the last problem.) Then P accepts all strings if and only if M does not halt on input x. Conclude that any decision procedure for the former statement can be used to decide the latter, which we know is undecidable.