Homework 7 – Due Thursday, March 17, 2016 before the lecture

Please refer to the general information handout for the full homework policy and options. *Your solution to each problem should be handed in on a separate sheet of paper.*

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the instructor if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Problems Please practice on exercises and solved problems in Chapter 4. The material they cover may appear on exams.

1. (**Decidable languages**)

   (a) **(**$\text{REVERSE}_{\mathsf{DFA}}$**)** Consider the problem of determining if the languages of two given DFAs are reverses of each other. Recall that $L^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in L\}$. Formulate this problem as a language $\text{REVERSE}_{\mathsf{DFA}}$ and show that it is decidable.

   (b) (**Decidable languages**) Consider the following problem: you are given an NFA and a PDA and you would like to know whether there exists a string that they both accept. Formulate this problem as a language and prove it is decidable.

2. (**Recognizable languages)**

   (a) **(Complement of** EQ_{CFG}**)** Prove that the complement of EQ_{CFG} is Turing-recognizable.

   (b) **(**2016_{TM}**)** Consider the problem of determining whether the language of a given Turing machine contains at least 2016 strings.

      i. Formulate this problem as a language 2016_{TM}.

      ii. Show that 2016_{TM} is Turing-recognizable.

3. (**Uncountable sets and diagonalization**)

   (a) Let $\mathcal{L}$ be the set of all languages over alphabet $\{0\}$. Show that $\mathcal{L}$ is uncountable, using a proof by diagonalization.

   (b) Your friend told you that he found a new C++ library that contains many useful functions. One example is a function `halt` which takes two arguments: a program `main.cpp` and a valid input string x for that program. It returns 1 if `main.cpp` produces an output on input string x, and returns 0 if `main.cpp` runs forever on x. You want to convince your friend that `halt` cannot be always correct. However, your friend does not want to hear about TMs because they cannot possibly be relevant to C++ programs.

   Give a diagonalization argument (similar to that on p. 207 of Sipser) to convince your friend that function `halt`, as specified above, does not exist. Your analogue of TM D on p. 207 should be a C++ program `src.cpp`. *Caution:* You cannot model it directly on TM D because your friend's claim is analogous to saying that $\mathsf{HALT}_{\text{TM}}$ is decidable, not that A_{TM} is decidable.

Your program may call function `fread` that reads a file and returns the content of a file as a string. If you do not know C++, you can use any programming language with C-like syntax (or consult Mahdy on which ones he can grade).

4* (**Optional, no collaboration is allowed**) If you wish, you can resubmit your solution to the optional problem on the previous homework. (Some students asked for more time.) We will grade the most recent solution.