CS 332: *Elements of Theory of Computation* Professor Sofya Raskhodnikova
Boston University October 11, 2019

Homework 5 – Due Friday, October 18, 2019 before noon

This homework contains 3 mandatory and 1 optional problem, worth 10 points each.

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 3. The material they cover may appear on exams.

1. (**Tape-saving TM**) A *tape-saving* Turing machine is similar to an ordinary (deterministic) one-tape Turing machine, but it can read and write on both sides of each tape square: front and back. In addition, it can move the head to the middle of nonblank portion of the tape. Specifically, if the first s squares of the tape are not blank, then the head can move to square $\lceil s/2 \rceil$.

   At each step, the head of the tape-saving TM can move left (L), right (R), switch sides of the current square (S) or jump to the middle (J).

   (a) Give a formal definition of the form of the transition function of tape-saving TM. (Modify Part 4 of Definition 3.3 on page 168 of the textbook.)

   (b) Show that tape-saving TMs recognize the class of Turing-recognizable languages, that is, they have exactly the same power as ordinary TMs.

2. (**FIFO Automaton**) A *FIFO automaton* is defined like a PDA except that instead of the stack, it has a first-in-first-out queue. When a FIFO automaton pushes a symbol, it is stored on the right-hand end of the queue; when it pops, the leftmost symbol is read and removed from the queue. Show that a language can be recognized by a deterministic FIFO automaton iff the languages is Turing-recognizable.

   Hint 1: Show how to simulate a FIFO automaton by a TM and vice versa.

   Hint 2: FIFO automaton can hold a small number of symbols in "memory" by using its states.

3. (**Decidable languages**)

   (a) (COMPLEMENT$_{\mathsf{DFA}}$) Consider the problem of determining if the languages of two given DFAs are complement of each other.

      i. Formulate this problem as a language COMPLEMENT$_{\mathsf{DFA}}$ and show that it is decidable.

      ii. Why does a similar approach fail to show that COMPLEMENT$_{\mathsf{PDA}}$ is decidable? (This language is defined analogously to COMPLEMENT$_{\mathsf{DFA}}$, with PDAs instead of DFAs as inputs.)

   (b) Let $A = \{\langle G, R\rangle \mid G \text{ is a context-free grammar, } R \text{ is a regular expression, and } L(G) \subseteq L(R)\}$.

      i. Give an example of G and R such that $\langle G, R\rangle \in A$.

      ii. Show that A is decidable.

4* (**Optional, no collaboration**) Let A be a Turing-recognizable language which is not decidable. (We will prove later in the course that such languages exist.) Consider a TM M that recognizes A. Prove that there are infinitely many strings on which M loops.