Professor Sofya Raskhodnikova

March 18, 2016

Homework 8 – Due Thursday, March 24, 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.

Exercises Use these exercises to practice, but do not hand them in. The material they cover may appear on exams.

1. Please practice on exercises and solved problems in Chapter 5.
2. (**True/False and Justify**) Decide whether each statement is true or false and briefly justify your answer.

T F There exists an enumerator TM that prints a set S of TM descriptions, such that S includes descriptions of TMs that decide infinitely many different languages.

T F There exists an enumerator TM that prints a set S of TM descriptions, such that S includes a description of at least one TM for each language over alphabet $\{0, 1\}$.

T F There exists an enumerator TM that prints the set S of TM descriptions that consists of descriptions of all TMs whose language is empty.

T F If A is a decidable language and B is a Turing-recognizable language, then $A \setminus B$ must be Turing-recognizable.

T F A two-dimensional Turing machine is like an ordinary Turing machine except that its tape storage consists of a two-dimensional tape, corresponding to the upper right quadrant of the plane. Each tape cell is a unit square. In one step, the single tape head can move left, right, up or down. A two-dimensional TM starts with its input written on consecutive cells, starting from the lowest leftmost cell (where the head is located) and going right. The class of languages recognized by two-dimensional Turing machines is exactly the Turing-recognizable languages.

Problems

1. (**Undecidable languages**) For each of the parts, formulate the given problem as a language and prove it is undecidable.
   (a) (**Sort-checker TM**) A TM is a *sort-checker* if it accepts a string if and only if this string is a comma-separated list of binary numbers, appearing in the sorted order (from smallest to largest). For example, it accepts "`001,111,1001`" and "`0,011,100111,11111111110`", but not "`1,0`". (It also accepts the empty string.) The problem is to determine whether a given TM is a sort-checker.

(b) (**464-TM**) You are given a TM and you would like to determine whether there exists some input w on which this TM moves its head to the left from the tape cell 464. (We number the tape cells from left to right, starting from 1.) Note that w is not given to you.

2. (**OVERLAP$_{\mathsf{DFA,TM}}$**) Let $\text{OVERLAP}_{\mathsf{DFA,TM}} = \{\langle D, M\rangle \mid D$ is a DFA and M is a TM and $L(D) \cap L(M) \neq \emptyset\}$.

   (a) Prove that $\text{OVERLAP}_{\mathsf{DFA,TM}}$ is undecidable.

   (b) Prove that $\text{OVERLAP}_{\mathsf{DFA,TM}}$ is Turing-recognizable.

   (c) Is $\overline{\text{OVERLAP}_{\mathsf{DFA,TM}}}$ Turing-recognizable? Prove or disprove.

3. (**Prime-length TM**) Let $\text{PL}_{\mathsf{TM}} = \{\langle M\rangle \mid M$ is a TM that accepts all strings whose length is a prime number and rejects all other strings$\}$. Prove the following statements about PL_{TM}.

   (a) PL_{TM} is not Turing-recognizable.

   (b) $\overline{\text{PL}_{\mathsf{TM}}}$ is not Turing-recognizable (i.e., PL_{TM} is not co-Turing-recognizable.)

4*. (**Optional, no collaboration**) Let $\Gamma = \{0, 1, \sqcup\}$ be the tape alphabet for all TMs in this problem. We number tape squares $1, 2, 3, \ldots,$ starting from the leftmost square. For each $k \in \mathbb{N}$, consider all k-state TMs that halt when started with a blank tape. Let $f(k)$ be the maximum position number of the rightmost square visited by such a TM when started with a blank tape. (The maximum is taken over all such TMs.) Show that function f is not computable.

   Hint: Use the fact that A_{TM} is undecidable.