CS 332: *Elements of Theory of Computation* Professor Sofya Raskhodnikova
Boston University October 25, 2019

Homework 7 – Due Friday, November 1, 2019 before noon

This homework contains 4 mandatory and 1 optional problem, worth 10 points each unless specified otherwise.

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 Please practice on exercises and solved problems in Chapter 4 as well as on the exercises below. 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) (**2019_{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 2019. (We number the tape cells from left to right, starting from 1.) Note that w is not given to you.

(b) (**Sorting TM**) A TM *correctly sorts* if, given a comma-separated list of binary numbers, it halts with the sorted (from smallest to largest) version of the list on its tape. (It does not matter what it does on other inputs.) Consider the problem of determining whether a TM *correctly sorts*. Formulate this problem as a language and prove it is undecidable.

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

   (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. (**OVERLAP$_{\mathsf{CFG}}$, 5 points**)
Let $\text{OVERLAP}_{\mathsf{CFG}} = \{\langle G_1, G_2\rangle \mid G_1 \text{ and } G_2 \text{ are } \mathsf{CFG}\text{s and } L(G_1) \cap L(G_2) \neq \emptyset\}$. Show that $\text{OVERLAP}_{\mathsf{CFG}}$ is undecidable by giving a reduction from the Post Correspondence Problem (see Section 5.2 of Sipser).

   Hint: Given an instance
$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\}$$
of the Post Correspondence Problem, construct CFGs G_1 and G_2 with the rules
$$\begin{aligned} G_1 &: \; T \to t_1 T \sigma_1 \mid \cdots \mid t_k T \sigma_k \mid t_1 \sigma_1 \mid \cdots \mid t_k \sigma_k \\ G_2 &: \; B \to b_1 B \sigma_1 \mid \cdots \mid b_k B \sigma_k \mid b_1 \sigma_1 \mid \cdots \mid b_k \sigma_k \end{aligned}$$
where $\sigma_1, ..., \sigma_k$ are new alphabet symbols. Prove that this reduction works.

4. (**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.)

5* (**Optional, no collaboration is allowed**) In this problem, you are asked to think about *LOSS* operations on languages. Each LOSS operation is specified by a set Σ of symbols. When the "*LOSS of* Σ" operation, denoted by $LOSS_\Sigma$, is applied to a string w, all characters in Σ disappear from w. For example, $LOSS_{\{\mathtt{1,3}\}}(\mathsf{121023}) = \mathsf{202}$ and $LOSS_{\{\mathtt{1,3}\}}(\mathsf{241222}) = \mathsf{24222}$, whereas $LOSS_{\{\mathtt{1,3}\}}(\mathsf{24222}) = \mathsf{24222}$. To apply $LOSS_\Sigma$ to a language, we apply it to every string in the language. For example, $LOSS_{\{\mathtt{0,1,3}\}}(\mathsf{0^*1^*2^*3}) = \mathsf{2}^*$. More formally,
$$LOSS_\Sigma(L) = \{LOSS_\Sigma(w) \mid w \in L\}.$$

   (a) Prove that the class of regular languages is closed under the LOSS operations.

   (b) Prove that the class of decidable languages is not closed under the LOSS operations.