Homework 7 – Due Tuesday, November 3rd, 2020 <u>before</u> 2:00PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff 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 3. The material they cover may appear on exams.

Problems There are 4 mandatory problems and one bonus problem.

1. (**Countable and uncountable sets**) Prove or disprove that each of the following sets is countable.
   (a) (**5 points**) The integer set $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$.
   (b) (**5 points**) The set of degree-d integer-coefficient univariate polynomials $\mathcal{P}_d$, that is each element in the set is a polynomial in variable x of the form $c_0 + c_1x + c_2x^2 + c_3x^3 + \cdots + c_dx^d$, where d is a non-negative integer and $c_0, ..., c_d$ are integers, called coefficients.
   (c) (**5 points**) The set of all integer-coefficient univariate polynomials $\mathcal{P} = \bigcup_{d \geq 0} \mathcal{P}_d$.
   (d) (**5 points**) The set of all finite sets of real numbers.
   (e) (**5 points**) The set of all possible English sentences.
   (f) (**5 points**) The set of all finite languages over the binary alphabet $\{0, 1\}$.
   (g) (**10 points**) The set of all languages over the unary alphabet $\{0\}$.

2. (**20 points**) Your friend told you that he found a new Python library that contains many useful functions. One example is a function `halt` which takes two arguments: a function `f` and a valid input x for that function. It returns 1 if `f` eventually returns on input string x, and returns 0 if `f` 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 Python programs.

   Give a diagonalization argument (similar to that on p. 207 of Sipser) to convince your friend that function `halt`, as specified above, cannot always terminate and give the correct answer. Your analogue of TM D on p. 207 should be a Python function (or program). *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 and D is used in the proof regarding A_{TM}.

   Your program may call function `halt`. If you do not know Python, you can use any programming language with C or Python-like syntax (or consult on Piazza for other languages).

   N.B. For this problem, assume the computer running these programs is allowed to call a function called `ask_human_for_ram` to ask for more RAM, for example, one memory stick of 4GB at a time. You can also assume that we have signed an agreement with the manufacturer and have an unlimited supply of RAM sticks for the computer.

3. (**20 points**) 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 that it is undecidable by reducing it to another undecidable language from the lectures.

4. (**Decidable/undecidable languages**) For each of the parts, formulate the given problem as a language and either prove or disprove it is decidable.
   Hint: Exactly one of them is decidable and the other is not.

   (a) (**20 points**) 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 2020. (We number the tape cells from left to right, starting from 1.) Note that w is not given to you.

   (b) (**20 points**) 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 *right* from the tape cell 2020.

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, $LOSS_{\{\mathtt{1},\mathtt{3}\}}(\mathtt{121023}) = \mathtt{202}$ and $LOSS_{\{\mathtt{1},\mathtt{3}\}}(\mathtt{241222}) = \mathtt{24222}$, whereas $LOSS_{\{\mathtt{1},\mathtt{3}\}}(\mathtt{24222}) = \mathtt{24222}$. To apply $LOSS_\Sigma$ to a language, we apply it to every string in the language. For example, $LOSS_{\{\mathtt{0},\mathtt{1},\mathtt{3}\}}(\mathtt{0}^*\mathtt{1}^*\mathtt{2}^*\mathtt{3}) = \mathtt{2}^*$. 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.