Theory of Computation: Final Exam

Arjun Chandrasekhar

Due Thursday, 04/28/2022 at 11:59 pm (50 points)

This is the midterm exam for CS 1502. You may only consult the following resources:

- Lecture slides
- Lecture notes
- Lecture recordings
- Any notes you took in class
- Your own written and programming assignment submissions
- Written and programming assignment solutions
- The course textbook (Sipser)
- Any email/discord correspondences that took place before this exam was released

You may **not** use any resources not explicitly listed above. In particular, you may not use any internet resources not listed above, **and you may not collaborate with other students.** If you have any clarification questions you should email the instructor.

You may reference any results that have already been proved in class, lecture slides, notes, the textbook, or homework assignments. You do not need to re-state the proofs for results that we have already established.

This exam is worth 50 points. You may notice that the total number of points adds up to 60 which is more than 50. Your total score will be based on the five problems that you did the best on. We will drop the problem on which you got the lowest score. You may attempt as many problems as you like in order to hedge your bets, but you will not receive more than 50 for successfully completing extra problems.

Please submit your solutions on Canvas. You will receive one point of extra credit for typesetting your assignment.

The assignment begins on the next page.

1. (10 points) Define a 2D-Turing Machine (2DTM) to be a Turing Machine whose tape is an infinite 2-dimensional grid (rather than an infinite linear tape). When the machine transitions, it may move left, right, up, or down. The 2D grid starts with the input string on a single row, and the head starts on the first character of the input. Everything else is the same. Figure 1 illustrates this model.

   Prove that a language is Turing-recognizable if and only if it is recognized by a 2DTM. (Hint: the 2DTM will only have a finite amount of non-empty rows in use at any given time).

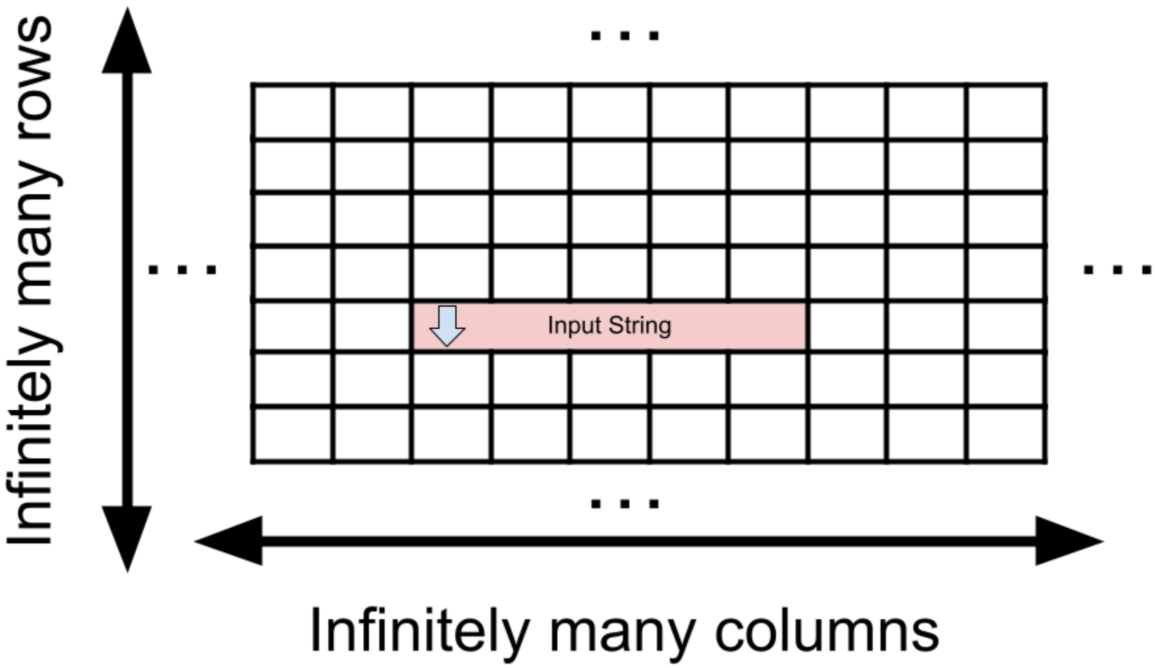

Figure 1: **2-Dimensional Turing Machine:** In the 2DTM model, we have an infinite 2D grid, and a tape head that can move up and down (in addition to left and right). The input starts out contiguously at an arbitrary row and column, and the tape head starts at the first symbol of the input.

2. (10 points) Recall that from homework 4 the DROP_OUT operation:

$$\text{DROP-OUT}(L) = \{w = xz | x, z \in \Sigma^*, xyz \in A \text{ for some } y \in \Sigma\}$$

   It is the set of all strings w that would be in L, except that they are missing a single character y. Put another way, there exists a way to split w into two substrings xz and then insert y in between xz, with the resulting string $xyz \in L$. However, we only receive w as input. We don't know how w should be split up into xy, we don't know what missing character y can be inserted in between xz to create a string $xyz \in L$.

   As an example, let $L = \{abc, 123\}$. Then

$$\text{DROP-OUT}(L) = \{ab, ac, bc, 12, 13, 23\}$$

   And if L were an infinite language then DROP-OUT(L) would also be infinite.

   Prove that Turing-recognizable languages are closed under the DROP_OUT operation.

3. (10 points) Prove that the following language is decidable

$$L = \{\langle D_1, D_2\rangle | D_1 \text{ and } D_2 \text{ are DFAs}, L(D_1) \subseteq L(D_2)\}$$

4. (10 points) Prove that that the following language is undecidable.

$$\text{ALL}_{\text{HALT}} = \{\langle M\rangle \mid M \text{ is a Turing machine, and } M \text{ halts on all inputs.}\}$$

5. Classify each of the following languages as either in P or NP-Complete. Justify your answer using the techniques we have discussed in this class.

   (a) (10 points) $L = \{\langle G, k\rangle | k \leq 10, G \text{ has a clique of size k}\}$

   (b) (10 points) $L = \{\langle G, k\rangle | k \geq 10, G \text{ has an independent set of size k}\}$

   Remember that to prove that L is NP-complete, you must prove that it is both in NP and that it is NP-Hard. Furthermore, to prove that L is NP-hard, remember that you must show that your reduction takes place in poly-time, yes maps to yes, and no maps to no (or yes maps to yes in the opposite direction).