Theory of Computation: Midterm Exam

Arjun Chandrasekhar

Due Thursday, 03/03/2022 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.

Any results that have already been established through other class materials - including lecture slides, lecture notes, instructional recordings, assignments, or the course text book - can be (re-)stated without (re-)proof. Please do not copy and paste proofs of results that we've already covered. You do not need to cite the result - if it's something we have already proven then you can just state the result and use it towards the problem you are trying to solve.

You may notice that this exam is worth 50 points, but the problems add up to 60 points. This is not an accident. You may attempt any combination of problems and parts of problems that you like. You will not receive more than 50 points (even if you get all the problems correct); but if you attempt more problems, you will be able to hedge your bets.

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. Classify each of the following languages as either regular or non-regular. Justify your answers using the techniques we have studied in this class.

   (a) (5 points) $L = \{a^{2n}b^{2m} | n, m \geq 0\}$

   (b) (5 points) $L = \{a^{2n}b^{2n} | n \geq 0\}$

   (c) (5 points) $L = \{a^n b^n c^n | n \geq 3\}$

   (d) (5 points) $L = \{a^n b^n c^n | n \leq 3\}$

2. (10 points) Suppose that we have a language L. Define the operation

$$\text{BOTHWAYS}(L) = \{w | w \in L \text{ and } w \in L^R\}$$

   Here, recall that L^R is the set of strings that are the reverse of the strings in L.

   Prove that the set of regular languages is closed under the BOTHWAYS operation. That is, prove that if L is a regular language, then BOTHWAYS(L) is also regular. You may use any techniques we have discussed in class.

3. (10 points) Define a **spooky expression** as follows:

   (a) $a \in \Sigma$

   (b) ϵ

   (c) $\emptyset$

   (d) $S_1 \cap S_2$ where S_1 and S_2 are spooky expressions (intersection)

   (e) S_1^c where S_1 is a spooky expression (complement)

   (f) $S_1 \circ S_2$ where S_1 and S_2 are spooky expressions (concatenation)

   (g) S_1^* where S_1 is a spooky expression (Kleene Star)

   In the spirit of Dr. Chandrasekhar's favorite holiday - Halloween - we will prove that a language L is regular if and only if it is described by a spooky expression.

   (a) Prove that if L is regular, then L can be described by a spooky expression. (**Hint**: Start with a regular expression for L.)

   (b) Prove that if L is described by a spooky expression, L is regular. (**Hint**: explain how to inductively construct a machine for L. You do not need a formal description.)

   Note: At first glance a spooky expression sure looks a lot like a regular expression, but its definition is technically different. In particular, spooky expressions do not include the union operator in their definition. Thus, you should not assume that spooky expressions describe regular languages just because regular expressions do.

4. (10 points) Define the operation XOR-SHUFFLE as follows:

$$\text{XOR-SHUFFLE}(A, B) = \{w = a_1 b_1 a_2 b_2 \ldots a_n b_n | a_1 a_2 \ldots a_n \in A, \textbf{ or } b_1 b_2 \ldots b_n \in B \textbf{ but not both}\}$$

   This is similar to perfect shuffle; the odd characters correspond to language A and the even characters correspond to language B. However, we only want to accept if **exactly one of the two substrings is a member of the corresponding language**.

   For example, let $A = a^*$ and $B = b^*$.

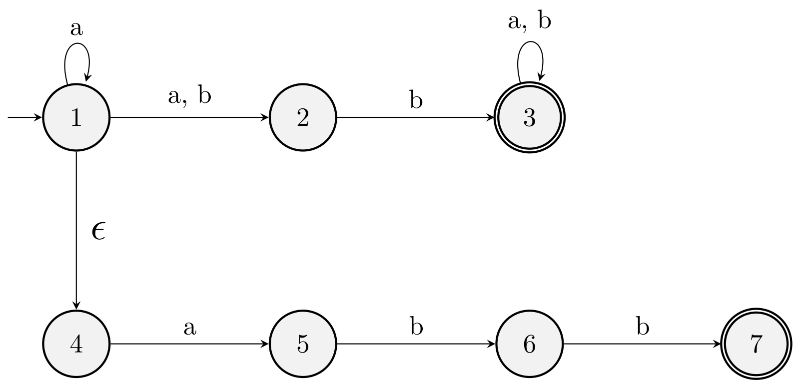
   - $ababaa \in \text{XOR-SHUFFLE}(A, B)$ because $aaa \in A, bba \notin B$
   - $abbbbb \in \text{XOR-SHUFFLE}(A, B)$ because $abb \notin A, bbb \in B$

- $bababa \notin$ XOR-SHUFFLE(A, B) because $bbb \notin A, aaa \notin B$. We need exactly one of them to work
- $ababab \notin$ XOR-SHUFFLE(A, B) because $aaa \in A, bbb \in B$. We need exactly one of them to work

Prove that regular languages are closed under XOR-SHUFFLE. Make sure your proof generalizes to *all* regular languages - not just the two in the example above. If you try to simply prove that the two languages used in the example are closed under XOR-SHUFFLE then you will not receive any points.

If you solve this problem by constructing a machine, make sure you give a complete formal description to receive full credit.

5. (10 points) Define a **restricted-NFA (rNFA)** to be the same as a regular NFA (same formal description), with one difference: it accepts a string if, after reading the characters of the string, the machine can reach one **and only one** accept state. For example, consider the following rNFA:



This will accept the string aab, because after reading aab the machine can end up in states 2, 3, and 6, and exactly one of of those is an accept state. The machine will also accept bba because after reading bba the machine can only be in state 3. However, **the machine will not accept abb**. After reading abb, the machine could be in states 3 or 7. The string abb can reach more than one accept state, so the machine rejects abb.

(a) Prove that a language is regular if and only if it is recognized by a restricted NFA. Remember, there are two directions. Give a formal description for any machine you construct.

(b) Recall the XOR operation from assignment 1

$$A \oplus B = \{w | w \in A \text{ or } w \in B \ \underline{\text{but not both}}\}$$

Use restricted NFAs to prove that regular languages are closed under the XOR operation. Your proof must use a restricted NFA. You will not receive any credit if you attempt to use De Morgan's laws. You will also not receive any credit if you attempt to use the Cartesian product construction to run two DFAs in parallel. Your proof must use nondeterminism, and it must take advantage of the unique properties of a restricted NFA.