Theory of Computation

CS-312 **Summer 2020**

Exam 1 **DUE TUESDAY, 4 August 2020, 11:59pm**

Instructions

- Complete all problems separately; each problem indicates the number of points possible. Show your work; partial credit may be awarded.
- All problems are to be completed individually. No collaboration with others is permitted.
- This exam is *open-book*, *open notes*. You are permitted to consult the textbook, your own class notes, class handouts, and class homework and solution sets. All information available from the course Blackboard site is permitted.

  You may use JFLAP to develop your solutions, *except where using JFLAP would defeat the purpose of the question.* You may *always* use the JFLAP automata editor to draw your solutions.

  No other outside sources, including Internet sources, are permitted. No collaboration with others is permitted. Violations are subject to severe penalties, up to and including course failure.
- Your answers must be submitted electronically via Blackboard by 11:59pm on Tuesday, 4 August (4th Tuesday). ***Retain your electronic copy of your completed answers, in case of difficulties.***

By submitting your examination, you certify that you have neither given nor received any unauthorized assistance on this examination.

Note: In all problems on this exam, let $\Sigma = \{0, 1\}$ be the alphabet from which strings are produced.

1. *10 points each.* Give *deterministic* finite automata which accept the following languages:

   (a) Strings in which every 0 is immediately preceded and followed by a 1. (Note: strings without any 0s, including ϵ, are in this language.)

   (b) Strings which end in 00.

2. *10 points each.* Give regular expressions for the following languages:

   (a) Strings where the number of 1s is divisible by 3.

   (b) Odd-length strings that start with 1, or even-length strings that start with 0.

3. *10 points.* Convert the following regular expression to a *nondeterministic finite automaton*:

$$1^*(0 + 10)^*1$$

   (**Note:** Do *not* use the automated regular expression conversion tool in JFLAP. You *may* use the JFLAP editor to draw and test your solution.)

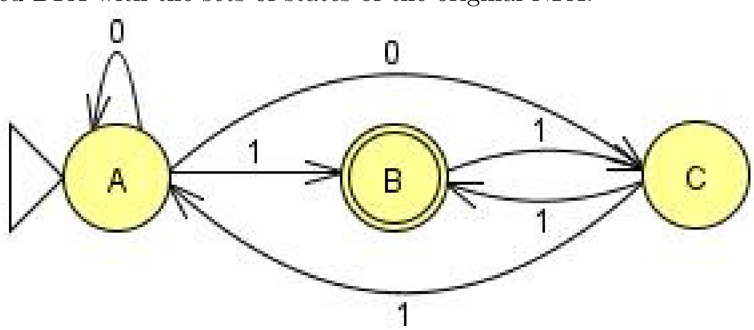
4. *10 points.* The POSIX standard for regular expressions includes many extensions to regular expressions. One of these is the *numerical quantifier*.

   Let r be a regular expression and n be a non-negative integer. The expression $r\{n,\}$ matches *n or more consecutive occurrences of r*. For example, $(01)\{3,\}$ matches 010101 and 0101010101, but not 0101.

   Show that the numerical quantifier operator does not change the class of languages accepted by regular expressions. That is: show how to transform any regular expression containing a numerical quantifier operator into an equivalent regular expression without a numerical quantifier operator.

5. *10 points.* Let L_1 and L_2 be regular languages. Describe an algorithm to decide if $L_1 \cap L_2 \neq \emptyset$. That is, is there any string w such that $w \in L_!$ and $w \in L_2$?

6. *15 points.* Convert the *nondeterministic* finite automaton shown below to a *deterministic* finite automaton, using the "powerset" construction presented in class (and the textbook). Label the states of your constructed DFA with the sets of states of the original NFA.



   (**Note:** Do *not* use the automated DFA conversion tool in JFLAP. You *may* use the JFLAP editor to draw and test your solution.)

7. *15 points.* Let $L = \{0^a 1^b 0^c : a = b + c\}$. Prove that L is not regular.