Test I: CS 3240
Total Points: 100
Time: 1:35 pm to 2:55 pm

Rules: 1. only fully correct and complete answers will get full credit; partial credit will be given to partially correct or partially complete answers.

2. Use name as well as GTID for identification on the answer books.
3. This is a closed book and closed notes exam. Use of internet not permitted
4. Abide by Georgia Tech honor code.
5. No collaborations are allowed. Late answer books not accepted.
6. All sub-questions carry equal points unless otherwise noted.
7. Total of 4 questions each with sub-questions

Question 1: DFAs (Choose any three questions out of five: 30 points)

Devise DFAs for:

1. All strings that start with 1 must end with a 0 and those which start with 0 must end with 1 (alphabet of this language is {0,1}), no null string
2. All strings from the alphabet {a, b} which contain an odd number of a's and even (but non-zero) number of b's
3. All strings that must have 0110 as the substring (alphabet {0,1})
4. All strings which have a length greater than or equal to 3 and ending on b or two consecutive a's
5. Strings that do not contain 3 consecutive a's

Question 2: Regular expressions (Choose any three questions out of five: 30 points)

Write regular expressions for:

1. Expressions that enumerate all positive integers (including 0) upto 100000 but without any leading zeroes
2. Strings made from {a, b} that start and end on the same letter (ie, strings starting with a end on a and those starting with b end on b)
3. Floats using decimal point representation with integer and fractional parts – no leading or trailing zeros and precision upto 4 places after decimal
4. Identifiers that start with a digit or lowercase letter following which one can optionally have one or more of digits or letters or underscores. Identifiers can not end on an underscore (consecutive underscores ok though)
5. Positive integers no leading zeros in which all 2's should occur only after 3's and all 1's should occur only after 2's (ie, no 2 should occur before a 3 or no 1 should occur before a 2).

<u>Question 3</u>: Regular Expression → NFA → DFA (30 points)

Convert the following regular expression into a NFA and convert the NFA to DFA showing the key steps (such as computing ε-closures of sets of states etc.) : $b[ab]^*$ Show all possible NFA transitions (using parallel tree) for the string babba and verify the state transitions in corresponding DFA

<u>Question 4:</u> State True or False (10 points)

a. Consider a language S=(a|b)*. Consider a Regular Language L, whose alphabet is from the set ∑= {a, b}. Let M be a DFA that Recognizes L. Let M' be a DFA obtained from M by changing all accepting states of the M into non-accepting states, and by changing all non-accepting states of M to accepting states. M' recognizes the complement of language L given by S – L
b. For every NFA and its equivalent DFA, the number of states in equivalent DFA must be at least equal to the number of states in the NFA.
c. Consider languages L and L' such that L ⊆ L'. Let M be a DFA that recognizes L and M' be DFA that recognizes L' then the number of states in M' must be equal to or greater than those in M.
d. Consider languages L and L' such that L ⊆ L'. Let M be a DFA that recognizes L and M' be DFA that recognizes L' then the number of states in M' must be lesser than or equal to those in M.
e. For every regular expression there can exist more than one DFA that recognizes the language described by the regular expression.

.