

CS 8803-008: Compilers: Theory and Practice

Homework I

Due: Sept 10th 2019, 7 am EST (submit via Canvas)

Total Points : 100

Guidelines:

1. Georgia Tech Honor Code will be enforced.
2. Answers should be concise, complete and precise

Question I: Rewrite the grammar on lecture1 slides to incorporate, multiple declarative statements followed by multiple assignment statements – modified grammar should be able to handle any length legal programs with respect to the above condition. Currently, the grammar handles only one declarative and one assignment statement. **(15 points)**

Question II: Give state diagrams of DFAs for the following languages (you can directly give DFA or, you can first do a NFA and convert it to DFA) (18 points)

1. $\{w \mid w \text{ starts with 1 and has an even length EXCLUSIVE OR } w \text{ starts with 0 and has a length which is multiple of 3}\}$, alphabet = $\{0, 1\}$
2. $\{w \mid w \text{ contains an odd number of 0's INCLUSIVE OR contains exactly two 1's and ends on a 0}\}$, alphabet = $\{0, 1\}$
3. $\{w \mid w \text{ is any string not in } a^*b^* \text{ that can be derived from } \{a, b\}\}$

Question III: Write regular expressions for the following (18 points)

1. All the strings of lengths three or higher that start with the letter 'b' and end with a letter 'd' or 'm' – alphabet is [a-z]
2. Unsigned integers in which digits occur in a non-decreasing order, can start with a 0. Alphabet = [0-9] e.g. 1129, 18999, 2456, 1 are legal, 1341, 0923 are illegal
3. All strings made from $\{0, 1\}$ which do not contain 00 as a substring

Question IV: Using **primitive NFAs** (start with NFA for each character in the regular expression), first assemble a full NFA for the regular expression: $(ab)^+ (a \mid b)^*$ Then use Brzozowski's algorithm to then convert the same to a minimized DFA. Show all the steps **(25 pts)**

Question V For the following languages, please let us know if they are regular or not and if they are, please implement them using regular expressions. If they are not regular, informally say why (point out what machine needs to do to recognize the expression and why is it incapable). (24 points)

- $L(k) = \{ a^*k \mid *k \text{ represents strings made out of a's such that the number of a's in them can range from 0 to k, k is a fixed positive integer for a given language } L(k), \text{ alphabet} = \{a, b\} \}$
- $L = \{ a \Omega b \mid \Omega \text{ represents strings made out of 3 or less a's followed by 4 or less b's, a string may contain 0 a's and/or 0 b's} \}$
- $L = \{ s \mid \text{strings made out of a's and b's which contain a perfect multiple of 6 in terms of number of a's, strings are non-empty} \}$
- $L = \{ s \mid \text{strings made out of a's whose length is a perfect square, strings are non-empty} \}$