CSE396, Spring 2019 Problem Set 1 Due Fri. 2/15, 11:59pm

Reading:

This coming week's lectures will cover *NFAs* and *regular expressions* in tandem. The previous reading instructions are much the same except to go up just a few more pages to page 69—stopping before you read the sentence with "generalized nondeterministic finite automaton" (GNFA). Also review the parts about closure under regular operations in section 1.2, connecting it to section 1.3.

This is a shorter set, due Friday midnight. Later sets will have a third and possibly fourth written problem and be due Thursday night for the written (PDF) part. There is a technical reason why the *TopHat* portion has a Friday morning deadline, but the intent is that it should be done *first* as it gives shorter takes on key concepts. Most of the questions will be "multi-choice"—that is, multiple choice where more than one of options (a)–(e) can be correct. *TopHat* doesn't give partial credits on those (yet?), so the strict policy is that you have to get the subset of true ones exactly. Thus it is also a kind of grouped true/false question.

Problem (2) will be submitted using the *CSE Autograder* system. This now set up with submission instructions at

`https://autograder.cse.buffalo.edu/courses/CSE396-s19/assessments/hw1`

Scans of handwritten sheets are fine provided they are *easily legible* and *do not have excessive file-size* (at most 5MB). They should have your name, student-ID, and recitation time at the top as if it were a hardcopy submission.

Homework—part online and all *individual work*—due Fri. 2/15, 11:59pm:

(1) Using *TopHat*, the "Worksheet" titled **Spr'19 HW1 Part (1)**. There will be 10 questions, each worth 2 points, for 20 total.

(2) Design a deterministic finite automaton M with alphabet $\Sigma = \{a, b\}$ that recognizes the language of strings that have at most one occurrence of the string aa in them. A technical point is that the string aaa is counted as having two such occurrences even though they overlap. For full credit, you must exhibit a *strategy* as well has giving a correct arc-node diagram (it is not necessary also to give a transition table). The strategy can be expressed via logical comments on the states and instructions—just like with code in Java or any other programming language—or and convert or combine simpler machines using theorems shown in class. (18+6 = 24 pts., for 44 total on the problem set)