CS 331 Compilers **Fall 2017**

Written Assignment 1

Prof. Szajda *Due Tuesday, September 26, 5:00 pm*

This assignment asks you to prepare written answers to questions on regular languages, finite automata, and lexical analysis. Each of the questions has a short answer. As usual, you may discuss the assignment with your colleagues in our class and work on the problems together. However, the empty-hands policy applies. I originally said that you were required to type all written assignments. For this one assignment I am suspending that, because you are required to draw some graphs (e.g., DFAs). Thus you may turn in hand written solutions. Your solution must be turned in, either to me at lecture (or lab), or by placing your work in the mailbox on my office door, by 5:00 pm on Tuesday, September 26. If you happen to decide you want to format your solutions electronically (non-trivial for this assignment), you must **print a hard copy** and turn that in by the deadline.

Note that unlike the programming projects, written assignments are not completed in teams (i.e., each of you must submit a solution to each written assignment).

1. Write regular expressions for the following languages over the alphabet $\Sigma = \{a, b\}$:

   (a) All strings that do not end with aa.

   (b) All strings that contain an even number of b's.

   (c) All strings which do not contain the substring ba.

2. Draw DFAs for each of the languages from question 1. None of your DFAs may contain more than 4 states.

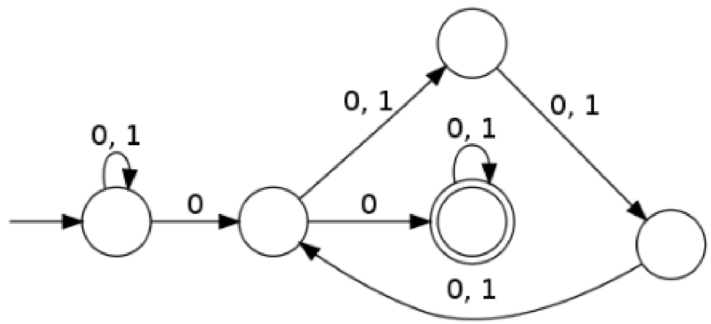
3. Consider the string

```
abaabaababababbaaabaab
```

and its tokenization

```
abaa ba ababa bba aa ba a b
```

Give a Jlex specification with the minimum number of rules that produces this tokenization. Each Jlex rule should be as simple as possible. You may not use regular expression union (i.e., $R1 + R2$) in your solution. Do not give any actions; just assume that the rule returns the string that it matches.



4. Consider the non-deterministic finite automaton (NFA) over the alphabet $\Sigma = \{0, 1\}$a pictured above. Give a one-sentence description of the language recognized by the NFA. Write a regular expression for this language.

5. Let $\Sigma_m = \{a_1, \ldots, a_m\}$ be an alphabet containing m elements, for some integer $m \geq 1$. Let L_m be the following language:

All strings in which at least one a_i occurs an even number of times (not necessarily consecutively), where $1 \leq i \leq m$.

The figure below shows an NFA for the language L_2. Construct a DFA for the language L_2 that has at most 6 states. Also construct an NFA for the language L_3 that has at most 7 states.

Aside: Non-deterministic finite automata (NFAs) are no more powerful than DFAs in terms of the languages that they can describe. However, NFAs can be exponentially more succinct than DFAs, as this problem demonstrates. For the language L_m, there exists an NFA of size at most $2m + 1$ while any DFA must have size at least 2^m. Note that the DFA for the language L_3 is not as easy to construct as the NFA for the language L_3.