CS164 Programming Language and Compilers **Spring 2016**

Written Assignment 1

Assigned: February 1 **Due:** February 8 at 10:30am

Instructions: This assignment asks you to prepare written answers to questions on regular languages and finite automata. Each of the questions has a short answer. You may discuss this assignment with other students and work on the problems together. However, your write-up should be your own individual work.

Remember that written assignments are to be turned in at the start of lecture. Please write your name, your account name, your TA's name, and your section time on your homework. We need this information so that we can give you credit for the assignment and so that we can return it to you.

1. Consider the following languages of binary numbers over the alphabet $\Sigma = \{0, 1\}$.

   - L_1: *All binary numbers that contain at least one 1* (e.g. $0010 \in L_1$)
   - L_2: *All even binary numbers* (e.g. $\{1000, 10, 1010\} \in L_2$)
   - L_3: *All binary numbers divisible by 3*
   - L_4: *All binary numbers that contain at most one* 0 *or no* 1*'s*

   Give a deterministic finite automaton (DFA) for all the languages above.

2. Consider the regular expression $R = (ab + ba)^* + bb$, note that language $L(R)$ is over the alphabet $\Sigma = \{a, b\}$

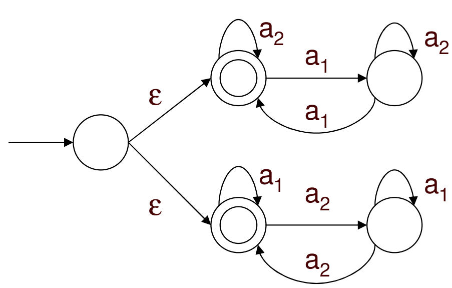
   - Construct an ϵ-NFA for the language $L(R)$.
   - Convert the above NFA to DFA.

   (**Hint:** use approach describe in the lecture : ϵ-NFA $\rightarrow$ NFA (label states) $\rightarrow$ DFA).

3. 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 that includes all strings in which at least one of the characters occurs an even number of times, i.e.

   All strings in which a_i occurs an even number of times for some i, where $1 \leq i \leq m$

   The following figure shows an NFA for the language L_2.



Construct a DFA for the language L_2. Also construct an NFA for the language L_3.

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.

4. (a) Determine whether or not the following languages are regular. Explain why in one or two sentences.
   - L_1: *All strings over the alphabet* $\{a, b, c\}$ *such that* $\#(c) \leq \#(b) \leq \#(a)$ *where* $\#(x)$ *represents number of* x *in the string.*
   - L_2: *All strings over the alphabet* $\{a, b\}$ *such that every* a *is always followed by two consecutive* b*'s.*
   - L_3: *All words in the Oxford English dictionary.* (**Hint:** assume dictionary has finite number of words).

   (b) The Cool expression language as described on page 16 of the Cool reference manual is not regular (The alphabet here is the set of all tokens, and the language is the set of all valid Cool programs). Give one reason why.