Written Assignment 1
Due February 2, 2006

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 either at the start of lecture or in the CS164 homework box in 283 Soda by 12:30 PM on the due date.

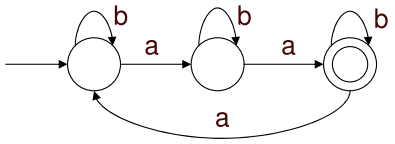
1. Consider the following languages over the alphabet $\Sigma = \{a, b\}$.

   - L_1: *All strings that contain at least two a's*
   - L_2: *All strings that contain at least one b*
   - L_3: *All strings that contain at least two a's and at least one b*
   - L_4: *All strings that contain at most one a or no b's*

   Give a deterministic finite automaton (DFA) for the languages L_1, L_2, L_3 and L_4.

   Aside: This example illustrates that regular languages are closed under intersection and complementation. Note that $L_3 = L_1 \cap L_2$ and $L_4 = \Sigma^* - L_3$, where Σ^* represents the language containing all strings over the alphabet Σ.

2. Consider the following DFA over the alphabet $\Sigma = \{a, b\}$.

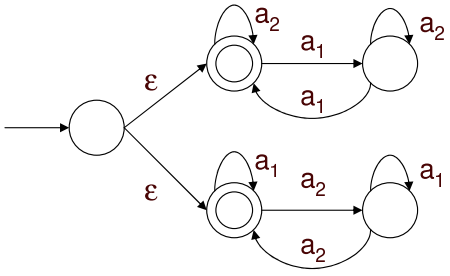


   Give a one-sentence description of the language recognized by the DFA. Write a regular expression for the same language.

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* $\{0, 1\}$ *that have equal number of 1's and 0's.*
   - L_2: *All strings over the alphabet* $\{0, 1\}$ *that are palindromes* [1].
   - L_3: *All words in the Oxford English dictionary.*

   (b) The Cool 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.

   Aside: This illustrates that we cannot use a lexer to *parse* the Cool language.

[1] strings that read the same when read left to right or right to left