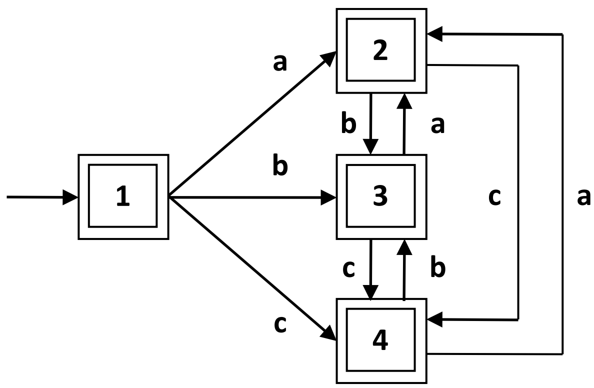
**CS 475/575 Exam 1** **Spring 2012** **Solution**

**Your score on this exam is based on your best 5 of these 6 problems.**

1. Let language L be the strings over alphabet {a,b,c} that do not contain two consecutive identical characters. That is, the following substrings are forbidden: aa, bb, cc. Draw a deterministic finite-state machine that accepts language L.




2. Let language L be as defined in problem 1, and let n be an arbitrary positive integer. Let $X_n$ denote the number of strings in L that have length exactly n. Write a formula for $X_n$, and prove by mathematical induction that your formula is correct.

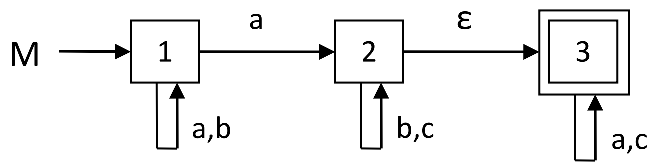
   **$X_n = 3*2^{n-1}$.**

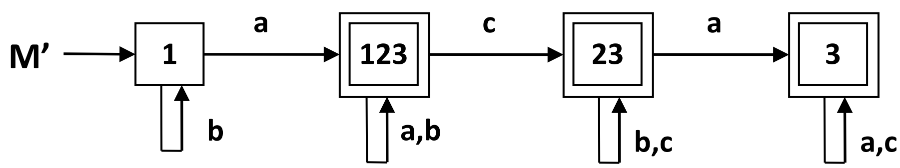
   **Basis: Three strings in L have length 1: {a, b, c}. So $X_1 = 3 = 3*2^{1-1}$.**

   **Inductive hypothesis: Assume that $n \geq 2$ and $X_{n-1} = 3*2^{(n-1)-1} = 3*2^{n-2}$.**

   **Inductive step: Let $S = d_1d_2\ldots d_{n-1}$ be any of the $X_{n-1}$ strings in L of length n–1, where each symbol $d_j \in \{a, b, c\}$. Notice that S can be extended to a string in L of length n by appending one additional character $d_n$, provided that $d_n \neq d_{n-1}$. So in each case there are exactly two possible characters to choose for $d_n$. Hence $X_n = X_{n-1}*2 = (3*2^{n-2})*2^1 = 3*2^{(n-2)+1} = 3*2^{n-1}$.**

3. Consider the non-deterministic finite-state machine M shown below. Draw a deterministic finite-state machine M’ that is equivalent to M, and label each state of M’ with its corresponding set of states from M.




4. Let machines M and M’ be same as in problem 3. For each of M and M’, write all the accepting computation sequences for the input string abaca. [If you omitted problem 3, note that you can still obtain a computation sequence for M’ by determining all possible states for M after M processes each prefix of the input string.]

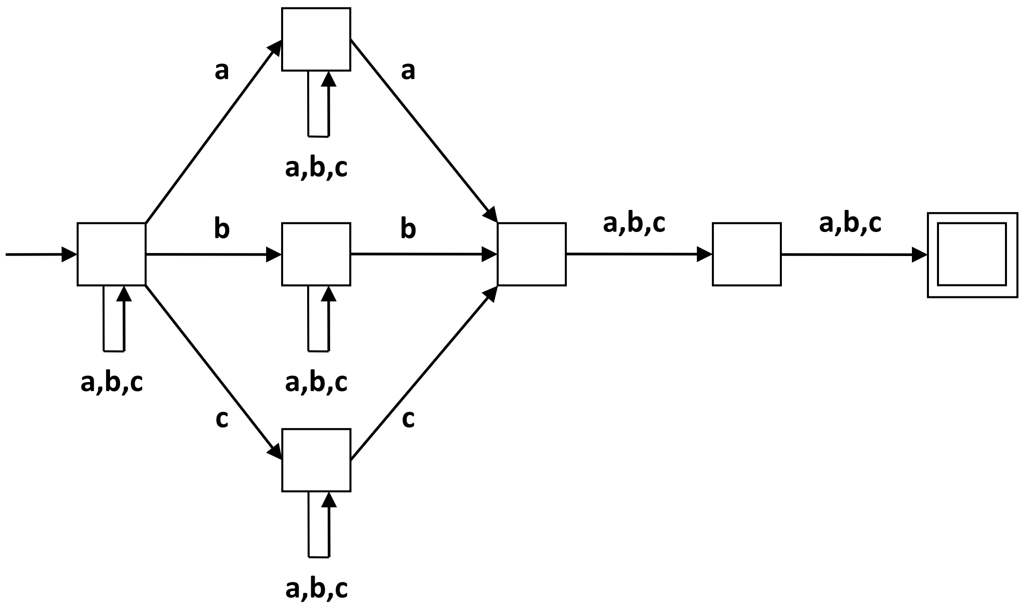
**M:** $1 \xrightarrow{a} 1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{c} 2 \xrightarrow{\varepsilon} 3 \xrightarrow{a} 3.$

**M:** $1 \xrightarrow{a} 1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{\varepsilon} 3 \xrightarrow{c} 3 \xrightarrow{a} 3.$

**M:** $1 \xrightarrow{a} 2 \xrightarrow{b} 2 \xrightarrow{\varepsilon} 3 \xrightarrow{a} 3 \xrightarrow{c} 3 \xrightarrow{a} 3.$

**M’:** $1 \xrightarrow{a} 123 \xrightarrow{b} 123 \xrightarrow{a} 123 \xrightarrow{c} 23 \xrightarrow{a} 3.$

5. Let language L be the strings over alphabet {a,b,c} such that the character in the third-rightmost position also appears somewhere earlier in the string. For example, in the string S = aca<u>b</u>ca<u>b</u>ac, the character b appears both in the third-rightmost position and also earlier in S. Draw a <u>non-deterministic</u> finite-state machine that accepts language L.




6. Draw a deterministic finite-state machine that models a memory that holds one value. The initial value is 0. The input alphabet {i,d,s} corresponds to these unary operations: increment, double, square. Each operation is performed using mod 7 arithmetic. The machine should accept a string iff the corresponding sequence of operations yields a prime number (2, 3, 5). For example, the string "idsid" yields the value 3, so it is accepted. But the string "disidis" yields the value 4, so it is rejected.