6.045J/18.400J: Automata, Computability and Complexity Prof. Nancy Lynch

Homework 2

Due: 16 February 2005 *Vinod Vaikuntanathan*

Problem 1: **Designing DFAs and NFAs.** For each of the following, draw a state diagram for a DFA or NFA (as required) that recognizes the specified language. In all cases the alphabet is $\{0, 1\}$.

(a) (Sipser, Exercise 1.6, part a)
$L_a = \{w | w \text{ begins with 1 and ends with 0}\}$. Provide a DFA recognizing L_a.

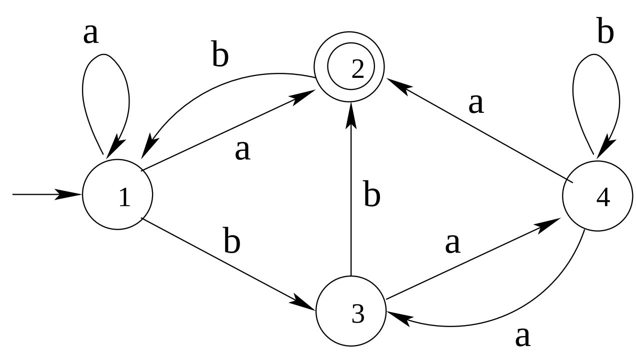
(b) (Sipser, Exercise 1.6, part l)
$L_b = \{w | w \text{ contains an even number of } 0s, \text{ or contains exactly two } 1s\}$. Provide a DFA recognizing L_b.

(c) (Sipser, Exercise 1.7, part e)
$L_c = 0^*1^*0^+$, that is, the set of strings consisting of some number (possibly zero) of 0s followed by some number of 1s followed by at least one 0. Provide an NFA recognizing L_c, with exactly three states.

(d) $L_d = \{w | w \text{ contains the substring 1100 or does not contain the substring 1010}\}$. Provide an NFA recognizing L_d.

Problem 2: **Proving an FA recognizes a language.** For one of the automata you designed in problem 1, prove that the machine recognizes *exactly* the specified language. To do this, you will need to prove that your automaton (1) accepts all strings in the language and (2) does not accept any string not in the language.

Problem 3: **NFA to DFA.** Consider the following state diagram.



(a) The state diagram above represents an NFA $N = (Q, \Sigma, \delta, q_0, F)$. Say what each of the components of the 5-tuple is, for this NFA.

(b) Apply the subset construction described in class to obtain a DFA $M = (Q', \Sigma, \delta', q_0', F')$ that is equivalent to N. State the corresponding element(s) in Q', δ', q_0', F', then describe δ' via either a transition table or a state diagram.

Problem 4: **Showing languages are regular.** (From Sipser, Problems 1.31 and 1.32)
For any string $w = w_1w_2 \ldots w_n$, the reverse of w, written $w^{\mathcal{R}}$, is the string w in reverse order, $w_n \ldots w_2w_1$. For any language A, let $A^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in A\}$.

1. Show that if A is a regular language, so is $A^{\mathcal{R}}$.

2. Let

$$\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

Σ_3 contains all size 3 columns of 0s and 1s. A string of symbols of Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let

$$B = \{w \in \Sigma_3^* \mid \text{ the bottom row of } w \text{ is the sum of the top two rows}\}.$$

Show that B is regular. (Hint: Use part 1).

Problem 5:

(a) **Closure construction.** Use the closure construction for concatenation described in class, and in Theorem 1.47, to show that L_1L_2 is regular, where L_1 is the set of strings containing 0 in every position that is a multiple of 3, and L_2 is the set of strings of length at least 3. (Construct the machines for L_1 and L_2 and illustrate the closure construction of the class)

(b) **Regular languages are closed under another operation** (Sipser, Exercise 1.43) Let A be any language. Define $DROPOUT(A)$ to be the language consisting of all strings that can be obtained by removing one symbol from a string in A. Thus, $DROPOUT(A) = \{xz | xyz \in A \text{ where } x, z \in \Sigma^*, y \in \Sigma\}$. Show that the class of regular languages is closed under the $DROPOUT$ operation. Give both a proof by picture and a more formal proof by construction as in Theorem 1.47.