Theory of Computation: Assignment 2

Arjun Chandrasekhar

Due Thursday, 02/03/2022 at 11:59 pm (50 points)

1. (10 points) This problem is taken from Sipser 1.4e. Consider the following language on the alphabet $\Sigma = \{a, b\}$:

$$L = \{w | w \text{ starts with an a and has at most one b}\}$$

This is the intersection of two smaller languages. Construct the DFAs for the two simpler languages. Then combine them into a DFA using the construction we described in class to give the DFA for L.

2. (10 points) This problem is taken from Sipser 1.5c. Consider the following language on the alphabet $\Sigma = \{a, b\}$:

$$L = \{w | w \text{ contains neither the substrings ab or ba}\}$$

This language is the complement of a simpler language. Construct the DFA for the simpler language. Then, use the construction we described in class to give the DFA for L.

3. (10 points) This problem is taken from Sipser 1.32. 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 0's and 1's. A string of symbols gives a row of 0's and 1's. Each row then makes a binary number.

Next, let

$$B = \{w \in \Sigma_3^* | \text{the top row + middle row = bottom row}\}$$

For example

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \in B$$

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \notin B$$

Show that B is regular. Note that in class we did a similar problem using base-10 numbers; you should follow the same approach. Once again, it will be more convenient to work with the reversed versions of the strings; therefore you may use (without proof) the fact that if B^r is regular, then B is regular.

4. (10 points) The **exclusive OR (XOR)** operation on languages is defined as follows:

$$A \oplus B = \{w \in A \text{ or } w \in B \text{ \underline{but not both}}\}$$

Prove that regular languages are closed under XOR.

(Side note: sometimes the OR that you are familiar with from computer programming is called the *inclusive OR*.)

5. (10 points) The EVERY-OTHER operation on languages is defined as follows:

$$\text{EVERY-OTHER}(A) = \{w | w = a_1 y_1 a_2 y_2 \ldots a_n y_n \text{ where } a_1 a_2 \ldots a_n \in A \text{ and the } y_i's \text{ can be anything}\}$$

Essentially, all of the odd characters (i.e., every other character) form a string in A, and the even characters don't matter. Note that the string must have an even length.

As an example, suppose $A = \{w | w \text{ only contains 0s}\}$. Then 01010101 and 00000101 are in EVERY-OTHER(A) because the odd characters are all 0's and the even characters can be either 0 or 1.

Prove that regular languages are closed under EVERY-OTHER