CS 334 Fall 2020: Problem Set 2.

Problem 1. (15 points)

a. Construct an FSA with 6 states to recognize $L_4 = \{1111\}$. Can you reduce the number of states below 6? (Hint: recall a basic property of directed graphs from CS 135!)
b. Use your argument to prove that, for all $k \geq 3$ there is a language that can be accepted by a k-state FSA that cannot be recognized by any FSA with fewer states.

Problem 2. (15 points) For any string $w = w_1 w_2 \cdots w_n$, the *reverse* of w, written w^R, is the string w in reverse order, $w_n \cdots w_2 w_1$. For any language A, let $A^R = \{w^R | w \in A\}$.
Show that if A is regular, so is A^R.

Problem 3. (20 points) In this problem you will design an FSA that checks if the sum of two numbers equals a third number. Each number is an arbitrarily long string of bits. At each step, the input to the FSA is a symbol that encodes 3 bits, one from each number. In other words, the alphabet of the FSA is:

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

In the input string $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix}$ the bits in the top row represent the number 01, the bits in the middle row represent 00 and the third row represents 11. In this case, since 01+00 $\neq$ 11, the FSA must reject the input. On the other hand, the input string $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix}$ will be accepted by the FSA since $001 + 011 = 100$.

Formally, let $B = \{w \in \Sigma^*: the\ bottom\ row\ of\ w\ equals\ the\ sum\ of\ the\ top\ two\ rows\}$, where Σ^* represents all finite strings over the alphabet Σ. Your goal is to design an FSA for the language B.

To get started, first design a 2-state FSA that recognizes B^R – this should be straightforward because the input arrives least significant bits first and most significant bits at the end. Next, use the technique of Problem 2 to design the final FSA for B.

Optional Problem. (20 points) For any string σ, over alphabet Σ, we define the string $SHIFT(\sigma)$ as follows: if $\sigma = aw, a \in \Sigma, w \in \Sigma^*$ then $SHIFT(\sigma) = wa$. For example, $SHIFT(0111) = 1110,\ and\ SHIFT(10110) = 01101.$
Prove that if L is regular, then so is $SHIFT(L) = \{SHIFT(\sigma): \sigma \in L\}$.