CS/ECE 374 A (Spring 2020)
Homework 3 (due Feb 13 Thursday at 10am)

Instructions: As in previous homeworks.

Problem 3.1: For each of the following languages in parts (a), (b), and (c), describe an NFA that accepts the language, using as few states as you can. Provide a short explanation of your solution.

(a) all strings $x \in \{0,1\}^*$ such that the length $|x|$ is divisible by 7 **or** x does **not** contain 00000 as a substring.

(b) the language defined by the regular expression $((101 + 11)^*(20)^*2 + 122)^*(1 + \varepsilon)$ (over the alphabet $\{0,1,2\}$).

(c) all strings $x \in \{0,1\}^*$ of length at least 4 such that the total number of 1's among the first two symbols and the last two symbols in x is 2.

(For example: 01110110 and 001011 are in the language, but 111001 is not, since 11 and 01 have a total of 3 1's.)

(d) Convert your NFA from part (c) to a DFA by using the subset construction (i.e., power set construction). Don't include unreachable states.

Problem 3.2: Given two languages L_1 and L_2 over the alphabet $\{0,1\}$, define

$$\text{INSERT}(L_1, L_2) = \big\{v_1u_1v_2u_2\cdots u_kv_{k+1} : \ u_1,\ldots,u_k \in L_1,\ v_1,v_2,\ldots,v_{k+1} \in \{0,1\}^*,\ k \geq 0, \text{ such that } v_1v_2\cdots v_{k+1} \in L_2\big\}.$$

Informally, a string is in $\text{INSERT}(L_1, L_2)$ iff it can be obtained by taking a string v in L_2 and inserting (possibly multiple) strings from L_1 at various positions in v. (For example, if $L_1 = \{0\}$ and $L_2 = \{110\}$, then $0100010 \in \text{INSERT}(L_1, L_2)$.)

Prove that if L_1 and L_2 are regular, then $\text{INSERT}(L_1, L_2)$ is regular.

[*Hint*: given regular expressions for L_1 and L_2, describe a recursive algorithm to produce a regular expression for $\text{INSERT}(L_1, L_2)$. Provide justification for your regular expression constructions. A formal proof of correctness is not required.]

Problem 3.3: For a string $x \in \{0,1\}^*$, let x^F denote the string obtained by changing all 0's to 1's and all 1's to 0's in x.

Given a language L over the alphabet $\{0,1\}$, define

$$\text{FLIP-SUBSTR}(L) = \{uv^Fw : \ uvw \in L,\ u,v,w \in \{0,1\}^*\}.$$

Prove that if L is regular, then $\text{FLIP-SUBSTR}(L)$ is regular.

(For example, $(1011)^F = 0100$. If $1011011 \in L$, then $1000111 = 10(110)^F11 \in \text{FLIP-SUBSTR}(L)$. For another example, $\text{FLIP-SUBSTR}(0^*1^*) = 0^*1^*0^*1^*$.)

[*Hint*: given an NFA (or DFA) for L, construct an NFA for FLIP-SUBSTR(L). Give a formal description of your construction. Provide an explanation of how your NFA works, including the meaning of each state. A formal proof of correctness is not required.]

Bonus ($\frac{3}{10}$ points[1]): Consider the modified language

$$\text{FLIP-SUBSTR}(L) \;=\; \{uv^Rw : \; uvw \in L, \; u, v, w \in \{0,1\}^*\},$$

were v^R denotes the reverse of v. Prove that if L is regular, then FLIP-SUBSTR(L) is regular.

[1] For bonus questions, no extra points (and no IDK!) unless your solution is very close to correct.