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. Below, \( #_0(x) \) and \( #_1(x) \) denote the number of 0’s and the number of 1’s in \( x \) respectively.

(a) (30 pts) all strings \( x \in \{0, 1\}^* \) such that \( x \) ends with 10101 or 11011 and \( #_0(x) \) is divisible by 3 or \( #_1(x) \) is divisible by 3.

(b) (30 pts) the language defined by the regular expression \(((01)^*0+2)(100)^*1)^* \cdot (1^* + 0^*2^*)\) over the alphabet \{0, 1, 2\}.

(c) (10 pts) all strings in \{0, 1\}^* that contains the pattern 0?1?0, where “?” denotes “don’t care” (i.e., a single symbol that is either 0 or 1); in other words, the language defined by the regular expression \((0 + 1)^* \cdot 0(0 + 1)1(0 + 1)0 \cdot (0 + 1)^*\).

(d) (30 pts) Convert your NFA from part (c) to a DFA by using the subset construction (i.e., power set construction). [Note: don’t include unreachable states; also, several accepting states can be collapsed into one in this DFA.]

Problem 3.2: Given a language \( L \) over the alphabet \( \Sigma \), define

\[
\text{MOVE-BACK}_8(L) = \{ xayz : xyaz \in L, x, y, z \in \Sigma^*, a \in \Sigma, |y| \leq 8 \}.
\]

Prove that if \( L \) is regular, then \( \text{MOVE-BACK}_8(L) \) is regular.

(For example, if 01001010011 \( \in L \), then 01100010100011 \( \in \text{MOVE-BACK}_8(L) \).) \[1\]

[Hint: given an NFA (or DFA) for \( L \), construct an NFA for \( \text{MOVE-BACK}_8(L) \). 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 of your NFA is not required.]

\[1\] and also 010101010010011 \( \in \text{MOVE-BACK}_8(L) \), and 010011010010011 \( \in \text{MOVE-BACK}_8(L) \), \ldots, 010010100110011 \( \in \text{MOVE-BACK}_8(L) \). 

For a different example: \( \text{MOVE-BACK}_8(0^*1^*) = 0^*1^* + 0^*101^* + 0^*1001^* + 0^*10^31^* + \cdots + 0^*10^61^* \).