ASSIGNMENT 7 (DUE ON 9 OCTOBER 2021 AT 11:59PM)

MATH2301, SEMESTER 2, 2021

INSTRUCTOR: ASILATA BAPAT

(1) Draw the state diagrams of DFAs recognising the following languages. The alphabet is $\{0,1\}$. Justifications are not required.
  (a) $L = \{0, 1, 11\}$

  (b) $L = \{w \in \Sigma^* \mid w \text{ starts with a } 1\}$

  (c) $L = \{w \in \Sigma^* \mid w \text{ any 1s in } w \text{ are next to each other in a single block}\}$

  (d) $L = \{w \in \Sigma^* \mid w \text{ contains an even number of zeroes}\}$

  (e) $L = \{w \in \Sigma^* \mid w \text{ contains the substring } 0101\}$

(2) Let $L \subseteq \Sigma^*$ be a language. The *complement of L*, denoted L^c, is the complement of L in Σ^*. That is, for every $w \in \Sigma^*$, we have $w \in L^c$ if and only if $w \notin L$.
  (a) Given a DFA M recognising a language $L = L(M)$, explain in words how to construct a DFA M' such that $L(M') = L^c$.

  (b) Construct a DFA recognising the following language:
$$L = \{w \in \Sigma^* \mid \text{every odd position of } w \text{ is } 1\}.$$

  (c) Now use your method from the first part to draw a DFA for the complement of the language L above. Justification not required.

  (d) Give a direct explanation in English (similar to the explanation given in the second part for L) of L^c.

(3) Let M_1 and M_2 be DFAs whose languages are $L_1 = L(M_1)$ and $L_2 = L(M_2)$.
  (a) Explain in words how to modify the product construction shown in class to construct a new DFA M such that $L(M) = L(M_1) \cap L(M_2)$.

  (b) The following language is the intersection of two simpler languages.
$$L = \{w \mid w \text{ has an odd number of 0s and ends with a } 1\}.$$

  Draw DFAs for each of the two simpler languages, and then use your modified product construction to draw a DFA recognising this language. Justifications are not required.

(4) Let $\Sigma = \{0,1\}$. Let
$$L = \{w \mid w \text{ contains an equal number of occurrences of the substrings 01 and 10}\}.$$
  (a) Find either a DFA or an NFA whose language is L. Justify your answer.

  (b) (Bonus, not for credit) Can you find a regular expression whose language is L?