Policy on homeworks

1. **Collaboration**: You may discuss a question with any other student currently taking CS81 provided:
   (i) both of you contribute equally; (ii) you come away from any discussion with an understanding in your mind (and no archived solution of any form is retained); (iii) your submission is your own work prepared by yourself on a separate occasion.

2. **Reference materials**: You should only refer to materials from this semester of CS81 (class notes, handouts, textbooks, grutors, instructor, etc).

3. **Submission**: Your submission should be legible or is prepared using TeX.

Regular languages

(A) Provide DFAs for the following languages.

   (i) \( L = \{ w \in \{0,1\}^* : w \text{ contains the same number of } 01 \text{'s and } 10 \text{'s} \}. \)

   (ii) \( L = \{ w \in \{0,1\}^* : w \text{ is the binary representation of a number that is divisible by } 3 \}. \)

   (iii) \( L = \{ w \in (\{0,1\} \times \{0,1\} \times \{0,1\})^* : w \text{ encodes binary addition} \}. \) For example,

   \[
   \begin{bmatrix}
   0 \\
   1 \\
   1 \\
   0 \\
   1 \\
   0 \\
   0 \\
   1 \\
   0
   \end{bmatrix} \in L, \quad \text{but} \quad \begin{bmatrix}
   0 \\
   1 \\
   1 \\
   1 \\
   1 \\
   0
   \end{bmatrix} \notin L.
   \]

(B) Consider the regular expression \( R = (0 \cup 1(01^*0)^*1)^* \).

   (i) Find the smallest DFA you can construct for \( L = L(R) \).

   (ii) Describe succinctly what \( L \) is: \( L = \{ w \in \{0,1\}^* : w \text{ is ...} \}. \)

(C) Consider the following assertions:

   (1) If \( L \) is regular, then so is \( L^c \) (the complement of \( L \)).

   *Proof. (sketch)* If \( L \) is regular, then there is an NFA \( M \) that accepts \( L \). Switch each state of \( M \) from accepting to non-accepting and vice versa. Call the new machine be \( M' \). Then \( M' \) accepts of \( L^c \). □

   Is the proof sketch above correct? If yes, provide the missing details (for example, why does \( M' \) accept \( L^c \)). If not, explain what is wrong and provide the correct proof.

   (2) If \( L_1 \) and \( L_2 \) are regular, then so is \( L_1 \cap L_2 \).

   *Proof. (sketch)* Note \( L_1 \cap L_2 = (L_1^c \cup L_2^c)^c \). Since regular languages are closed under union and complementation (see previous part), \( L_1 \cap L_2 \) is regular. □

   i. Is this proof sketch correct? If not, explain why and provide a correct proof. If yes, provide any missing details if any.
ii. Describe a direct construction of a DFA for \( L_1 \cap L_2 \) from the DFAs for \( L_1 \) and \( L_2 \).

(3) If \( L \) is regular, then so is \( \text{Suffix} (L) \), where \( \text{Suffix} (L) = \{ w \in \Sigma^* : xw \in L, \text{ for some } x \in \Sigma^* \} \).

Proof. (sketch) If \( L \) is regular, then let \( M \) be any DFA that accepts \( L \). Add \( \epsilon \)-moves from the start state of \( M \), say \( s \), to each state \( q \) of \( M \). Call the new machine \( M' \). Then \( M' \) accepts \( \text{Suffix} (L) \).

Is the proof sketch above correct? If yes, provide the missing details (for example, why does \( M' \) accept \( \text{Suffix} (L) \)). If not, explain what is wrong and provide the correct proof.