Figure 1: Question 1
Figure 2: Question 2
Figure 3: Question 3
So \((0 \cup 11 \cup 0(01)^*)^*\) is a regular expression describing the language recognized by \(M\).

(4)

(a)

**Using regular expressions:** Let \(R\) be a regular expression describing the regular language \(L'\). Let \(\Sigma\) denote a regular expression for the union of the symbols in the alphabet. Then \(\Sigma^* . R . \Sigma^*\) describes \(\text{Ext}(L')\). Since \(R\) and \(\Sigma\) are regular expressions, \(\Sigma^* . R . \Sigma^*\) is a regular expression. So \(\text{Ext}(L')\) is a regular language.

**Using NFAs:** We have defined \(\text{Ext}(L')\) as –

\[
\text{Ext}(L') = \{ w : w \text{ has a substring } x \text{ such that } x \in L' \}
\]

Suppose \(M' = (Q', \Sigma, \delta', q'_0, F')\) is a DFA that recognizes the regular language \(L'\). We will define an NFA \(N = (Q, \Sigma, \delta, q_0, F)\) that accepts \(\text{Ext}(L')\). This will show that \(\text{Ext}(L')\) is indeed a regular language. Set \(Q = Q' \cup \{ q_0, q_F \}\), where \(q_0, q_F \notin Q'\). Define \(\delta\) as follows –

\[
\delta(q_0, a) = \{ q_0 \} \text{ for } a \in \Sigma,
\]

\[
\delta(q_0, \epsilon) = \{ q'_0 \},
\]

\[
\delta(q, a) = \{ q' \} \text{ for } a \in \Sigma, q \in Q', \text{ and } \delta'(q, a) = q',
\]

\[
\delta(q, \epsilon) = \{ q_F \} \text{ for } q \in F',
\]

\[
\delta(q, \epsilon) = \phi, \text{ for } q \in Q' \setminus F'
\]

Set \(F = \{ q_F \}\).

How would \(N\) accept a string \(w \in \text{Ext}(L')\)? \(w\) will have a substring \(x \in L'\), by the definition of \(\text{Ext}(L')\). Before the first symbol of \(x\) is encountered, \(N\) remains in the start state \(q_0\). \(N\) non-deterministically “guesses” the first symbol of the string \(x\), and using the \(\epsilon\)-transition moves to the state \(q'_0\). After that the computation in \(N\) mimics the computation of \(M'\) on \(x\), and eventually reaches one of the final states of \(M'\). Then \(N\) non-deterministically applies the \(\epsilon\)-transition to move to the final state \(q_F\). On the other hand, if \(w \notin \text{Ext}(L')\), there is no computation of \(N\) that will reach the state \(q_F\), and hence get accepted by \(N\) (you can show this by contradiction).

(b)

This language is the complement of the language defined in the previous question. Since complement of a regular language is again a regular language, this is also a regular language.

(c)

Let \(R_1\) and \(R_2\) be two regular expressions according to our formal definition.

\(+\): \(R_1^+ = R_1 \cup R_1^*\). We have expressed an extended regular expression containing \(+\) in terms of only concatenation and Kleene star.

**Intersection:** \(R_1 \cap R_2 = \overline{R_1} \cup \overline{R_2}\), using De Morgan’s law. We have expressed intersection in terms of union and complementation. We will be done if we can express complementation in terms of union, concatenation and Kleene star.

**Difference:** \(R_1 \setminus R_2 = R_1 \cap \overline{R_2} = \overline{R_1} \cup \overline{R_2}\), using the the definition of difference and De Morgan’s law. Again, we will be done if we can express complementation in terms of union, concatenation and Kleene star.

**Complementation:** Choose a complemented regular expression \(R\) in the extended regular expression so that \(R\) does not itself contain a complement. Since \(R\) is a regular expression, \(L(R)\), the language described by \(R\), is a regular language. Let \(D\) be a DFA that recognizes \(L(R)\). Complementing the final/non-final states
of $D$, we will get a DFA for $\overline{L(R)}$. Convert this DFA into a GNFA, and then to a regular expression $R'$. Replace $\overline{R}$ in the original extended regular expression by $R'$. Repeat this procedure until no complements remain.

\[5\]

Step (f) is not a valid argument. We have only shown that regular languages are closed under finite union. But in step (f) we are assuming that this closure holds for infinite union as well. In fact, we can easily produce a counter example. By using pumping lemma we can show that the language $L = \{0^n1^n : n \geq 0\}$ is not regular. Now consider the infinite class of languages – $L_i = \{0^i1^i\}$, for $i \geq 0$. Clearly, each of the $L_i$’s are finite, hence regular. But $L = \cup_i L_i$. This example shows that regular languages are not necessarily closed under infinite union.