Solution 1:

Correct Answer: C
Explanation: Clearly option C represents "Set of all binary strings which contains 00 as a substring. Now consider a binary string \( w \) belongs to regular expression defined by option (C), say \( R \). Since \( w \in R \) \( w \) contains at least one occurrence of 00. Partition \( w \) into substring \( xyz \), where \( y = 00 \), and \( y \) is the first occurrence of 00 in \( w \).

Now we claim that we will be at state \( q_0 \) after reading \( x \). Because we can see any number of 1 at \( q_0 \) and as we see a 0 we go \( q_1 \). Now at this point \( x \) cannot be finished. Since \( y \) is the first occurrence of 00, \( x \) must end with a 1. So we will come back to \( q_0 \) and so on. As we read \( y \) we reach to \( q_2 \) and here we can read any combination of 0 and 1 because of the loop. Hence given DFA accept all string consist of substring 00.

Solution 2:

Correct Answer: A,D
Explanation: Option A is correct because \((1 + 01)^*\) can generate \(\epsilon\) (as any RE of the form \(a^*\) can generate \(\epsilon\)). Similarly, \(0^*\) can generate \(\epsilon\). The concatenation of these two will result in an \(\epsilon\).

Option B and C is incorrect because the string 00 can only be generated by \(0^*\) in the given Regular expression. This cannot be followed by even a single 1.

Option D is correct because \((1 + 01)\) can generate 1 and 01. Hence \((1 + 01)^*\) can generate 1 1 1 01 01 1 1 1. The last three 0’s can be generated by RE \(0^*\).

Solution 3:

Correct Answer: B
Explanation: The RE \(1^*(01^*0)^*\) will generate a string containing even number of 0’s. The RE \(1^*01^*\) will generate a string containing a single 0. When concatenated, the given RE will generate a string containing odd number of 0s. Hence option B is correct.

Solution 4:

Correct Answer: B,D
Explanation: Option A is incorrect because by definition of a DFA, the set of states \(Q\) must be a finite set.

Option B is correct because, for example, the RE \(0^*\) can generate infinitely many strings, and by equivalence of DFA and RE, we know that there exists a DFA to accept a language containing those strings.

Option C is incorrect because it is possible for an NFA with \(n\) states to have an equivalent DFA with no less than \(2^n\) states.

Option D is correct because any language containing finitely many strings is a regular language and can be accepted by a DFA.

Solution 5:

Correct Answer: A,C
Explanation: Option A is correct. By taking string of length 0, we can generate \(\epsilon\) from \(\phi^*\).

Option B is incorrect because \(\epsilon^* = \epsilon\)

Option C is correct. The union of the two RE, \(\phi\) and \(\epsilon\) will be \(\epsilon\).

Option D is incorrect because \(\phi \cdot \epsilon = \phi\).
Solution 6:

Correct Answer: A

Explanation: We first prove that any language accepted by the DFA can be generated by the RE \((01^*0 + 10^*1)^*(01^* + 10^*)\). Consider any string \(w\) accepted by the DFA. We partition \(W\) into strings \(xy\) such that \(x\) is the largest possible string after which DFA is at state \(q_0\). We further partition \(x\) into substrings \(x_1x_2...x_n\) where each \(x_i\) represent a portion of string between two consecutive \(Q_0\)'s. Clearly, each \(x_i\) can be generated by either \(01^*0\) or \(10^*1\) (depending upon whether the DFA goes through the state \(Q_2\) or \(Q_1\)). Also, the string \(y\) can be generated by either \(01^*\) or \(10^*\) (depending upon whether the acceptance is by state \(Q_2\) or \(Q_1\)).

In a similar way, it can also be proved that any string generated by \((01^*0 + 10^*1)^*(01^* + 10^*)\) can be accepted by the DFA. Hence Option A is correct.

Solution 7:

Correct Answer: D

Explanation: The RE \((R + P)^*\) can generate infinite strings. Hence option D is correct.

Solution 8:

Correct Answer: A,D

Explanation: Counter example for option B can be generated by setting \(R = 1\) and \(S = 0\) as string 10 \(\in\) \((1^* + 0^*)^*\) but 10 \(\notin\) \(1^* + 0^*.\) Same setting works for C also, \(\epsilon\) \(\in\) \((1^*0)^*\) but \(\epsilon\) \(\notin\) \((1 + 0)^*0.\)

Now let's look at the correctness of A. Let \(w \in L((RS + R)^*R)\) then the general form of \(w\) will be \(w = r_1s_1r_2s_2...r_is_i...r_n\) where \(r_i \in L(R)^* - \{\epsilon\}\) and \(s_i \in L(S)\). Since \(r_1 \in L(R)^* - \{\epsilon\}\) and \(s_1r_2s_2...r_is_i...r_n \in (SR + R)^*\) therefore we can say that \(w \in L(R(SR + R)^*).\) Similarly we can prove in other direction.

Finally let's look at option D. Let \(w \in L((R + S)^*)\) then the general form of \(w\) will be \(w = w_1w_2...w_n\) where \(w_1 \in L(R + S)\). Since \(w_1 \in L(R^*S^*)\) therefore \(w \in (R^*S^*)^*\). For other direction, let \(w \in (R^*S^*)^*\) then the general form of \(w\) will be \(w = w_1w_2...w_n\) where \(w_1 \in L(R^*S^*)\), clearly \(w_i \in L((R + S)^*)\) so \(w \in L((R + S)^*)\) and since \((R + S)^*)^* = (R + S)^*\) it follows that \(w \in L((R + S)^*)\).

Solution 9:

Correct Answer: B,C,D

Explanation: Notice that option B,C and D are same. Because both \(q_2^*\) and \(q_4^*\) are the final states and there is no transition going from these states to any other states.

Clearly the language accepted by NFA is, set of all binary string which contain one of 00 or 11 as a substring. Now we will show the same language is accepted by the DFA given in option B. This can be shown by the similar technique, instructor discussed in lecture 3. Following are the states and the properties of the strings ending at those states. (Idea is very much similar to the solution of question 1 in this assignment)

- \(q_3^*\): All string that neither contain 00 nor 11 as a substring and ending with 0.
- \(q_4^*\): All string that neither contain 00 nor 11 as a substring and ending with 1.

Case 1: If we are at \(q_3^*\) and read another 0, we will immediately go to \(q_4^*\) and then we can read any combination of 0 and 1.

Case 2: Similarly if we are at state \(q_1^*\) and read another 1 we immediately go to \(q_2^*\) and then can read any combination of 0 and 1.

Case 1 and 2 are exclusive i.e both cannot be executed for the same string. Hence string will either contain 00 or 11 as a substring.

2
Solution 10:

Correct Answer: B,D

Explanation: To show that option A is incorrect, let \( L_1 = L_2 = \{0^n \mid n \geq 2 \text{ is an integer}\} \) and \( L_3 = \{0^n \mid n \text{ is a prime number}\} \). Since \( L_3 \) is a non-regular language and clearly the equality holds therefore option A is incorrect.

Correctness of the option B follows from the 3 simple closure properties of union, concatenation, star operator described in week 2 lecture 2.

For proving C is incorrect let \( L_1 = \{0^n \mid n \geq 2 \text{ is an integer}\} \cup \{\epsilon\} \) and \( L_3 = \{0^n \mid n \text{ is a prime number}\} \). Clearly if \( w \in L_3 \) then \( w \in L_1 \). To see the other direction, if \( w \in L_1 \) and \( w = \epsilon \) then clearly \( w \in L_3^* \). On the other hand, if \( w \in L_1 \) and of the form \( 0^n \) then depending on \( n \) is odd or even it can be written as summation of \((2+2+\ldots+2+3)\) or \((2+2+\ldots+2)\) and thus \( w \in L_3^* \).

Finally to prove that option D is correct we will construct the DFA of \( L_3^R \) since regular languages are closed under reversal. Let \( A \) be the DFA of \( L_3^R \), construct a DFA \( A' \) same as \( A \) except that in \( A' \) make only those states final which on \( a \) moves to a state which is a final state in \( A \). It is easy to see that \( L(A') = L_3^R \).