1. Find all strings in \( L((a + b)^*b(a + ab)^*) \) of length less than four.

   **Answer.**
   
   The strings with length 1: \( \{ \lambda b \lambda \} = \{ b \} \);
   
   The strings with length 2: \( \{ ab \lambda, bb \lambda, \lambda b a \} = \{ ab, bb, ba \} \);
   
   The strings with length 3: \( \{ (aa) b \lambda, (ab) b \lambda, (ba) b \lambda, (bb) b \lambda, (a) b (a), (b) b (a), \lambda b (ab) \} = \{ aab, abb, bab, bbb, aba, bba, bab \} \).

2. Find a regular expression for the set \( \{ a^n b^m : (n + m) \text{ is even} \} \).

   **Answer.**
   
   There are two cases:
   
   - \( n \) and \( m \) are even: \( (aa)^*(bb^*) \);
   
   - \( n \) and \( m \) are odd: \( a(aa)^*(bb^*) \);
   
   Thus, a regular expression for the set \( \{ a^n b^m : (n + m) \text{ is even} \} \) is \( (aa)^*(bb^*) + a(aa)^*(bb^*) \).

3. Give a regular expression for \( L = \{ a^n b^m : n < 4, m \leq 3 \} \).

   **Answer.**
   
   A regular expression for \( L_1 = \{ a^n : n < 4 \} \) is \( \lambda + a + aa + aaa \);
   
   A regular expression for \( L_2 = \{ b^m : m \leq 3 \} \) is \( \lambda + b + bb + bbb \);
   
   Thus, a regular expression for \( L = L_1 L_2 \) is \( (\lambda + a + aa + aaa)(\lambda + b + bb + bbb) \).

4. What languages do the expressions \( (\emptyset^*)^* \) and \( a\emptyset \) denote?

   **Answer.**
   
   \( L((\emptyset^*)^*) = (L(\emptyset^*))^* = (\emptyset^*)^* = \{ \lambda \} \);
   
   \( L(a\emptyset) = L(a)L(\emptyset) = \{ a \} \{ \} = \emptyset \).

5. Find a regular expression for \( L = \{ vuv : v, w \in \{ a, b \}^*, |v| \leq 3 \} \).

   **Answer.**
   
   For any string \( x \in \{ a, b \}^* \), we can treat it as \( x = \lambda x \lambda \), where \( v = \lambda \), \( x = w \). Therefore, any string \( x \in \{ a, b \}^* \) is in \( L \). Thus, \( L = (a + b)^* \).

6. Give a regular expression for all strings that contain no run of \( a \)'s of length greater than two. (\( \Sigma = \{ a, b, c \} \)).

   **Answer.**
A regular expression for \( \{ a^n : n \leq 2 \} \) is \( \lambda + a + aa \).
Thus, for \( \Sigma = \{a, b, c\} \), a regular expression for all strings that contain no run of a’s of length greater than two is \( ((\lambda + a + aa)(b + c))^*(\lambda + a + aa) \). □

7. Give a regular expression for all strings with at most two occurrences of the substring 00. \( (\Sigma = \{0, 1\}) \).

**Answer.**

There are three cases for a string with at most two occurrences of 00:
- 0 occurrence of 00: \( \lambda \);
- 1 occurrence 00: 00;
- 2 occurrences 00: 000, 0011*00;

Thus, a regular expression for all strings with at most two occurrences of the substring 00 is \( 1 + 01)^*(\lambda + 00 + 000 + 0011*00)(1 + 10)^* \) □

8. Determine whether or not the following claim is true for all regular expressions \( r_1 \) and \( r_2 \). The symbol \( \equiv \) stands for equivalence regular expressions in the sense that both expressions denote the same language. \( r_1^*(r_1 + r_2)^* \equiv (r_1 + r_2)^* \).

**Answer.**

Since \( L(r_1^*(r_1 + r_2)^*) \subseteq L((r_1 + r_2)^*(r_1 + r_2)^*) = L((r_1 + r_2)^*) \) and \( L((r_1 + r_2)^*) = L(\lambda(r_1 + r_2)^*) \subseteq L(r_1^*(r_1 + r_2)^*) \), they are equivalent. □

9. Find an NFA that accepts the language \( L(aa^*(a + b)) \).

**Answer.**

The following graph represents the NFA \( M = (\{q_1, q_2, q_3\}, \{a, b\}, \delta, q_1, \{q_3\}) \) that accepts \( L(aa^*(a + b)) \), where \( \delta \) is described as in the graph. □
10. Use the construction in Theorem 3.1 to find an nfa that accepts the language $L(ab^*aa + bba^*ab)$.

**Answer.**

By Theorem 3.1, the automata for $L(a)$ is

![Automata for L(a)](image)

By Theorem 3.1, the automata for $L(a^*)$ is

![Automata for L(a*)](image)

The automata for $L(b)$ and $L(b^*)$ can be constructed in a similar way. Then by Theorem 3.1, the automata for $L(ab^*aa)$ is

![Automata for L(ab^*aa)](image)

Then by Theorem 3.1, the automata for $L(bba^*ab)$ is

![Automata for L(bba^*ab)](image)

Thus, by Theorem 3.1, the automata for $L(ab^*aa + bba^*ab)$ is

![Automata for L(ab^*aa + bba^*ab)](image)
11. Find DFA that accept \( L = L(ab^*a^*) \cap L((ab)^*ba) \).

**Answer.** \( L = L(ab^*a^*) \cap L((ab)^*ba) = \{abba\} \). The following graph represents the DFA \( M = (\{q_0, q_1, \ldots, q_5\}, \{a, b\}, \delta, q_0, \{q_4\}) \) that accepts \( L \), where \( \delta \) is described as in the graph.

![DFA Diagram](image)

12. Find regular expression for the language accepted by the following automata.

**Answer.**

We first convert the given NFA to a complete GTG as follows.

![NFA Diagram](image)

Then we reduce the state \( q_1 \) to obtain a two states one as follows.

![Reduced Diagram](image)
Then we have:

Finally, we obtain a regular expression:

\[ a^* \left( ab^* + c + a(a + b)^*(a + b) \right) (a + b)^* \]

13. Find a regular expression for \( L = \{ w : (n_a(w) - n_b(w)) \mod 3 = 1 \} \) on \( \{a, b\} \).

\textbf{Answer.}

The following figure shows the DFA of the language \( L \), where

- \( q_0 \): the state for \( n_a(w) - n_b(w) \mod 3 = 0 \);
- \( q_1 \): the state for \( n_a(w) - n_b(w) \mod 3 = 1 \);
- \( q_2 \): the state for \( n_a(w) - n_b(w) \mod 3 = 2 \);

Then, we use reduce the steps to obtain the following NFA:

Finally the regular expression is \( ((ba)^*(a + bb))((ab)^* + (b + aa)(ba)^*(a + bb))^* \).
14. Construct a dfa that accepts the language generated by the grammar

\[ S \rightarrow abA, \]
\[ A \rightarrow baB, \]
\[ B \rightarrow aA|bb. \]

**Answer.**

The dfa is constructed as follows, where \( q_x \) corresponds to variable \( x \), \( x \in \{S, A, B\} \).

15. Find a regular grammar for \( L = \{ w : |(n_a(w) - n_b(w))|\text{ is odd} \} \) on \( \{a, b\} \).

**Answer.**

\[ S \rightarrow aA|bA \]
\[ A \rightarrow aS|bS|\lambda \]

\( \square \)