Problem 1

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA where

\[ Q = \{q_0, q_1, q_2\}, \quad \Sigma = \{x, y\}, \quad F = \{q_1, q_2\}, \]

and where $\delta$ is defined by the transition table

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_0$</td>
<td>$q_1$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_0$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_0$</td>
<td>$q_2$</td>
</tr>
</tbody>
</table>

Draw the visual representation of $M$.

Problem 2

Give the formal descriptions of each of the following DFAs (i.e. set of states, transition table, etc.).
Problem 3
Prove that the following languages are regular by giving an example of a DFA or an NFA that recognizes them.

a) \( L_1 = \{ w \in \{0, 1\}^* \mid w \text{ begins and ends with a 0} \} \)

b) \( L_2 = \overline{L_1} \) (i.e. the strings that start or end with 1)

Note that we are only considering bit strings in this problem; therefore \( L_2 \) is also a subset of \( \{0, 1\}^* \).

Problem 4
Let \( \Sigma = \{a, b\} \cup \{N, /, *\} \) be the alphabet for this problem where \( N \) represents the “new line” character.

In Java, there are three different types of comments

1. */...*/,
2. //...N, and
3. /**...*/,

where the third type is a special Javadoc comment.

a) Construct an NFA that recognizes the language of all such comments. Note that the entire input string needs to be a comment, therefore your NFA should accept strings like ///abbaN and reject strings like ab///ababN and even //aaa (since the terminating newline is missing). We also point out that */aba/*aa*/ is a valid string of the first kind, but */aba*/aaa*/ is NOT (because the first */ terminates the comment).

b) IDEs often need to distinguish between regular comments and special Javadoc comments (e.g. for syntax highlighting). Construct an NFA that only recognizes strings of the first and second kind (i.e. it rejects Javadoc comments).

Note that /** is a valid comment of the first kind but /**/** is a comment of the third kind.
Bonus Problem: Ignoring whitespace

When parsing text files, it can be useful to ignore whitespace characters when searching for a string. If we have a string such as "professor", we might want to match strings like " p r o f e s s or". We say that strings of this form padded with the whitespace character. However, in this problem we will use the ⊥ symbol to represent whitespace (since a whitespace character is hard to draw).

More formally, if $A$ is a language over the alphabet $\Sigma$, then the padding of $A$ with $\bot$ is the language

$$
\left\{ w \in \Sigma^* \mid w = \bot^{k_1} a_1 \bot^{k_2} a_2 \cdots \bot^{k_n} a_n \bot^{k_{n+1}} \\
\text{where } a_1 a_2 \cdots a_n \in A \\
\text{and } k_i \in \mathbb{Z}_{\geq 0} \text{ for all } 1 \leq i \leq n + 1 \right\}.
$$

Note that if $A = \{abba\}$, then the padding of $A$ includes strings with an arbitrary number of whitespace characters inserted in between all the characters of $abba$, and therefore includes strings like $\bot\bot a \bot b b \bot a \bot$

Prove that the class of regular languages is closed under padding.