Problem 1

The figure below gives a state diagram for Machine $R$. Write down $R$ formally by specifying $(Q, \Sigma, \delta, q_0, F)$. Give as concise a description as possible of the language $G$ that $R$ recognizes. Then formally prove that $R$ recognizes $x$ if and only if $x \in G$.

$R$ is defined as the five-tuple $(Q, \Sigma, \delta, q_0, F)$ as follows:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- $\delta = \{$
  $(q_0, a) \rightarrow q_1$
  $(q_0, b) \rightarrow q_3$
  $(q_1, a) \rightarrow q_0$
  $(q_1, b) \rightarrow q_2$
  $(q_2, a) \rightarrow q_3$
  $(q_2, b) \rightarrow q_1$
  $(q_3, a) \rightarrow q_2$
  $(q_3, b) \rightarrow q_0$
$\}$
- $q_0 = q_0$
- $F = \{q_0\}$

$G$ is the language $R$ recognizes and it can either be the empty string or it consists of even numbers of $a$’s and $b$’s.
Proof.

1. If \( x \in G \), then \( R \) recognizes \( x \):

   • If \( x \) is the empty string, \( R \) accepts \( x \) at \( q_0 \)
   • If \( x \) has an even number of \( a \)'s and \( b \)'s, then \( x \) must be made up of matching \( a \)'s, \( b \)'s, or \((ab \mid ba)\)'s.
   • If \( x \) has only \( aa \)'s or \( bb \)'s, it stays in \( q_0 \)
   • If \( x \) consists only of \((ab \mid ba)(ab \mid ba)\)'s, it will go to \( q_3 \) then return to \( q_0 \)
   • If \( x \) contains \( a \)'s, \( b \)'s and \((ab \mid ba)(ab \mid ba)\)'s, it will still go to \( q_3 \) and return to \( q_0 \).

   Thus it always ends in \( q_0 \) while computing \( x \), so \( R \) accepts \( x \) if \( x \in G \)

2. If \( R \) recognizes \( x \), then \( x \in G \). \( R \) only accepts at \( q_0 \). To end in \( q_0 \), \( x \) can either be the empty string or contains only matching pairs of \( a \)'s, \( b \)'s, or \((ab \mid ba)\)'s. If \( x \) contains an odd number of \( a \)'s and \( b \)'s, it will not end in \( q_0 \), and \( R \) will not accept. If \( R \) accepts \( x \), then \( x \in G \).

\[ \square \]

Problem 2

Consider the set of all strings over the 26-letter lowercase English alphabet. Suppose we wish to design a DFA \( J \) that will accept any such finite string, provided it contains the substring “jumbo” somewhere in it. Can you provide the state diagram for \( J \)? Now we instead want to design the DFA \( T \) that will instead accept any string that contains the substring “tufts”. Can you design \( T \)?

![Figure 1: DFA J](image-url)
Figure 2: DFA $T$