1. Is there any input for which the following Turing machine that accepts \( L = \{a^n b^n : n \geq 1\} \) goes into an infinite loop?

\[
M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F),
\]

where \( Q = \{q_0, q_1, \ldots, q_4\} \), \( F = \{q_4\} \), \( \Sigma = \{a, b\} \), \( \Gamma = \{a, b, x, y, \square\} \), and the transitions are

\[
\begin{align*}
\delta(q_0, a) &= (q_1, x, R), \\
\delta(q_1, a) &= (q_1, a, R), \\
\delta(q_1, y) &= (q_1, y, R), \quad \text{and} \\
\delta(q_1, b) &= (q_2, y, L).
\end{align*}
\]

**Ans.** No, the Turing machine is always halt for any inputs.

- If \( w = a^n b^n \in L \), the machine halts in the final state \( q_4 \).
- If \( w = a^k b^\ell \not\in L \),
  - \( k > \ell \): The machine halts in state \( q_1 \).
  - \( k < \ell \): The machine halts in state \( q_3 \).
  - \( k = 0 \) or \( \ell = 0 \): The machine halts in state \( q_1 \) or \( q_0 \).

2. What language is accepted by the Turing machine whose transition graph is in the figure below?

**Ans.** \( L = ab^* + bb^*a \).

3. Given \( \Sigma = \{a, b\} \), construct a Turing machine that accepts \( L = \{a^n b^{2n} : n \geq 0\} \).

**Ans.** A Turing machine that accepts \( L \) is \( M = (\{q_0, q_1, \ldots, q_5\}, \{a, b\}, \{a, b, x, y, \square\}, \delta, q_0, \square, \{q_5\}) \) with the following transition graph.
4. Design a Turing machine to compute $f(x) = x/2$ if $x$ is even and $f(x) = (x+1)/2$ if $x$ is odd, where $x$ is a positive integer represented in unary.

**Ans.** A Turing machine that computes $f(x)$ is $M = (\{q_0, q_1, \ldots, q_4\}, \{0, 1\}, \{0, 1, x, \square\}, \delta, q_0, \square, \{q_4\})$ with the following transition graph.

![Turing machine transition graph]

5. Using adders, subtracters, comparers, copiers, or multipliers, draw block diagrams for Turing machines that computes the function $f(n) = 2^n$ for all positive integers $n$.

**Ans.**

![Block diagram for $f(n) = 2^n$]

6. Give convincing arguments that any language accepted by an off-line Turing machine is also accepted by some standard machine.

**Ans.** The following shows that the off-line Turing machine and standard Turing machine are equivalent.

- The behavior of any standard Turing machine can be simulated by some off-line Turing machine. All that needs to be done by the simulating machine is to copy the input from the input file to the tape. The it can proceed in the same way as the standard machine.
- A standard Turing machine $\hat{M}$ can simulate the computation of an off-line Turing machine $M$ by using the four-track arrangement. The first track has the input, the second marks the position at which the input is read, the third represents the tape of $M$, and the fourth shows the position of $M$’s read-write head. The simulation of each move of $M$ requires a number of moves of $\hat{M}$. Starting from some standard position, say the left end, and with the relevant information marked by special end markers, $\hat{M}$ searches track 2 to locate the position at which the input file of $M$ is read. The symbol found in the corresponding cell on track 1 is remembered by putting the control unit of $\hat{M}$ into a state chosen for this purpose. Next, track 4 is searched for the position of the read-write head of $M$. With the remembered input and the symbol on track 3, we now know that $M$ is to do. This information is again remembered by $\hat{M}$ with an appropriate internal state. Next, all four tracks of $\hat{M}$’s tape are modified to reflect the move of $M$. Finally, the read-write head of $\hat{M}$ returns to the standard position for the simulation of the next move.
Therefore, a language accepted by an off-line Turing machine is also accepted by some standard machine.

7. Suppose we make the requirement that a Turing machine can halt only in a final state, that is, we ask that \( \delta(q, a) \) be defined for all pairs \((q, a)\) with \(a \in \Gamma\) and \(q \notin F\). Does this restrict the power of the Turing machine?

**Ans.** The following shows that the halt-in-final Turing machine and standard Turing machine are equivalent.

- Since a halt-in-final Turing machine is clearly an extension of the standard Turing machine, it is obvious that any standard Turing machine can be simulated by some halt-in-final Turing machine.
- A standard Turing machine \(\tilde{M}\) can simulate the computation of a halt-in-final Turing machine by using the following arrangement.
  - Create a new trap-state \(q_{\text{trap}}\) with transitions to itself for all symbol \(a \in \Gamma\), i.e., \(\delta(q_{\text{trap}}, a) = (q_{\text{trap}}, a, L\) or \(R)\).
  - For each non-final state \(q\), we define a new transition that bring each unused symbol \(a\), which causes halt in state \(q\), to the trap-state \(q_{\text{trap}}\).

Therefore, the modification in the halt-in-final Turing machine does not restrict the power of the Turing machine.

8. Outline how one would write a program for nondeterministic Turing machine to accept the language \(L = \{ww^Rw : w \in \{a, b\}^+\}\).

**Ans.**

- Given an input string \(w = w_1w_2 \ldots w_\ell\). If \(\ell = 0\) or \(\ell \mod 3 \neq 0\), reject input. Otherwise, separate \(w\) into three parts \(w^{(1)} = w_1w_2 \ldots w_{\ell/3}\), \(w^{(2)} = w_{\ell/3+1}w_{\ell/3+2} \ldots w_{2\ell/3}\), and \(w^{(3)} = w_{2\ell/3+1}w_{2\ell/3+2} \ldots w_\ell\).
- Start at the left of \(w\). Remember \(w^{(1)}\)'s first non-\(X\) symbol \(a\) and last non-\(X\) symbol \(b\) by putting the machine in the appropriate state and replace them with \(X\). If \(w^{(1)}\) remains one non-\(X\) symbol, remember its first non-\(X\) symbol \(a\) and last non-\(X\) symbol \(b\) with \(a = b\) and replace it with \(X\).
- Move the read-write head to \(w^{(2)}\)'s first non-\(Y\) symbol and last non-\(Y\) symbol, and \(w^{(3)}\)'s first non-\(Y\) symbol and last non-\(Y\) symbol, and compare them with the remembered symbols \(b, a, a, b\), respectively. If \(w^{(2)}\) and \(w^{(3)}\) remain one non-\(Y\) symbol, the read-write head would stop at the same positions and do the comparison. If any of them do not match, reject the input. Otherwise, replace them with \(Y\).
- Accept \(w\).

9. Give a complete encoding, using the suggested method, for the Turing machine with \(\delta(q_1, a_1) = (q_1, a_1, R), \delta(q_1, a_2) = (q_3, a_1, L), \delta(q_3, a_1) = (q_2, a_2, L)\).

**Ans.**

```
 1010101011 00 101101110101 00 1110101101101.
  first transition  second transition  third transition
```