Exercise  (Example 8.2 from textbook)

Construct a Turing Machine accepting the language

\[ \{0^m1^n \mid m \geq 1\} \]

Solution

The idea is that the TM M that we construct needs the leftmost 0, turns it into X, and moves right until it reaches a 1, that is turned into Y. Then the head moves left again to the leftmost 0 (on the right to a X), and starts again until all 0's and 1's are turned into X's and Y's respectively.

If the input is not in 0*1*, M will fail to find a move and it won't accept. If M changes the last 0 and the last 1 in the same round, it will go into the final state and accept.

\[ Q = \{ q_0, q_1, q_2, q_3, q_4 \} \]
\[ \Sigma = \{ 0, 1 \} \]
\[ \Gamma = \{ 0, 1, X, Y, \delta \} \]

\(q_0\) : start state
\(f = \{ q_9 \}\)

In \(q_0\) is the state in which M is when the head precedes the leftmost 0. In state \(q_1\), M moves right skipping 0's and Y's until it gets to a 1. In state \(q_2\), M moves left while skipping Y's and 0's again, until it gets to a X and goes again in \(q_0\).
Starting from $q_0$, if a 1 is read instead of a 0, H goes in $q_3$ and moves right; if a 1 is found, then there are more 1's than 0's; if a 0 is read, then the initial string is accepted (transition to $q_4$).

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>X</th>
<th>Y</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$(q_1, x, R)$</td>
<td>–</td>
<td>–</td>
<td>$(q_3, Y, R)$</td>
<td>–</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$(q_2, 0, R)$</td>
<td>$(q_2, y, L)$</td>
<td>–</td>
<td>$(q_2, y, R)$</td>
<td>–</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$(q_2, 0, L)$</td>
<td>–</td>
<td>$(q_0, x, R)$</td>
<td>$(q_2, y, L)$</td>
<td>–</td>
</tr>
<tr>
<td>$q_3$</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>$(q_3, y, R)$</td>
<td>$(q_4, 5, R)$</td>
</tr>
<tr>
<td>$q_4$</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Exercise**

Show the computation of the TM above when the input string is:

(a) 00
(b) 000111

**Solution**

(a) $q_0 \delta 00 \rightarrow xq_1 \delta 01 \rightarrow x0q_2 \delta 1$

and the TM halts.

(b) $q_0 \delta 000111 \rightarrow xq_2 \delta 0111 \rightarrow x0q_3 \delta 111 \rightarrow$

$x00q_4 \delta 11 \rightarrow x0q_2 \delta 11 \rightarrow xq_2 \delta 011 \rightarrow q_2 \delta 00y11 \rightarrow$

$q_2 \delta 0y11 \rightarrow xxq_4 \delta 11 \rightarrow xxq_2 \delta 11 \rightarrow xx0q_3 \delta 11 \rightarrow$

$xx0q_2 \delta 11 \rightarrow xxq_0 \delta 11 \rightarrow xq_0 \delta y11 \rightarrow$

$xq_2 \delta yy1 \rightarrow xxq_2 \delta yy1 \rightarrow q_2 \delta 0yy1 \rightarrow xq_0 \delta 0yy1 \rightarrow$

$xxq_4 \delta yy1 \rightarrow xxq_2 \delta yy1 \rightarrow xxxq_2 \delta yy1 \rightarrow xxxyq_2 \delta yy1 \rightarrow$

$xxxq_2 \delta yy1 \rightarrow xq_2 \delta yy1 \rightarrow xxyq_2 \delta yy1 \rightarrow$

$xxxq_2 \delta yy1 \rightarrow xxyq_3 \delta yy1 \rightarrow xxyq_5 \delta yy1 \rightarrow$
Exercise (8.2.3 from textbook):

Design a Turing Machine that takes as input a number $N$ in binary and turns it into $N+1$ (in binary); the number $N$ is preceded by the symbol $\$, which may be destroyed during the computation. For example, $111$ is turned into $1000$; $1001$ is turned into $1010$.

Solution

The idea is to toggle the rightmost digit, and, from right to left, all consecutive $1$'s until we get to the first $0$ (which is also toggled). If there is no $0$ to be toggled, a $1$ is added on the left of the first digit (i.e., in place of the $\$).

We need three states, where only $q_2$ is the final state; we briefly describe what the TM does in the different states.

$q_0$: The TM goes right until it reaches $\$, after the rightmost digit. When $\$ is reached, the TM goes into $q_2$.

$q_1$: Goes left toggling all $1$'s and the first $0$ (from right); when $0$ or $\$ is reached, the symbol is turned into $1$.

$q_2$: Final state; the TM does nothing.

<table>
<thead>
<tr>
<th></th>
<th>$$</th>
<th>0</th>
<th>1</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$(q_0, $, R)</td>
<td>$(q_0, 0, R)$</td>
<td>$(q_0, 1, R)$</td>
<td>$(q_2, 5, L)$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$(q_1, 1, L)$</td>
<td>$(q_1, 1, L)$</td>
<td>$(q_1, 0, L)$</td>
<td>—</td>
</tr>
<tr>
<td>$q_2$</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Exercise (8.22 from textbook)

Design a Turing machine accepting the following language:
\[
\{ w \in \{0,1\}^* \mid \text{has an equal number of 0's and 1's} \}
\]

Solution

The idea is that the head of our TM \( M \) moves back and forth on the tape, "deleting" one 0 for each 1; if there are no 0's and 1's in the end, the string is accepted.

When in state \( q_2 \), \( M \) has found a 1 and looks for a 0; in state \( q_2 \) it is the other way around.

Note that the head never moves left of any \( X \), so that there are never unmatched 0's and 1's on the left of an \( X \).

From initial state \( q_0 \), \( M \) picks up a 0 or a 1 and turns it into \( X \). The only final state is \( q_4 \). In state \( q_3 \), \( M \) moves head left looking for the rightmost \( X \).

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
 & 0 & 1 & \varepsilon & X & Y \\
\hline
q_0 & (q_2, x, R) & (q_2, x, R) & (q_4, x, R) & - & (q_0, y, R) \\
q_1 & (q_3, y, L) & (q_2, x, R) & - & - & (q_2, y, R) \\
q_2 & (q_2, 0, R) & (q_2, y, L) & - & - & (q_2, y, R) \\
q_3 & (q_3, 0, L) & (q_3, y, L) & - & (q_0, x, R) & (q_3, y, L) \\
q_4 & - & - & - & - & - \\
\hline
\end{array}
\]