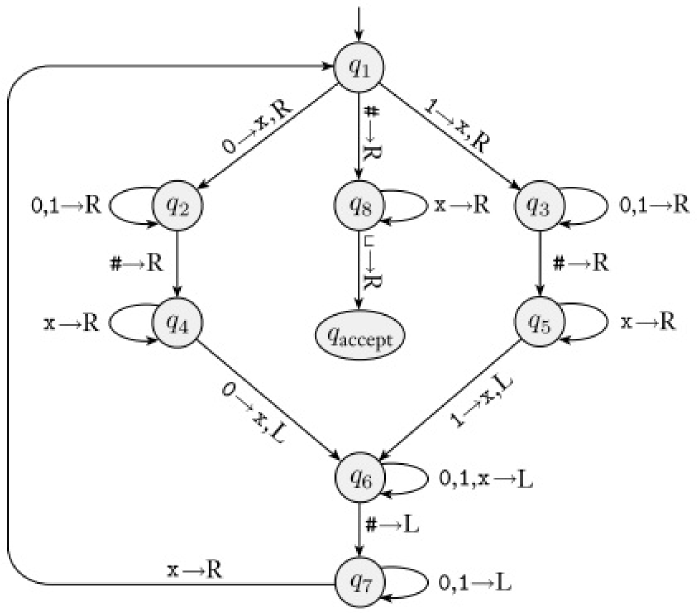
3.2 This exercise concerns TM M_1, whose description and state diagram appear in Example 3.9. In each of the parts, give the sequence of configurations that M_1 enters when started on the indicated input string.

[A] **a.** 11.
b. 1#1.
c. 1##1.
d. 10#11.
e. 10#10.

EXAMPLE 3.9

The following is a formal description of $M_1 = (Q, \Sigma, \Gamma, \delta, q_1, q_{\text{accept}}, q_{\text{reject}})$, the Turing machine that we informally described (page 167) for deciding the language $B = \{w\#w \mid w \in \{0,1\}^*\}$.

- $Q = \{q_1, \ldots, q_8, q_{\text{accept}}, q_{\text{reject}}\}$,
- $\Sigma = \{0,1,\#\}$, and $\Gamma = \{0,1,\#,x,\sqcup\}$.
- We describe δ with a state diagram (see the following figure).
- The start, accept, and reject states are q_1, q_{accept}, and q_{reject}, respectively.



a.) Inp 11

	Estado	conf
0	q_1	$q_1 11$
1	q_3	$Xq_3 1$
2	q_3	$X1q_3$

Final X1 //

c.) Inp 1##1

0	q_1	$q_1 1\#\#1$
1	q_3	$Xq_3 \#\#1$
2	q_5	$X\#q_5\#1$

Final X##1 //

b.) Inp 1#1

	E	conf
0	q_1	$q_1 1\#1$
1	q_3	$Xq_3\#1$
2	q_5	$X\#q_5 1$
3	q_6	$Xq_6\#X$
4	q_7	$q_7 X\#X$
5	q_1	$Xq_1\#X$
6	q_8	$X\#q_8 X$
7	q_8	$X\#Xq_8$

Aceptada //

d.) 10#11

0	q_1	$q_1 10\#11$
1	q_3	$Xq_3 0\#11$
2	q_3	$X0q_3\#11$
3	q_5	$X0\#q_5 11$
4	q_6	$X0q_6\#X1$
5	q_7	$Xq_7 0\#X1$
6	q_7	$q_7 X0\#X1$
7	q_1	$Xq_1 0\#X1$
8	q_2	$XXq_2\#X1$
9	q_4	$XX\#q_4 X1$
10	q_4	$XX\#Xq_4 1$

Final XX#X1 //

e. Inp 10#10

0	q_1	$q_1 10\#10$
1	q_3	$Xq_3 0\#10$
2	q_3	$X0q_3\#10$
3	q_5	$X0\#q_5 10$
4	q_6	$X0q_6\#X0$
5	q_7	$Xq_7 0\#X0$
6	q_7	$q_7 X0\#X0$
7	q_1	$Xq_1 0\#X0$
8	q_2	$XXq_2\#X0$
9	q_4	$XX\#q_4 X0$
10	q_4	$XX\#Xq_4 0$
11	q_6	$XX\#q_6 XX$
12	q_6	$XXq_6\#XX$
13	q_7	$Xq_7 X\#XX$
14	q_1	$XXq_1\#XX$
15	q_8	$XX\#q_8 XX$
16	q_8	$XX\#Xq_8 X$
17	q_8	$XX\#XXq_8$
18	Aceptada //	

3.6 In Theorem 3.21, we showed that a language is Turing-recognizable iff some enumerator enumerates it. Why didn't we use the following simpler algorithm for the forward direction of the proof? As before, $s_1, s_2, \ldots$ is a list of all strings in Σ^*.

$E =$ "Ignore the input.
  1. Repeat the following for $i = 1, 2, 3, \ldots$.
  2. Run M on s_i.
  3. If it accepts, print out s_i."

THEOREM 3.21

A language is Turing-recognizable if and only if some enumerator enumerates it.

PROOF First we show that if we have an enumerator E that enumerates a language A, a TM M recognizes A. The TM M works in the following way.

$M =$ "On input w:
  1. Run E. Every time that E outputs a string, compare it with w.
  2. If w ever appears in the output of E, *accept*."

Clearly, M accepts those strings that appear on E's list.
Now we do the other direction. If TM M recognizes a language A, we can construct the following enumerator E for A. Say that $s_1, s_2, s_3, \ldots$ is a list of all possible strings in Σ^*.

$E =$ "Ignore the input.
  1. Repeat the following for $i = 1, 2, 3, \ldots$.
  2. Run M for i steps on each input, $s_1, s_2, \ldots, s_i$.
  3. If any computations accept, print out the corresponding s_j."

If M accepts a particular string s, eventually it will appear on the list generated by E. In fact, it will appear on the list infinitely many times because M runs from the beginning on each string for each repetition of step 1. This procedure gives the effect of running M in parallel on all possible input strings.

3.8 Give implementation-level descriptions of Turing machines that decide the following languages over the alphabet {0,1}.

[redacted]

b. $\{w \mid w$ contains twice as many 0s as 1s$\}$

Ejemplo

Inp	010
q_0	q_0 1 0
q_2	x q_2 1 0
q_3	x y q_3 0
q_6	x q_6 y y
q_6	q_6 x y y
q_0	y q_0 y y
q_0	y y q_0 y
q_0	y y y q_0

Aceptado