Montogue

# Quiz EL503
# Turing Machines

Lucas Monteiro Nogueira

**Note:** In the following problems, the blank state in a Turing machine is interchangeably denoted by $B$ or #.

## PROBLEMS

### Problem 1

Regarding the theory of Turing machines, true or false?

**1.( )** Every language accepted by a one-tape, read-only deterministic Turing machine is regular.

**2.( )** Some of the languages computed by a two-way infinite one-tape deterministic Turing machine are not Turing-computable.

**3.( )** In the formal definition of a one-tape deterministic Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$, the tape alphabet $\Gamma$ can be the same as the input alphabet $\Sigma$.

**4.( )** The set of Turing machines is both infinite and uncountable.

**5.( )** If $L$ is a language enumerated by a Turing machine, then $L$ is certain to be recursively enumerable.

**6.( )** The following is a description of a legitimate Turing machine.

$M_{bad}$ = "On input $\langle p \rangle$, a polynomial over variables $x_1, ..., x_k$:
1. Try all possible settings of $x_1, ..., x_k$ to integer values.
2. Evaluate $p$ on all of these settings.
3. If any of these settings evaluates to 0, *accept*; otherwise, *reject*."

**7.( )** A $k$-tape Turing machine can be simulated by a single-tape TM with $2k$ tracks.

**8.( )** If a Turing machine has transitions specified by

$$\delta(q_0, a) = \{(q_1, b, R), (q_2, c, L)\}$$

we can surmise that the machine is nondeterministic. ($q_0$ is the initial state.)

**9.( )** The one-tape deterministic Turing machine described by the following transition table accepts all strings having even occurrences of 2's over the alphabet $\Sigma = \{1,2\}$. (The arrow next to $q_0$ indicates that it is the initial state; state $H$ is the halt state.)

| $\delta$ | 1 | 2 | $B$ |
|---|---|---|---|
| $\to q_0$ | $q_0, 1, R$ | $q_1, 2, R$ | $H, B, R$ |
| $q_1$ | $q_1, 1, R$ | $q_0, 2, R$ | – |
| $H$ | – | – | – |

**10.( )** The following table describes the transition rules for a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$, where $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $F = \{q_4\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, x, y, B\}$. Accept state is $q_{accept} = \{q_4\}$, and reject states $q_{reject}$ are any $q_i \in Q - \{q_4\}$. The arrow next to $q_0$ indicates that it is the start state. It can be shown that string *aabb* is rejected by this machine.

| $\delta$ | $a$ | $b$ | $x$ | $y$ | $B$ |
|---|---|---|---|---|---|
| $\to q_0$ | $q_1, x, R$ | $q_1, 2, R$ | $H, B, R$ | $q_3, y, R$ | – |
| $q_1$ | $q_1, a, R$ | $q_2, y, L$ | – | $q_1, y, R$ | – |
| $q_2$ | $q_2, a, L$ | – | $q_0, x, R$ | $q_2, y, L$ | – |
| $q_3$ | – | – | – | $q_3, y, R$ | $q_4, B, R$ |
| $q_4$ | – | – | – | – | – |

## Problem 2 (Sipser, 2013, w/ permission)

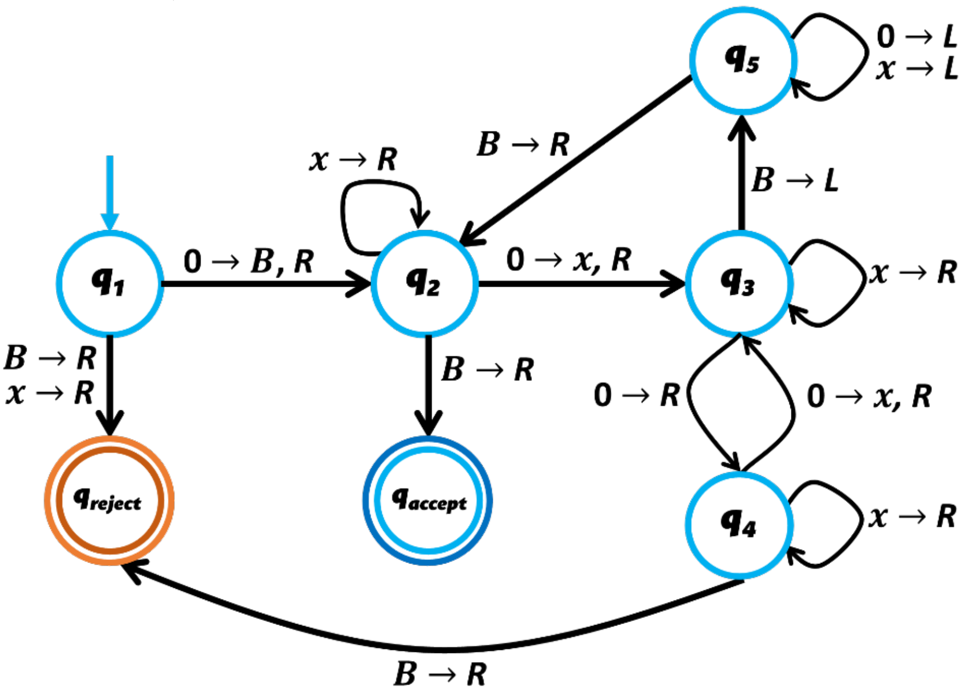
The following transition diagram describes the Turing machine $M_2$ which decides $A = \{0^{2^n} \mid n \geq 0\}$, the language consisting of all strings of 0's whose length is a power of 2. Formally, $M_1$ is described by a 7-tuple $M_1 = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$ such that

$$Q = \{q_1, q_2, \ldots, q_5, q_{accept}, q_{reject}\}$$
$$\Sigma = \{0\}$$
$$\Gamma = \{0, x, B\}$$

In each of the following parts, give the sequence of configurations that $M_1$ enters when started on the indicated input string.

**Problem 2.1:** Input 00

**Problem 2.2:** Input 000000




## Problem 3 (Sipser, 2013, w/ permission)

The following transition diagram describes the Turing machine $M_1 = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$ that decides the language

$$A = \{w\#w \mid w \in \{0,1\}^*\}$$

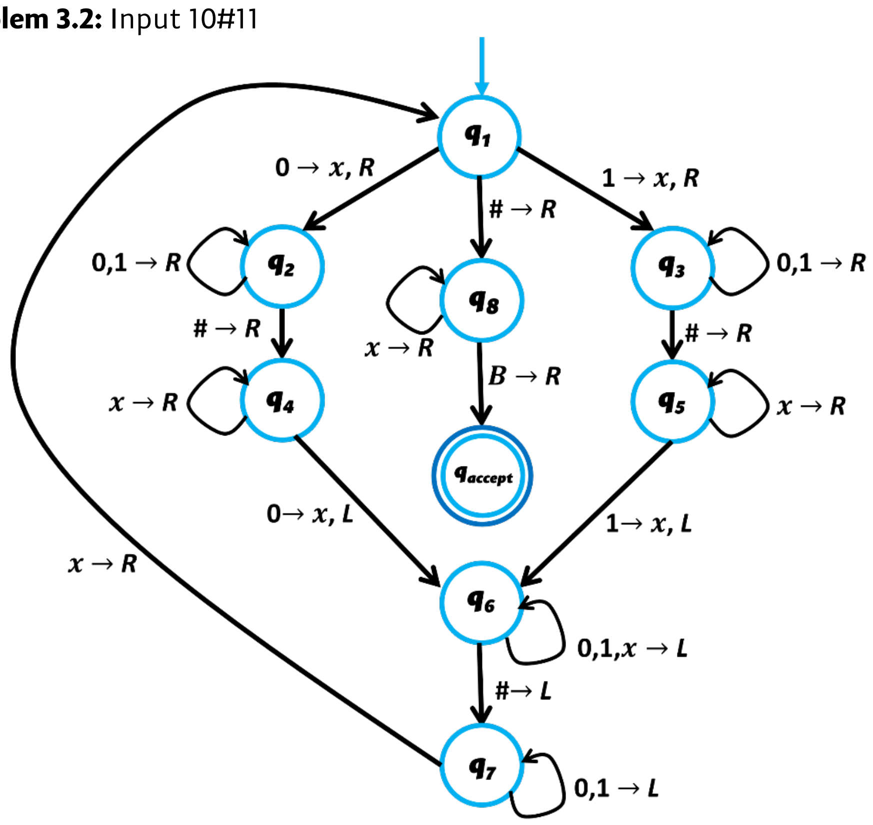
where the sets $Q$, $\Sigma$, and $\Gamma$ are described as

$$Q = \{q_1, q_2, \ldots, q_8, q_{accept}, q_{reject}\}$$
$$\Sigma = \{0, 1, \#\}$$
$$\Gamma = \{0, 1, \#, x, B\}$$

In each of the following parts, give the sequence of configurations that $M_2$ enters when started on the indicated input string.

**Problem 3.1:** Input 1#1

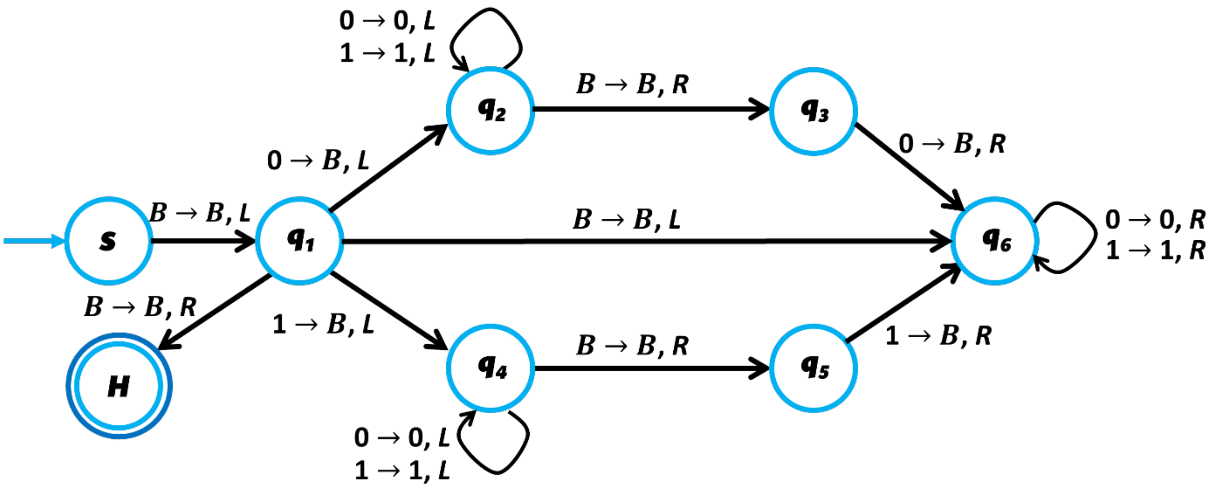
**Problem 3.2:** Input 10#11



## Problem 4

Using the transition diagram shown in continuation, fill up the table with the appropriate transition rules. The starting state is $s$, and the halt state is $H$.

| Present state | 0 | 1 | $B$ |
|---|---|---|---|
| $\rightarrow s$ | | | |
| $q_1$ | | | |
| $q_2$ | | | |
| $q_3$ | | | |
| $q_4$ | | | |
| $q_5$ | | | |
| $q_6$ | | | |
| $H$ | | | |




## Problem 5

Construct a transition diagram for the Turing machine whose transfer function is described by the following table. The arrow next to $q_0$ indicates that $q_0$ is the initial state.

| Present state | 1 | $B$ |
|---|---|---|
| $\rightarrow q_0$ | $q_0, 1, R$ | $q_2, 1, R$ |
| $q_1$ | $q_1, 1, R$ | $q_2, B, L$ |
| $q_2$ | $q_3, B, L$ | — |
| $q_3$ | $q_3, 1, L$ | $H, B, R$ |
| $H$ | — | — |

## Problem 6 (Sipser, 2013, w/ permission)

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

**Problem 6.1:** $\{w \mid w$ contains twice as many 0's as 1's$\}$

**Problem 6.2:** $\{w \mid w$ does not contain twice as many 0's as 1's$\}$

## Problem 7

Design a Turing machine for the following language.

$\{w \mid w \in$ set of strings with an equal number of 0's and 1's$\}$

## Problem 8 (Hopcroft *et al.*, 2001, w/ permission)

**Problem 8.1:** A $k$-tape Turing machine can be simulated by a one-tape TM with $2k$ tracks. One such technique is explained in the box at the Additional Information section. In a simulation of this kind, suppose the technique is used to simulate a 5-tape TM that had a tape alphabet of seven symbols. How many tape symbols would the one-tape TM have?

**Problem 8.2:** An alternative way to simulate $k$ tapes by one is to use a $(k + 1)$st track to hold the head positions of all $k$ tapes, while the first $k$ tracks simulate the $k$ tapes in the obvious manner. Note that in the $(k + 1)$st track, we must be careful to distinguish among the tape heads and to allow for the possibility that two or more heads are at the same cell. Does this method reduce the number of tape symbols needed for the one-tape TM?

## Problem 9 (Sudkamp, 2006)

Let $L$ be the set of palindromes over $\{a, b\}$. Build a two-tape TM that accepts $L$ in which the computation with input $w$ should take no more than $3length(u) + 4$ transitions.

## Problem 10 (Sudkamp, 2006)

**Problem 10.1[1]:** Construct a deterministic Turing machine with input alphabet $\{a, b\}$ to accept the language

$$L = \{w \mid w \text{ has same number of } a\text{'s and } b\text{'s}\}$$

**Problem 10.2:** Construct a nondeterministic Turing machine whose language is the set of strings over $\{a, b\}$ that contain a substring $u$ satisfying the following two properties:

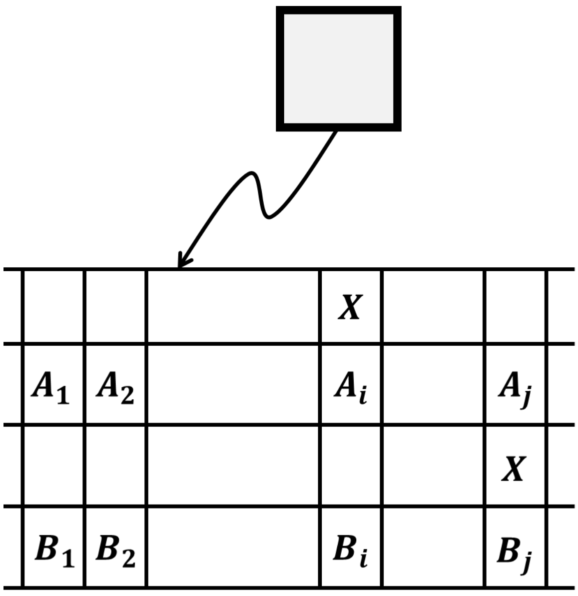
*1)* $length(u) \geq 3$.

*2)* $u$ contains the same number of $a$'s and $b$'s.

## ADDITIONAL INFORMATION

**Simulating a $k$-tape deterministic Turing machine with a one-tape DTM**
(*Excerpt from Hopcroft* et al. *(2001)*)

Suppose a language $L$ is accepted by a $k$-tape Turing machine $M$. We can simulate $M$ with a one-tape Turing machine $N$ whose tape we think of as having $2k$ tracks. Half these tracks hold the tapes of $M$, and the other half of the tracks hold only a single marker that indicates where the head for the corresponding $M$ is currently located. The figure below assumes $k = 2$. The second and fourth tracks hold the contents of the first and second tapes of $M$, track 1 holds the position of the head of tape 1, and track 3 holds the position of the second tape head.




To simulate a move of $M$, $N$'s head must visit the $k$ head markers. So that $N$ not get lost, it must remember how many head markers are to its left at all times; that count is stored as a component of $N$'s finite control. After visiting each head marker and storing the scanned symbol in a component of its finite control, $N$ knows what tape symbols are being scanned by each of $M$'s heads. $N$ also knows the state of $M$, which it stores in $N$'s own finite control. Thus, $N$ knows what move $M$ will make.

$N$ now revisits each of the head markers on its tape, changes the symbol in the track representing the corresponding tapes of $M$, and moves the head markers left or right, if necessary. Finally, $N$ changes the state of $M$ as recorded in its own finite control. At this point, $N$ has simulated one move of $M$.

We select as $N$'s accepting states all those states that record $M$'s state as one of the accepting states of $M$. Thus, whenever the simulated $M$ accepts, $N$ also accepts, and $N$ does not accept otherwise.

[1] Similar to Problem 7.

## SOLUTIONS

### P.1 ➡ Solution

**1. True.** Nothing to add here.

**2. False.** In actuality, every language accepted by a two-way infinite one-tape DTM is Turing-computable. Also, every language that is accepted by a two-way infinite one-tape DTM is Turing-acceptable.

**3. False.** $\Gamma$ and $\Sigma$ cannot be the same because $\Gamma$ contains the empty symbol $B$, which never occurs in $\Sigma$.

**4. False.** In actuality, the set of Turing machines, although infinite, is countable. A simple enumeration procedure for Turing machines is outlined in Linz's textbook. In Linz's treatment, each Turing machine can be encoded using 0 or 1. With this encoding, we proceed as follows.

*Step 1*: Generate the next string in $\{0, 1\}^+$ in proper order.

*Step 2:* Check the generated string to see if it defines a Turing machine. If so, write it on the tape in the appropriate form (see definition 10.4 of Linz's textbook). If not, ignore the string.

*Step 3:* Return to step 1.

Since every Turing machine has a finite description, any specific machine will eventually be generated by this process.

*Reference*: Linz (2017).

**5. True.** Put simply, a recursively enumerable language is a formal language for which there exists a Turing machine that can enumerate all valid strings of the language.

**6. False.** The variables $x_1, \ldots, x_k$ have infinitely many possible strings. A Turing machine would require infinite time to try them all. However, by definition a Turing machine is only feasible if it can perform a given computation stage in a finite number of steps.

**7. True.** This topic is elaborated upon in Problem 8.

**8. True.** The machine is nondeterministic because two different transition rules stem from the same state-input combination.

**9. True.** It is easy to see that the halt state $H$ will only be achieved once an even number of 2's has been read.

**10. False.** The sequence of transitions is shown below.

$$q_0aabb \Rightarrow xq_1abb \Rightarrow xaq_1bb \Rightarrow xq_2ayb \Rightarrow q_2xayb \Rightarrow xq_0ayb \Rightarrow xxq_1yb \Rightarrow xxyq_1b \Rightarrow$$
$$xxyq_1b \Rightarrow xxq_2yy \Rightarrow xq_2xyy \Rightarrow xxq_0yy \Rightarrow xxyq_3y \Rightarrow xxyyq_3B \Rightarrow xxyyBq_4B$$

Since the machine ends the transition sequence in state $q_4$, we conclude that the string in question is *accepted* by the TM at hand.

### P.2 ➡ Solution

**Problem 2.1:** Consider first the operation of the machine under input 00. We begin by writing the state-input string $q_1 00$. Referring to the state diagram, we see that upon reading an input 0 at state $q_1$, the TM transitions to $q_2$, is displaced to the right, and writes a $B$. That is,

$$q_1 00 \Rightarrow Bq_2 0$$

Now, upon reading an input 0 at state $q_2$, the machine transitions to $q_3$, is displaced to the right, and writes an $x$. Accordingly,

$$q_1 00 \Rightarrow Bq_2 0 \Rightarrow Bxq_3B$$

Upon reading an input $B$ at state $q_3$, the machine transitions to $q_5$, is displaced to the left, and writes nothing. It follows that

$$q_1 00 \Rightarrow Bq_2 0 \Rightarrow Bxq_3B \Rightarrow Bq_5xB$$

At state $q_5$, the machine reads an $x$, remains in the same state, is displaced to the left, and writes nothing. Thus,

$$q_1 00 \Rightarrow Bq_2 0 \Rightarrow Bxq_3B \Rightarrow Bq_5xB \Rightarrow q_5BxB$$

At state $q_5$, the machine reads a $B$, transitions to $q_2$, and writes nothing. Thus,

$$q_1 00 \Rightarrow Bq_2 0 \Rightarrow Bxq_3B \Rightarrow Bq_5xB \Rightarrow q_5BxB \Rightarrow Bq_2xB$$

At state $q_2$, the machine reads an $x$, remains in the same state, is displaced to the right, and writes nothing. Thus,

$$q_1 00 \Rightarrow Bq_2 0 \Rightarrow Bxq_3B \Rightarrow Bq_5xB \Rightarrow q_5BxB \Rightarrow Bq_2xB \Rightarrow Bxq_2B$$

Finally, at state $q_2$ the machine reads a $B$, transitions to $q_{\text{accept}}$, is displaced to the right, and writes nothing. Thus,

$$q_1 00 \Rightarrow Bq_2 0 \Rightarrow Bxq_3 B \Rightarrow Bq_5 xB \Rightarrow q_5 BxB \Rightarrow Bq_2 xB \Rightarrow Bxq_2 B \Rightarrow BxBq_{accept}$$

The TM has entered the accept state. Thus, string input 00 is accepted.

**Problem 2.2:** We begin by writing the state-input string $q_1$000000. Upon receiving an input 0 at $q_1$, the TM transitions to $q_2$, is displaced to the right, and writes a $B$. That is,

$$q_1 000000 \Rightarrow Bq_2 00000$$

Now, upon reading an input 0 at state $q_2$, the machine transitions to $q_3$, is displaced to the right, and writes an $x$. Accordingly,

$$q_1 000000 \Rightarrow Bq_2 00000 \Rightarrow Bxq_3 0000$$

At state $q_2$, the machine reads a 0, transitions to $q_4$, is displaced to the right, and writes nothing. It follows that

$$q_1 000000 \Rightarrow Bq_2 00000 \Rightarrow Bxq_3 0000 \Rightarrow Bx0q_4 000$$

At state $q_4$, the machine reads a 0, transitions to $q_3$, is displaced to the right, and writes an $x$. Thus,

$$q_1 000000 \Rightarrow Bq_2 00000 \Rightarrow Bxq_3 0000 \Rightarrow Bx0q_4 000 \Rightarrow Bx0xq_3 00$$

At state $q_3$, the machine reads a 0, transitions (back) to $q_4$, is displaced to the right, and writes nothing. Thus,

$$q_1 000000 \Rightarrow Bq_2 00000 \Rightarrow Bxq_3 0000 \Rightarrow Bx0q_4 000 \Rightarrow Bx0xq_3 00 \Rightarrow Bx0x0q_4 0$$

The full transition sequence is shown below.

$$q_1 000000 \Rightarrow Bq_2 00000 \Rightarrow Bxq_3 0000 \Rightarrow Bx0q_4 000 \Rightarrow Bx0xq_3 00 \Rightarrow Bx0x0q_4 0 \Rightarrow$$
$$Bx0x0xq_3 B \Rightarrow Bx0x0q_5 xB \Rightarrow Bx0xq_5 0xB \Rightarrow Bx0q_5 x0xB \Rightarrow Bxq_5 0x0xB \Rightarrow$$
$$Bq_5 x0x0xB \Rightarrow q_5 Bx0x0xB \Rightarrow Bq_2 x0x0xB \Rightarrow Bxq_2 0x0xB \Rightarrow Bxxq_3 x0xB \Rightarrow$$
$$Bxxxq_3 0xB \Rightarrow Bxxx0q_4 xB \Rightarrow Bxxx0xq_4 B \Rightarrow Bxxx0xBq_{reject}$$

Since the machine is at state $q_{reject}$ after the last transition, we conclude that input 000000 is rejected by the TM in question.

## P.3 ➡ *Solution*

**Problem 3.1:** We begin with the state-input string $q_1$1#1. The TM begins by reading a 1, which causes it to move from $q_1$ to $q_3$, displaces the tape to the right, and writes an $x$. That is,

$$q_1 1\#1 \Rightarrow xq_3 \#1$$

At state $q_3$, the machine reads a #, transitions to $q_5$, displaces the tape to the right, and writes nothing. Accordingly,

$$q_1 1\#1 \Rightarrow xq_3 \#1 \Rightarrow x\#q_5 1$$

At state $q_5$, the machine reads a 1, transitions to $q_6$, displaces the tape to the left, and writes an $x$. It follows that

$$q_1 1\#1 \Rightarrow xq_3 \#1 \Rightarrow x\#q_5 1 \Rightarrow xq_6 \#x$$

At state $q_6$, the machine reads a #, transitions to $q_7$, displaces the tape to the left, and writes nothing. Thus,

$$q_1 1\#1 \Rightarrow xq_3 \#1 \Rightarrow x\#q_5 1 \Rightarrow xq_6 \#x \Rightarrow q_7 x\#x$$

At state $q_7$, the machine reads a $x$, transitions to $q_1$, displaces the tape to the right, and writes nothing. Thus,

$$q_1 1\#1 \Rightarrow xq_3 \#1 \Rightarrow x\#q_5 1 \Rightarrow xq_6 \#x \Rightarrow q_7 x\#x \Rightarrow xq_1 \#x$$

At state $q_1$, the machine reads a #, transitions to $q_8$, displaces the tape to the right, and writes nothing. Thus,

$$q_1 1\#1 \Rightarrow xq_3 \#1 \Rightarrow x\#q_5 1 \Rightarrow xq_6 \#x \Rightarrow q_7 x\#x \Rightarrow xq_1 \#x \Rightarrow x\#q_8 x$$

At state $q_8$, the machine reads an $x$, stays in the same state, displaces the tape to the right, and writes nothing. Thus,

$$q_1 1\#1 \Rightarrow xq_3 \#1 \Rightarrow x\#q_5 1 \Rightarrow xq_6 \#x \Rightarrow q_7 x\#x \Rightarrow xq_1 \#x \Rightarrow x\#q_8 x \Rightarrow x\#xq_8 B$$

At state $q_8$, the machine reads a $B$, transitions to $q_{accept}$, displaces the tape to the right, and writes nothing. Thus,

$$q_1 1\#1 \Rightarrow xq_3\#1 \Rightarrow x\#q_5 1 \Rightarrow xq_6\#x \Rightarrow q_7 x\#x \Rightarrow xq_1\#x \Rightarrow x\#q_8 x \Rightarrow x\#xq_8 B \Rightarrow x\#xBq_{accept}$$

At this point, the accept state $q_{accept}$ has been reached. Input 1#1 is accepted by the TM at hand.

**Problem 3.2:** We start with the state-input 2-tuple $q_1 10\#11$. The TM begins by reading a 1, which causes it to move from $q_1$ to $q_3$, displaces the tape to the right, and writes an $x$. That is,

$$q_1 10\#11 \Rightarrow xq_3 0\#11$$

At state $q_3$, the machine reads a 0, stays in the same state, displaces the tape to the right, and writes nothing. Accordingly,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11$$

At state $q_3$, the machine reads a #, changes to state $q_5$, displaces the tape to the right, and writes nothing. It follows that

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11$$

At state $q_5$, the machine reads a 1, changes to $q_6$, displaces the tape to the left, and writes an $x$. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1$$

At state $q_6$, the machine reads a #, changes to state $q_7$, displaces the tape to the left, and writes nothing. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1$$

At state $q_7$, the machine reads a 0, stays in the same state, displaces the tape to the left, and writes nothing. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1 \Rightarrow q_7 x0\#x1$$

At state $q_7$, the machine reads an $x$, changes to state $q_1$, displaces the tape to the right, and writes nothing. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1 \Rightarrow q_7 x0\#x1 \Rightarrow xq_1 0\#x1$$

At state $q_1$, the machine reads a 0, changes to state $q_2$, displaces the tape to the right, and writes an $x$. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1 \Rightarrow q_7 x0\#x1 \Rightarrow xq_1 0\#x1 \Rightarrow xxq_2\#x1$$

At state $q_2$, the machine reads a #, changes to state $q_4$, displaces the tape to the right, and writes nothing. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1 \Rightarrow q_7 x0\#x1 \Rightarrow xq_1 0\#x1 \Rightarrow xxq_2\#x1 \Rightarrow xx\#q_4 x1$$

At state $q_4$, the machine reads an $x$, remains in state $q_4$, displaces the tape to the right, and writes nothing. Thus,

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1 \Rightarrow q_7 x0\#x1 \Rightarrow xq_1 0\#x1 \Rightarrow xxq_2\#x1 \Rightarrow xx\#q_4 x1 \Rightarrow xx\#xq_4 1$$

At state $q_4$, the machine would read a 1, but there's no transition for such an input in the diagram. Thus, the machine moves to state $q_{reject}$.

$$q_1 10\#11 \Rightarrow xq_3 0\#11 \Rightarrow x0q_3\#11 \Rightarrow x0\#q_5 11 \Rightarrow x0q_6\#x1 \Rightarrow xq_7 0\#x1 \Rightarrow q_7 x0\#x1 \Rightarrow xq_1 0\#x1 \Rightarrow xxq_2\#x1 \Rightarrow xx\#q_4 x1 \Rightarrow xx\#xq_4 1 \Rightarrow xx\#x1q_{reject}$$

String 10#11 is rejected by the TM.

## P.4 ➡ *Solution*

Starting at $s$, the machine may read an input $B$ to transition to $q_1$ and "write" a $B$. Also, the tape is displaced to the left. It follows that the table entry for row $s$ and column $B$ is $(q_1, B, L)$, as shown.

| Present state | 0 | 1 | $B$ |
|---|---|---|---|
| $\to s$ | | | $q_1, B, L$ |
| $q_1$ | | | |
| $q_2$ | | | |
| $q_3$ | | | |
| $q_4$ | | | |
| $q_5$ | | | |
| $q_6$ | | | |
| $H$ | | | |

Once at $q_1$, the machine may read a 0, change to state $q_2$, and "write" a $B$. The tape is then displaced to the left. The entry for row $q_1$ and column 0 is $(q_2, B, L)$, as shown.

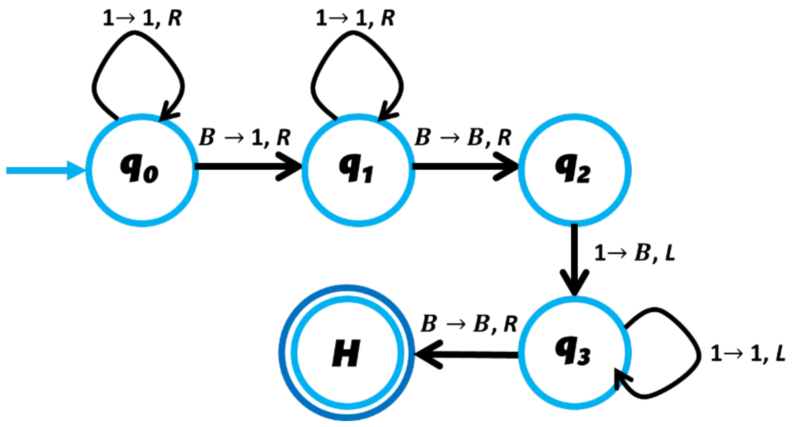
| Present state | 0 | 1 | $B$ |
|---|---|---|---|
| $\to s$ | | | $q_1, B, L$ |
| $q_1$ | $q_2, B, L$ | | |
| $q_2$ | | | |
| $q_3$ | | | |
| $q_4$ | | | |
| $q_5$ | | | |
| $q_6$ | | | |
| $H$ | | | |

The full table is shown in continuation.

| Present state | 0 | 1 | $B$ |
|---|---|---|---|
| $\to s$ | – | – | $q_1, B, L$ |
| $q_1$ | $q_2, B, L$ | $q_4, B, L$ | $H, B, R$ |
| $q_2$ | $q_2, 0, L$ | $q_2, 1, L$ | $q_3, B, R$ |
| $q_3$ | $q_6, B, R$ | – | – |
| $q_4$ | $q_4, 0, L$ | $q_4, 0, L$ | $q_5, B, R$ |
| $q_5$ | – | $q_6, B, R$ | – |
| $q_6$ | $0, q_6, R$ | $1, q_6, R$ | – |
| $H$ | – | – | – |

## P.5 ➡ *Solution*

The transition diagram in question is shown below.




## P.6 ➡ *Solution*

**Problem 6.1:** On input string $w$, the machine works as follows.
*1.* Scan the tape and mark the first 0 which has not been marked. If there is no unmarked 0, proceed to stage 5.
*2.* Continue scanning and mark the next unmarked 0. If there are not any on the tape, *reject*. Otherwise, move the head to the front of the tape.
*3.* Scan the tape and mark the first 1 which has not been marked. If there is no unmarked 1, *reject*.
*4.* Move the head to the front of the tape and repeat stage 1.
*5.* Move the head to the front of the tape. Scan the tape for any unmarked 1's. If there are none, *accept*. Otherwise, *reject*.

**Problem 6.2:** On input string $w$, the machine works as follows.
*1.* Scan the tape and mark the first 0 which has not been marked. If there is no unmarked 0, go to stage 5.
*2.* Continue scanning and mark the next unmarked 0. If there are not any on the tape, *accept*. Otherwise, move the head to the front of the tape.

3. Scan the tape and mark the first 1 which has not been marked. If there is no unmarked 1, *accept*.
4. Move the head to the front of the tape and repeat stage 1.
5. Move the head to the front of the tape. Scan the tape for any unmarked 1's. If there are none, *reject*. Otherwise, *accept*.

Note that the machine in this part works similarly to the one in the previous part, only with the opposite (complementary) positioning of *accept* and *reject* decisions.

## P.7 ➡ Solution

The language $L$ in question can be described with the Turing machine $M$ such that $M = \{Q, \Sigma, \Gamma, q_0, q_{\text{accept}}, q_{\text{reject}}\}$, where $Q = \{q_0, q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1, X, Y\}$, $\Gamma = \{0, 1, X, Y, B\}$, and $q_0$ as the initial state. The accept state $q_{\text{accept}}$ is $q_f$, and the reject state set $q_{\text{reject}}$ is any state $\in Q$ other than $q_f$. The transition function of the TM is described by the following table.

| $\delta$ | 0 | 1 | $B$ | X | Y |
|---|---|---|---|---|---|
| $q_0$ | $q_2, X, R$ | $q_1, X, R$ | $q_f, B, R$ | — | $q_0, Y, R$ |
| $q_1$ | $q_3, Y, L$ | $q_1, 1, R$ | — | — | $q_1, Y, R$ |
| $q_2$ | $q_2, 0, R$ | $q_3, Y, L$ | — | — | $q_2, Y, R$ |
| $q_3$ | $q_3, 0, L$ | $q_3, 1, L$ | — | $q_0, X, R$ | $q_3, Y, L$ |
| $q_f$ | — | — | — | — | — |

The TM works as follows. It makes repeated excursions back and forth along the tape. Symbols $X$ and $Y$ are used to replace 0's and 1's that have been cancelled one against another. The difference is that an $X$ guarantees that there are no unmatched 0's and 1's to its left (so the head never moves left of an $X$), while a $Y$ may have 0's or 1's to its left.

Initially in state $q_0$, the Turing machine picks up a 0 or 1, remembering it in state ($q_1$ = found a 1; $q_2$ = found a 0), and cancels what it found with an $X$. As an exception, if the TM sees the blank state $q_0$, then all 0's and 1's have matched, so the input is accepted by going to state $q_f$.

In state $q_1$, the TM moves right, looking for a 0. If it finds it, the 0 is replaced with $Y$, and the TM enters state $q_3$ to move left and look for an $X$. Similarly, state $q_2$ looks for a 1 to match against a 0.

In state $q_3$, the TM moves left until it finds the rightmost $X$. At that point, it enters state $q_0$ again, moving right over $Y$'s until it finds a 0, 1, or blank, and the cycle begins again.

## P.8 ➡ Solution

**Problem 8.1:** The single-tape TM used in the simulation would have $2k = 2 \times 5 = 10$ tracks. Five of the tracks hold one of the symbols from the tape alphabet, so there are $7^5 \approx 16{,}800$ ways to select these tracks. The other five tracks hold either $X$ or blank, so these tracks can be selected in $2^5 = 32$ ways. The total number of symbols is thus $7^5 \times 2^5 \approx 538{,}000$.

**Problem 8.2:** The number of symbols is the same. The five tracks with tape symbols can still be chosen in $7^5$ ways. The sixth track has to tell which subset of the five tapes have their head at that position. There are $2^5$ possible subsets, and therefore 32 symbols are needed for the 6th track. The number of symbols continues to be $7^5 \times 2^5$.

## P.9 ➡ Solution

The two-tape Turing machine with tape alphabet $\{a, b, B\}$ and accepting state $q_4$ is described in the following table. Using these transition rules, the machine can be made to accept palindromes over $\{a, b\}$.

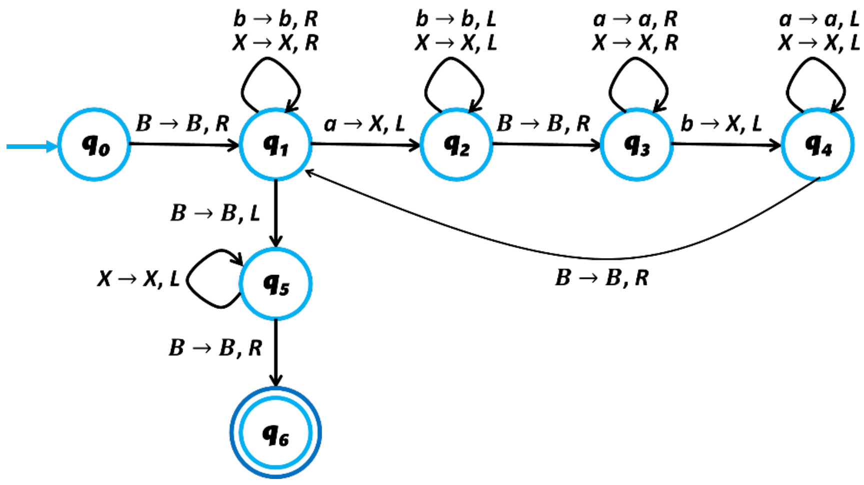
| $\delta$ | Observation |
|---|---|
| $\delta(q_0, B, B) = [q_1; B, R; B, R]$ | |
| $\delta(q_1, x, B) = [q_1; x, R; x, R]$ | $\forall\, x \in \{a, b\}$ |
| $\delta(q_1, B, B) = [q_2; B, L; B, S]$ | |
| $\delta(q_2, x, B) = [q_2; x, L; B, S]$ | $\forall\, x \in \{a, b\}$ |
| $\delta(q_2, B, B) = [q_3; B, R; B, L]$ | |
| $\delta(q_3, x, x) = [q_3; x, R; x, L]$ | $\forall\, x \in \{a, b\}$ |
| $\delta(q_3, B, B) = [q_4; B, S; B, S]$ | |

The machine functions as follows. First, the input is copied onto tape 1 in state $q_1$. State $q_2$ returns the head reading tape 1 to the initial position. With tape head 1 moving left-to-right and head 2 moving right-to-left, the strings on the two tapes are compared. If both heads simultaneously read a blank, the computation terminates in $q_4$.

The maximal number of transitions of a computation with an input string of length $n$ occurs when the string is accepted. Tape head 1 reads right-to-left, left-to-right, and then right-to-left through the input. Each pass requires $n + 1$ transitions. The computation of a string that is not accepted halts when the first mismatch of symbols on tape 1 and tape 2 is discovered. Thus, the maximal number of transitions for an input string of length $n$ is $3(n + 1)$.

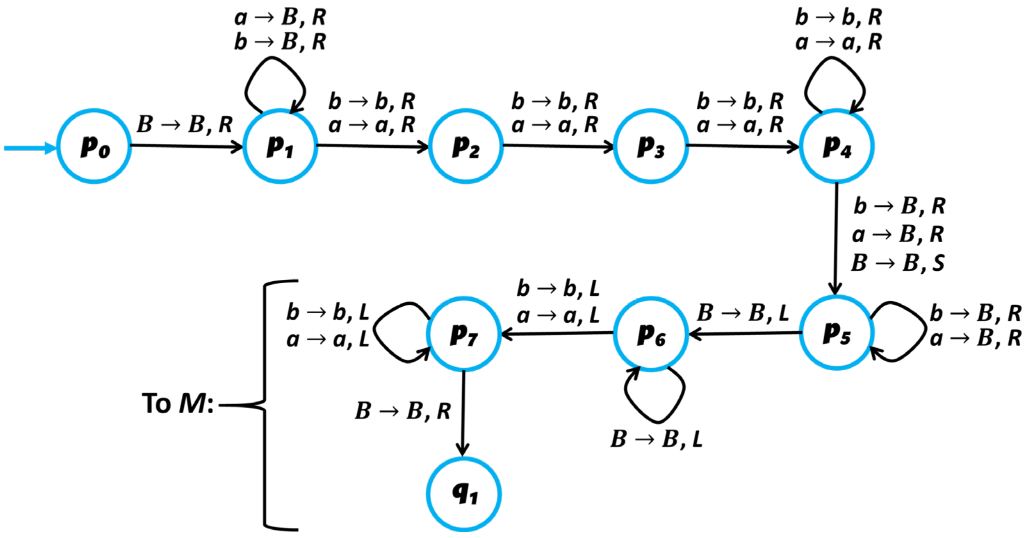
## P.10 ➡ *Solution*

**Problem 10.1:** The diagram for the Turing machine we aim for is shown below.




A computation on this machine begins by finding the first $a$ on the tape and replacing it with an $X$ (state $q_1$). The tape head is then returned to position zero and a search is initiated for a corresponding $b$. If a $b$ is encountered in state $q_3$, and $X$ is written and the tape head is repositioned to repeat the cycle $q_1, q_2, q_3, q_4$. If no matching $b$ is found, the computation halts in state $q_3$, rejecting the input. After all the $a$'s have been processed, the entire string is read in $q_1$ and $q_5$ is entered upon reading the trailing blank. The computation halts in the accepting state $q_6$ if no $b$'s remain on the tape.

**Problem 10.2:** A deterministic machine $M$ was constructed in 10.1 that accepts strings over $\{a, b\}$ with the same number of $a$'s and $b$'s. Using $M$, a composite machine is constructed to determine whether an input string contains a substring of length three or more with the same number of $a$'s and $b$'s. Nondeterminism is used to "choose" a substring to be examined. States $p_0$ to $p_7$ nondeterministically select a string of length three or more from the input.




The transition to state $q_1$ of $M$ begins the computation that checks whether the chosen substring has the same number of $a$'s and $b$'s. The accepting states of the composite machine are the accepting states of $M$.

This problem illustrates two important features in the design of Turing machines. The first is the ability to use existing machines as components in more complex computations. The machine constructed in 10.1 provides the evaluation of a single substring needed in this computation. The second feature is the ability of a nondeterministic design to remove the need for the consideration of all possible substrings. The computation of a deterministic machine to solve this problem must select a substring of the input and decide if it satisfies the conditions. If not, another substring is generated and examined. This process must be repeated until an acceptable substring is found or all substrings have been generated and evaluated.

## REFERENCES


- DU, D.-Z. and KO, K.-I. (2001). *Problem Solving in Automata, Languages, and Complexity*. Hoboken: John Wiley and Sons.
- HOPCROFT, J.E., MOTWANI, R. and ULLMAN, J.D. (2001). *Introduction to Automata Theory*, *Languages, and Computation*. 2nd edition. Upper Saddle River: Addison-Wesley.
- LINZ, P. (2017). *An Introduction to Formal Languages and Automata.* 6th edition. Burlington: Jones and Bartlett Learning.
- SIPSER, M. (2013). *Introduction to the Theory of Computation*. 3rd edition. Boston: Cengage Learning.
- SUDKAMP, T.A. (2006). *Languages and Machines*. 3rd edition. Upper Saddle River: Addison-Wesley.