

# Lab10-Turing Machine

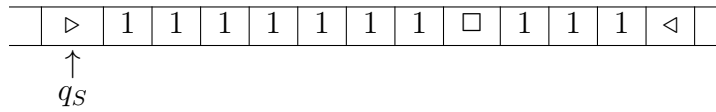
CS214-Algorithm and Complexity, Xiaofeng Gao & Lei Wang, Spring 2021.

\* If there is any problem, please contact TA Yihao Xie.

\* Name: Renyang Guanajuato    Student ID: 519021911058    Email: guanrenyang@sjtu.edu.cn

1. Design a one-tape TM  $M$  that computes the function  $f(x, y) = \lfloor x/y \rfloor$ , where  $x$  and  $y$  are positive integers ( $x > y$ ). The alphabet is  $\{1, 0, \square, \triangleright, \triangleleft\}$ , and the inputs are  $x$  "1"s,  $\square$  and  $y$  "1"s. Below is the initial configuration for input  $x = 7$  and  $y = 3$ . The result  $z = f(x, y)$  should also be represented in the form of  $z$  "1"s on the tape with pattern of  $\triangleright 111 \cdots 111 \triangleleft$ , which is  $\triangleright 11 \triangleleft$  for the example.

Initial Configuration



- (a) Please describe your design and then write the specifications of  $M$  in the form like  $\langle q_s, \triangleright \rangle \rightarrow \langle q_1, \triangleright, R \rangle$ . Explain the transition functions in detail.
- (b) Please draw the state transition diagram.
- (c) Show briefly and clearly the whole process from initial to final configurations for input  $x = 7$  and  $y = 3$ . You may start like this:

$$(q_s, \triangleright 1111111 \square 111 \triangleleft) \vdash (q_1, \triangleright 1111111 \square 111 \triangleleft) \vdash^* (q_1, \triangleright 1111111 \square 111 \triangleleft) \vdash (q_2, \triangleright 1111111 \square 111 \triangleleft)$$

(Note that for simplicity, we write  $(q_1, \triangleright 1111111 \square 111 \triangleleft) \vdash^* (q_1, \triangleright 1111111 \square 111 \triangleleft)$  if the corresponding transaction repeats on multiple inputs with the same state.)

**Solution.**

- (a) **Transition functions:**

$$\begin{aligned}
 &\langle q_s, 1 \rangle \rightarrow \langle q_s, 1, R \rangle & \langle q_s, \triangleright \rangle \rightarrow \langle q_s, \triangleright, R \rangle \\
 &\langle q_s, \square \rangle \rightarrow \langle q_1, \square, R \rangle & \langle q_1, 1 \rangle \rightarrow \langle q_2, 0, L \rangle \\
 &\langle q_1, 0 \rangle \rightarrow \langle q_1, 0, R \rangle & \langle q_1, \square \rangle \rightarrow \langle q_1, \square, R \rangle \\
 &\langle q_1, \triangleleft \rangle \rightarrow \langle q_3, \triangleleft, R \rangle & \langle q_2, 1 \rangle \rightarrow \langle q_1, \square, R \rangle \\
 &\langle q_2, 0 \rangle \rightarrow \langle q_2, 0, L \rangle & \langle q_2, \triangleright \rangle \rightarrow \langle q_5, \square, R \rangle \\
 &\langle q_2, \square \rangle \rightarrow \langle q_2, \square, L \rangle & \langle q_3, 1 \rangle \rightarrow \langle q_3, 1, R \rangle \\
 &\langle q_3, \square \rangle \rightarrow \langle q_4, 1, L \rangle & \langle q_4, 1 \rangle \rightarrow \langle q_4, 1, L \rangle \\
 &\langle q_4, 0 \rangle \rightarrow \langle q_4, 1, L \rangle & \langle q_4, \square \rangle \rightarrow \langle q_1, \square, R \rangle \\
 &\langle q_4, \triangleleft \rangle \rightarrow \langle q_4, \triangleleft, L \rangle & \langle q_5, 1 \rangle \rightarrow \langle q_5, \square, R \rangle \\
 &\langle q_5, 0 \rangle \rightarrow \langle q_5, \square, R \rangle & \langle q_5, \square \rangle \rightarrow \langle q_5, \square, R \rangle \\
 &\langle q_5, \triangleleft \rangle \rightarrow \langle q_6, \triangleright, R \rangle & \langle q_6, 1 \rangle \rightarrow \langle q_6, 1, R \rangle \\
 & & \langle q_6, \square \rangle \rightarrow \langle q_{halt}, \triangleleft, S \rangle
 \end{aligned}$$

**Intuition of the algorithm:**

To compute  $\lfloor x/y \rfloor$ , we do  $x \leftarrow x - y$  and add 1 to the result circularly until  $x < Y$ . Since  $x$  and  $y$  are represented by 1s, each time one bit of  $y$  is read, the bit is set to 0 to mark that this bit has been read. Then we set the corresponding bit of  $x$  to  $\square$  (setting one bit of  $x$  to  $\square$  means  $x$  is subtracted by 1).

**Explanation:**

**State  $q_s$**  scans the input sequence until the separator  $\square$  is found. As a result, the head moves right when it read  $\triangleright$  or 1.

**State  $q_1$**  locates the first 1 in the sequence of  $y$  from the separator  $\square$ , so the head moves right when it reads 0 and  $\square$ . When the head reads 1, it sets the cell to 0 and moves left to set one cell of  $x$  to  $\square$  ( $q_2$ ).

**State  $q_2$**  will set the cell on the right side of  $x$  to  $\square$  after the head reads 1, so the head moves left when it reads  $\square$  and 0. If the head reads  $\triangleleft$  in state  $q_2$ , which means the left 1s of  $x$  is not enough to be divided by  $y$ , the head replaces  $\triangleleft$  with  $\square$  go to state  $q_5$ .

**State  $q_3$**  add 1 to the result of the division after one  $x \leftarrow x - y$ , which is signed by reading  $\triangleleft$  in state  $q_1$ . The result is stored after  $\triangleleft$  so the head will move right when it reads 1 and replaces the first  $\square$  it meet with 1.

**State  $q_4$**  resets  $y$ , currently being signed by 0, to 1. Consequently, the head moves left when it reads 1 and  $\triangleleft$ , and replace 0 with 1 before move left. Then next  $x \leftarrow x - y$  begins and the machine returns to state  $q_1$  when the head reads  $\square$ , which means the sequence of  $y$  has been scanned.

**State  $q_5$**  replace the cells before the result with  $\square$  except for the cell next to the result, which is replaced by  $\triangleright$ . Consequently, the head replaces 1 and 0 with  $\square$  and  $\triangleleft$  with  $\triangleright$  before moving right. The machine goes into state  $q_6$  after modifying  $\triangleleft$ .

**State  $q_6$**  set the cell next to the end of the result sequence to  $\triangleleft$  to ensure the format of the output.

(b) **Transition diagram:**

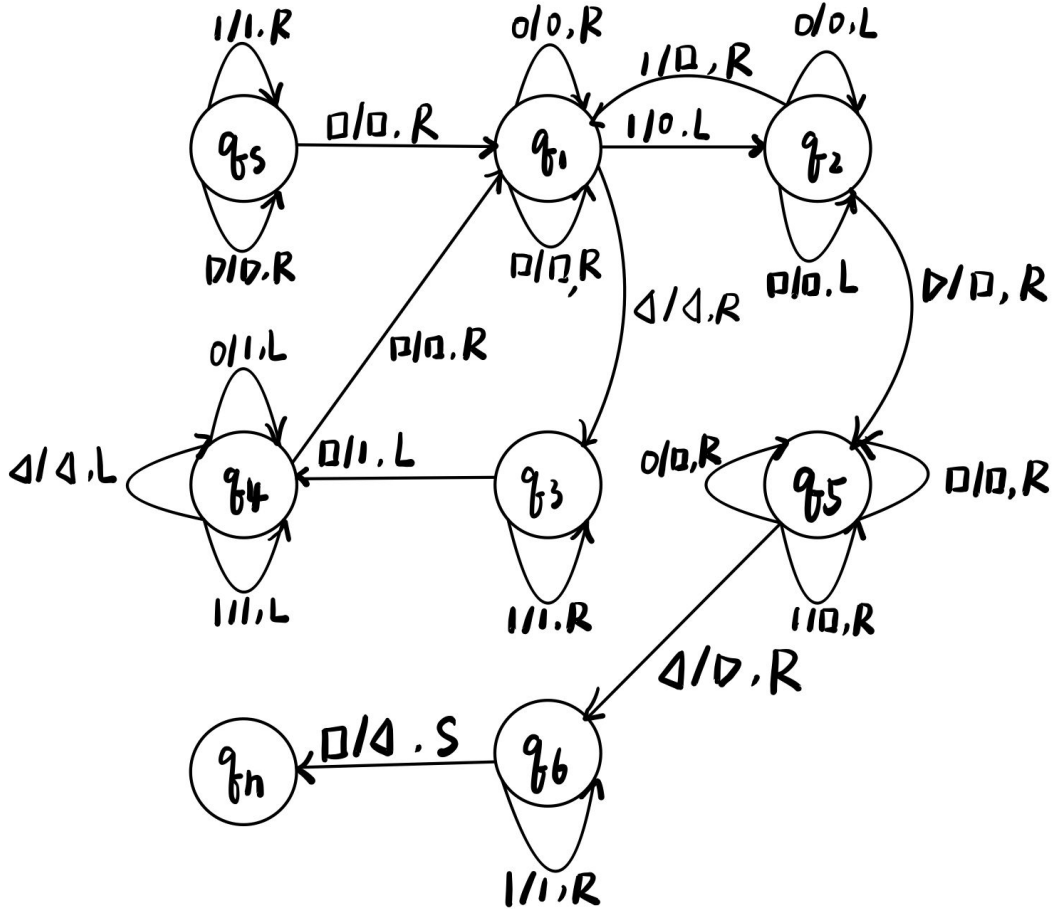


图 1: State transition diagram.

(c) *The whole process:*

$$\begin{aligned}
& (q_s, \sqsupset 1111111 \square 111 \triangleleft) \vdash^* (q_s, \triangleright 1111111 \square 111 \triangleleft) \vdash (q_1, \triangleright 1111111 \square \underline{1} 11 \triangleleft) \\
& \vdash (q_2, \triangleright 1111111 \square 011 \triangleleft) \vdash (q_2, \triangleright 111111 \underline{1} \square 011 \triangleleft) \vdash (q_1, \triangleright 111111 \square \square 011 \triangleleft) \\
& \vdash (q_1, \triangleright 111111 \square \square \underline{0} 11 \triangleleft) \vdash (q_1, \triangleright 111111 \square \square 0 \underline{1} 1 \triangleleft) \vdash^* (q_2, \triangleright 111111 \square \square 001 \triangleleft) \\
& \vdash^* (q_2, \triangleright 111111 \underline{1} \square \square 001 \triangleleft) \vdash (q_1, \triangleright 11111 \square \square \square 001 \triangleleft) \vdash^* (q_1, \triangleright 11111 \square \square \square \underline{0} 01 \triangleleft) \\
& \vdash^* (q_1, \triangleright 11111 \square \square \square 00 \underline{1} \triangleleft) \vdash (q_2, \triangleright 11111 \square \square \square 000 \triangleleft) \vdash^* (q_2, \triangleright 11111 \square \square \square 000 \triangleleft) \\
& \vdash^* (q_2, \triangleright 11111 \underline{1} \square \square \square 000 \triangleleft) \vdash (q_1, \triangleright 1111 \square \square \square \square 000 \triangleleft) \vdash^* (q_1, \triangleright 1111 \square \square \square \square \underline{0} 00 \triangleleft) \\
& \vdash^* (q_1, \triangleright 1111 \square \square \square \square 000 \trianglelefteq) \vdash (q_3, \triangleright 1111 \square \square \square \square 000 \triangleleft \square) \vdash (q_4, \triangleright 1111 \square \square \square \square 000 \trianglelefteq 1) \\
& \vdash (q_4, \triangleright 1111 \square \square \square \square 000 \trianglelefteq 1) \vdash^* (q_4, \triangleright 1111 \square \square \square \square \underline{1} 11 \triangleleft 1) \vdash (q_1, \triangleright 1111 \square \square \square \square \underline{1} 11 \triangleleft 1) \\
& \vdash (q_2, \triangleright 1111 \square \square \square \square \underline{0} 11 \triangleleft 1) \vdash^* (q_2, \triangleright 1111 \square \square \square \square 011 \triangleleft 1) \vdash (q_1, \triangleright 111 \square \square \square \square 011 \triangleleft 1) \\
& \vdash^* (q_1, \triangleright 111 \square \square \square \square \underline{0} 11 \triangleleft 1) \vdash (q_1, \triangleright 111 \square \square \square \square 0 \underline{1} 1 \triangleleft 1) \vdash (q_2, \triangleright 111 \square \square \square \square \underline{0} 01 \triangleleft 1) \\
& \vdash (q_2, \triangleright 111 \square \square \square \square \underline{0} 01 \triangleleft 1) \vdash^* (q_2, \triangleright 11 \square \square \square \square \underline{0} 01 \triangleleft 1) \vdash (q_1, \triangleright 11 \square \square \square \square \underline{0} 01 \triangleleft 1) \\
& \vdash^* (q_1, \triangleright 11 \square \square \square \square \underline{0} 01 \triangleleft 1) \vdash^* (q_1, \triangleright 11 \square \square \square \square \underline{0} 0 \underline{1} \triangleleft 1) \vdash (q_2, \triangleright 11 \square \square \square \square \underline{0} 00 \triangleleft 1) \\
& \vdash^* (q_2, \triangleright 11 \square \square \square \square \underline{0} 000 \triangleleft 1) \vdash^* (q_2, \triangleright 1 \square \square \square \square \underline{0} 000 \triangleleft 1) \vdash (q_1, \triangleright 1 \square \square \square \square \underline{0} 000 \triangleleft 1) \\
& \vdash^* (q_1, \triangleright 1 \square \square \square \square \underline{0} 00 \triangleleft 1) \vdash^* (q_1, \triangleright 1 \square \square \square \square \underline{0} 00 \trianglelefteq 1) \vdash (q_3, \triangleright 1 \square \square \square \square \underline{0} 00 \triangleleft \underline{1}) \\
& \vdash (q_3, \triangleright 1 \square \square \square \square \underline{0} 00 \triangleleft 1 \square) \vdash (q_4, \triangleright 1 \square \square \square \square \underline{0} 00 \triangleleft \underline{1} 1) \vdash (q_4, \triangleright 1 \square \square \square \square \underline{0} 00 \trianglelefteq 11) \\
& \vdash (q_4, \triangleright 1 \square \square \square \square \underline{0} 00 \triangleleft 11) \vdash^* (q_4, \triangleright 1 \square \square \square \square \underline{1} 11 \triangleleft 11) \vdash (q_1, \triangleright 1 \square \square \square \square \underline{1} 11 \triangleleft 11) \\
& \vdash (q_2, \triangleright 1 \square \square \square \square \underline{0} 11 \triangleleft 11) \vdash^* (q_2, \triangleright 1 \square \square \square \square \underline{0} 11 \triangleleft 11) \vdash (q_1, \triangleright \square \square \square \square \underline{0} 11 \triangleleft 11) \\
& \vdash^* (q_1, \triangleright \square \square \square \square \underline{0} 11 \triangleleft 11) \vdash (q_2, \triangleright \square \square \square \square \underline{0} 01 \triangleleft 11) \vdash^* (q_2, \triangleright \square \square \square \square \underline{0} 01 \triangleleft 11) \\
& \vdash (q_5, \square \square \square \square \underline{0} 01 \triangleleft 11) \vdash^* (q_6, \square \square \square \square \square \square \square \square \triangleright \underline{1} 1) \\
& \vdash^* (q_6, \square \square \square \square \square \square \square \square \triangleright 11 \square) \\
& \vdash (q_{halt}, \square \square \square \square \square \square \square \square \triangleright 11 \trianglelefteq)
\end{aligned}$$

□

2. Given the alphabet  $\{1, 0, \square, \triangleright, \triangleleft\}$ , design a time efficient 3-tape TM  $M$  to compute  $f : \{0, 1\}^* \rightarrow \{0, 1\}$  which verifies whether the number of 0 and the number of 1 are the same in an input consisting of only 0's and 1's.  $M$  should output 1 if the numbers are the same, and 0 otherwise. For example, for the input tape  $\triangleright 001101 \triangleleft$ ,  $M$  should output 1

- (a) Please describe your design and then write the specifications of  $M$  in the form like  $\langle q_s, \triangleright, \triangleright, \triangleright \rangle \rightarrow \langle q_1, \triangleright, \triangleright, R, R, S \rangle$ . Explain the transition functions in detail.
- (b) Show the time complexity for one-tape TM  $M'$  to compute the same function  $f$  with  $n$  symbols in the input and give a brief description of such  $M'$ .

**Solution.**

(a) **Transition functions:**

*Start state*

$$\langle q_s, \triangleright, \triangleright, \triangleright \rangle \rightarrow \langle q_c, \triangleright, \triangleright, R, R, R \rangle$$

*Begin to copy*

$$\langle q_c, 1, \square, \square \rangle \rightarrow \langle q_c, 1, \square, R, R, S \rangle$$

$$\langle q_c, 0, \square, \square \rangle \rightarrow \langle q_c, \square, \square, R, S, S \rangle$$

$$\langle q_c, \triangleleft, \square, \square \rangle \rightarrow \langle q_t, \square, \square, L, L, S \rangle$$

*Begin to compare*

$$\langle q_t, 0, 1, \square \rangle \rightarrow \langle q_t, 1, \square, L, L, S \rangle$$

$$\langle q_t, 1, 1, \square \rangle \rightarrow \langle q_t, 1, \square, L, S, S \rangle$$

$$\langle q_t, 0, \triangleright, \square \rangle \rightarrow \langle q_r, \triangleright, 0, S, S, R \rangle$$

$$\langle q_t, \triangleright, \triangleright, \square \rangle \rightarrow \langle q_r, \triangleright, 1, S, S, R \rangle$$

$$\langle q_t, 1, \triangleright, \square \rangle \rightarrow \langle q_r, \triangleright, \square, L, S, S \rangle$$

*Ready to terminate*

$$\langle q_r, 0, \triangleright, \square \rangle \rightarrow \langle q_h, \triangleright, \triangleleft, S, S, S \rangle$$

$$\langle q_r, \triangleright, \triangleright, \square \rangle \rightarrow \langle q_h, \triangleright, \triangleleft, S, S, S \rangle$$

**Explanation:**

**State  $q_s$**  is the starting state where the three heads are reading  $\triangleright$ . Move them right without writing to starting executing the Turing Machine.

**State  $q_c$**  copies all the 1s in the input sequence to the second tape.

**State  $q_t$**  compares the contents of the first tape with that of the second tape. Since the second head always reads 1 before reading  $\triangleright$ , if the first head reads 0, both of them will move to the next position; else if the first head reads 1, the first head will move to the next position while the second one will stay. After many times of such operation, the loop will terminate with reading  $\langle \triangleright, \triangleright \rangle$  or  $\langle 0, \triangleright \rangle$ . The situation of  $\langle 0, \triangleright \rangle$  the number of 0s is bigger than that of 1s while the other situation mean the number of 0s is equal to that of 1S.

**State  $q_r$**  formats the output by adding  $\triangleleft$  to the end of result and transform the state to  $h_{halt}$  to stop executing.

- (b) **Time complexity of  $M'$ :** Assume that the initial problem is computable in time  $T(n)$  by a Turing Machine  $M$  with three tapes, then it is computable in time  $15T(n)^2$  by a single-tape Turing machine  $\tilde{M}$ .

**Brief description:** The machine  $\tilde{M}$  places  $\triangleright$  after the input sequence and then start copying the input to the imaginary input tape. During this process whenever an input symbol is copied it is overwritten by  $\triangleright$ .

The  $n + 2$ -cell and  $n + 3$ -cell indicate the initial head position while the  $n + 1$ -cell is reserved for the input.  $\tilde{M}$  sweeps  $kT(n)$  cells from the  $n + 1$ -cell the right and  $kT(n)$  cell from right to left to update using the transitions of  $M$ .

□

3. Define the corresponding decision or search problem of the following problems and give the "certificate" and "certifier" for each decision problem provided in the subquestions or defined by yourself.
- (a) *3-Dimensional Matching*. Given disjoint sets  $X, Y, Z$  all with the size of  $n$ , and a set  $M \subseteq X \times Y \times Z$ . Is there a subset  $M'$  of  $M$  of size  $n$  where no two elements of  $M'$  agree in any coordinate?
  - (b) *Travelling Salesman Problem*. Given a list of cities and the distances between each pair of cities, find the shortest possible route that visits each city exactly once and returns to the origin city.
  - (c) *Job Sequencing*. Given a set of unit-time jobs, each of which has an integer deadline and a nonnegative penalty for missing the deadline. Does there exist a job sequence that has a total penalty  $w \leq k$ ?

**Solution.**

- (a) It is a **decision problem**, the corresponding search problem is: *Given disjoint sets  $X, Y$ , and  $Z$ , find a subset  $M'$  of  $M$  of size  $n$  where no two elements of  $M'$  agree in any coordinate.*  
**Certificate:** A pair of triplets  $t$  in  $M'$  with at least one of their coordinates agree. Note that such a certificate exists iff the set  $M'$  doesn't exist.  
**Certifier:** Check each pair of triplets of possible  $M'$  to determine whether such a set exists or not.
- (b) It is a **search problem**, the corresponding decision problem is: *Is there a possible route with total distance less than  $k$ , which visits each city exactly once and returns to the origin city.*  
**Certificate:** All the possible travelling routines.  
**Certifier:** If there exists a path satisfying that it visits each city once and returns to the origin city, check if the total distance is less than  $k$ . If the proposition is true, the decision problem is true. Vice versa.
- (c) It is a **decision problem**, the corresponding search problem is: *Given such unit-time jobs, find the job sequence with the least total penalty.*  
**Certificate:** All the possible job sequences.  
**Certifier:** Check each possible job sequence and compute the total penalty, if the total penalty is less or equal than  $k$ , the original proposition is true.

□

**Remark:** Please include your .pdf, .tex files for uploading with standard file names.