

# Lab10-Turing Machine

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

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

\* Name: Zirui Liu    Student ID: 519021910343    Email: L.prime@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

$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$	
↑													
$q_s$	$px$												

- (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):

$$\begin{array}{ll}
 \langle q_s, \triangleright \rangle \rightarrow \langle q_1, \triangleright, R \rangle & \langle q_1, 0 \rangle \rightarrow \langle q_1, 0, R \rangle \\
 \langle q_1, 1 \rangle \rightarrow \langle q_1, 1, R \rangle & \langle q_1, \square \rangle \rightarrow \langle q_1, \square, R \rangle \\
 \langle q_1, \triangleright \rangle \rightarrow \langle q_1, \triangleright, R \rangle & \langle q_1, \triangleleft \rangle \rightarrow \langle q_2, \triangleleft, R \rangle \\
 \langle q_2, 0 \rangle \rightarrow \langle q_2, 0, L \rangle & \langle q_2, 1 \rangle \rightarrow \langle q_3, 0, L \rangle \\
 \langle q_2, \square \rangle \rightarrow \langle q_7, \square, R \rangle & \langle q_7, 0 \rangle \rightarrow \langle q_7, 1, R \rangle \\
 \langle q_7, 1 \rangle \rightarrow \langle q_7, 1, R \rangle & \langle q_7, \triangleleft \rangle \rightarrow \langle q_5, \triangleleft, L \rangle \\
 \langle q_3, 1 \rangle \rightarrow \langle q_3, 1, L \rangle & \langle q_3, \square \rangle \rightarrow \langle q_4, \square, L \rangle \\
 \langle q_4, 0 \rangle \rightarrow \langle q_4, 0, L \rangle & \langle q_4, 1 \rangle \rightarrow \langle q_1, 0, R \rangle \\
 \langle q_4, \triangleright \rangle \rightarrow \langle q_8, \triangleleft, R \rangle & \langle q_5, 0 \rangle \rightarrow \langle q_5, 0, L \rangle \\
 \langle q_5, 1 \rangle \rightarrow \langle q_5, 1, L \rangle & \langle q_5, \square \rangle \rightarrow \langle q_5, \square, L \rangle \\
 \langle q_5, \triangleright \rangle \rightarrow \langle q_6, \triangleright, L \rangle & \langle q_6, 0 \rangle \rightarrow \langle q_6, 0, L \rangle \\
 \langle q_6, 1 \rangle \rightarrow \langle q_6, 1, L \rangle & \langle q_6, \square \rangle \rightarrow \langle q_1, 1, R \rangle \\
 \langle q_8, 0 \rangle \rightarrow \langle q_8, \square, R \rangle & \langle q_8, 1 \rangle \rightarrow \langle q_8, \square, R \rangle \\
 \langle q_8, \square \rangle \rightarrow \langle q_8, \square, R \rangle & \langle q_8, \triangleleft \rangle \rightarrow \langle q_9, \square, L \rangle \\
 \langle q_9, \square \rangle \rightarrow \langle q_9, \square, L \rangle & \langle q_9, \triangleleft \rangle \rightarrow \langle q_{10}, \triangleleft, L \rangle \\
 \langle q_{10}, 1 \rangle \rightarrow \langle q_{10}, 1, L \rangle & \langle q_{10}, \square \rangle \rightarrow \langle q_e, \triangleright, L \rangle
 \end{array}$$

(b):

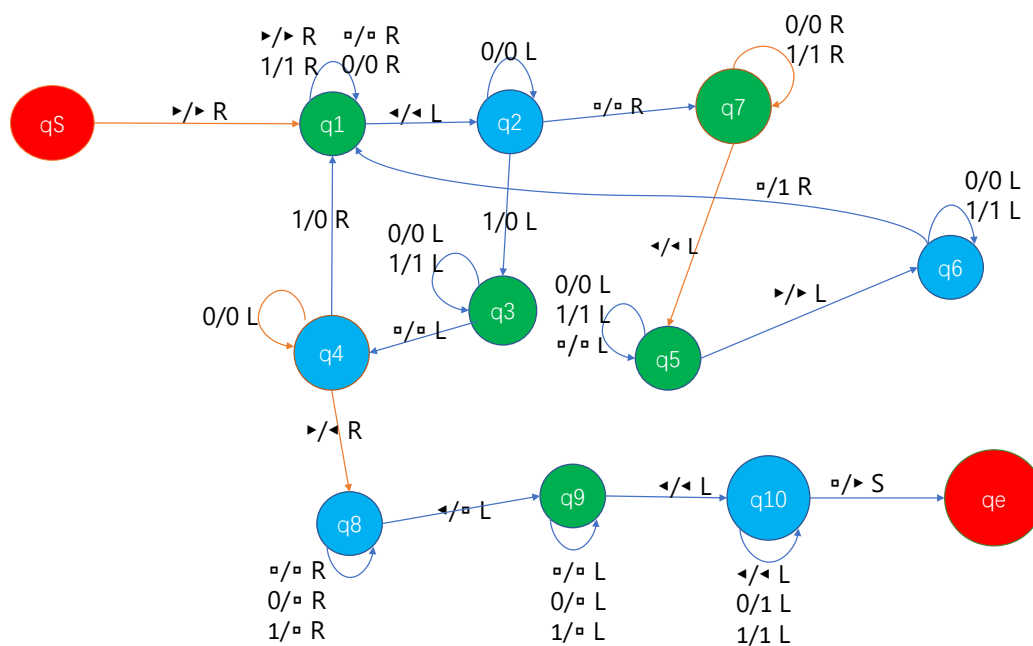


图 1: Turing Machine

(c):

$$\begin{array}{l}
(q_s, \triangleright 1111111 \square 111 \triangleleft) \\
\vdash (q_1, \triangleright 1111111 \square \underline{111} \triangleleft) \\
\vdash (q_3, \triangleright 1111111 \square 1 \underline{10} \triangleleft) \\
\vdash (q_1, \triangleright 1111110 \square 110 \triangleleft) \\
\vdash (q_2, \triangleright 1111110 \square 11 \underline{0} \triangleleft) \\
\vdash (q_3, \triangleright 1111110 \square 100 \triangleleft) \\
\vdash (q_1, \triangleright 111110 \underline{0} \square 100 \triangleleft) \\
\vdash (q_2, \triangleright 1111100 \square \underline{100} \triangleleft) \\
\vdash (q_4, \triangleright 1111 \underline{100} \square 000 \triangleleft) \\
\vdash (q_2, \triangleright 1111000 \square 00 \underline{0} \triangleleft) \\
\vdash (q_7, \triangleright 1111000 \square 1 \underline{00} \triangleleft) \\
\vdash (q_5, \triangleright 1111000 \square 11 \underline{1} \triangleleft) \\
\vdash (q_6, \square \square \square \triangleright 1111000 \square 111 \triangleleft) \\
\vdash (q_2, \square \square 1 \triangleright 1111000 \square 11 \underline{1} \triangleleft) \\
\vdash (q_4, \square \square 1 \triangleright 111100 \underline{0} \square 110 \triangleleft) \\
\vdash (q_1, \square \square 1 \triangleright 1110000 \square 11 \underline{0} \triangleleft) \\
\vdash (q_3, \square \square 1 \triangleright 1110000 \square \underline{100} \triangleleft) \\
\vdash (q_4, \square \square 1 \triangleright 11 \underline{10000} \square 100 \triangleleft) \\
\vdash (q_2, \square \square 1 \triangleright 1100000 \square 10 \underline{0} \triangleleft) \\
\vdash (q_4, \square \square 1 \triangleright 110000 \underline{0} \square 000 \triangleleft) \\
\vdash (q_1, \square \square 1 \triangleright 1000000 \square 000 \underline{\triangleleft}) \\
\vdash (q_7, \square \square 1 \triangleright 1000000 \square \underline{000} \triangleleft) \\
\vdash (q_5, \square \square 1 \triangleright 1000000 \square 111 \triangleleft) \\
\vdash (q_1, \square 1 \underline{1} \triangleright 1000000 \square 111 \triangleleft) \\
\vdash (q_3, \square 1 \underline{1} \triangleright 1000000 \square 1 \underline{10} \triangleleft) \\
\vdash (q_4, \square 1 \underline{1} \triangleright \underline{1000000} \square 110 \triangleleft) \\
\vdash (q_2, \square 1 \underline{1} \triangleright 0000000 \square 11 \underline{0} \triangleleft) \\
\vdash (q_3, \square 1 \underline{1} \triangleright 0000000 \square \underline{100} \triangleleft) \\
\vdash (q_8, \square 1 \triangleleft \underline{0000000} \square 100 \triangleleft) \\
\vdash (q_8, \square 1 \triangleleft \square \square \square \square \square \square \square \square \square \triangleleft) \vdash (q_9, \square 1 \triangleleft \square \square \square \square \square \square \square \square \square \triangleleft) \vdash (q_9, \square 1 \triangleleft) \\
\vdash (q_{10}, \square 1 \underline{\triangleleft})
\end{array}
\quad
\begin{array}{l}
\vdash (q_1, \triangleright \underline{1111111} \square 111 \triangleleft) \vdash (q_1, \triangleright 1111111 \square \underline{111} \triangleleft) \\
\vdash (q_1, \triangleright 1111111 \square 11 \underline{1} \triangleleft) \vdash (q_2, \triangleright 1111111 \square 11 \underline{1} \triangleleft) \\
\vdash (q_3, \triangleright 1111111 \square \underline{110} \triangleleft) \vdash (q_4, \triangleright 111111 \underline{1} \square 110 \triangleleft) \\
\vdash (q_1, \triangleright 1111110 \square 11 \underline{0} \triangleleft) \vdash (q_1, \triangleright 1111110 \square 110 \underline{\triangleleft}) \\
\vdash (q_2, \triangleright 1111110 \square 11 \underline{0} \triangleleft) \vdash (q_3, \triangleright 1111110 \square 1 \underline{00} \triangleleft) \\
\vdash (q_4, \triangleright 111111 \underline{0} \square 100 \triangleleft) \vdash (q_4, \triangleright 11111 \underline{10} \square 100 \triangleleft) \\
\vdash (q_1, \triangleright 1111100 \square 100 \underline{\triangleleft}) \vdash (q_2, \triangleright 1111100 \square 100 \underline{\triangleleft}) \\
\vdash (q_3, \triangleright 1111100 \square \underline{000} \triangleleft) \vdash (q_4, \triangleright 11111 \underline{00} \square 000 \triangleleft) \\
\vdash (q_1, \triangleright 11110 \underline{00} \square 000 \triangleleft) \vdash (q_1, \triangleright 1111000 \square 000 \underline{\triangleleft}) \\
\vdash (q_2, \triangleright 1111000 \square \underline{000} \triangleleft) \vdash (q_7, \triangleright 1111000 \square \underline{000} \triangleleft) \\
\vdash (q_7, \triangleright 1111000 \square 11 \underline{0} \triangleleft) \vdash (q_7, \triangleright 1111000 \square 11 \underline{\triangleleft}) \\
\vdash (q_5, \triangleright 1111000 \square \underline{111} \triangleleft) \vdash (q_5, \square \square \square \triangleright 1111000 \square 111 \triangleleft) \\
\vdash (q_1, \square \square 1 \triangleright 1111000 \square 111 \triangleleft) \vdash (q_1, \square \square 1 \triangleright 1111000 \square 11 \underline{\triangleleft}) \\
\vdash (q_3, \square \square 1 \triangleright 1111000 \square 1 \underline{10} \triangleleft) \vdash (q_3, \square \square 1 \triangleright 1111000 \square 110 \underline{\triangleleft}) \\
\vdash (q_4, \square \square 1 \triangleright 111 \underline{1000} \square 110 \triangleleft) \vdash (q_1, \square \square 1 \triangleright 1110 \underline{000} \square 110 \triangleleft) \\
\vdash (q_2, \square \square 1 \triangleright 1110000 \square 11 \underline{0} \triangleleft) \vdash (q_2, \square \square 1 \triangleright 1110000 \square 1 \underline{10} \triangleleft) \\
\vdash (q_3, \square \square 1 \triangleright 1110000 \square \underline{100} \triangleleft) \vdash (q_4, \square \square 1 \triangleright 111000 \underline{0} \square 100 \triangleleft) \\
\vdash (q_1, \square \square 1 \triangleright 110 \underline{0000} \square 100 \triangleleft) \vdash (q_1, \square \square 1 \triangleright 1100000 \square 100 \underline{\triangleleft}) \\
\vdash (q_2, \square \square 1 \triangleright 1100000 \square \underline{100} \triangleleft) \vdash (q_3, \square \square 1 \triangleright 1100000 \square \underline{000} \triangleleft) \\
\vdash (q_4, \square \square 1 \triangleright 1 \underline{100000} \square 000 \triangleleft) \vdash (q_1, \square \square 1 \triangleright 10 \underline{00000} \square 000 \triangleleft) \\
\vdash (q_2, \square \square 1 \triangleright 1000000 \square 00 \underline{0} \triangleleft) \vdash (q_2, \square \square 1 \triangleright 1000000 \square \underline{000} \triangleleft) \\
\vdash (q_7, \square \square 1 \triangleright 1000000 \square 11 \underline{\triangleleft}) \vdash (q_5, \square \square 1 \triangleright 1000000 \square 11 \underline{\triangleleft}) \\
\vdash (q_6, \square \square 1 \triangleright 1000000 \square 11 \underline{\triangleleft}) \vdash (q_6, \square \square 1 \triangleright 1000000 \square 11 \underline{\triangleleft}) \\
\vdash (q_1, \square 1 \underline{1} \triangleright 1000000 \square 11 \underline{\triangleleft}) \vdash (q_2, \square 1 \underline{1} \triangleright 1000000 \square 11 \underline{\triangleleft}) \\
\vdash (q_3, \square 1 \underline{1} \triangleright 1000000 \square \underline{110} \triangleleft) \vdash (q_4, \square 1 \underline{1} \triangleright 100000 \underline{0} \square 110 \triangleleft) \\
\vdash (q_1, \square 1 \underline{1} \triangleright \underline{0000000} \square 110 \triangleleft) \vdash (q_1, \square 1 \underline{1} \triangleright 0000000 \square 110 \underline{\triangleleft}) \\
\vdash (q_2, \square 1 \underline{1} \triangleright 0000000 \square 1 \underline{10} \triangleleft) \vdash (q_3, \square 1 \underline{1} \triangleright 0000000 \square \underline{100} \triangleleft) \\
\vdash (q_4, \square 1 \underline{1} \triangleright 000000 \underline{0} \square 100 \triangleleft) \vdash (q_4, \square 1 \underline{1} \triangleright 00000000 \square 100 \triangleleft) \\
\vdash (q_8, \square 1 \triangleleft \square \underline{0000000} \square 100 \triangleleft) \vdash (q_8, \square 1 \triangleleft \square \square \square \square \square \square \square \square \square \underline{100} \triangleleft) \\
\vdash (q_{10}, \underline{\triangleleft} 1 \triangleleft) \vdash (q_e, \triangleright 1 \triangleleft)
\end{array}$$

□

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

1. 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.
2. 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):

**Begin To Copy**

$$\begin{aligned}\langle q_s, \triangleright, \square, \square \rangle &\rightarrow \langle q_1, \triangleright, \square, R, R, S \rangle \\ \langle q_1, 0, \square, \square \rangle &\rightarrow \langle q_1, 0, \square, R, S, S \rangle \\ \langle q_1, 1, \square, \square \rangle &\rightarrow \langle q_1, \square, 1, R, S, R \rangle \\ \langle q_1, \triangleleft, \square, \square \rangle &\rightarrow \langle q_2, \triangleleft, \square, S, L, L \rangle\end{aligned}$$

**Check Equal Or Not**

$$\begin{aligned}\langle q_2, \triangleleft, 0, 1 \rangle &\rightarrow \langle q_2, 0, \square, S, L, L \rangle \\ \langle q_2, \triangleleft, 0, \square \rangle &\rightarrow \langle q_e, 0, 0, S, S, S \rangle \\ \langle q_2, \triangleleft, \triangleright, \square \rangle &\rightarrow \langle q_e, \triangleright, 1, S, S, S \rangle\end{aligned}$$

**Output**

$$\begin{aligned}\langle q_2, \triangleleft, \triangleright, 1 \rangle &\rightarrow \langle q_3, \triangleright, \square, S, S, L \rangle \\ \langle q_3, \triangleleft, \triangleright, 1 \rangle &\rightarrow \langle q_3, \triangleright, \square, S, S, L \rangle \\ \langle q_3, \triangleleft, \triangleright, \square \rangle &\rightarrow \langle q_e, \triangleright, 0, S, S, S \rangle\end{aligned}$$

(b):

We know that if  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  is computable in time  $T(n)$  by using  $k$  tapes, then it is computable in time  $5kT(n)^2$  using one tape TM. In problem *a*, we have  $k = 3$  and  $T(n) = O(n)$ , so we can know that the time complexity for one tape TM is  $O(n^2)$ .

For such  $M'$ , we thought that it is proper to match every 0 and 1. The head should scan round and round and every time the head matches one 0 and one 1, it should turn both of them into  $\square$ . If the head can't match such one 0 and one 1 (when the tape is not empty) then we knew it won't match, so we will output 0. On the other hand, it would output 1. The TM will go through at most  $n$  rounds, and each round costs  $n$  movements, For the fact that one tape TM has nearly the same complexity as multi tape TM, so in conclusion, the time complexity is  $O(n^2)$ . □

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.

1. *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?
2. *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.
3. *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):

**Certificate:** Give permutation for all the three dimension tuple.

**Certifier:** Check for all the permutation, does it satisfy the condition of no two elements having the same coordinate.

(b):

**Certificate:** A permutation of the  $n$  nodes.

**Certifier:** Check that the permutation contains each node in  $V$  exactly once(except the start point, which also being the end point,would be contained twice) , and that there is an edge between each pair of adjacent nodes in the permutation to form a circle.

(c):

**Certificate:** We can apply the greedy algorithm. First we sort all the jobs in decreasing order of their penalties so that minimum of penalties will be charged.

**Certifier:** Using contradiction, it is easy to prove the optimality of the greedy algorithm.

□

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