Lab10-Turing Machine

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

* Name: Haoyi You Student ID: 519030910193 Email: yuri-you@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 $\{1, 0, \square, \rhd, \lhd\}$, 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 $\rhd 111 \cdots 111 \lhd$, which is $\rhd 11 \lhd$ for the example.

Initial Configuration

	$\rhd$	1	1	1	1	1	1	1	$\square$	1	1	1	$\lhd$
	$\uparrow$ q_S												

(a) Please describe your design and then write the specifications of M in the form like $\langle q_S, \rhd \rangle \to \langle q_1, \rhd, R \rangle$. 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:

$(q_s, \underline{\rhd}1111111\square111\lhd) \vdash (q_1, \rhd\underline{1}111111\square111\lhd) \vdash^* (q_1, \rhd1111111\underline{\square}111\lhd) \vdash (q_2, \rhd1111111\square\underline{1}11\lhd)$

Solution. (a)

$$\begin{aligned}
< q_s, \rhd > &\to < q_b, \rhd, R > \\
< q_b, (1, 0, \square) > &\to < q_b, same, R > \\
< q_b, \lhd > &\to < q_c, \lhd, R > \\
< q_c, \square > &\to < q_1, \rhd, L > \\
< q_1, 1 > &\to < q_L, 0, L > \\
< q_1, 0, \lhd > &\to < q_1, same, L > \\
< q_1, \square > &\to < q_r, \square, L > \\
< q_L, (1, 0, \square) > &\to < q_L, same, L > \\
< q_L, \rhd > &\to < q_2, \rhd, R > \\
< q_2, 0 > &\to < q_2, 0, R > \\
< q_2, 1 > &\to < q_R, 0, R > \\
< q_2, \square > &\to < q_f, \square, R > \\
< q_2, \lhd > &\to < q_1, \lhd, R > \\
< q_R, \lhd > &\to < q_1, \lhd, L > \\
< q_R, (1, 0, \square) > &\to < q_L, same, L > \\
< q_r, 0 > &\to < q_r, 1, R > \\
< q_r, (1, 0, \rhd, \lhd) > &\to < q_r, same, R > \\
< q_r, \square > &\to < q_3, 1, L > \\
< q_3, 1 > &\to < q_3, 1, L > \\
< q_3, \rhd > &\to < q_1, \rhd, L > \\
< q_f, (1, 0, \rhd, \lhd) > &\to < q_f, same, R > \\
< q_f, \square > &\to < q_H, \lhd, S >
\end{aligned} \tag{1}$$

q_s start Turing machine
q_b begin to turn right,until arrive at the written block and write the first $\triangleright$
q_1 change one of y from 1 to 0
q_L turn left until the $\triangleright$
q_2 change one of x from 1 to 0
q_R turn right until the $\triangleleft$
q_r rewrite the y to 1 and add an 1 in the end
$q3$ finish adding 1 and turn left to do a new divide operation
q_f end the divide operation and write the last $\triangleleft$ in the end
q_H stop Turing machine.
'same' means do not change the alphabet.

(b) Please refer to figure 1

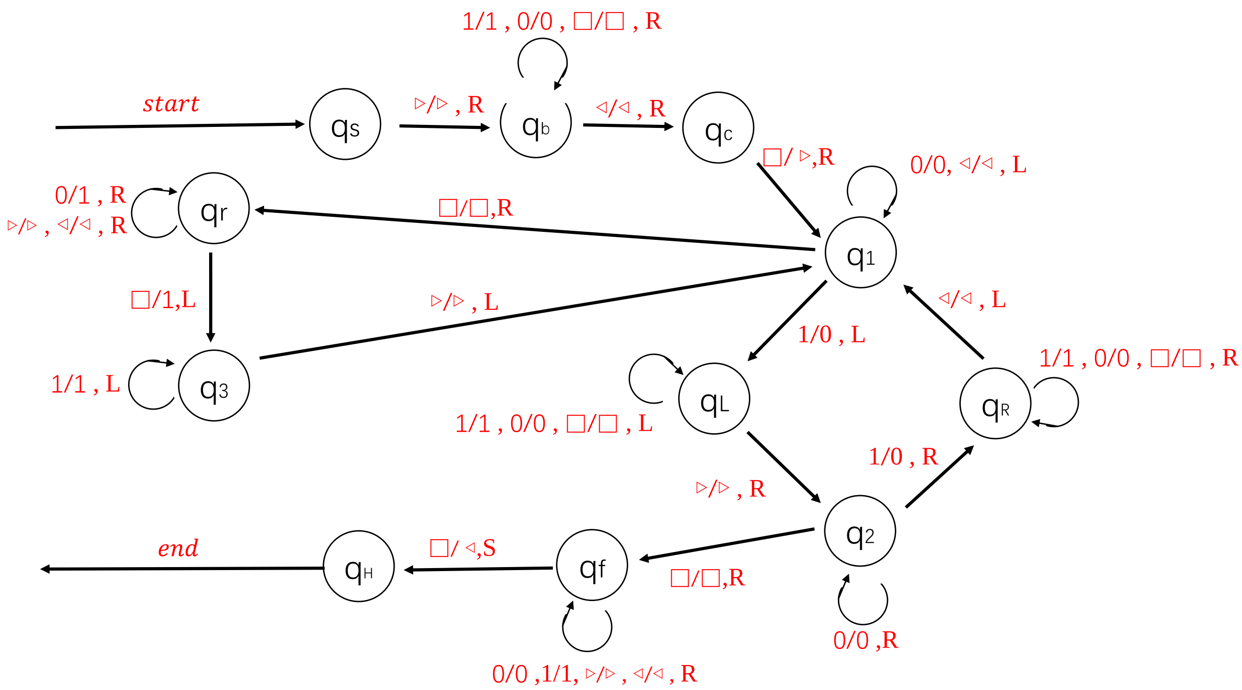

图 1: state transition diagram

(c)

$$
\begin{gathered}
(q_s, \underline{\triangleright}1111111\Box111\triangleleft) \vdash (q_b, \triangleright\underline{1}111111\Box111\triangleleft) \vdash (q_b, \triangleright1111111\Box111\underline{\triangleleft}) \\
\vdash (q_c, \triangleright1111111\Box111\triangleleft\_) \vdash (q_1, \triangleright1111111\Box111\underline{\triangleleft}\triangleright) \vdash (q_1, \triangleright1111111\Box11\underline{1}\triangleleft\triangleright) \\
First\ round \\
\vdash (q_L, \triangleright1111111\Box1\underline{1}0\triangleleft\triangleright) \vdash (q_L, \underline{\triangleright}1111111\Box110\triangleleft\triangleright) \vdash (q_2, \triangleright\underline{1}111111\Box110\triangleleft\triangleright) \\
\vdash (q_R, \triangleright0\underline{1}11111\Box110\triangleleft\triangleright) \vdash (q_R, \triangleright0111111\Box110\underline{\triangleleft}\triangleright) \vdash (q_1, \triangleright0111111\Box11\underline{0}\triangleleft\triangleright) \\
\vdash (q_1, \triangleright0111111\Box1\underline{1}0\triangleleft\triangleright) \vdash (q_L, \triangleright0111111\Box\underline{1}00\triangleleft\triangleright) \vdash (q_L, \underline{\triangleright}0111111\Box100\triangleleft\triangleright) \\
\vdash (q_2, \triangleright\underline{0}111111\Box100\triangleleft\triangleright) \vdash (q_R, \triangleright0\underline{0}11111\Box100\triangleleft\triangleright) \vdash (q_R, \triangleright0011111\Box100\underline{\triangleleft}\triangleright) \\
\vdash (q_1, \triangleright0011111\Box10\underline{0}\triangleleft\triangleright) \vdash (q_1, \triangleright0011111\Box\underline{1}00\triangleleft\triangleright) \vdash (q_L, \triangleright0011111\underline{\Box}000\triangleleft\triangleright) \\
\vdash (q_L, \underline{\triangleright}0011111\Box000\triangleleft\triangleright) \vdash (q_2, \underline{\triangleright}0011111\Box000\triangleleft\triangleright) \vdash (q_2, \triangleright00\underline{1}1111\Box000\triangleleft\triangleright) \\
\vdash (q_R, \triangleright000\underline{1}111\Box000\triangleleft\triangleright) \vdash (q_R, \triangleright0001111\Box000\underline{\triangleleft}\triangleright) \vdash (q_1, \triangleright0001111\Box00\underline{0}\triangleleft\triangleright) \\
\vdash (q_1, \triangleright0001111\underline{\Box}000\triangleleft\triangleright) \vdash (q_r, \triangleright0001111\Box\underline{0}00\triangleleft\triangleright) \vdash (q_r, \triangleright0001111\Box111\underline{\triangleleft}\triangleright) \\
\vdash (q_r, \triangleright0001111\Box111\triangleleft\triangleright\_) \vdash (q_3, \triangleright0001111\Box111\triangleleft\underline{\triangleright}1) \vdash (q_1, \triangleright0001111\Box111\underline{\triangleleft}\triangleright1) \\
Second\ round \\
\vdash (q_L, \triangleright0001111\Box1\underline{1}0\triangleleft\triangleright1) \vdash (q_L, \underline{\triangleright}0001111\Box110\triangleleft\triangleright1) \vdash (q_2, \triangleright\underline{0}0011111\Box110\triangleleft\triangleright1) \\
\ldots\ldots \qquad similar\ to\ the\ First\ round \\
\vdash (q_1, \triangleright0000001\Box00\underline{0}\triangleleft\triangleright1) \vdash (q_1, \triangleright0000001\underline{\Box}000\triangleleft\triangleright1) \vdash (q_r, \triangleright0000001\Box\underline{0}00\triangleleft\triangleright1) \\
\vdash (q_r, \triangleright0000001\Box111\underline{\triangleleft}\triangleright1) \vdash (q_r, \triangleright0000001\Box111\triangleleft\triangleright1\_) \vdash (q_3, \triangleright0000001\Box111\triangleleft\triangleright\underline{1}1) \\
\vdash (q_3, \triangleright0000001\Box111\triangleleft\underline{\triangleright}11) \vdash (q_1, \triangleright0000001\Box111\underline{\triangleleft}\triangleright11) \vdash (q_1, \triangleright0000001\Box11\underline{1}\triangleleft\triangleright11) \\
Third\ round(unable\ to\ finish) \\
\vdash (q_L, \triangleright0000001\Box1\underline{1}0\triangleleft\triangleright11) \vdash (q_L, \underline{\triangleright}0000001\Box110\triangleleft\triangleright11) \vdash (q_2, \triangleright\underline{0}000001\Box110\triangleleft\triangleright11) \\
\vdash (q_2, \triangleright000000\underline{1}\Box110\triangleleft\triangleright11) \vdash (q_R, \triangleright0000000\underline{\Box}110\triangleleft\triangleright11) \vdash (q_R, \triangleright0000000\Box110\underline{\triangleleft}\triangleright11) \\
\vdash (q_1, \triangleright0000000\Box11\underline{0}\triangleleft\triangleright11) \vdash (q_1, \triangleright0000000\Box1\underline{1}0\triangleleft\triangleright11) \vdash (q_L, \triangleright0000000\Box\underline{1}00\triangleleft\triangleright11) \\
\vdash (q_L, \underline{\triangleright}0000000\Box100\triangleleft\triangleright11) \vdash (q_2, \triangleright\underline{0}000000\Box100\triangleleft\triangleright11) \vdash (q_2, \triangleright0000000\underline{\Box}100\triangleleft\triangleright11) \\
Ready\ to\ stop \\
\vdash (q_f, \triangleright0000000\Box\underline{1}00\triangleleft\triangleright11) \vdash (q_f, \triangleright0000000\Box100\triangleleft\triangleright11\_) \vdash (q_H, \triangleright0000000\Box100\triangleleft\triangleright11\underline{\triangleleft})
\end{gathered}
\tag{2}
$$

□

2. Given the alphabet $\{1, 0, \Box, \triangleright, \triangleleft\}$, design a time efficient 3-tape TM M to compute $f : \{0,1\}^* \to \{0,1\}$ 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 eample, 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 $\langle q_S, \triangleright, \triangleright, \triangleright\rangle \to \langle q_1, \triangleright, \triangleright, R, R, S\rangle$. 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) Firstly, copy all the 0 from tape 1 to tape 2

$$
\begin{aligned}
< q_s, \triangleright, \triangleright, \triangleright > \quad &\to \quad < q_c, \triangleright, \triangleright, R, R, R > \\
< q_c, 0, \Box, \Box > \quad &\to \quad < q_c, 0, \Box, R, R, S > \\
< q_c, 1, \Box, \Box > \quad &\to \quad < q_c, \Box, \Box, R, S, S > \\
< q_c, \triangleleft, \Box, \Box > \quad &\to \quad < q_L, \triangleleft, \Box, L, L, S >
\end{aligned}
\tag{3}
$$

Secondly, move to the beginning of the tape

$$\begin{array}{rcl} < q_L, 1, 0, \Box > & \rightarrow & < q_L, 1, 0, L, L, S > \\ < q_L, 0, 0, \Box > & \rightarrow & < q_L, 0, 0, L, L, S > \\ < q_L, 1, \triangleright, \Box > & \rightarrow & < q_L, 1, \triangleright, L, S, S > \\ < q_L, 0, \triangleright, \Box > & \rightarrow & < q_L, 0, \triangleright, L, S, S > \\ < q_L, \triangleleft, \triangleright, \Box > & \rightarrow & < q_1, \triangleleft, \triangleright, R, R, S > \end{array} \tag{4}$$

Thirdly, begin to compare,if read 1 in tape 1,move tape 2 an unit right,else do not move in tape 2.If read 1 in tape 1 but $\triangleleft$ in tape 2(means ending in tape 2), output 0. If read $\triangleleft$ in tape 1 and tape 2 in the same time, output 1.

$$\begin{array}{rcl} < q_1, 0, 0, \Box > & \rightarrow & < q_1, 0, 0, R, S, S > \\ < q_1, 1, 0, \Box > & \rightarrow & < q_1, 1, 0, R, R, S > \\ < q_1, 0, \triangleleft, \Box > & \rightarrow & < q_1, 0, \triangleleft, R, S, S > \\ < q_1, 1, \triangleleft, \Box > & \rightarrow & < q_f, 1, \triangleleft, S, S, R > \\ < q_1, \triangleleft, \triangleleft, \Box > & \rightarrow & < q_f, \triangleleft, 1, S, S, R > \\ < q_1, \triangleleft, 0, \Box > & \rightarrow & < q_f, \triangleleft, 0, S, S, R > \end{array} \tag{5}$$

At last, end the Turing machine.($\forall$ means not caring what).Then we will get the output in tape 3.("$\triangleright 1 \triangleleft$" or "$\triangleright 0 \triangleleft$").

$$< q_f, \forall, \forall, \Box > \quad \rightarrow \quad < q_H, \triangleleft, \triangleleft, S, S, S > \tag{6}$$

(b) i.
Except for the finishing state, we always moves in tape 1, so we only need to count the movement in tape 1.
First we move from the beginning to the end and return to the beginning, which cost $2 * n$ times.
Then when comparing, we at most move the tape 1 from beginning to the end, which cost n times.
As a result, the time complexity is $O(n)$.

ii. When there is only one tape, we have use the same idea but need more time.
Firstly, we copy all the 0 to the end of the tape. We only need to marked the 0 that have been moved by changing it to $\Box$.Then we have 2 parts,one is the input, and the other is all the 0s, which are blocked by $\triangleright$.Every moving costs $2 * n = O(n)$ time complexity.
Secondly, during the comparing time, when reading an 1 in the first part, change a 0 into 1 in the 1 in the second part.If there is no 0 ready for changing or there are 0s left in the end , change to fail state to end, else change to success state to end.Every compare costs $2 * n = O(n)$ time complexity
Lastly, according to the state, output 1 or 0.As every operation cost $O(n)$ time complexity and we total need $O(n)$ operations (according to the method above), so the total time complexity is $O(n^2)$.

$\Box$

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 $w \leqslant k$?

Solution. (a) This is a decision problem.
Certificate: A subset M'

Algorithm 1: 3-Dimensional Matching certifier

Input: A set M'
Output: *True* or *False*

```
for a ∈ M' do
    for b ∈ M' do
        if a ≠ b then
            if a_x == b_x || a_y == b_y || a_z == b_z then
                return False
return True
```

We can define corresponding search problem that Given X, Y, Z, M,
1)find the subset M' of M that of the largest size
or 2)find a subset M' of M of size n if it exists

(b) This is a search problem. When can define the decision problem that If there is a route S that its weight is no larger than k.
Certificate: A route S

Algorithm 2: Travelling Salesman Problem.

Input: A route S,range k
Output: *True* or *False*

```
if route S visits each city exactly once and returns to the origin city then
    a ← 0
    for edge ∈ S do
        a ← a + weight(edge)
    if a <= k then
        return True
return False
```

(c) This is a decision problem.
Certificate: A Job sequence S

Algorithm 3: Job Sequencing.

Input: A Job sequence S,range k
Output: $True$ or $False$

$\omega \leftarrow 0$
for $job \in S$ **do**
  $\omega \leftarrow \omega + penalty(job)$
if $\omega <= k$ **then**
  return True
else
  return False

We can define the search problem that find the job sequence S' with the least penalty.

$\square$

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