Lab10-Turing Machine

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

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

$*$ Name:Xin Xu   Student ID:519021910726   Email: xuxin20010203@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	$\triangleleft$

$\uparrow$
q_S

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

$(q_s, \underline{\triangleright}1111111\square111\triangleleft) \vdash (q_1, \triangleright\underline{1}111111\square111\triangleleft) \vdash^* (q_1, \triangleright1111111\underline{\square}111\triangleleft) \vdash (q_2, \triangleright1111111\square\underline{1}11\triangleleft)$

(Note that for simplicity, we write $(q_1, \triangleright\underline{1}111111\square111\triangleleft) \vdash^* (q_1, \triangleright1111111\underline{\square}111\triangleleft)$ if the corresponding transaction repeats on multiple inputs with the same state.)

Solution. (a) My design is to transfer x, y "1" to "0" everytime encountering y "1" and add "1" when encountering $\triangleright$. The specifications are below:

$\langle q_S, \triangleright \rangle \to \langle q_1, \triangleright, R \rangle$
$\langle q_1, 1 \rangle \to \langle q_1, 1, R \rangle$
$\langle q_1, \triangleleft \rangle \to \langle q_1, \triangleleft, R \rangle$
$\langle q_1, 0 \rangle \to \langle q_1, 0, R \rangle$
$\langle q_1, \square \rangle \to \langle q_2, \square, R \rangle$

q_1 is used to go right and across $\square$. After acrossing, the state becomes q_2.

$\langle q_2, 0 \rangle \to \langle q_2, 0, R \rangle$
$\langle q_2, 1 \rangle \to \langle q_3, 0, L \rangle$
$\langle q_2, \triangleleft \rangle \to \langle q_w, \triangleleft, L \rangle$

q_2 is used to change 1 to 0 so that after all 1 in y has to be changed, the quotient can increase by 1.

$\langle q_3, 0 \rangle \to \langle q_3, 0, L \rangle$
$\langle q_3, 1 \rangle \to \langle q_3, 1, L \rangle$
$\langle q_3, \square \rangle \to \langle q_4, \square, L \rangle$

q_3 is used to go left and across $\square$. After acrossing, the state becomes q_4.

$\langle q_4, 1 \rangle \to \langle q_1, 0, R \rangle$
$\langle q_4, 0 \rangle \to \langle q_4, 0, L \rangle$
$\langle q_4, \triangleleft \rangle \to \langle q_4, \triangleleft, L \rangle$
$\langle q_4, \triangleright \rangle \to \langle q_c, \triangleright, R \rangle$

q_4 is used to change 1 to 0 in x.

$\langle q_w, 0\rangle \rightarrow \langle q_w, 0, L\rangle$
$\langle q_w, \Box\rangle \rightarrow \langle q_d, \Box, L\rangle$
q_w is used to go left and across $\Box$. After acrossing, the state becomes q_d.
$\langle q_d, 0\rangle \rightarrow \langle q_b, \triangleleft, R\rangle$
$\langle q_d, \triangleleft\rangle \rightarrow \langle q_d, \triangleleft, L\rangle$
q_d is used to increase the quotient. The number of $\triangleleft$ by the left of $\Box$ is the quotient.
$\langle q_b, \triangleleft\rangle \rightarrow \langle q_b, \triangleleft, R\rangle$
$\langle q_b, \Box\rangle \rightarrow \langle q_f, \Box, R\rangle$
$\langle q_f, 0\rangle \rightarrow \langle q_f, 1, R\rangle$
$\langle q_f, \triangleleft\rangle \rightarrow \langle q_f, \triangleleft, L\rangle$
$\langle q_f, 1\rangle \rightarrow \langle q_f, 1, L\rangle$
$\langle q_f, \Box\rangle \rightarrow \langle q_2, \Box, R\rangle$
q_b and q_f are used to regain the divisor. q_b is used to go right and q_f is used to regain.
$\langle q_c, 0\rangle \rightarrow \langle q_c, 0, R\rangle$
$\langle q_c, \triangleleft\rangle \rightarrow \langle q_c, 1, R\rangle$
$\langle q_c, \Box\rangle \rightarrow \langle q_p, 0, R\rangle$
$\langle q_p, 1\rangle \rightarrow \langle q_p, 0, R\rangle$
$\langle q_p, \triangleleft\rangle \rightarrow \langle q_H, \triangleleft, S\rangle$
q_c and q_p are used to transfer $\triangleleft$ to 1 and transfer other symbols to 0 and finally terminate the machine.

(b) The diagram is below:

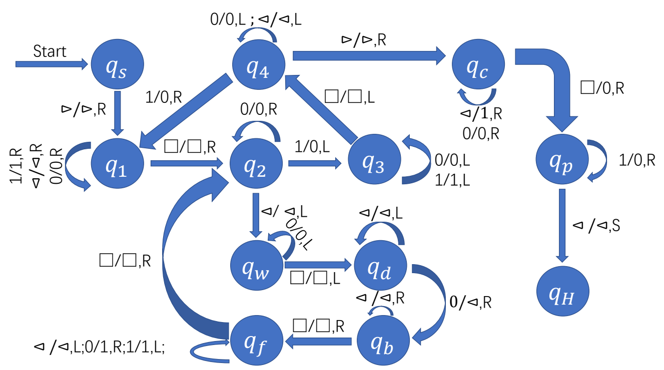

图 1: The transition diagram of the one type TM

(c)

$(q_s, \underline{\triangleright}1111111\Box111\triangleleft) \vdash (q_1, \triangleright\underline{1}111111\Box111\triangleleft) \vdash^* (q_1, \triangleright1111111\underline{\Box}111\triangleleft) \vdash (q_2, \triangleright1111111\Box\underline{1}11\triangleleft)$

$\vdash (q_3, \triangleright1111111\underline{\Box}011\triangleleft) \vdash (q_4, \triangleright111111\underline{1}\Box011\triangleleft) \vdash (q_1, \triangleright1111110\underline{\Box}011\triangleleft) \vdash (q_2, \triangleright1111110\Box\underline{0}11\triangleleft)$

$\vdash (q_2, \triangleright1111110\Box0\underline{1}1\triangleleft) \vdash (q_3, \triangleright1111110\underline{\Box}001\triangleleft) \vdash (q_4, \triangleright111111\underline{0}\Box001\triangleleft) \vdash (q_4, \triangleright11111\underline{1}0\Box001\triangleleft)$

$\vdash (q_1, \triangleright1111100\underline{\Box}001\triangleleft) \vdash (q_2, \triangleright1111100\Box\underline{0}01\triangleleft) \vdash (q_2, \triangleright1111100\Box00\underline{1}\triangleleft) \vdash (q_3, \triangleright1111100\underline{\Box}000\triangleleft)$

$\vdash (q_4, \triangleright111110\underline{0}\Box000\triangleleft) \vdash (q_4, \triangleright1111\underline{1}00\Box000\triangleleft) \vdash (q_1, \triangleright1111000\underline{\Box}000\triangleleft) \vdash (q_2, \triangleright1111000\Box\underline{0}00\triangleleft)$

$\vdash (q_2, \triangleright1111000\Box000\underline{\triangleleft}) \vdash (q_w, \triangleright1111000\Box00\underline{0}\triangleleft) \vdash (q_w, \triangleright1111000\underline{\Box}000\triangleleft) \vdash (q_d, \triangleright111100\underline{0}\Box000\triangleleft)$

$\vdash (q_b, \triangleright111100\triangleleft\underline{\Box}000\triangleleft) \vdash (q_f, \triangleright111100\triangleleft\Box\underline{0}00\triangleleft) \vdash (q_f, \triangleright111100\triangleleft\Box1\underline{0}0\triangleleft) \vdash (q_f, \triangleright111100\triangleleft\Box11\underline{0}\triangleleft)$

$\vdash (q_f, \triangleright111100\triangleleft\Box111\underline{\triangleleft}) \vdash (q_f, \triangleright111100\triangleleft\underline{\Box}111\triangleleft) \vdash (q_2, \triangleright111100\triangleleft\Box\underline{1}11\triangleleft) \vdash (q_3, \triangleright111100\triangleleft\underline{\Box}011\triangleleft)$

$\vdash (q_4, \triangleright111100\underline{\triangleleft}\Box011\triangleleft) \vdash (q_4, \triangleright11110\underline{0}\triangleleft\Box011\triangleleft) \vdash (q_4, \triangleright111\underline{1}00\triangleleft\Box011\triangleleft) \vdash (q_1, \triangleright111000\triangleleft\underline{\Box}011\triangleleft)$

$$\vdash (q_2, \rhd 111000\lhd\Box\underline{0}11\lhd) \vdash (q_2, \rhd 111000\lhd\Box 0\underline{1}1\lhd) \vdash (q_3, \rhd 111000\lhd\underline{\Box}001\lhd) \vdash (q_4, \rhd 111000\underline{\lhd}\Box 001\lhd)$$
$$\vdash (q_4, \rhd 11\underline{1}000\lhd\Box 001\lhd) \vdash (q_1, \rhd 110000\lhd\underline{\Box}001\lhd) \vdash (q_2, \rhd 110000\lhd\Box\underline{0}01\lhd) \vdash (q_2, \rhd 110000\lhd\Box 00\underline{1}\lhd)$$
$$\vdash (q_3, \rhd 110000\lhd\underline{\Box}000\lhd) \vdash (q_4, \rhd 110000\underline{\lhd}\Box 000\lhd) \vdash (q_4, \rhd 1\underline{1}0000\lhd\Box 000\lhd) \vdash (q_1, \rhd 100000\lhd\underline{\Box}000\lhd)$$
$$\vdash (q_2, \rhd 100000\lhd\Box\underline{0}00\lhd) \vdash (q_2, \rhd 100000\lhd\Box 000\underline{\lhd}) \vdash (q_w, \rhd 100000\lhd\Box 00\underline{0}\lhd) \vdash (q_w, \rhd 100000\lhd\underline{\Box}000\lhd)$$
$$\vdash (q_d, \rhd 100000\underline{\lhd}\Box 000\lhd) \vdash (q_d, \rhd 10000\underline{0}\lhd\Box 000\lhd) \vdash (q_b, \rhd 10000\lhd\lhd\underline{\Box}000\lhd) \vdash (q_f, \rhd 10000\lhd\lhd\Box\underline{0}00\lhd)$$
$$\vdash (q_f, \rhd 10000\lhd\lhd\Box 1\underline{0}0\lhd) \vdash (q_f, \rhd 10000\lhd\lhd\Box 11\underline{0}\lhd) \vdash (q_f, \rhd 10000\lhd\lhd\Box 111\underline{\lhd}) \vdash (q_f, \rhd 10000\lhd\lhd\underline{\Box}111\lhd)$$
$$\vdash (q_2, \rhd 10000\lhd\lhd\Box\underline{1}11\lhd) \vdash (q_3, \rhd 10000\lhd\lhd\underline{\Box}011\lhd) \vdash (q_4, \rhd 10000\lhd\underline{\lhd}\Box 011\lhd) \vdash (q_4, \rhd\underline{1}0000\lhd\lhd\Box 011\lhd)$$
$$\vdash (q_1, \rhd 0\underline{0}000\lhd\lhd\Box 011\lhd) \vdash (q_1, \rhd 00000\lhd\lhd\underline{\Box}011\lhd) \vdash (q_2, \rhd 00000\lhd\lhd\Box\underline{0}11\lhd) \vdash (q_2, \rhd 00000\lhd\lhd\Box 0\underline{1}1\lhd)$$
$$\vdash (q_3, \rhd 00000\lhd\lhd\underline{\Box}001\lhd) \vdash (q_4, \rhd 00000\lhd\underline{\lhd}\Box 001\lhd) \vdash (q_4, \underline{\rhd}00000\lhd\lhd\Box 001\lhd) \vdash (q_c, \rhd 00000\underline{\lhd}\lhd\Box 001\lhd)$$
$$\vdash (q_c, \rhd 000001\underline{\lhd}\Box 001\lhd) \vdash (q_c, \rhd 0000011\underline{\Box}001\lhd) \vdash (q_p, \rhd 00000110\underline{0}01\lhd) \vdash (q_p, \rhd 0000011000\underline{1}\lhd)$$
$$\vdash (q_p, \rhd 00000110000\underline{\lhd}) \vdash (q_H, \rhd 00000110000\underline{\lhd})$$

□

2. Given the alphabet $\{1, 0, \Box, \rhd, \lhd\}$, 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 $\rhd 001101\lhd$, M should output 1

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

$\langle q_S, \rhd, \rhd, \rhd\rangle \to \langle q_1, \rhd, \rhd, R, R, S\rangle$
$\langle q_1, 0, \Box, \rhd\rangle \to \langle q_1, 0, \rhd, R, R, S\rangle$
$\langle q_1, 1, \Box, \rhd\rangle \to \langle q_1, \Box, \rhd, R, S, S\rangle$
$\langle q_1, \lhd, \Box, \rhd\rangle \to \langle q_2, \lhd, \rhd, L, L, S\rangle$
q_1 is used from left to right to record the number of 0.
$\langle q_2, 0, 0, \rhd\rangle \to \langle q_2, 0, \rhd, L, S, S\rangle$
$\langle q_2, 1, 0, \rhd\rangle \to \langle q_2, \Box, \rhd, L, L, S\rangle$
$\langle q_2, \rhd, 0, \rhd\rangle \to \langle q_H, 0, 0, S, S, S\rangle$
$\langle q_2, 0, \rhd, \rhd\rangle \to \langle q_2, \rhd, \rhd, L, S, S\rangle$
$\langle q_2, 1, \rhd, \rhd\rangle \to \langle q_H, \rhd, 0, L, S, S\rangle$
$\langle q_2, \rhd, \rhd, \rhd\rangle \to \langle q_H, \rhd, 1, S, S, S\rangle$
q_2 is used from right to left to compare 1 with 0.
q_H is the terminate state.

(b) Suppose the 3-type TM cumputes f with time complexity of $T(n)$. Firstly, when 3-type TM is changed to 1-type TM, everytime searching the elements in the original 3-type TM needs $3T(n)$, and there are $T(n)$ times computes, so this transition needs $3T(n)^2$. Secondly, to compute n symbols, we need $\log n$ cells to show one symbol. In conclusion, the total time complexity is $O(T(n)^2 \log n)$.
The machine M' has one type in which the elements in 3-type are put in order: a_1, b_1, c_1, a_2, $b_2, c_2, \ldots .(a, b, c$ are the original types). and every element is expressed by $\log n$ 0's and 1's.

□

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) Decision problem: Is there a subset M' of M of size n where no two elements of M' agree in any coordinate?
Search problem: Find a subset M' of M of size n where no two elements of M' agree in any coordinate if it exists.
Certifier:

Algorithm 1: 3-Dimensional Matching

Input: M, M'
Output: True or False

if *$M' \nsubseteq M ||$ the size of $M' \neq n$* **then**
  return *False*;
if *no two elements of M' agree in any coordinate* **then**
  return *True*;
return *False*;

(b) Decision problem: Is there a shortest possible route with the length $\leq k$ that visits each city exactly once and returns to the origin city?
Search problem: Find the possible route that visits each city exactly once and returns to the origin city.
Certifier:

Algorithm 2: Travelling Salesman Problem

Input: A city graph $G(V, E, W)$ and the route R and maximum length k.
Output: True or False

if *R isn't the Hamiltonian cycle* **then**
  return *False*;
else
  if *the length of $R > k$* **then**
    return *False*;
  else
    return *True*;

(c) Decision problem: Does there exist a job sequence that has a total penalty $w \leqslant k$?
Search problem: Find the job sequence that has a lowest total penalty.

Certifier:

Algorithm 3: Job Sequencing

Input: A job sequence S, the deadline of each job D, the penalty of each job P and the maximum penalty k.

Output: True or False

```
if S isn't the Job Sequence then
    return False;
else
    if the total penalty of S > k then
        return False;
    else
        return True;
```

□

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