Lab10-Turing Machine

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

∗ If there is any problem, please contact TA Yihao Xie.
∗ Name: Wendi Chen Student ID: 519021910071 Email: chenwendi-andy@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 $\langle q_S, \triangleright \rangle \rightarrow \langle q_1, \triangleright, R \rangle$. 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:

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

   (Note that for simplicity, we write $(q_1, \triangleright\underline{1}111111\square 111\triangleleft) \vdash^* (q_1, \triangleright 1111111\underline{\square}111\triangleleft)$ if the corresponding transaction repeats on multiple inputs with the same state.)

Solution.

(a) Let us show the **basic idea** first. Generally, we want to compare the symbols on the both sides of the"splitting $\square$" and use "0" to denote the symbol that has been compared. And after one iteration, the tape become

	$\triangleright$	1	1	1	1	0	0	0	$\square$	0	0	0	$\triangleleft$	

Then, we go to the every left of the tape and change the first symbol which is not "$\square$" or "$\triangleright$" to "$\square$", which represents a counter.

	$\triangleright$	$\square$	1	1	1	0	0	0	$\square$	0	0	0	$\triangleleft$	

Next, we update the tape so that we can do the next iteration. Remember that we need to compensate one "1".

	$\triangleright$	$\square$	1	1	1	1	$\square$	1	1	1	$\triangleleft$	0	$\triangleleft$	

After that, we can finish the next iteration.

	$\triangleright$	$\square$	$\square$	1	$\square$	1	1	1	$\triangleleft$	0	$\triangleleft$	0	$\triangleleft$	

And when we do the third iteration, there is no enough symbol on the left for comparison.

	$\triangleright$	$\square$	$\square$	0	$\square$	0	0	1	$\triangleleft$	0	$\triangleleft$	0	$\triangleleft$	

So we can do the finishing steps: change the first "0" or the "splitting $\square$" to "$\triangleleft$" and replace other "$\square$" on the left with "1".

	$\triangleright$	1	1	$\triangleleft$	$\square$	0	0	1	$\triangleleft$	0	$\triangleleft$	0	$\triangleleft$

Here we get the final result.

Next, we'll divide the whole process into several parts and try to write the specifications of M.
In part one, we want to do some preparations for the comparison. We need skip the first few "$\square$" and move the the right of the "splitting $\square$". After we change the state to q_3, we're ready to do the comparison.

$$\langle q_S, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle \quad \langle q_1, \square\rangle \to \langle q_1, \square, R\rangle \quad \langle q_1, 1\rangle \to \langle q_2, 1, R\rangle$$
$$\langle q_2, 1\rangle \to \langle q_2, 1, R\rangle \quad \langle q_2, \square\rangle \to \langle q_3, \square, R\rangle$$

In part two, we want to do the comparison. Here, we use several states to represents the direction that the head is moving. And we also need to distinguish whether we are now on the left or the right side of the "splitting $\square$".

$$\langle q_3, 0\rangle \to \langle q_3, 0, R\rangle \quad \langle q_3, 1\rangle \to \langle q_4, 0, L\rangle \quad \langle q_4, 0\rangle \to \langle q_4, 0, L\rangle$$
$$\langle q_4, \square\rangle \to \langle q_5, \square, L\rangle \quad \langle q_5, 0\rangle \to \langle q_5, 0, L\rangle \quad \langle q_5, 1\rangle \to \langle q_6, 0, R\rangle$$
$$\langle q_6, 0\rangle \to \langle q_6, 0, R\rangle \quad \langle q_6, \square\rangle \to \langle q_3, \square, R\rangle$$

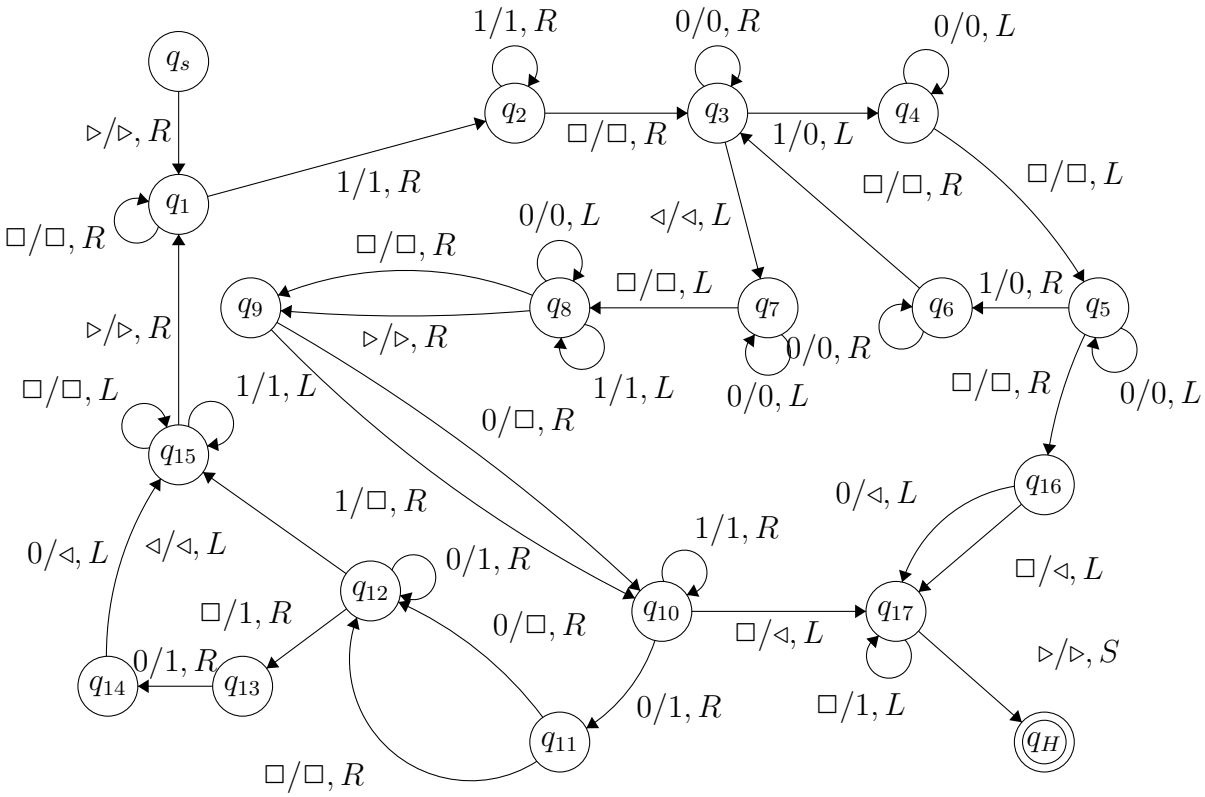
In part three, we need to update the tape after finishing one iteration. There are many details here, but we don't have to worry. All we need to remember is to move to the every left first and use "$\square$" as a counter. Then maintain the number of 1's in the string through a series of operations and try to revert to a pattern similar to the initial tape.

$$\langle q_3, \triangleleft\rangle \to \langle q_7, \triangleleft, L\rangle \quad \langle q_7, 0\rangle \to \langle q_7, 0, L\rangle \quad \langle q_7, \square\rangle \to \langle q_8, \square, L\rangle$$
$$\langle q_8, 0\rangle \to \langle q_8, 0, L\rangle \quad \langle q_8, 1\rangle \to \langle q_8, 1, L\rangle$$
$$\langle q_8, \triangleright\rangle \to \langle q_9, \triangleright, R\rangle \quad \langle q_8, \square\rangle \to \langle q_9, \square, R\rangle$$
$$\langle q_9, 1\rangle \to \langle q_{10}, \square, R\rangle \quad \langle q_{10}, 1\rangle \to \langle q_{10}, 1, R\rangle \quad \langle q_{10}, 0\rangle \to \langle q_{11}, 1, R\rangle$$
$$\langle q_{11}, 0\rangle \to \langle q_{12}, \square, R\rangle \quad \langle q_{12}, 0\rangle \to \langle q_{12}, 1, R\rangle \quad \langle q_{12}, \square\rangle \to \langle q_{13}, 1, R\rangle$$
$$\langle q_9, 0\rangle \to \langle q_{10}, \square, R\rangle \quad \langle q_{11}, \square\rangle \to \langle q_{12}, \square, R\rangle \quad \langle q_{12}, \triangleleft\rangle \to \langle q_{15}, \triangleleft, L\rangle$$
$$\langle q_{13}, 0\rangle \to \langle q_{14}, 1, R\rangle \quad \langle q_{14}, 0\rangle \to \langle q_{15}, \triangleleft, L\rangle \quad \langle q_{15}, \square\rangle \to \langle q_{15}, \square, L\rangle$$
$$\langle q_{15}, 1\rangle \to \langle q_{15}, 1, L\rangle \quad \langle q_{15}, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle$$

At last, let us do the finishing steps. We change "$\square$" to "1" to represent the result and mark the "$\triangleleft$".

$$\langle q_5, \square\rangle \to \langle q_{16}, \square, R\rangle \quad \langle q_{16}, 0\rangle \to \langle q_{17}, \triangleleft, L\rangle \quad \langle q_{16}, \square\rangle \to \langle q_{17}, \triangleleft, L\rangle$$
$$\langle q_{10}, \square\rangle \to \langle q_{17}, \triangleleft, L\rangle \quad \langle q_{17}, \square\rangle \to \langle q_{17}, 1, L\rangle \quad \langle q_{17}, \triangleright\rangle \to \langle q_H, \triangleright, S\rangle$$

(b) Here is the state transition diagram.



(c) In this section, we'll show the whole process. For simplicity, we only retain the symbols that is between the first "$\triangleright$" and the corresponding "$\triangleleft$".

$$(q_s, \underline{\triangleright}1111111\square111\triangleleft) \vdash (q_1, \triangleright\underline{1}111111\square111\triangleleft) \vdash (q_2, \triangleright1\underline{1}11111\square111\triangleleft) \vdash^* (q_2, \triangleright1111111\underline{\square}111\triangleleft)$$
$$\vdash (q_3, \triangleright1111111\square\underline{1}11\triangleleft) \vdash (q_4, \triangleright1111111\underline{\square}011\triangleleft) \vdash (q_5, \triangleright1111111\underline{1}\square011\triangleleft) \vdash (q_6, \triangleright1111110\underline{\square}011\triangleleft)$$
$$\vdash (q_3, \triangleright1111110\square\underline{0}11\triangleleft) \vdash (q_3, \triangleright1111110\square0\underline{1}1\triangleleft)$$

Next few steps follow the same pattern, so we omit it.

$$(q_5, \triangleright1111\underline{1}00\square000\triangleleft) \vdash (q_6, \triangleright1111\underline{0}0\square000\triangleleft) \vdash^* (q_6, \triangleright1111000\underline{\square}000\triangleleft)$$
$$\vdash (q_3, \triangleright1111000\square\underline{0}00\triangleleft) \vdash^* (q_3, \triangleright1111000\square000\underline{\triangleleft}) \vdash (q_7, \triangleright1111000\square00\underline{0}\triangleleft)$$
$$\vdash^* (q_7, \triangleright1111000\underline{\square}000\triangleleft) \vdash (q_8, \triangleright1111000\square000\triangleleft) \vdash^* (q_8, \underline{\triangleright}1111000\square000\triangleleft)$$
$$\vdash (q_9, \triangleright\underline{1}111000\square000\triangleleft) \vdash (q_{10}, \triangleright\square\underline{1}11000\square000\triangleleft) \vdash^* (q_{10}, \triangleright\square111\underline{0}00\square000\triangleleft)$$
$$\vdash (q_{11}, \triangleright\square1111\underline{0}0\square000\triangleleft) \vdash (q_{12}, \triangleright\square1111\square\underline{0}\square000\triangleleft) \vdash (q_{12}, \triangleright\square1111\square1\underline{\square}000\triangleleft)$$
$$\vdash (q_{13}, \triangleright\square1111\square11\underline{0}00\triangleleft) \vdash (q_{14}, \triangleright\square1111\square111\underline{0}0\triangleleft) \vdash (q_{15}, \triangleright\square1111\square11\underline{1}\triangleleft)$$
$$\vdash^* (q_{15}, \underline{\triangleright}\square1111\square111\triangleleft) \vdash (q_1, \triangleright\underline{\square}1111\square111\triangleleft)$$

Next several steps follow the above pattern and we omit it.

$$(q_3, \triangleright\square\square0\square0\underline{1}1\triangleleft) \vdash (q_4, \triangleright\square\square0\square\underline{0}01\triangleleft) \vdash (q_4, \triangleright\square\square0\underline{\square}001\triangleleft)$$
$$\vdash (q_5, \triangleright\square\square\underline{0}\square001\triangleleft) \vdash (q_5, \triangleright\square\underline{\square}0\square001\triangleleft) \vdash (q_{16}, \triangleright\square\square\underline{0}\square001\triangleleft)$$
$$\vdash (q_{17}, \triangleright\square\underline{\square}\triangleleft) \vdash (q_{17}, \triangleright\underline{\square}1\triangleleft) \vdash (q_{17}, \underline{\triangleright}11\triangleleft) \vdash (q_H, \underline{\triangleright}11\triangleleft)$$

In addition, I also write a program (you can refer to *Code-TM-Program.txt*) and test it through a simulator (*https://alistat.eu/online/turingmachinesimulator*). Here 2 stands for $\triangleright$, 3 stands for $\triangleleft$ and 4 stands for $\square$. You can see our design returns the right answer (Figure 1).

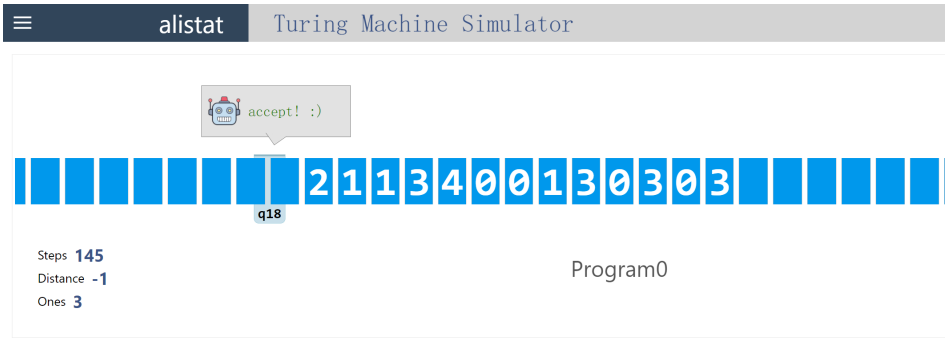


Figure 1: Using the Simulator to Check the Design

□

2. Given the alphabet $\{1, 0, \square, \triangleright, \triangleleft\}$, 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) Initially the input string is located on the first tape like "$\triangleright 001101 \triangleleft \square\square\cdots$", strings on all other tapes are "$\triangleright\square\square\cdots$". The idea is to copy the "1"s on the first tape to the second tape and then compare the number of "1"s on the second tape with the number of "0"s on the first tape. Thus, let us write the specifications of M.
Start State: In this stage, we do some preparations for copying.

$$\langle q_S, \triangleright, \triangleright, \triangleright\rangle \to \langle q_C, \triangleright, \triangleright, R, R, R\rangle$$

Begin to copy: When copying, we just copy "1"s to the second tape. Therefore, we skip those "0"s. When arriving at the every right of the input tape, we finish this stage.

$$\begin{aligned}
\langle q_C, 0, \square, \square\rangle &\to \langle q_C, \square, \square, R, S, S\rangle \\
\langle q_C, 1, \square, \square\rangle &\to \langle q_C, 1, \square, R, R, S\rangle \\
\langle q_C, \triangleleft, \square, \square\rangle &\to \langle q_l, \square, \square, L, S, S\rangle
\end{aligned}$$

Return back to the leftmost: This stage is quite trivial. We just move the head of the input tape to the every left of it.

$$\begin{aligned}
\langle q_l, 0, \square, \square\rangle &\to \langle q_l, \square, \square, L, S, S\rangle \\
\langle q_l, 1, \square, \square\rangle &\to \langle q_l, \square, \square, L, S, S\rangle \\
\langle q_l, \triangleright, \square, \square\rangle &\to \langle q_t, \square, \square, R, L, S\rangle
\end{aligned}$$

Begin to compare: When doing the comparison, we want to match one "1" on the second tape with one "0" on the input tape. We should be careful to determine when we should change the state. There're three cases and we can solve them one by one.

$$\begin{aligned}
\langle q_t, 0, 1, \square\rangle &\to \langle q_t, 1, \square, R, L, S\rangle \\
\langle q_t, 1, 1, \square\rangle &\to \langle q_t, 1, \square, R, S, S\rangle \\
\langle q_t, 1, \triangleright, \square\rangle &\to \langle q_t, \triangleright, \square, R, S, S\rangle \\
\langle q_t, \triangleleft, \triangleright, \square\rangle &\to \langle q_r, \triangleright, 1, S, S, R\rangle \\
\langle q_t, 0, \triangleright, \square\rangle &\to \langle q_r, \triangleright, 0, S, S, R\rangle \\
\langle q_t, \triangleleft, 1, \square\rangle &\to \langle q_r, 1, 0, S, S, R\rangle
\end{aligned}$$

Ready to terminate: There're also three cases accordingly.

$$\begin{aligned}
\langle q_r, \triangleleft, \triangleright, \square\rangle &\to \langle q_h, \triangleright, \triangleleft, S, S, S\rangle \\
\langle q_r, \triangleleft, 1, \square\rangle &\to \langle q_h, 1, \triangleleft, S, S, S\rangle \\
\langle q_r, 0, \triangleright, \square\rangle &\to \langle q_h, \triangleright, \triangleleft, S, S, S\rangle
\end{aligned}$$

(b) In fact, according to (a), if we want to compute f with n symbols in the input through the 3-tape TM M, the time complexity will be $O(n)$. That's because in every step the head of the first tape will move to left or right until M is about to terminate. So during the whole process, we just sweep the input tape from left to right twice and backwards once. Thus, the time complexity is $T(n) = O(n)$. Now, we want to use a one-tape TM $M^{'}$ to compute the same function. We need to transfer the 3-tape TM to a one-tape TM.

The idea is simple. We interleave 3 tapes into one tape, which means we use locations $1, 3+1, 2+1, \cdots$ to encode the first tape and use locations $2, 3+2, 2+2, \cdots$ to encode the second tape... For every symbol a in M's alphabet, $M^{'}$ will contain both the symbol a and the symbol $\hat{a}$. Only one symbol in each original tape will be hatted, which indicates the head of each tape. To simulate one step of M, $M^{'}$ will do two sweeps of the work tape. In the first time, it sweeps from left to right to records the 3 symbols that are hatted. And then $M^{'}$ will use M's transition function to determine the new state, symbols and head movements and do the second sweep from right to left and update the tape. Because M never reaches more than location $T(n)$ of any of its tapes, $M^{'}$ will never reach more than location $3T(n)$. Thus, for every step in M, M will do $3cT(n)$ steps to simulate M, where c is a constant because there are some additional works like recording and updating. Thus, the time complexity of $M^{'}$ will be $3cT^2(n) = O(n^2)$.

□

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 problem is a decision problem. We can define a corresponding search problem like this: find the subset M' of M with the maximum size n where no two elements of M' agree in any coordinate.

Certificate: a subset M'.

Algorithm 1: Certifier for (a)

Input: A subset M' of M
Output: A boolean that represents the result

```
1 flag ← True;
2 for A ∈ M' do
3     for B ∈ M' and A ≠ B do
4         if A_x = B_x or A_y = B_y or A_z = B_z then
5             flag ← False;
6 return flag;
```

(b) This problem is a search problem. We can define a corresponding decision problem like this: does there exist a route that has a $length \leq k$ and visits each city exactly once and returns to the origin city?

Certificate: a route R.

Algorithm 2: Certifier for (b)

Input: A route R
Output: A boolean that represents the result

```
1 if R doesn't visit each city exactly once and return to the origin city then
2     return False;
3 if the length of R ≤ k then
4     return True;
5 else
6     return False;
```

(c) This problem is a decision problem. We can define a corresponding search problem like this: fine the job sequence with the minimum penalty.
Certificate: a job sequence S.

Algorithm 3: Certifier for (c)

Input: A job sequence S
Output: A boolean that represents the result

```
1 if the penalty of S ≤ k then
2     return True;
3 else
4     return False;
```

□

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