数据科学与计算机学院 软件工程 郑卓民 18342138

## Problem:

1. What a Turing machine is?
2. How a Turing Machine executes a program?

## What a Turing machine is?

➔ A Turing machine is a hypothetical machine thought of by the mathematician Alan Turing in 1936. Despite its simplicity, the machine can simulate ANY computer algorithm, no matter how complicated it is!

A Turing machine is a mathematical model of computation that defines an abstract machine, which manipulates symbols on a strip of tape according to a table of rules. Despite the model's simplicity, given any computer algorithm, a Turing machine capable of simulating that algorithm's logic can be constructed.

The machine operates on an infinite memory tape divided into discrete "cells". The machine positions its "head" over a cell and "reads" or "scans" the symbol there. Then, as per the symbol and its present place in a "finite table" of user-specified instructions, the machine (i) writes a symbol (e.g., a digit or a letter from a finite alphabet) in the cell (some models allowing symbol erasure or no writing), then (ii) either moves the tape one cell left or right (some models allow no motion, some models move the head), then (iii) (as determined by the observed symbol and the machine's place in the table) either proceeds to a subsequent instruction or halts the computation.

## How a Turing Machine executes a program?

### Operation principle:

Turing's basic idea is to use machines to simulate people's mathematical operations with paper and pen.

At any one time, the machine has a head which is positioned over one of the squares on the tape. With this head, the machine can perform three very basic operations:

1. Read the symbol on the square under the head.
2. Edit the symbol by writing a new symbol or erasing it.
3. Move the tape left or right by one square so that the machine can read and edit the symbol on a neighbouring square.

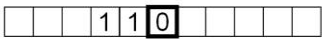
### A simple program:

With the symbols "1 1 0" printed on the tape, let's attempt to convert the 1s to 0s and vice versa. This is called bit inversion, since 1s and 0s are bits in binary. This can be done by passing the following instructions to the Turing machine, utilising the machine's reading capabilities to decide its subsequent operations on its own. These instructions make up a simple program.

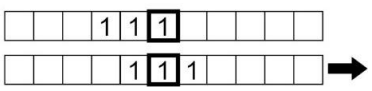
| State | Symbol read | Write instruction | Move instruction | Next state |
|---|---|---|---|---|
| State 0 | Blank | None | None | Stop state |
| | 0 | Write 1 | Move tape to the right | State 0 |
| | 1 | Write 0 | Move tape to the right | State 0 |

The machine will first read the symbol under the head, write a new symbol accordingly, then move the tape left or right as instructed, before repeating the read-write-move sequence again.

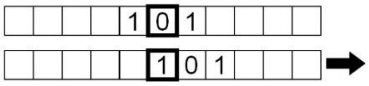
Let's see what this program does to our tape from the previous end point of the instructions:



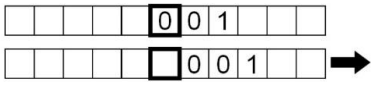

1. The current symbol under the head is 0, so we write a 1 and move the tape right by one square.



2. The symbol being read is now 1, so we write a 0 and move the tape right by one square.




3. Similarly, the symbol read is a 1, so we repeat the same instructions.




4. Finally, a 'blank' symbol is read, and stop the “read-write-move”(Next state).

**Summary:**

A Turing machine is a seven tuple, {Q, ∑, Γ, δ, Q0, qaccept, qreject}, where Q, ∑, Γ are all finite sets and satisfy the following requirements:

1. Q is the state set;
2. Σ is the input alphabet, which does not contain special blank characters;
3. Γ is alphabetic, where □∈Γ and ∑∈Γ;
4. δ: Q × Γ→ Q × Γ×{L, R} is the transfer function, where L, R indicates whether the read / write head moves to the left or to the right;
5. q0 ∈ Q is the initial state;
6. qaccept is the accepted state.
7. qreject is in reject status, and qreject ≠ qaccept.

Turing machine M = (Q, ∑, Γ, δ, q0, qaccept, qreject) will operate as follows:
At the beginning, fill in the input symbol string from left to right on the No. box of the paper tape, and leave the other boxes blank (i.e. fill in the blank character). M's read / write head points to grid 0, and M is in state q0. After the machine starts to run, it is calculated according to the rules described in the transfer function δ. For example, if the current state of the machine is q and the symbol in the cell referred to by the read / write head is x, set δ (q, x) = (q', x', L), the machine will enter the new state q', change the symbol in the cell referred to by the read / write head to X', and then move the read / write head one cell to the left. If at a certain time, the read-write head refers to grid No. 0, but according to the transfer function, it will continue to move to the left, and then it will stop in place. In other words, the read-write head never moves out of the left edge of the tape. If M enters the state qaccept according to the transfer function at a certain time, it stops immediately and accepts the input string; if M enters the state qreject according to the transfer function at a certain time, it stops immediately and rejects the input string.

Note that the transfer function δ is a partial function. In other words, for some q, x, δ (q, x), it may not be defined. If the next operation is not defined in operation, the machine will shut down immediately.