Turing Machine

Ken Q Pu, Faculty of Science, Ontario Tech University

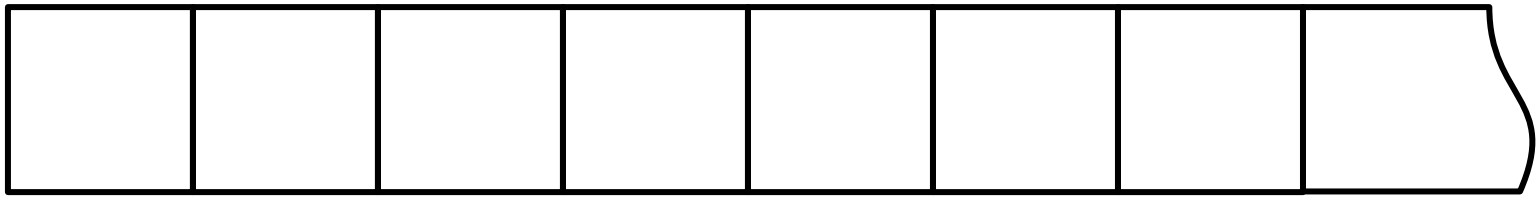
Turing Machine

- TM is the most powerful computing device possible.
- TM is mostly mechanically possible.
- The impossible requirements of TM are:
  - TM has unlimited and unrestricted memory.
  - TM has no upper bounds on how long it takes to execute.

Components of a TM

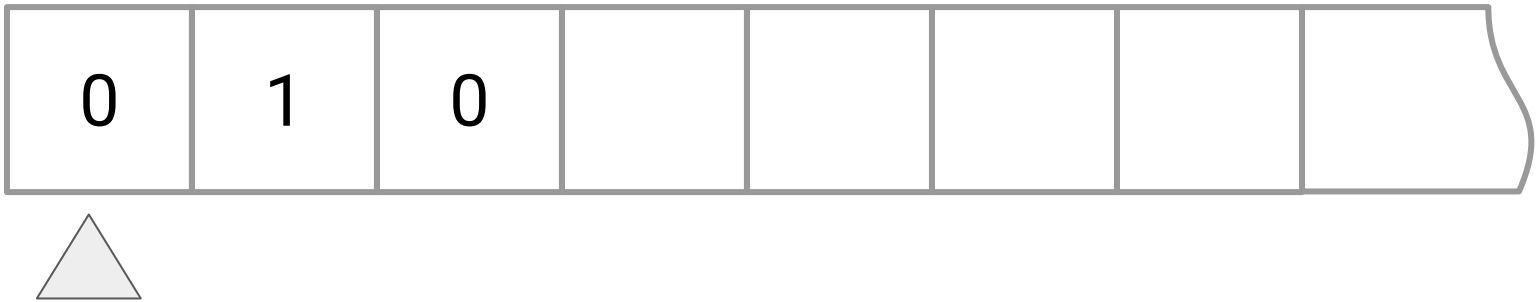
- Tape to store an array of cells
- Head that can move along the tape and modify the cell contents
- Control logic that can react to head dynamics

Components of TM: Tape



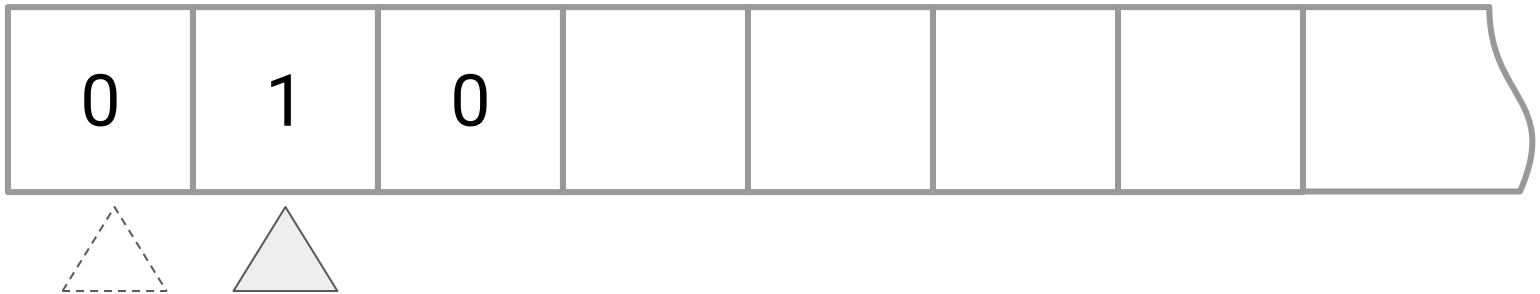
- Each cell can be blank or store one symbol from the alphabet.
- There are infinitely many cells to the right of the tape.

Components of TM: Head



- The head is always positioned at some cell along the tape.
- The head can move left or right, one cell at a time.
- The head reads the content of the **current** cell.
- The head can be instructed to modify the content of the current cell **after** its read.

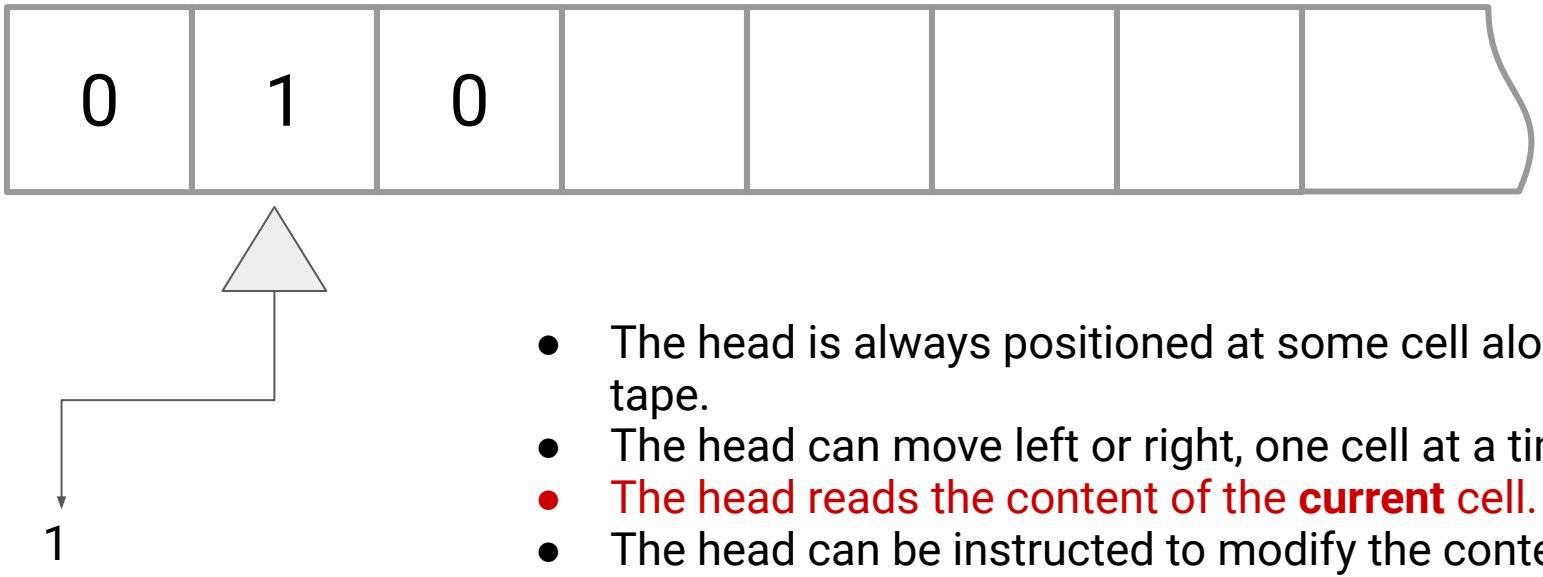
Components of TM: Head



Move right

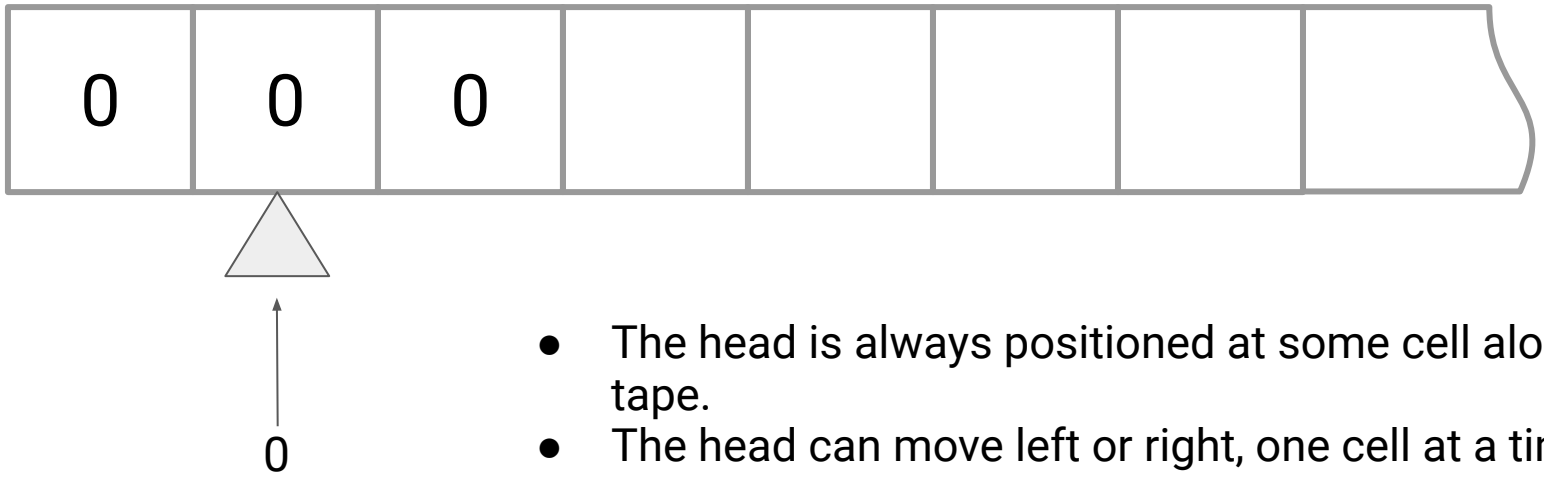
- The head is always positioned at some cell along the tape.
- The head can move left or right, one cell at a time.
- The head reads the content of the **current** cell.
- The head can be instructed to modify the content of the current cell **after** its read.

Components of TM: Head



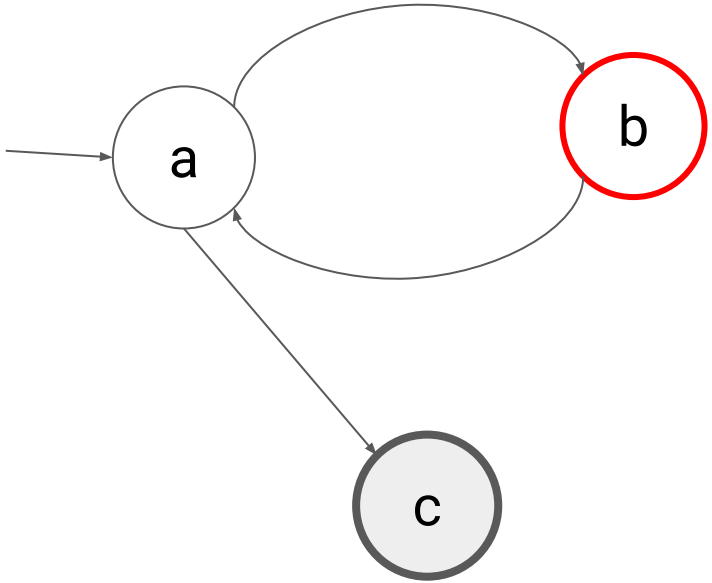
- The head is always positioned at some cell along the tape.
- The head can move left or right, one cell at a time.
- The head reads the content of the **current** cell.
- The head can be instructed to modify the content of the current cell **after** its read.

Components of TM: Head



- The head is always positioned at some cell along the tape.
- The head can move left or right, one cell at a time.
- The head reads the content of the **current** cell.
- The head can be instructed to modify the content of the current cell **after** its read.

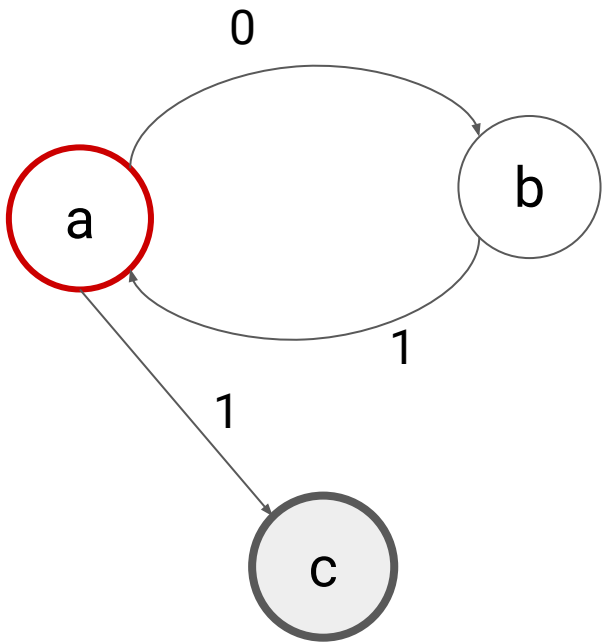
Component of TM: Control Logic



- States
  - Initial state
  - Current state
  - Final states
- TM stops when it reaches any of its final states
  - Halting

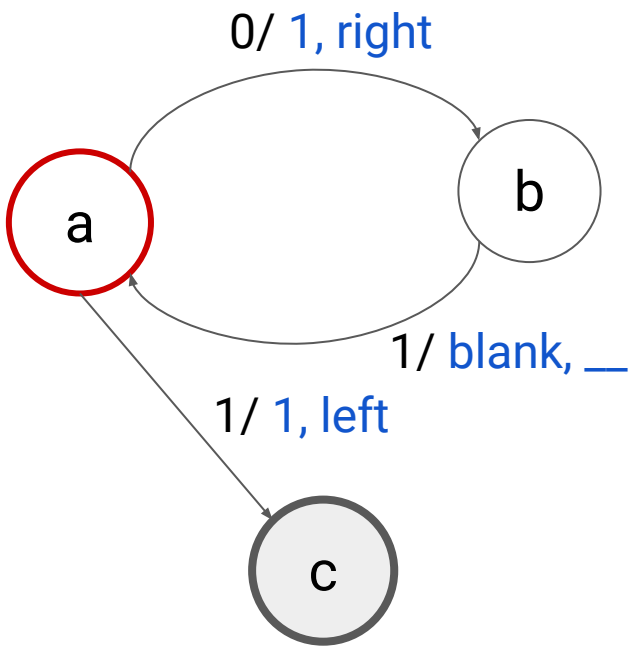
Ken Q Pu, Faculty of Science, Ontario Tech University

Component of TM: Control Logic



- States
  - Initial state
  - Current state
  - Final states
- Transitions triggered by what the head is reading

Component of TM: Control Logic



- States
  - Initial state
  - Current state
  - Final states
- Transitions triggered by what the head is reading
- Each transition also has an instruction for the head:
  - What symbol to write to current cell
  - Which direction to move the head

Turing Machine and decision problem

- Consider a decision problem, P. It has (infinitely) many instances.
- There is an encoding function, ENC, that maps each instance to a string over the alphabet.
- We need a TM that is specifically designed for solving P. Call it M.

Now, we can solve P for any input x:

1. Encode x into a string using ENC(x)
2. Write ENC(x) on the tape of M
3. Start running M
4. If M halts, we get our answer
5. If M never halts, then
    a. M is badly designed. 99.999% of the times
    b. P might be undecidable.

Getting the answer

Relying on final halting state:

- Assign which accepting state corresponds to YES, and which corresponds to NO.
- When M halts, check the state it's in.

Relying on the tape content:

- Rely on the TM to modify the tape content to determine whether the answer is YES or NO
- E.g. if the tape starts with 1, then that's YES. Otherwise it's NO.