Data Structures

R. K. Ghosh

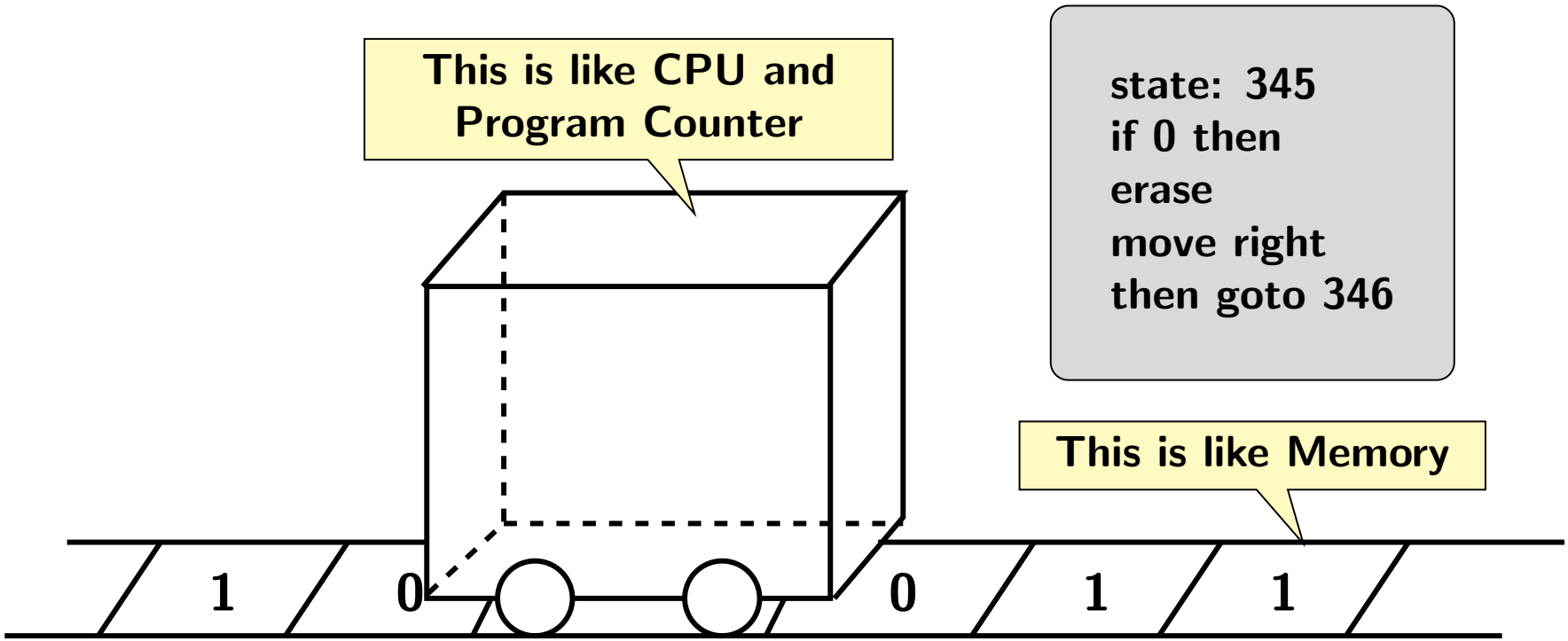
IIT Bhilai

Data structures: Computational Model

Turing Machine

- A. M. Turing gave an abstract definition of a computation using abstract machine called TM.
- A TM manipulates a string of 0s, 1s and spaces on a strip of tape according a table of rule (program book).
- There is control (automaton):
  - It has a knowledge of its current state.
  - It examines each cell on the tape at a time.
  - It consults a program book which tells it what to do in the current state.

Turing Machine



- After examining current input, RW-head of TM either moves left or right.
- Changes its state as specified by the program.

Turing Machine

- **Initial conditions**: entire input string w is present on the tape surrounded by infinite number of blanks.
- **Final state**: if TM halts in final state then it accepts w
- TM halts in a non final state w is rejected.
- In general a transition is expressed as: $\delta(q, X) = (p, Y, D)$,
  - q: current state,
  - X: TM's RW-head at tape symbol X
  - Y: Output symbol, RW-head erases X and replaces it by Y.
  - p: New state
  - D: could be R or L specifying movement of RW-head

Computation versus Language

- **Calculation**: Takes an input value and outputs a value.
- **Language**: A set of string meeting certain criteria.
- So, language for a calculation basically a set of strings of the form ”<input, output>”, where output correspond to value calculated from the input.

Computation versus Language

L_{add} could consists of strings

<0+0, 0>	<0+1,1>	<0+2,2>	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
<5+7, 12>	<5+8,13>	<5+9,14>	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Membership question: Verifying a solution <13+12, 25> belongs to L_{add} or not?

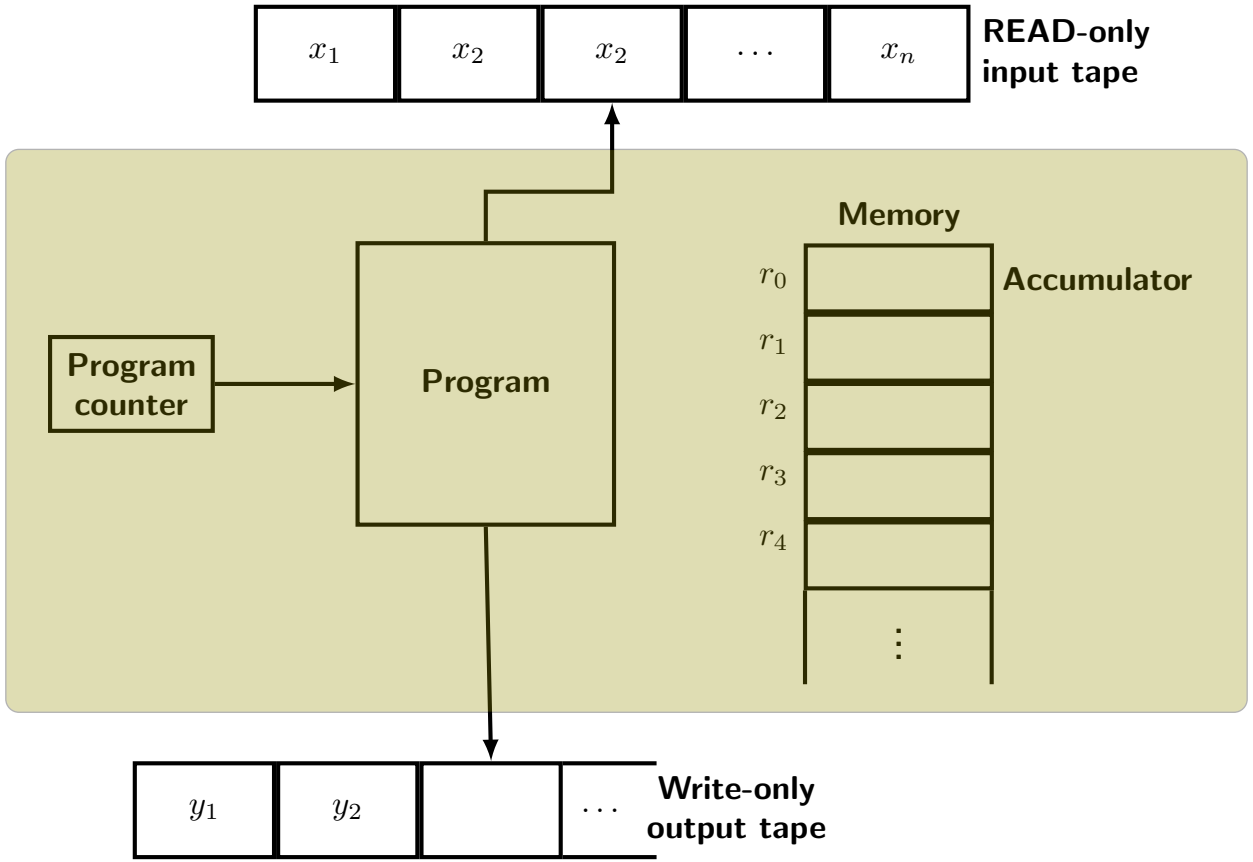
Random Access Machine

- Disconnect between a TM and real computer is sequential tape vs random access memory.
- A RAM is a simplified abstraction of real world computer
  - It has an unbounded memory and capable of storing an arbitrarily large integer in each memory cell.
  - A RAM can access content of any random memory cell.
  - However, to access a random cell, RAM needs to read the address for the cell in a different register.
  - For description of algorithms it is practical to use RAM, since it is closest to a real program.

RAM Model

- Instructions are executed sequentially.
- Impractical to define instructions of each machine, and their corresponding costs.
- Therefore, a set of commonly found instructions in a computer are assumed:
  - **Arithmetic**: ADD, SUB, MULTI, DIV,
  - **Data movement**: LOAD, STORE, WRITE, READ
  - **Control**: JUMP, JGTZ, JZERO, HALT.
- Assume each instruction takes one unit of time.
- A RAM program is not stored in memory of RAM, so instructions cannot be modified.

RAM Model


x_1
x_2
x_2
$\cdots$
x_n
READ-only
input tape
Memory
r_0
Accumulator
r_1
r_2
r_3
r_4
Program
counter
Program
y_1
y_2
$\cdots$
Write-only
output tape

RAM Model

- Programs of RAM not stored in the memory, so cannot be modified.
- All computation take place in register r_0 (accumulator)
- An operand can be one of the following type:
  - Immdiate Addressing ($= i$): integer i itself.
  - Direct Addressing (i): $c(i)$ contents of register r_i.
  - Indirect Addressing ($*i$): $c(c(i))$, if $c(c(i)) < 0$, machine halts.
- Initially $c(i) = 0$ for all $i \geq 0$.
- LC (PC) is set to first instruction of program P.
- After execution of k instruction LC = $k + 1$, automatically unless k instruction is JUMP, JGTZ, or JZERO.

Meaning of an Instruction & Program

- Value $v(a)$ of an operand a is defined as follows:
  - $v(=i) = i$, $v(i) = c(i)$, $v(*i) = c(c(i))$.
- Program is essentially defines a mapping of input tape to output tape.
- Since, program may not halt for some input, the mapping is only partial.