EXTENSIONS OF THE TURING MACHINE

Dr. A. Beulah

AP/CSE

LEARNING OBJECTIVE

- To Design Turing machines for any Languages (K3)
  - To Understand the concept of Turing Machine

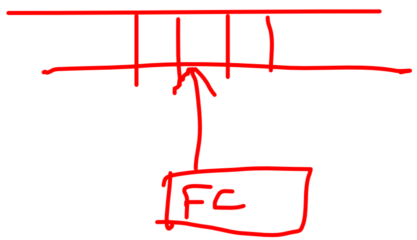
POSSIBLE EXTENSIONS

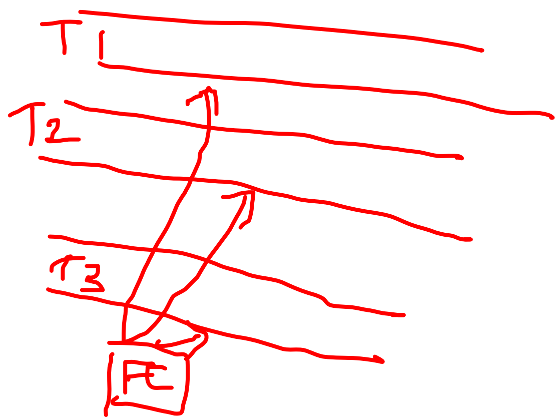
- Multiple tapes
- Two-way infinite tapes
- Two-dimensional tapes
- Multiple heads
- Random access
- Nondeterministic

MULTIPLE TAPE TURING MACHINE

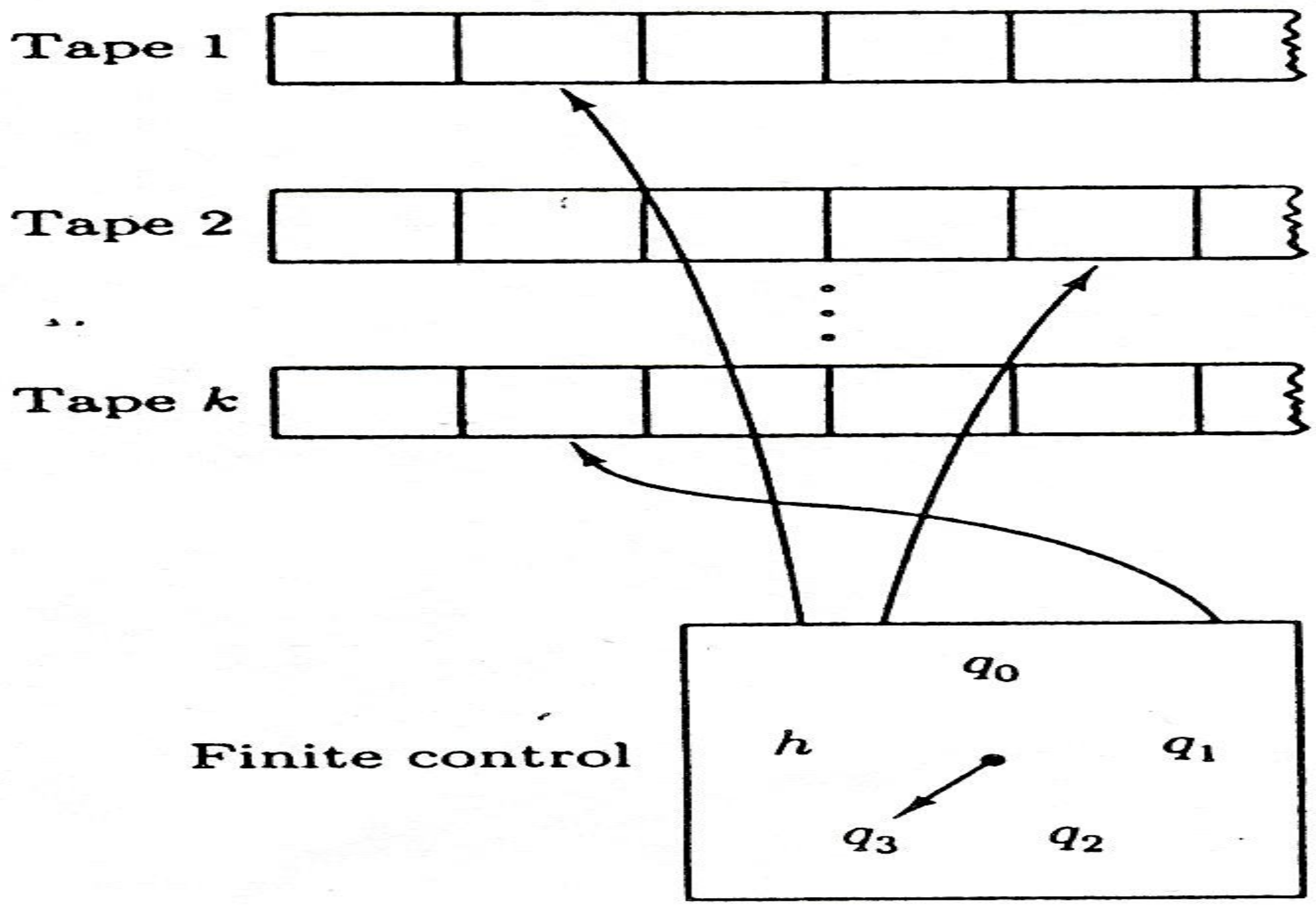
- Each tape is connected to the finite control by means of a read/write head
- For any fixed integer k ≥ 1
  - A k-tape Turing machine is a Turing machine equipped with k tapes and corresponding heads

Multitrack one head





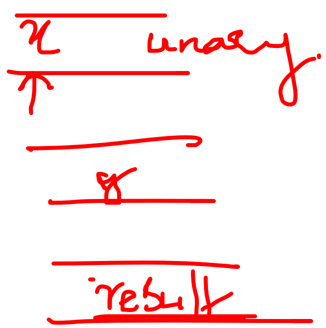
MULTIPLE TAPE TURING MACHINE



EX FOR THE MULTIPLE TAPE TM

x + y

- X on first tape, y on second tape and results written to third tape
  - – move 1 and 2 heads to right end, move head 3 to right max (|x|,|y|)
  - – move tape 3 head right one bit for overflow



EX FOR THE MULTIPLE TAPE TM

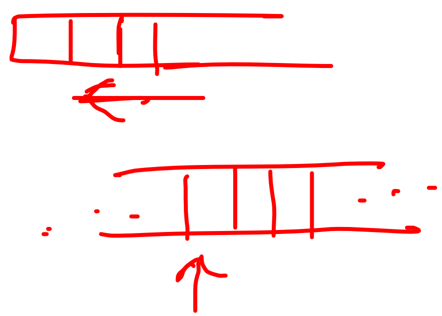
- add 1 and 2, bit by bit and writes each intermediate result as follows
  - reads bits at 1 and 2 plus carry from previous bits
  - if sum is 0 or 1, write it to tape 3
  - if sum is 2 or 3, set carry and write 0 or 1 on tape 3
- if one string ends (beginning of tape marker) use 0 for that input and do not move that head

USAGE OF MULTIPLE TAPE TM

- The use of a k-tape Turing machine:
  - computing a function
  - deciding or semideciding a language

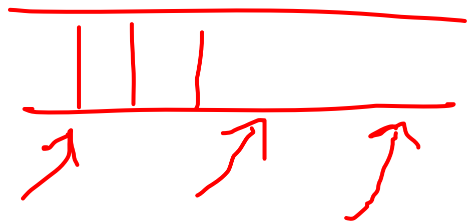
TWO-WAY INFINITE TM

- The tape is infinite in both directions
- All squares are blank (exception: those containing the input)
- It can be simulated by a 2-tape machine:
  - Tape 1: contains the part of the tape to the right of the square containing the first input symbol
  - Tape 2: contains the remaining part of the tape to the left.



MULTIPLE HEADS TURING MACHINE

- Uses a single tape and multiple heads
- In any state only one head can write or move
- The heads all sense the scanned symbols and move or write independently



MULTIPLE HEADS TURING MACHINE

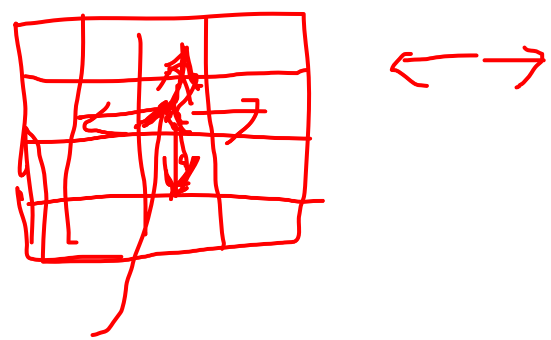
- L = ($a^n b^n c^n$ | n = 0, 1, 2, …)
  - given string w, position first head at beginning of input
  - position second head past all a's to the first b
  - position third head past all a's and all b's to first c
  - enter loop verifying that, on each iteration, head 1 reads an a, head 2 reads a b and head 3 reads a c
  - if third head reaches end of input string at the same time head 1 reads the first b and head 2 reads the first c, machine erases the input string and writes a 1 into cell 1 to signify acceptance

$a^2b^2c^2$

→ a a b b c c B

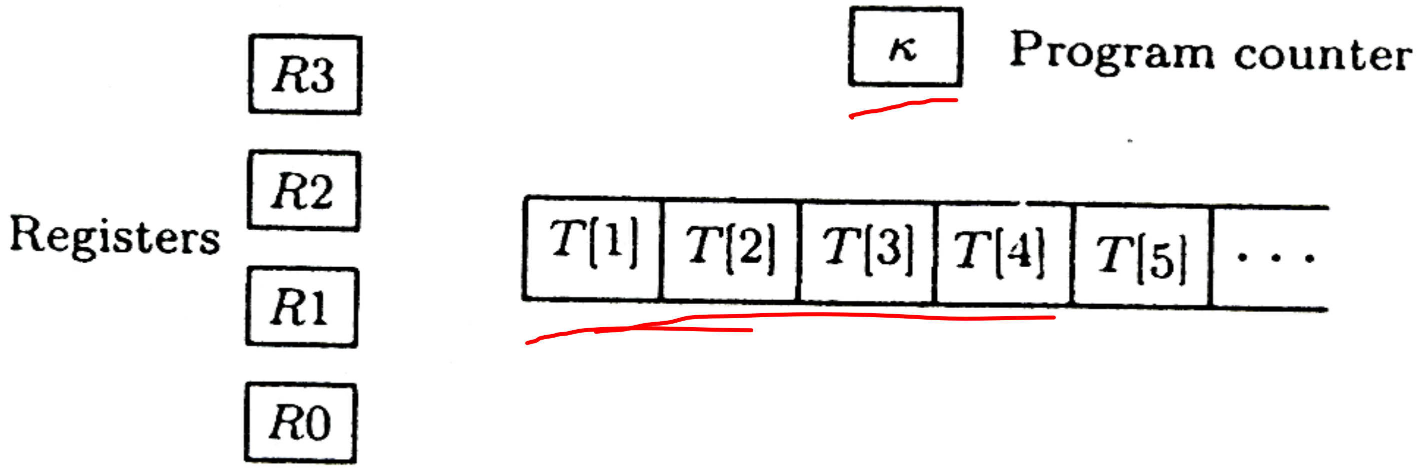
2D TAPE TURING MACHINE

- the input string is placed on the first tape, such as in case of a standard Turing machine
- tape: an infinite two-dimensional grid
- one head on a two dimensional grid that could expand indefinitely down and to the right
- head can move in four different directions
- end of tape markers on left and top sides



RANDOM ACCESS TURING MACHINES

- A random access Turing machine has:
  - a fixed number of registers
  - a one-way infinite tape
  - a program counter
  -



SEQUENCE OF INSTRUCTIONS

Instruction	Operand	Sematics
read	j	$R_0 := T\,[R_j]$
write	j	$T\,[R_j] := R_0$
store	j	$R_j := R_0$
load	j	$R_0 := R_j$
load	=c	$R_0 := c$
add	j	$R_0 := R_0 + R_j$
add	=c	$R_0 := R_0 + c$
sub	j	$R_0 := \max\{R_0 - R_j, 0\}$
sub	=c	$R_0 := \max\{R_0 - c, 0\}$
half		$R_0 := [R_0 / 2]$
jump	s	$k := s$
jpos	s	if $R_0 > 0$ then $k := s$
jzero	s	if $R_0 = 0$ then $k := s$
halt		$k := 0$

SEQUENCE OF INSTRUCTIONS

- j stands for a register number, $0 \leq j < k$
- T [i] denotes the current contents of tape square *i*
- *Rj* denotes the current contents of Register *j*
- $s \leq p$ denotes any instruction number in the program
- *c* is any natural number
- All instructions change *k to k+1*, unless explicitly stated otherwise

EXAMPLE

- program of a random access Turing machine, deciding the language $\{a^n b^n c^n : n \geq 0\}$.

acount := bcount := ccount := 0, n :=1

while T[n] = 1 do : n := n +1, acount :=acount +1

while T[n] = 2 do : n := n +1, bcount :=bcount +1

while T[n] = 3 do : n := n +1, ccount :=ccount +1

if acount = bcount = ccount and T[n] = 0 then accept else reject

EXAMPLE

- We are assuming here that E(a) = 1, E(b) = 2, E(c) = 3
- We are using the variables acount, bcount, and ccount to stand for the number of a’s, b’s, and c’s
- We are also using the abbreviation accept for ”load =1, halt” and reject for ”load =0, halt”

NONDETERMINISTIC TURING MACHINES

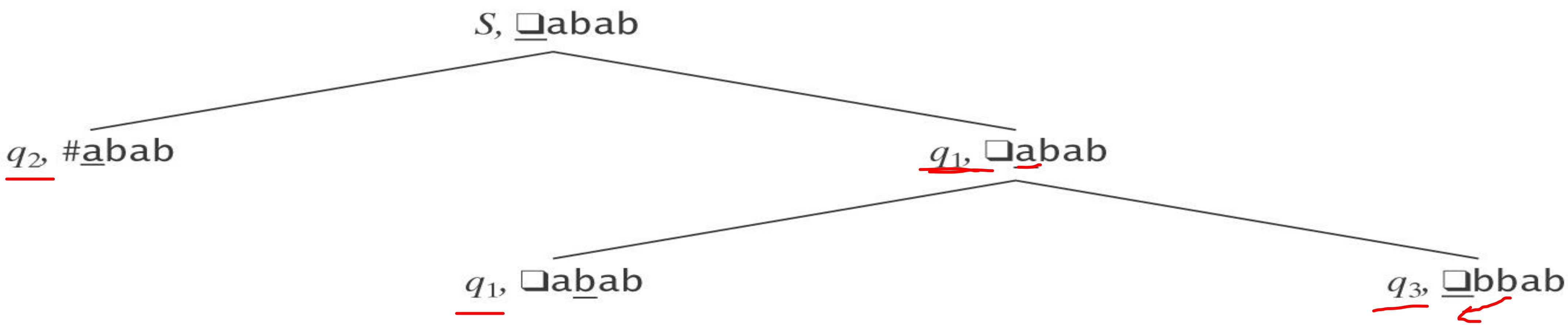
- At any state it is in and for the tape symbol it is reading, can take any action selecting from a set of specified actions rather than taking one definite predetermined action.
- Formally a **nondeterministic Turing machine** is a Turing machine whose transition function takes values that are

$Q \times \Gamma \rightarrow$ subsets of $(Q \times \Gamma \times \{L,R\})$

$Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$2^{(\quad)}$

NONDETERMINISTIC TURING MACHINES



SUMMARY

- Extentions of Turing Machine

REFERENCE

- Hopcroft J.E., Motwani R. and Ullman J.D, “Introduction to Automata Theory, Languages and Computations”, Second Edition, Pearson Education, 2008