Module-5
Turing Machine

Suppose language

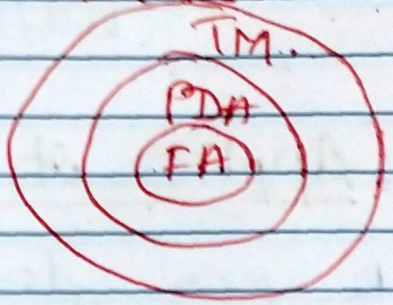
$a^n b^n c^n$, $n \geq 1$

comparision increases here.

so, Push Down Automata fails here. so we required more powerful machine.

Here, the introduction of Turing machine comes in picture.

Which is having 2 stacks.



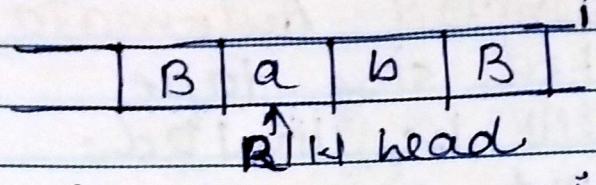
Turing Machine :-

seven Tuples.

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q - set of finite states.
Σ - input symbols.
Γ - symbols allowed on i/p o/p taps. $\Sigma \subseteq \Gamma$ because i/p symbols are always allowed plus some extra symbols.
q_0 - initial state.
B - Blank symbols
F - set of final states.

Subject _____ Date _____



i/p - o/p tape will have i/p symbols plus blank.

R/W head

in PDA only Reading was allowed.
in TM Read & Write both allowed.
R/W head can move into left and right direction

$$\delta = Q \times \Gamma \rightarrow Q \times \Gamma \times (L, R)$$

ese. $(q_0, a) \rightarrow (q_1, x, R)$

Standard / Deterministic TM.

- for single i/p symbol we can either go into right direction or left direction

- for Non-Deterministic TM

$$2^{Q \times \Gamma \times (L,R)}$$

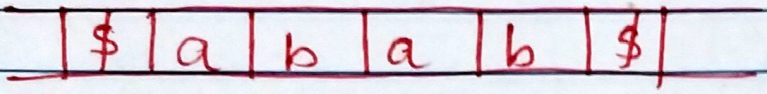
ese. $(q_0, a) \rightarrow (q_1, a, R)$
$\hookrightarrow (q_0, x, L)$ } two choises.

LBA (Linear Bounded Automata)

- TM with limited size tape
- i/p o/p tape will be limited dependent on input size.

LBA = TM + Input size tape

-



Based on power LBA stands

FA ⊂ PDA ⊂ LBA ⊂ TM.

- FA → Regular Language
- PDA → Context Free Lang.
- LBA → (CSL) Context Sensitive Language
- TM → Recursive or Recursive Enumerable language

FA → Deterministic
→ Non-Deterministic

PDA → Deterministic
→ Non-Deterministic

TM → Deterministic
→ Non-Deterministic.

LBA → Only single

Subject _____ Date

Standard examples of LBA

1. $L = \{a^n b^n c^n, n \geq 1\}$
2. $L = \{a^n, n \text{ is prime}\}$
3. $L = \{a^n, n \text{ is non-prime}\}$
4. $L = \{a^{n!}, n \geq 0\}$
5. $L = \{ww, w \in (a,b)^+\}$
6. $L = \{www^R, w \in (a,b)^+\}$
7. $L = \{w^n, w \in (a,b)^+, n \geq 1\}$
8. $L = \{a^n, n = n^2, n \geq 1\}$

Example 1

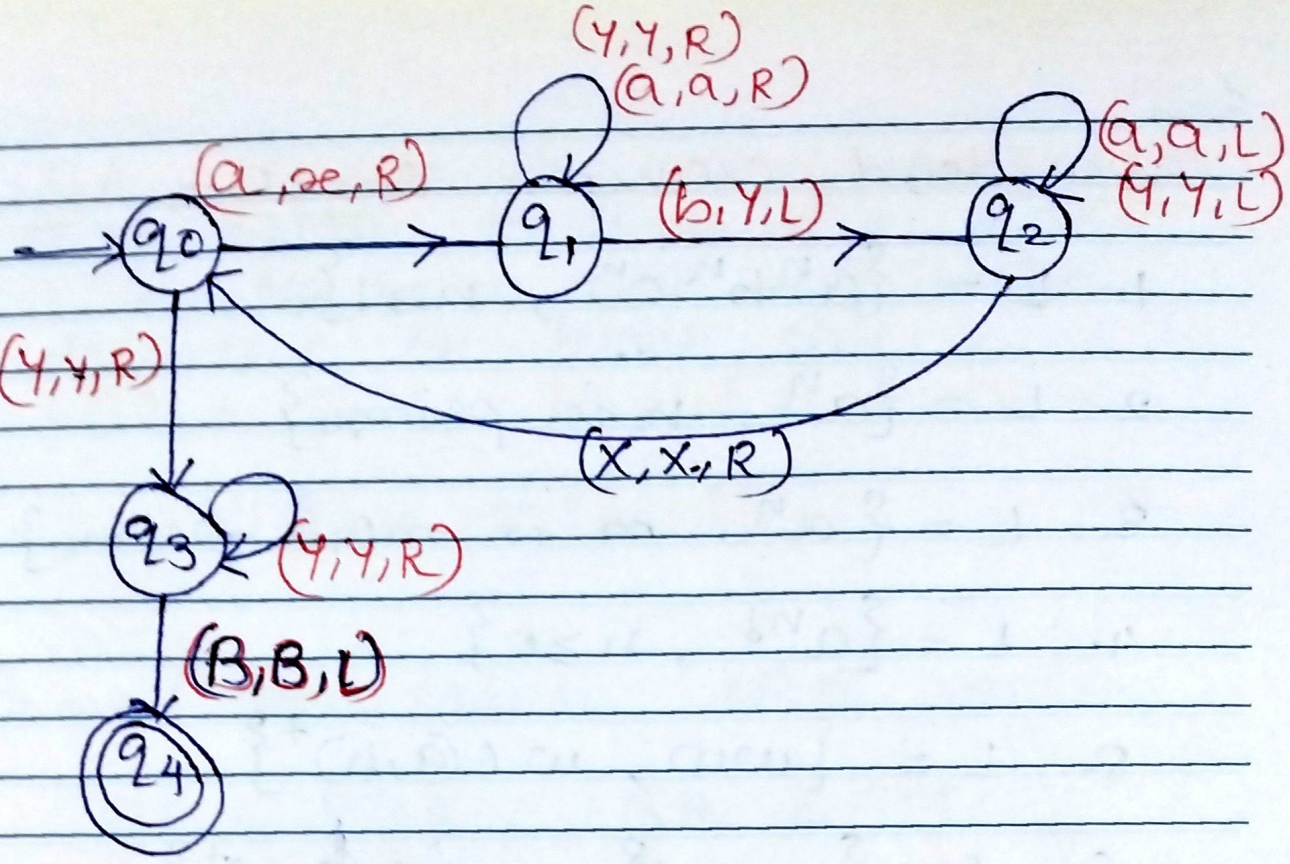
Design a TM for $\{a^n b^n / n \geq 1\}$

L = B|a|a|b|b|B

also can be

B	B	a	a	a	b	b	b	B	B
		↑							
		R/W head							

Here, the Read/Write head can move into right & left direction as per need.



String aabbb will not be accepted. You will stuck at q_3 only.

Example 2

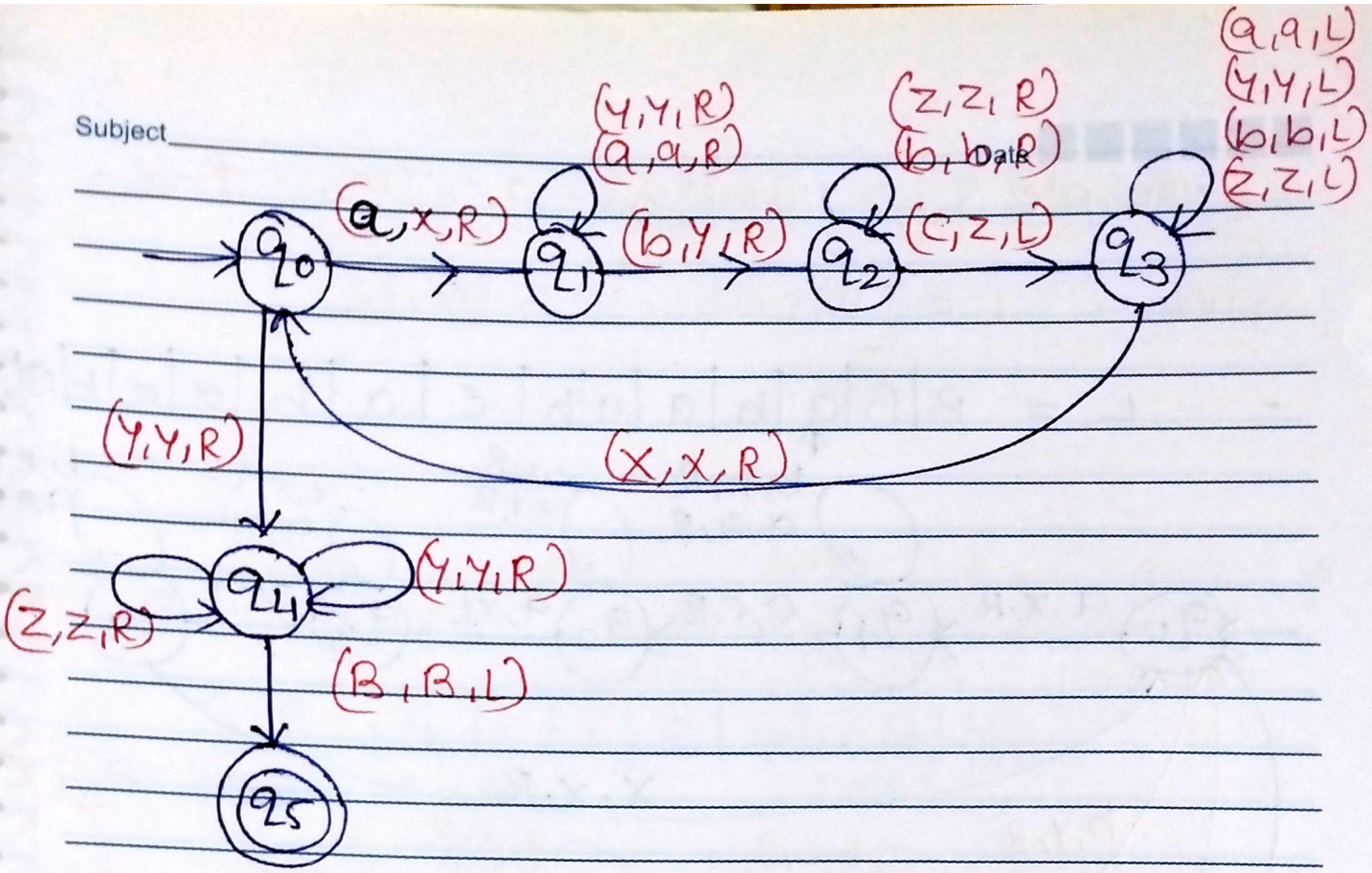
Design a Turing Machine for $\{a^n b^n c^n \mid n \geq 1\}$

L =

B	B	a	a	a	b	b	b	c	c	c	B	B

↑R/W head

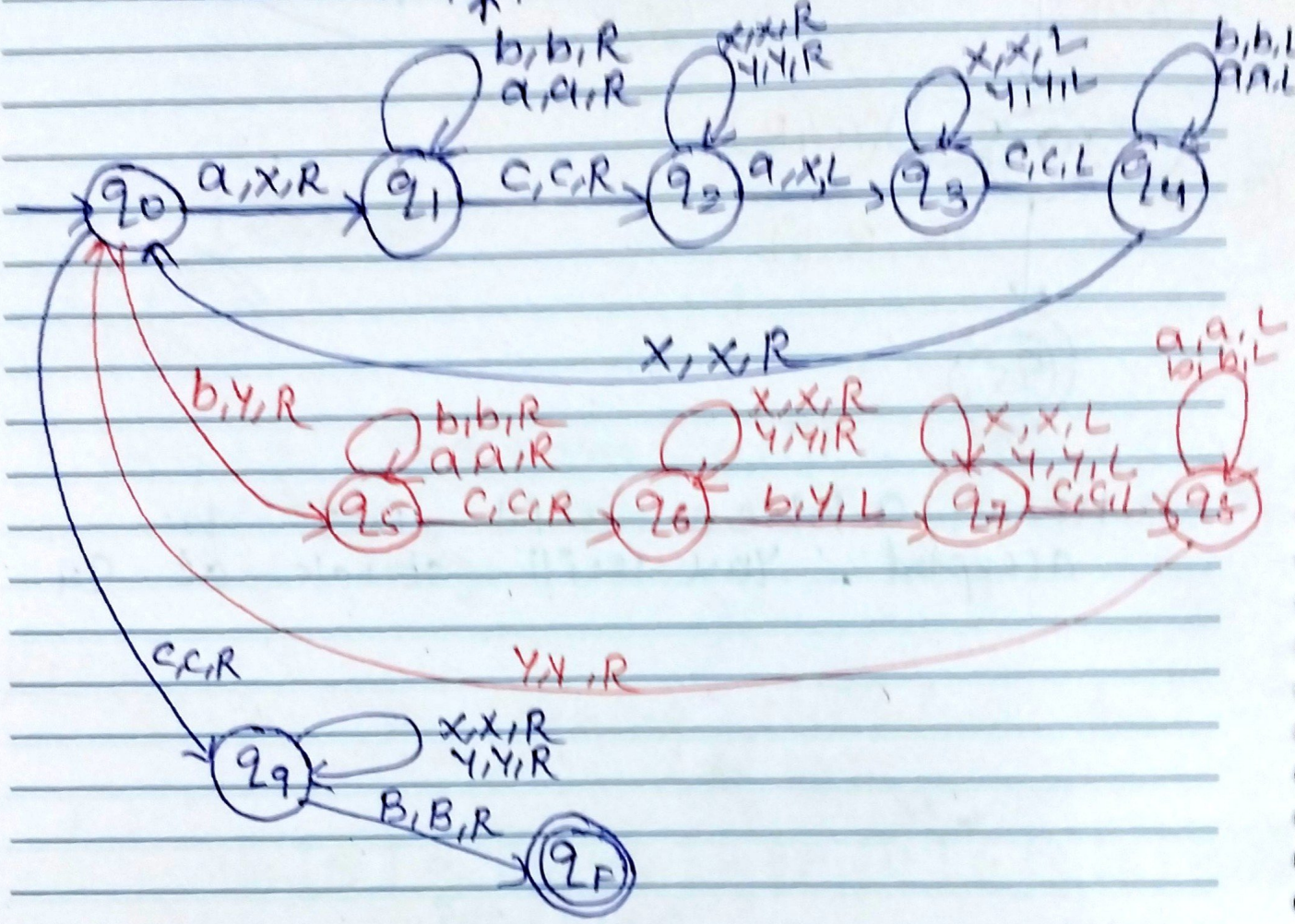
Subject



Here, abbcc will not be accepted. You will stuck at q_4.

Example 3 Design a TM. fn given $L = \{wcw \mid w \in \{a,b\}^*\}$

- L = B|B|a|b|a|a|b|c|a|b|a|a|b|B|B

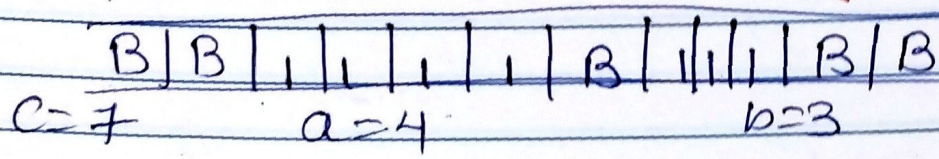


* Turing Machine is language acceptor as well as language generator and language transducer also.
- Transducer — can perform mathematical ops. addition, Multiplication, ~~tttt~~ Binary op, etc..

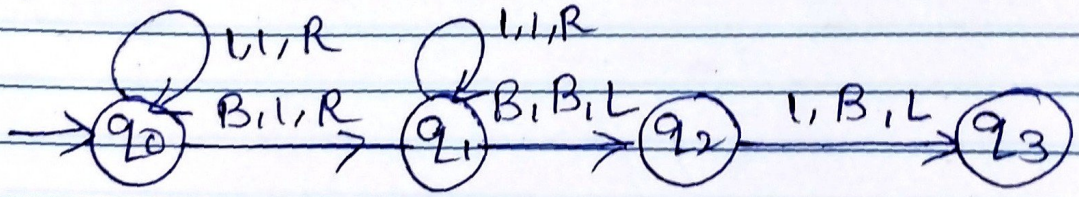
Subject _____ Date _____

* Turing Machine as adder

Ex:- perform addition opn for a=4 & b=3 using unary opn.



Replace middle B by 1 till B in right direction. & then come back to left direction



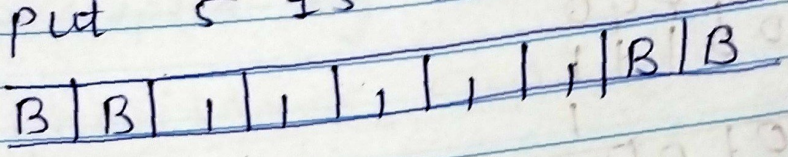
Subject

Date

Ex. TM as unary to binary converter

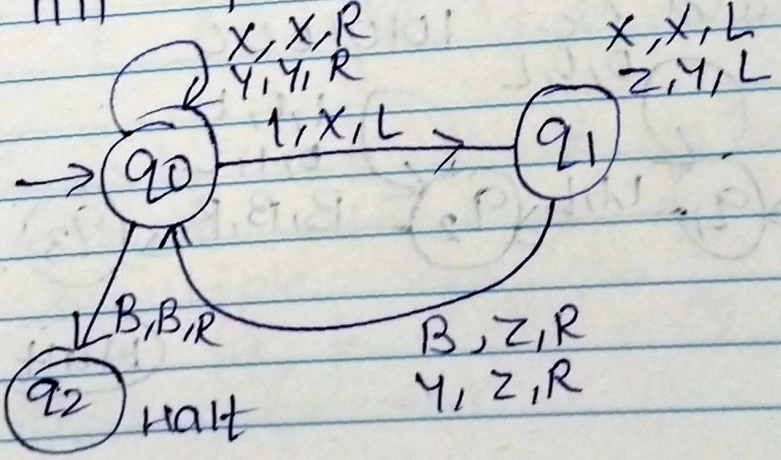
Unary number represented with the help of that many times of 1.
Suppose decimal value = 5
Then unary 5 = 11111

put 5 1's on i/p o/p tape of TM.



in binary 5 = 101.
Here to represent
0 - Y will be used
and 1 - Z will be used.
So, final answer 101 = ZYZ.

Unary	Binary
1	1
11	10
111	11
1111	100



Date

Turing Machine for 2's complement

2's complement means 1's complement plus 1.

~~1 0 1 0 1 0 0 0~~

~~1's comp 0 1 0 1 0 1 1 1~~

~~+ 1~~

~~0~~

Ex. 0 1 0 1 1 0 0 0

1's comp 1 0 1 0 0 1 1 1

\+ 1

1 0 1 0 1 0 0 0

Observation

i/p - 0 1 0 1 1 0 0 0 NO change till 1

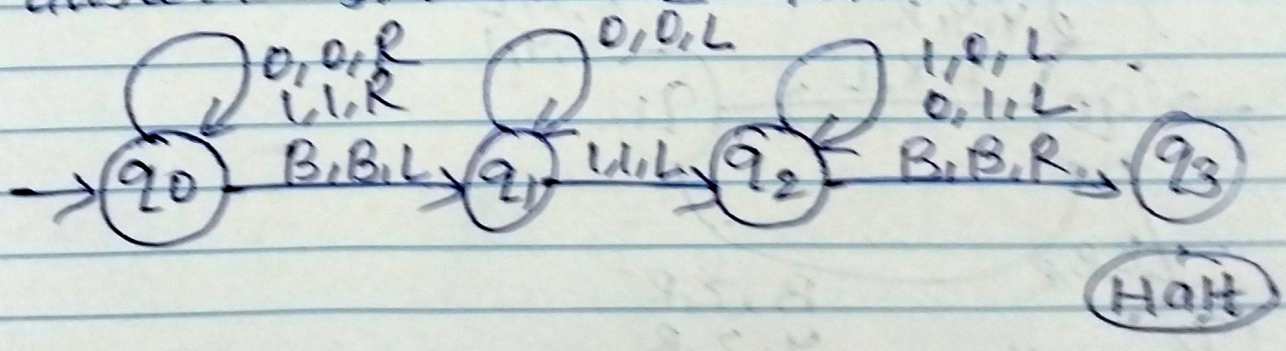
o/p - 1 0 1 0 1 0 0 0

↳ after 1 comes digits are altered.

B	B	0	1	0	1	1	0	0	0	B	B

↑

answer should be 10101000



Subject

* Turing

Subject

Date

Variations of TM

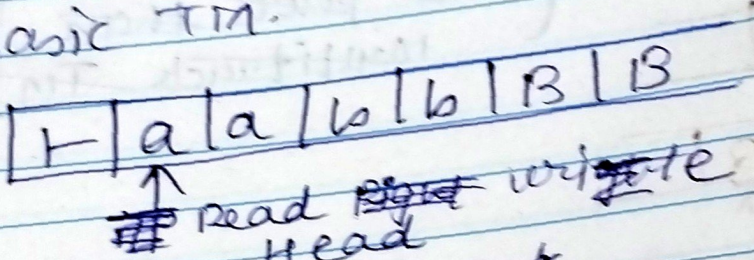
Modification in the structure of TM

But the power of TM will not be changed.

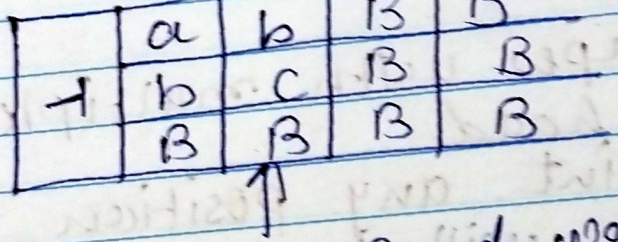
If the language is accepted by basic TM then that will be accepted by variations & vice versa.

1. Multitrack Turing Machine

Basic TM



But in Multitrack multiple tracks are used.



- Only one Read write head
- Here at a time the head can see multiple symbols

$\vdash$ $(q, b, c, B) = (q_{next}, x, y, z, R)$

Transition fun

Example 3 40 min.

Date

Basic TM + multitrack both are same power of TM.

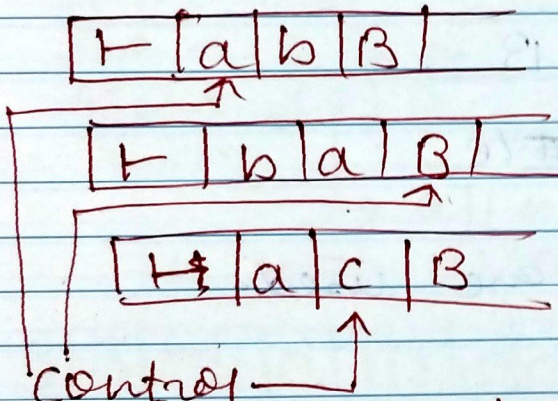
⊢	a	a	b	B
	b	c	c	B

$\Sigma = \{a, b, c\}$

⊢	a	a	b	B	B
	b	c	c	B	B

$\Sigma = \{\begin{matrix} a & a & a & b & b \\ b & a & c & a & b \end{matrix} \cdots\}$

2. Multitape TM (more power than multitrack TM.)



multiple tapes with multiple Read write head

Head can point any position

All things will be same as TM.

Transition fun will be diff.

$(q, a, B, c) = (q_{next}, x, y, z, L, R, L)$

All heads can have different moves.

Universal TM

A TM for all other TM.

The language

$A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a Turing machine and accepts } w \}$

is Turing Recognizable

A_{TM} – language

It is Recognizable but not decidable

Because we don't know what this TM is doing with different type of language. accepts, rejects or in loop.

So, TM is always recognizable but not decidable.

Given the description of TM and some input can we determine whether the machine accepts it?

- Just simulate / run the TM on the input.

M Accepts w – Our algo. will halt & accept
M Rejects w – Our algo. will halt & reject
M loops on w – Our algo. will not Halt.

Subject _____ Date

Input:-
M - description of some TM.
w - an input string for M.

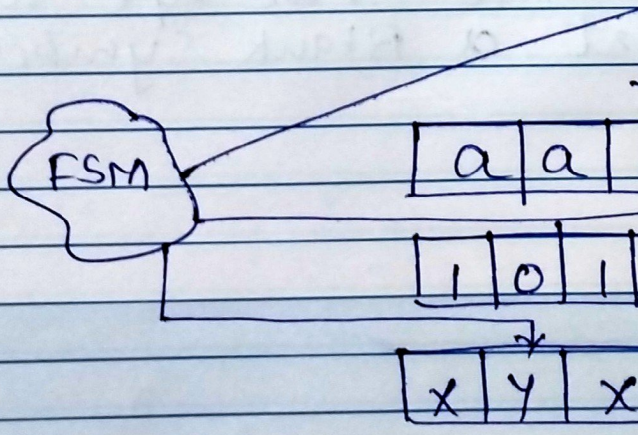
Action:-
Simulate M.
Behave just like M would (may accept or reject)

The UTM is a recognizer but not a decider for
$A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \}$

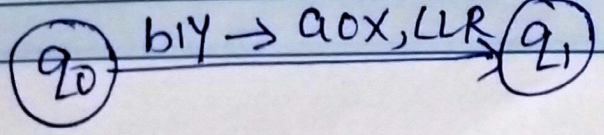
* Equivalence of single and Multitape TMs.

→ Every Multitape TM has an equivalent single Tape TM.

Given a Multitape TM show how to build a Single Tape TM.

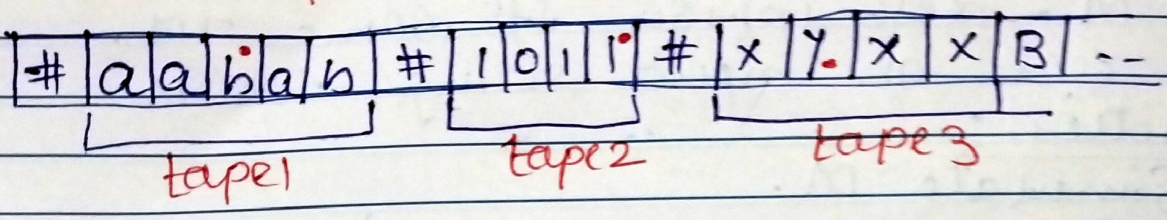


several Tapes Each has its own Tape Head



An example machine with k=3 Tapes

Single tape Turing Machine



- Add "dots" to show where head 'k' is
- By process of marking we can put dots
- To simulate a transition from state q, we must scan our tape to see which symbols are under the k tape head
- Once we determine this and are ready to make the transition, we must scan across the tape again to update the cells and move the dots.
- whenever one head moves right end, we must shift our tape so we can insert a Blank symbol.