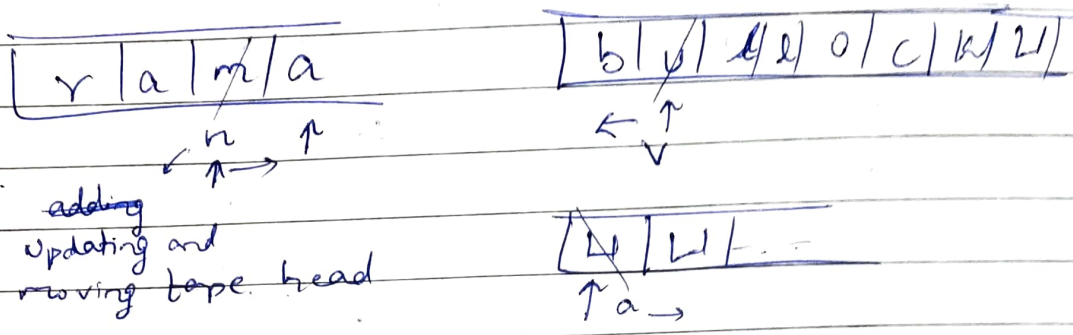
referring to ramu, bullock example. let the transistion of it be to



so M_2 will be

r a m a # b u l l o c k # a ⊔ # ...

$$\delta(q, [m, u, \sqcup]) = (q', [n, v, a], [R, L, R])$$

So DTM with k tapes also doesn't have extra computational power.
It is same as that of a standard DTM.

start of this lecture

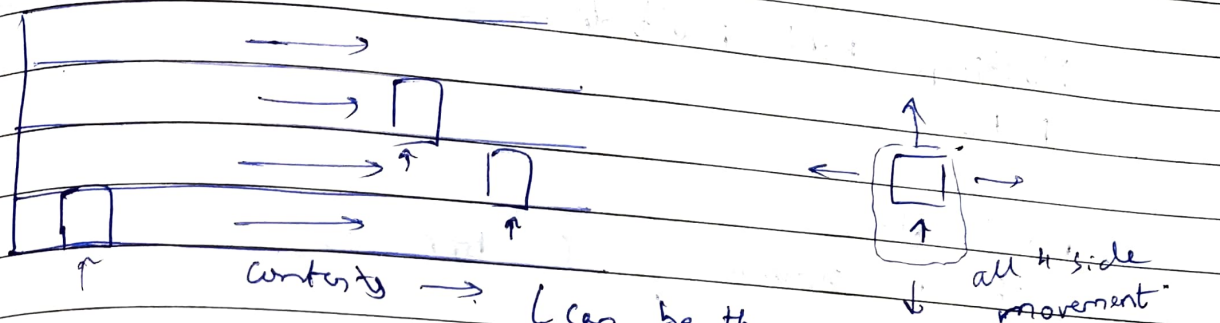
If tape 1 contents are

a b c d ⊔ ⊔ ⊔.

and head is on blank. then we must include it in our single tape.

a b c d ⊔ # ← tape 2 → # ...

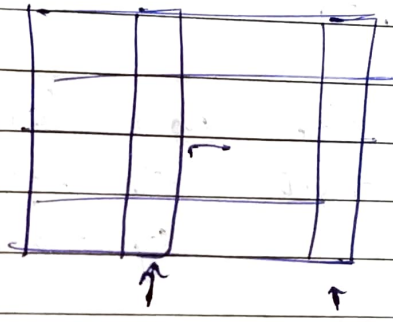
eg Multi dimensional DTM



(can be thought of a matrix, where we can move from one cell to another)

head position shifts to these individual cells.
head can move up, down, left, right.

Multi track means tapes are multiple but whole column has one head



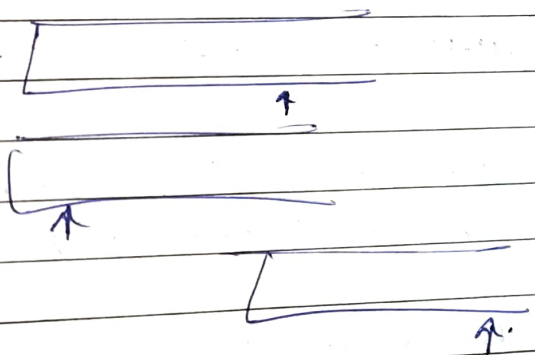
(head shifts like this)

Symbols can be thought of as a vector [4x1]

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \rightarrow \begin{bmatrix} k \\ e \\ m \\ n \end{bmatrix}$$

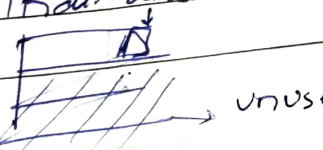
(shifts)

Multi-tape DTM means multiple individual tapes having their own heads



* Simulating multi-tape DTM to multi-dimensional DTM.

Simulate multi-tape to single tape and then just use 1st tape of multidimensional DTM.
So this is simulated

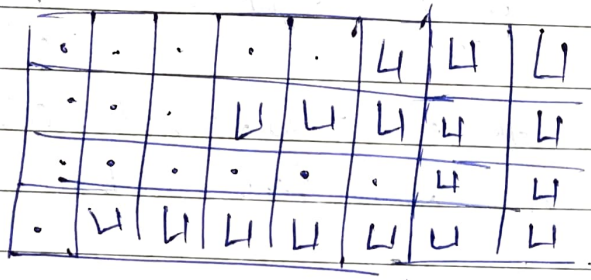


- Simulating

Multitape ~~DTM~~ Multidimensional DTM ~~on~~ multitape DTM.

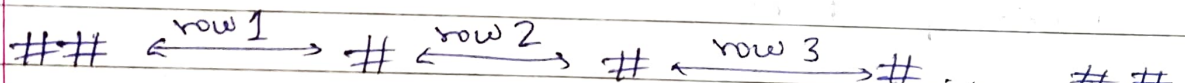
~~non blank~~

if multidimensional DTM is



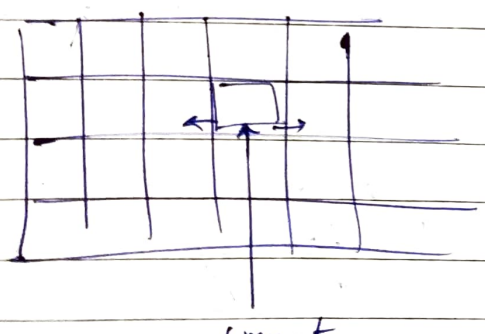
just consider the non blank portion
(max column length = till where it is non-blank/# significant symbol is placed)

I'll place all contents in a single tape as shown below

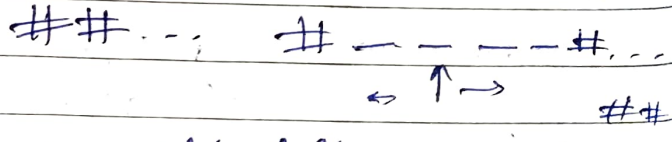


We would have same notation: symbol having current head will have dot on it.

For right, left movements.



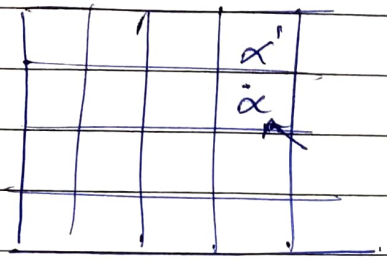
on single tape (using first tape of multitape DTM)

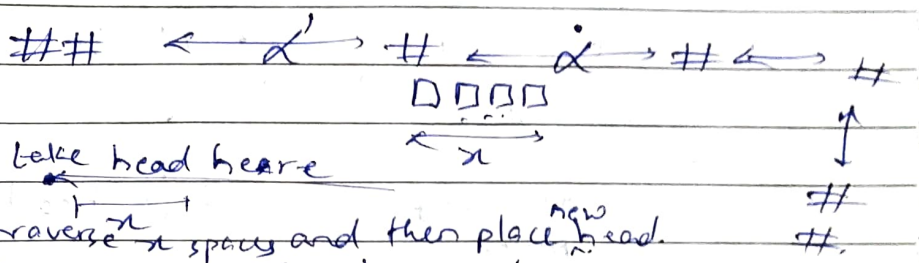


so right, left can easily be simulated

Suppose we have vertical upwards, downwards movement

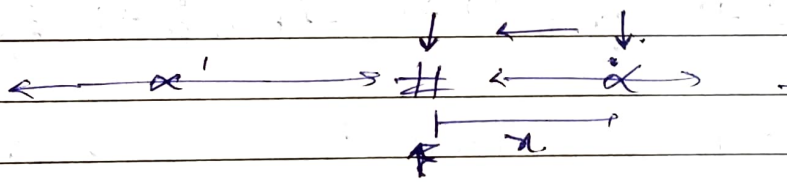
Consider upward movement. α to α'



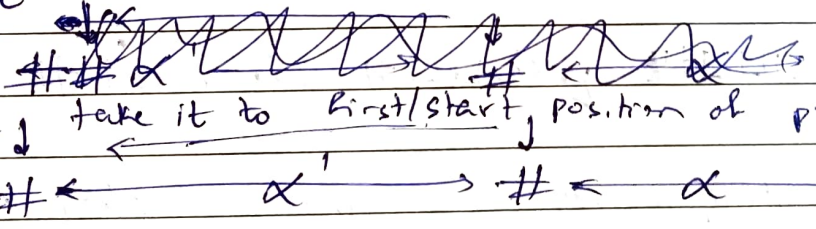


So to shift from $\dot{\alpha}$ to α'
we will calculate no. of symbols before $\dot{\alpha}$ till first sharp let it be x

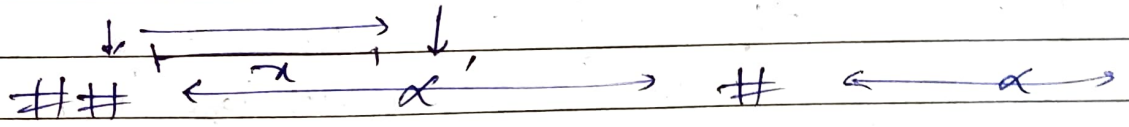
So we will go to previous row, and from starting of it, leave x places and then place our tape head



~~How to~~ look at next page to see how to implement counting



Then



so place dot on α'.

~~Dow~~ Downward movement can also be generated similarly.

Date: YOUVA

~~Non blanks~~ If tape head moves inside the selected non blank portion, then it is fine but what if.

p	q	r	⊔	⊔	⊔	⊔
q	r	s	t	⊔	⊔	⊔
k	l	m	n	a	⊔	⊔
f	g	h	b	⊔	⊔	⊔

our selected matrix

tape head.

What if tape head moves right.

Then we would have to include one more column So all the rows will include one more position

Here.

previously

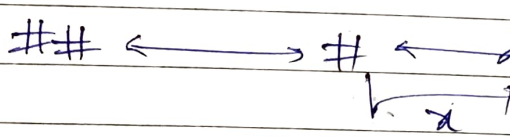
pqr⊔⊔ # qrst ⊔ # klmna # fghb⊔##

New config

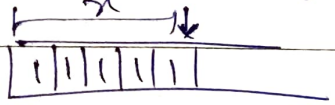
if tape head moves from a to right blank

pqr⊔⊔⊔## qrst⊔⊔ # klmna⊔ # fgh b⊔⊔

To implement counting, we will use another tape (remember we are simulating on multi-tape DTM)



We move from x to one place back, on second tape move one place forward. similarly till we move till #, move ahead on second tape, and finally we have our count.



(unary representation) or counting can be done any way.

- We are doing this simulations to make proofs easy.
Like if we have some partial recursive function and we are asked to simulate it in DTM.
So then we could use multitape, multidimensional tape models to easily simulate, rather than having difficult time proving it only on single tape DTM. And all multi-tape, multi dimensional tape, single tape, double ∞ tape are interconvertible.

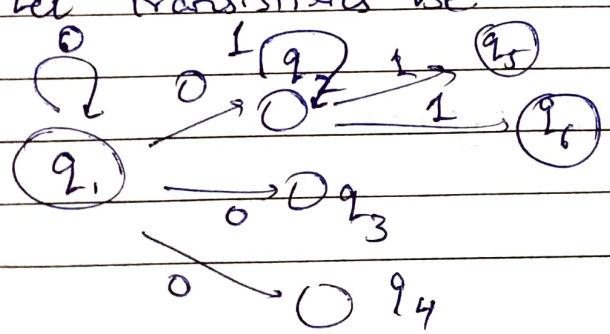
- Simulating NTM (single tape) on multi-tape DTM.

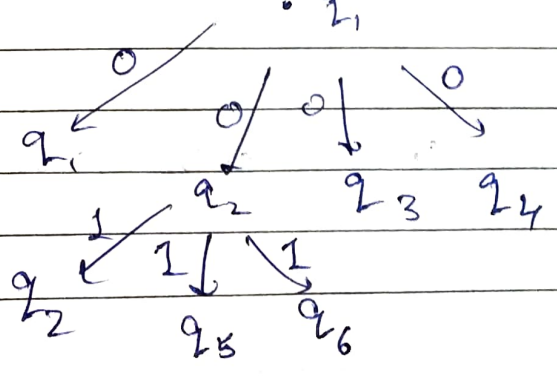
Firstly using single tape NTM to simulate single tape DTM.

If NTM has tape, non-determinism (the guesses) aren't involved in DTM, then single tape NTM is same as single tape DTM.
and thus single tape NTM can also be used to simulate multitape DTM
Now other way round. Using multi tape DTM to simulate NTM

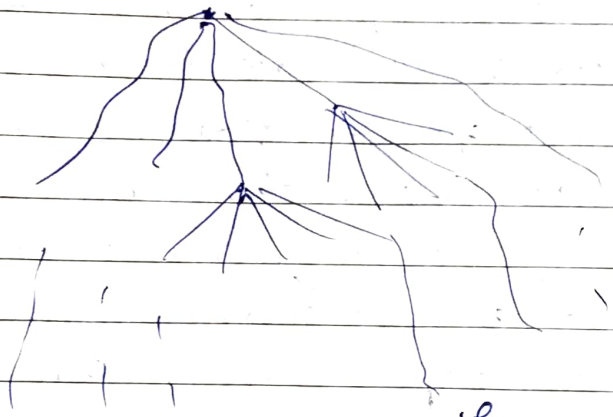
Lets suppose there is a NTM. corresponding to NTM, this is called computation tree.
input w = 101011.
we are here (q_1)

Let transistions be.





for some input symbols, there is a computation tree.



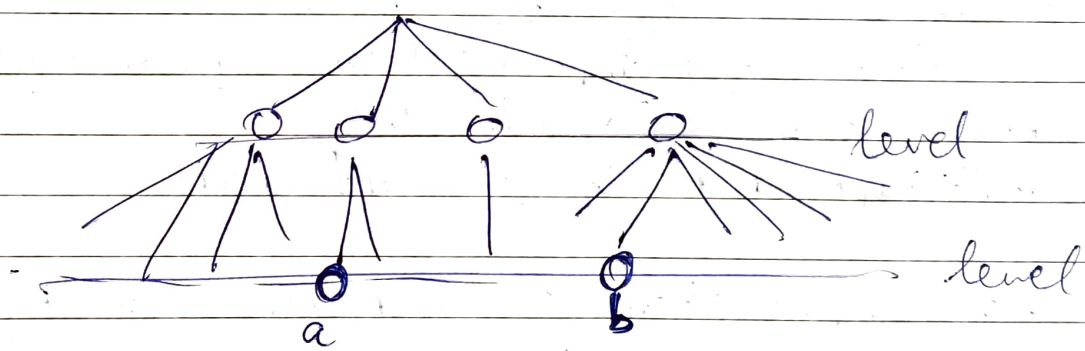
so if a branch ends in accept state for input it is accepted and all other branches are killed

If a branch ends in reject state, it kills itself, all others continue to run

It may also happen that a branch may not halt, continuously loop, so there we may not reach end of branch and continue it infinitely.

~~Due to non halting possibility~~

So then instead of traversing branch by branch, we go level by level



same levels are explored together

If suppose we are exploring node a tape would be like

| 2 | 1 |

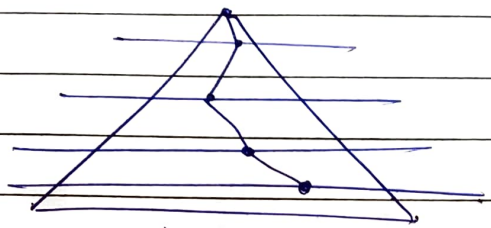
2nd branch → 1st branch so on

for b node | 4 | 2 |

So we can use multi-tapes to explore different paths.

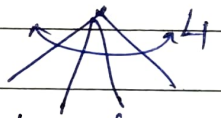
- Some arbitrary case, reaching final accept state.

| 4 | 2 | 5 | 8 |



computation tree.

Suppose maximum branching in tree is b



branching is 4.

different tapes look like. (4 levels)

0,0,0,0 traverse lexicographically.

0,0,0,1

0,0,0,2

0,0,0,b

0,0,

So each element will go from 1-b.

1, 1, 1, 1

1, 1, 1, 2

:

1, 1, 1, b

1, 1, 2, 1

1, 1, 2, 2

:

1, 1, 2, b, similarly

It may happen that some id (1, b, b, b for eg) does not exist, then it will be rejected and next id will be traversed. Hence simulation done //.