~~Data~~ → Fill ~~cells~~ one diagonal in each step.

Step 1 :

Fill the first diagonal (1,1) (2,1) (3,1) (4,1) (5,1)

NOTE :

* Since b, can be produced only by X, we write only X in first cell. and a by X, Y, A

Step 2 :

- Fill second diagonal (1,2) (2,2) (3,2) (4,2)

- Take cartesian product to calculate value of cells - with left, and top cell.

For (1,2)

$(1,2) = (1,1) \times (2,1)$

$= \{X\} \times \{X, Y, A\}$

$= \{XX, XY, XA\}$

$\phi \cup S \cup X = \{S, X\}$

→ Check if these symbols exist at right side of any production

For (2,2)

$(2,2) =$ ~~(1,2)~~ $(2,1) \times (3,1)$

Date: Day:

$\{X, Y, A\} \times \{X\}$

$\{XX, Y\cancel{X}, AX\}$

$\emptyset \cup \emptyset \cup \emptyset \rightarrow \{ \}$

For (3,2)

$(3,2) = (3,1) \times (4,1)$

$= \{X\} \times \{X, Y, A\}$

$= \{XX, XY, XA\}$

$\emptyset \cup S \cup X \quad \{S, X\}$

For {4,2}

$(4,2) = \{4,1\} \times \{5,1\}$

$= \{X, Y, A\} \times \{X, Y, A\}$

$= \{\underset{\emptyset}{XX}, \underset{S}{XY}, \underset{X}{XA}, \underset{\emptyset}{YX}, \underset{\emptyset}{YY}, \underset{\emptyset}{YA}, \underset{\emptyset}{AX}, \underset{Y}{AY}, \underset{\emptyset}{AA}\}$

$= \{S, X, Y\}$

Step 3:

- Fill diagonal 3 (1,3) (2,3) (3,3)

* {Take cartesian product of left most cell with immediate top of a cell}

Union {then immediate left cartesian product with top most}

Date ______ Day ______

For $(1,3)$

$$(1,3) = \{(1,1) \times (2,2)\} \cup \{(1,2) \times (3,1)\}$$

$$= \{\{X\} \times \{\phi\}\} \cup \{\{S,X\} \times \{X\}\}$$

$$= \{\phi \cup \{\underbrace{SX, XX}_{\phi}\}\}$$

$$= \phi$$

For $2,3$

$$(2,3) = \{\{2,1\} \times \{3,2\}\} \cup \{\{2,2\} \times \{4,1\}\}$$

$$= \{\{X,Y,A\} \times \{S,X\}\} \cup \{\underbrace{\{\phi\} \times \{X,Y,A\}}_{\phi}\}$$

$$= \{\{XS, XX, YS, YA, AS, AX\} \cup \phi\}$$

$$= \phi$$

For $(3,3)$

$$= \{S, X\}$$

Date: Day:

Step 4 :

Similar to Step3,

For $(1,4)$

$(1,4) = \{\{\underset{\emptyset}{(1,1) \times (2,3)}\} \cup \{\underset{\emptyset}{(1,2) \times (3,2)}\} \cup \{\underset{\emptyset}{(1,3) \times (4,1)}\}\}$

$= \emptyset$

For $(2,4)$

$= \{ (2,1) \times (3,3) \cup (2,2) \times (4,2) \cup (2,3) \times (5,1) \}$

$= \emptyset$

step 5 :-

For $(1,5)$.

$(1,5) = \emptyset$

Un

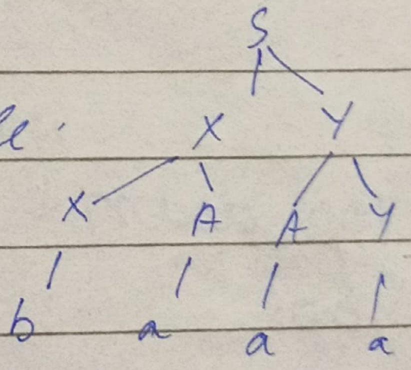
Conclusion :

* If S ~~exists~~ does not exists in step. It means string is not part of language.

* If S exist → part of language.

String baaa

acceptable.



			a
		a	X,Y,A
	a	X,Y,A	S,X,Y
b	X,Y,A	S,X,Y	SXY
X	S,X	S, X	S,X

acceptable, as S present.

Step 1:

b	a	a	a
			a
		a	aa
	a	aa	aa(a) ∪ a(aa)
b	ba	b(aa) ∪ ba(a)	b(aaa) ∪ ba(aa) ∪ baa(a)

aa produced

aaa produced by two ways.

baaa produced here by 3 ways

ba being produced here, one way

baa produced here by two ways

Non-deterministic

→ **Top down ↓ Parser**

Steps

1. Push starting variable on stack.
2. If top of stack element is variable then replace it non-deterministically with right side. ~~else if~~
3. else → top of stack is terminal
   pop if terminal and input are same.
   else
   crash
3. jump to step 2

→ If you asked for PDA, you can't provide parser for it

Date: Day:

$S \rightarrow T\$$

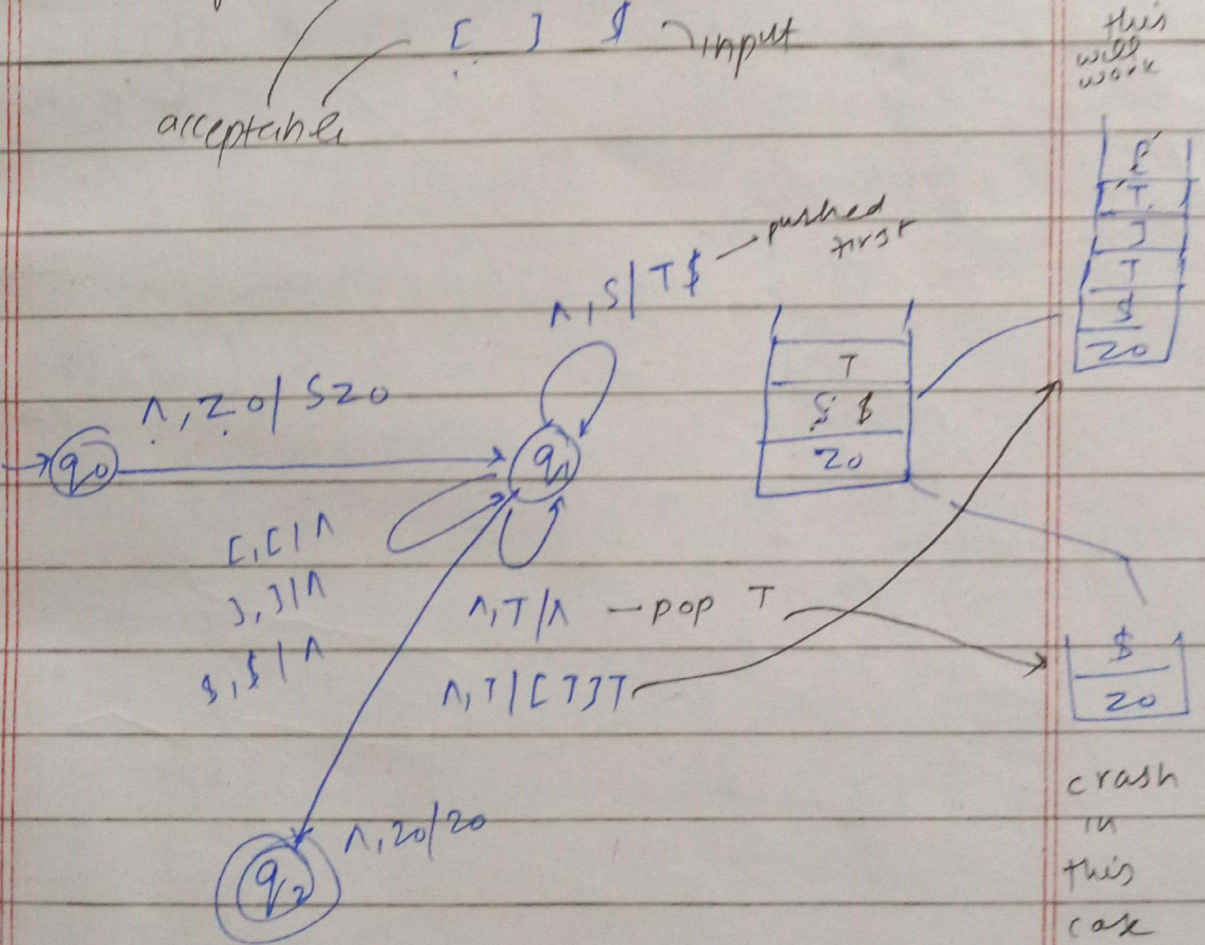
$T \rightarrow [T]T \mid \Lambda$.

Terminal = { $, [,] }

Strings [[[ ]]] $.

[ ] $ → input

acceptable



$(q_0, [\,]\$, z_0) \rightarrow (q_1, [\,]\$, SZ_0)$

$\rightarrow (q_1, [\,]\$, T\$Z_0) \rightarrow$

Splitting of tree indicator of non-determinism parser

Name ____________ Date ____________

Non deterministic

Tree:

$(q_0, []\$, z_0)$

$\downarrow$

$(q_1, []\$, Sz_0)$

$\downarrow$

$(q_1, []\$, T\$z_0)$.

$T \to \Lambda$ $\qquad$ $T \to [T]T$.

$(q_1, []\$, \$z_0)$ $\qquad$ $(q_1, []\$, [T]T\$z_0)$

top of stack

$\downarrow$

crash

$(q_1, []\$, T]T\$z_0)$

$T \to \Lambda$ $\qquad$ $T \to [T]T$

$(q_1,]\$,]T\$z_0)$. $\qquad$ $(q_1,]\$, [T]T]T\$z_0)$

$\downarrow$

crash

$(q_1, \$, T\$z_0)$

$T \to \Lambda$ $\qquad$ $T \to [T]T$

$(q_1, \$, \$z_0)$ $\qquad$ $(q_1, \$, [T]T\$z_0)$

$\downarrow$ $\qquad$ $\downarrow$

crash.

top of stack & input are different

(q_1, Λ, z_0)

$\downarrow$

(q_2, Λ, z_0)

For parser language should be in LL1 form.

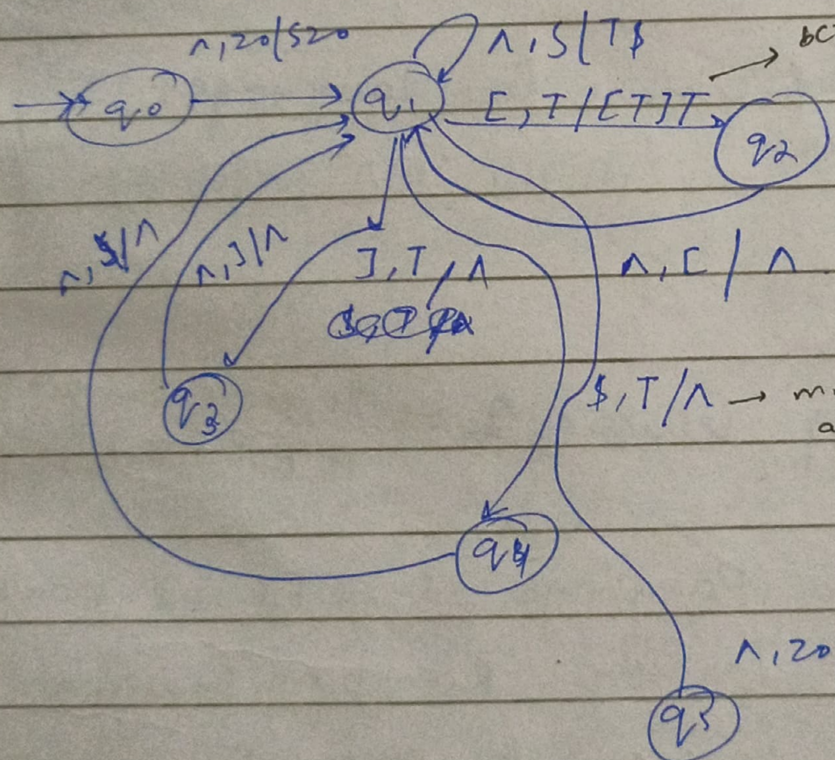
look ahead language 1 — we decide after looking 1 element which is ahead

Date: Day:

How to change it to Deterministic Parser

input	Top of stack	
[	T	T → [T]T
]	T	T → Λ
$	T	T → Λ

Deterministic parser



bcz we consumed our input [here, in next step we don't consume input and pop top of stack.

$, T/Λ → meaning we are replacing T by null.

* We have to understand the rules of non-deterministic parser, by traversing tree on a string.

Conditions for Parser:

① → Grammar should not be ambigous. Only one tree for each string. One branch of tree should accept only.

Date

② No right factoring $\cancel{A \to Ab \mid Ac}$ should not be like this

③ should be in LL1 form

$L = a^i b^i c^i, \; i \geq 0$

↓

Non-context free language. can't do three comparision with help of one stack

↓

can't draw PDA for it.

Pumping lemma 2: — To prove language is not context free

* To prove language is CFL we provide PDA for it. Pumping lemma can't be used to prove language is CFL.

$S \to vAz$

$A \to wAy / x$

acceptable → vxz, $vwxyz$, $vwwxyyz$; vw^3xy^3z

Date: Day:

→ self loop on variable

× max leaf nodes, h starts from 0

$$\boxed{h \rightarrow 2^{h-1}}$$

G → P variables. G → SABC

↓

height → (P+1)

S
↓
A
↓
B
↓
C
↓
terminal]+1

P

$\boxed{\text{leaf nodes}}$

↓

$2^{P+1-1} = 2^P$.

if

→ height ≥ P+2, then ~~leaf nodes~~ → $2^{P+2-1} = 2^{P+1} = n$. string

length of ~~these~~

↓

some variable is repeated → recursion

× if variable is not repeated and is this, it will not be acceptable

(length "n")

S → vAz.

A → x | wAy.

before recursion $v\underbrace{w^i x y^i}_{\text{recursion}} z \in L$. after recursion

for all $i \geq 0$.

× $|u| \geq n$ $n = 2^{P+1}$

① $|wy| > 0$ → both can't be null simultaneously

② $|wxy| \leq n$

Date

Date

→ $L = \{ a^i b^i c^i ;$ where $i \geq 0 \}$

Assumption Must → Let L is CFL → CFG with P variables.

$a^n b^n c^n$; → instance

$|u| = n+n+n = 3n > n$.

"Take instance with length greater than n"

$n = 2^{P+1}$

$a^{30} b^{30} c^{30}$
↓
invalid instance.

→ Now, divide u in 5 portions

One division

(1) $v = a^n$, $wxy = b^n$, $z = c^n$
$v = a^n$, $w = b^{n/2}$, $x = \Lambda$, $y = b^{n/2}$, $z = c^n$

Other division.

(2) $v = \Lambda$, $wxy = a^n$, $z = b^n c^n$
$v = \Lambda$, $w = a^{n/2}$, $x = a^{n/2}$, $y = \Lambda$, $z = b^n c^n$.

(3) $v = a^n$, $wxy = b^{n/2}$, $z = b^{n/2} c^n$.
// $w = b^{n/4}$, $x = \Lambda$, $y = b^{n/4}$, $z = b^{n/2} c^n$

if for any division of instance, we can't conclude changing that instance

Date: Day:

→ If we give some power to w, y no. of b's will become different than no. of a, c.

so, valid instance.

4 $v = a^n b^n$, $wxy = c^n$, $z = \wedge$.

$w = c^k$

$x = \wedge$

$y = c^{n-k}$.

$vw^ixy^iz = (a^n b^n)(c^k)^i \wedge (c^{n-k})^i \wedge$.

$\boxed{\text{for } i = 2}$

$= a^n b^n c^{2k} c^{2n-2k}$

$= a^n b^n c^{2n} \notin L$

↓

language not CFL, Proved

Disproving on 3

$vw^ixy^iz = (a^n)(b^{n/4})^i \wedge (b^{n/4})^i (b^{n/2}) c^n$

$\boxed{i = 16}$

$= a^n b^{4n} b^{4n} b^{n/2} c^n$ — $8n + n/2 \geq n$ & so ② property is not followed here

$= a^n b^{8n + n/2} c^n \notin L$

↓

Not CFL.

Conclusion "Must"

Since this is not part of language, so our assumption that this was CFL contradicts.

$L = \{ SS$ where $S \in \{a, b\}^* \}$.

let L is CFL → CFG with P variables

① $n + n = 2n \geq n$

$|u| \geq n$

② $a^{n/2}\ a^{n/4}\ a^{n/4}\ a^n$

$vwxyz = a^{n/2}\ (a^{n/4})^i\ (a^{n/4})^i\ a^n$

$\boxed{u = a^n a^n}$ → instance

$\boxed{\text{Put } i = 4}$

$a^{n/2}\ a^n a^n\ a^n$

$a^{2n + n/2}\ a^n$

$a^{3n + n/2} = a^{\frac{6n+n}{2}} = a^{7n/2}$

since, we don't know n, n could be anything, and we may get even no after putting

Date: Day:

some value in n. In that case we can evenly divide "a". So, we can not conclude, This is not a proper instance. We are not able to disproof with this instance.

$$u = \overbrace{a^n b}^{s} \overbrace{a^n b}^{s}.$$

$$|u| = n+1+n+1 = 2n+2 \geq n.$$

Division:

1. $v = \Lambda$, $wxy = a^n$, $z = ba^n b$.
2. $v = a^n$, $wxy = b$, $z = a^n b$
3. $v = a^n b$, $wxy = a^n$, $z = b$.
4. $v = a^n b a^{n/2}$, $wxy = a^{n/2} b$, $z = \Lambda$

whenever we will put some value of "i" in $w^i x y^i$, we will get imbalance in no of a's or b's

Doing for 1

$|wxy| = n \leq n$

$|wy| = n > 0$

$uv^i x y^i z = (\wedge)(a^{n/2})^i \wedge (a^{n/2})^i (ba^n b)$

$\boxed{\text{Put } i=0}$

$= ba^n b \notin L$. can't put n=0 to ~~check~~ bcz $n=2^{p+1}$

we can't divide it in such two parts such that both parts are equal, so instance was valid.

→ Give conclusion

$L_1 = CFL$ → S_1 , $L_2 = CFL$ → S_2

1) $L_1 \cup L_2$ → $S \to S_1 / S_2$

2) $L_1 L_2$ $S \to S_1 S_2$

3) L_1^* $S \to S_1 S / \wedge$ $L_1 = \{a^n b^n c^k\}$ $n, k \geq 0$

4) $L_1 \cap L_2$ → Non CFL $L_2 \{a^k b^n c^n\}$

5) $\overline{L_1}$ // // $L_1 \cap L_2$ $a^n b^n c^n$ → not CFL

6) $L_1 - L_2$ // //

Not closed under CFL

Date: Day:

$L_1 = RL$, $L_2 = CFL$.

1) $L_1 \cup L_2$ CFL.

2) L_1^* R.L.

3) $L_1 L_2$ CFL.

4) $L_1 \cap L_2$ CFL. $a^n b^n \cap a^n b^n \rightarrow a^n b^n$.

5) $\overline{L_1}$ RL. bcz L1 was regular.

6) $L_2 - L_1$ CFL.

TURING MACHINES

Languages

- R.L
  - RE
  - FA
    - DFA
    - NFA
    - NFA-Λ
- NON-RL.
  - Turing Machine
  - CFL
    - CFG
    - PDA
      - DPDA
      - NDPDA

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

Σ: input set

Γ: allowed symbol on tape

δ: transition

$\Sigma \subseteq \Gamma$

B: blank symbol (Δ)

F: set of final state

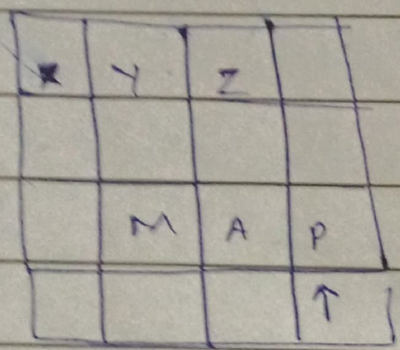
$\delta = Q \times \Gamma \rightarrow Q \times \Gamma \times (L, R)$

(L, R): direction of transition

Non deterministic behavior → $(q_0, a) \to (q_1, a, R)$
$(q_0, a) \to (q_1, b, L)$ — can go in two direction on one input

Non deterministic — can go in two direction on one input.

Allan Turing → said machine work as human, should be limitless like human brain, and have limitless memory, not just stack.



2D

⟹ |Δ|Δ|Δ|Δ|Δ|

delta Δ means blank.

Instead of online input, we assume input is on tape.

- → input shall be b/w two delta.

  $x = abb$ $\quad \Delta x \Delta$ |Δ|a|b|b|Δ|Δ|Δ| - - -

- → head pointer can move left, right or stay stationary

- → if head ptr is on first cell then it cannot move left, we place '#' on first block to know it is first cell.

|#|Δ|Δ|Δ| - - -

→ there can be many ways of forming tape

- |#|Δ|Δ| - -
- - - |Δ|Δ|#|
- - - |Δ|#|Δ| - -

Date: Day:

L = {x | x has substring aba} Σ{a, b}

we cannot make partial decision, we have to make final decision, so we shall be in halt state before making decision

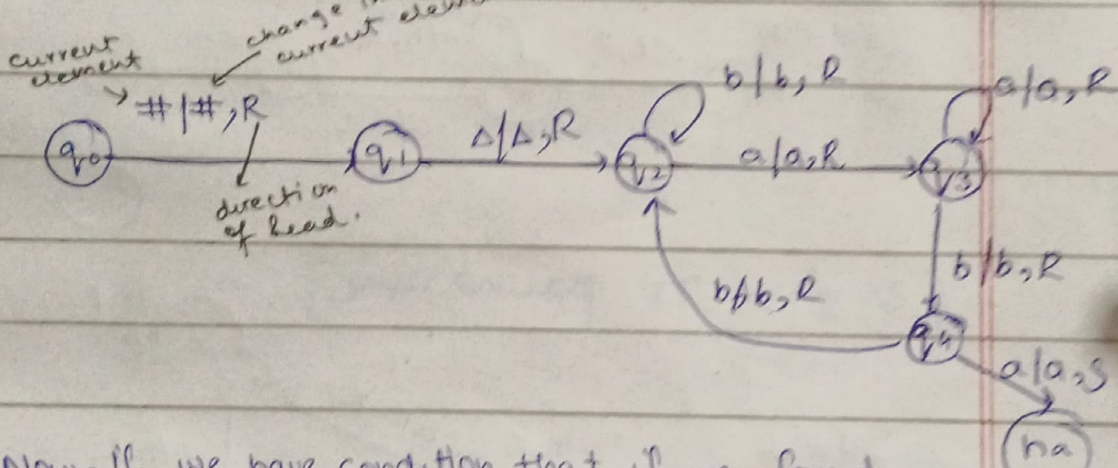
halt state

ha — explicit

hr — implicit

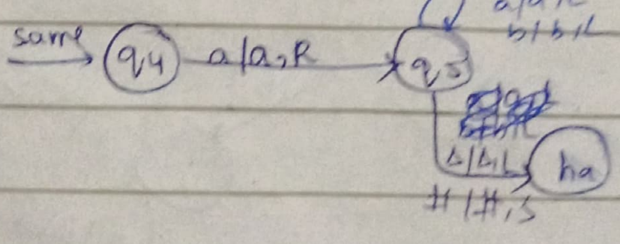
| # | Δ | b | a | a | a | b | a | b | b | Δ ... this string is input

P.T.O

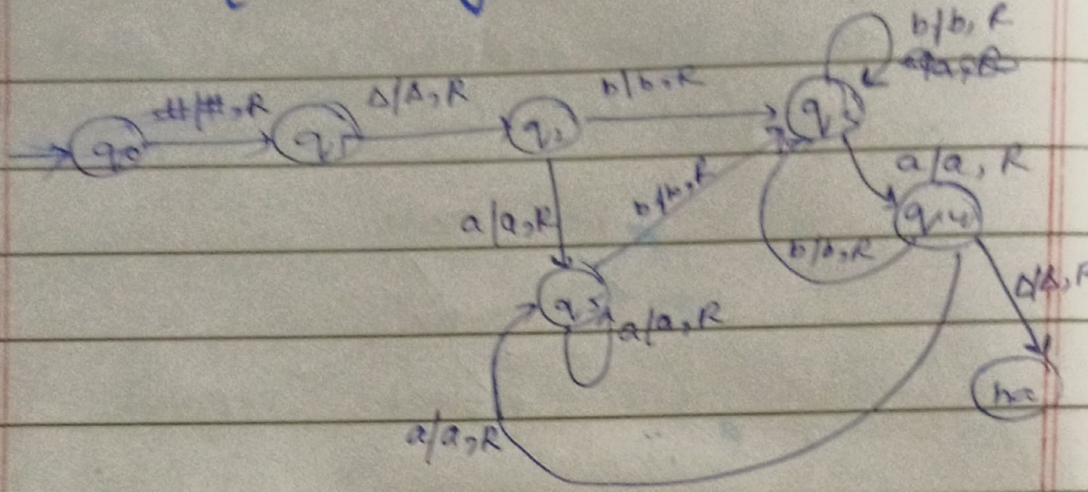


- Now, if we have condition that if we found aba, first move back to # then halt

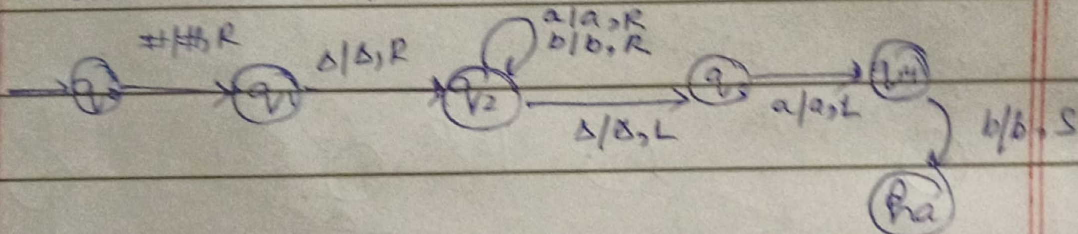
→ we add a loop to move back to #



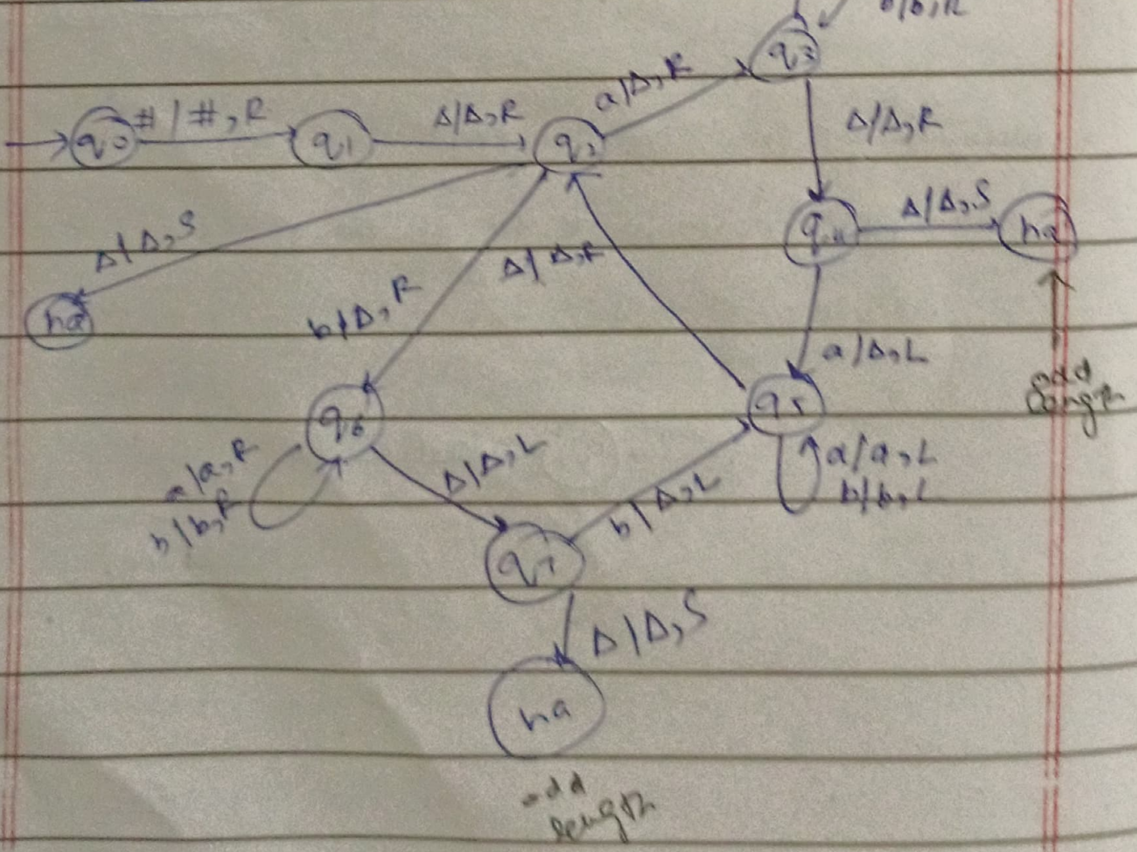
→ L = { x string ends with ba }



better sol, can be that we move to the end, and check if last two cells are ba.



L{ x | x is palindrome }.

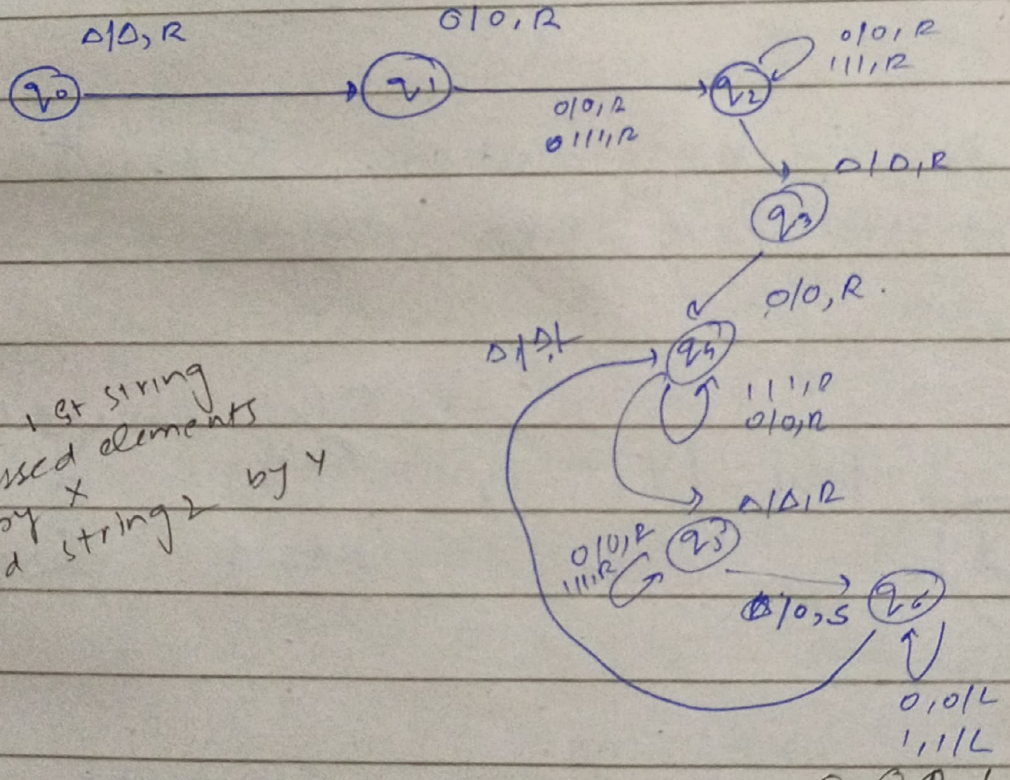


Date: Day:

Develop TM that is given 2 inputs $x, y \in \Sigma^{*}$, you have to compute bit wise AND.

$|X| = |Y|$, $\Sigma = \{0, 1\}$.

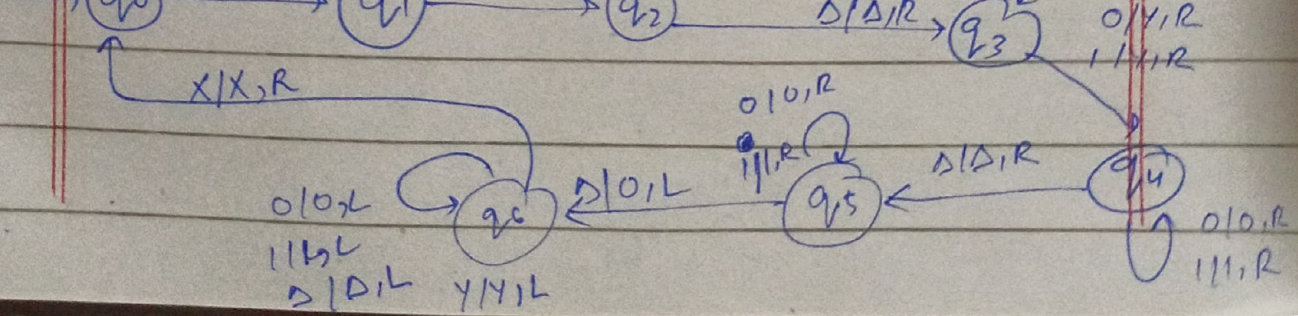
| Δ | x | Δ | y | Δ | z |



Replace 1st string processed elements by X and string 2 by Y

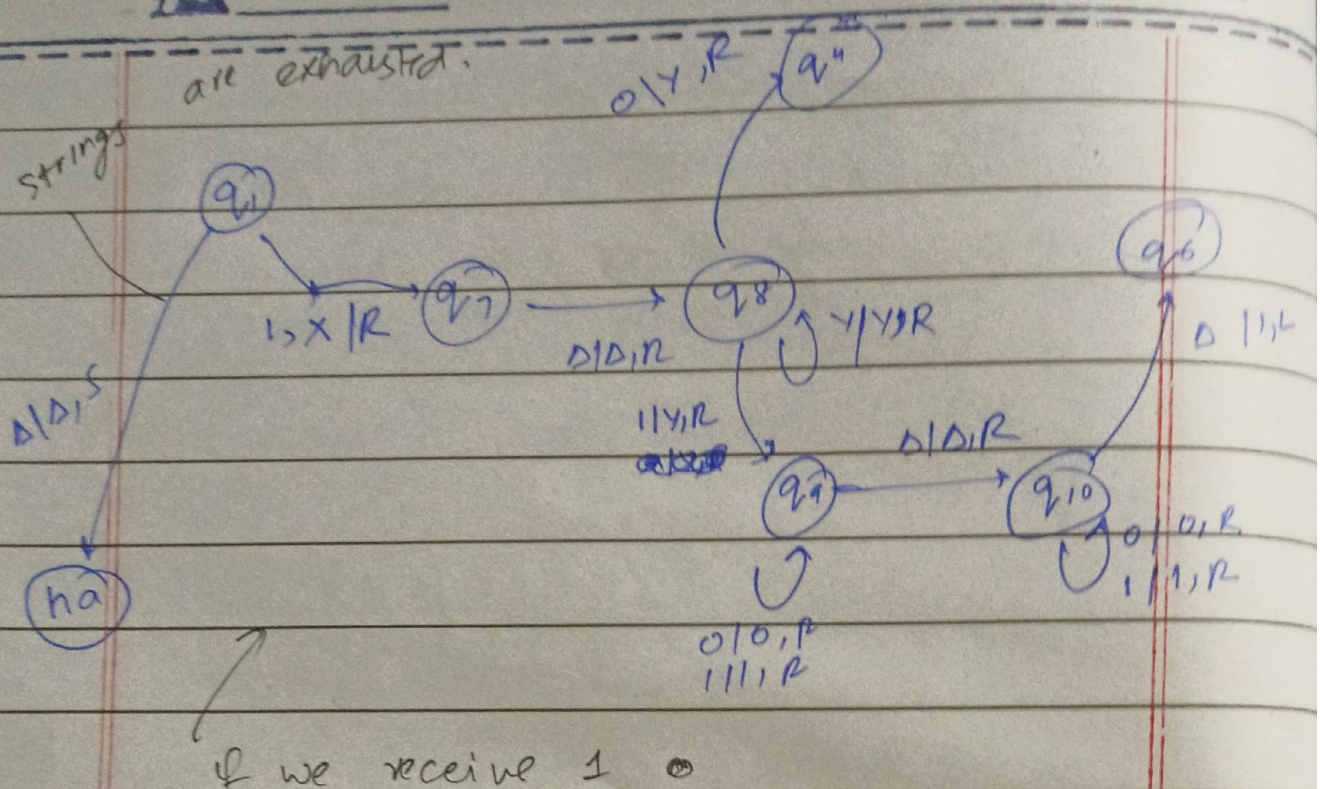
If we receive 0

| Δ | 0 | 1 | 1 | 0 | 1 | Δ | 1 | 0 | 1 | 1 | Δ | Δ | Δ | Δ |



Date ________ Date ________

are exhausted.



⇒ If we have multiple tapes, it would have been easy

HW

$|X| = |Y|$, $x > y$ output 0

$x \leq y$ 1

MUITI TAPE TURING Machine

First copy "y" from 1st tape to 2nd.

	Δ	x	Δ	y
	Δ	Δ	Δ	Δ
z →	Δ	Δ	Δ	Δ