



# Computer Science

## Theory of Computation

Turing Machine

Lecture No.- 1



Mallesham Devasane Sir

# Recap of Previous Lecture



**Topic**

**Context Free Languages**





# Topics to be Covered



**Topic**

**TM**

**Topic**

**LBA Vs HTM Vs TM**

**Topic**

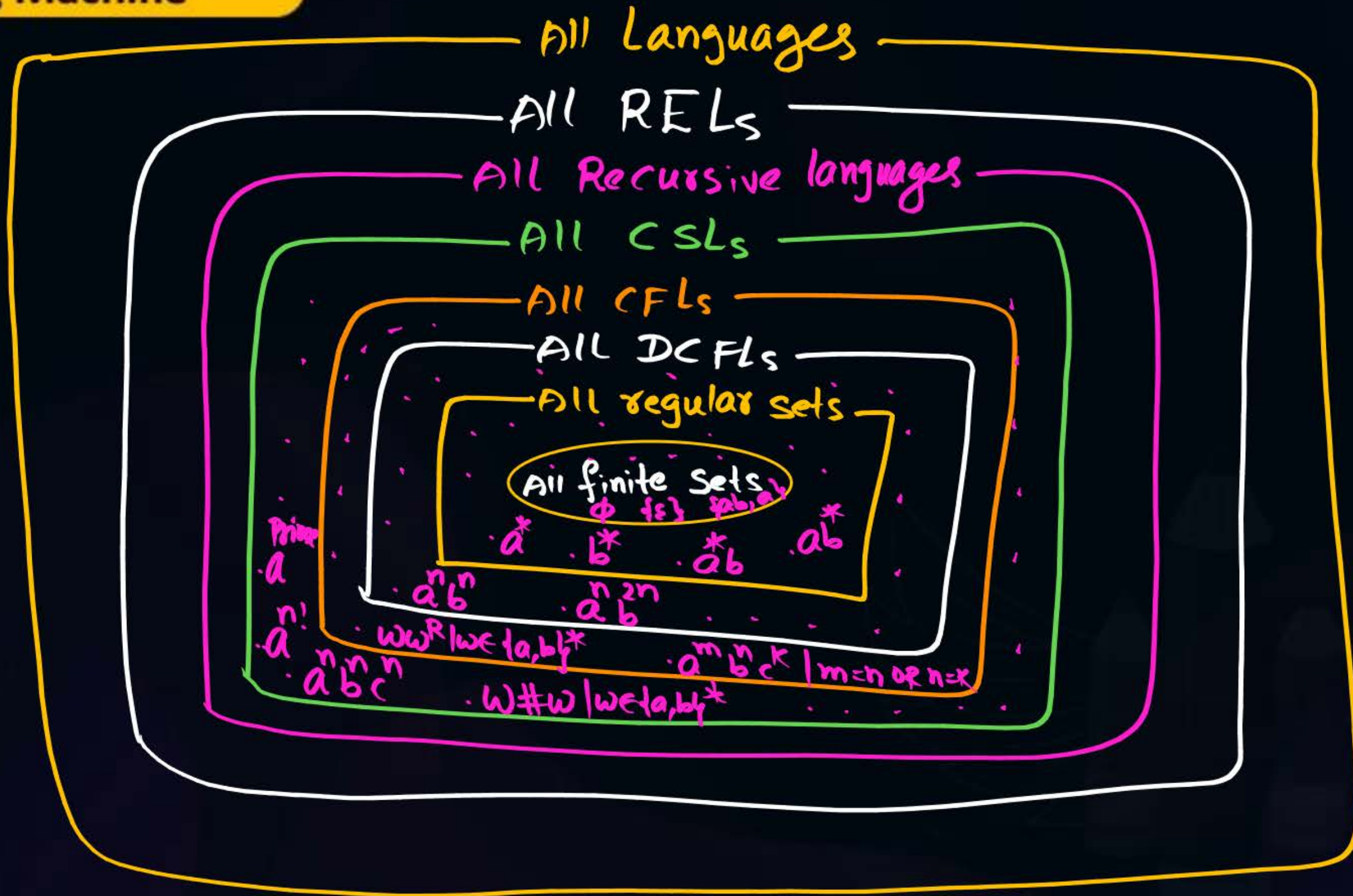
**TM Construction**

**Topic**

**Recursive Vs REL**

**Topic**

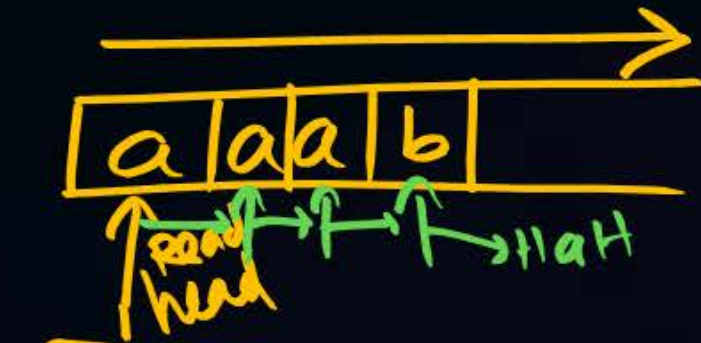
**Closure properties**





FA

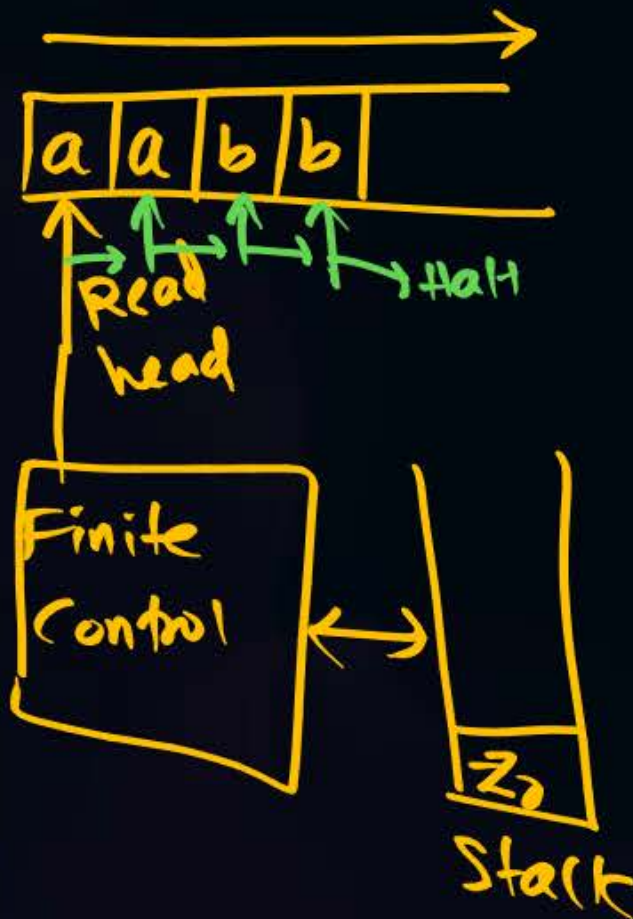
$(Q, \Sigma, \delta, q_0, F)$



Finite control

PDA

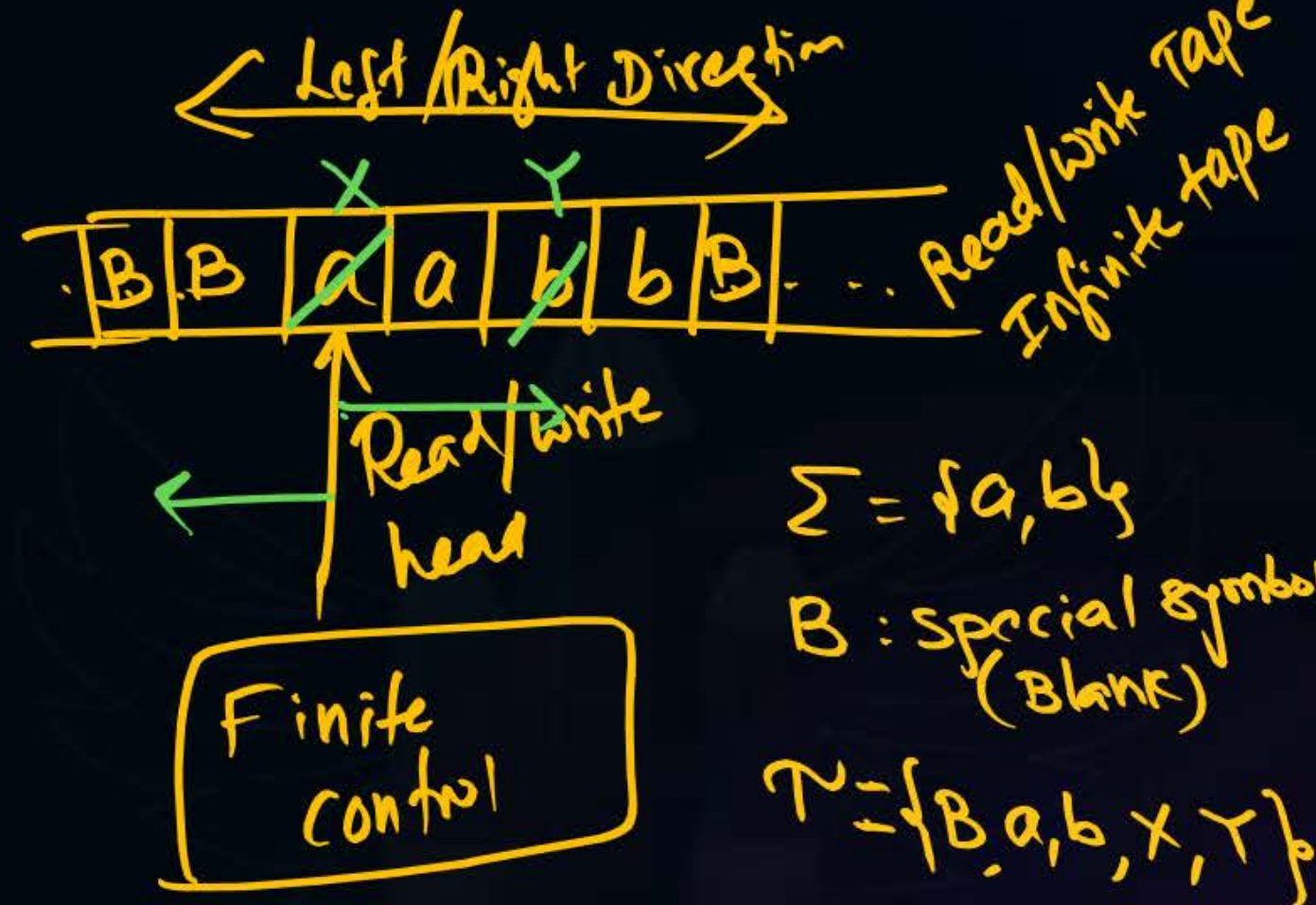
$(Q, \Sigma, \delta, q_0, F, Z_0, \Gamma)$



Stack alphabet

TM

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



Tape Alphabet

$\Sigma = \{a, b\}$   
 $B$  : special symbol (Blank)  
 $\Upsilon = \{B, a, b, x, y\}$



FA

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

PDA

$(Q, \Sigma, \delta, q_0, F, z_0, \Gamma)$

TM

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

$DFA \cong NFA$

$DFA: Q \times \Sigma \rightarrow Q$

$NFA: Q \times \Sigma_{\epsilon} \rightarrow 2^Q$

$PDA: Q \times \Sigma_{\epsilon} \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$

$DTM \cong NTM$

$DTM: Q \times \overset{\text{read}}{\tau} \rightarrow Q \times \overset{\text{write}}{\tau} \times \{L, R\}$

$NTM: Q \times \tau \rightarrow 2^{Q \times \tau \times \{L, R\}}$

FA



$$\delta(1, a) = 2$$

$$Q \times \Sigma \rightarrow Q$$

$$1 \in Q \quad a \in \Sigma \quad 2 \in Q$$

PDA



$$\delta(1, a, x) = (2, xx)$$

$$\text{ppda: } Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$$

$$\begin{matrix} 1 \in Q & & 2 \in Q \\ a \in \Sigma & & xx \in \Gamma^* \\ x \in \Gamma & & \end{matrix}$$

TM



$$\delta(1, a) = (2, y, R)$$

state read

state

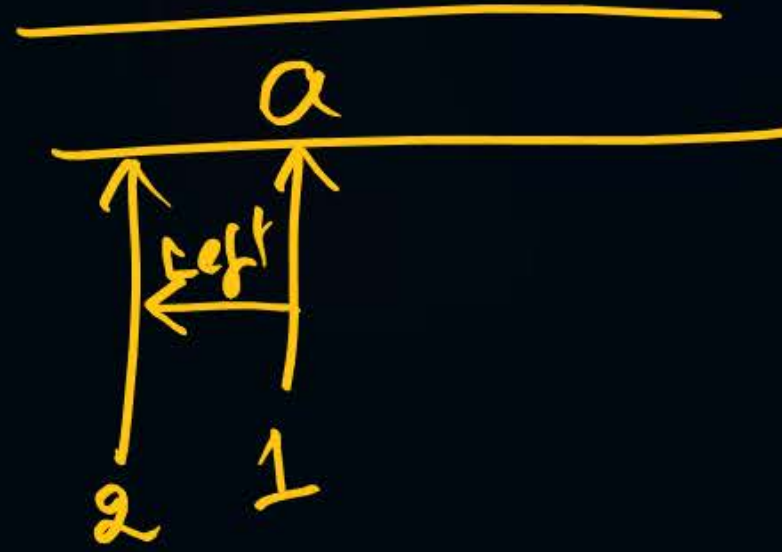
write

Direction



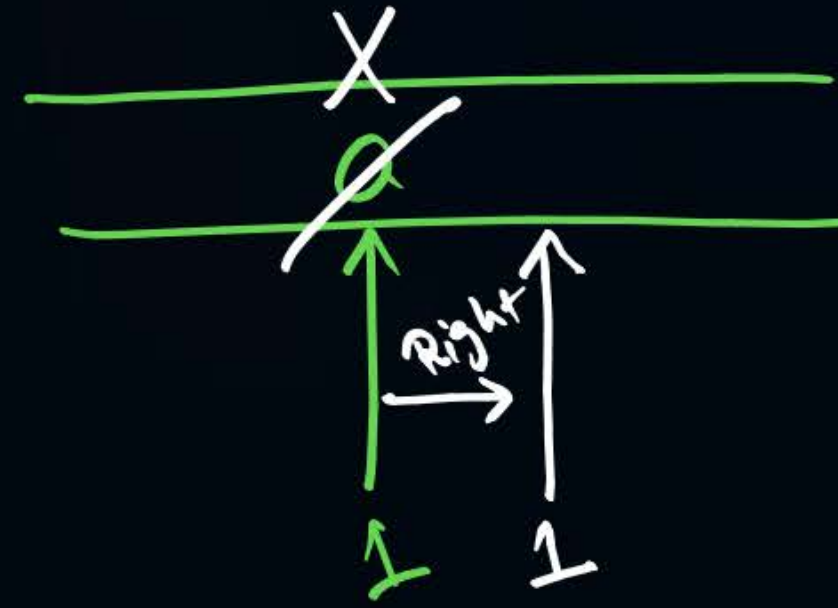
$$\delta(1, a) = (2, a, L)$$

from state 1,  
by reading a,  
goes to state 2  
by writing with a  
and moves to left direction.





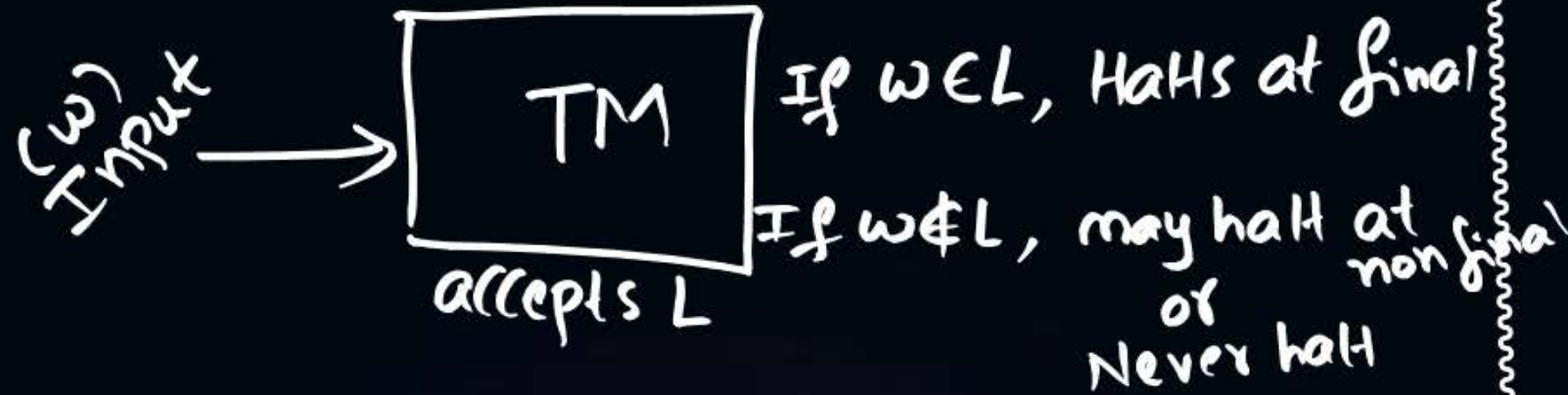
$$\delta(1, a) = (1, x, R)$$



- It is powerful machine
- It is equivalent to computer/program
- It represents "Recursively Enumerable Language" (REL)  
(Semi-decidable language)  
(Enumerable language)  
(Recognizable language)



## Acceptor

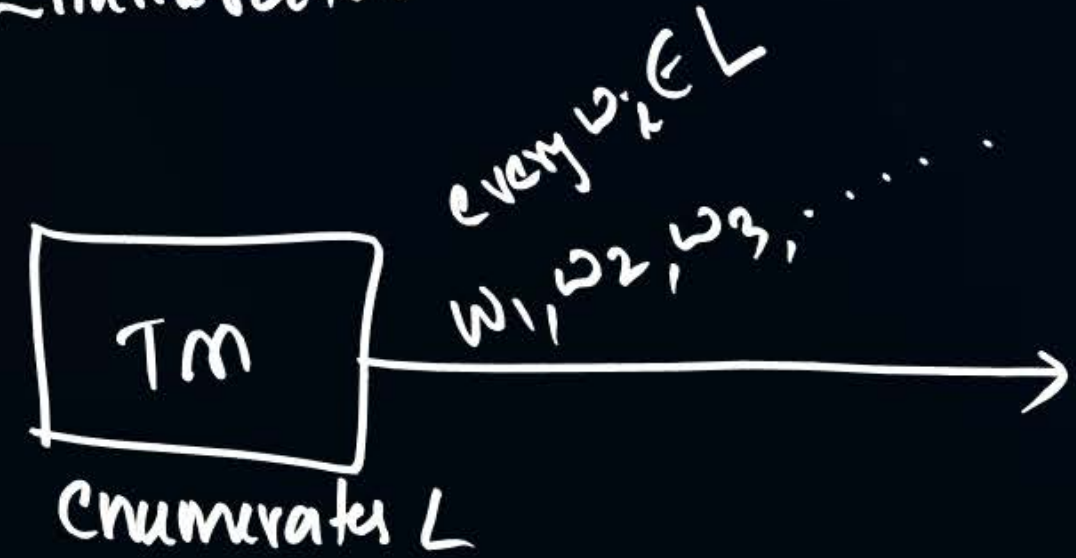


TM accepts RE L.

$L(TM)$  is RE L

$TM \equiv RE L$

## Enumerator



TM enumerates RE L

$L(TM)$  is RE L

\*\*\*

If <sup>Input</sup> String is valid,  
w ∈ L

Always halts at final.  
Logic exist

If <sup>Input</sup> String is invalid,  
w ∉ L

either halts at nonfinal  
or  
never halts

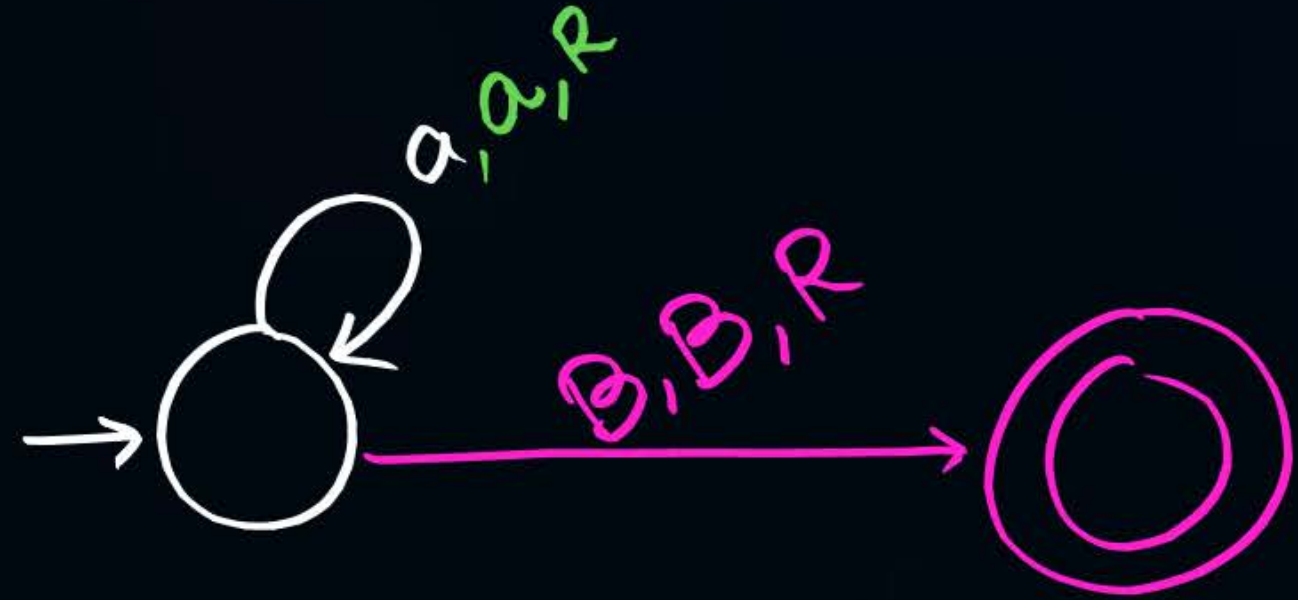
Logic may or may not exist



$$\textcircled{1} \quad L = a^*$$



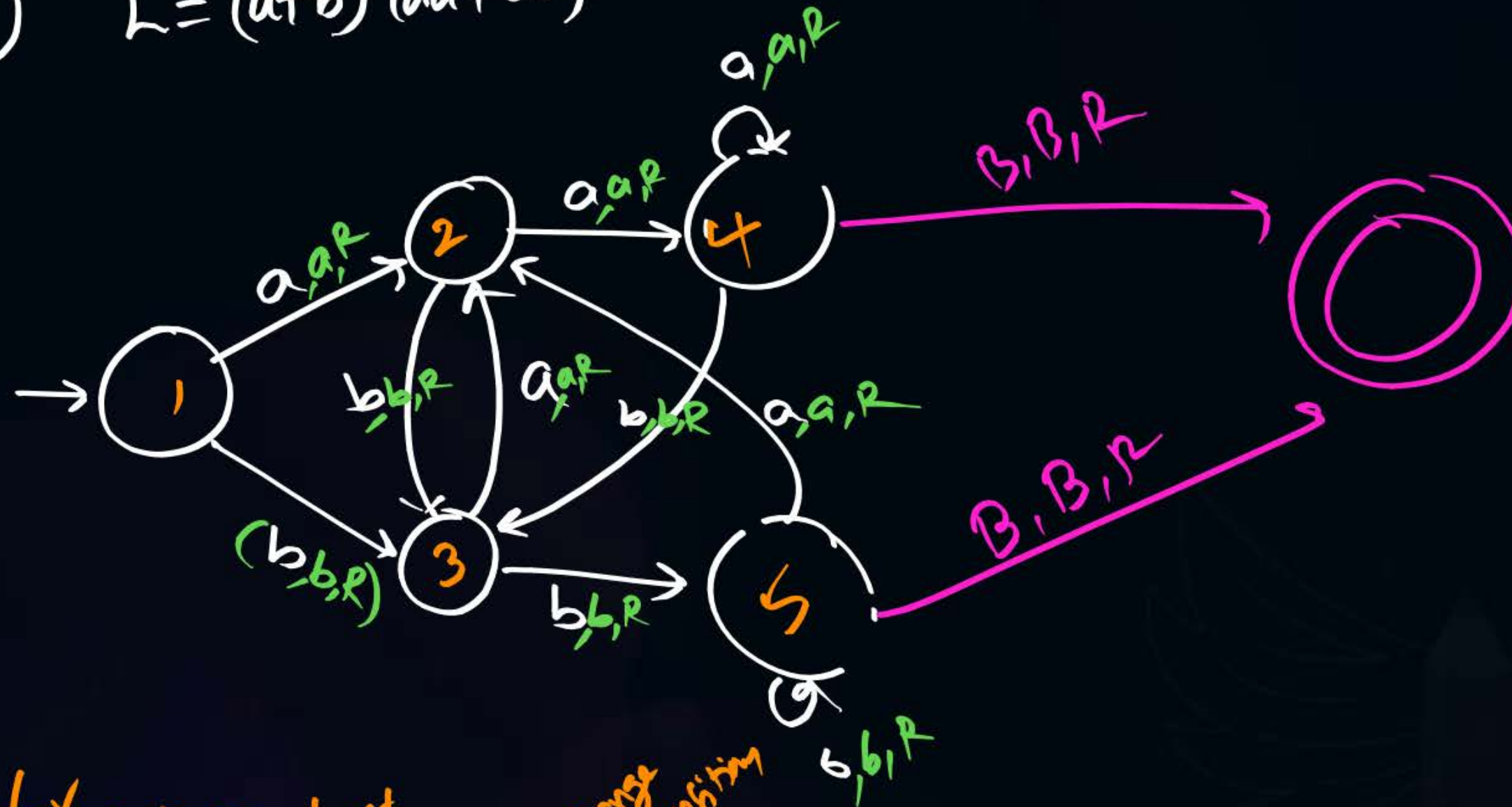
FA



TM

For every regular, we can design TM

②  $L = (a+b)^*(aa+bb)$



is Regular  $\Rightarrow$  construct DFA or NFA

change transition  $\Rightarrow$  TM



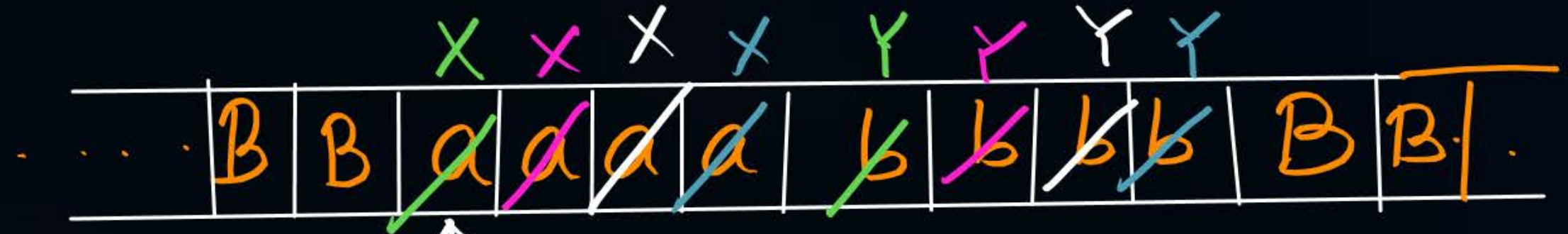
Note: We can convert any FA to TM



If  $q$  is final in FA



$$(3) \quad L = \{a^n b^n \mid n \geq 1\}$$



In each scan,

we replace  $a$  with  $X$   
 $b$  with  $Y$

we can remember 1a and 1b

remember 1a and 1b



1<sup>st</sup> scan:

... B B ~~a~~ a a a ~~b~~ b b b B B |

The diagram illustrates the state transitions for finding the first 'x' in a string. It shows states  $q_0$ ,  $q_1$ , and  $q_2$ . Transitions include 'R' (right), 'L' (left), and 'skip all a's' (green arrows).

$$(3) L = \{a^n b^n \mid n \geq 1\}$$

3<sup>rd</sup> scan



$q_0$ : Replace a with X

$q_1$ : skip a's } Replace b with Y  
Y's

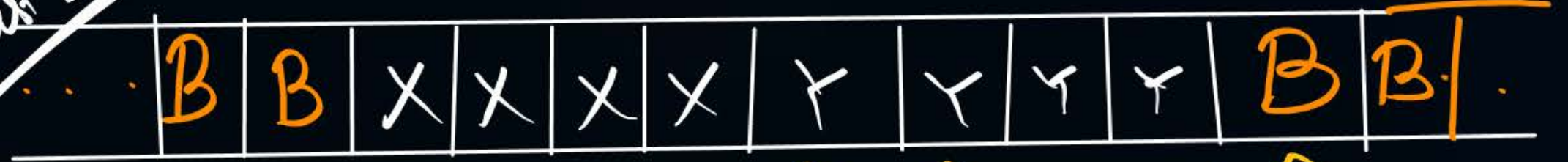
$q_2$ : skip a's } Find 1<sup>st</sup> X in reverse  
Y's





$$(3) L = \{a^n b^n \mid n \geq 1\}$$

Last scan:

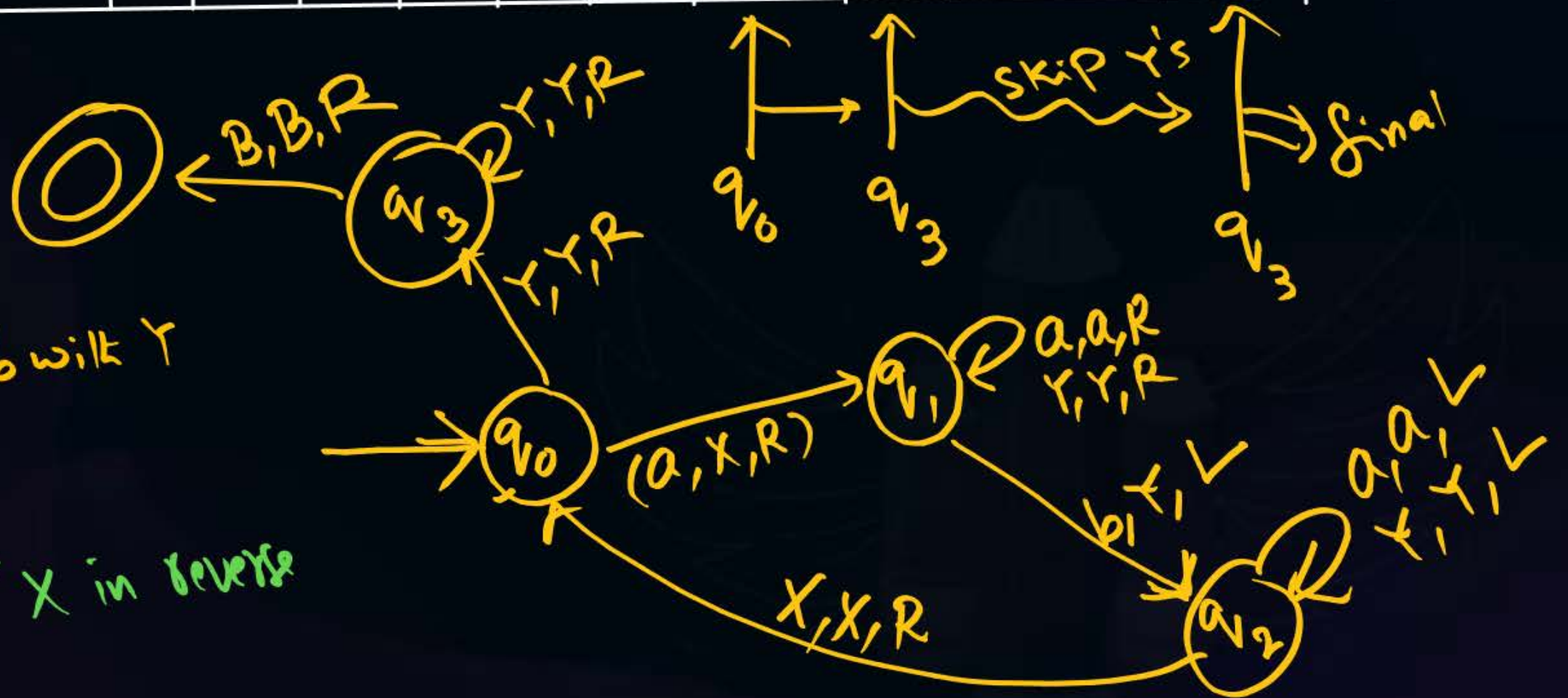


$q_3$ : skip all Y's to reach B.

$q_0$ : Replace a with X

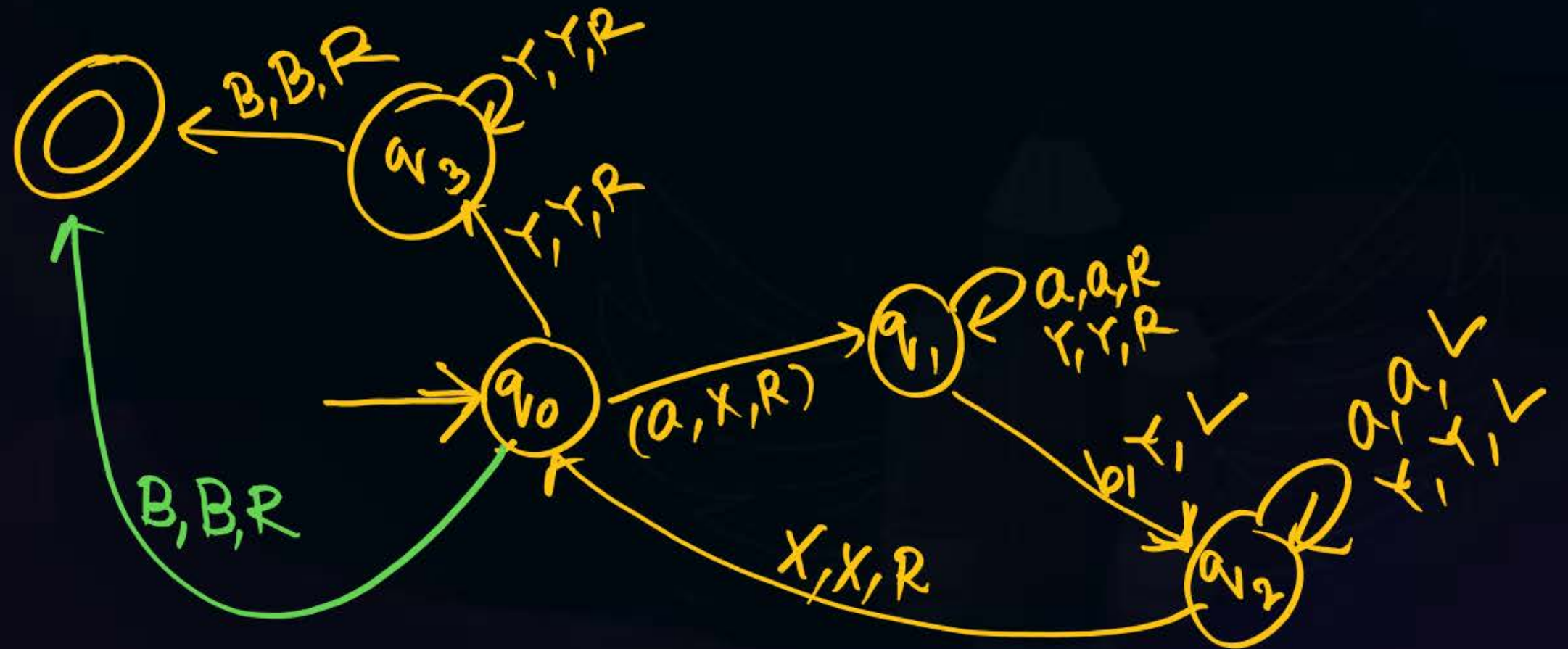
$q_1$ : skip a's } Replace b with Y  
Y's

$q_2$ : skip a's } Find 1<sup>st</sup> X in reverse  
Y's



④  $L = \{a^n b^n \mid n \geq 0\}$

B B B  
↑



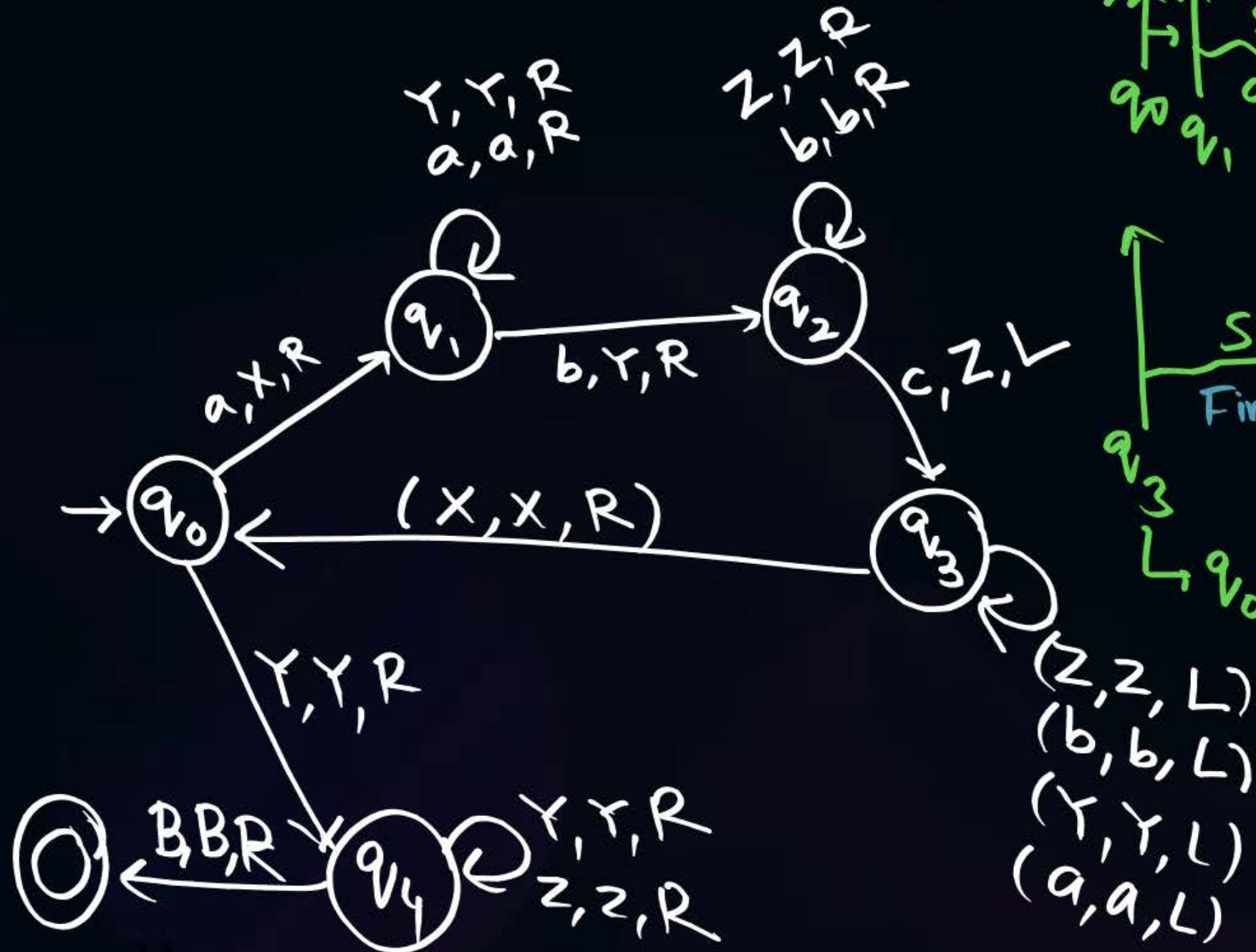


⑤  $\{a^n b^n c^n \mid n \geq 1\}$

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

skip  
as, Y's  
bs, Z's

skip Z's, b's, Y's, a's  
Find 1<sup>st</sup> x in reverse



$q_0$ : ~~a~~ X

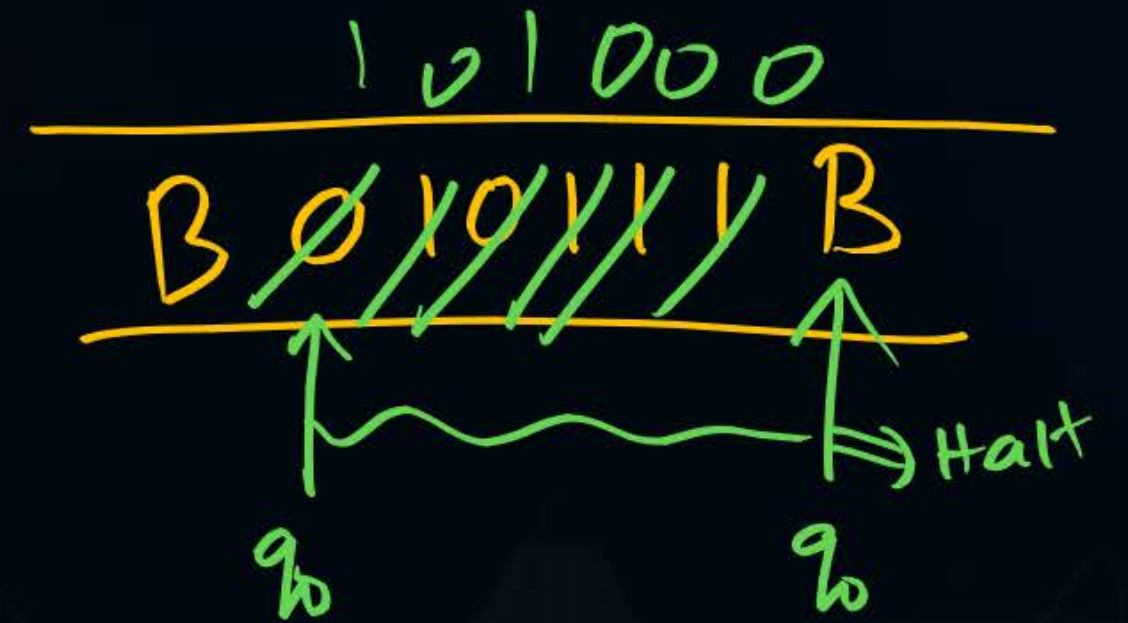
$q_1$ : ~~b~~ Y

$q_2$ : ~~c~~ Z

$q_3$ : To find 1<sup>st</sup> X in reverse

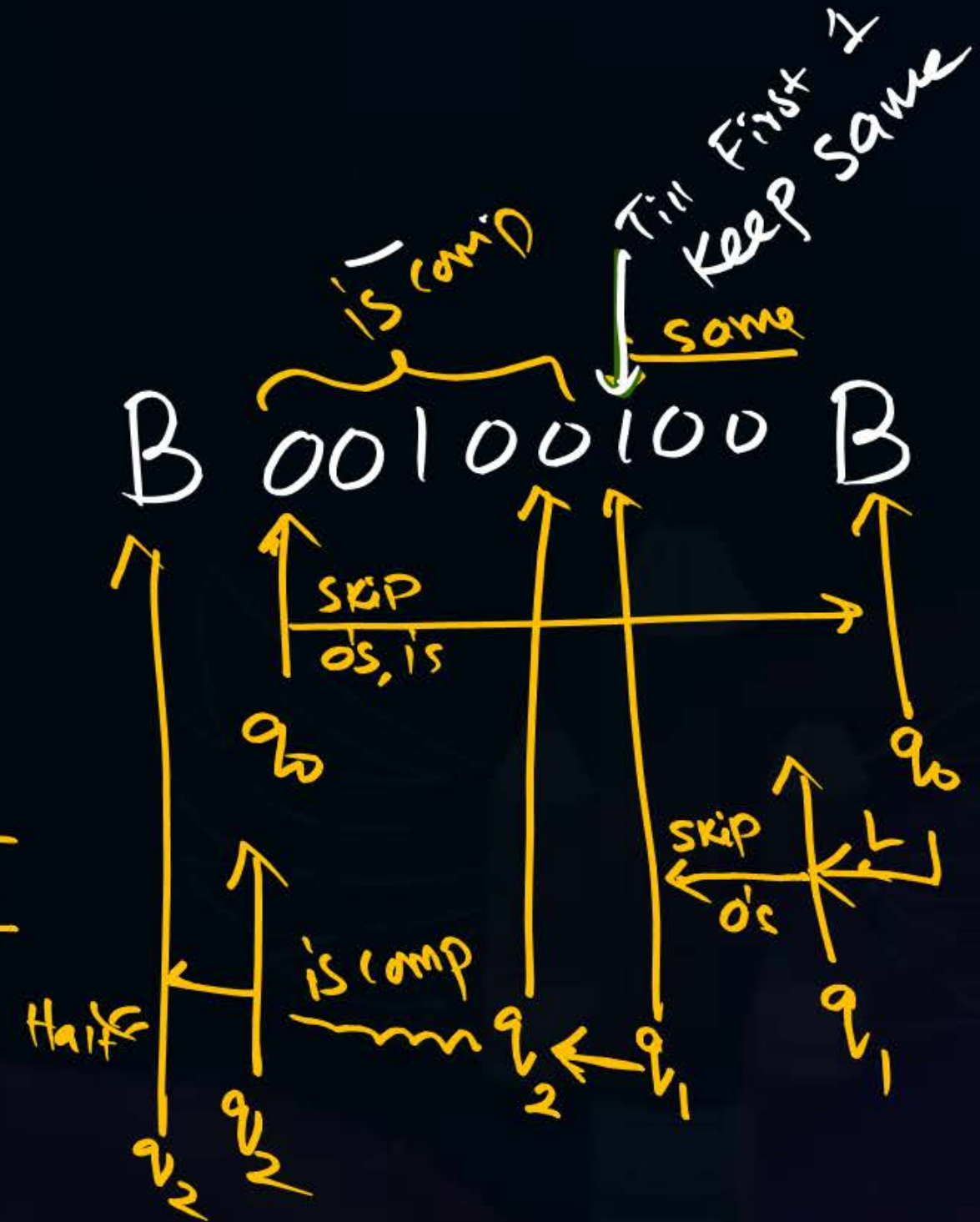
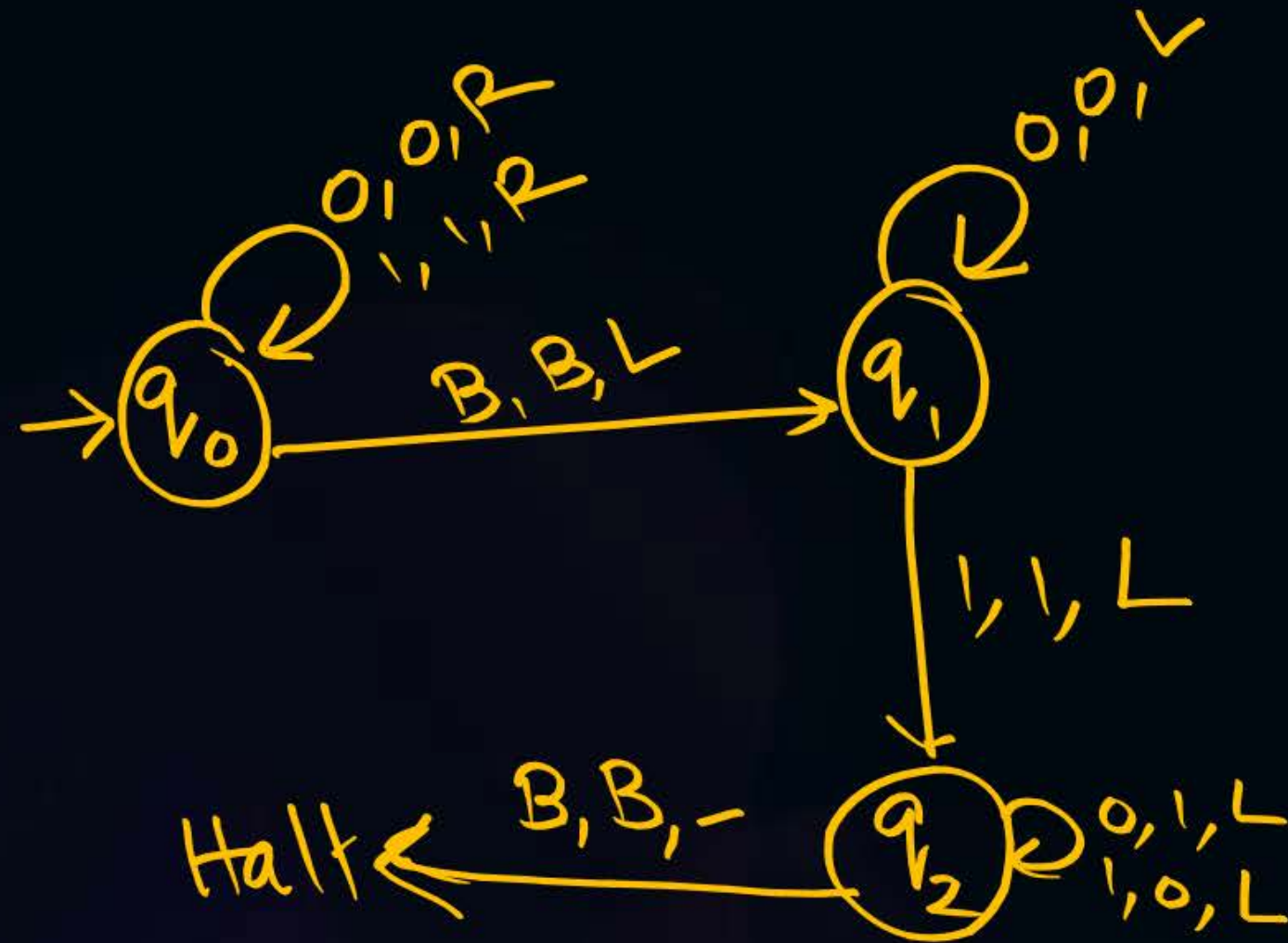
$q_4$ : To find B

⑥ 1's complement of binary



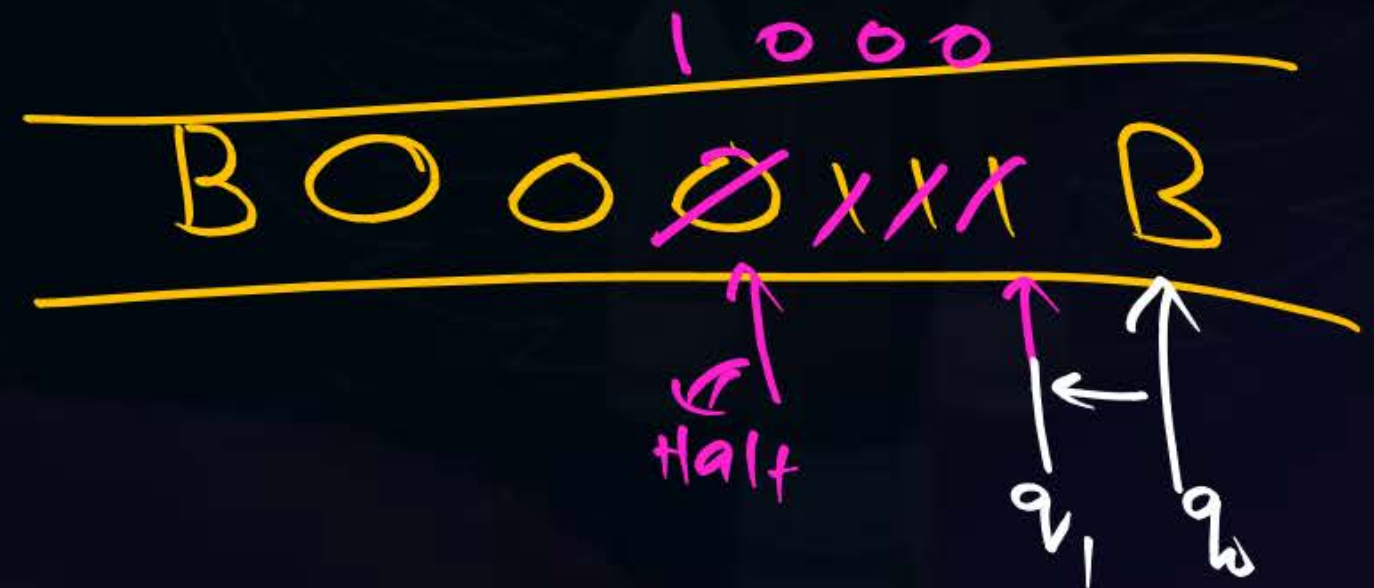
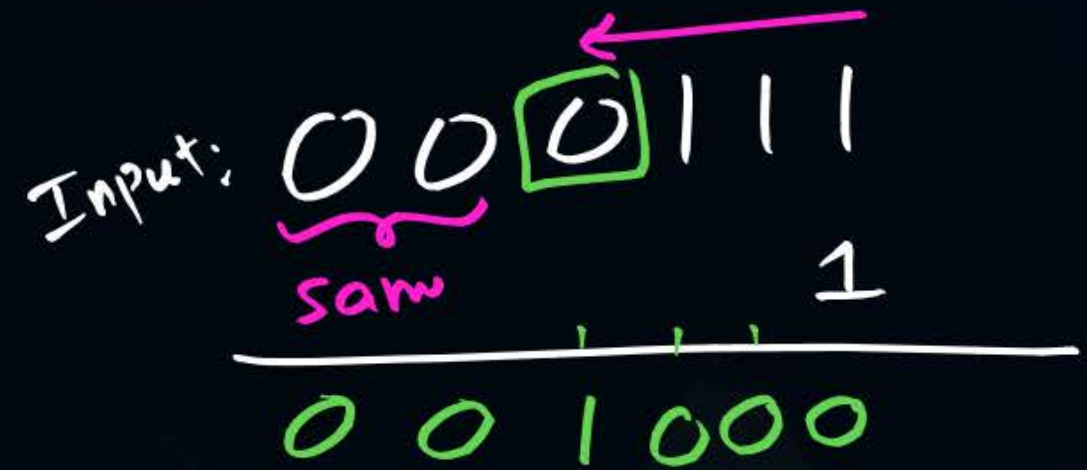
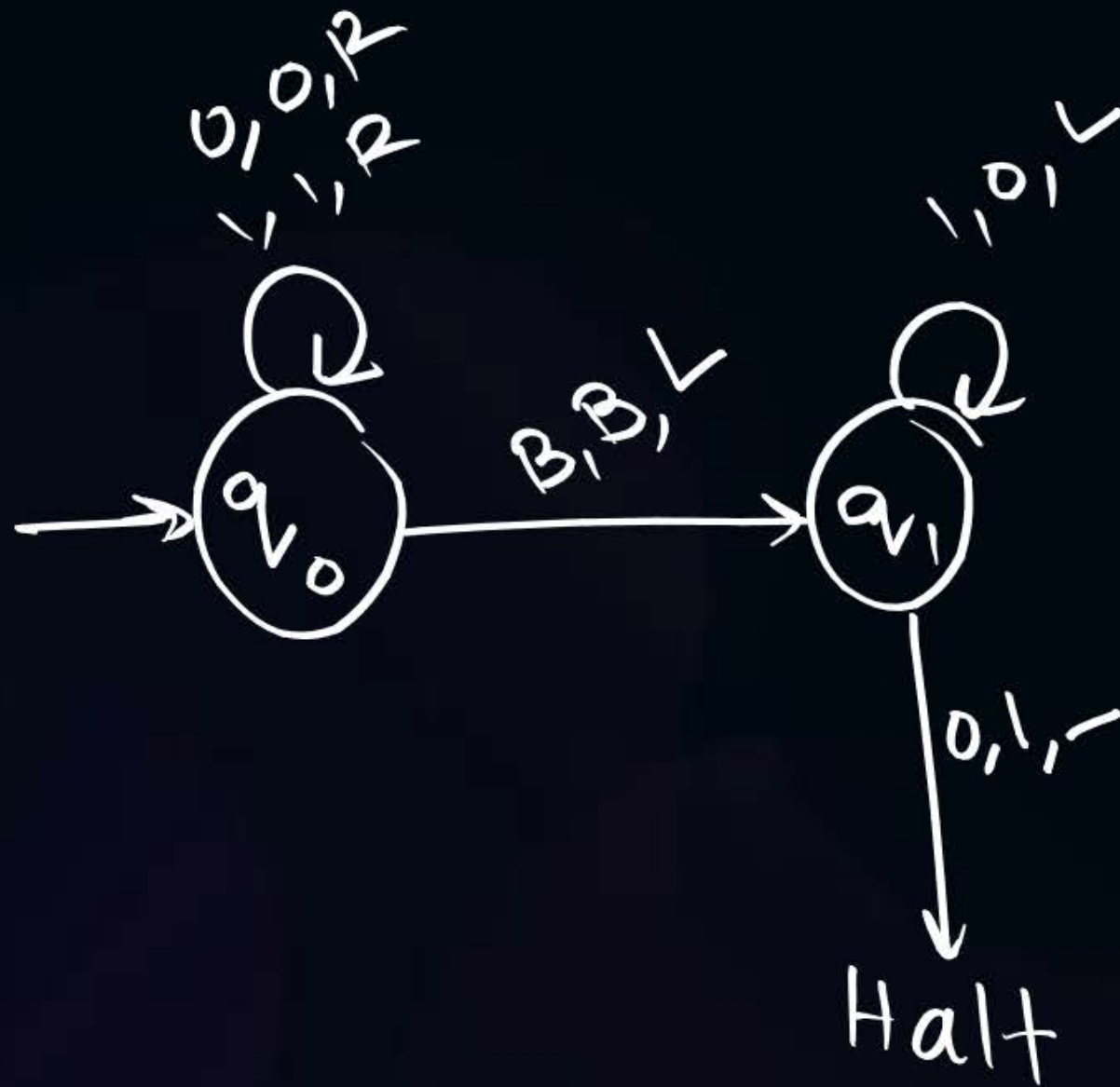


⑦ 2's complement of binary



## ⑧ Increment of Binary

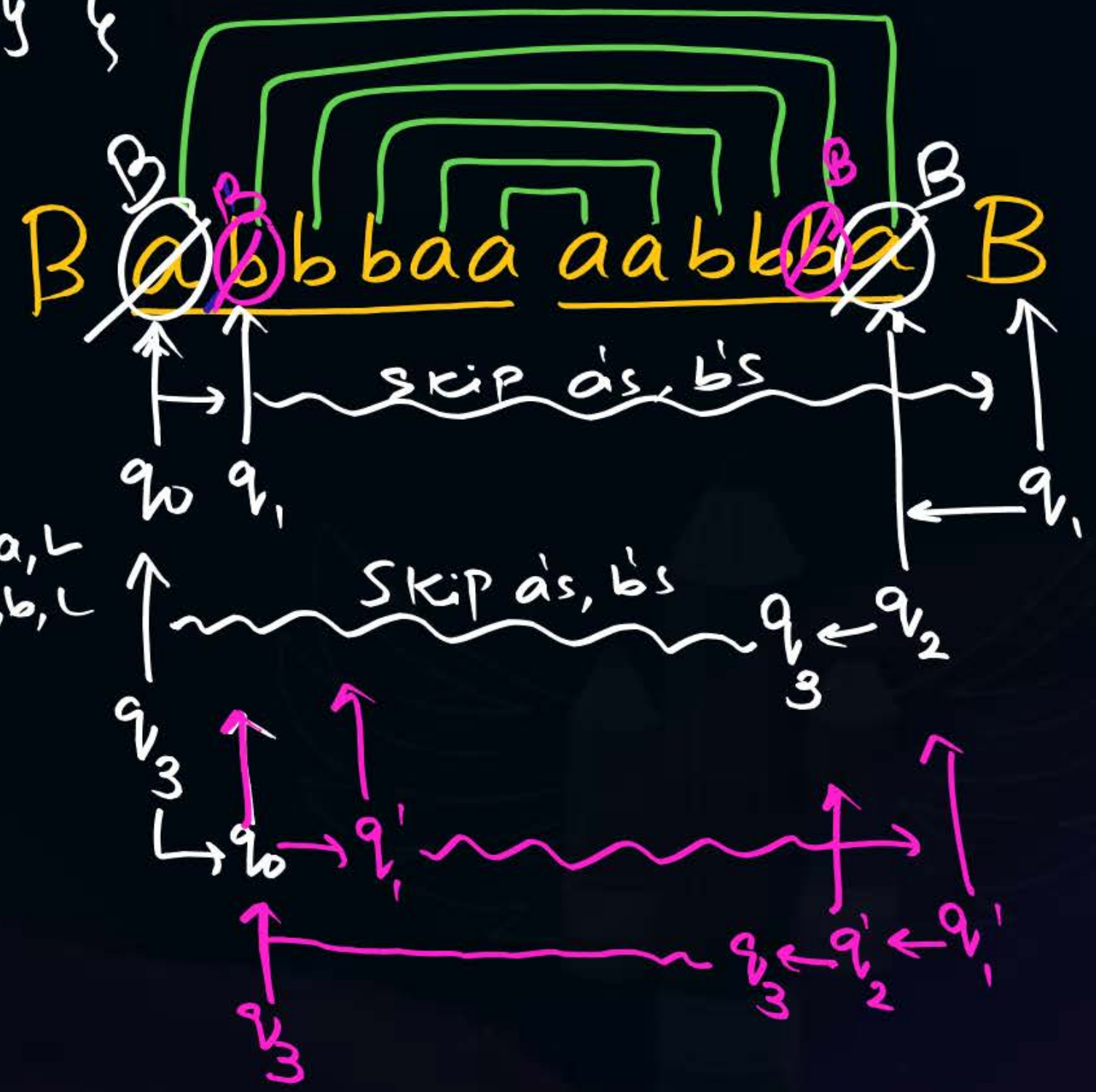
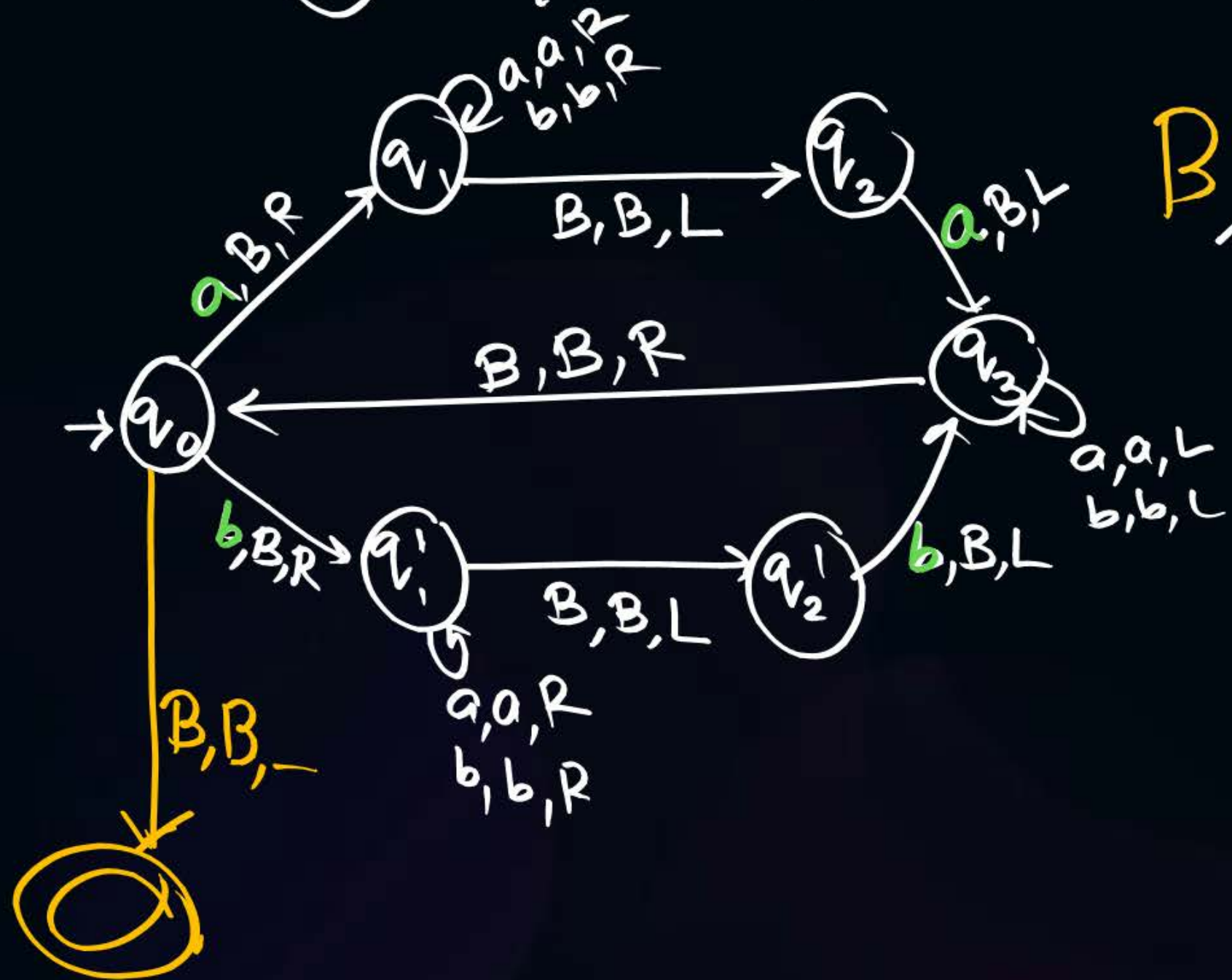
$$f(x) = x + 1$$





Handwritten diagram illustrating the iterative reduction of a string  $abbbbaa aabbbba$  to a single character  $a$  using the rules of the Collatz conjecture. The string is shown in yellow on a black background. The first  $a$  and the last  $a$  are circled in white. A long white arrow points from the first  $a$  to the last  $a$ . Above the string, several green curved arrows represent the iterative reduction process, starting from the first  $a$  and ending at the last  $a$ . Below the string, the same process is shown in white, with the string  $abbbbaa aabbbba$  being reduced to  $bbbaaabbba$ , then to  $baaaaabbb$ , and finally to  $baaaaabbb$ .

⑨  $\{ww^R \mid w \in \{a,b\}^*\}$





- H.W. {
- ⑩  $\{w\#w^R \mid w \in \{a,b\}^*\}$
  - ⑪  $\{w\#w \mid w \in \{a,b\}^*\}$



## 2 mins Summary



Topic

→ TM construction

Next: closure properties  
Recursive vs R.E.L



**THANK - YOU**