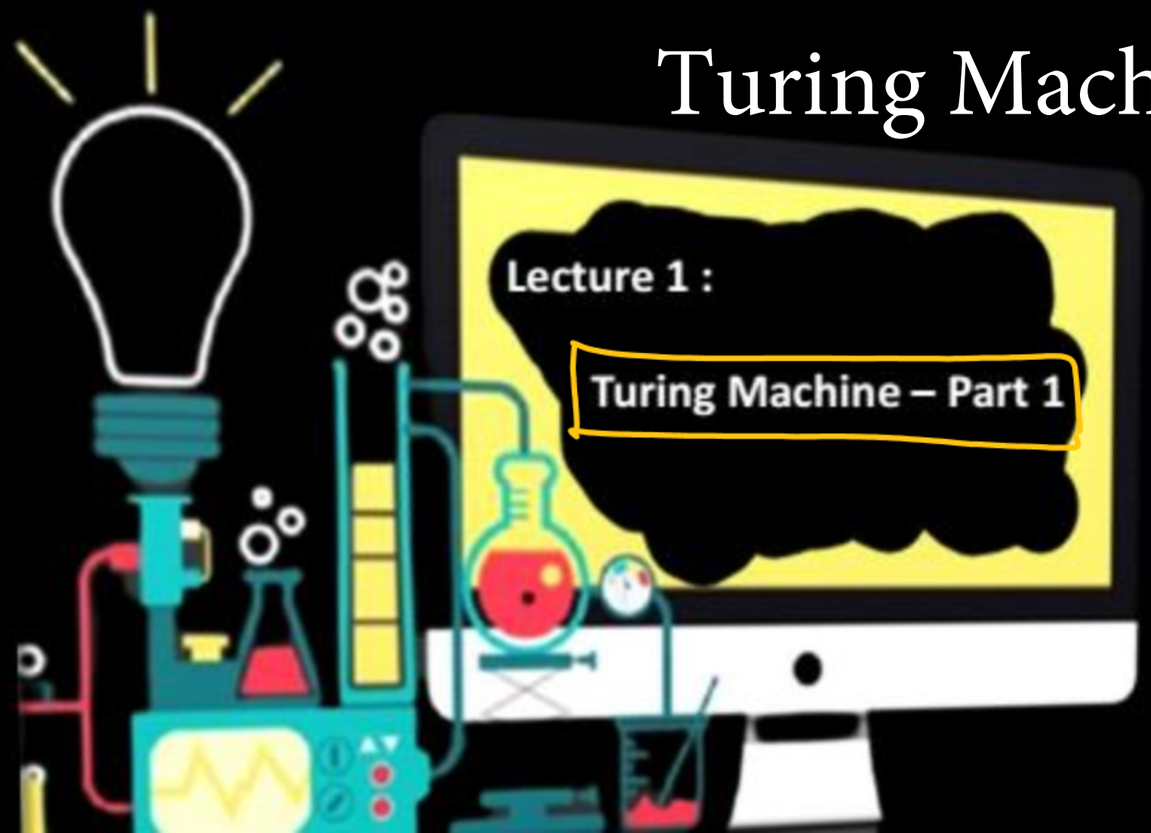
CS & IT Engineering

Turing Machine

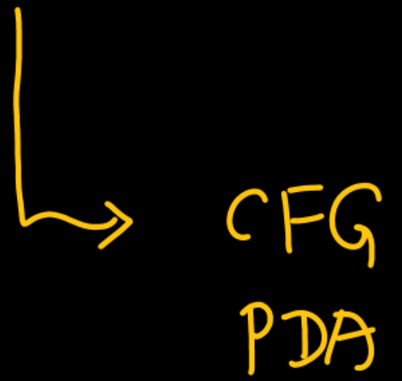


Deva sir

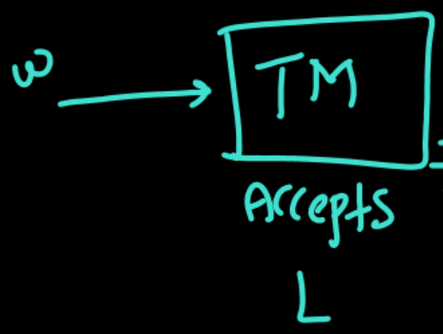
Topics to be covered:

- TM (Turing M/C)
  - What is TM?
  - Configuration of TM
  - Recursive and RE
  - TM Vs HTM Vs LBA
  - types of TMs

Topics Covered in Previous Session:

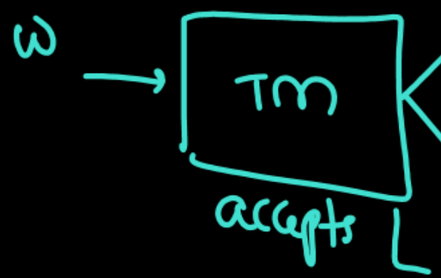


TM



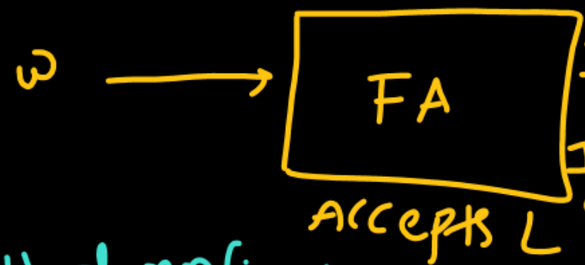
If $w \in L$, Halts at final

If $w \notin L$, either it may halt at nonfinal
or
never halts



Valid string always Halts

Invalid string (Don't know)



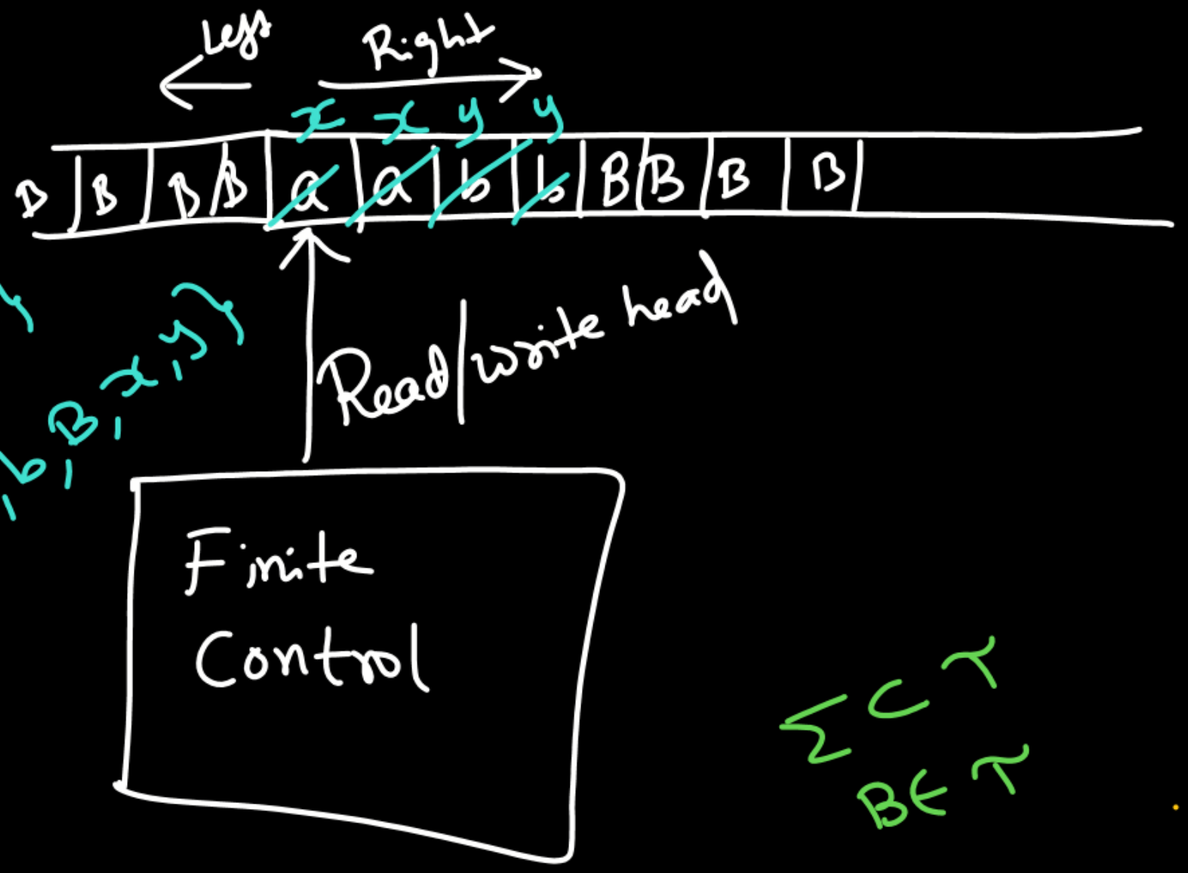
If $w \in L$, Halts at final

If $w \notin L$, Halts at nonfinal

If $w \in L$, Halts at final

If $w \notin L$, Halts at nonfinal

TM



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

$\Gamma = \{a, b, B, X, Y\}$

$\Sigma \subset \Gamma$

$B \in \Gamma$

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

Γ → Tape Alphabet

B → Blank symbol

TM ≅ FA + R/W tape + Bidirectional Head

Tm ≅ PDA + 1 stack ≅ 2 stack PDA

Tm ≅ FA + 2 Stacks

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

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

TM

DTM ≅ NTM ≅ TM

I) DTM :

present state, read, next state, write, Direction

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

II) NTM :

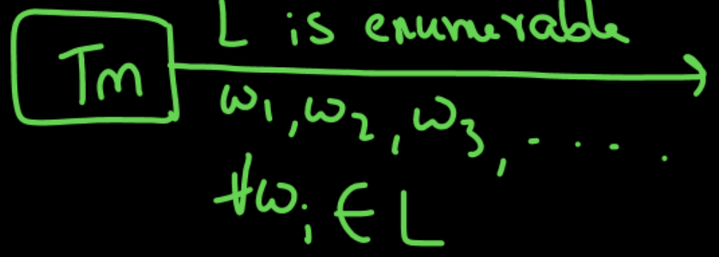
$$\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$$

TM

Recursively Enumerable Language

① It is Acceptable by TM
(Recognizable)

② It is enumerable by TM



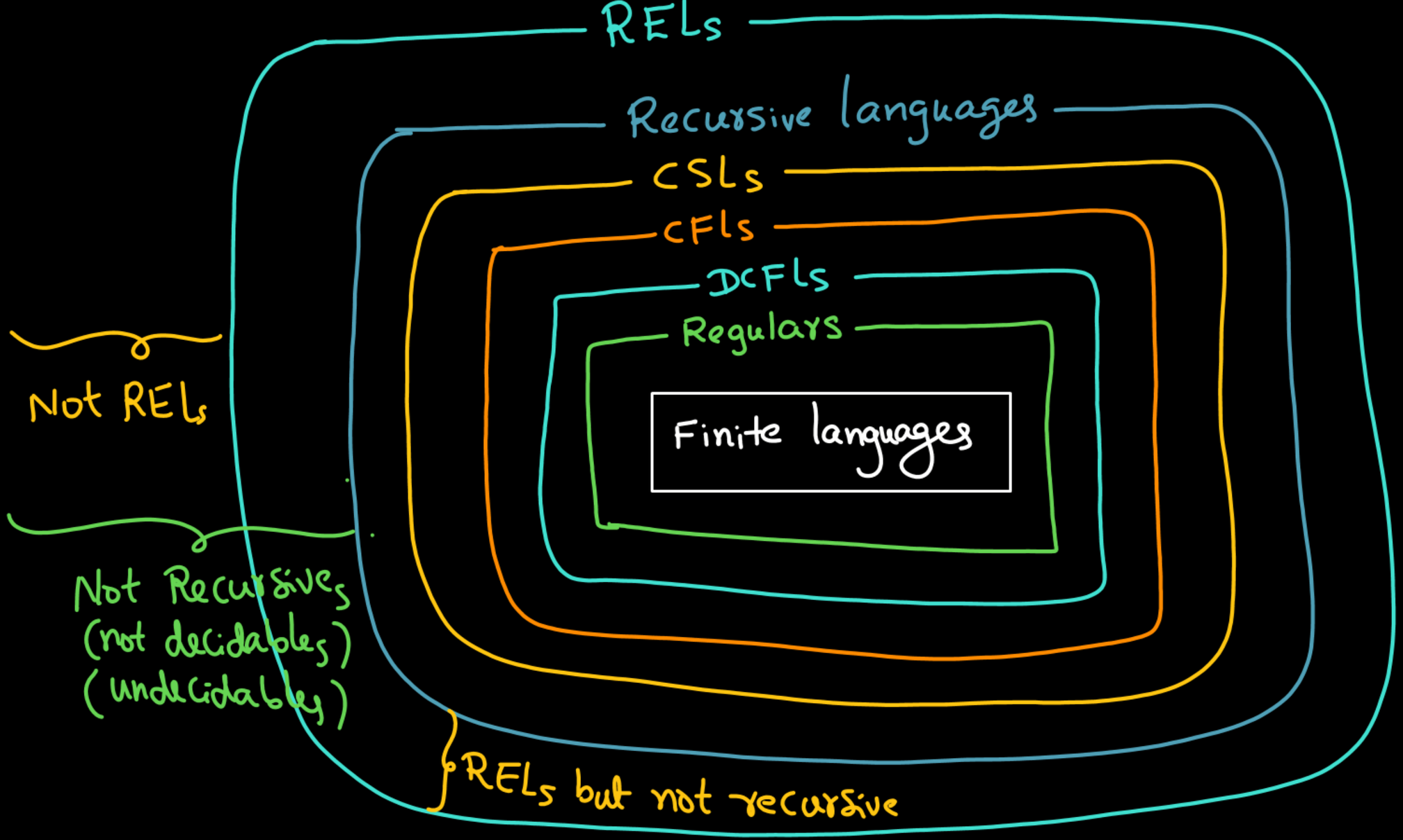
(Decidable Language) Recursive Language

① It is decidable by Tm
(It is acceptable by Tm that always halts)

② It is lexicographically enumerable by Tm

I) Every recursive language is Recursively Enumerable Set

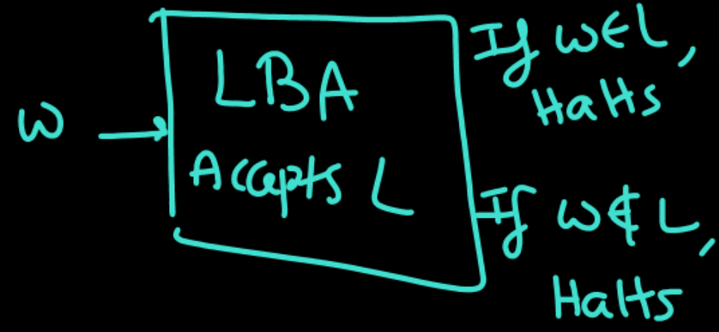
II) REL need not be Recursive Set


RELs
Recursive languages
CSLs
CFLs
DCFLs
Regulars
Finite languages
Not RELs
Not Recursives
(not decidables)
(undecidables)
RELs but not recursive

TM

LBA [CSL]

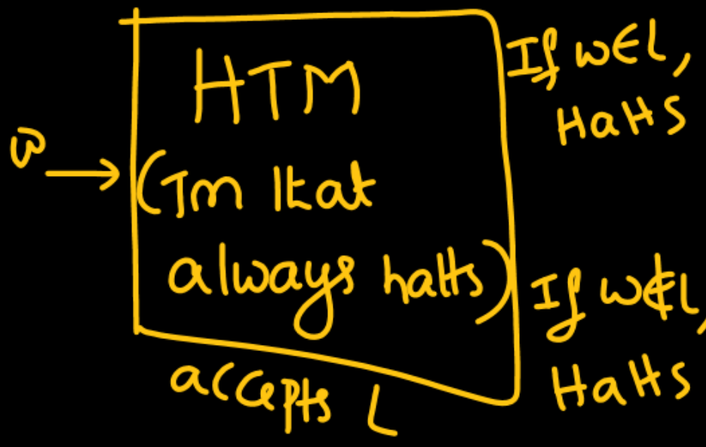
→ It is HTM but tape length is restricted to linearly bounded



Tape length is linearly bounded

HTM [Recursive Set]

→ It is TM but always halts



Halting always

TM [REL]

Every valid string, it halts

Types of TMs :

- Turing M/c
  - ≡
- One way infinite tape TM
  - ≡
- Two way infinite tape TM
  - ≡
- Single head TM
  - ≡
- multi-head TM
  - ≡
- Single tape TM
  - ≡
- multi tape TM
  - ≡
- Single dimensional tape TM
  - ≡
- multi dimensional tape TM

- multi head and multi tape TM
  - ≡
- Universal TM
  - ≡
- 2 stack PDA
  - ≡
- FA + 2 stacks
- PDA + 1 stack
  - ≡
- n stack PDA $(n \geq 2)$

→ If Turing m/c head is unidirectional then $TM \cong FA$

→ If TM tape is read only then $TM \cong FA$

→ If TM tape is read only and head is unidirectional then $TM \cong FA$

→ If TM always halts and tape is linearly bounded then $TM \cong LBA$

TM

TM
≅
Program
≅
$\forall w \in L$, Halts

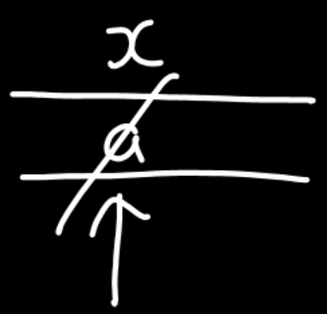
Logic exists for valid strings
We don't know about invalid
≅
Recursively Enumerable Set
≅
Semi-decidable language

HTM
≅
Algorithm
≅
Halting Program
≅
$\forall w \in L$, Halts
$\forall w \notin L$, Halts

logic exists for both valid & invalid
≅
Recursive language
≅
decidable language

TM construction:

In Tm:



i) change state

ii) write with some other symbol

iii) change state and write with some other symbol

Remember 'a'

In FA:

change state

In PDA:

i)

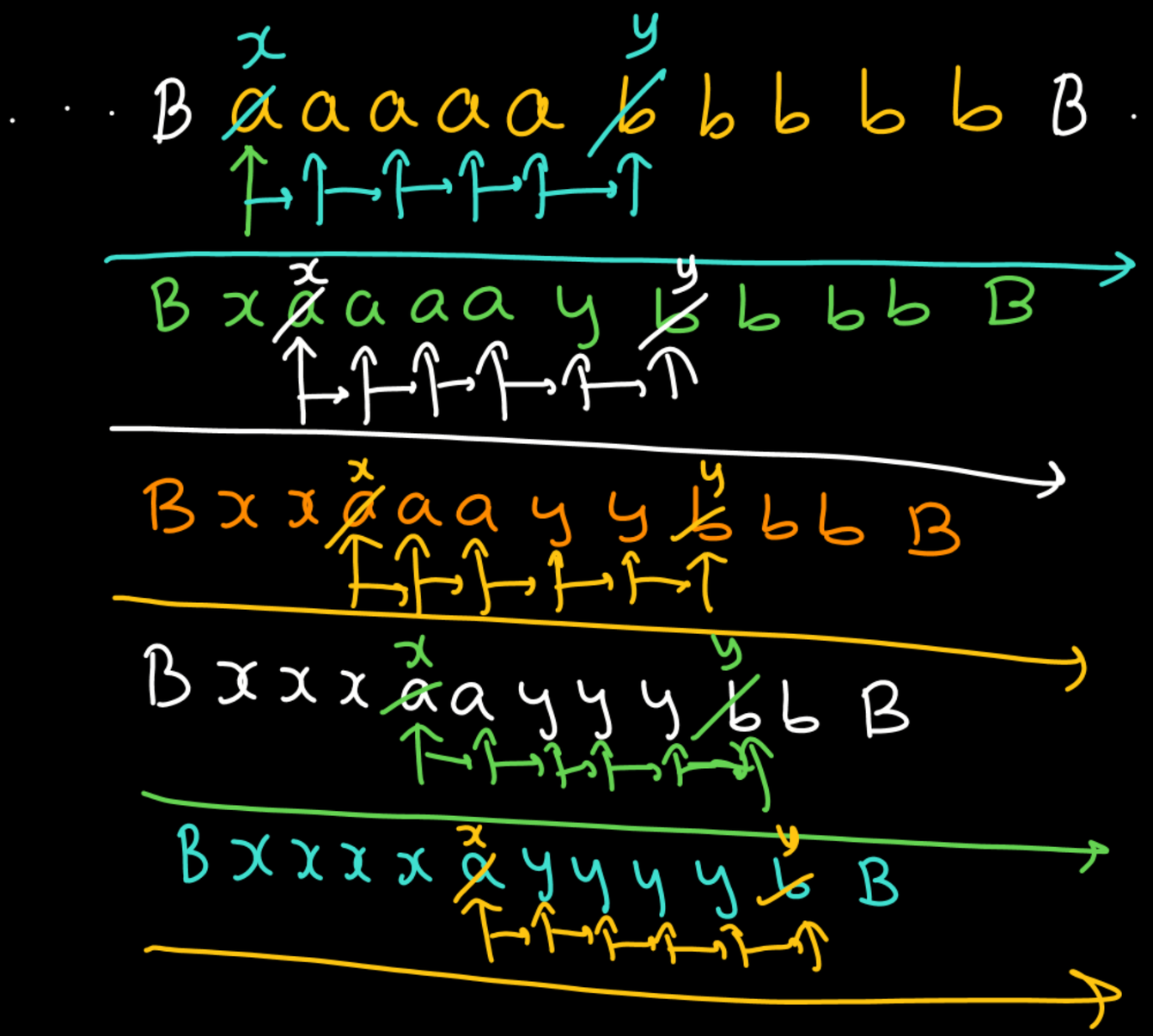
change state

ii) Push symbol onto stack

iii) Push symbol and change state

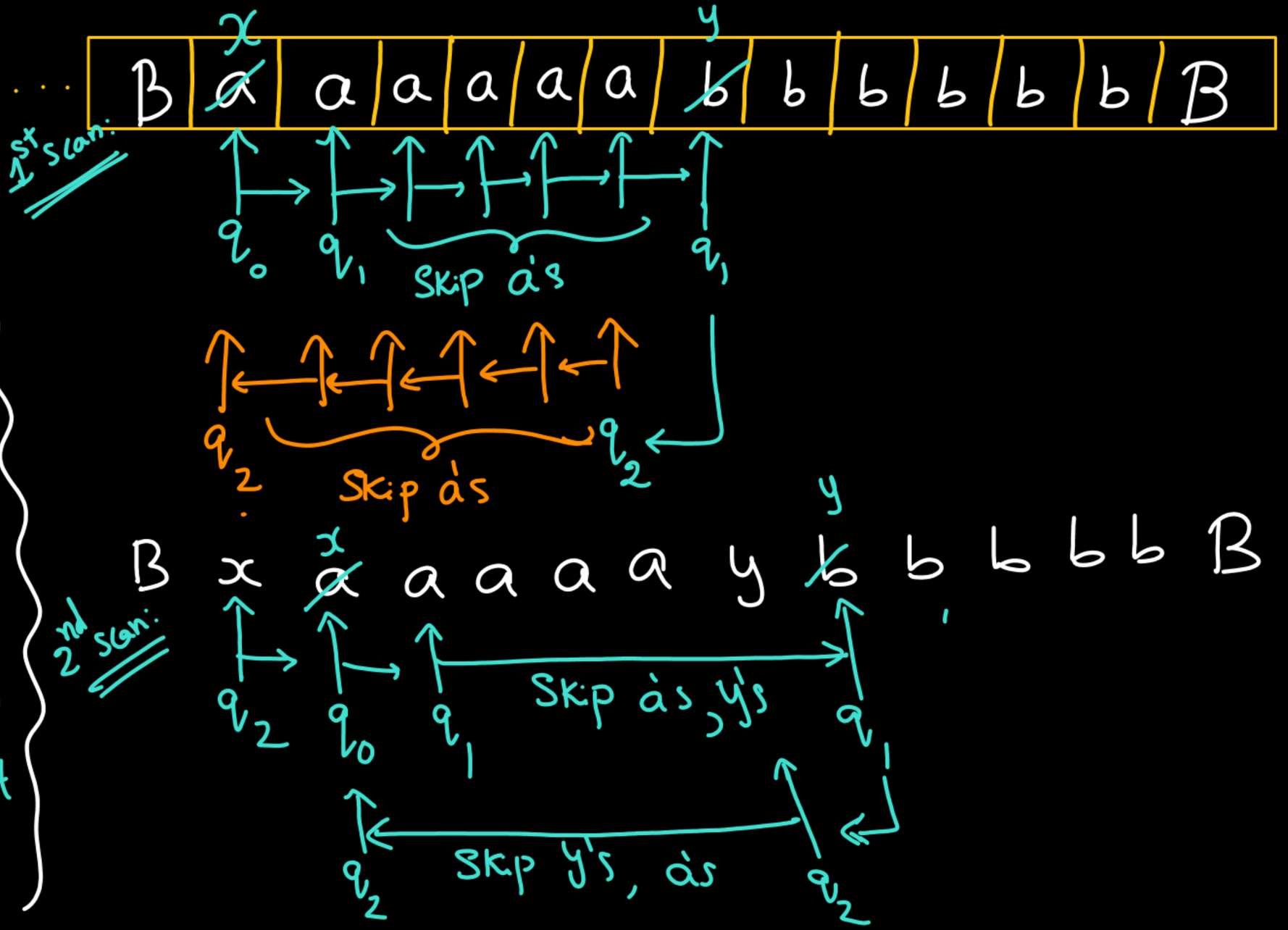
TM

① $L = \{a^n b^n \mid n \geq 0\}$



TM

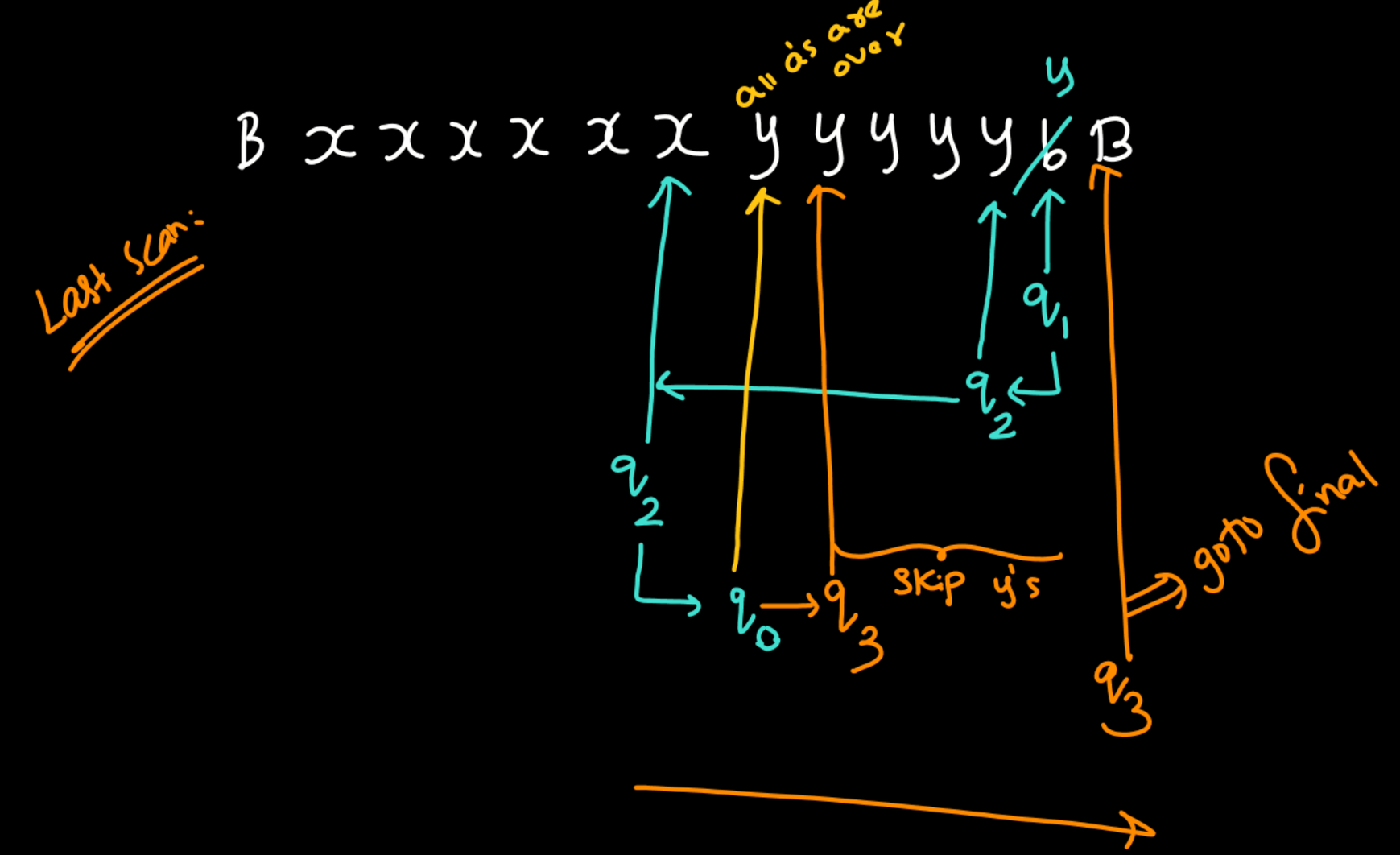
① $\{a^n b^n \mid n \geq 0\}$

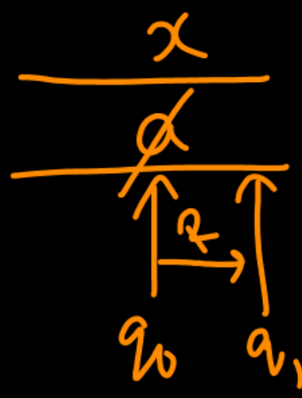


q_0 : Replace a with x

q_1 : Replace b with y

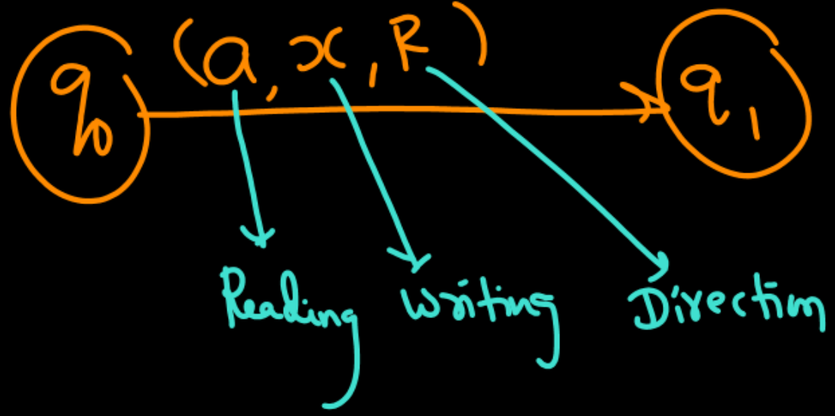
q_2 : Reverse scan to find first x

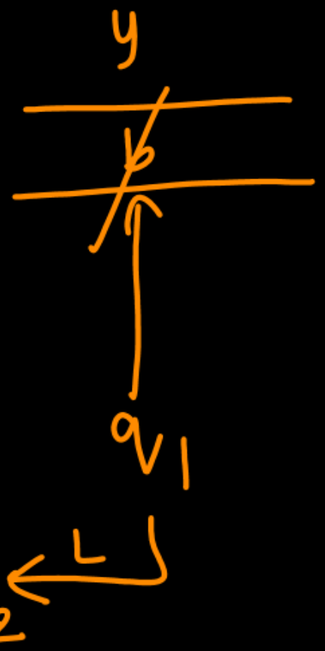

Last Scan:
B x x x x x x y y y y y b B
all a's are over
y
q_1
q_2
q_2
q_0
q_3
skip y's
q_3
goto final



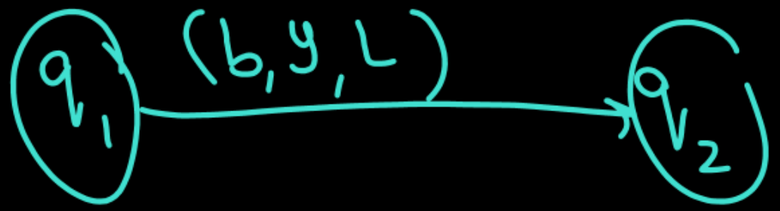
$$\delta(q_0, a) = (q_1, x, R)$$

Read, Write, Direction



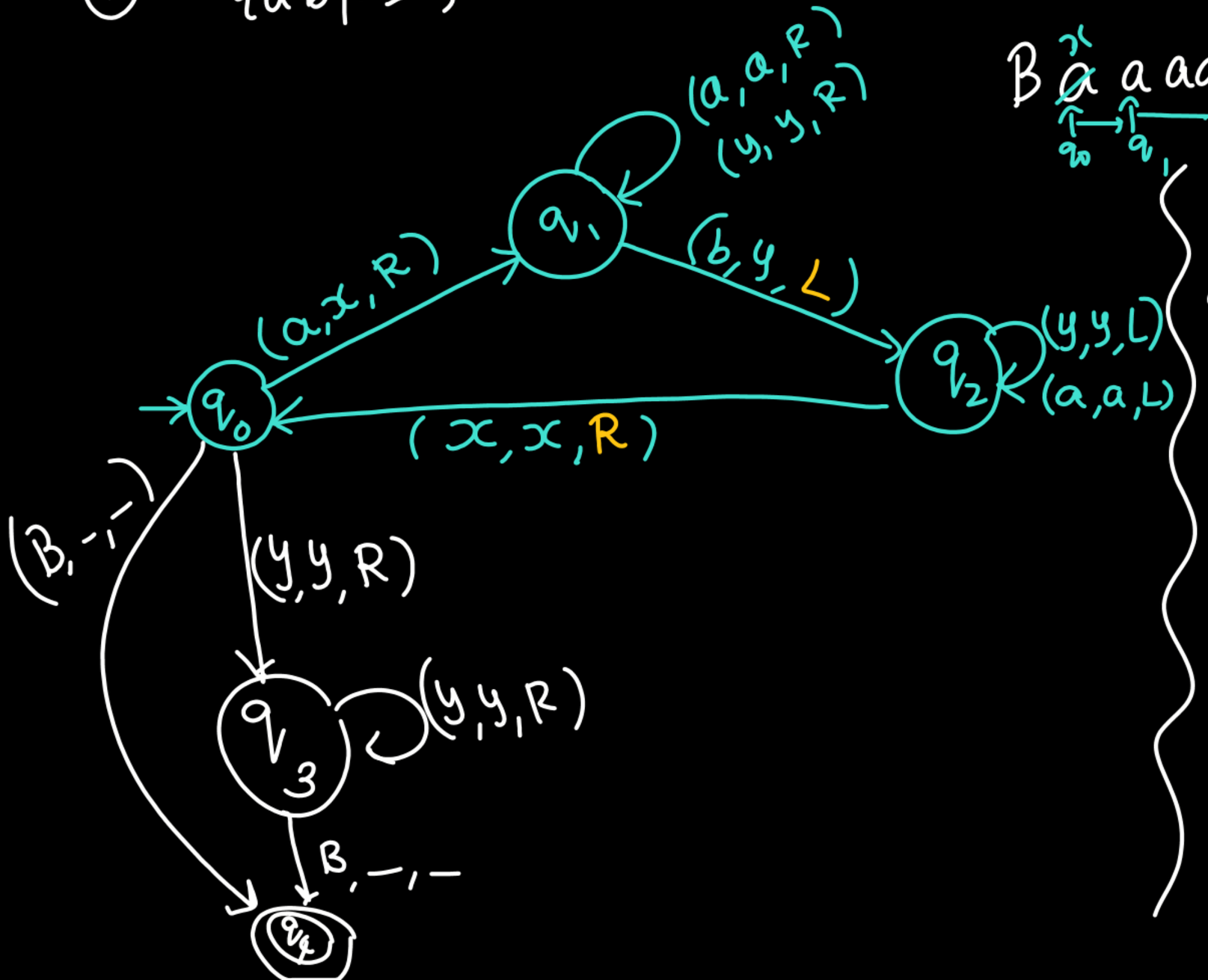


$$\delta(q_1, b) = (q_2, y, L)$$



TM

① $\{a^n b^n \mid n \geq 0\}$



q_0 : x ⊘ [Whenever y appears, move to q_3]

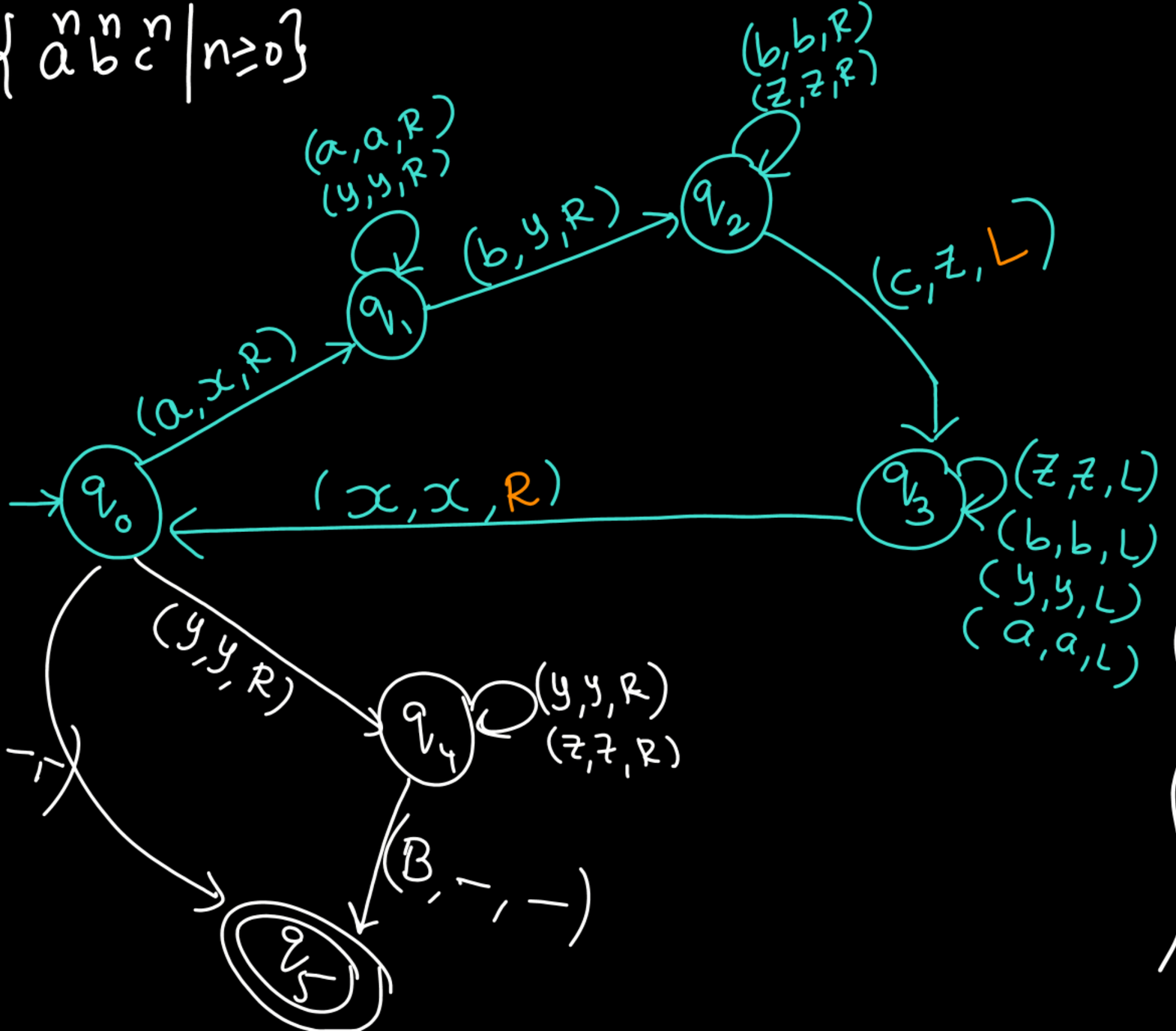
q_1 : y ⊘ (Skip a's & y's)

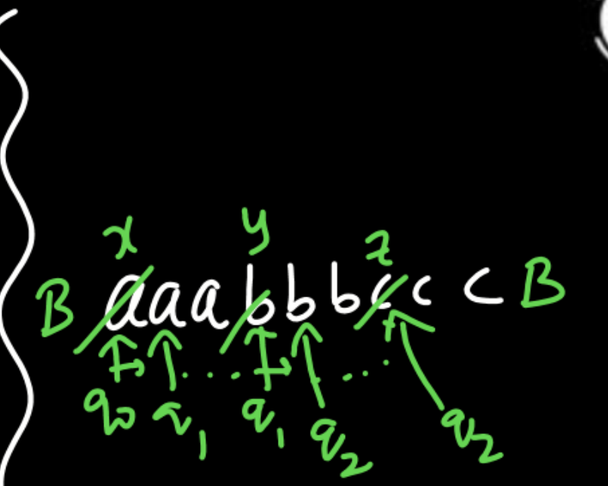
q_2 : Find 1st x in reverse direction (Skip y's & a's)

q_3 : Skip all y's to reach B

TM

② $\{a^n b^n c^n \mid n \geq 0\}$





q_0 : $\not{a}^{x}$

q_1 : $\not{b}^{y}$ (Skip às, ýs)

q_2 : $\not{c}^{z}$ (Skip ƀs, ẑs)

q_3 : Reverse sCan (Skip ẑs, ƀs, ýs, às)

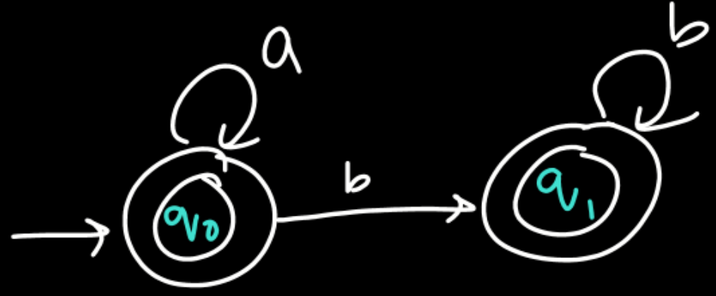
q_4 : Skip all ýs, ẑs

TM

③ $L = \{a^* b^*\}$

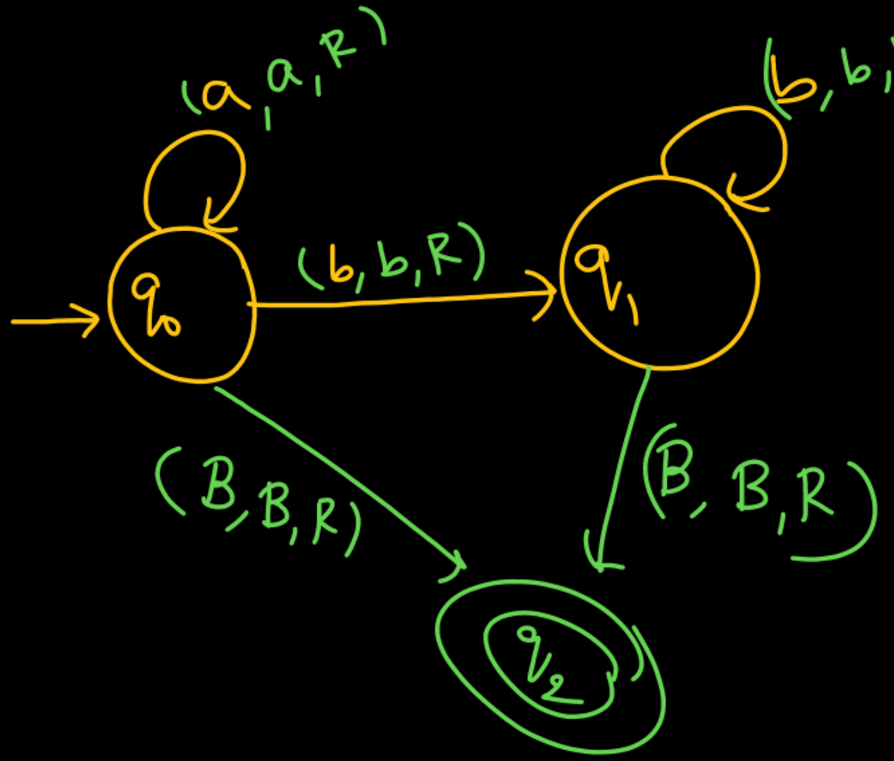
Every FA ⇓ be can convert to TM

FA:

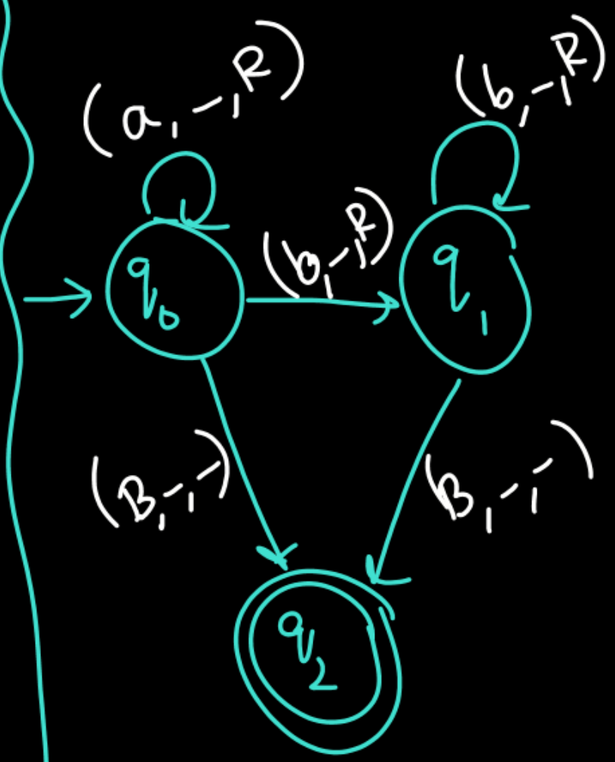


Tm:

Method 1:



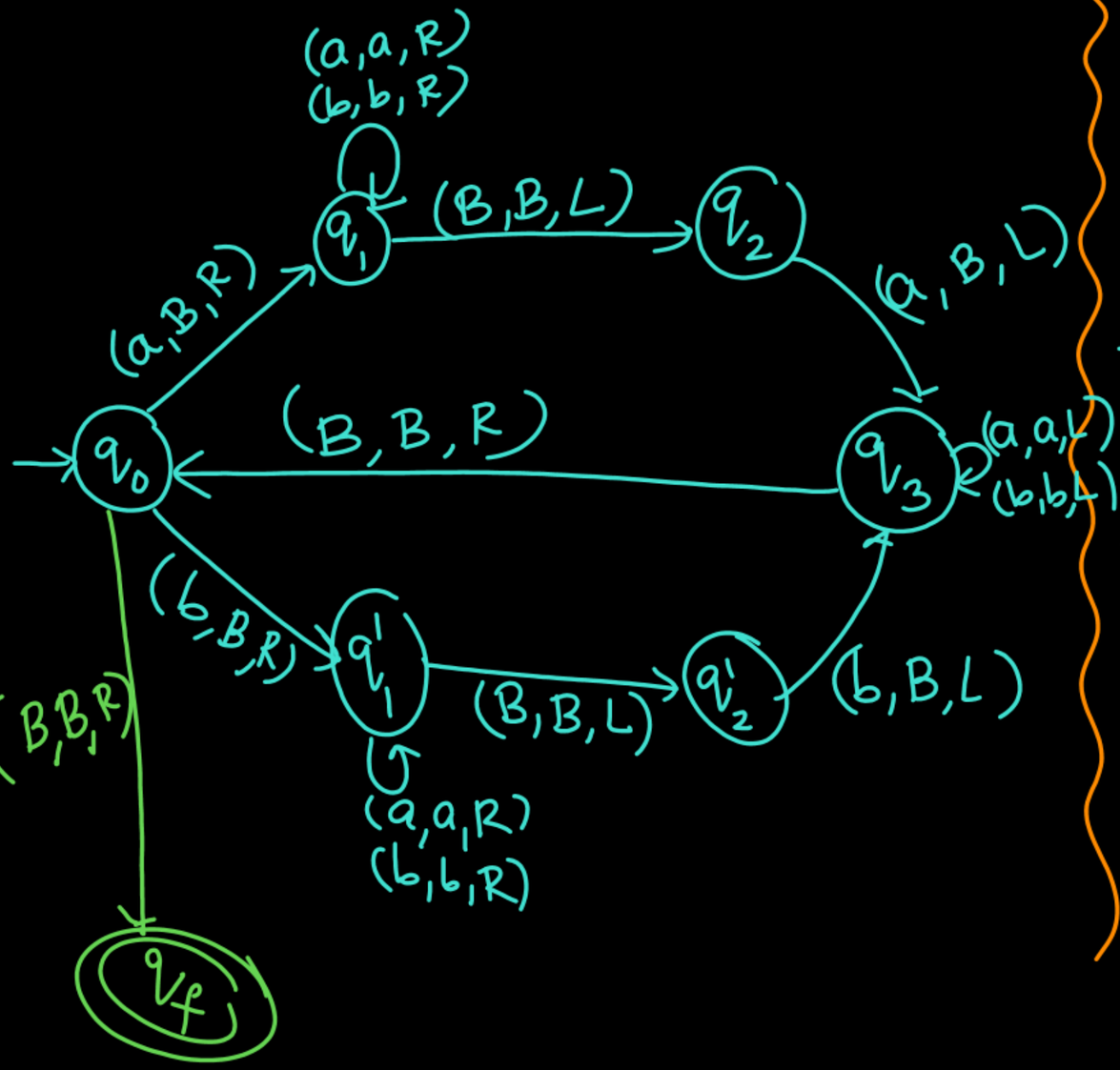
Method 2:

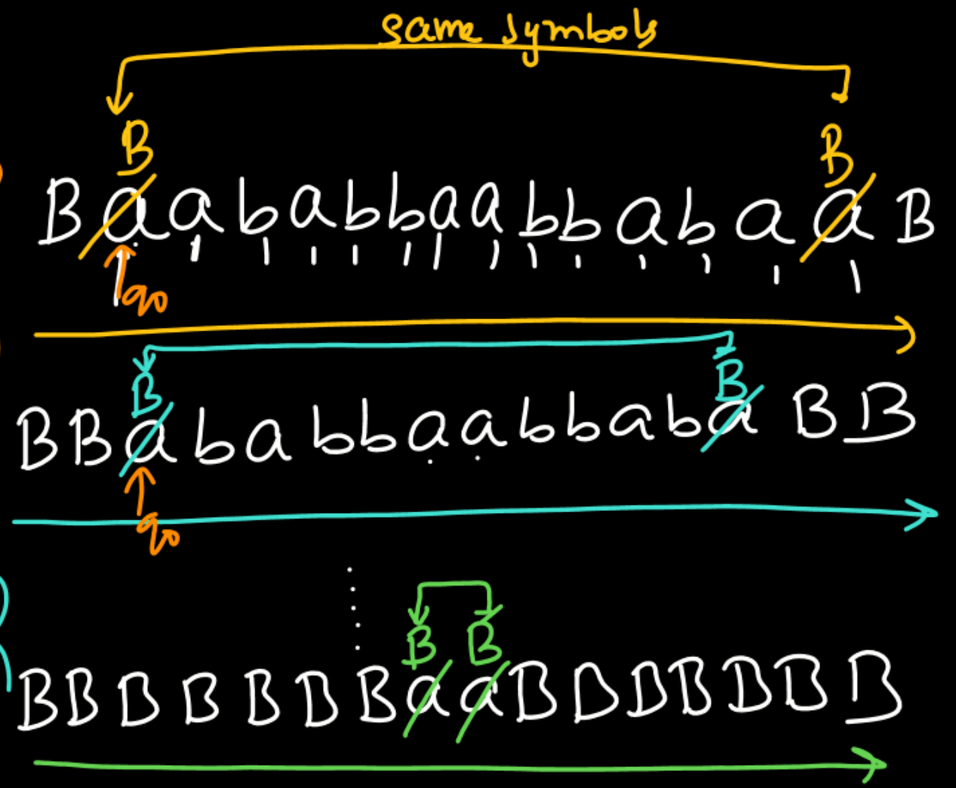


↦ Every regular language, we can design TM

TM

(4) $L = \{ ww^R \mid w \in \{a,b\}^* \}$





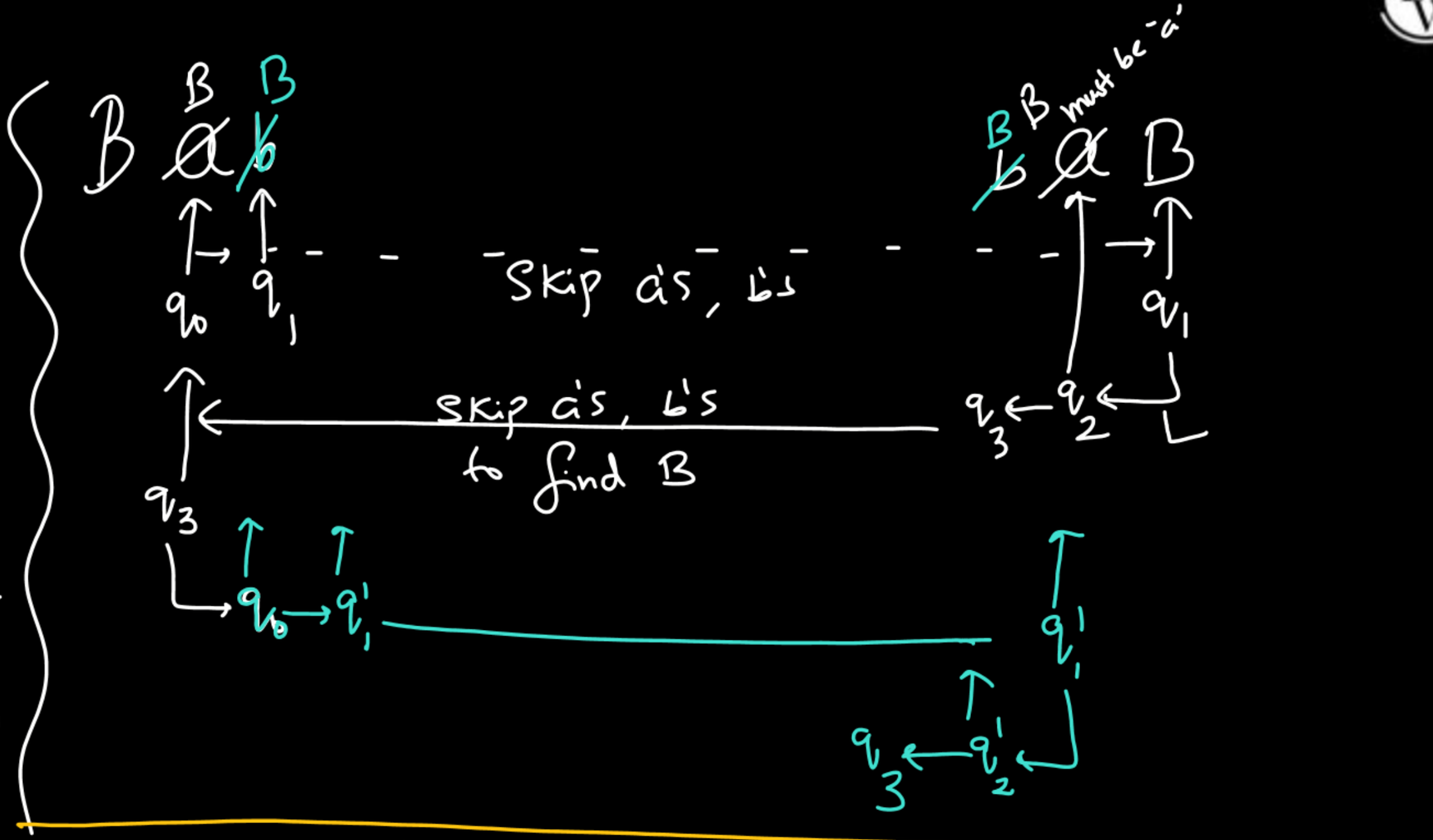
ε ✓

aa ✓

bb ✓

aaaa ✓

abba ✓



q_0 :

i) If input a, replace with Blank but goto q_1

ii) If input b, replace with Blank goto q_1'

iii) If B, goto final

q_1 : Skip a's, b's to find B

q_2 : Replace a with B

q_3 : Reverse Scan to find B

q_1' : Skip a's, b's to find B

q_2' : Replace b with B

Summary

H.W.: (5) $\{w\#w^R \mid w \in \{a,b\}^*\}$

(6) 1's complement of binary input

(7) 2's complement of binary input

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

(9) Decrement $f(x) = x-1$

(10) Unary addition

TM ✓

LBA Vs HTM Vs TM ✓

Recursive Vs RE ✓

Thank you