CS & IT ENGINEERING

THEORY OF COMPUTATION

Pushdown Automata

Lecture – 01

By- Venkat sir

Recap of Previous Lecture

Topic

Grammar

?????

Types of Grammar

Ambiguous Grammars

Regular Grammars → L.L.G, → R.L.G

CFG Simplification

CFG Normal forms → CNF, → GNF

Topics to be Covered

Topic Push down automata (PDA)

Topic ?? PDA Construction

Topic ?? Context free language

Topic ?? DCFL

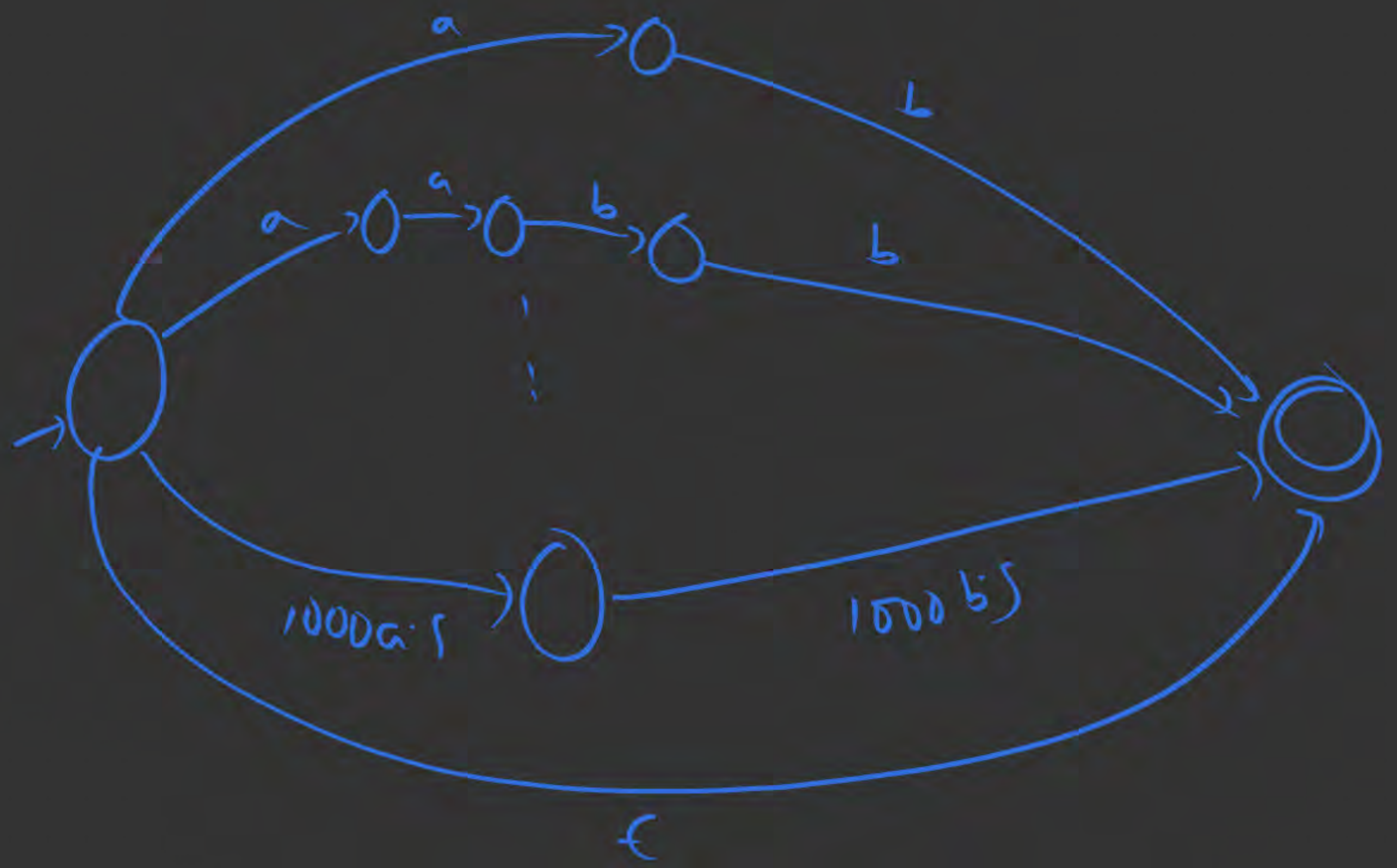
Topic : Pushdown Automata

F.A fails if Infinite memory required

(Infinite Dependency)

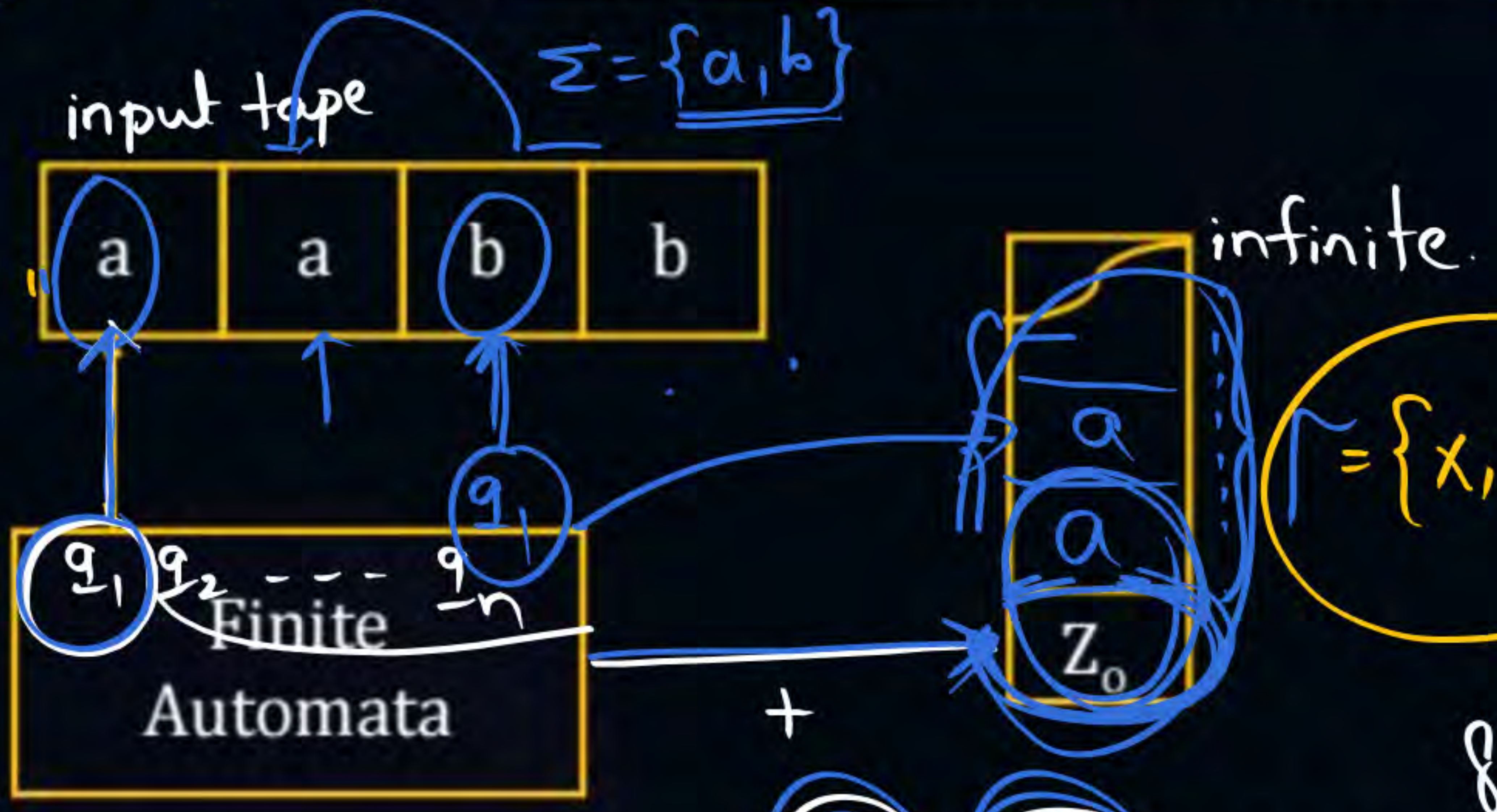
non regular $L_1 = \{a^n b^n \mid n \geq 1\} \to$ F.A fails

$L_2 = \{a^n b^n \mid n \leq 1000\} \to$ F.A accepts.


a
b
a
a
b
b
1000a's
1000b's
ε

Pushdown Automata (PDA)

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



$\Gamma = \{x, a, Z_0\}$

a's push

a's pop

aaa bbb

+

F.A + Stack = PDA

$\delta(q_1, i, \Gamma) =$

L R

transition function : $Q \times \Sigma \cup \{\epsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$

$(q_1, \text{Input}, \text{Stack}) = (q_1, \text{stack operation})$

push :

pop :

skip :

$(q_1, a, Z_0) = (q_1, aZ_0)$ push

$\delta(q_1, b, a) = (q_2, \epsilon)$ pop

Topic : PDA

- Finite Automata having additional power form of stack known as Push down automata.
- Size of stack in Push Down automata is infinite
- There exist only one type of push down automata i.e. “language recognisor”
- Push down automata can accept language in (deterministic way) or non-deterministic way

PDA
- → DPDA
- → NPDA

formal Definition

$$\text{PDA } (\overbrace{Q, \Sigma, \delta, q_0, F}^{F.A} \overbrace{Z_0, \Gamma}^{+ \text{ stack}})$$

Q :- Finite number of states

Σ :- Input alphabet $\longrightarrow$

q_0 :- initial state $\rightarrow$ only one

F :-set of final states $\rightarrow$ any final states

Z_0 :-initial stack symbol ✓

Γ :-stack alphabet $\neq$ set of symbols allowed into stack.

δ :- transition function

$$\boxed{Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \rightarrow Q \times \Gamma^{*})}$$

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

Q :- Finite number of states

Σ :- Input alphabet

δ :- Initial State

q_0 :- Set of final states

F :- Initial stack elements

Z_0 :- Stack alphabet

Γ :- transition function

$$\boxed{Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \rightarrow Q \times \Gamma\,)}$$

Topic : Note:

Note:- The following operation possible with PDA stack.

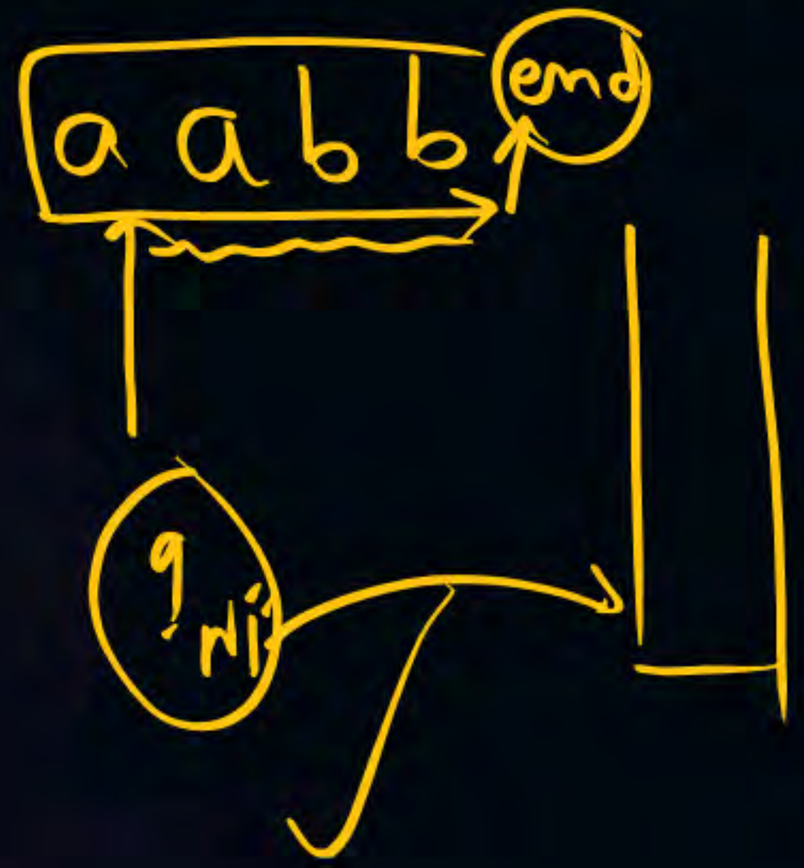
Push operation:- Moving i/p symbol from i/p buffer stack.

POP operation:- removing element from stack.

By pass operation:- don't push & don't pop (just reading symbol only)

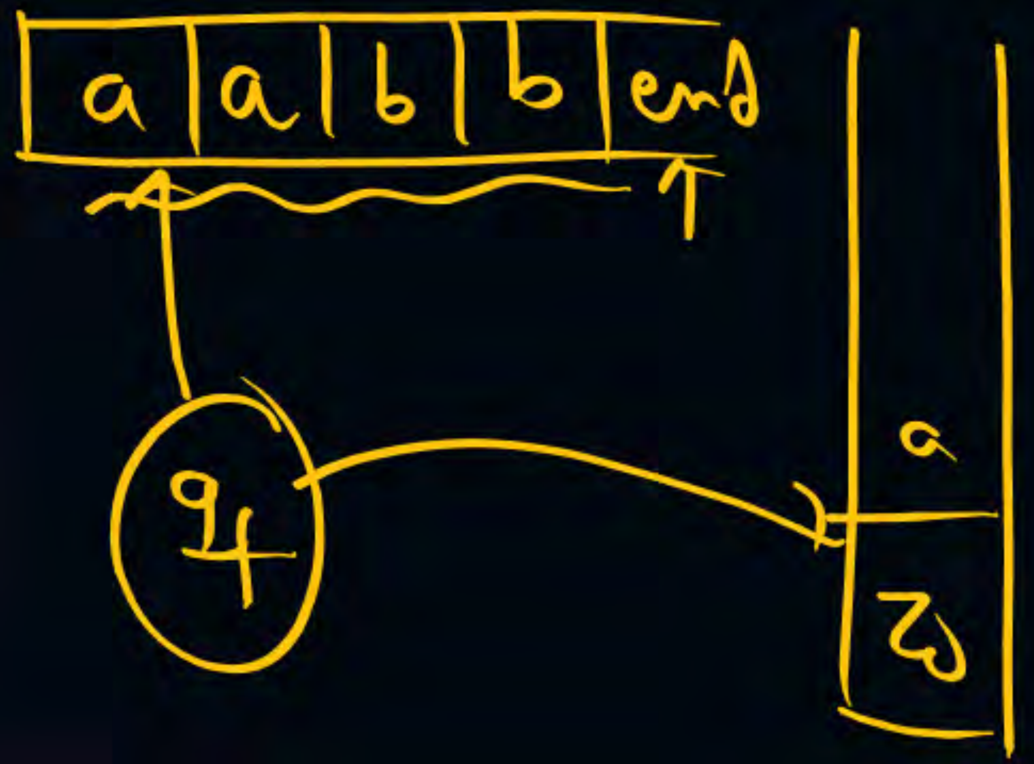
Topic : Empty Stack

By reading the string from left to Right by end of the string, if stack of the PDA is empty, then given string is accepted and irrelevent ~~inrelicent~~ of No of final state.



Topic : Final State

By reading the string from left to right, end the string PDA enters into final state then given string is accepted and ~~irradiant~~ irrelevent about stack is empty (or) not.



Topic : Note

PDA

DPDA < NPDA (more expressive power)

Note:- Number of language accepted by empty stack method and final state method is same not in PDA. (DPDA)

The language L is accepted by empty stack if and only if L should be final state.

The Default PDA is NPDA only

Topic PDA

- The expressive power of NPDA is more than DPDA.
- By Default PDA means NPDA.
- PDA practically used in compilers as parser.
- There are two types of acceptance method in PDA they are acceptance by empty stack and acceptance by final stack.

Notations:-

- Transition diagram
- Transitions

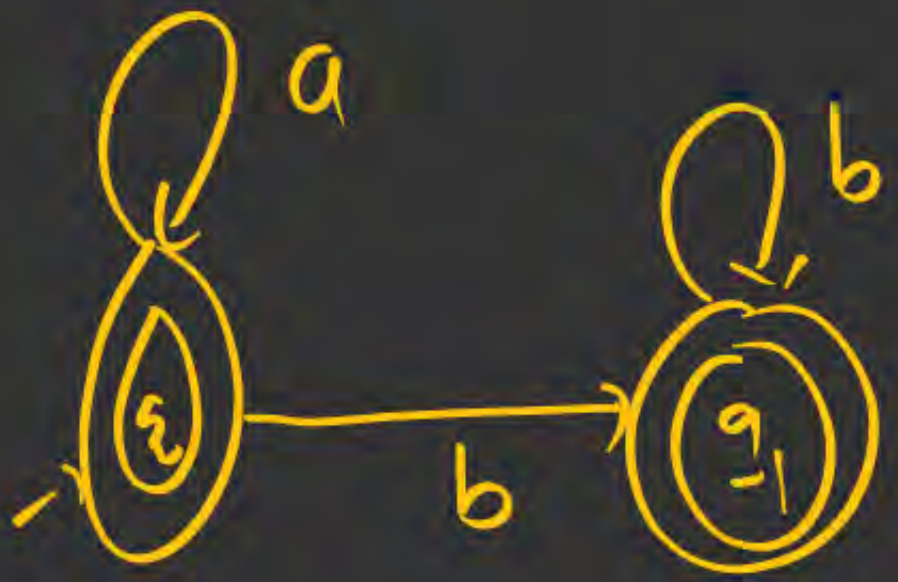
PDA (Acceptor)

- DPDA
- NPDA

a^*b^*

X

aba



$Q \times |\Sigma \cup \{\epsilon\}| \times \Gamma \rightarrow Q \times \Gamma^*$

input

Topic : Pushdown Automata

Construct DPDA for $L = \{a^n b^n \mid n \geq 1\}$

1. $(q_0, a, Z_0) = (q_0, aZ_0)$
2. $(q_0, a, a) = (q_0, aa)$
3. $(q_0, b, a) = (q_1, \epsilon)$
4. $\delta(q_1, b, a) = (q_1, \epsilon)$
5. $\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$ final state

(or)

$= (q_1, \epsilon)$ Empty stack

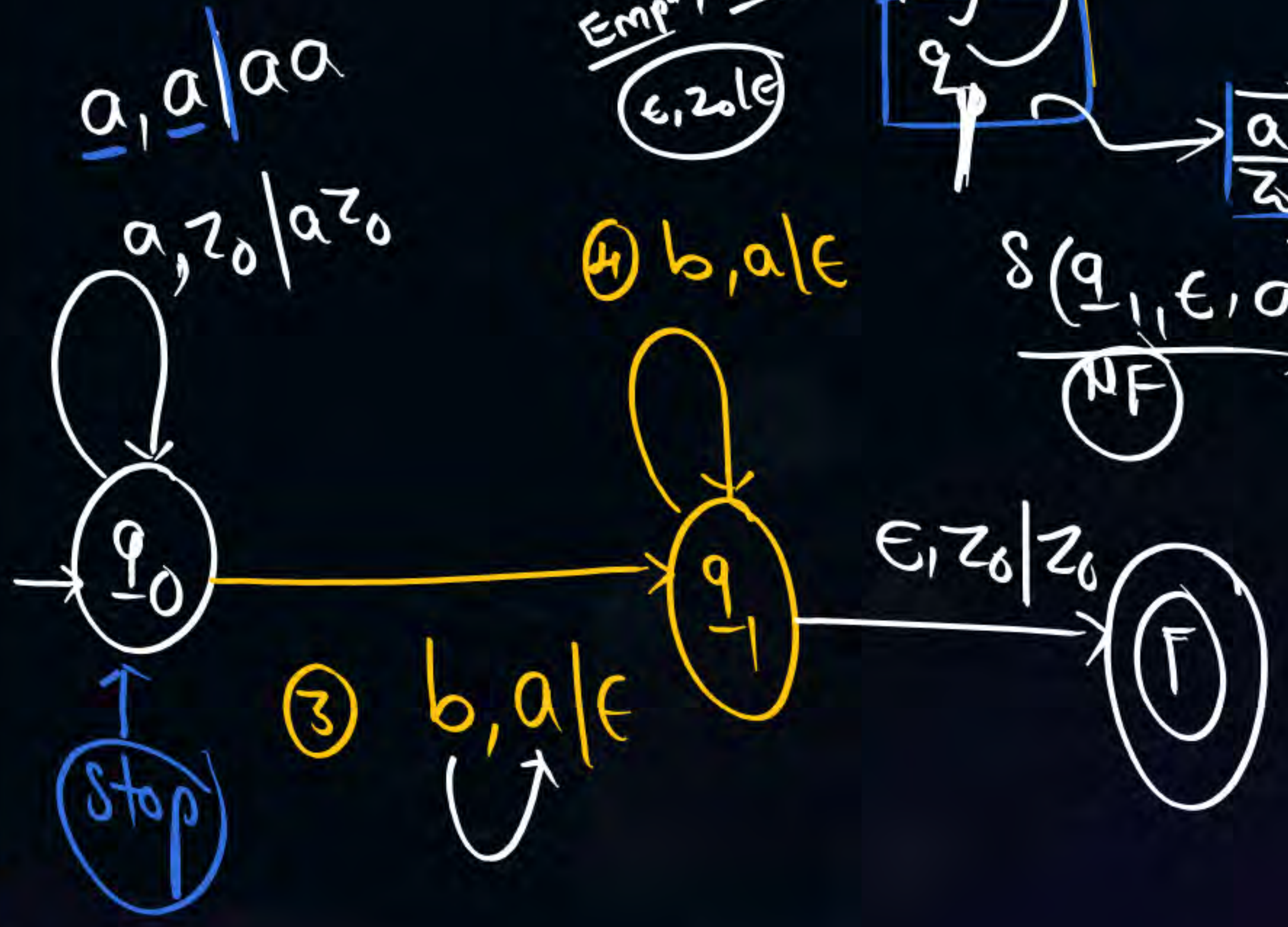
reject $a \mid a \mid b \mid \epsilon$ ×

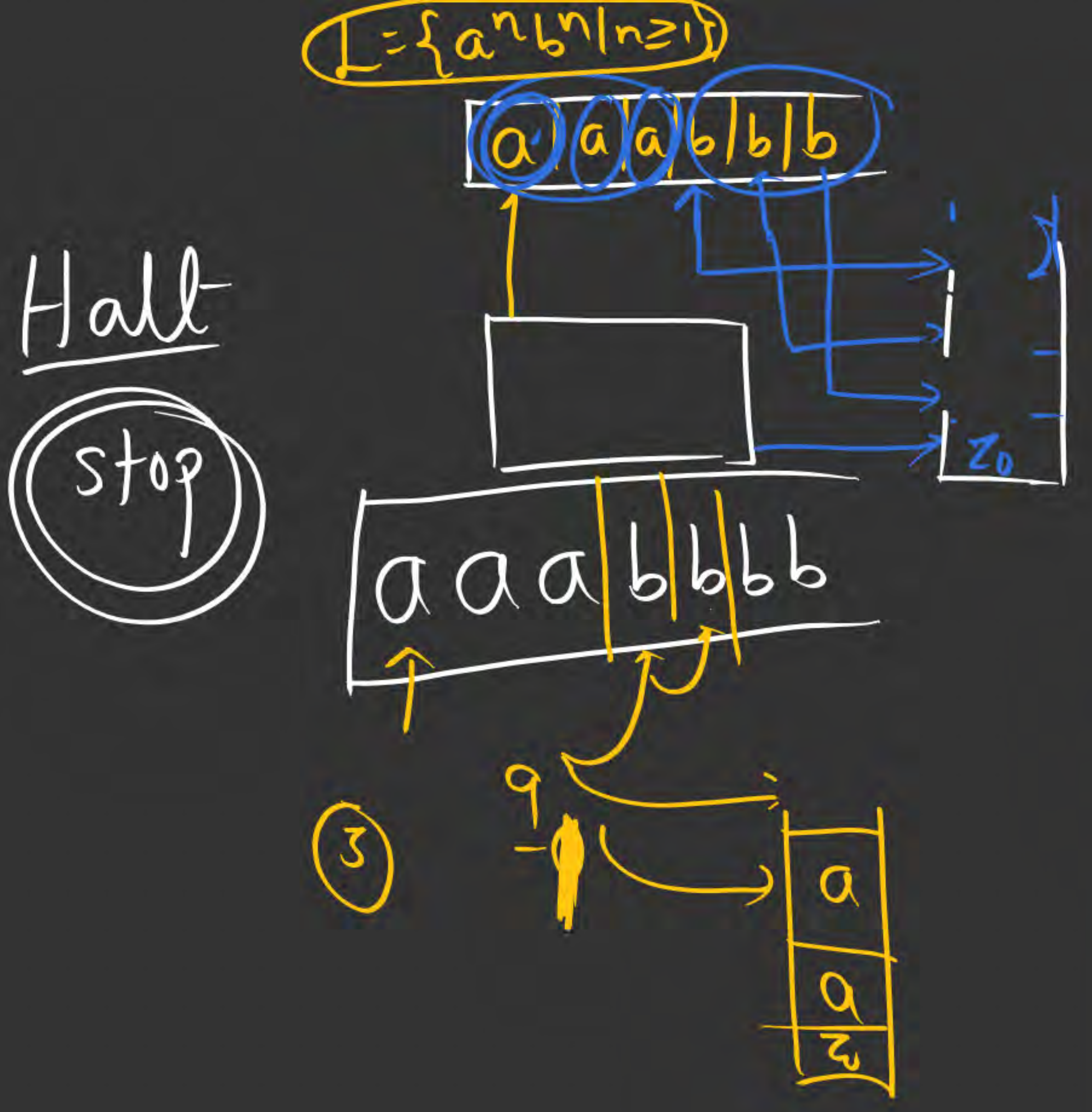
Empty stack

$\epsilon, Z_0 \mid \epsilon$

$\delta(q_1, \epsilon, a)$

NF





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

Logic:-

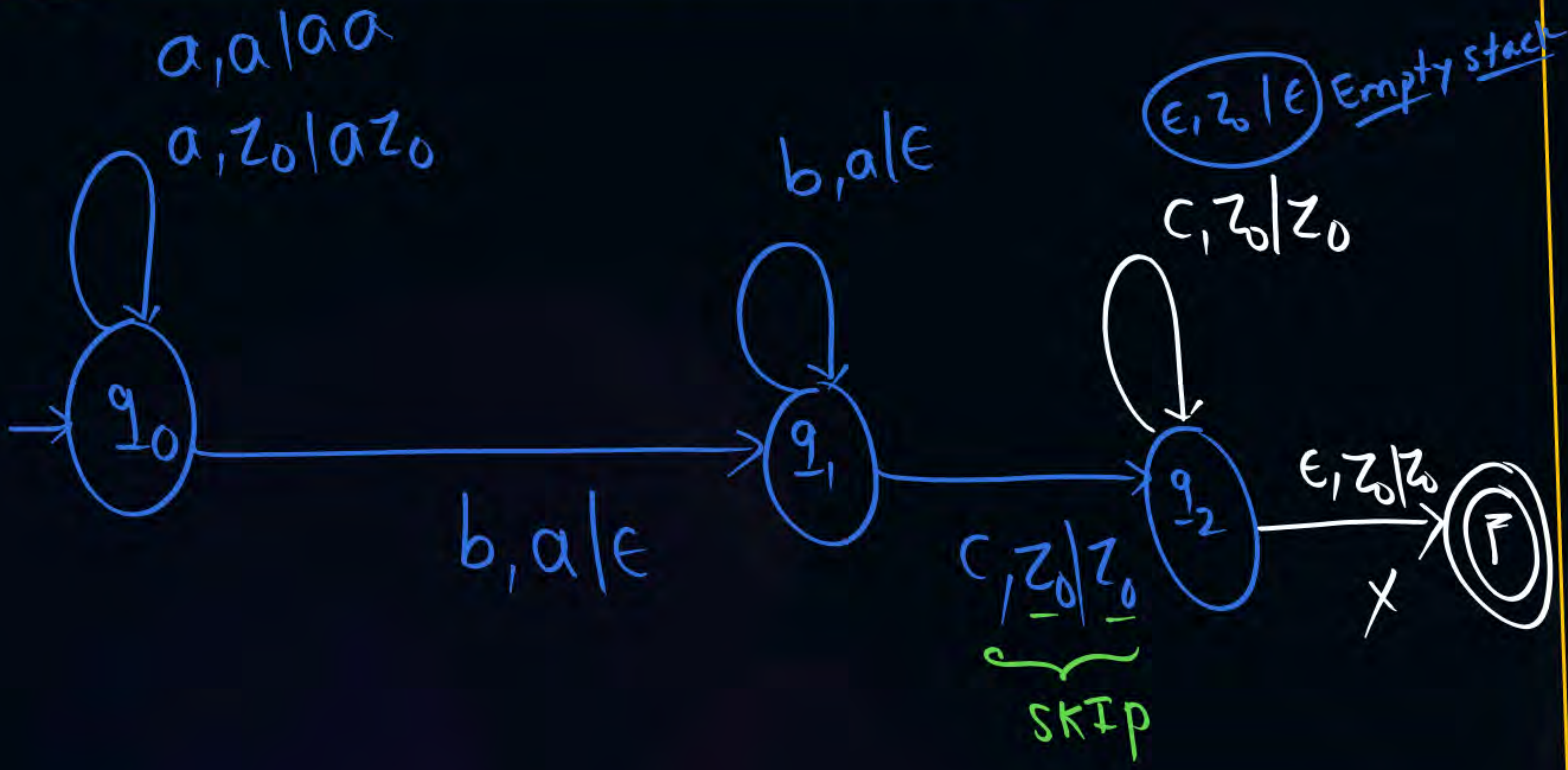
① All a's push into stack

② For every b one a's pop from stack

③ If input ended and top of stack is z_0 then accept

Construct PDA for $L = \{a^n b^n c^m \mid n, m \geq 1\}$

Topic : Pushdown Automata

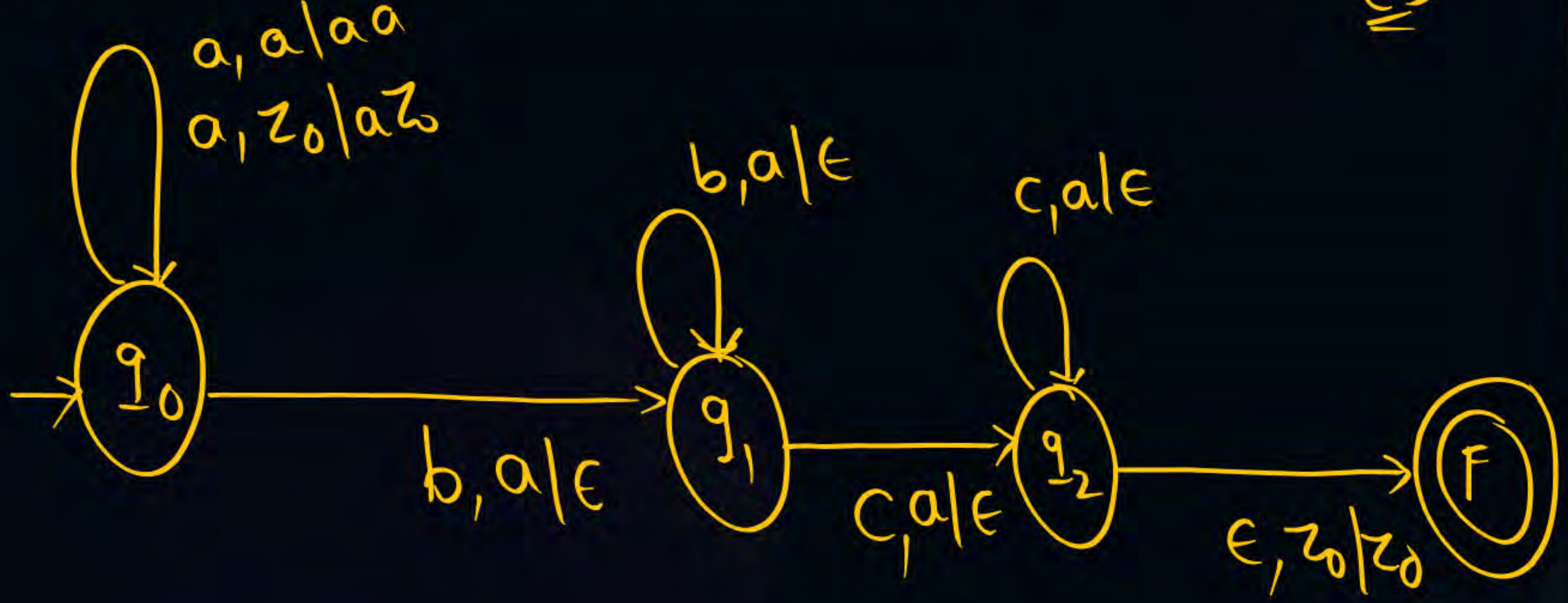


Logic:-

1. All a's push into stack
2. for every one b pop one a from stack
3. SKIP all c's
4. accept

Construct PDA for $L = \{ a^{n+m} b^n c^m \mid n, m \geq 1 \}$ ((final state))

Topic : Pushdown Automata



Logic:-

1. All a's → push.
2. for every b pop a
3. for every c pop a
4. If input ended & stack having Z_0 then accept.

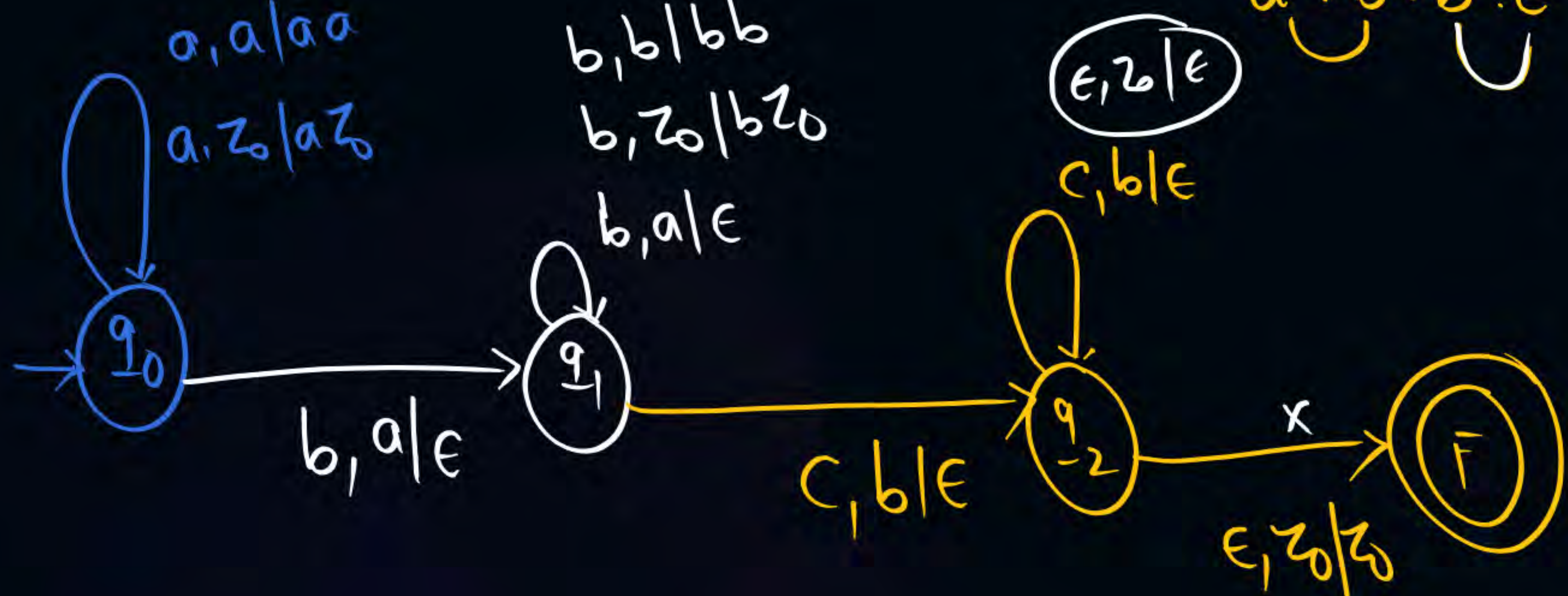
Home Work.

DPDA possible ✓ (final state)

Topic : Pushdown Automata

Construct PDA for $L = \{a^n b^{n+m} c^m \mid n, m \geq 1\}$

$a^n . b^n . b^m . c^m$



Logic :

① All a's → push

② for b's → pop a's

b's again push into stack

③ for every c → pop b

Home Work

Construct PDA for following languages.

1. $L = \{ a^n b^m \mid n > m,\ n, m \geq 1 \}$

2. $L = \{ a^n b^{2n} \mid n \geq 1 \}$

3. $L = \{ a^n b^{2^n} \mid n \geq 1 \}$

[MCQ]

#Q. Let N1 n_1 is number of language accepted by using empty stack method. N2 n_2 is number of lang accepted by using final state then which of the following is true for PDA ?

A. $n_1 = n_2$

B. $n_1 > n_2$

C. $n_1 < n_2$

D. We can't say

[MCQ]

#Q. Size of the stack is restricted to 10000 element only in PDA than the lang accepted by that type of PDA is-

A Regular Lang

B CFL but Not Reg.

C Finite lang

D Reg. but not Reg.

Note:-

- Lang accepted by push down automata known as CFL

Context free language

- The expressive power of PDA is more then finite automata because PDA can accept regular language as well as CFL.

F.A → Regular only

PDA → CFL & Regular

PDA > F.A

-

Topic : Drawback of PDA

PDA fails to accept language which requires more than one stack.

Ex:- L= $\{a^n b^n c^n | h \geq 1|\}$

The language for which PDA Not possible known as non-cfL.

Construct PDA for the language L = $\{a^n b^n / n \geq 1\}$

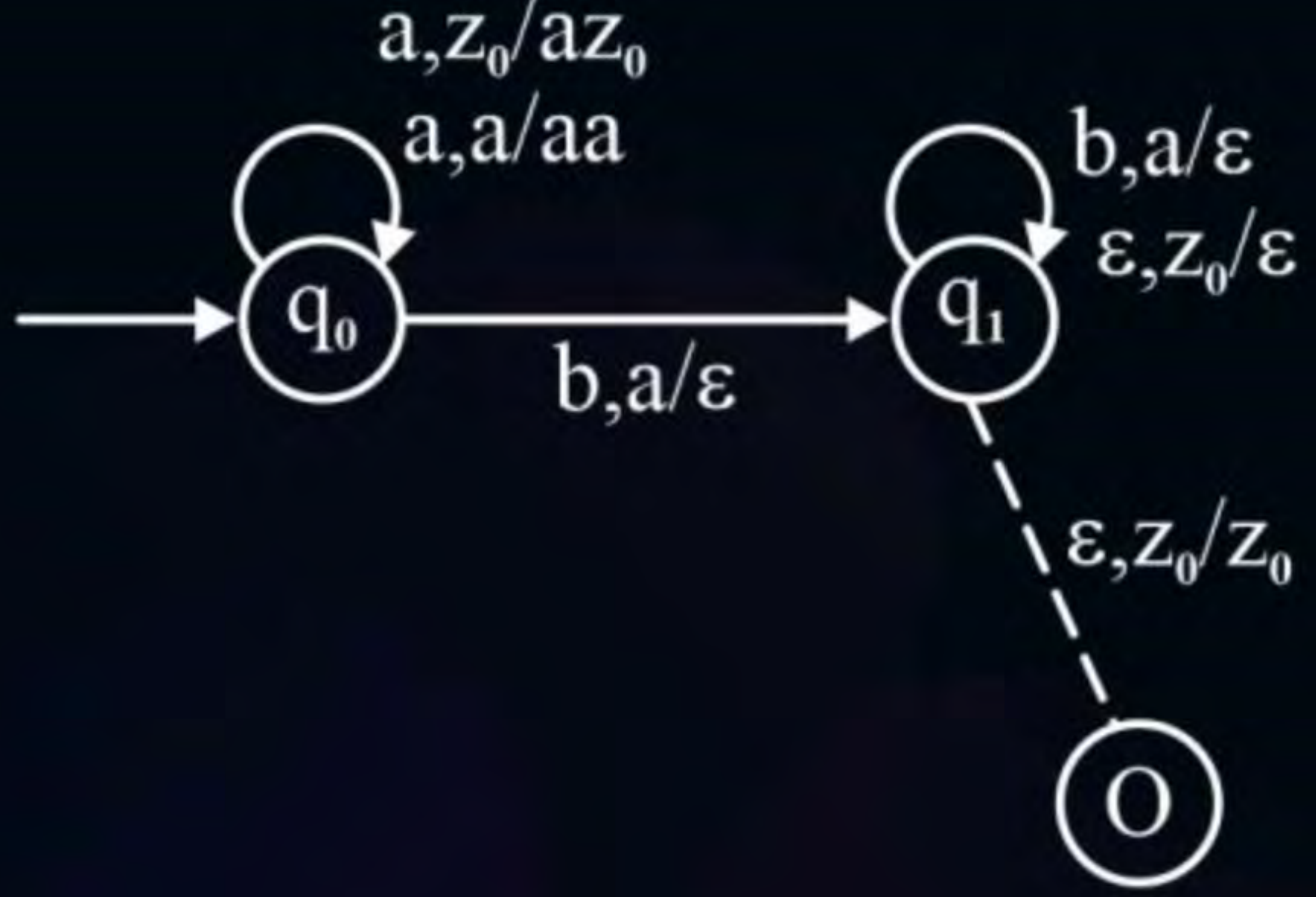
$$\delta: \; Q \times \Sigma \cup B \times \Gamma \rightarrow Q \times \Gamma)$$

a's Push $\left[\begin{array}{l} S(q_0, a, z_0) = (q_0, az_0) \\ S(q_0, a, a) = (q_0, aa) \end{array}\right.$

b's PED $\left[\begin{array}{l} S(q_0, b, a) = (q_1, \varepsilon) \\ S(q_1, b, a) = (q_1, \varepsilon) \end{array}\right.$

accepted $\left[S(q_0, \varepsilon, z_0) = (q_1, \varepsilon) \right.$

Empty stack



a,z0/az0
a,a/aa
b,a/ε
ε,z0/ε
q0
b,a/ε
q1
ε,z0/z0
O

Topic : Note

Note:- By reading the input string by the end of the string stack is non empty or starting is not ended is-

Whenever m/c is halted then that i/p is rejected.

- the input is valid only string is ended end 2 not be in there stack.
- In final state mechanism i/p is valid only when automata enters into final state whenever m/c is halted.

2 mins Summary

Topic One

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU