# CS & IT


# ENGINEERING

## THEORY OF COMPUTATION

### Pushdown Automata

Lecture – 01


By- Venkat sir

# Recap of Previous Lecture

**Topic**

Grammar

?????

1. Construction
2. language Detection
3. types of grammar
4. Ambiguou grammar
5. Simplification of grammar
6. Normal form of grammar

# Topics to be Covered

**Topic** Push down automata (PDA)

**Topic** ?? PDA Constuction

**Topic** ?? PDA $\Rightarrow$ language

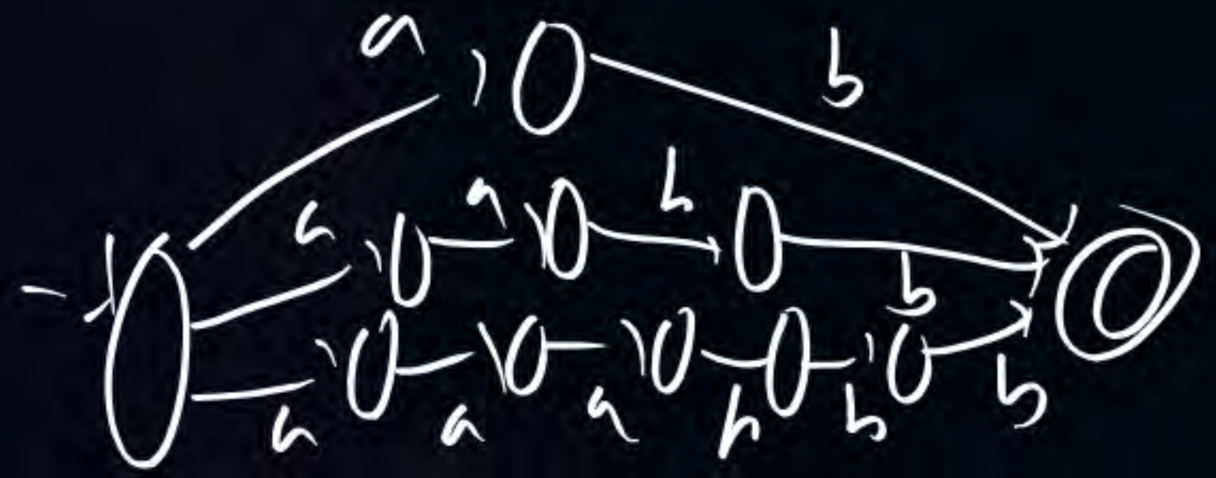
**Topic** ?? CFL detection

Closure properties

# Topic : Pushdown Automata

$L_1 = \{a^n b^n \mid n \geq 1\}$ ✗

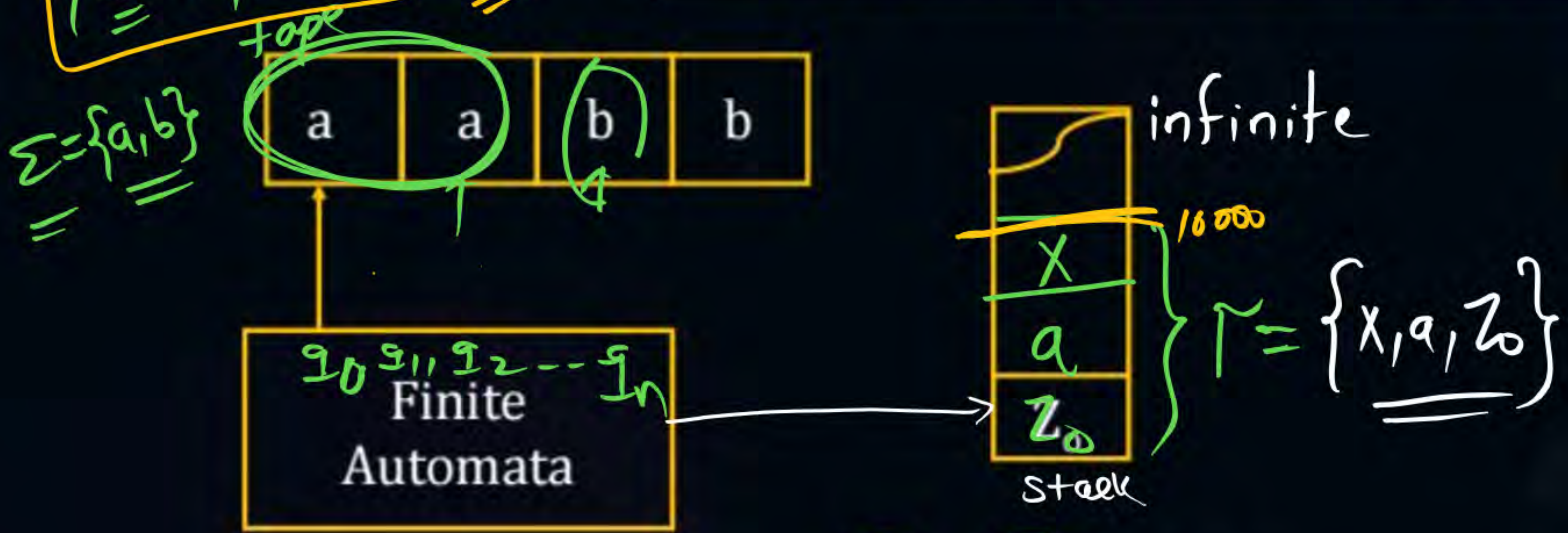
$L_2 = \{a^n b^n \mid n \leq 5\}$ = F.A ✓

# Pushdown Automata (PDA)

(CFL)

$\{a^n b^n / n \geq 1\}$ CFL




F.A. + Stack = PDA

Reg

## Topic : PDA

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

{ formal Definition }

stack

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

$Q$ :- Finite number of states

$\Sigma$ :- Input alphabet

$q_0$ :- initial state

$F$ :-set of final states

$Z_0$ :-initial stack symbol

$\Gamma$ :-stack alphabet

$\delta$ :- $\boxed{\text{transition function}}$


op

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

$\delta(q_1, a, X) = (q_2, aX)$ (push)

$\delta(q_2, b, X) = (q_3, \epsilon)$ pop

$\delta(q_3, a, Y) = (q_3, Y)$ (skip)

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

## Topic : Empty Stack

①

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 of No of final states.

## 2 Topic : Final State

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

## Topic : Note

**Note**:- Number of language accepted by empty stack method and final state method is same in PDA.

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

# Topic PDA

PDA

DPDA < NPDA

≠

DFA = NFA

- (1) The expressive power of NPDA (more) is more than DPDA (less).
- (2) By Default PDA (NPDA) means NPDA.
- (3) 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:

- ① Transition diagram
- ② Transitions

PDA (Acceptor)

- ① DPDA
- ② NPDA

## Topic : Pushdown Automata

(Q) Construct PDA for $L=\{a^n b^n \mid n \ge 1\}$

Logic


Transitions $\boxed{Q \times \Sigma \cup \{\epsilon\} \times \Gamma \to Q \times \Gamma^*}$

$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$ {push} all a's

$$\delta(q_0, a, a) = (q_0, aa)$$

$$\delta(q_0, b, a) = (q_1, \epsilon)$$ (pop) a's from stack

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

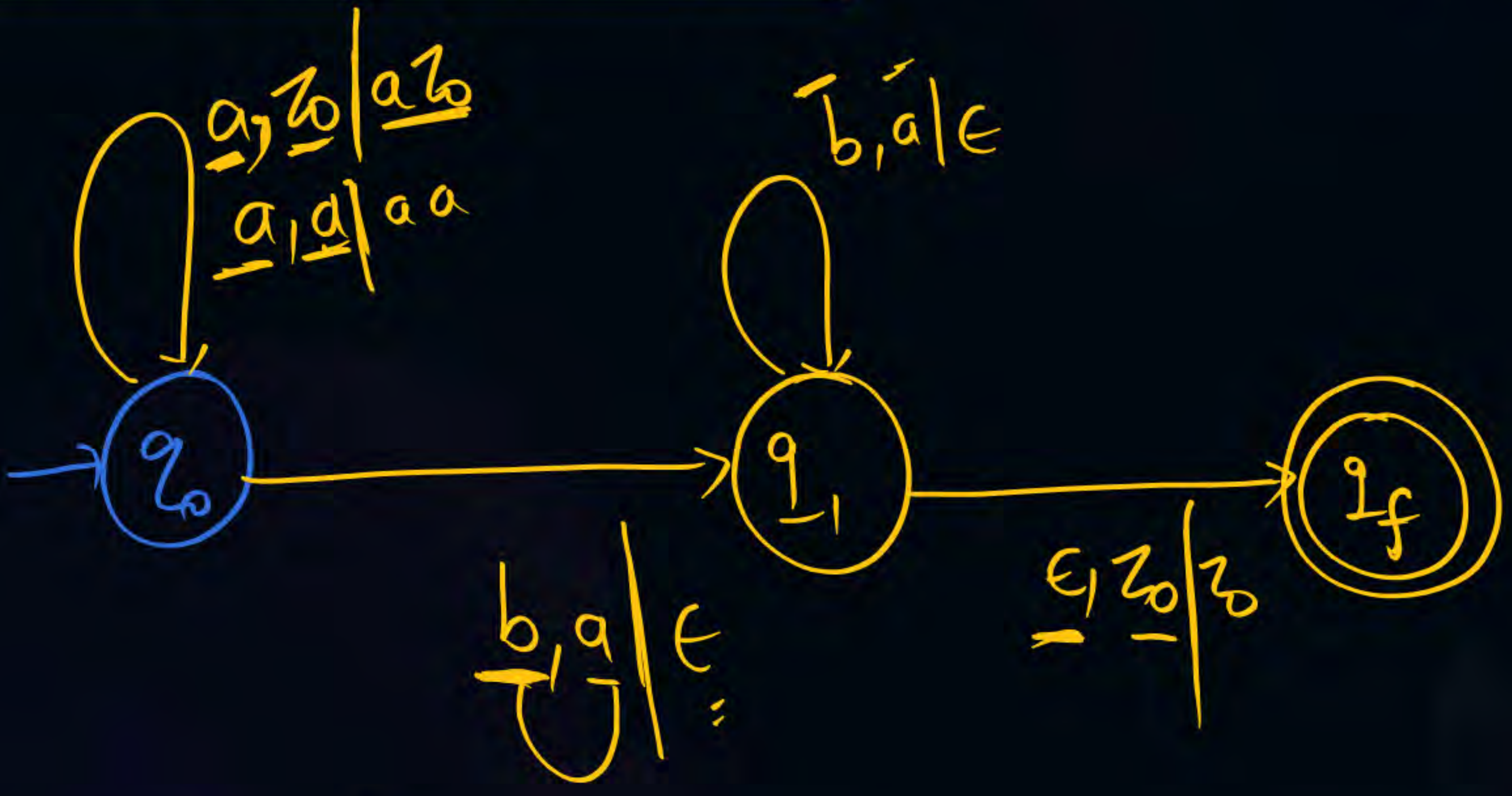
$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$$ (final state)

Logic

1. all a's push into stack
2. for every 1 b -> 1 a pop from stack.
3. accept

Topic : Pushdown Automata

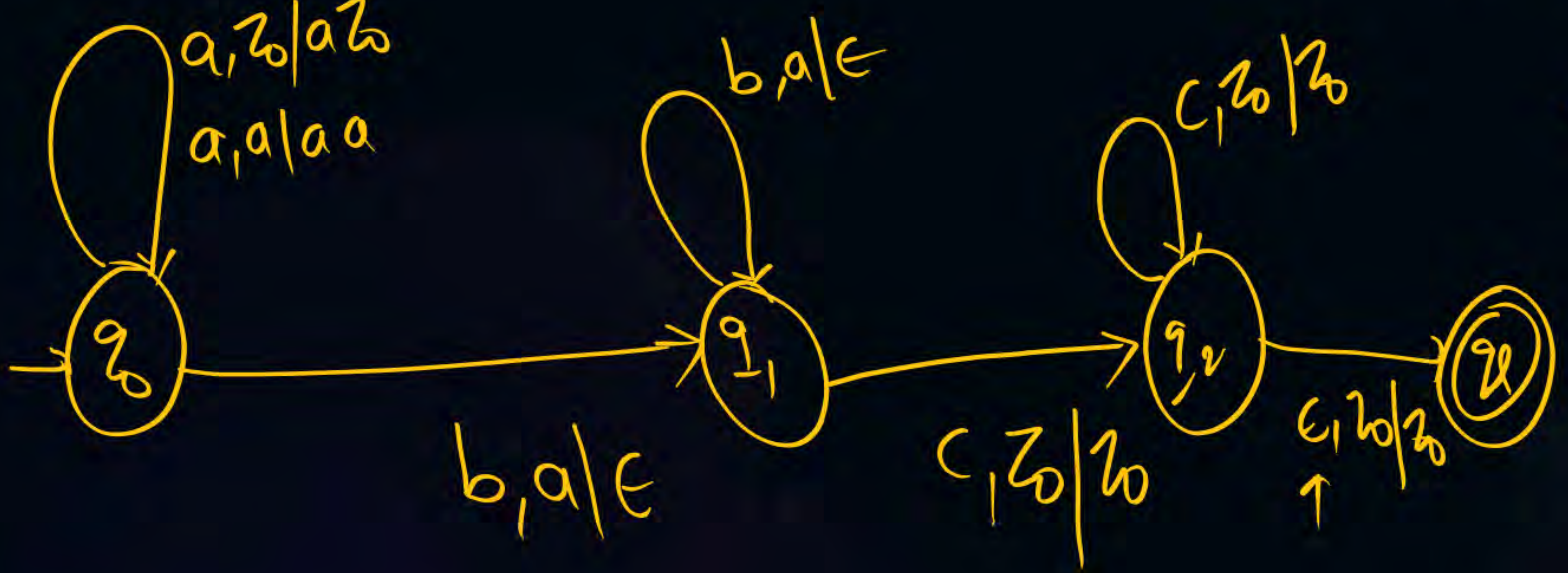
$\{a^n b^n \mid n \geq 1\}$




transition diagram

## Topic : Pushdown Automata

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



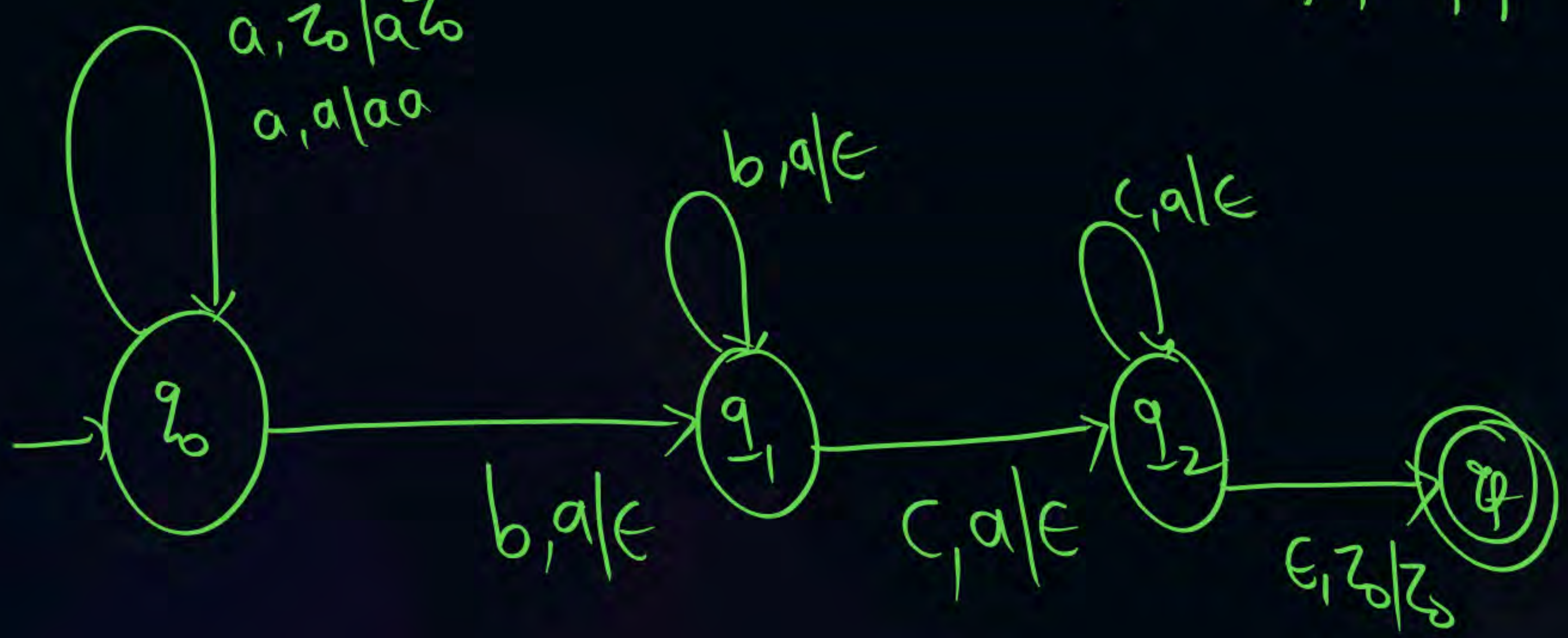

Logic

1. a's → push
2. b's → pop a's
3. c's → SKIP

## Topic : Pushdown Automata

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

push — $a^{n+m}$; pop — $b^m$; pop — $c^n$



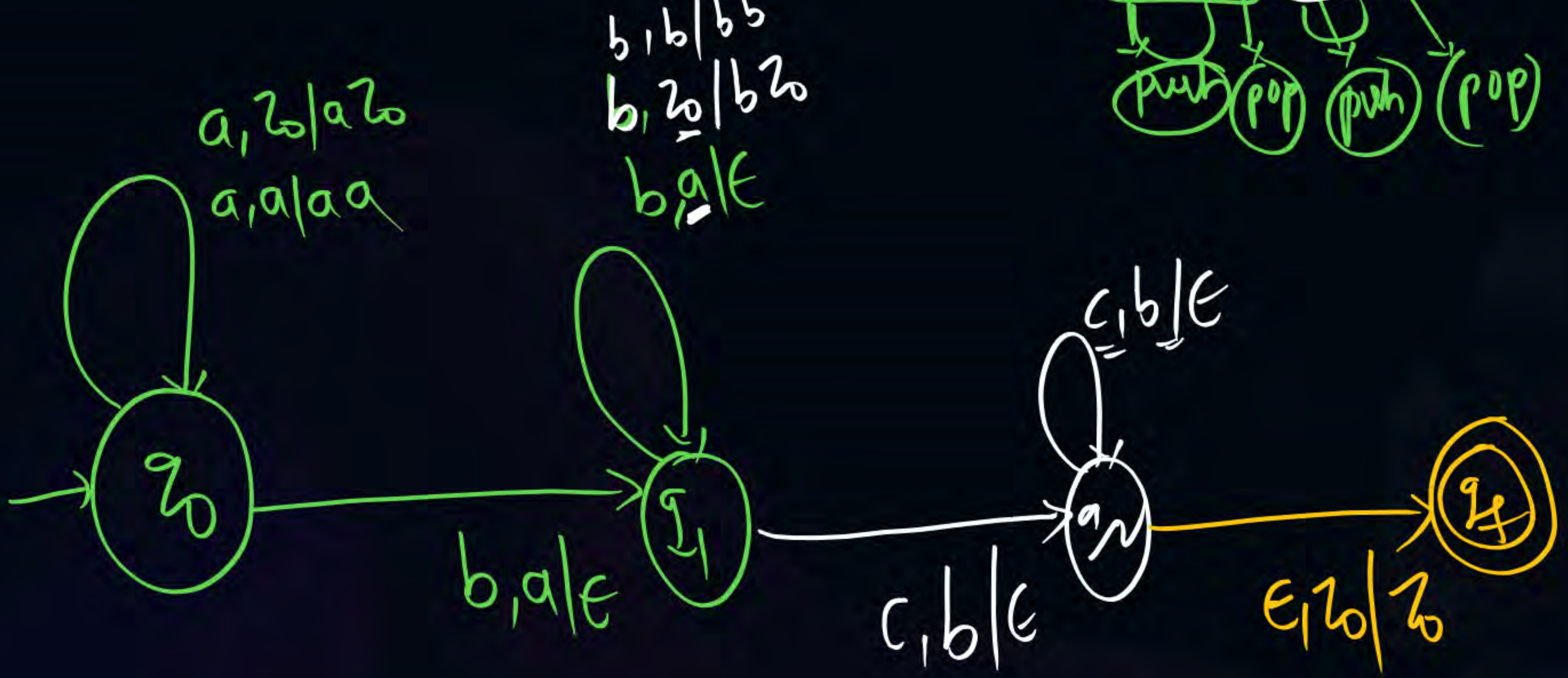
Topic : Pushdown Automata

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

$\boxed{a^n b^n}\ \boxed{b^m c^m}$

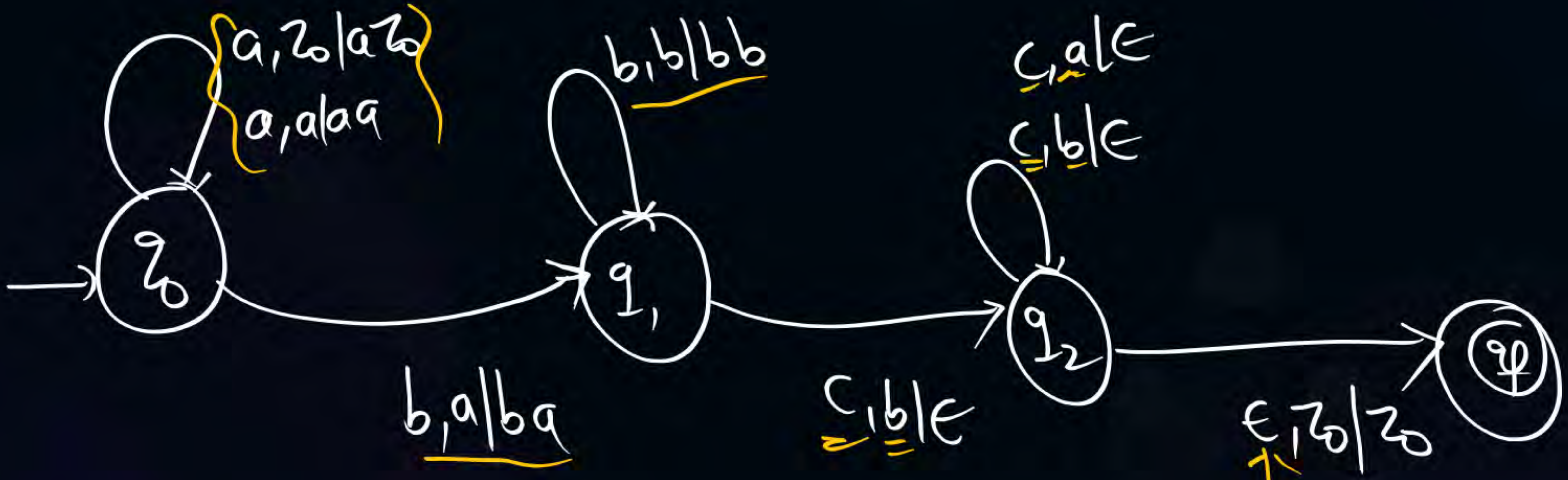
push pop push pop

Logic



## Topic : Pushdown Automata

(Q) Identify languge accepted by given PDA ?




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

# Topic : Pushdown Automata

(Q) Construct PDA for $L = \{ x \mid n_a(x) = n_b(x) \}$, $x \in (a+b)^*$

Empty stack

$\{\epsilon, ab, ba, abab, baba \ldots\}$


$a, z_0 / a z_0$

$b, z_0 / b z_0$

$a, a / aa$

$b, b / bb$

$a, b / \epsilon$

$b, a / \epsilon$

$\epsilon, z_0 / \epsilon$ {empty stack}

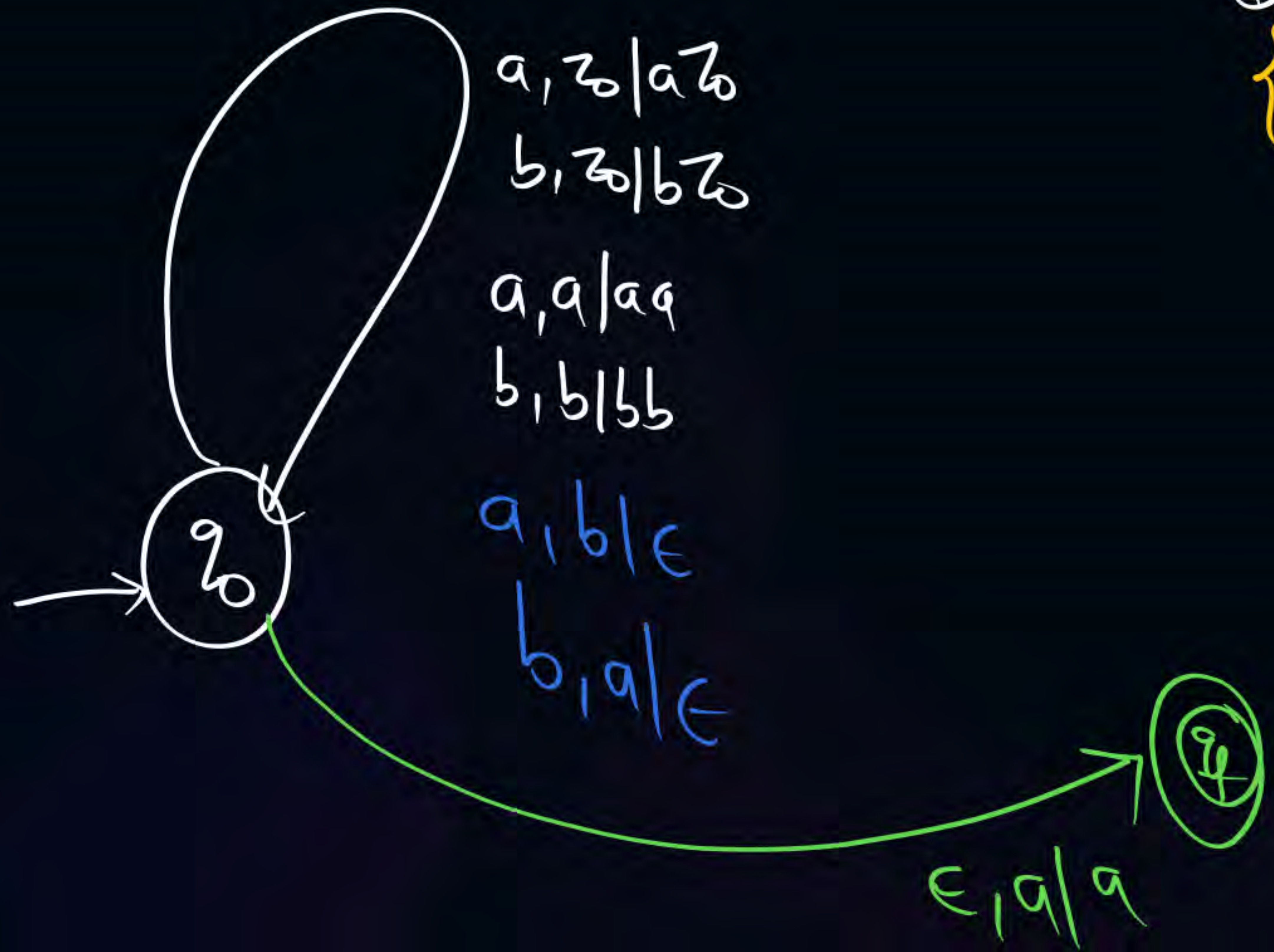
**Logic**

1. Initial $(a, z_0)$, $(b, z_0)$ } push
2. same $(a, a)$, $(b, b)$ } → push
3. diff $(a, b)$, $(b, a)$ } pop

## Topic : Pushdown Automata

(Q) Construct PDA for $L = \{ x \mid n_a(x) \geq n_b(x) \}$ (Empty stack), $x \in (a+b)^*$

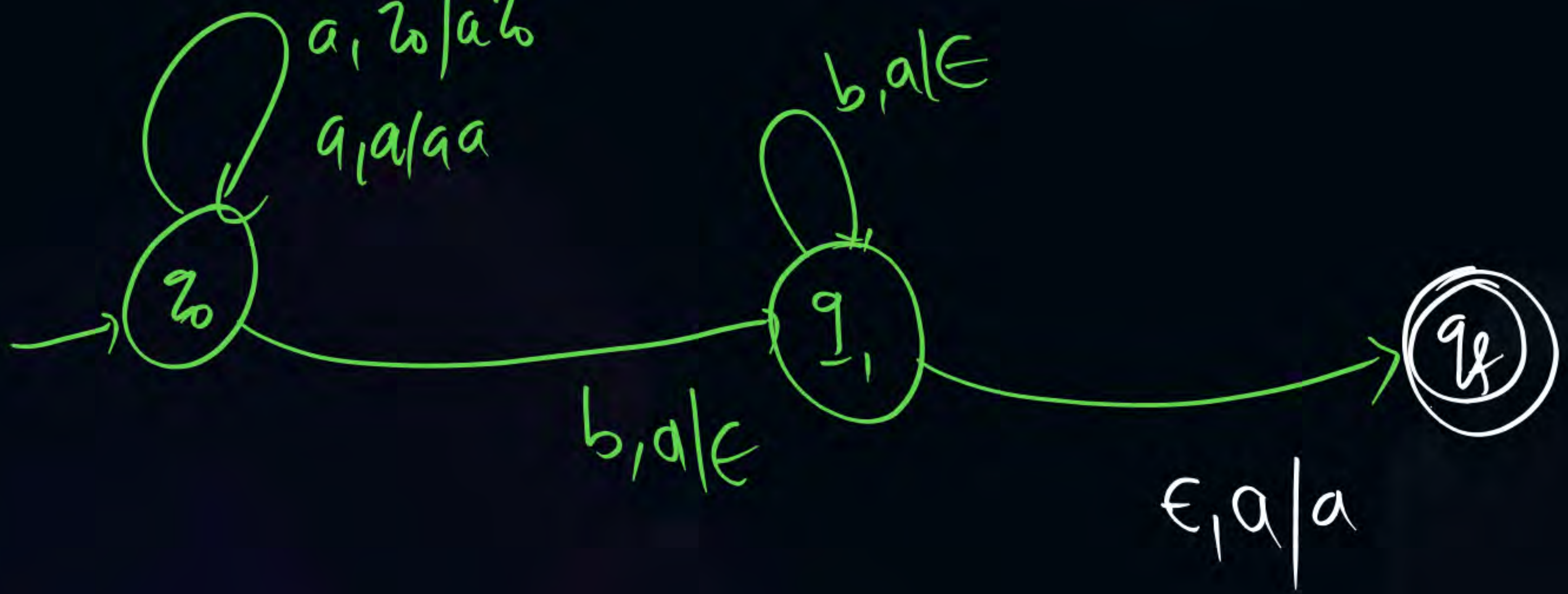
$\{ \epsilon, ab, ba, abab, baba \ldots \}$




**Logic**

1. Initial $(a, z_0)$, $(b, z_0)$ } push
2. Same $(a, a)$, $(b, b)$ } → push
3. diff $(a, b)$, $(b, a)$ } pop

(Q) PDA for $L = \{a^n b^m \mid n > m\}$ $\{n, m \geq 1\}$




$a^n b^n$

(Q) Construct PDA for $L = \{x \mid n_a(x) > n_b(x)\}$ final state

$x \in (a+b)^*$

{Logic}

① $\{n_a(x) = n_b(x)\}$

② $(\epsilon, a) \Rightarrow$ accept

## [MCQ]

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

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-

PDA ⇒ CFL

A. Regular Lang

B. CFL but Not Reg.

C. Finite lang

D. Reg. but not ~~Reg.~~ CFL.

Note:-

Context

1. Lang accepted by push down automata known as CFL

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

PDA > F.A

-

## Topic : Drawback of PDA

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

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

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

THANK - YOU