DATE: / /
PAGE:

Push Down Automata (PDA)

A pushdown automaton is a way to implement a context free grammar in a similar way we design DFA for a regular grammar.

⇒ A DFA can remember a finite amount of information, but

⇒ A PDA can remember an infinite amount of information.

Basically a PDA is –
~~"Finite State machine"~~
"Finite State Machine" + "a ~~stack~~ stack"

⇒ PDA has <u>three</u> components

1. An input tape, (Unread data are stored in i/p tape)
2. A control unit and
3. A stack with infinite size

The <u>stack head</u> scans the top symbol of the stack.

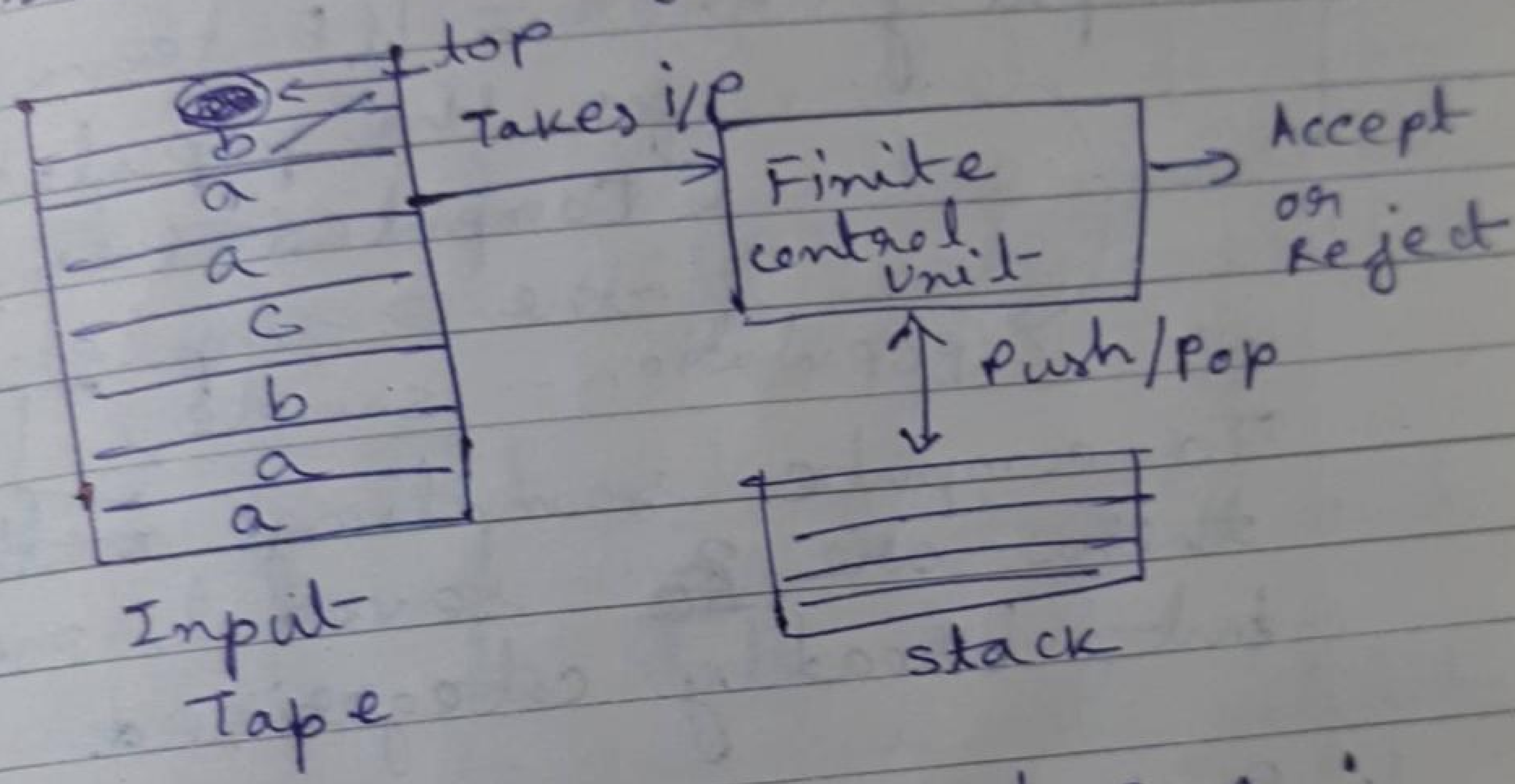
A stack does two operations —

→ Push – A new symbol is added at the <u>top</u>.

DATE: / /
PAGE:

→ Pop — The top symbol is read and removed.

A PDA may or may not read an input symbol, but it has to read the top of the stack in every transaction.



Formal Definition of PDA :

A PDA can be formally described as a 7-tuple $(Q, \Sigma, S, \delta, q_0, I, F)$

Q → Finite no. of states

Σ → Input alphabets

S → stack symbol

δ → Transaction function:
$Q \times (\Sigma \cup \{\epsilon\}) \times S \times Q \times S^*$

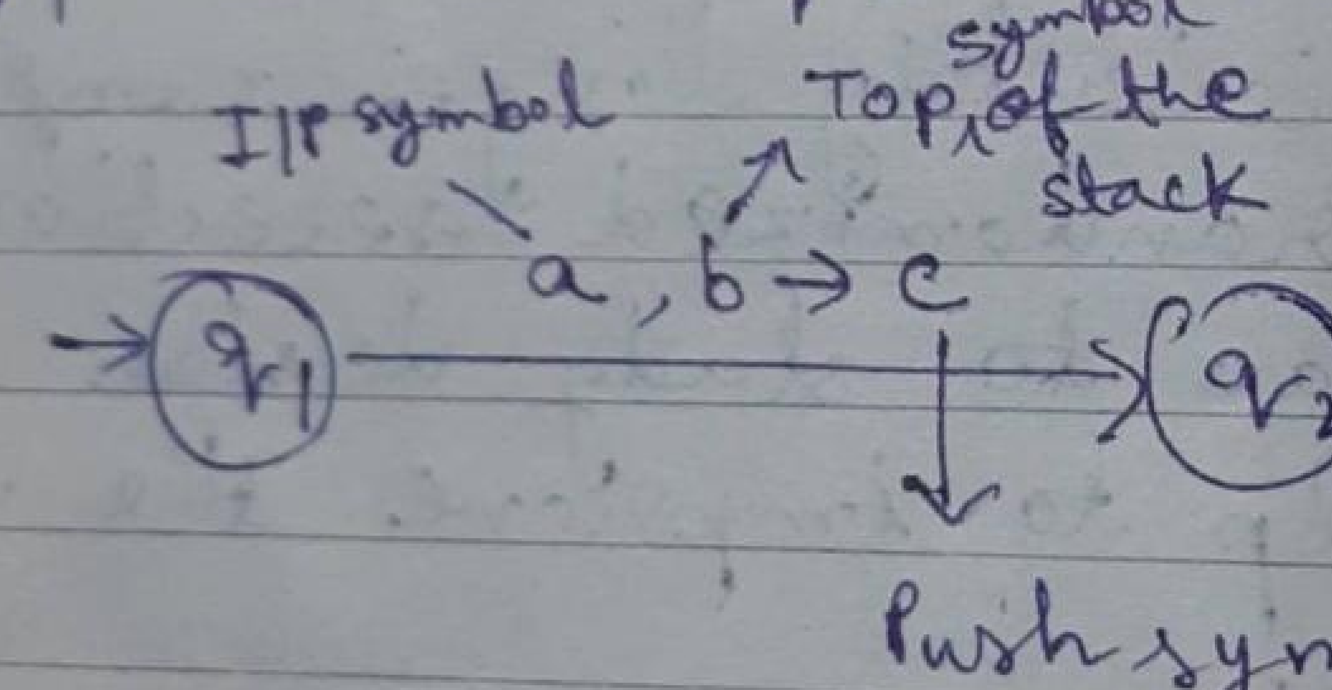
- $q_0 \rightarrow$ Initial state ($q_0 \in Q$)

$I \rightarrow$ Initial stack top symbol ($I \in S$)

:

$F \rightarrow$ set of accepting state ($F \in Q$)

The following diagram shows a transition in a PDA from a state q_1 to state q_2, labeled as $a, b \rightarrow c$



This means at state q_1, if we encounter an i/p string 'a' and top symbol of the stack is 'b' then we pop 'b', push 'c' on top of the stack and moved to q_2.

<u>Terminologies Related to PDA</u>:

1. <u>Instaneous Description</u> (ID)

The instantaneous description

DATE: / /
PAGE:

(ID) of a PDA is represented by ~~to~~ a triple (q, w, s) where
→ q is the state
→ w is unconsumed i/p
→ s is the stack contents

ID is an informal notation of how a PDA computes an i/p string and make a decision that string is accepted or rejected.

2. <u>Turnstile Notation</u>:

It is used for connecting pairs of IDs' that represent one or ~~more~~ many moves of a PDA.
The process of transition is denoted by the turnstile symbol "⊢"

Fallen T

→ ⊢ It represents one move
⊢* " " a sequence of moves.

eg: ~~(q0, b, T) ⊢ (q1, w, s)~~

suppose a → b
whenever an i/p symbol is changed the state has to be changed.

Triple IDs

DATE: / /

PAGE:

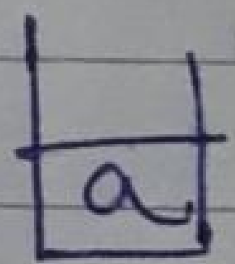
eg. $(p, b, T) \vdash (q, w, a)$

state — Unconsumed i/p — stack contents

We are moving from state p to state q,

~~the inputs~~

While taking a transition from P to q, the i/p symbol 'b' is consumed and top of the stack is 'T' is represented a new string a.



Push Down Automata Acceptance :-

There are 2 different ways to define PDA acceptability

1. Final State Acceptability —

A PDA accepts a string when, after reading the entire string, the PDA in final state.

→ From the starting state, we can make moves that end up in a final state with any stack values.

→ The stack values are irrelevant as long as we end up in a final state.

For a PDA $(Q, \Sigma, \delta, S, q_0, I, F)$ the lang accepted by the set of final states F is —

$$L(PDA) = \{ W \mid \overbrace{(q_0, w, I)}^{ID} \vdash^* (q, \epsilon, x), q \in F \}$$

for any i/p string x.

2. <u>Empty stack Acceptability</u>:

Here a PDA accepts a string when, after reading the entire string, the PDA has emptied its stack.

For a PDA $(Q, \Sigma, \delta, S, q_0, I, F)$ the lang accepted by <u>empty stack</u> <u>is</u>

$$L(PDA) = \{ W \mid (q_0, w, I) \vdash^* (q, \underset{\substack{\text{Empty} \\ \text{i/p}}}{\epsilon}, \underset{\substack{\text{Empty} \\ \text{stack}}}{\epsilon}), q \in Q \}$$

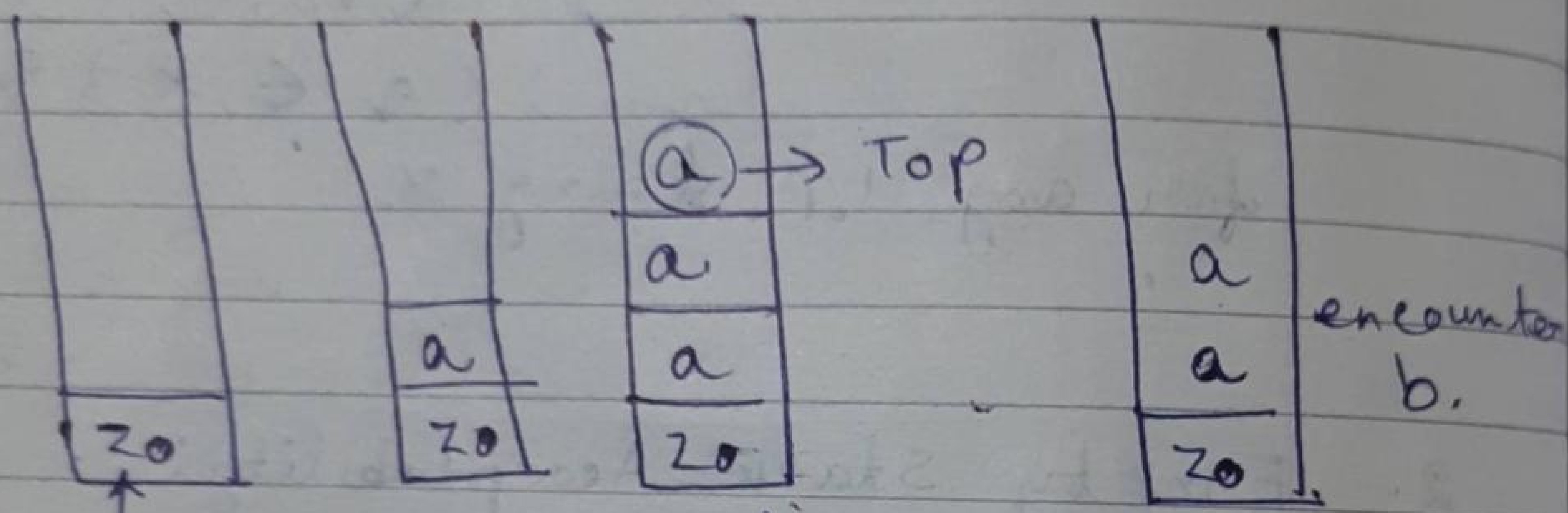
DATE: / /
PAGE:

Examples :

Q.1. Construct PDA that accepts
$L = \{ a^n b^n \mid n \geq 1 \}$

sol^n:- $a^n b^n$ = Every string contains n no. of a's followed n no. of b's.

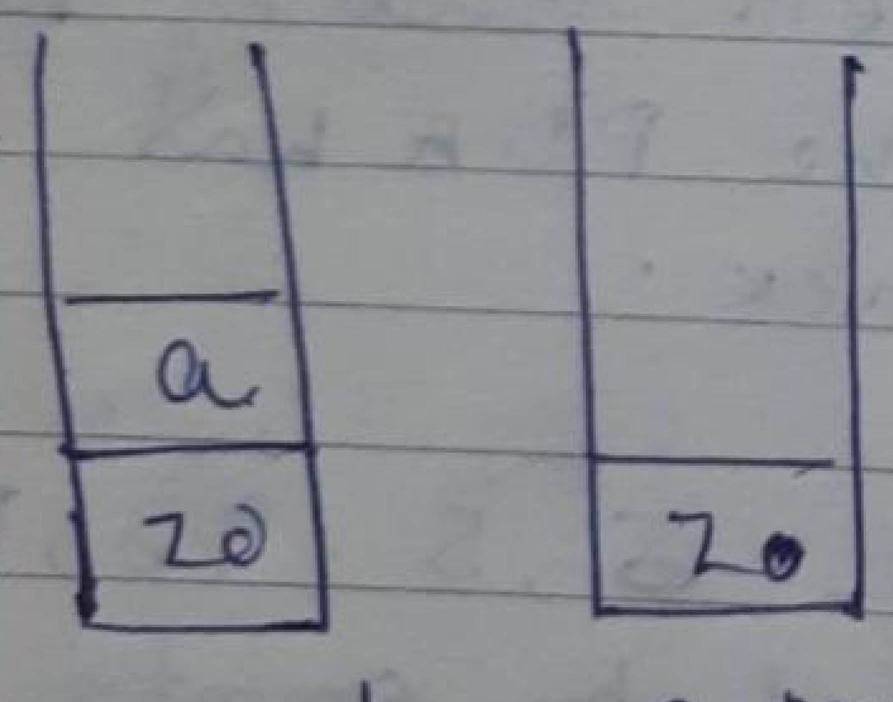
If $n = 3$, $a^3 b^3 \Rightarrow aaabbb$



Z_0 means empty stack

Iterations of a

whenever a b is encountered the top element is popped up. This process is goes on until & unless we are not reaching to Z_0.



encounter 'b'   encounter 'b'

$\therefore$ No. of ~~i/p~~ a's as an i/p is 3 &
" " b' " 3.

$\therefore$ The string is accepted.

Transition diagram

$M = \{Q, \Sigma, S, \delta, q_0, I, F\}$

$Q = \{q_0, q_1, q_2\}$ [no. of states]

$F = q_2$ $\quad \Sigma = \{a, b\}$

** If i/p values are n then no. of states are n+1

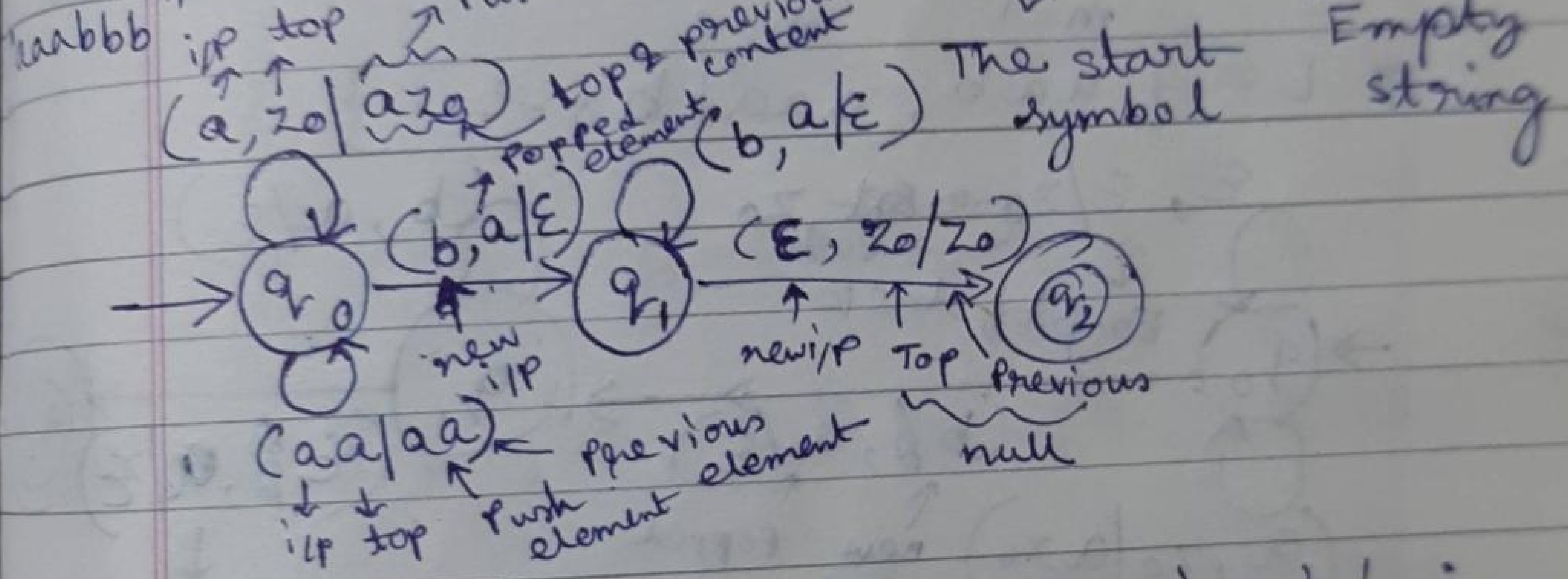
In our example we have n = 2

∴ No of states are n+1 = 3

stack S contain = $\{a, Z_0\}$

a → The start symbol

Z_0 → Empty string



In q_0 whenever i/p 'a' is encountered and top of the stack is 'Z_0' push 'a' on to the top of the stack.

** Whenever a new i/p element is encountered the state must be changed.

Here a new i/p element 'b' has encountered and we reached to state 'q_1' and the top element 'a' should be popped and the pop notation is shown as ε.

Finally we have to check for ε. So new i/p element 'ε' is encountered we have reached to new state 'q'_2'

Q2: Construct a PDA for L = $\{a^n b^m \mid n \geq m, m \geq 1\}$

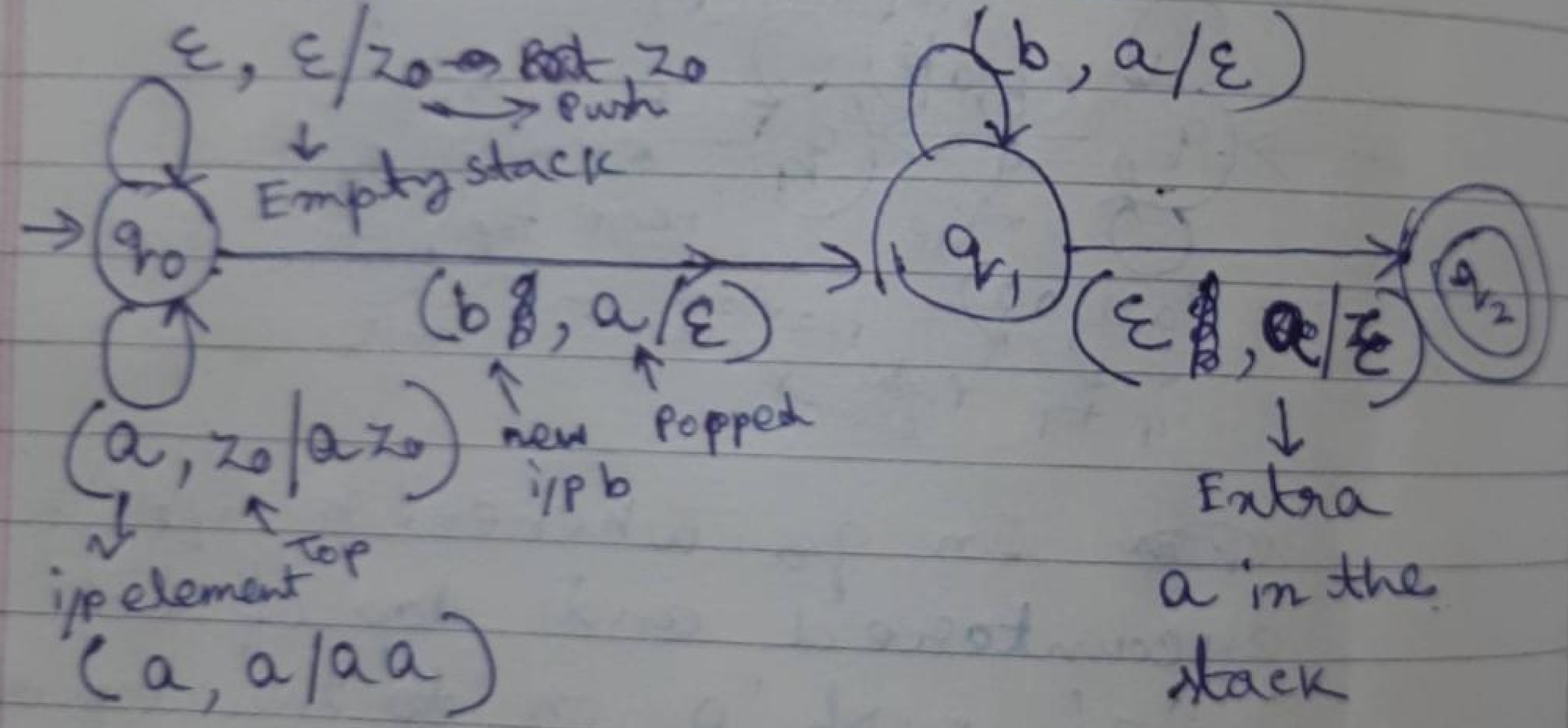
⇓

At least 1 b's should be there

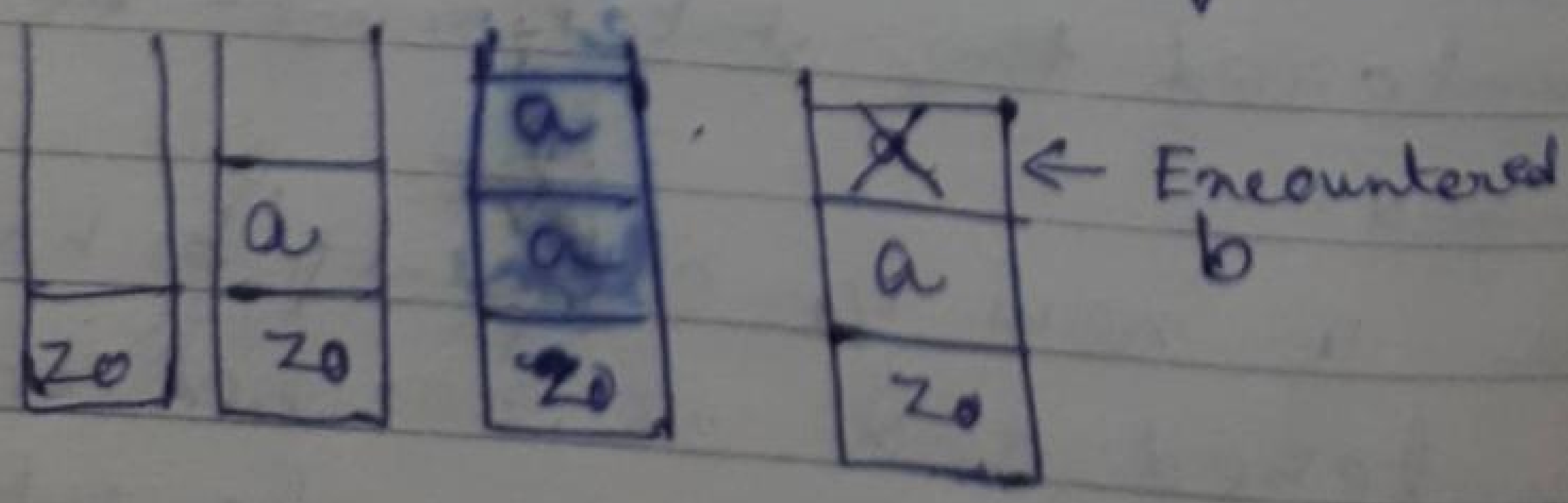
Sol^n:

$m=1$ $m=2$

$L = \{aab, aaabb \ldots\}$



Whenever an extra 'a' is there the string is accepted.



DATE: / /
PAGE:

~~Deterministic PDA & Non PDA~~

~~$\delta : Q \times \{\Sigma \cup \varepsilon\} \times S \to Q \times S^*$~~

Another method to represent PDA

$\Rightarrow$ Transition fn. for Ex. 1

$\delta(q_0, a, z_0) = (q_0, a z_0)$

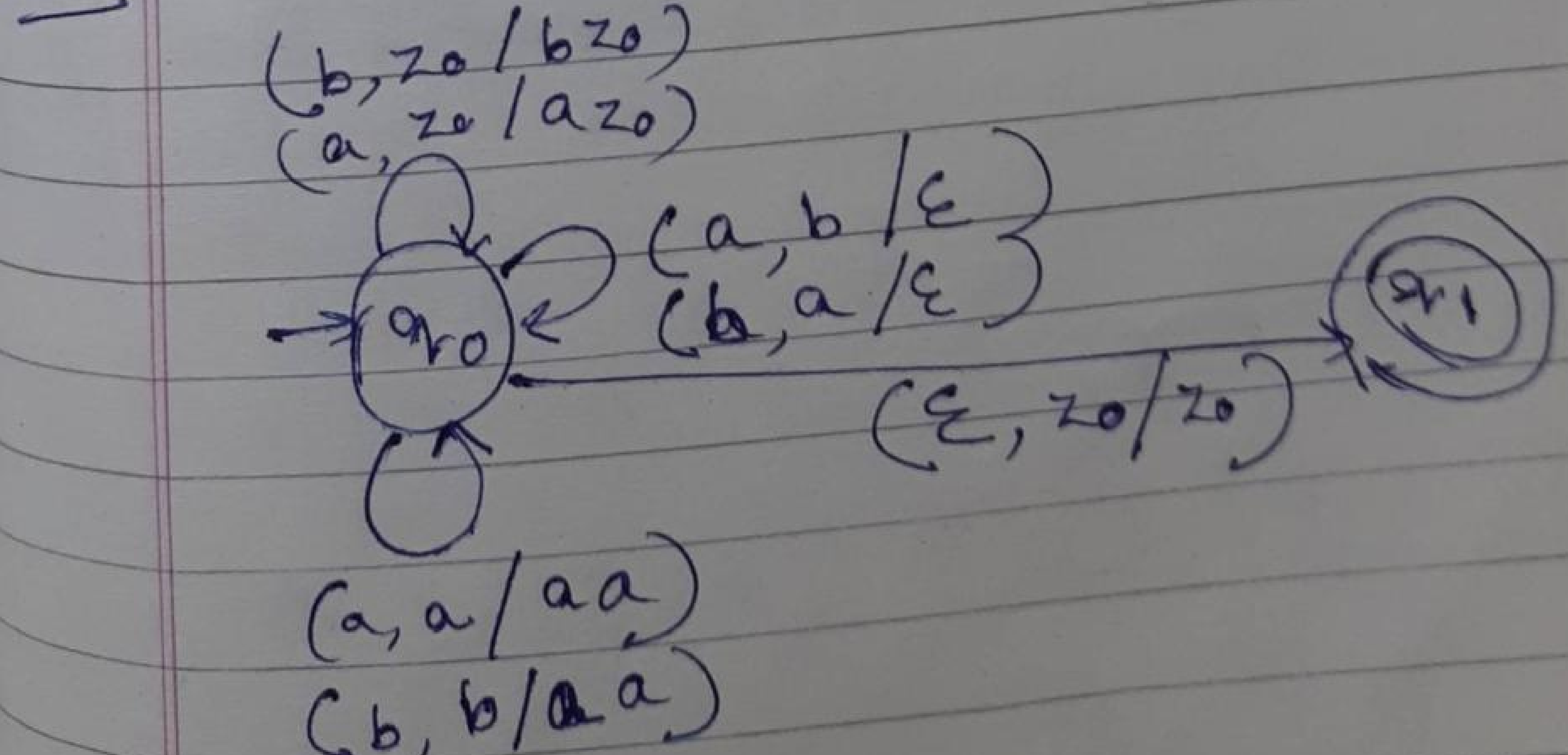
$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \varepsilon)$

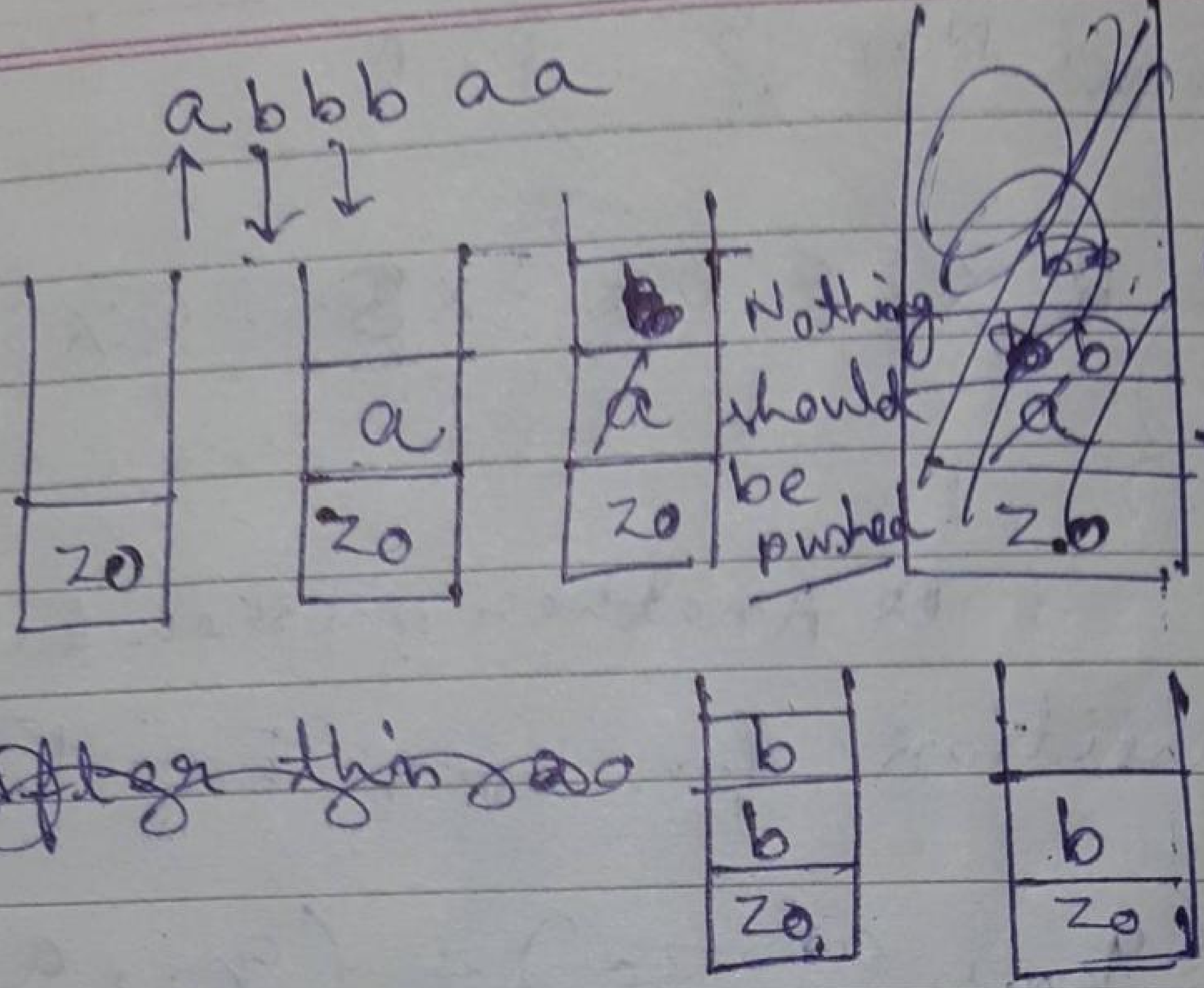
$\delta(q_1, b, a) = (q_1, \varepsilon)$

$\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$

Q3. $w \mid n_a(w) = n_b(w) \mid a^n b^n / n \geq 1$



abbb aa



Nothing should be pushed

~~After this~~ an encountered 'b' stack is empty.

~~After this~~

PDA & CFG

Conversion (vice versa)

DATE: / /
PAGE:

Conversion from CFG to PDA

eg.

$S \rightarrow aSa$
$S \rightarrow bSb$
$S \rightarrow c$

Production rules of CFG

Sol^n:

1st write the production rules & then pop the elements.

ID → Instantaneous Description

state

1. $\delta(q_0, \varepsilon, \varepsilon) = (q_0, \varepsilon)$

Initial; Nothing is there at top of the stack; value on the stack

2. $\delta(q_0, \varepsilon, S) = (q_0, aSa)$

Top of the stack

3. $\delta(q_0, \varepsilon, S) = (q_0, bSb)$

4. $\delta(q_0, \varepsilon, S) = (q_0, c)$

production rules

Popping:

5. $\delta(q_0, a, a) = (q_1, \varepsilon)$

6. $\delta(q_1, b, b) = (q_2, \varepsilon)$

7. $\delta(q_2, c, c) = (q_3, \varepsilon)$

Transition Table

S.No.	state	Unconsumed i/p	stack	Transition no.
1.	q_0	"abbcbba"	ε	1.
2.	q_0	abbcbba	S	1.
3.	q_0	abbcbba	~~a~~sa similar so pop it **	2.
4.	q_1	bbcbba	sa	5.
5.	q_0	~~b~~bcbba	~~b~~sba pop b	3.
6.	q_2	bcbba	sba	6.
7.	q_0	~~b~~cbba	~~b~~sbba pop b	3.
8.	q_2	cbba	sbba	6
9.	q_0	~~c~~bba	~~c~~bba	4.
10.	q_3	~~b~~ba	~~b~~ba	7.

S.No.	state			
11.	q_2	ba	ba	6
12.	q_1	ε	ε	5