PUSH DOWN AUTOMATA

A Pushdown automaton PDA is similar to FSA, except that PDA has an auxillary stack which provides an unlimited amount of memory. A language L is recognized by a pushdown automaton, iff L is content free.

Thus, pushdown automaton accepts a rich classes of languages (which may be regular or non regular) with an unlimited memory capacity, in the form of stack. PDA contributes mainly in the area of parsing and compiler construction.

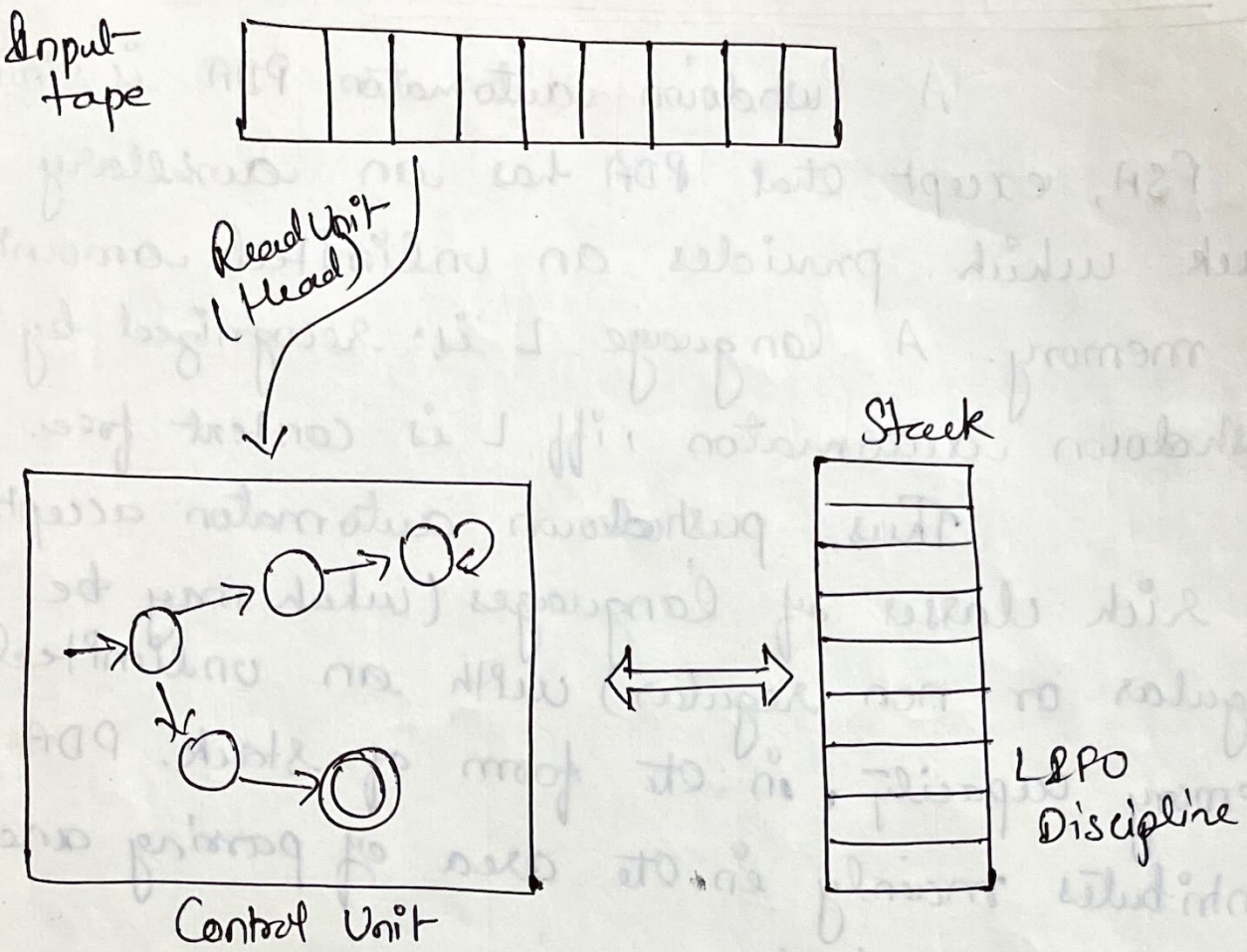
Definition :-

There exist the following equivalent definitions for the concept of PDA :-

(a) a PDA is a way to represent the language class called content-free languages. In other words, PDA are abstract devices that recognize content free languages.

(b) PDA is a generalization of FSA and a PDA changes from state to state, reading input symbols. Unlike FSA, transitions also update the stack either by popping symbols or by pushing them.

Components of PDA



The PDA is usually described as consisting of four components:-

- Input tape
- Read Unit
- Control Unit
- Stack (Memory Unit)

Elements of PDA :- $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

Q = Set of finite states of PDA
Σ = finite set of i/p symbol.
Γ = finite set of stack symbols.
q_0 = $q_0 \in Q$ initial state
z_0 = initial stack symbol
δ = transition function of PDA
F = final state

Description of PDA :-

There are different ways to describe the task of PDA, as follows:-

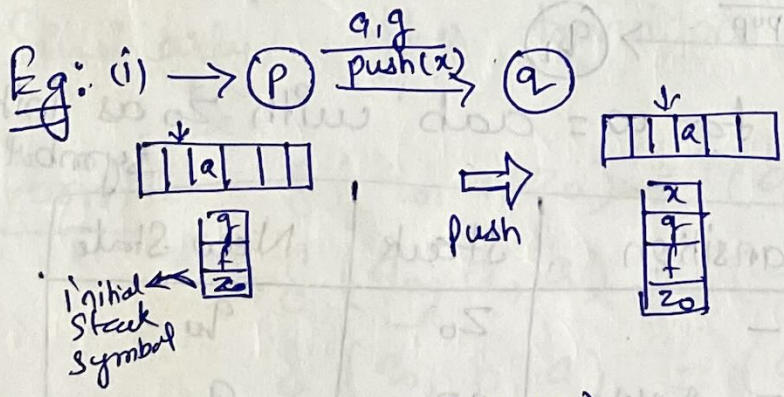
1) Transition Diagram :-

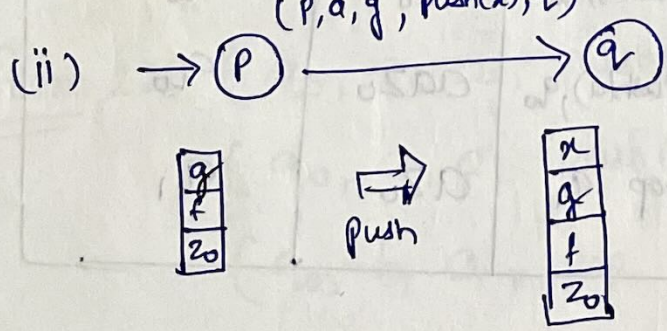
In a directed graph, for an arc going from the vertex which corresponds to state p, to the vertex that corresponds to state q, the edge labelling can be represented in different forms as follows:-

form-1 $\frac{\text{input symbol, top i/p symbol of stack}}{\text{Operations on stack}}$

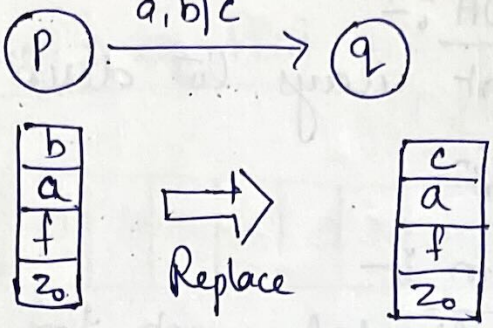
form-2 (p, input symbol, top symbol of stack, operation on stack, q)

form-3 input symbol, pop old stack symbol / Push new stack symbol

Eg: (i) 

(ii) 

(iii)

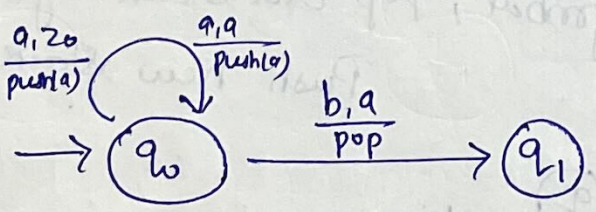


2) Transition Table

The description of operation of a PDA, for a given input string, can be represented in a tabular format called transition table. The table format is shown below:-

Unread i/p	Transition	Stack	New State

E.g.:



Transition table for w = aab with z_0 as initial symbol on stack.

Unread i/p	Transition	Stack	New State
aab	–	z_0	q_0
ab	q_0, a, z_0, push(a), q_0	az_0	q_0
b	q_0, a, a, push(a), q_0	aaz_0	q_0
∈	q_0, b, a, pop, q_1	az_0	q_1

3) Transitions of PDA :-

The transitions of PDA can be represented in different ways, as follows:-

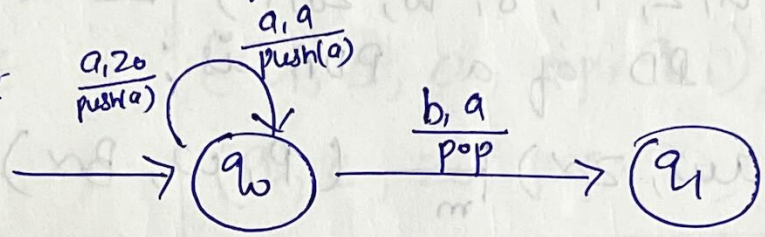
Form-1 :-

δ (current state, current i/p symbol, current stack top)
= (new state, new stack top)

Form-2 :-

(Current State, Current i/p symbol, current-Stack top, Operation on Stack, new State)

E.g:-



Transition Diagram

Form 1 :- $\delta(q_0, a, z_0) = (q_0, a)$

Form 2 :- $(q_0, a, z_0, \text{push}(a), q_0)$

Similarly, other transitions of above dig. can be -

Form 1 :- $\delta(q_0, a, z_0) = (q_0, a)$

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

$\delta(q_0, b, a) = (q_1, \epsilon)$ (ϵ means pop a Current Stack hp)

Form 2 :- $(q_0, a, z_0, \text{push}(a), q_0)$

$(q_0, a, a, \text{push}(a), q_0)$

$(q_0, b, a, \text{pop}, q_1)$

Q
Σ
Γ
z_0
q_0
F
δ

4) Instantaneous Description (ID) of PDA

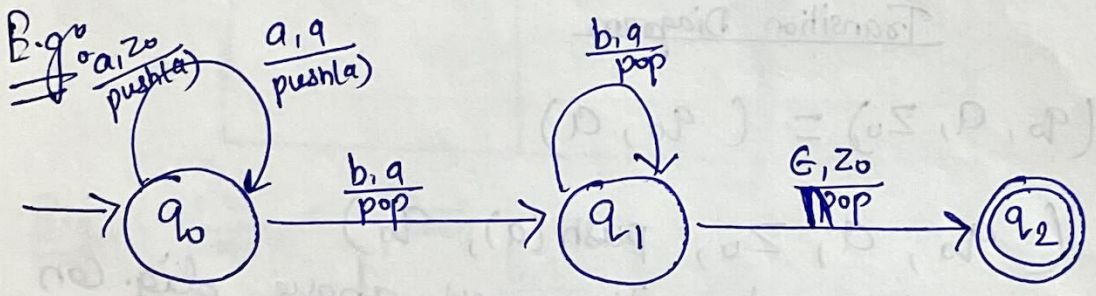
ID of a PDA is defined as triple (q, w, s) where q is the current state, w is the remaining input and s is the current stack contents.

$$(q, \quad w, \quad s)$$

- q ↑ Current State
- w ↑ Remaining Inputs
- s ↑ current Stack Contents

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA, then the ID of a PDA is;

$$(q, aw, z\alpha) \vdash_m (P, w, B\alpha)$$



Check $w = aaabbb$ is accepted by PDA or not.

Soln:

$(q_0, aaabbb, z_0) \vdash (q_0, aabbb, az_0)$
$\vdash (q_0, abbb, aaz_0)$
$\vdash (q_0, bbb, aaaz_0)$
$\vdash (q_1, bb, aaz_0)$
$\vdash (q_1, b, az_0)$
$\vdash (q_1, \epsilon, z_0)$
$\vdash (q_2, z_0)$

q_2 final state reached so accepted

$\therefore (q_0, aaabbb, z_0) \vdash_M^* (q_2, z_0)$

Transition Table

Unread i/p	Transition	Stack	New State
aaabbb	–	Z_0	q_0
aabbb	$(q_0, a, Z_0, push(a), q_0)$	aZ_0	q_0
abbb	$(q_0, a, a, push(a), q_0)$	aaZ_0	q_0
bbb	$(q_0, a, a, push(a), q_0)$	$aaaZ_0$	q_0
bb	(q_0, b, a, pop, q_1)	aaZ_0	q_1
b	(q_1, b, a, pop, q_1)	aZ_0	q_1
$\in$	(q_1, b, a, pop, q_1)	Z_0	q_1
–	$(q_1, \in, Z_0, pop, q_2)$	Z_0	q_2

Design of PDA's

The basic design strategy for PDA is as follows:-

(a) Understand the language properties, for which the PDA has to designed.

(b) Determine the state and alphabet set required.

(c) Identify the initial, accepting and dead states of PDA.

(d) Decide on the stack symbols required.

(e) Determine the initial stack symbol from the Stack symbol set.

(f) For each state, decide on the transition to be made for each character of the input string.

(g) For each state transition, decide on the stack operation to be performed.

(h) Obtain the transition diagram and table for PDA.

(i) Test, the PDA obtained, on short strings.

Q. Design a PDA to accept the language

$$L = \{ w C w^R \mid w \in (0+1)^* \}$$

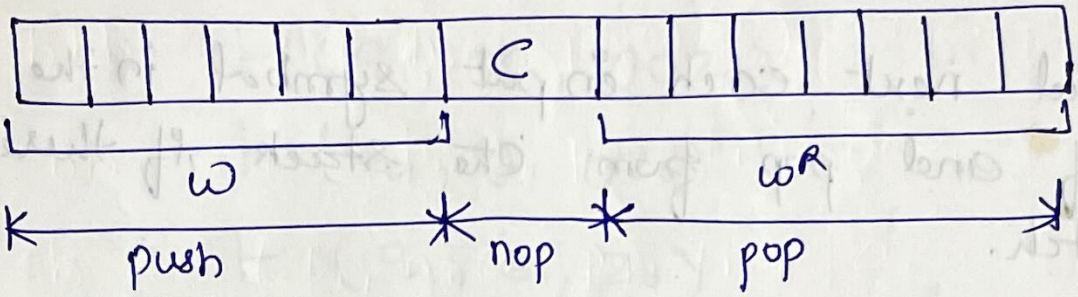
by an empty stack

or

Design a PDA that accepts the language of odd Palindrome.

0 1 0 C 0 1 0

Soln:

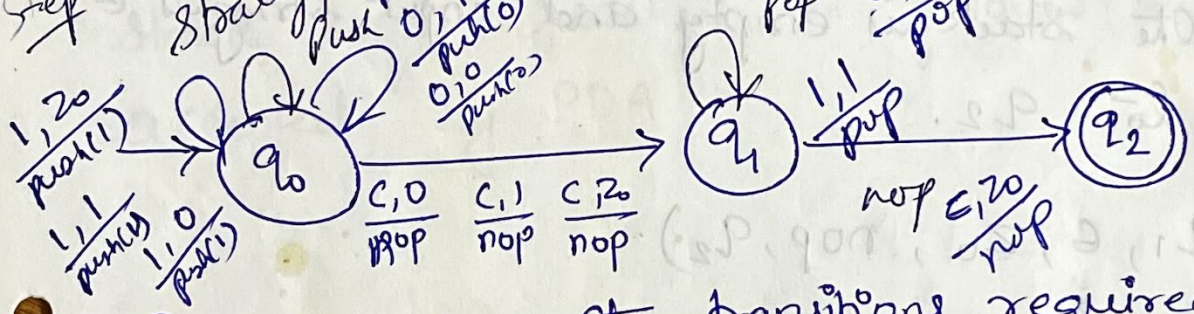


Push all input into stack & remain in q_0

Once C is encountered Shift to q_1.

pop top symbol from the stack if its same as next i/p symbol.

Step 1:- Design Strategy



0 1 C 0 0

Following are the transitions required:-

① Read each symbol of i/p string & push onto the stack till C is encountered.

ⓐ $(q_0, \in, Z_0, \text{pop}, q_0)$

ⓑ $(q_0, 0, Z_0, \text{push}(0), q_0)$

ⓒ $(q_0, 0, 0, \text{push}(0), q_0)$

ⓓ $(q_0, 0, 1, \text{push}(0), q_0)$

ⓔ $(q_0, 1, Z_0, \text{push}(1), q_0)$

ⓕ $(q_0, 1, 0, \text{Push}(1), q_0)$

ⓖ $(q_0, 1, 1, \text{Push}(1), q_0)$

② Once 'c' is encountered shift from q_0 to q_1.

(h) $(q_0, \epsilon, z_0, pop, q_1)$

(i) $(q_0, c, 0, nop, q_1)$

(j) $(q_0, c, 1, nop, q_1)$

③ Read next each input symbol in the second half and pop from the stack, if there is a match.

(k) $(q_1, 0, 0, pop, q_1)$

(l) $(q_1, 1, 1, pop, q_1)$

④ Once the stack is empty and input string is ϵ move to q_2.

(m) $(q_1, \epsilon, z_0, nop, q_2)$

PDA action for the i/p String

Transition Table :- $\omega = 001c100$

Unread i/p	Transition	Stack	New State
001c100	-	z_0	q_0
01c100	$(q_0, 0, z_0, push(0), q_0)$	$0z_0$	q_0
1c100	$(q_0, 0, 0, push(0), q_0)$	$00z_0$	q_0
c100	$(q_0, 1, 0, push(1), q_0)$	$100z_0$	q_0
100	$(q_0, c, 1, nop, q_1)$	$100z_0$	q_1
00	$(q_1, 1, 1, pop, q_1)$	$00z_0$	q_1
0	$(q_1, 0, 0, pop, q_1)$	$0z_0$	q_1
ϵ	$(q_1, 0, 0, pop, q_1)$	z_0	q_1
-	$(q_1, \epsilon, z_0, nop, q_2)$	z_0	q_2

$w = 001c100$ accepted — By PDA

ID:- $(q_0, 001c100, Z_0) \vdash (q_0, 01c100, 0Z_0)$

$\vdash (q_0, 1c100, 00Z_0)$

$\vdash (q_0, c100, 100Z_0)$

$\vdash (q_1, 100, 100Z_0)$

$\vdash (q_1, 00, 00Z_0)$

$\vdash (q_1, 0, 0Z_0)$

$\vdash (q_1, \in, Z_0)$

$\vdash (q_2, Z_0)$

Since, the final state is q_2 and the empty stack Z_0 is reached, it follows that w is accepted by PDA.

Q2: Obtain a PDA to accept the language

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

Soln:-

Design Strategy

Transitions Required

Transition Diagram

PDA action required for the I/P string

- Transition Table
- PDA ID

Design Strategy:- The PDA must operate in the following way. As it reads the i/p symbol a it will keep on adding the elements to the stack and as it is going to encounter first b it will move to state q_1 & pop one 'a' from stack & go to state q_1 and from then for each 'b' it will pop 'a' from the stack. And after the input string when ϵ is reached it will move to state q_2 (final state).

Transitions Required:-

1. Read each symbol of the input string i.e 'a' and push onto the stack, until 'b' is encountered.
   - (a) $(q_0, a, Z_0, \text{push}(a), q_0)$
   - (b) $(q_0, a, a, \text{push}(a), q_0)$

(2) Once b is encountered move to state q_1. by performing pop operation.

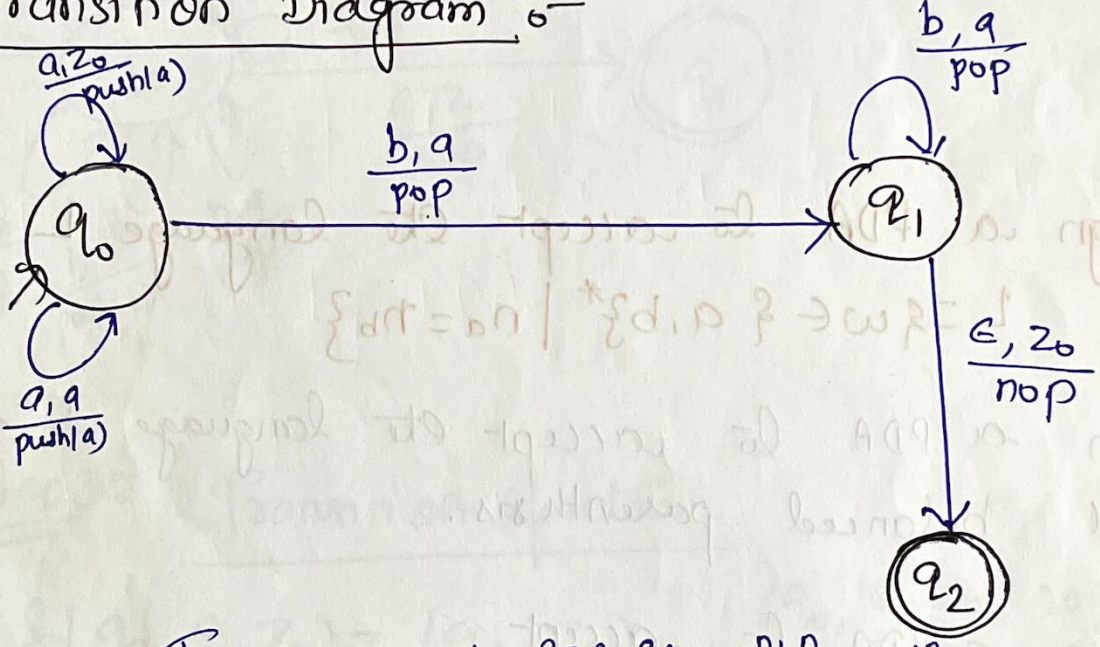
(c) (q_0, b, a, pop, q_1)

(d) (q_1, b, a, pop, q_1)

(3) Once the stack is empty or input is ϵ, change to q_2.

(e) $(q_1, \epsilon, z_0, nop, q_2)$

* Transition Diagram :-



Transition Dig for PDA $\{L = a^n b^n : n \geq 1\}$

* PDA action for the input string :-

$w = aaabbb$

PDA ID. $(q_0, aaabbb, z_0) \vdash (q_0, aabbb, az_0)$
$\vdash (q_0, abbb, aaz_0)$
$\vdash (q_0, bbb, aaaz_0)$
$\vdash (q_1, bb, aaz_0)$
$\vdash (q_1, b, az_0) \vdash (q_1, \epsilon, z_0)$
$\vdash (q_2, z_0)$

Q:: Design a PDA to accept the language L

$$L = \{ w \in \{a,b\}^* \mid n_a = n_b \}$$

Q:: Design a PDA to accept the language L of nested balanced parenthesis.

Q:: Design a PDA to accept

$$L = \{ w \mid w \in (a,b)^* \}, \text{ such that}$$

(i) $n_a(w) < n_b(w)$ and

(ii) $n_b(w) < n_a(w)$.

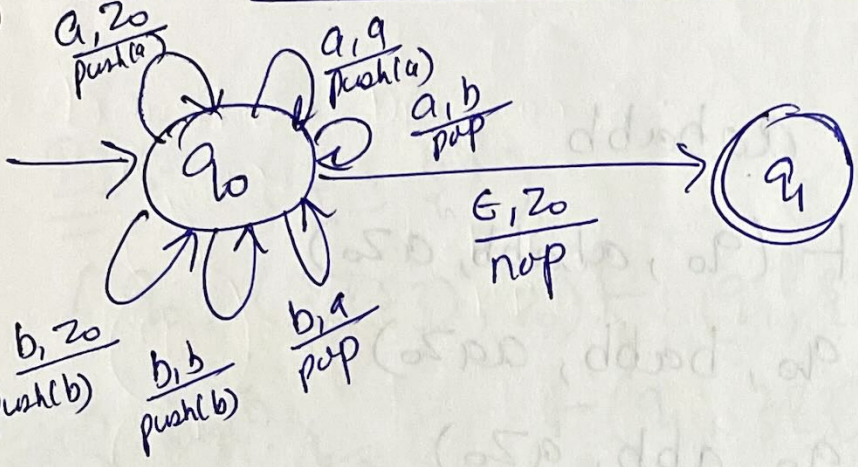
$$L = \{ a a^n b^m a^n \; ; \; m, n \geq 1 \}$$

$n = 1$
$n = 2$
$n = 3$
$n = 4$

Q1. Design a PDA for $L = \{w \in (a,b)^* \mid n_a = n_b\}$

Soln:- Step1:- Design Strategy:- As we will read a from the input tape we will apply push operations and add 'a' to the Stack. For each b encountered in the Stack we will pop a from the stack.

Step 2:- Transition Graph:-



Step 3:- Transition function

$\delta(q_0, \epsilon, Z_0) = (q_1, Z_0)$ $(q_0, \epsilon, Z_0, nop, q_1)$

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ $(q_0, a, Z_0, push(a), q_0)$

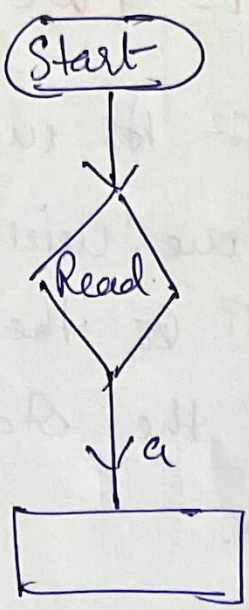
$\delta(q_0, a, a) = (q_0, aa)$ $(q_0, a, a, push(a), q_0)$

$\delta(q_0, b, Z_0) = (q_0, bZ_0)$ $(q_0, b, Z_0, push(b), q_0)$

$\delta(q_0, b, b) = (q_0, bb)$ $(q_0, b, b, push(b), q_0)$

$\delta(q_0, a, b) = (q_0, \epsilon)$ (q_0, a, b, pop, q_0)

$\delta(q_0, b, a) = (q_0, \epsilon)$ (q_0, b, a, pop, q_0)



PD for the String aababb

$\delta(q_0, aababb, z_0) \vdash (q_0, ababb, az_0)$

$\vdash (q_0, babb, aaz_0)$

$\vdash (q_0, abb, az_0)$

$\vdash (q_0, bb, aaz_0)$

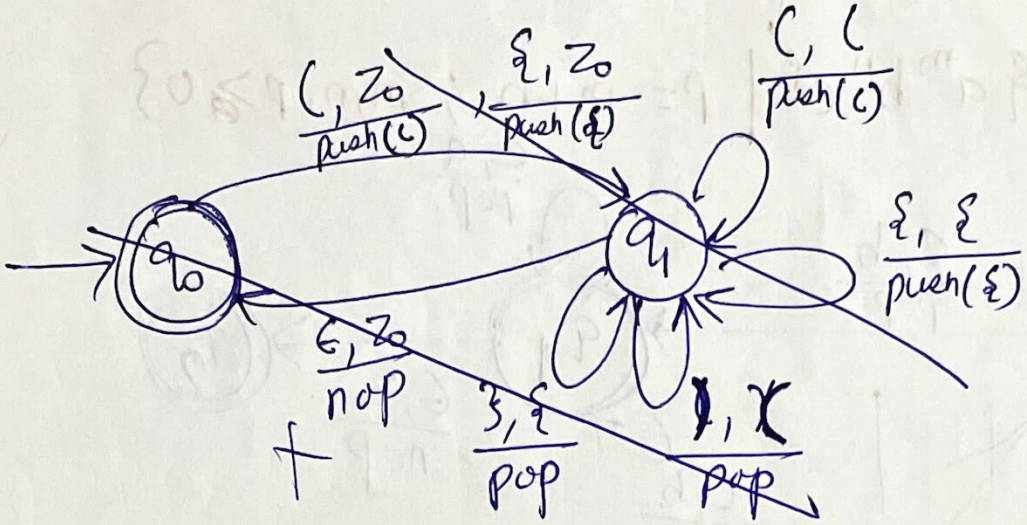
$\vdash (q_0, b, az_0)$

$\vdash (q_0, \epsilon, z_0)$

$\vdash (q_1, z_0)$

↳ Accept State

Q:- Design a PDA to accept a string of balanced paranthesis. $\Sigma = [(,), \{, \}]$



ID:- ({}{})

$(q_0, (\{\}\{\}), Z_0) \vdash (q_1, \{\}\{\}), (Z_0)$

$\vdash (q_1, \}\{\}), \{(Z_0)$

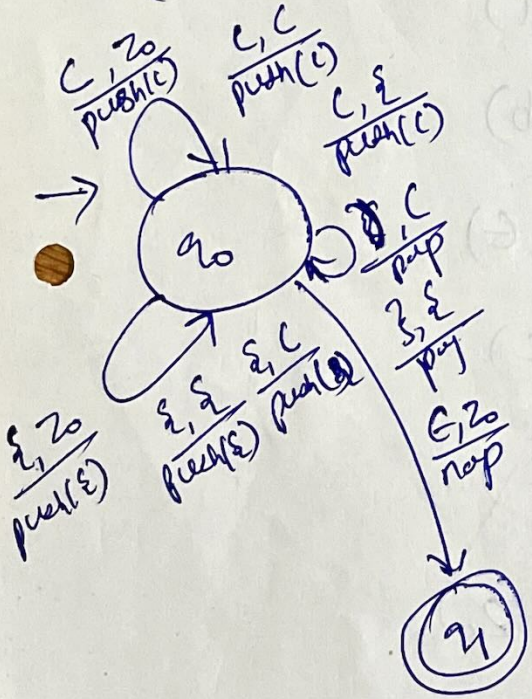
$\vdash (q_1, \{\}), (Z_0)$

$\vdash (q_1, \}), \{(Z_0)$

$\vdash (q_1,), (Z_0)$

$\vdash (q_1, \epsilon, Z_0)$

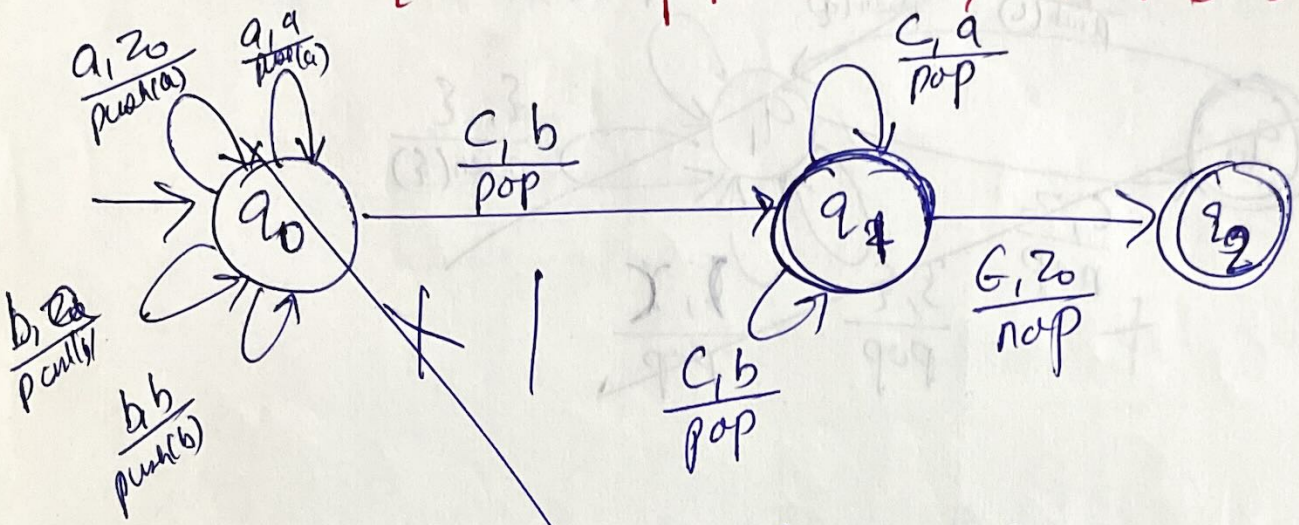
$\vdash (q_0, Z_0)$ – Accept State



({ ( ) })

Q:- Construct PDA for

$$L = \{ a^m b^n c^p \mid p = m+n \; ; \; m, n, p \geq 0 \}$$



$(q_0, a, Z_0) = (q_0, aZ_0)$

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

$(q_0, b, a) = (q_0, ba)$

$(q_0, b, b) = (q_0, bb)$

$(q_0, c, b) = (q_1, G)$

$(q_1, c, b) = (q_1, G)$

$(q_1, c, a) = (q_1, G)$

$(q_1, G, Z_0) = (q_2, Z_0)$

↳ Accept State