Note:- 1) Every transition of PDA is associated with one of the three operation ie. PUSH, POP, No-operation (Skip)

2) If atleast one transition represents more than one operation then the PDA is nondeterministic (NPDA)

3) PDA can accept the RL iff every transition represents No-operation.

4) PDA accept the NRL using stack & empty stack mechanism. iff atleast one transition represents PUSH operation

Push Down Automata

DPDA

(Deterministic PDA)

1) The PDA has atmost one choice of move in any state is called as DPDA

2) Every CFL can't be recognized by DPDA

3) DPDA is less powerful than NPDA

4) A palindrome can't be accepted by DPDA (ww^R)

5) DPDA's are very useful in programming languages
ex:- Parser's are deterministic

6) Every DPDA is NPDA

7) $\delta : Q \times \{\Sigma \cup \epsilon\} \times \Gamma \Rightarrow Q \times \Gamma^*$

If you are going to ~~pus~~ one state then ~~pushing~~ it is DPDA

NPDA

(Nondeterministic PDA)

1) In NPDA, there could be multiple moves under a situation

2) Every CFL can be recognized by NPDA.

3) NPDA is more powerful than DPDA

4) A palindrome can be accepted by NPDA (ww^R

5) Syntax of most of the programming language is described by DCFL's

6) Not every NPDA has a DPDA.

7) $\delta : Q \times \{\Sigma \cup \epsilon\} \times \Gamma \Rightarrow 2^{(Q \times \Gamma^*)}$

If you are going to more than one state then print more than one symbol it is NPDA
(more operations)

Ex:- Construct a PDA that accept $L=\{a^n b^n \mid n \geq 1\}$ through empty stack:

1> **Using a set of equations:-**

1> $\delta(\overset{\text{Current State}}{q_0}, \overset{I/P}{a}, \overset{\text{Topmost}}{Z_0}) = (q_0, \overset{\text{New Topmost}}{aZ_0})$ {First 'a' of $a^n b^n$ is pushed onto the stack}

2> $\delta(q_0, a, a) = (q_0, aa)$ {Subsequent of a's of $a^n b^n$ are pushed onto the stack one by one}

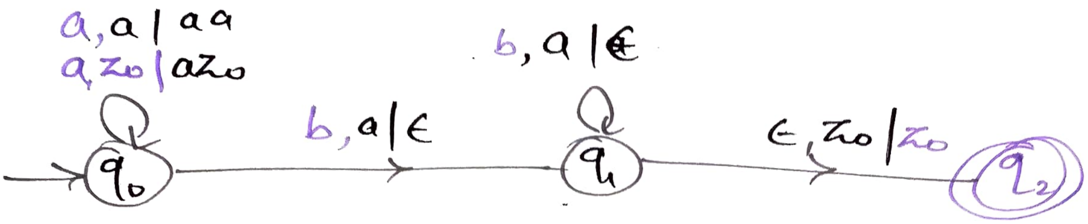
3> $\delta(q_0, b, a) = (q_1, \epsilon)$ {on seeing first b, the machine will make a move to q_1 state with pop operation}

4> $\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$ {After the end of input string machine will transit to a final state}

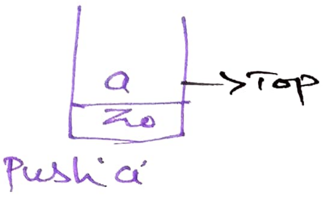
(OR)

$\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$ => By empty stack.

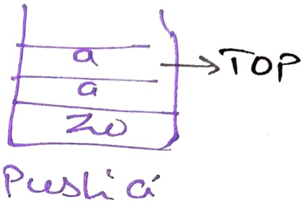
② By Using Graphical representation (transition diagram)



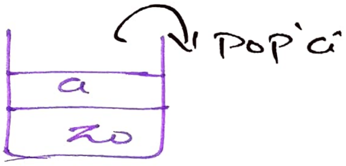
1



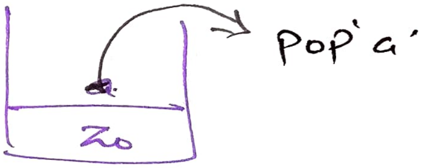
2



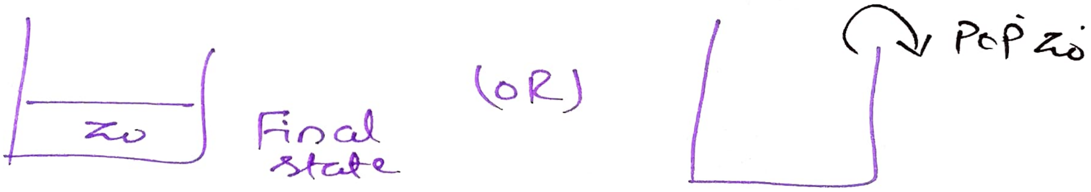
3



4



5



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

* Language of PDA :- [Acceptance ~~of~~ by PDA]

⇒ The language L can be accepted by a PDA in two ways –

① Acceptance by final state

② Acceptance by Empty stack.

1) Acceptance by Final State :-

After reading the complete I/P string, if PDA reaches the final state. then I/P string is accepted by PDA & this mechanism is known as acceptance by final state

$$L(M) = \{ w \mid (q_0, w, z_0) \xRightarrow[M]{*} (q, \epsilon, \alpha) \}$$

where, $q \in F. \alpha$

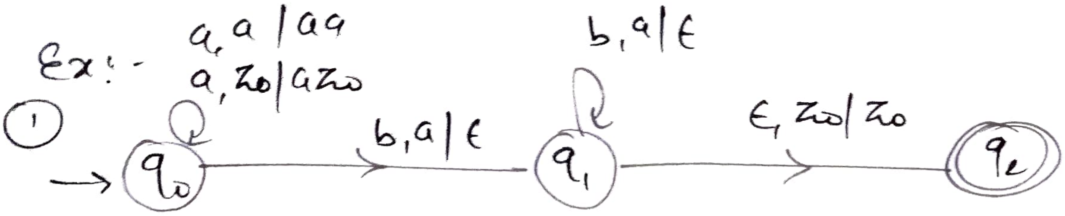
② Acceptance by Empty stack :-

After reading the complete i/p string if the stack is empty then the PDA accept the i/p string then this mechanism is known as acceptance by empty stack.

$$L(M) = \{ w \mid (q_0, w, z_0) \xRightarrow[M]{*} (q, \epsilon, \epsilon) \}$$

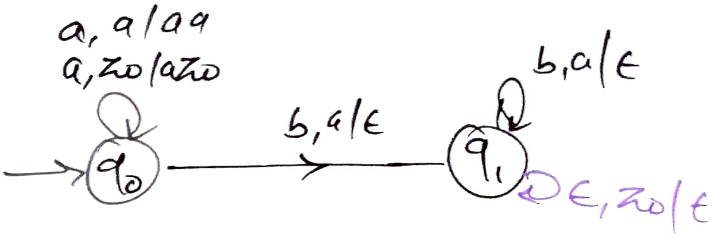
Ex:-

①



← By final State

②



⇐ By empty stack