Unit-3

PUSHDOWN AUTOMATA. (PDA)

PDA Definitions - Moves - Instantaneous Descriptions - DPDA - Equivalence PDA & CFL - Pumping lemma for CFL - Problems based on PL.

PDA: Introduction:-

- As far as we are concerned with finite Automaton (Linear) the Computational Capability is very less. Limited memory.
- The Computational result is gnl. by FA is either 'yes' (accepted) or 'No' (Not accepted).

⇒ To overcome this Pbm Backus & Naur designed a new automaton called PDA.

* Why PDA is Needed ?

⇒ To overcome the Limitations of FA, PDA has Unlimited amount of memory, accessible in the form of STACK. (LIFO) Recursive.

⇒ The CFG is defined by the special type of automata namely PDA.

⇒ PDA is an extension of the NFA-∈ with the addition of a STACK.

↳ Used to read the Symbols, Push and Pop only at the top of the Stack.

- PDA can remember an Infinite amt of String Infm.
- PDA can recognize only Context free Languages.
- PDA has more powerful than FA which has finite memory.

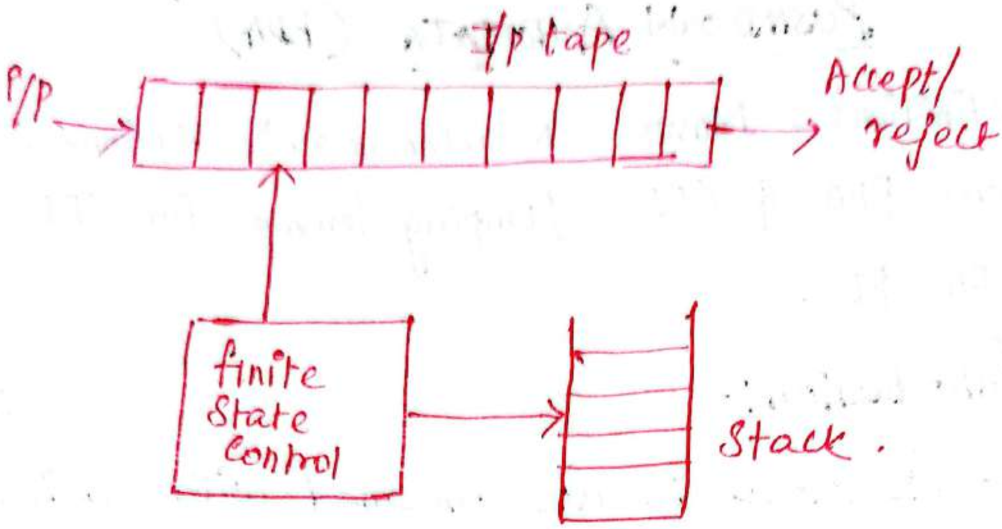


Fig) PDA.

- The PDA consists of the following:
  - Control unit (finite State control)
  - Input tape
  - Stack (memory unit).
  - Reading head

I/p tape: is divided into many cells.
- At each cell only one i/p symbol is placed.
- reads the i/p one symbol at a time.

Finite ctrl: has some pointer which points the current symbol which is to read.

End of i/p: $ or Δ or φ. (Blank symbol)

Stack: Last in first out.

- Push
- Pop.

→ In PDA, finite control consists of a set of transitions, i/p state & set of final states.

→ The State Transition takes place not only on the current i/p & current state, but also based on current symbol which is available on the top of the stack.

→ The PDA can also make a spontaneous transition using ϵ as its i/p symbol.

⊛ The Activities down by the PDA is as follows:-

i) Read the i/p symbol from i/p tape. if ϵ is the i/p, then no input is consumed.

ii) It makes the transition with the current state, i/p symbol, symbol at the top of stack. So after transition, the control may enter into either a new state or previous state.

iii) Replaces the symbol at the top of the stack by any other symbol.

There are 3 ways to recognize a PDA :-

1) Recognition by empty stack.
2) Recognition by reaching the final state.
3) Recognition by empty stack and reaching final states.

Formal Definitions of PDA :-

It has 7 components.

$$\boxed{\text{In PDA } P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)}$$

Where

$Q \rightarrow$ finite set of states.

$\Sigma \rightarrow$ finite set of symbols.

$\Gamma \rightarrow$ A finite stack alphabet (This is the set of symbols that are pushed to the stack)

$\delta \rightarrow$ Transition function.

$q_0 \rightarrow$ Initial state.

$Z_0 \rightarrow$ Starting symbol of stack.

$F \rightarrow$ Set of accepting states of final states.

δ :-

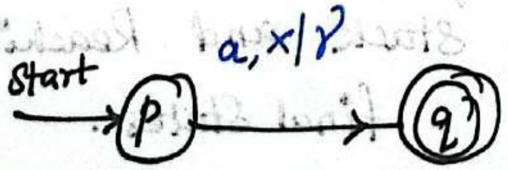
$$\boxed{\delta(q, a, x) = (P, \gamma)}.$$

Graphical Notations of PDA :-

i) The Nodes corresponds to the States of PDA.

ii) An arrow Labeled 'Start' Indicates the Starting symbol. & double circle States $\Rightarrow$ accepting States.

eg) $\delta(P, a, x) = (q, \gamma)$



P – current state

a – i/p

x – old top stack symbol.

q $\rightarrow$ new state

$\gamma \rightarrow$ New top stack symbol.

$\rightarrow$ An arc Indicates the transition of PDA each are labeled as $a, x|\gamma$ from state p to q means as $\delta(P, a, x)$ goes to (q, γ).

Moves :- Moves :-

Three Important Principles about ID :-

1) if the sequence of ID's are legal for PDA P, then the computation formed by adding the same additional i/p string to the end of the i/p in each ID is also legal.

ex:1 if $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a PDA and $(q, x, a) \vdash (P, y, \beta)$ then for any string w in Σ^* and γ in Γ^*, it is also true that,

$$(q, xw, \alpha\gamma) \vdash_P^* (P, yw, \beta\gamma)$$

If $\gamma = \epsilon$ then,

$$(q, xw, \alpha) \vdash_P^* (P, yw, \beta)$$

And the same i/p string to the end of the i/p.

if $w = \epsilon$ then

$$(q, xw, \alpha\gamma) \vdash_P^* (P, yw, \beta\gamma)$$ Add the same stack symbols below the stack.

2) if the computational is legal for PDA P, then the computation formed by adding the same additional stack symbols below the stack in each ID's are also legal.

ex:1

3) if a computation is legal for PDA P and some tail of the i/p is not consumed, then we can remove this tail from the i/p in each ID and the resulting computation is also legal.

Ex:2 if $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a PDA and $(q, xw, \alpha) \vdash (P, yw, \beta)$ then for any string w in Σ^* and γ in Γ^*, it is also true that,

$$(q, x, \alpha) \vdash_P^* (P, y, \beta).$$

Remove the unconsumed input string.

Moves :- Moves :-

The Interpretation of

$\delta(q, a, Z) = \{(P_1, \gamma_1), (P_2 \gamma_2) \ldots (P_m \gamma_m)\}$

where q and $P_i \rightarrow$ States.

a in $\Sigma \rightarrow$ i/p

$Z \rightarrow$ Stack symbol.

γ_i is in Γ^*.

Turnstile Notation

eg) $\delta(q, w, Y) \vdash (P, \epsilon, A)$

(i) Single move $\vdash$ (ii) Sequence of moves or multiple moves $\vdash^*$.

The Interpretation of

$\delta(q, \epsilon, Z) = \{(P_1 \gamma_1)(P_2 \gamma_2) \ldots (P_m \gamma_m)\}$.

Instanteneous Descriptions : (ID) :

we define an ID to be a triple (q, w, γ) where

$q \rightarrow$ state ; $w \rightarrow$ string ; $\gamma \rightarrow$ Stack symbol.

if $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a PDA. We say

$(q, aw, Z\alpha) \vdash_m (P, w, \beta\alpha)$ if

$\delta(q, a, Z)$ contains (P, β).

The Language of a PDA :-

→ A PDA can accept its i/p by consuming it and entering into an accepting state.

→ There are 2 ways of language acceptances.

(i). Acceptance by Final State.

(ii) Acceptance by Empty Stack.

i). Acceptance by Final State :-

*. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA, then the Lang. accepted by a final state is,

$$L(M) = \{ W \mid (q_0, W, Z_0) \vdash^* (P, \epsilon, \gamma) \text{ for some } P \text{ in } F \text{ and } \gamma \text{ in } \Gamma^* . \}$$

⇒ When the PDA consumes all the i/p symbols, and goes to accepting state with initial stack symbol is called Lang. accepted by final state.

i). Acceptance by Empty Stack / null stack :

*. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA, then the Lang. accepted by a empty stack is,

$$N(M) = \{ W \mid (q_0, W, Z_0) \vdash^* (P, \epsilon, \epsilon) \text{ for some } p \text{ in } Q. \}$$

— All the i/p has been consumed & makes the Stack empty is called Lang. accepted by empty Stack.

Equivalence of Acceptance.

I. Converting a Language accepted by Empty Stack to final State:-

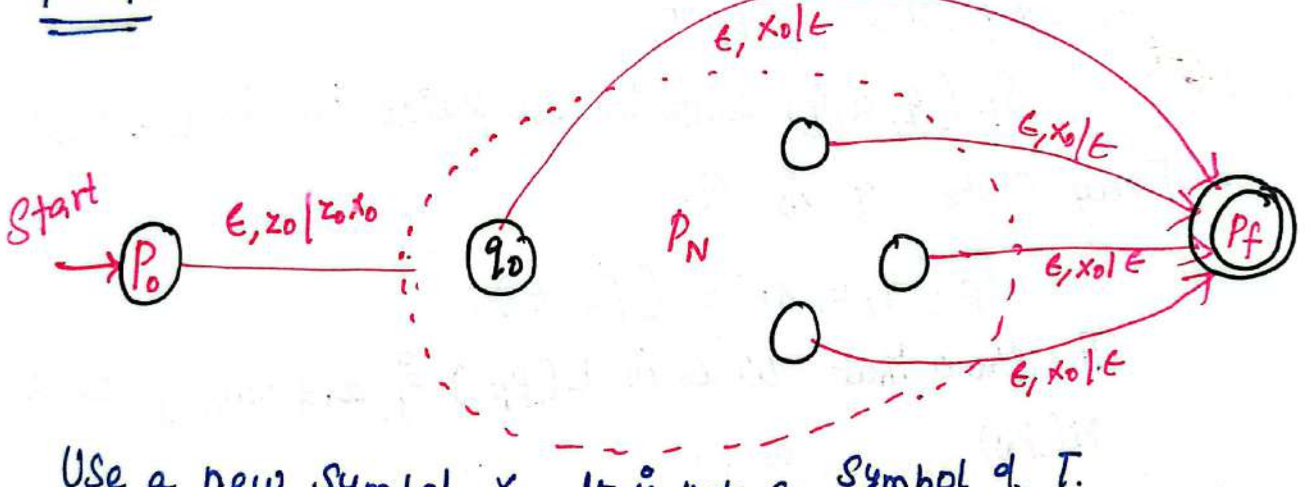
Theorem:-

Consider the Class of Languages, $L(P)$ is Language acceptance by final state and $N(P)$ is Language acceptance by empty stack.

→ first take PDA P_N that accepts a Lang L by empty stack and take PDA P_F that accepts L by final state.

If $L = N(P_N)$ for some PDA $P_N = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ then there is a PDA P_F such that $L = L(P_F)$.

Proof:



Use a new symbol X_0, it is not a symbol of Γ.

X_0 is Start Symbol of P_F and also a marker for the bottom of the Stack. Only then we can understand P_N has reached an empty stack.

ie) If P_F sees X_0 on the top of its stack, then P_N would empty its stack on the same i/p. Create a new start state P_0, it is used to push Z_0, (Start symbol of P_N), onto the top of the stack and enters into state q_0, (start state of P_N).

P_F simulates P_N, until the stack of P_N is empty, which P_F detects because it sees X_0 on the top of the stack. Finally, we include the new state P_f, which is the accepting state of P_F.

Let $P_F = (Q \cup \{p_0, p_f\}, \Sigma, \Gamma \{X_0\}, \delta_F, p_0, X_0, \{p_f\})$ where trans fn δ_F is defined as follows,

- The stack state P_F makes a simultaneous transition to the start state of P_N by pushing Z_0 onto the stack.

$$\delta_F(p_0, \epsilon, X_0) = \{q_0, Z_0 X_0\}.$$

- For all states q in Q, input a in Σ or $a = \epsilon$ and stack symbols Y in Γ.

$$\delta_F(q, a, y) \text{ contains all pairs in } \delta_N(q, a, y)$$

- Every state q in Q.

$$\delta_F(q, \epsilon, X_0) = (P_f, \epsilon).$$

To show that w is in $L(P_F)$ if and only if w is in $N(P_N)$.

<u>If-part:</u>

Given that $(q_0, w, Z_0) \vdash^*_{P_N} (q, \epsilon, \epsilon)$ for some state q.

$$(q_0, w, Z_0 X_0) \vdash^*_{P_N} (q, \epsilon, X_0).$$

P_F has all the moves of P_N,

$$(q_0, w, Z_0 X_0) \vdash^*_{P_F} (q, \epsilon, X_0).$$

$\Rightarrow$ Consider the sequence of moves together with the initial and final moves.

$(p_0, w, X_0) \vdash_{P_F} (q_0, w, Z_0 X_0) \vdash^*_{P_F} (q_0, \epsilon, X_0) \vdash_{P_F} (P_f, \epsilon, \epsilon)$.

$\therefore$ P_F accepts w by final state.

Only-if part:-

$(q_0, w, Z_0) \vdash^*_{P_N} (q, \epsilon, \epsilon)$

Thus, w is in $N(P_N)$.

II Converting a Language accepted by final state to empty stack:-

Let PDA P_F accepts a Lang L by final state and construct another PDA P_N that accepts L by empty stack.

Let L be $L(P_F)$ for Some PDA $P_F = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$. Then there is a PDA P_N such that $L = N(P_N)$.

$$L = L(P_F)$$
$$L = N(P_N).$$

Proof:-

P_F simulates by P_N.

$\Rightarrow$ for each accepting state of P_F, we can add the new ϵ transition to the new state P. When we are in state P, P_N pops its stack and doesnot consume any i/p. When P_F enters an accepting state after consuming input w, P_N will empty its stack after consuming w.

P_N must start in a new state P_0, to push the start symbol of P_F on the stack and go to the start state of P_F.

Let $P_N = (Q \cup \{p_0, p\}, \Sigma, \Gamma\{X_0\}, \delta_F, P_0, X_0)$ where, transition function δ_N is defined as,

- Pushing the Start Symbol of P_F onto the Stack and going to the Start state of P_F.

$$\delta_F(P_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\}.$$

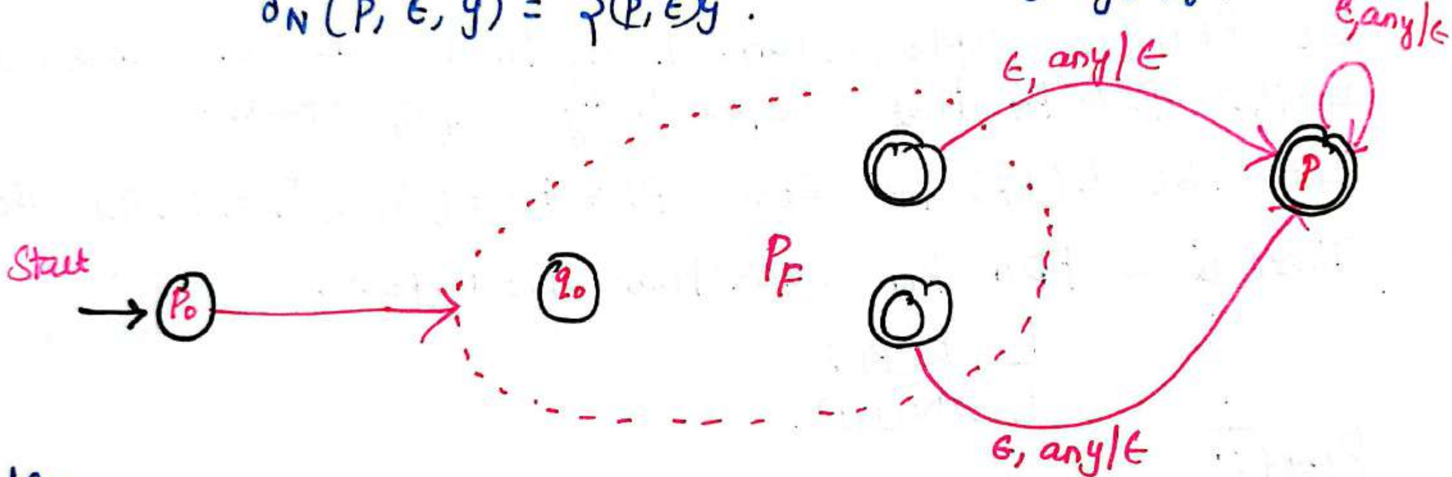
- P_N Simulates P_F (ie)

$\delta_N(q, a, y)$ contains every pair in $\delta_F(q, a, y)$ for all states q in Q and input symbols a in Σ or $a = \epsilon$ and y in Γ.

- When P_F accepts, P_N can empty its stack without consuming any more i/p:

$\delta_N(q, \epsilon, y)$ contains (p, ϵ). Where, q is in the final state F, y in Γ or $y = X_0$.

$$\delta_N(p, \epsilon, y) = \{(p, \epsilon)\}.$$



<u>If-part:</u>

Every transition of P_F is a move of P_N.

$$(q_0, w, Z_0) \vdash^*_{P_F} (q, \epsilon, \alpha).$$

Where, $q \to$ accepting state, $\alpha \to$ stack String.

We know that,

$(q_0, w, Z_0X_0) \vdash^*_{P_N} (q, \epsilon, \alpha X_0)$ then P_N can do the Same.

$$(P_0, w, X_0) \vdash_{P_N} (q_0, w, Z_0X_0) \vdash^*_{P_N} (q, \epsilon, \alpha X_0) \vdash^*_{P_N} (p, \epsilon, \epsilon).$$

<u>Only-if part:</u>

- $(P_0, w, X_0) \vdash_{P_N} (q_0, w, Z_0X_0) \vdash^*_{P_N} (q, \epsilon, \alpha X_0) \vdash^*_{P_N} (p, \epsilon, \epsilon)$

Where q is an accepting State of P_F.

- $(q_0, w, X_0) \vdash^*_{P_F} (q, \epsilon, \alpha)$ without X_0 on the stack P_F accepts w by the final State.

Thus, w is in $L(P_F)$.

Problems - Construction of PDA.

① Design a PDA that accepts $wcw^R \mid w$ in $(0+1)^*$ by empty stack.

Solu:

Idea:-

→ For each Move, the PDA writes a symbol on the top of the Stack.

→ q_0 is the Start state, On reading the Symbols 0's and 1's, the PDA remains in the Same States (push).

→ On reading the input Symbol c, the PDA goes to the State q_1. From q_1, the PDA goes to the State q_2 till reads the ϵ-Symbol. Compare the Stack Symbol with the i/p Symbol, if it matches, pop the Stack symbol.

→ Repeat the Process till reaches the Final State or empty Stack.

$$\therefore \boxed{M = P = (\underbrace{\{q_0, q_1, q_2\}}_{Q}, \underbrace{\{0, 1\}}_{\Sigma}, \underbrace{\{0, 1, Z_0\}}_{\Gamma}, \delta, q_0, Z_0, \{q_2\})}$$

δ:

(i)

$\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$

$\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$

$\delta(q_0, 0, 0) = (q_0, 00)$

$\delta(q_0, 1, 0) = (q_0, 10)$

$\delta(q_0, 0, 1) = (q_0, 01)$

$\delta(q_0, 1, 1) = (q_0, 11)$

PUSH (W.)

(q, q, Z)
0 0
1 1
Z_0

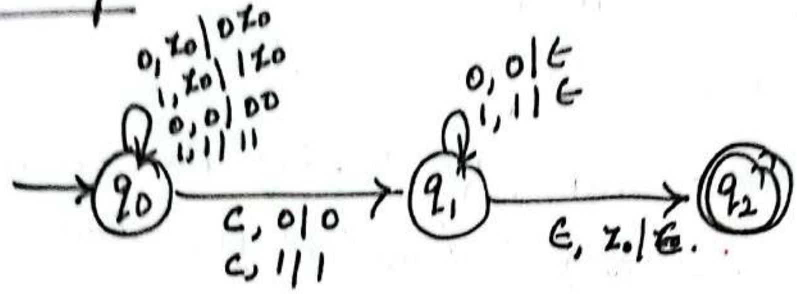
~~1~~
0
Z_0

(ii) $\delta(q_0, c, 1) = \{q_1, 1)$
$\delta(q_0, c, 0) = \{q_1, 0)$. | Accept the seperator.

iii) $\delta(q_1, 0, 0) = (q_1, \epsilon)$
$\delta(q_1, 1, 1) = (q_1, \epsilon)$ | Pop.

iv) $\delta(q_1, \epsilon, z_0) = (q_f, \epsilon)$ (q_2, ϵ)

Trans. diagram:



Eg):

i) $\delta(q_0, 100c001, z_0)$
$\vdash (q_0, 00c001, 1z_0)$
$\vdash (q_0, 0c001, 01z_0)$
$\vdash (q_0, c001, 001z_0)$
$\vdash (q_1, \cancel{0}01, \cancel{0}01z_0)$
$\vdash (q_1, \cancel{0}1, \cancel{0}1z_0)$
$\vdash (q_1, 1, 1z_0)$
$\vdash (q_1, \epsilon, z_0)$
$\vdash (q_2, \epsilon)$.

Accepted.

ii) $\delta(q_0, 110c10, z_0)$
$\vdash (q_0, 10c10, 1z_0)$
$\vdash (q_0, 0c10, 11z_0)$
$\vdash (q_0, c10, 011z_0)$
$\vdash (q_1, 10, 011z_0)$

Rejected.

2) Design a PDA that accept the language $L = \{ww^R \mid w = \{0,1\}^*\}$

Solu:

- When we are in state q_0, and read the i/p symbol w and push it onto the stack.
- Whenever we read 'ϵ' → from state q_0 to q_1 changed. (Leaving intact whatever symbol is at the top of the stack).
- Then Read w^R → match the i/p symbols and pop it.

$$\boxed{P = (\{q_0, q_1, q_2\}, \{0,1\}, \{0,1,Z_0\}, \delta, q_0, Z_0, \{q_2\}).}$$

δ:

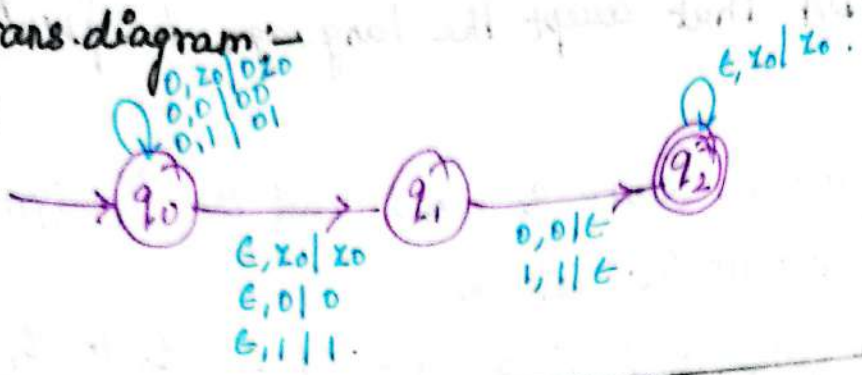
1. $\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$
   $\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$
   $\delta(q_0, 0, 0) = (q_0, 00)$
   $\delta(q_0, 1, 0) = (q_0, 10)$
   $\delta(q_0, 0, 1) = (q_0, 01)$
   $\delta(q_0, 1, 1) = (q_0, 11)$ — PUSH W.

2. $\delta(q_0, \epsilon, Z_0) = (q_1, Z_0)$
   $\delta(q_0, \epsilon, 0) = (q_1, 0)$
   $\delta(q_0, \epsilon, 1) = (q_1, 1)$ — Sponteneous trans on 'ϵ'.

3. $\delta(q_1, 0, 0) = (q_1, \epsilon)$
   $\delta(q_1, 1, 1) = (q_1, \epsilon)$ — POP W^R

4. $\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$.

Trans. diagram :-



3) Design a PDA to accept the Lang $L = \{0^n 1^n / n \geq 1\}$ accepting by final State.

<u>Solu</u>: <u>Step 1</u>:

$L = \{0^n 1^n / n \geq 1.$

here we have to construct a PDA such that the String contains equal no. of zero's & one's.

→ We read the i/p symbols $0^n \Rightarrow$ PUSH.

→ $1^n \Rightarrow$ POP.

<u>Step 2</u>:

$\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$

$\delta(q_0, 0, 0) = (q_0, 00)$ push

~~$\delta(q_0, 0, 1) = (q_0, 01)$~~

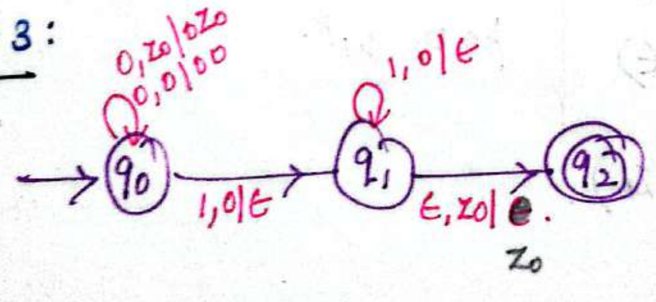
$\delta(q_0, 1, 0) = (q_1, \epsilon)$; $\delta(q_1, 1, 0) = (q_1, \epsilon)$

$\delta(q_0, 1, 1) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) = (q_2, \cancel{\epsilon}) \Rightarrow (q_2, Z_0)$

So, PDA $\boxed{P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, \cancel{1}, Z_0\}, \delta, q_0, Z_0, \{q_2\})}$

<u>Step 3</u>:



Step 4: Ex: 0011

$\delta(q_0, 0011, Z_0) \vdash (q_0, 011, 0Z_0)$
$\vdash (q_0, 11, 00Z_0)$
$\vdash (q_1, 1, 0Z_0)$
$\vdash (q_1, \epsilon, Z_0)$
$\vdash (\underset{f}{q_2}, Z_0)$ // hence the string is accepted.

Ex: 2
w = 011.

$\delta(q_0, 011, Z_0) \vdash (q_0, 11, 0Z_0)$
$\vdash (q_1, 1, Z_0)$
There is no transition. So it is not accepted.

---

4) $L = \{a^n b^n / n \geq 0\}$ accepting by final state.

Solu:

Step 1: Here the string contains equal no. of a's and equal no. of b's.

⇒ q_0, is the Initial state, we read 'a's initially PUSH all the a's to the stack.

⇒ When we read 'b' we try to delete from the stack.

⇒ For each 'a' in the stack, we delete the 'a' from the stack for reading each 'b'.

- we delete all the symbols & we reach the final state.

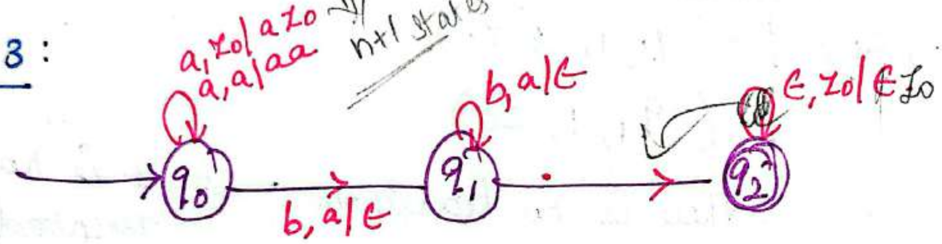
Step 2: δ :

$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$
$$\delta(q_0, a, a) = (q_0, aa), \cancel{(q_0, a, b)} \vdash (q_0, ab)$$
$$\delta(q_0, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_2, \cancel{\epsilon} Z_0)$$

$$\therefore \boxed{P = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, \cancel{b}, Z_0\}, \delta, q_0, Z_0, \{q_2\})}$$

n+1 states

Step 3:



Step 4:

Ex: $w = aabb$.

2) $w = aab$.
↳ rejected.

$\delta(q_0, aabb, Z_0)$

$\vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_1, b, aZ_0)$

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

$\vdash (q_2, \epsilon)$ accepted.

5) Design a PDA with the Set of strings with twice as many b's than a's with a as the Starting String. (or)

Design PDA for the Set of all strings {a,b} with two occurences of b's for each a's. (or)

Design PDA for the Lang $L = \{a^n b^{2n} / n \geq 0\}$.

$L = \{0^n 1^{2n}$.

Solu:

• Here the idea to design this PDA is what when we read single 'a' we insert or PUSH two a's on the Stack. Then we read 'b', we pop each 'a' on the top of the Stack & then reach the final State.

2ways
① for every a – push 2 a's.
② for every 2 b's – pop 1 a.

Step 1:

$\delta(q_0, a, Z_0) = (q_0, aaZ_0)$

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

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

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

$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$.

// $\cancel{\delta(q_0, \epsilon, Z_0) = (q_2, \epsilon)}$.

another method

$L = \{abb, aabbbb \ldots\}$

a a b b b b → pop
push ↑ ↑ pop one 'a'
no change   no change

odd no b's as it is
even no b's pop a

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

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

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

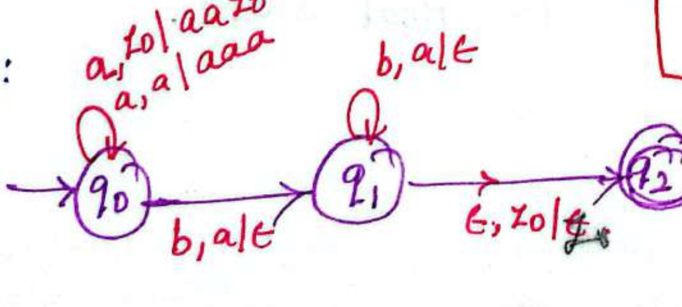
$\delta(q_1, b, a) = (q_2, \epsilon)$.

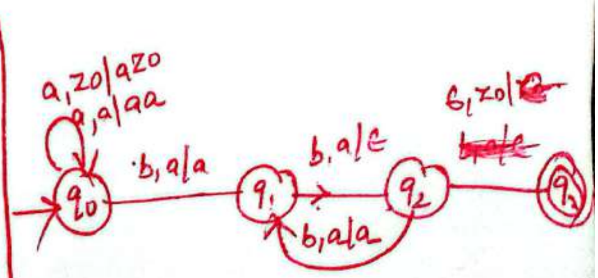
$\delta(q_2, b, a) = (q_1, a)$

$\cancel{\delta(q_2, b, a) = (q_3, \epsilon)}$

$\delta(q_2, \epsilon, Z_0) = (q_3, \cancel{\epsilon})$

Step 2:





Ex:

$w = aabbbb$.

$\delta(q_0, aabbbb, z_0) \vdash (q_0, abbbb, aaz_0)$

$\vdash (q_0, bbbb, aaaaz_0)$.

$\vdash (q_1, bbb, aaaz_0)$

$\vdash (q_1, bb, aaz_0)$

$\vdash (q_1, b, az_0)$

$\vdash (q_1, \epsilon, z_0)$

$\vdash (q_2, \epsilon)$. hence the string is accepted.

---

6) Design a PDA for the Lang $L = \{0^{2n}1^n / n \geq 0\}$.

Solo:

$0^{2n}1^n$

$\Rightarrow$ $n=1$. $n=2$

001. 000011

Here, we design the PDA, Such that when reading first '0' we go to Some state and when we read the Second zero we push the Single 0 to the Stack. and we move to the Initial State itself.

- Then, while reading 1 we try pop zero
- When reading empty String & z_0 at the top stack symbol we enter the final state.

$M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \Gamma, \delta, q_0, z_0, \{q_3\})$

$\delta(q_0, 0, 0) = (q_1, 0)$

$\delta(q_0, 0, z_0) = (q_1, z_0)$

$\delta(q_1, 0, z_0) = (\underline{q_0}, 0z_0)$. $\delta(q_1, 0, 0) = (q_0, 00)$.

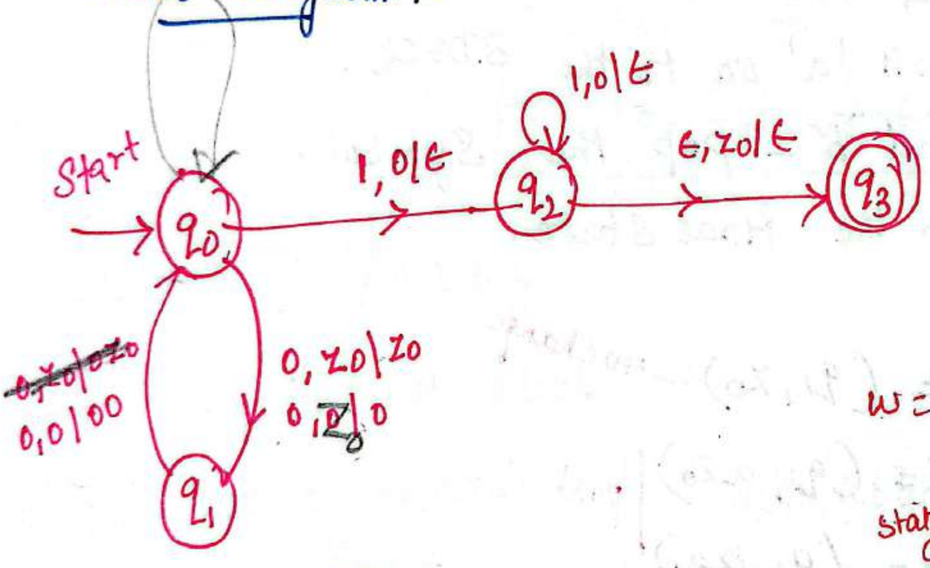
q_2.

$\delta(q_0, 1, 0) = (q_2, \epsilon)$.

$\delta(q_2, 1, 0) = (q_2, \epsilon)$.

$\delta(q_2, \epsilon, z_0) = (q_3, \epsilon)$.

Trans. diagram:-



$w = 0\ 0\ 1$

state change $q_0 \to q_1$ | push | pop.

Ex: $w = 0^{2n} 1^{n}$. $\Rightarrow$ 001.

$\delta(q_0, 001, z_0) \vdash (q_1, 01, z_0)$

$\vdash (q_0, 1, 0z_0)$

$\vdash (q_2, \epsilon, z_0)$

$\vdash (q_3, \epsilon)$. // hence it is accepted.

001

$\underline{00}\,\underline{00}\,\overline{11}$

$q_1, q_1.\ q_2\quad q_3$

push pop.

7) Design a PDA that accepts the Lang $L = \{a^{n+1} b^n / n \geq 0\}$.

Solu:- $L = \{a^{n+1} b^n / n \geq 0$.

$\therefore n = 1$ $L = \{aab ..$

$n = 2$ $L = \{aaabb$.

→ When we read first 'a' with top of stack Z_0, just Change the state and don't do any operation on the Stack.

→ Then we read next 'a' with 'a' or 'Z_0' on the top stack Symbol. PUSH 'a' on to the STACK.

→ When we read 'b' – pop the Symbol.

→ At last reach the final State.

δ:

$\delta(q_0, a, Z_0) = (q_1, Z_0)$ – No change

$\left.\begin{array}{l}\delta(q_1, a, Z_0) = (q_1, aZ_0) \\ \delta(q_1, a, a) = (q_1, aa)\end{array}\right\}$ push

$\left.\begin{array}{l}\delta(q_1, b, a) = (q_2, \epsilon) \\ \delta(q_2, b, a) = (q_2, \epsilon)\end{array}\right\}$ pop

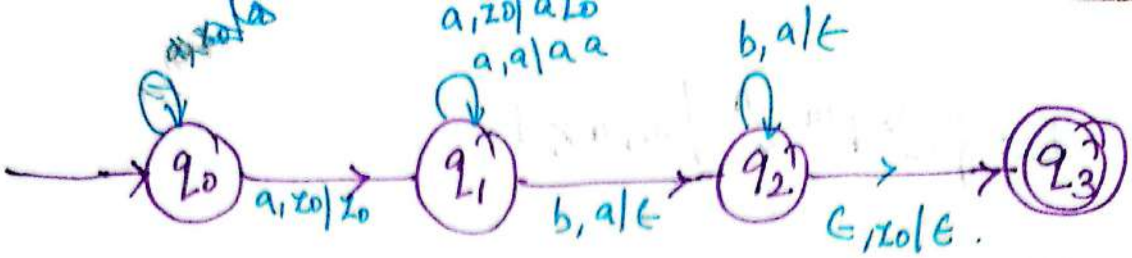
$\delta(q_2, \epsilon, Z_0) = (q_3, \epsilon)$.

The PDA P is

$P = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \Gamma, \delta, q_0, Z_0, \{q_3\})$

$q_0 \to$ initial State.

$q_3 \to$ final State.



Step 3 :

$W = aaabb.$

$\delta(q_0, aaabb, z_0) \vdash (q_1, aabb, z_0)$

$\vdash (q_1, abb, a z_0)$

$\vdash (q_1, bb, aa z_0)$

$\vdash (q_2, b, a z_0)$

$\vdash (q_2, \epsilon, z_0)$

$\vdash (q_3, \epsilon)$. // accepted.

---

8) $L = \{a^n b^{n+1} / n \geq 0\}$.

Solu: $L = \{abb, aabbb \ldots$

$\delta(q_0, a, z_0) = (q_0, a z_0)$	$\delta(q_0, abb, z_0)$
$\delta(q_0, a, a) = (q_0, aa)$	$\vdash (q_0, bb, a z_0)$.
$\delta(q_0, b, a) = (q_1, a)$	$\vdash (q_1, b, a z_0)$
$\cancel{\delta(q_0, b, z_0) = (q_1, z_0)}$	$\vdash (q_1, \epsilon, z_0)$
$\delta(q_1, b, a) = (q_2, \epsilon)$.	$\vdash (q_2, \epsilon, \epsilon)$.
$\delta(q_1, \epsilon, z_0) = (q_2, \epsilon)$.	

a b b.

push | Change the State only | pop

9) $L = \{a^n b^m c^n \mid m,n \geq 1.$

<u>Solu</u>:

- when we read 'a' with stack 'Z_0' or 'a' $\Rightarrow$ PUSH.

→ when reading 'b' with top of stack $\Rightarrow$ Just change the state only & don't perform any operations.

→ Then while reading 'c' with top stack symbol $\Rightarrow$ Pop.

∴ PDA δ:

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

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

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

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

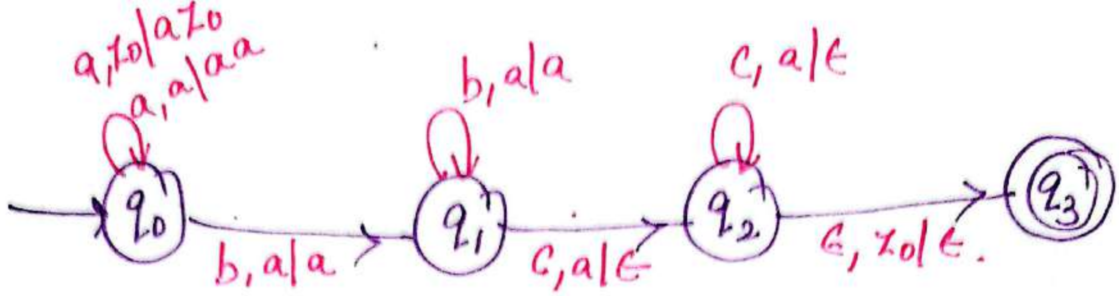
$\delta(q_1, c, a) = (q_2, \epsilon)$

$\delta(q_2, c, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, Z_0) = (q_3, \cancel{\epsilon})Z_0$

$\underbrace{aa}_{push}\ \underbrace{bb}_{\text{no change}}\ \underbrace{cc}_{pop}$

PDA $\boxed{P = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \Gamma, \delta, q_0, Z_0, \{q_3\}}$



Ex:

$w = aabbcc.$

$(q_0, aabbcc, z_0) \vdash (q_0, abbcc, az_0)$

$\vdash (q_0, bbcc, aaz_0).$

$\vdash (q_1, bcc, aaz_0)$

$\vdash (q_1, cc, aaz_0)$

$\vdash (q_2, c, az_0)$

$\vdash (q_2, \epsilon, z_0)$

$\vdash (q_3, \epsilon)$. // Accepted.

10) $L = \{a^{n+1} b^m c^n \,/\, m, n \geq 1\}.$

eg) $\delta(q_0, aabc, z_0)$

$\vdash (q_1, abc, z_0)$

$\vdash (q_1, bc, az_0)$

$\vdash (q_2, c, az_0)$

$\vdash (q_3, \epsilon, z_0)$

$\vdash (q_4, \epsilon).$

a a b c ← pop.

Change the state only. (q_1)

push

Change the state. q_2.

Equivalence of PDA & CFG.

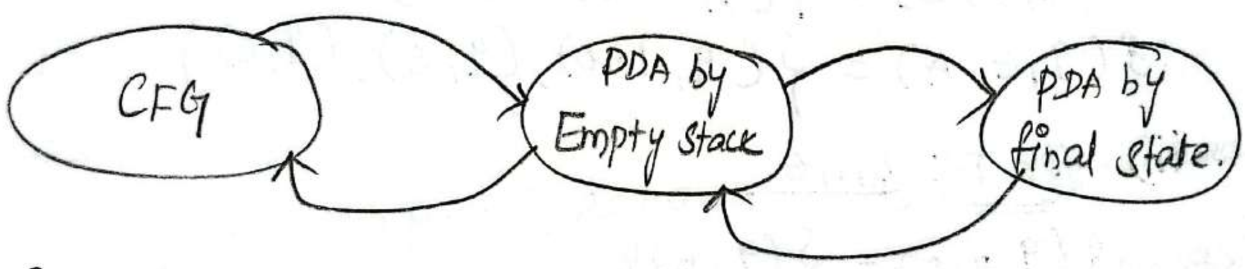
*. The CFG is recognized by PDA. The languages defined by PDA's are the CFG.

*. There are 3 Classes of languages equivalent to each other and they are,

i) The Context Free Languages.

ii) The Languages that are accepted by ~~same~~ final state of PDA.

iii) The Languages that are accepted by Empty Stack of PDA.



I. Converting CFG to PDA :-

Rules :-

a) For each Nonterminal 'A' in CFG

push $\delta(q, \epsilon, A) = \{(q, B)\}$ [A → B is a pdn of CFG).

b) For each terminal 'a' in CFG

pop $\delta(q, a, a) = \{(q, \epsilon)\}$ [a ∈ T in CFG).

c) $(q_0, \epsilon, \epsilon) = (q_0, \epsilon)$

Problems :-

1) Convert the grammar to a PDA that accepts the languages by empty stack.

$$S \rightarrow 0S1 \mid A$$
$$A \rightarrow 1A0 \mid S \mid \epsilon .$$

Solu:

$N.T \Rightarrow S, A$

$T \Rightarrow 0, 1, \epsilon$

Trans. fn of PDA:

i) For NT: S, A

$\delta(q, \epsilon, S) = \{(q, 0S1), (q, A)\}$

$\delta(q, \epsilon, A) = \{(q, 1A0), (q, S), (q, \epsilon)\}$

ii) For T: 0, 1, ϵ

$\delta(q, 0, 0) = \{(q, \epsilon)\}$

$\delta(q, 1, 1) = \{(q, \epsilon)\}$

$\delta(q, \epsilon, \epsilon) = \{q, \epsilon)\}$

ex: ID.

$w = 0101$.

$\delta(q, 0101, S) \vdash (q, 0101, 0S1)$

$\vdash (q, 101, S1)$

$\vdash (q, 101, A1)$

$\vdash (q, 101, 1A01)$

$\vdash (q, 01, A01) \vdash (q, 01, \epsilon 01)$

$\vdash (q, 01, 01) \vdash (q, 1, 1)$

$\vdash (q, \epsilon, \epsilon)$. accept.

2) Convert the gr to PDA by empty stack.

$S \to aAA$

$A \to aS \mid bS \mid a$

Solu: N.T $\Rightarrow S, A$

T $\Rightarrow a, b$.

Trans. fn of PDA:-

i) for N.T : S, A.

$\delta(q, \epsilon, S) = \{(q, aAA)\}$

$\delta(q, \epsilon, A) = \{(q, aS), (q, bS), (q, a)\}$.

ii) for T: a, b.

$\delta(q, a, a) = \{(q, \epsilon)\}$

$\delta(q, b, b) = \{(q, \epsilon)\}$.

eg) ID:

$w = aabaaa$.

$(q, aabaaa, S) \vdash (q, aabaaa, aAA) \vdash (q, abaaa, AA)$
$\vdash (q, abaaa, aA) \vdash (q, baaa, A) \vdash (q, baaa, bS)$
$\vdash (q, aaa, S) \vdash (q, aaa, aAA) \vdash (q, aa, AA)$
$\vdash (q, aa, aA) \vdash (q, a, A) \vdash (q, a, a)$
$\vdash (q, \epsilon)$.

= Thus the string "aabaa" is accepted.

3) Construct a PDA for the gr.

$S \to aB / bA$

$B \to b / bS / aBB$

$A \to a / aS / bAA$.

Solu:

$NT \Rightarrow S, A, B$

$T \Rightarrow a, b$.

I. Transition fn for PDA:

i) for NT: S, A, B.

$\delta(q, \epsilon, A) = \{(q, a), (q, aS), (q, bAA)\}$

$\delta(q, \epsilon, S) = \{(q, aB), (q, bA)\}$

$\delta(q, \epsilon, B) = \{(q, b), (q, bS), (q, aBB)\}$.

ii) for T: a, b

$\delta(q, a, a) = \{(q, \epsilon)\}$

$\delta(q, b, b) = \{(q, \epsilon)\}$.

4) $S \to AA / 0$, $A \to SS / 1$ } convert the gr. to PDA.

i) NT:

$\delta(q, \epsilon, S) = \{(q, AA), (q, 0)\}$

$\delta(q, \epsilon, A) = \{(q, SS), (q, 1)\}$

ii) T:

$\delta(q, 0, 0) = \{(q, \epsilon)\}$

$\delta(q, 1, 1) = \{(q, \epsilon)\}$

5) $E \to E+E \mid E*E \mid a$. Convert the gr to PDA:-

Solu:

i) NT:

$$\delta(q,\epsilon,E) = \{(q,E+E),(q,E*E),(q,a)\}$$

ii) T:

$$\delta(q,+,+) = \{(q,\epsilon)\}$$
$$\delta(q,*,*) = \{(q,\epsilon)\}$$
$$\delta(q,a,a) = \{(q,\epsilon)\}.$$

6) $E \to E+E \mid E*E \mid E-E \mid E/E \mid I$

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1.$

Solu:

NT: $$\delta(q,\epsilon,E) = \{(q,E+E),(q,E*E),(q,E-E),(q,E/E),(q,I)\}$$

$$\delta(q,\epsilon,I) = \{(q,a),(q,b),(q,Ia),(q,Ib),(q,I0),(q,I1).$$

T: $$\delta(q,+,+) = \{(q,\epsilon)\}.$$

$\delta(q,*,*) =$ "

$\delta(q,-,-) =$ "

$\delta(q,/,/) =$ "

$\delta(q,a,a) =$ "

$\delta(q,b,b) =$ "

$\delta(q,0,0) =$ "

$\delta(q,1,1) =$ ".

II. Converting PDA to CFG.

Theorem:-

Suppose M is a PDA, then there is a grammar G Such that $L(G) = L(M)$ ie $L(M)$ is Context Free gr.

Proof:-

We shall construct $G = (V, T, P, S)$ where the set of variables 'V' consists of

i) The special starting symbol S.

ii) All the symbols of the form $[PXq]$ where P & q are states in Q, X is a stack sym in Γ.

The Productions of the gr G as follows:

→ For all states P, 'G' has pdn

$$\boxed{S \rightarrow [q_0 Z_0 P]}$$

This $[q_0 Z_0 P]$ generates all string from state q_0 to P. That is $(q_0, w, Z_0) \vdash (P, \epsilon, \epsilon)$. This states that the starting symbol S will generate all strings.

→ Let $\delta(q, a, x)$ contains the pair $(r, y_1 y_2 \ldots y_k)$ where

i) 'a' is either symbol in Σ or $a = \epsilon$

ii) 'k' can be any number including 0, the pair is (r, ϵ).

Then for all list of states $(r_1 r_2 \ldots r_k)$ the gr G has the pdn:

$$[q x r_k] \rightarrow a [r, Y_1 r_1] [r_1 Y_2 r_2] \ldots [r_{k-1} Y_k r_k]$$

⇒ This Pdn says that one way to pop X & go to from State q to state r_k is to read a then use some i/p to pop Y_1 off the Stack while going from State r to r_1 then read some i/p that pops Y_2 off the Stack & goes from state r_1 to r_2 and so on.

Theorem:-

If $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ is a PDA, then there exists CFG G which is accepted by that PDA:-

Algorithm to find CFG from PDA:-

The G can be defined $G = (V, T, P, S)$ where $S \rightarrow$ Start sym.

$V = \{S, [q_i, z, q_j]\}$ where q_i & $q_j \in Q$, & Z is the Stack elt.

Rule: 1 The start symbol pdn is gn/. by:

$S \rightarrow [q_0, z_0, q_f]$ where $q_f \in Q$.

Rule 2: if the mapping fn/. of PDA is,

$\delta(q_i, a, Z) = (q_j, \epsilon)$ where $q_i, q_j \in Q$.

Then, pdn rule is written as

$[q_i, Z, q_j] \rightarrow a$.

Rule 3: if the mapping fun of PDA $\delta(q_i, a, Z) = (q_j, AB)$ then the pdn rule is written as

$[q_i, Z, q_k] \rightarrow a[q_i, A, q_k][q_k, B, q_j]$ where $q_i, q_j \in Q$, and q_k the all the States in Q.

Rule 4: For $\delta(q_i, a, Z) = (q_j, A)$

The pdn rule is,

$[q_i, Z, q_k] \rightarrow a[q_j, A, q_k]$.

1) Convert the PDA $P = (\{q,p\}, \{0,1\}, \{X,Z_0\}, \delta, q, Z_0)$ to a CFG if δ is given by

$\delta(q,1,Z_0) = \{(q, XZ_0)\}$

$\delta(q,1,X) = \{(q, XX)\}$

$\delta(q,0,X) = \{(p, X)\}$

$\delta(q,\epsilon,X) = \{(q,\epsilon)\}$

$\delta(p,1,X) = \{(p,\epsilon)\}$

$\delta(p,0,Z_0) = \{(q, Z_0)\}$

Soln:

For the above PDA, $G = (V, T, P, S)$

States $= \{q, p\}$,

Stack symbols $= \{X, Z_0\}$.

Initial stack symbol $= Z_0$.

Initial state $= q$.

$\therefore$ CFG for the above PDA is $G = \underline{(V, T, P, S)}$.

$V = (S, [q,X,q], [q,X,p], [p,X,q], [p,X,p],$
$[q,Z_0,q], [q,Z_0,p], [p,Z_0,q], [p,Z_0,p])$.

$= \{S, [q_i, z, q_j]\}$

$q_i, q_j \in Q$

z - stack sym

$T = \{0,1\}$.

$S = S \cdot \{q\}$

<u>P</u>:

$S \rightarrow [q_0, Z_0, q_f]$ $\quad q_f \in Q$.

$\boxed{Q = \{q, p\}}$

I. i) $S \rightarrow [q_0, Z_0, q]$

ii) $S \rightarrow [q_0, Z_0, p]$.

$V = (S, [q,X,p], [q,X,q] [p,X,p] [p,X,q]$

II. $\underline{\delta(\overset{q_i,\,a,\,Z}{q,1,Z_0}) = \{(\overset{q_j,\,A\,B}{q, XZ_0})\}}$

Rule 3: $\delta(q_i, a, Z) = (q_j, AB)$

$[q_i, Z, q_k] \to a[q_i, A, q_k][q_k, B, q_j]$

$q_k \to$ all the states in Q.

$q_i, q_j \in Q$.

q_k – all states in Q.

$\Downarrow$

$\{q, P\}$.

3) $[q, Z_0, q] \to 1[q, X, q][q, Z_0, q]$

4) $ \to 1[q, X, P][P, Z_0, q]$

5) $[q, Z_0, P] \to 1[q, X, q][q, Z_0, P]$

6) $ \to 1[q, X, P][P, Z_0, P]$

III. $\underline{\delta(q, 1, X) = \{(q, XX)\}}$. Rule 3

7) $[q, X, q] \to 1[q, X, q][q, X, q]$

8) $ \to 1[q, X, P][P, X, q]$

9) $[q, X, P] \to 1[q, X, q][q, X, P]$

10) $ \to 1[q, X, P][P, X, P]$.

III. $\underline{\delta(\overset{q_i\ \ a\ \ Z}{q, 0, X}) = \{(\overset{q_j,\ A}{P, X})\}}$

Rule 4: $\delta(q_i, a, Z) = (q_j, A)$

$[q_i, Z, q_k] \to a[q_j, A, q_k]$

11) $[q, X, q] \to 0[P, X, q]$

12) $[q, X, P] \to 0[P, X, P]$.

IV. $\underline{\delta(\overset{q_i\ \ a\ \ Z}{q, \epsilon, X}) = \{(\overset{q_j\ \ \epsilon}{q, \epsilon})}$

13) $\overset{q_i,\ Z,\ q_j}{[q, X, q]} \overset{\to a}{\Rightarrow} \epsilon$.

Rule 2: $\delta(q_i, a, Z) = (q_j, \epsilon)$

$[q_i, Z, q_j] \to a$

At last write all the PdAs:
1
2
3
⋮
16

V. $\underline{\delta(\overset{q_i\ \ a\ \ Z}{P, 1, X}) = \{(P, \epsilon)\}}$

14) $[P, X, P] \to 1$

VI. $\underline{\delta(\overset{q_i,\ a,\ Z}{P, 0, Z_0}) = \{(\overset{q_i,\ A}{q, Z_0})\}}$

$[q_i, Z, q_k] \to a[q_j, A, q_k]$

15) $[P, Z_0, q] \to 0[q, Z_0, q]$

16) $[P, Z_0, P] \to 0[q, Z_0, P]$.

3) Give the equivalent CFG for the following PDA

$M = \{ \{q_0, q_1\}, \{a, b\}, \{Z, Z_0\}, \delta, q_0, Z_0, \phi \}$ where

δ is defined as follows:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$

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

$\delta(q_0, b, Z) = (q, Z)$

$\delta(q_0, a, Z) = (q_1, Z)$

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

$\delta(q_1, a, Z_0) = (q_0, Z_0)$.

<u>Solu:</u> For the above PDA, States $\{q_0, q_1\}$, Stack Symbols $= \{Z, Z_0\}$, Initial stack symbol $= \{Z_0\}$.

Initial State $= \{q_0\}$.

$\therefore$ CFG for the above PDA is, (V, T, P, S).

$V = (S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0] [q_1, Z_0$

$[q_0, Z_0, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1 Z q$

$T = \{a, b\}$

$S = S$.

<u>P:</u>

I. $S \rightarrow [q_0, Z_0, q_0] \,/\, [q_0 Z_0 q_1]$

II. $\underline{\delta(q_0, b, Z_0) = (q_0, ZZ_0)}$.

$[q_0, Z_0, q_0] \rightarrow b [q_0, Z, q_0], [q_0, Z_0, q_0]$

$\rightarrow b [q_0, Z, q_1], [q_1, Z_0, q_0]$

$[q_0, Z_0, q_1] \rightarrow b [q_0, Z, q_0] \; [q_0, Z_0, q_1]$

$\rightarrow b [q_0 Z q_1] \; [q_1, Z_0, q_1]$.

III. $\underline{\delta(q_0, b, Z) = (q_0, ZZ)}$

$[q_0, Z, q_0] \rightarrow b[q_0, Z, q_0]\,[q_0, Z, q_0]$
$\rightarrow b\,[q_0, Z, q_1]\,[q_1, Z, q_0]$.

$[q_0, Z, q_1] \rightarrow b[q_0, Z, q_0]\,[q_0, Z, q_1]$
$\rightarrow b\,[q_0, Z, q_1]\,[q_1, Z_0, q_1]$.

IV. $\underline{\delta(q_0, a, Z) = (q_1, Z)}$

$[q_0, Z, q_0] \rightarrow a\,[q_1, Z, q_0]$

$[q_0, Z, q_1] \rightarrow a\,[q_1, Z, q_1]$.

V. $\underline{\delta(q_1, b, Z) = (q_1, \epsilon)}$

$[q_1, Z, q_1] \rightarrow b$

VI. $\underline{\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)}$

$[q_0, Z_0, q_0] \rightarrow \epsilon$.

VII. $\underline{\delta(q_1, a, Z_0) = (q_0, Z_0)}$

$[q_1, Z_0, q_0] \rightarrow a\,[q_0, Z_0, q_0]$

$[q_1, Z_0, q_1] \rightarrow a\,[q_0, Z_0, q_1]$.

4) Construct the grammar for the following PDA.

$M = (\{q_0, q_1\}, \{0,1\}, \{x, z_0\}, \delta, q_0, z_0, \phi\}$ and

where δ is gn/. by.

$\delta(q_0, 0, z_0) = \{(q_0, xz_0)\}$

$\delta(q_0, 0, x) = \{(q_0, xx)\}$

$\delta(q_0, 1, x) = \{(q_1, \epsilon)\}$

$\delta(q_1, 1, x) = ((q_1, \epsilon)\}$

$\delta(q_1, \epsilon, x) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, z_0) = \{(q_1, \epsilon)\}$

<u>Solu:</u>

The states are = $\{S, [q_0, x, q_0], [q_0, x, q_1]$
$[q_1, x, q_0]\ [q_1, x, q_1]$
$[q_0, z_0, q_0]\ [q_0, z_0, q_1]$
$[q_1, z_0, q_0]\ [q_1, z_0, q_1]\}$.

I. $S \rightarrow [q_0, z_0, q_0] / [q_0, z_0, q_1]$

II. $\delta(q_0, 0, z_0) = (q_0, xz_0)$

$[q_0, z_0, q_0] \rightarrow 0[q_0, x, q_0][q_0, z_0, q_0]$
$\rightarrow 0[q_0, x, q_1].[q_1, z_0, q_0]$

$[q_0, z_0, q_1] \rightarrow 0[q_0, x, q_0][q_0, z_0, q_1]$
$\rightarrow 0[q_0, x, q_1][q_1, z_0, q_1]$

II. $\delta(q_0, 0, x) = (q_0, xx)$

$[q_0, x, q_0] \rightarrow 0[q_0, x, q_0][q_0, x, q_0]$
$\rightarrow 0[q_0, x, q_1][q_1, x, q_0]$

$$[q_0, X, q_1] \rightarrow 0[q_0, X, q_0][q_0, X, q_1]$$
$$\rightarrow 0[q_0, X, q_1][q_1, X, q_1]$$

III. $\delta(q_0, 1, X) = (q_1, \epsilon)$

$$[q_0, X, q_1] \rightarrow 1.$$

IV. $\delta(q_1, 1, X) = (q_1, \epsilon)$

$$[q_1, X, q_1] \rightarrow 1.$$

V. $\delta(q_1, \epsilon, X) = (q_1, \epsilon)$

$$[q_1, X, q_1] \rightarrow \epsilon.$$

VI. $\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon).$

---

5) Construct PDA to accept the Language $L = \{wcw^R\}$ where $w = \{a, b\}^*$ and use the Productions of PDA to construct the CFG generating $N(M)$.

Solu:

The PDA for the Lang $L = \{wcw^R \mid w = \{a, b\}^*$ is as follows:-

1. $\delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}$
2. $\delta(q_0, b, Z_0) = \{(q_0, BZ_0)\}$
3. $\delta(q_0, a, A) = \{(q_0, AA)\}$
4. $\delta(q_0, b, A) = \{(q_0, BA)\}$

5. $\delta(q_0, a, B) = \{(q_0, AB)\}$

6. $\delta(q_0, b, B) = \{(q_0, BB)\}$

7. $\delta(q_0, c, Z_0) = \{(q_1, Z_0)\}$

8. $\delta(q_0, c, A) = \{(q_1, A)\}$

9. $\delta(q_0, c, B) = \{(q_1, B)\}$

10. $\delta(q_1, a, A) = \{(q_1, \epsilon)\}$

11. $\delta(q_1, b, B) = \{(q_1, \epsilon)\}$

12. $\delta(q_1, \epsilon, Z_0) = \{(q_2, \epsilon)\}$.

The PDA P is defined as,

$P = (\{q_0, q_1, q_2\}, \{a, b\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \{q_2\})$.

CFG, $G = (V, T, P, S)$.

$V = (S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_0, A, q_0], [q_0, A, q_1], [q_1, A, q_0]$ $[q_1, A, q_1]$
$[q_0, B, q_0]$ $[q_0, B, q_1], [q_1, B, q_0]$ $[q_1, B, q_1].)$

$T = \{a, b, c\}$

$S = S$.

Pumping lemma for CFL.

$z = uvwxy$

i) $|vx| \geq 1$

ii) $|vwx| \leq n$

iii) for all $i \geq 0$, $uv^iwx^iy \in L$.

1) Prove that the Lang L is not a CFL:

Solu: $L = \{a^n b^n c^n \mid n \geq 1$.

$L = \{0^n 1^n 2^n \mid n \geq 1$

$L = \{012, 001122 \ldots$

By PL, $z = 0^n 1^n 2^n$.

$z = uvwxy$

$|vx| \geq 1$

$|vwx| \leq n$

$uv^iwx^iy \in L$.

we prove that $uv^iwx^iy \notin L$.

$z = \underset{u}{0^n}\ \underset{vwx}{1^n}\ \underset{y}{2^n}$.

$= 0^n\ \underline{1\ 1^{n-2}\ 1}\ 2^n$. ($u$, v, w, x)

for all i, $0^n\ \underline{1}^i\ 1^{n-2}\ \underline{1}^i\ 2^n$

$i = 1$, $z \in L$.

$i = 2$ $z \notin L$. ∴ proved.

2. P.T the Lang $L = \{ww \mid w$ is a bit string$\}$ & is not a CFL.

Solo :

$L = \{ww$

$L = \{\underbrace{0^n 1^n}_{w} \underbrace{0^n 1^n}_{w}$.

$L = \{0101, 00110011 \dots\}$

By Pumping Lemma we consider $z = uvwxy$

$$\boxed{\begin{array}{l} |vx| \geq 1 \\ |vwx| \leq n \\ \forall i \text{ and } i \geq 0 \\ uv^i w x^i y \in L \end{array}}$$

To prove that $uv^i wx^i y \notin L$

$|n| = \underset{u}{0^n} \underset{vwx}{1^n} \underset{y}{0^n 1^n}$

$= \underset{u}{0^n} \underset{v}{1^1} \underset{w}{1^{n-2}} \underset{x}{1^1} \underset{y}{0^n 1^n}$

$z = uv^i wx^i y$ for all i, $0^n 1^i 1^{n-2} 1^i 0^n 1^n$

$i = 1 \Rightarrow 0^n 1^1 1^{n-2} 1^1 0^n 1^n \in L$

$i = 2 \Rightarrow 0^n 1^2 1^{n-2} 1^n 0^n 1^n \notin L$

The Language is not CFL

Univ. Q.P - III-unit

1) Differentiate PDA acceptance by empty stack method with acceptence by the final state method. (Apr/may 17).
2) When is PDA is said to be deterministic? (Nov/dec 16)
3) what are the conventional notations of PDA? (")
4) Define PDA?
5) Does a PDA have memory? Justify. } may/june 16.
6) What are the different ways of language acceptances by a PDA and define them?
7) Convert the following CFG to a PDA } Nov/dec 15.

   $S \rightarrow aAA$

   $A \rightarrow aS \mid bS \mid a$.

Part-B.

1) Convert the following CFG to PDA & verify that for (a+b) and a++.

   $I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$.

   $E \rightarrow I \mid E+E \mid E*E \mid (E)$.

2) i) Construct a DPDA for even length palindrome.

   ii) Prove : if PDA P is constructed from CFG G by the above construction, then $N(P) = L(G)$.

} APR/may 17.

1) i) Outline an Instantenous description of a PDA.

   ii) State & explain the pumping lemma for CFG.

2) With an example, explain the procedure to obtain a PDA from the given CFG.

} Nov/dec 16.

may / June 16

1) i) $L = \{ww^R \mid w \in \Sigma^+\}$. $\Sigma = \{a, b\}$. (10)m.

ii) What is an Instantaneous description that the PDA? (6)
How will you represent it? And also give three Important Principles of ID and their transactions.

2) i) Explain acceptance by final state and acceptance by empty stack of a PDA. (8)

ii) State the PL for CFL. Use PL to show that the Lang $L = \{a^i b^j c^k \mid i < j < k\}$ is not a CFL. (8)

Nov / dec 15

1) i) Design a PDA $\{0^n 1^n \mid n > 1\}$. draw the transition diagram for the PDA. Show by Instantaneous description that the PDA accepts the string "0011". (10 m)

ii) State the PL for CFL & show that the Lang
$L = \{a^n b^n c^n \mid n >= 1\}$ is not a CFL.

(or)

i) Convert PDA to CFG. PDA is given by
$P = (\{P, q\}, \{0, 1\}, \{X, Z\}, \delta, q, Z)$, δ is defined by

$\delta(P, 1, Z) = \{(P, XZ)\}$, $\delta(P, \epsilon, Z) = \{(P, \epsilon)\}$
$\delta(P, 1, X) = \{(P, XX)\}$
$\delta(q, 1, X) = \{(q, \epsilon)\}$ (10m)
$\delta(P, 0, X) = \{(q, X)\}$
$\delta(q, 0, Z) = \{(P, Z)\}$

ii) What are deterministic PDA's? Give example for Non-deterministic & deterministic PDA.

Types OF PDA :-

i) → Non-deterministic PDA.
ii) → Deterministic PDA.

(i) NDPDA:

It can have a no. of moves for the same combination of state, i/p and stack symbol and it can have a choice for some combination of state and stack symbol, between reading an i/p symbol and making transition without reading one.

(ii) D-PDA :-

if PDA is deterministic, there is never a choice of move. The PDA is said to be deterministic if it has no moves in some configurations.

→ For a single i/p it moves only into the same state or a new state.

Let $p = (Q, \Sigma, T, \delta, q_0, Z_0, F)$ be a DPDA, if it satisfies the following conditions.

1. For any given i/p symbol and stack symbol, DPDA has at most one move.

   $\delta(q, a, X)$ has at most one member for q in Q, a in Σ, X in T.

2. When ϵ-move is possible for some PDA, no input consuming alternative is available.

   $\delta(q, a, X)$ is non empty for a in Σ then $\delta(q, \epsilon, X)$ must be empty.

A deterministic PDA accepts a deterministic Context-free Language (DCFL).

Regular Lang & DPDA:-

DPDA's accepts the class of Languages b/w CFG & RL. if L is a regular Lang (RL) then

$L = L(P)$ for Some PDA P.

DPDA's and CFL:-

The languages accepted by DPDAs by final state, includes both the regular Lang & also CFL's.

DPDA's and ambiguous grammar:-

DPDA's accepts only unambigous grammars.

if $L = N(P)$ for some PDA P, then L has an unambiguous CFG.

if $L = L(P)$ for Some PDA P, then L has an unambiguous CFG.