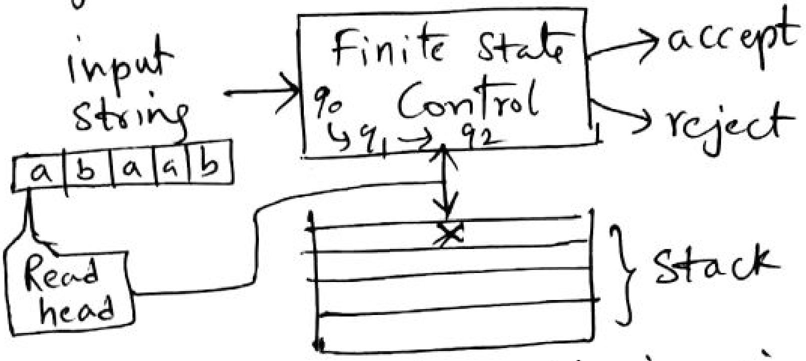
Introduction to Push Down Automata :- (PDA)

PDA is basically an ϵ-NFA with ϵ-Transistions Permitted and a Stack on which it can store a string of Stack Symbols. A PDA can be represented Pictorially as



A PDA is designed to accept set of strings in a given Language and reject other strings. The Languages accepted by PDA are called Context Free Languages.

Working of PDA :- Push Down Automata begins with initial state q_0 and read head pointing to the leftmost symbol of input string. PDA also reads top symbol of the stack. The PDA is in some present state q reads current symbol a of input string or ϵ and top symbol say X of stack.

(1) PDA may goto a next state which may or may not be the previous state. that is

$$\delta(q, a \text{ or } \epsilon, X) = (q_1, X)$$

(2) Replaces top symbol of stack (X) by string say Y_2.

$$\delta(q, a, X) = (q, Y_2)$$

(3) Pop the top symbol of the stack.

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

(4) PDA may not change the top symbol of stack.

$$\delta(q, a, X) = (q, X).$$

On reading current symbol of input string the read head moves to point to the next symbol. In this way the PDA reads each symbol of the input string and behaves in one of the 4 ways. The PDA finally enters accepting state or empty the stack for the Valid string and enters Non-Final State or does not empty the stack for invalid string.

Formal Definition of Push Down Automata (PDA)

A PDA involves Seven Components. $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$. That is PDA Consist of

(1) Finite Set of States, denoted by Q

(2) Finite Set of input symbols denoted by Σ. It is the input symbols that form the input string.

(3) Finite Set of stack symbols denoted by Γ (tou). these Stack symbols are pushed onto and poped out of the stack.

(4) Transistion function denoted by δ which takes 3 arguments (Present state, Present input symbol or ϵ, top symbol of stack) and returns next state and string of stack symbols that replaces top of stack

Formally, δ takes as arguments a tripple $\delta(q, a, X)$

where 1) q is a state in Q

2) a is either an input symbol in Σ or $a = \epsilon$, the empty string which is not the input symbol.

3) X is a top symbol of the stack.

The output of δ is a pair (p, γ) where p is a new state and γ is a string of stack symbols that replaces X at the top of the stack.

If $\gamma = \epsilon$, stack is poped, if $\gamma = X$, then stack is unchanged and if $\gamma = YZ$, then X is replaced by Z and Y is pushed onto the stack.

(5) initial state denoted by q_0

(6) initial stack symbol denoted by Z_0.. Initially, stack contains initial stack symbol say Z.

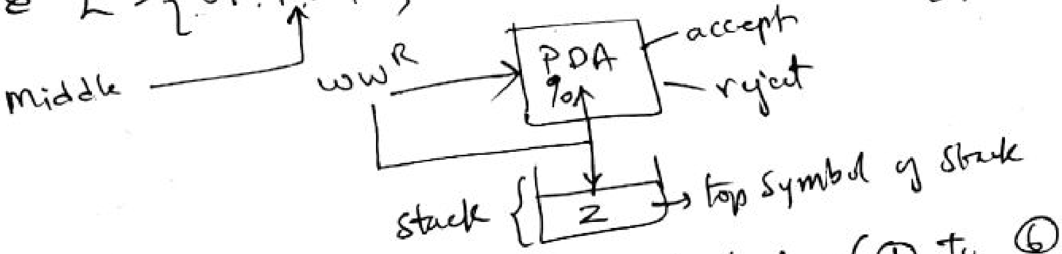
(7) Set of accepting states.

<u>Problems</u>:- Design NPDA for the Language

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

The given language is the set of <u>evenlength Palindrome</u> strings over $\Sigma = \{0, 1\}$

ϵ $L = \{ \overbrace{0101}^{w}\overbrace{1010}^{w^R}, \overbrace{1010}\overbrace{0101}, \overbrace{110}\overbrace{011}, \overbrace{11}\overbrace{11}, \overbrace{011}\overbrace{110}, \overbrace{0011}\overbrace{1100}, \underbrace{1100}\underbrace{0011}, \ldots \}$



<u>Assumptions</u>:-

(1) <u>Middle of string is not reached</u>:- (① to ⑥)
Read input symbols of w and store them onto stack (push), stay in q_0.

(2) <u>Middle string is reached</u>: (⑦ ⑧ & ⑨)
Donot read input symbols (ϵ) of w and simply move to state q_1 leaving stack top symbol as it is. We are now in q_1.

After the middle string is reached, Now read each symbol of w^R if the symbol of w^R matches with top symbol of stack, Pop the top symbol of the stack, Repeat this for all symbols of w^R. (⑩ & ⑪).

<u>Note</u>: NPDA (Non-Deterministic PDA. <u>Refer</u> Definition of DPDA.

After we finish reading the symbols of w^R, only Z will be on the stack. Therefore without reading input symbol (ε) goto state q_2 leaving the top symbol Z as it is (acceptance by Final state) or pop ~~the~~ Z (acceptance by empty stack).

The PDA for the given Problem is

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where

δ is given by:

(1) Present state, Present input symbol, Top Stack symbol, Next state, Top Stack Contents

w

(1) $\delta(q_0, 0, Z) = (q_0, 0Z)$ Push 0 onto stack

(2) $\delta(q_0, 1, Z) = (q_0, 1Z)$ Push 1 —"—

(3) $\delta(q_0, 0, 0) = (q_0, 00)$ Push 0 —"—

(4) $\delta(q_0, 1, 1) = (q_0, 11)$ Push 1 —"—

(5) $\delta(q_0, 0, 1) = (q_0, 01)$ Push 0 —"—

(6) $\delta(q_0, 1, 0) = (q_0, 10)$ Push 1 —"—

middle

(7) $\delta(q_0, \epsilon, Z) = (q_1, Z)$ goto state q_1

(8) $\delta(q_0, \epsilon, 0) = (q_1, 0)$ —"—

(9) $\delta(q_0, \epsilon, 1) = (q_1, 1)$ —"—

w^R

(10) $\delta(q_1, 0, 0) = (q_1, \epsilon)$ PoP 0

(11) $\delta(q_1, 1, 1) = (q_1, \epsilon)$ PoP 1

(12) $\delta(q_1, \epsilon, Z) = (q_2, Z)$ enter q_2 (Final state)

OR

(12) $\delta(q_1, \epsilon, Z) = (q_1, \epsilon)$ PoP Z and empty the stack

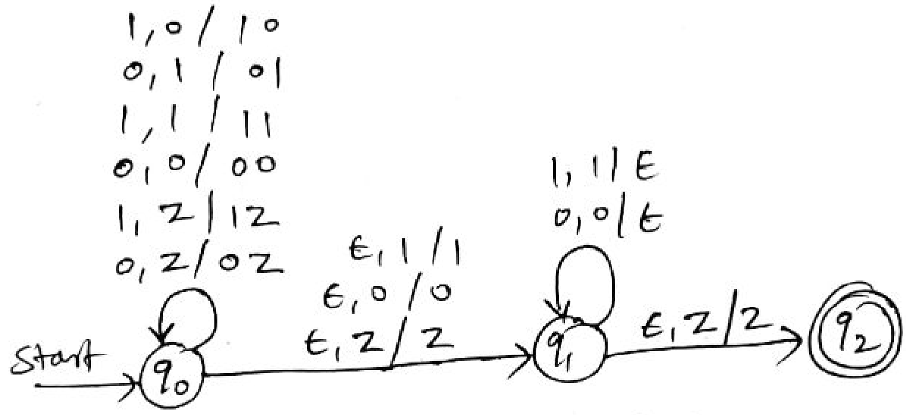
$Q = \{q_0, q_1, q_2\}$ $q_0 = q_0$ initial state of PDA

$\Sigma = \{0, 1\}$ $Z_0 = Z$ (initial stack symbol

$\Gamma = \{Z_0, 0, 1\}$ $F = \{q_2\}$ set of final states

Transistion Diagram :-

Convert the transistiory functions (δ) into equivalent Transistion Diagram of PDA.



Instantaneous Description (ID) of PDA :-

The PDA moves from one Configuration to another while processing the given input string. We represent the Configuration of PDA by a tripple (q, w, γ) where (gama)

q is present state of the PDA.

w is the remaining input string to be read

γ is the stack Contents.

We show the top of the stack to the left end of γ.

Such a tripple is Called Instantaneous Description (ID) of the PDA.

Language of PDA :- (Set of strings accepted by the PDA)

(1) **Acceptance by Final State** :- Let $P = (Q, \Sigma, \Gamma, \delta, q_0 Z_0, F)$ be a PDA. The Language accepted by PDA P by Final state denoted as $L(P_F)$ is :-

$$L(P_F) = \{ w \mid (q_0, w, Z_0) \overset{*}{\underset{P}{\vdash}} (q, \epsilon, \alpha) \}$$

for Some state q in F and any stack string α.

That is Starting With initial state q_0 with string w Waiting on the input, P Consumes w and enters the accepting state. The Contents of stack at that time are irrelevant.

(2) Acceptance by Empty Stack :-

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. The Language accepted by PDA P by empty stack denoted by $L(P_E)$ is :-

$$L(P_E) = \{ w \mid (q_0, w, Z_0) \overset{*}{\underset{P}{\vdash}} (q, \epsilon, \epsilon) \}$$

for any state q. That is $L(P_E)$ is the set of input strings that PDA P can consume and at the same time empty its stack.

Problem: Write Sequence of moves (ID's) made by PDA of previous problem for the input string $w = 1111$

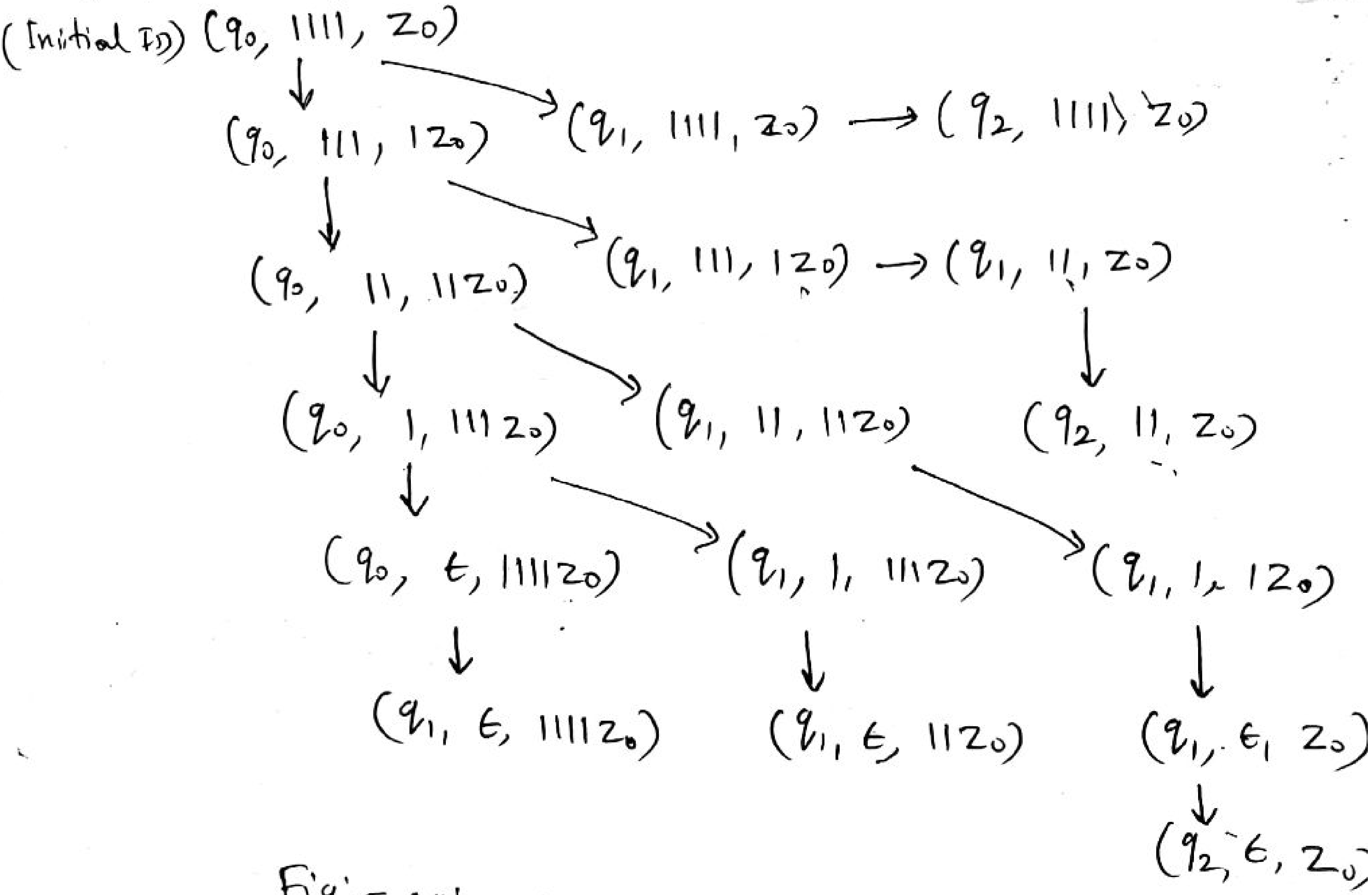
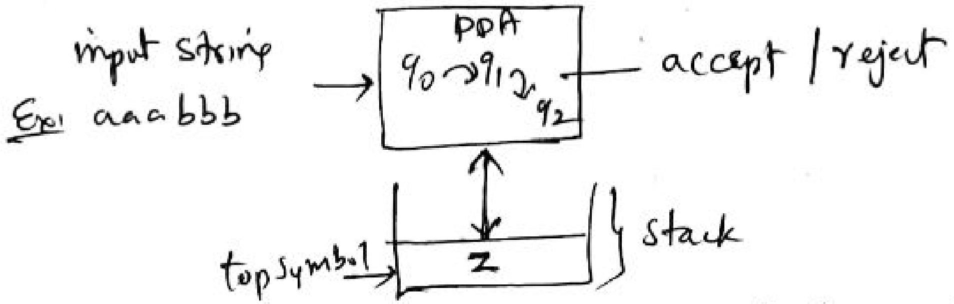


Fig:- ID's of PDA on input 1111

Problems: (1) Construct PDA that accepts the Language $L = \{a^n b^n \mid n \geq 1\}$

Solution: Let $L = \{ab, aabb, aaabbb, \ldots\}$



Procedure: 1. Read input symbols a and Push a onto the Stack

2. Read input symbols b and Pop a out of the Stack.

3. Read empty string (ε) when Z is top symbol and enter accepting state OR Pop Z

1) $\delta(q_0, a, z) = (q_0, az)$ read a when z is top symbol and Push a

2) $\delta(q_0, a, a) = (q_0, aa)$ read a when a is top symbol and Push a

3) $\delta(q_0, b, a) = (q_1, \epsilon)$ read b when a is top symbol. Pop a and enter state q_1

4) $\delta(q_1, b, a) = (q_1, \epsilon)$ read b when a is top Symbol, Pop a. ~~and~~

5) $\delta(q_1, \epsilon, z) = (q_2, z)$ read empty string (ε) when Z is top Symbol and enter Final State q_2

OR

(q_1, ϵ) read empty string (ε) when Z is top Symbol, ~~and~~ Pop Z and thereby empty the stack.

⇓

Instantaneous Descriptions: of PDA (ID)

We represent Configuration of PDA by a tripple (q, w, γ) where q is a present state of PDA. w is the Remaining input symbols to be read. γ is the Stack Contents. We show the top symbol of the Stack to the left end of γ. Such a tripple is called Instantaneous Description (ID)

NOTE:- PDA moves from one Configuration to another Configuration during the process of Reading the Symbols of the input string. ID is the Description of Configuration of PDA at time t.

change from one Configuration to another is represented using the symbol $\vdash$

Problem(2):- Use the transistion functions ① to ⑤ of the previous PDA. Write instantaneous Descriptions (ID) of PDA for the input string $W = aaabbb$.

$(q_0, aaabbb, Z) \vdash (q_0, aabbb, aZ) \vdash (q_0, abbb, aaZ)$

$\vdash (q_0, bbb, aaaZ)$

$\vdash (q_1, bb\ aaZ)$

$\vdash (q_1, b, aZ)$

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

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

OR

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

The PDA for $L = \{a^n b^n \mid n \geq 1\}$ is:

⇒ $P_1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

where $Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

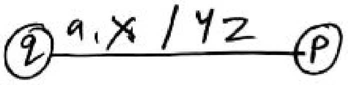
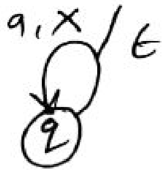
$\Gamma = \{Z, a, b\}$

$q_0 = q_0$ (start state of PDA

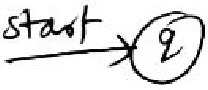
$Z_0 = Z$ (initial stack symbol)

$F = \{q_2\}$ set of accepting state of PDA

Transistion Diagram of PDA :- It is possible to Construct Transistion Diagram of a PDA. It is a pictorial (graphical) representation of Push Down Automaton (PDA). Transistion Diagram can be constructed as follows

General transistion function of PDA	Equivalent Transistion Diagram of PDA
$\delta(\underbrace{q}_{\text{Present State}}, \underbrace{a}_{\text{Current input symbol}}, \underbrace{X}_{\text{Top symbol of stack}}) = (p, YZ)$	 Next state Symbols that replace X
$\delta(q, a, X) = (q, \epsilon)$	

As Usual, Initial & Final States are represented by



and

repectively

Problem (3): Construct Transistion Diagram of the PDA Constructed for Problem ①.

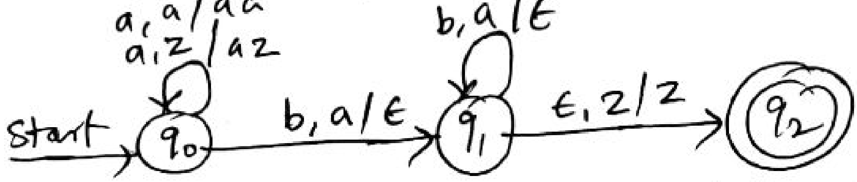


Fig (1): T.D. of PDA that accepts by Final State

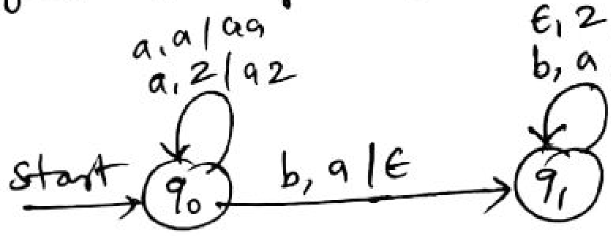


fig: T.D. of PDA that accepts by empty stack.

Problem(3) Construct PDA accepting the following Language: $L = \{ w \mid w \in (a+b)^* \text{ and } n_a(w) = n_b(w) \}$.

Solution:- The Language Consist of Set of all Strings Consisting of equal numbers of a's and b's

Let L = { ab, ba, abab, abba, baab, aabb, baba, abbababa, babbaa, ... }

Procedure: 1. read the first Symbol of input String (a or b) and push it on to the Stack

2. read a when a is Top Symbol and Push a

3. read b when b is Top Symbol and Push b

4. read a when b is Top Symbol Pop b

5. read b when a is Top Symbol Pop a

6. read empty string (ϵ) (no Symbol) when z is the top Symbol and enter accepting state.

∴ The PDA is

1) $\delta(q_0, a, z) = (q_0, az)$ read a when z is top Symbol and Push a

2) $\delta(q_0, b, z) = (q_0, bz)$ read b when z is top Symbol Push b

3) $\delta(q_0, a, a) = (q_0, aa)$ read a when a is top Symbol Push a

4) $\delta(q_0, b, b) = (q_0, bb)$ read b when b is top Symbol Push b

5) $\delta(q_0, a, b) = (q_0, \epsilon)$ read a when b is top Symbol, Pop b

6) $\delta(q_0, b, a) = (q_0, \epsilon)$ read b when a is top Symbol, Pop a

7) $\delta(q_0, \epsilon, z) = (q_f, z)$ read no Symbol (ϵ) when Z is Top Symbol, Enter accepting state.

∴ The PDA for the given Problem is ;

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

Where $Q = \{q_0, q_1\}$

$\Sigma = \{a, b\}$

$\Gamma = \{Z_0, a, b\}$

δ = Transistion Functions ① to ⑦

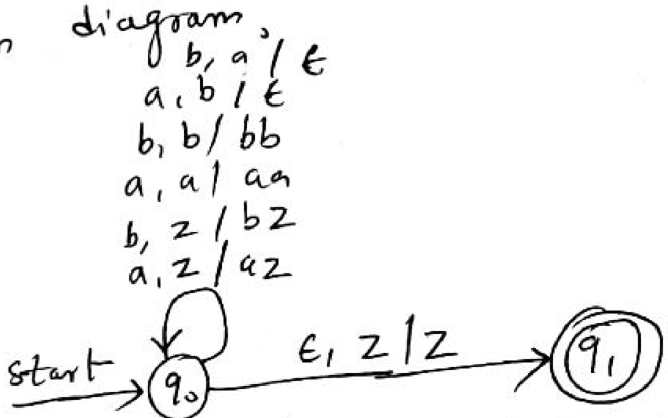
$q_0 = q_0$ start State

$Z_0 = Z$ (initial stack Symbol)

$F = \{q_1\}$ (set of final states.

Problem:- Write graphical representation of PDA of Previous Problem.

Use transistion functions ① to ⑦ to Construct Transistion diagram.



Problem:- Write instantaneous Description (ID) of PDA of problem (1) for the string W = abba

$$(\underline{q_0}, \underline{a}bba, \underline{Z}) \vdash (q_0, \underline{b}ba, \underline{a}Z) \vdash (\underline{q_0}, \underline{b}a, \underline{Z})$$

(initial Configuration)

$$\vdash (\underline{q_0}, \underline{a}, \underline{b}Z)$$

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

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

(Final Configuration)

Define Language of PDA (Important)

The Set of strings (Language) accepted by PDA Can be defined as follows

(1) Acceptance by Final state :- Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. The language accepted by PDA denoted as $L(P_F)$ by final state Can be Written as

$$L(P_F) = \left\{ w \;\middle|\; (q_0, w, Z_0) \overset{*}{\underset{P}{\vdash}} (q, \epsilon, \alpha) \right\}$$

for Some State q in F and any Stack String α. That is starting With initial ID with String w as input string, P Consumes w and enters accepting state. The Contents of stack at that time are irrelevant.

(2) Acceptance by empty stack : Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. The Language accepted by PDA P by empty stack is :

$$L(P_E) = \left\{ w \;\middle|\; (q_0, w, Z_0) \overset{*}{\underset{P}{\vdash}} (q, \epsilon, \epsilon) \right\}$$

for any state q. That is $L(P_E)$ is the Set of all input strings that P Can Consume and at the Same time emty its stack.

Problem(4) Construct PDA accepting the following Language

$$L = \{ w \mid w \in (a+b)^* \text{ and } n_a(w) > n_b(w) \}$$

The Language Consist of Set of all strings whose number of a's is greater than no. of b's

$L = \{ aab, \; aabaab, \; baabaaab. , \ldots \}$

The given problem is Similar to Problem (3) except that the string has excess a's. By the time we finish reading Symbols of the input string, if the stack still Contains Some a's in it, then the input string has excess a's than b's & it must be accepted. otherwise string must be rejected.

Transistion function (δ) from (1) to (6) are Similar to previous problem. and include the following transistion function. (7) $\delta(q_0, \epsilon, a) = (\underset{\substack{\downarrow \\ \text{(accepting state)}}}{q_1}, a)$

Problem(5) Construct PDA accepting the Language

$$L = \{ w \mid w \in (a+b)^* \text{ and } n_a(w) < n_b(b) \}$$

Here each string has excess b's than a's.

include the transistion function

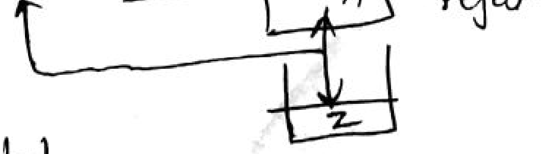
(7) $\delta(q_0, \epsilon, b) = (q_1, b)$

Problem(6) :- Design PDA accepting the Language

$L = \{ a^n b^{2n} \mid n \geq 1 \}$ or $L = \{ 0^n 1^{2n} \mid n \geq 1 \}$

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

The Language Consist of Set of all strings with a's followed by b's and number of b's is twice as the number of a's.



Procedure :-

(1) Read a and Push two a's onto the Stack

(2) Read b when a is stack top symbol, Pob b. Repeat this for all b's in input string.

(3) Finally without reading input string (ϵ) and Z is on top of the stack, enter accepting state.

δ is given by:

(1) $\delta(q_0, a, z) = (q_0, aa)$ Read a Push two a's

(2) $\delta(q_0, a, a) = (q_0, aa)$ Read a Push two a's

(3) $\delta(q_0, b, a) = (q_1, \epsilon)$ Read b when a is top Stack symbol, PoP a Enter State q_1.

(4) $\delta(q_1, b, a) = (q_1, \epsilon)$ Read b when a is top stack symbol, PoP a

(5) $\delta(q_1, \epsilon, z) = (q_2, z)$ read ϵ when z is on top of stack, enter final state q_2.

$\therefore P = (Q, \Sigma, \delta, q_0, z_0, F)$

where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$ $\Gamma = \{z, a, b\}$, $q_0 = q_0$

$z_0 = z_0$, $F = \{q_2\}$

Problem (6):- Design PDA accepting the language

$L = \{a^{2n} b^n \mid n \geq 1\}$ or $L = \{0^{2n} 1^n \mid n \geq 1\}$

The Language consist of set of all string with a's followed by b's and number of a's is twice as the no. of b's.

(1) $\delta(q_0, a, z) = (q_0, az)$

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

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

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

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

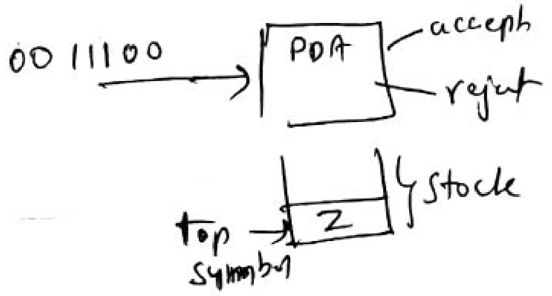
(6) $\delta(q_2, \epsilon, z) = (q_3, \epsilon)$

$\therefore$ PDA $P = (\underbrace{\{q_0, q_1, q_2, q_3\}}_{Q}, \underbrace{\{a, b\}}_{\Sigma}, \delta, \underbrace{q_0}, \underbrace{z}_{z_0}, \underbrace{\{q_2\}}_{F})$

Problem (7): Design PDA accepting the Language

$L = \{ 0^n 1^m 0^n \mid m, n \geq 1 \}$

$L = \{ 0011100,\ 00011000,\ 00111100, \ldots \}$



General Procedure:

(1) Read 0 and Push 0 onto stack.

(2) Read 1 when 0 is on top of stack (Repeat (2) for all 1's on input string)

(3) Read 0 when 0 on top of stack PoP 0.

(4) Read ε when Z on top of stack. enter accepting state.

1) $\delta(q_0, 0, Z) = (q_0, 0Z)$ Read 0, Push 0

2) $\delta(q_0, 0, 0) = (q_0, 00)$ Read 0 Push 0

3) $\delta(q_0, 1, 0) = (q_1, 0)$ Read 1 move to q_1

4) $\delta(q_1, 1, 0) = (q_1, 0)$ Read 1

5) $\delta(q_1, 0, 0) = (q_1, \epsilon)$ PoP 0

6) $\delta(q_1, \epsilon, Z) = (q_2, Z)$ enter accepting state.

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

Equivalence of PDA and CFG (Grammer).

CFG - Context Free Grammer.

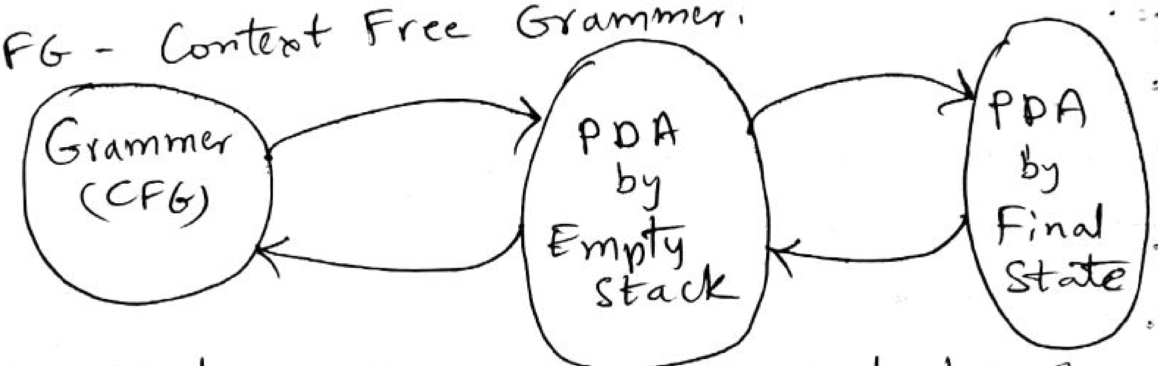


Fig:- Equivalence of 3 notations that describe Context Free Language (CFL)

A Grammer Can be Converted into its equivalent PDA that accept by empty Stack and Vice-Versa.

From Grammer to PDA :- Given a CFG $G=(V, T, P, S)$ It Can be Converted into its equivalent PDA $P=(Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ that accepts by empty Stack Such that $L(G) = L(P) = L$.

The General Procedure is :- Begin with initial state q_0 of PDA.

1) Push Start Variable Say S of grammer onto the Stack. i.e. $\delta(q_0, \epsilon, Z) = (q_1, SZ)$.

2) Apply Rule 1 to Convert each production of given grammer into equivalent transistion function of PDA.

Rule 1 is :-

For each Production $A \to \alpha$ the equivalent transistion function of PDA is $\delta(q_1, \epsilon, A) = (q_1, \alpha)$

3) Apply Rule 2 to Convert each terminal to δ of PDA

Rule 2 is :-

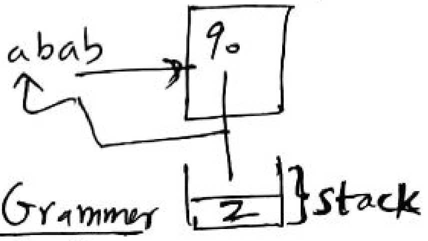
for each Terminal a, the equivalent Transistion Function δ is $\delta(q, a, a) = (q, \epsilon)$

① Convert the following grammer to PDA that accepts by empty stack. $S \to aSb \mid bSa \mid SS \mid \epsilon$

Solution:- Variables - S Terminals - a, b,
Start Variable - S

Begin with initial state q_0 of PDA and Z on top of stack.

Push Start Variable of grammer onto stack.



PDA

∴ <1> $\delta(q_0, \epsilon, z) = (q_1, SZ)$

Apply Rule1 to Convert each production of Grammer into its equivalent Transistion function of PDA.

Production Transistion Function

In general, $A \to \beta$ $\delta(q_1, \epsilon, A) = (q_1, \beta)$

$S \to aSb \mid bSa \mid SS \mid \epsilon$ (2) $\delta(q_1, \epsilon, S) = \{(q_1, aSb), (q_1, bSa), (q_1, SS), (q_1, \epsilon)\}$

Apply Rule2 to Convert each terminal of grammer into transistion function of PDA.

Terminal Transistion function

In general, for each a in T $\delta(q_1, a, a) = (q_1, \epsilon)$

a (3) $\delta(q_1, a, a) = (q_1, \epsilon)$

b (4) $\delta(q_1, a, a) = (q_1, \epsilon)$

include (5) $\delta(q_1, \epsilon, z) = (q_1, \epsilon)$

Therefore PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0)$ where

$Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$ $\Gamma = \{z, S, a\}$

δ - Transistion functions ① to ⑤ $q_0 = q_0$ (initial state)

$z_0 = z$ (initial stack Symbol)

δ is: (1) $\delta(q_0, \epsilon, z) = \{(q_1, z)\}$

(2) $\delta(q_1, \epsilon, S) = \{(q_1, aSb), (q_1, bSa), (q_1, SS), (q_1, \epsilon)\}$

(3) $\delta(q_1, a, a) = \{(q_1, \epsilon)\}$

(4) $\delta(q_1, b, b) = \{(q_1, \epsilon)\}$

(5) $\delta(q_1, \epsilon, z) = \{(q_1, \epsilon)\}$

② Convert the following grammer into its equivalent PDA that accepts by empty stack.

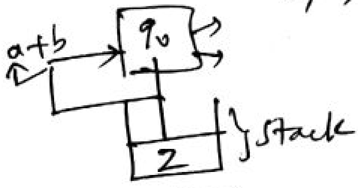
$E \to E+E \mid E*E \mid (E) \mid I$

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

Solution: Variables – E, I
start Variable – E

Terminal symbols } – a, b, 0, 1, +, *, (,)

Begin with initial state q_0 of PDA and Z on top of the stack. Push start Variable onto stack



PDA

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

Apply Rule 1 :-

Production	Transistion function of PDA
$E \to E+E \mid E*E \mid (E) \mid I$	② $\delta(q_1, \epsilon, E) = \{(q_1, E+E), (q_1, E*E), (q_1, (E)), (q_1, I)\}$
$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$	③ $\delta(q_1, \epsilon, I) = \{(q_1, a), (q_1, b), (q_1, Ia), (q_1, Ib), (q_1, I0), (q_1, I1)\}$

Apply Rule 2 :-

Terminal	Transistion function
a	④ $\delta(q_1, a, a) = \{(q_1, \epsilon)\}$
b	⑤ $\delta(q_1, b, b) = \{(q_1, \epsilon)\}$
0	⑥ $\delta(q_1, 0, 0) = \{(q_1, \epsilon)\}$
1	⑦ $\delta(q_1, 1, 1) = \{(q_1, \epsilon)\}$
+	⑧ $\delta(q_1, +, +) = \{(q_1, \epsilon)\}$
*	⑨ $\delta(q_1, *, *) = \{(q_1, \epsilon)\}$
(	⑩ $\delta(q_1, (, () = \{(q_1, \epsilon)\}$
)	⑪ $\delta(q_1,),)) = \{(q_1, \epsilon)\}$
	⑫ $\delta(q_1, \epsilon, Z) = \{(q_1, \epsilon)\}$

Exercize :

③ Convert following grammers to PDA

(I) $S \to 0AA$
$A \to 0S \mid 1S \mid 0$

(II) $S \to S0S1S0S \mid S0S0S1S \mid S1S0S0S \mid \epsilon$

(III) $S \to aABC$
$A \to aB \mid a$
$B \to bAB \mid \epsilon$
$C \to a$

Exercize 3. (i) Convert following grammer to PDA.

$$S \to 0AA$$
$$A \to 0S \mid 1S \mid 0$$

- Begin with q_0 (initiate state) of PDA
- Push S onto stack ① $\delta(q_0, \epsilon, Z) = (q_1, SZ)$
- Apply Rule 1 :

Production	Transistion Function (δ)
$S \to 0AA$	② $\delta(q_1, \epsilon, S) = \{(q_1, 0AA)\}$
$A \to 0S \mid 1S \mid 0$	③ $\delta(q_1, \epsilon, A) = \{(q_1, 0S),$ $(q_1, 1S),$ $(q_1, 0)\}$

- Apply Rule 2 :

Terminal Symbol	Transistion function (δ)
0	④ $\delta(q_1, 0, 0) = (q_1, \epsilon)$
1	⑤ $\delta(q_1, 1, 1) = (q_1, \epsilon)$

- include the transistion function ⑥ $\delta(q_1, \epsilon, Z) = (q_1, \epsilon)$.

$$\therefore \text{PDA } P = (\underbrace{\{q_0, q_1\}}_{Q}, \underbrace{\{0, 1\}}_{\Sigma}, \underbrace{\{Z, S, A, 0, 1\}}_{\Gamma}, \delta, q_0, \underbrace{Z}_{Z_0})$$

Exercize 3 (ii) : Convert following grammer into its equivalent PDA

$$S \to S0S1S0S \mid S0S0S0S \mid S1S0S0S \mid \epsilon$$

- Push S onto stack ① $\delta(q_0, \epsilon, Z) = (q_1, SZ)$
- Apply Rule 1 : ② $\delta(q_1, \epsilon, S) = \{(q_1, S0S1S0S),$ $(q_1, S0S0S0S)$ $(q_1, S1S0S0S)\}$
- Apply Rule 2 :

③ $\delta(q_1, 0, 0) = (q_1, \epsilon)$

④ $\delta(q_1, 1, 1) = (q_1, \epsilon)$

⑤ $\delta(q_1, \epsilon, Z) = (q_1, \epsilon)$

Deterministic Push Down Automata (DPDA).

Definition :- A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is said to be deterministic PDA or DPDA if and only if the following conditions are met.

1) $\delta(q, a, X)$ has atmost one member for any state q in Q, any symbol a in Σ or $a = \epsilon$ and X is in Γ.

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

For example the PDA for $L = \{a^n b^n \mid n \geq 1\}$ is a Deterministic PDA where as PDA for $L = \{ww^R \mid w \text{ in } (0+1)^*\}$ is a Non-Deterministic PDA.

Since 1. $\delta(q_0, 0, z) = \overbrace{(q_0, 0z)}^{\text{Nonempty}}$

2. $\delta(q_1, 1, z) = (q_1, 1z)$

⋮

7 $\delta(q_0, \epsilon, z) = \overbrace{(q_1, z)}^{\text{non-empty}}$

8. $\delta(q_0, \epsilon, 0) = (q_1, 0)$

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

Design PDA for the following language
$L = \{ wcw^R : w \in \{a, b\}^* \}$

method: Read symbols of string w and push the equivalent symbols onto the stack. Read character c (middle of string) and enter state q_1. Now ~~Com~~ Read symbols of w^R with symbols on top of the stack. If they match pop the top symbol of the stack. After wcw^R string is read enter Final state or empty the stack.

example of string wcw^R : $\underbrace{abbab}_{w}c\underbrace{babba}_{w^R}$

Note: w^R is the reverse of string w.

Current State, Current Symbol, top Symbol of Stack, Next State, operation performed.

(1) $\delta(q_0, a, z) = (q_0, az)$ Push a

(2) $\delta(q_0, b, z) = (q_0, bz)$ Push b

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

(4) $\delta(q_0, b, b) = (q_0, bb)$ Push b

(5) $\delta(q_0, a, b) = (q_0, ab)$ Push a

(6) $\delta(q_0, b, a) = (q_0, ba)$ Push b

(7) $\delta(q_0, c, b) = (q_1, b)$ enter state q_1

(8) $\delta(q_0, c, a) = (q_1, a)$ enter state q_1

(9) $\delta(q_1, a, a) = (q_1, \epsilon)$ Pop a

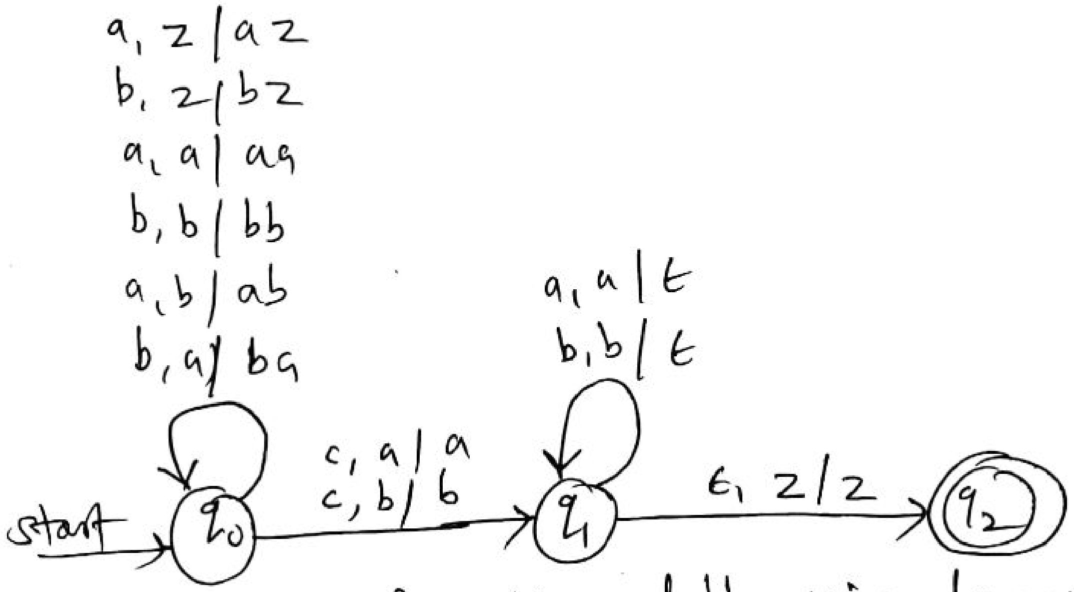
(10) $\delta(q_1, b, b) = (q_1, \epsilon)$ Pop b

(11) $\delta(q_1, \epsilon, z) = (q_2, z)$ enter state q_2

or ~~or~~

(11) $\delta(q_1, \epsilon, z) = (q_1, \epsilon)$ Pop z.

Transition Diagram:



Design PDA for the following language

$L = \{ a^n b^{2n} \mid n \geq 1 \}$

example of string w = aabbbb

method: Read a and push two a's onto the stack. Read b and if a is top symbol of stack, Pop a. Repeat this for every symbol b and every top symbol a on stack. Finally after reading string w enter final state

(1) $\delta(q_0, a, z) = (q_0, aa)$ Push aa

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

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

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

(5) $\delta(q_1, \epsilon, z) = (q_2, z)$ enter state q_2

Deterministic PDA :- (DPDA).

Definition: A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is said to be deterministic PDA or DPDA if and only if the following conditions are met.

(1) $\delta(q, a, X)$ has atmost one member for any state q in Q, any symbol a in Σ and X in Γ.

(2) If $\delta(q, a, X)$ has one member for a in Σ then $\delta(q, \epsilon, X)$ don't have any member.

example.

$$\delta(q, a, X) = \overbrace{(P, YZ)}^{\text{exactly one member,}}$$

or

$$(P, \epsilon)$$

$\delta(q, \epsilon, X)$ — don't have any member (empty)

Non-Deterministic PDA :-

Definition: A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is said to be Non-Deterministic PDA iff following conditions are met

(1) $\delta(q, a, X)$ has two members for any state q in Q, symbol a in Σ and X in Γ

(2) if $\delta(q_1, a, X)$ has one member for any a in Σ then $\delta(q, \epsilon, X)$ also have one member.

For example! The PDA for the language $L = \{ ww^R : w \in \{a, b\}^* \}$ is Non-Deterministic

Since $\delta(q_0, a, X) = (q_0, aX)$
and
$\delta(q_0, \epsilon, X) = (q_1, X)$