Unit-4

Pushdown Automata (PDA)

→ A pushdown automata is a type of automata that extends a finite automata by adding a stack, allowing it to recognize a wider class of languages, specifically context free languages, which is more powerful than regular languages.

→ A pushdown automata (PDA) is finite automata supplemented with a storage device, specifically a stack onto which we can push symbols. The transition in the pushdown automata read input symbols, just as a finite automaton does.

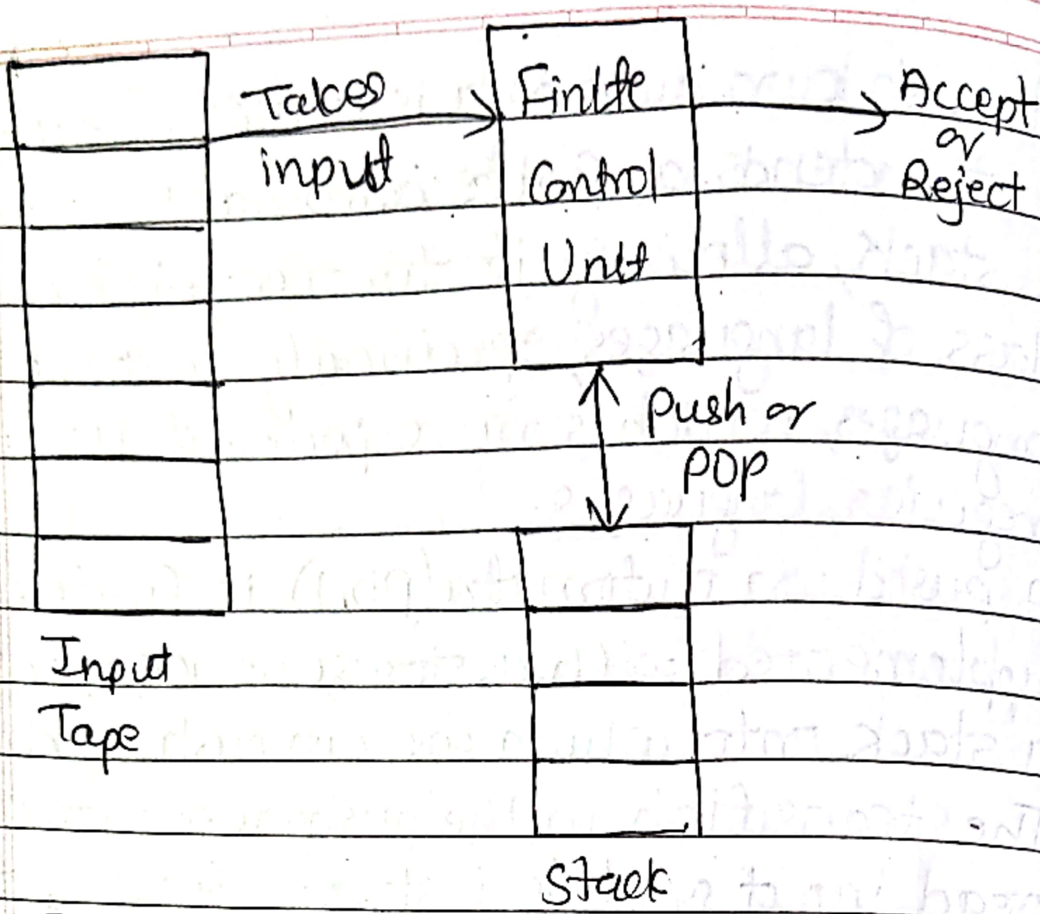
→ A pushdown automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar:

- It is more powerful than Finite State Machine.
- FSM has a very limited memory but PDA has more memory.
- PDA = FSM + stack

→ A pushdown automata has 3 components:

1. An input tape
2. A Finite Control Unit
3. A stack with infinite size

HK107



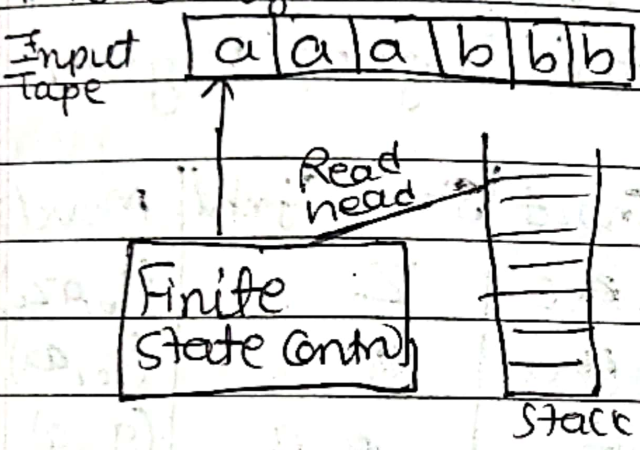
✡ Comparision between PDA and DFA

PDA	DFA
(i) PDA stands for Pushdown Automata.	(i) DFA stands for Deterministic Finite Automata.
(ii) PDA can recognize all CFL because it has stack	(ii) DFA cannot recognize all CFL because it has finite memory.
(iii) It has 5-tuple i.e. $M = (Q, \Sigma, \delta, q_0, A)$.	(iii) It has 7 tuple i.e. $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, A)$
(iv) It's transition function $\delta \rightarrow Q \times \Sigma \rightarrow Q$	(iv) It's transition function $Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow Q \times \Gamma^*$
(v) pushdown automata is used for Type-2 Grammar	(V) DFA is used for Type-3 grammar.

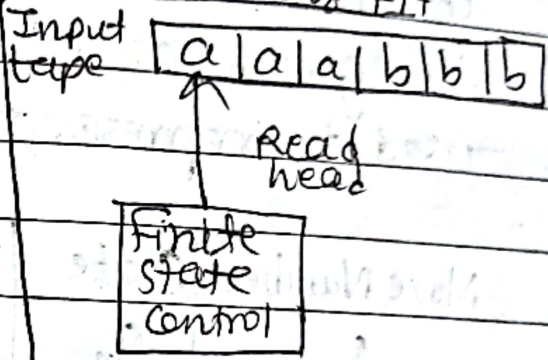
AK108

(vi) Input alphabets are accepted by going to empty stacks and final states.	(vi) Input alphabets are accepted by going to the final states.
(vii) Alphabets can be stored	(vii) Alphabets cannot be stored.

(viii) ~~PDA is used for reg~~
Model of PDA



(viii) Model of FA



The formal definition of PDA:

A PDA can be defined as 7-tuple:

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

where,

$Q \rightarrow$ Set of finite states

$\Sigma \rightarrow$ Non-empty finite set of input symbols

$\Gamma \rightarrow$ Set of stack symbols/alphabets

$\delta \rightarrow$ Transition function

$$Q\times(\Sigma\cup\epsilon)\times\Gamma \longrightarrow Q\times\Gamma^*$$

$q_0\in Q$ is the start state

$Z_0\in\Gamma$ is the initial symbol on the stack

$A\in Q$ is the set of final state

HK109

Representation of PDA

PDA can be described by using two techniques:

(i) Transition Table

(ii) Transition diagram

Transition Table

Consider a context free language defined by the grammar $a^n b^n$. Then the transition table used to represent this language is given as:

Move Number	State	Input	Stack Symbol	Move(s)
1	q_0	a	Z_0	(q_0, aZ_0)
2	q_0	a	a	(q_0, aa)
3.	q_0	ϵ	a	(q_1, a)
4	q_1	b	a	(q_1, ϵ)
5	q_1	ϵ	Z_0	(q_2, Z_0)

Here, q_0 is the starting state of PDA

q_2 is the final state " "

Z_0 is the initial stack symbol.

Description

move1 :- $\delta(q_0, a, Z_0) = (q_0, aZ_0)$

move2 :- $\delta(q_0, a, a) = (q_0, aa)$

move3 :- $\delta(q_0, \epsilon, a) = (q_1, a)$

move4 :- $\delta(q_1, b, a) = (q_1, \epsilon)$

move5 :- $\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$

H.K.119.

The transition perform by PDA is depends on:

(i) Current state

(ii) Next input symbol

(iii) The symbol on top of the stack

✡ Instantaneous Description (ID)

The current configuration of PDA at any given instant can be described by an instantaneous description (ID). Id gives the current state of the PDA, the remaining string to be processed and the entire contents of the stack. Thus, an Id can be defined as:

let,

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, A)$ be a PDA. An ID is defined as 3-Tuple:

$$(q, \omega, \alpha)$$

where, q is the current state

ω is the string to be processed

α is the current content of stack.

If the transition defined as

HKM

$$\delta(q, a, z) = (p, \beta)$$

Then the new configuration obtained will be

$$(p, w, \beta, \alpha)$$

✡ Acceptance of language by PDA

There are two cases where string w is accepted by a PDA. They are:

(i) Get the final state with empty inputs:

- It means that when all the symbols in the string w have been read and when the machine is in final state, the final content of stack is irrelevant."

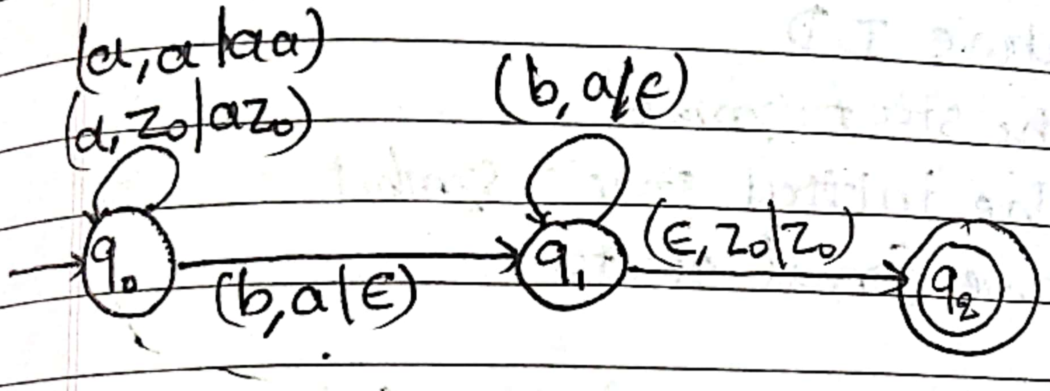
~~- Given~~

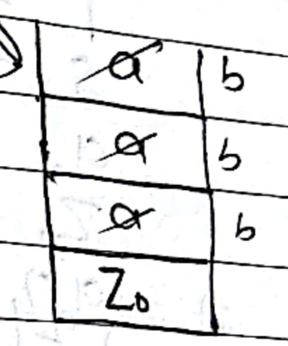
(ii) Get an empty stack from the start state:

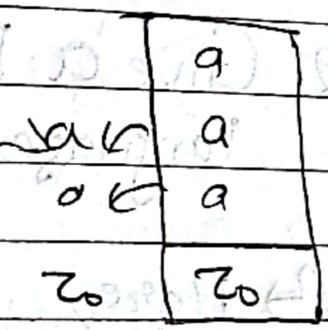
It means that when the string w is accepted by empty stack, the input should be completely read and stack should be empty.

Q.1. Design a PDA for the following language
$L = \{a^n b^n \mid n \geq 1\}$

Sol^n →
$L = \{ab, aabb, aaabbb, \ldots\}$







$(a, Z_0 | aZ_0)$

↑ input symbol ↑ Top of the stack

ID

present state, input symbol, stack → next state → operation result

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

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

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

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

$(q_1, \epsilon, Z_0) = (q_2, Z_0)$ or (q_2, ϵ)

HK113

PDA is given as $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, A)$ is given

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, Z_0\}$
δ = above ID
q_0 is the start symbol
Z_0 is the initial stack symbol
$A = \{q_2\}$ is the final state

Q.2. Give a PDA to accept the following language

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

Soln→ Here,

$$L = \{abb, aabbbb\epsilon, \text{---}$$

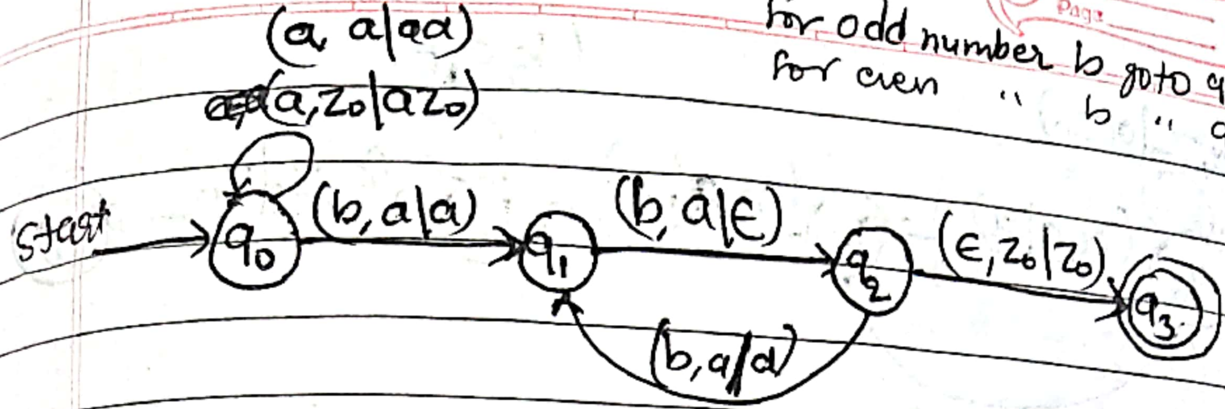
ID

a	a	b	b	b	b	∈

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, a)$
$\delta(q_1, b, a) = (q_2, \epsilon)$
$\delta(q_2, b, a) = (q_1, a)$
$\delta(q_2, \epsilon, Z_0) = (q_3, Z_0)$ or (q_3, ϵ)

a
a
Z_0

for odd number b goto q_1
for even " b " q_2



PDA is given $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, A)$ as

$Q = \{q_0, q_1, q_2, q_3\}$

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

$\Gamma =$ ~~above ID~~ $\{a, Z_0\}$.

$\delta =$ above ID

q_0 $q_0 \subseteq Q$ is the start symbol

Z_0 is the initial stack symbol

$A = \{q_3\}$ is the final state

Q3. Design a PDA to accept the language

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

soln → $L = \{aab, aaaabb, \ldots\}$

a	a	a	a	b	b	ϵ

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

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

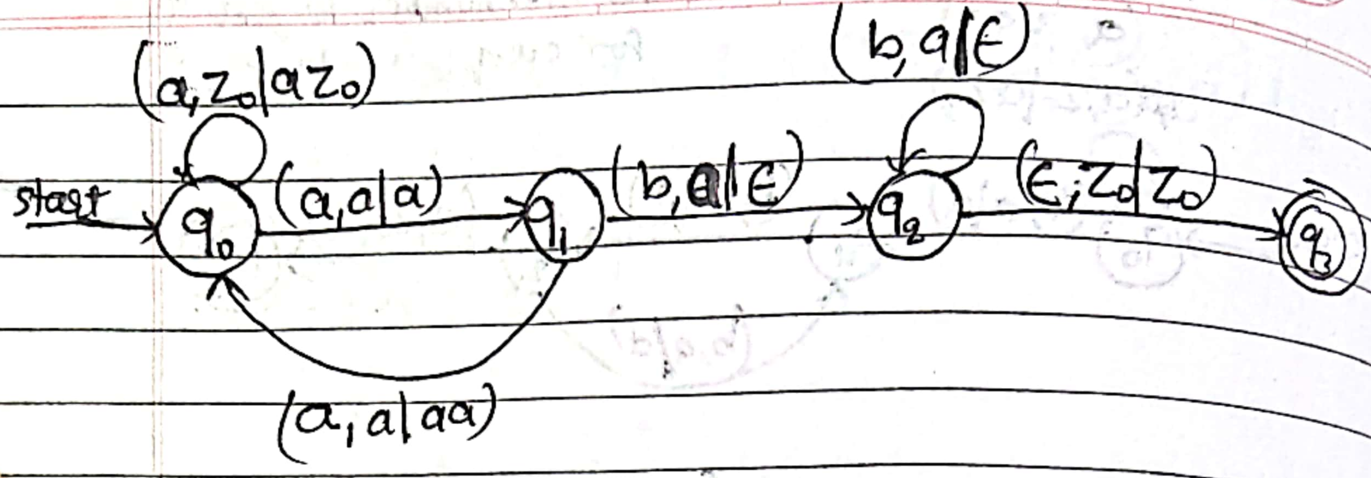
$\delta(q_1, a, a) = (q_0, aa)$

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

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

$\delta(q_2, \epsilon, Z_0) = (q_3, Z_0)$ or (q_3, ϵ)

HK115



PDA is given $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, A\}$ where

$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, Z_0\}$
δ = above ID.
$q_0 \subseteq Q$ is the start symbol.
Z_0 is the start stack
$A = \{q_3\}$ is the final state.

Q.4. Construct a PDA for $L = \{a^n b^m c^n \mid n \geq 1, m \geq 1\}$

Soluⁿ→ $L =$ abbc, aabbbcc, ...

a	a	b	b	b	c	c	ϵ

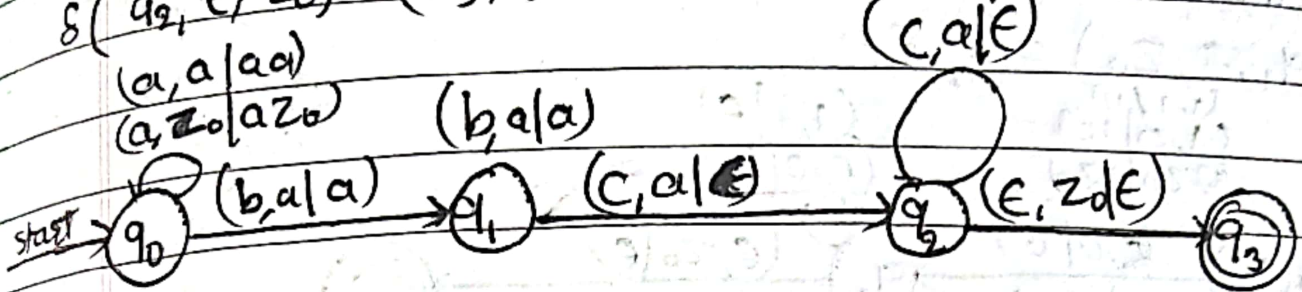
ID

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, a)$
$\delta(q_1, b, a) = (q_1, a)$
⋮ for more b

a
a
Z_0

HKIIIB

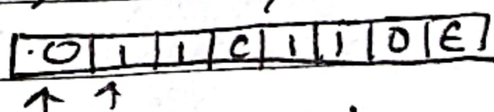
$\delta(q_1, c, a) = (q_2, \epsilon)$
$\delta(q_2, c, a) = (q_2, \epsilon)$
$\delta(q_2, \epsilon, Z_0) = (q_3, \epsilon)$

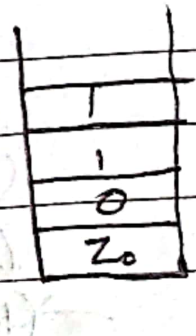


PDA $P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, A\}$ is given as

$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, Z_0\}$
$\delta =$
$q_0 \subseteq Q$ is the initial state
$Z_0 \subseteq \Gamma$ is the initial stack
$A = \{q_3\}$ is the final state

Q.5. Construct a PDA(DPDA) for $L = \{wcw^R \mid w \text{ is in } (0+1)^+\}$
Sol^n
δ
$L = 000c000, 110c011, 011c110, \ldots$





ID
$\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$
$\delta(q_0, 1, 0) = (q_0, 10)$
$\delta(q_0, 1, 1) = (q_0, 11)$
$\delta(q_0, c, 1) = (q_1, 1)$

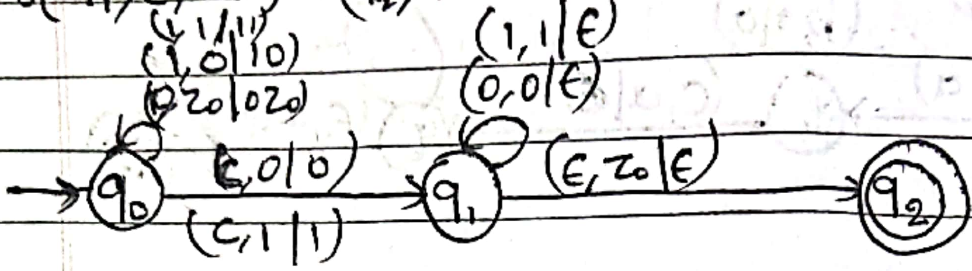
HKIT

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

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

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

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



Q.6. Construct a PDA for $L = \{0^n 1^n \mid n \geq 1\}$

Sol^n :- $L = \{01, 0011, 000111, \ldots\}$

0	0	0	1	1	1	ε

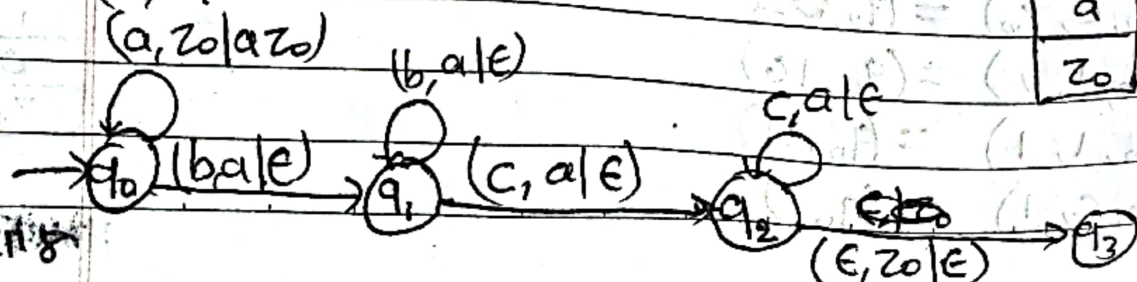
Q.7. Construct PDA for equal number of a's and equal number of b's. i.e $L = \{w \mid n_a(w) = n_b(w)\}$

Q.8. Construct PDA for $L = \{a^{m+n} b^m c^n \mid m, n \geq 1\}$

Sol^n → $L = aabc, aaabbc, \ldots$

a	a	a	b	b	c	ε

a
a
a
Z_0



HKM8

Deterministic PDA (DPDA)

A deterministic PDA is a type of Pushdown Automaton (PDA) where, for each state, input symbol, and stack symbol, there is at most one possible transition, unlike a non-deterministic PDA which can have multiple choices.

A PDA $M = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, A\}$ is deterministic

if every $q_0 \in Q$,

$a \in \Sigma \cup \epsilon$,

$b \in \Gamma$.

(i) $\delta(q_0, a, b)$ contains at most one element

(ii) if $\delta(q_0, \epsilon, b)$ is not empty, then $\delta(q_0, c, b)$ must be empty ~~for~~ empty for every $c \in \Sigma$ ~~be empty~~

Non-deterministic PDA (NPDA) :-

A Non-deterministic Pushdown Automaton is a type of automaton that, unlike deterministic pushdown automaton (DPDA), can have multiple possible transitions from a given state and input symbol allowing for a greater flexibility in recognizing Context-free languages

HIGI19

- Formally, an NPDA is defined as a tuple.

~~Q - A finite set of states~~

$$M = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, A\}$$

Where,

Q = A finite set of states

Σ = Finite set of input symbol

Γ = finite set of stack symbols

δ = Transition function:

$$Q \times \Sigma \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}$$

q_0 = The initial state

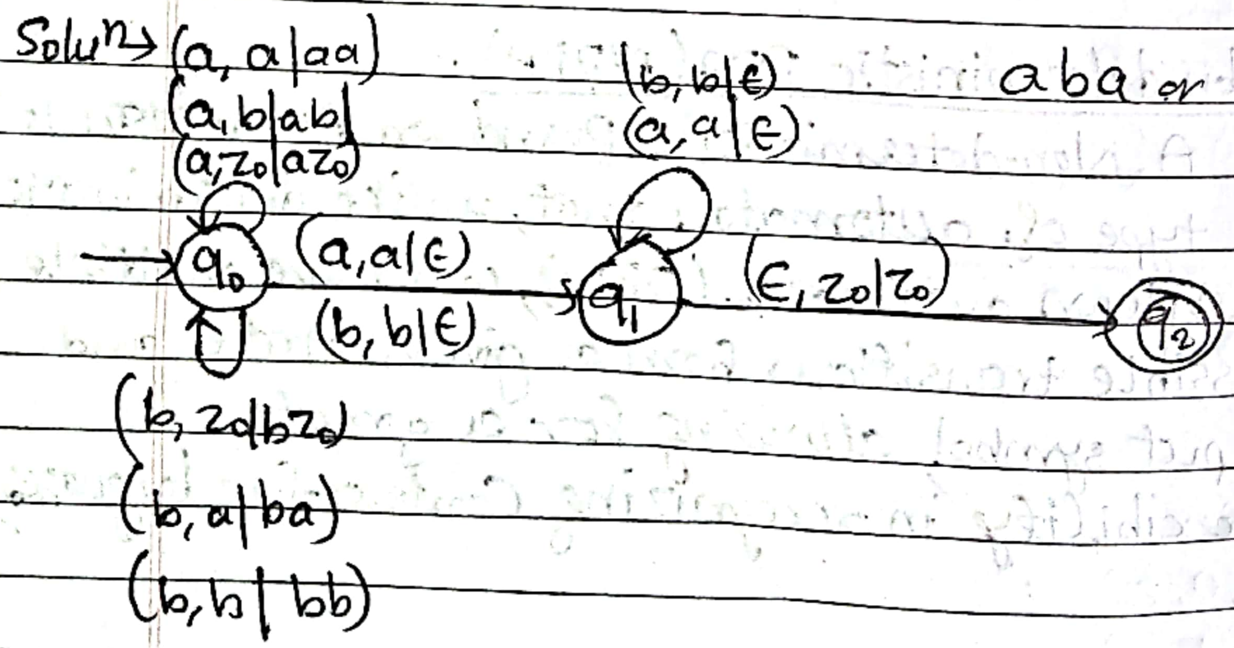
Z_0 = Initial stack symbol

A = Final state

Q. 9. Construct NPDA for $L = \{ww^R \mid w \in \{a,b\}^+\}$

Solun →

aba or bab



classmate

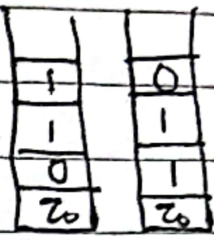
Date ____
Page ____

Q.10. Construct Non-deterministic PDA for
$L = \{ ww^R \mid w \text{ is in } (0+1)^+ \}$

Ans/Sol => L = 011110, 110011, ...

ID

011110
110011



operation 1: $\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$

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

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

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

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

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

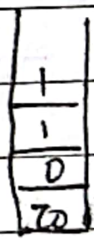
100001

operation 2: $\delta(q_0, \epsilon, 0) = (q_1, 0)$

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

011110

↑ ↑ ↑

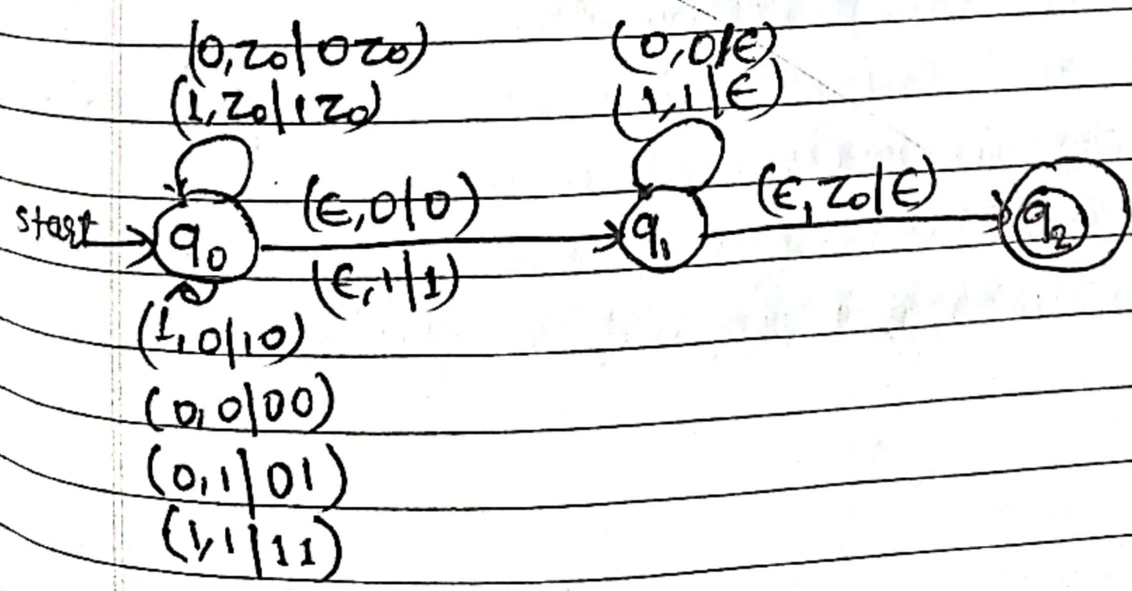


operation 3:- $\delta(q_1, 0, 0) = (q_1, \epsilon)$

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

~~δ(~~

operation 4:- $\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$



HK121

Where NPDA is given

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, A)$$

where,

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

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, z_0\}$

δ = above ID

$q_0 \in Q$ Initial state

$z_0 \in \Gamma$ Initial stack

$A = \{q_2\}$ is final state

4.2 Pushdown Automata and Context Free Grammars-

For every CFG, there is PDA that accepts it. It is easy to get PDA from CFG. This is possible only from a CFG which is in GNF. For any given Grammar first obtain the grammar in GNF and then obtain PDA.

To ~~obtain~~ construct PDA for given grammar follows the following steps:

Step1: Convert the Grammar or CFG into GNF

Step2: Let q_0 be the start state and Z_0 is the initial symbol on the stack. Without consuming any input, push the start symbol S on the stack and change the state to q_1. The transition for this is given as

$$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$$

Step3: For each production of the form $\boxed{A \to a\alpha}$ where

$$a \in T$$
$$\alpha \in (V \cup T)^*$$

Introduce the transition,

$$\delta(q_1, a, A) = (q_1, \alpha)$$

Eg.

$$S \to aABC$$
$$\delta(q_1, a, S) = (q_1, ABC)$$

1/11/23

Step 4:- Finally, at state q_1, without consuming any input change the state to q_f which is final state. The transition for this can be written as

$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$$

Q.1. Obtain the corresponding PDA for the grammar given below:

$$S \rightarrow aABC$$
$$A \rightarrow aB \mid a$$
$$B \rightarrow bA \mid b$$
$$C \rightarrow a$$

Soln → Let q_0 is the start state and Z_0 be the initial symbol on a stack. The given grammar is in GNF. Push the start symbol S on the stack and change the state to q_1. The transition for this is

$$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$$

For each production of the form $A \rightarrow a\alpha$, introduce the transition

$$\delta(q_1, a, A) = (q_1, \alpha)$$

HK124

Production	Transition function for PDA
$S \rightarrow aABC$	$\delta(q_1, a, S) = (q_1, ABC)$
$A \rightarrow aB$	$\delta(q_1, a, A) = (q_1, B)$
$A \rightarrow a$	$\delta(q_1, a, A) = (q_1, \epsilon)$
$B \rightarrow bA$	$\delta(q_1, b, B) = (q_1, A)$
$B \rightarrow b$	$\delta(q_1, b, B) = (q_1, \epsilon)$
$C \rightarrow a$	$\delta(q_1, a, C) = (q_1, \epsilon)$

Finally in state q_1, without consuming any input change the state to q_f. The transition for this is

$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$$

Hence, the PDA is given by

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

where,

$$Q = \{q_0, q_1, q_f\}$$
$$\Sigma = (a, b)$$
$$\Gamma = \{S, A, B, C, Z_0\}$$
$$\delta = \{$$
$$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$$
$$\delta(q_1, a, S) = (q_1, ABC)$$
$$\delta(q_1, a, A) = (q_1, B)$$
$$\delta(q_1, a, A) = (q_1, \epsilon)$$
$$\delta(q_1, b, B) = (q_1, A)$$
$$\delta(q_1, b, B) = (q_1, \epsilon)$$
$$\delta(q_1, a, C) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$$

MK125

$q_0 \in Q$ is the initial state
$Z_0 \subseteq \Gamma$ is the initial stack
$A \subseteq Q$ is the final state. $\{q_f\}$

Q.2. Convert the following grammar to a PDA that accepts the same language.

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

Soln: The CFG can be first simplified by eliminating unit products

$S \rightarrow 0S1 \mid 1S0 \mid \epsilon$

Now, we will convert this CFG to GNF:

$S \rightarrow 0SX \mid 1SY \mid \epsilon$
$X \rightarrow 1$
$Y \rightarrow 0$

Production	Transition function for PDA
$S \rightarrow 0SX$	$\delta(q_1, 0, S) = (q_1, SX)$
$S \rightarrow 1SY$	$\delta(q, 1, S) = (q_1, SY)$
$S \rightarrow \epsilon$	$\delta(q_1, \epsilon, S) = (q_1, \epsilon)$
$X \rightarrow 1$	$\delta(q_1, 1, X) = (q_1, \epsilon)$
$Y \rightarrow 0$	$\delta(q_1, 0, Y) = (q_1, \epsilon)$

HK126

Finally in state q_1, without consuming any input change the state to q_f. The transition for this is

$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$$

Hence, the PDA is given by

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

where,

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

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

$\Gamma = \{S, X, Y, Z_0\}$

$\delta = \{$

$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

$\delta(q_1, 0, S) = (q_1, SX)$

$\delta(q_1, 1, S) = (q_1, SY)$

$\delta(q_1, \epsilon, S) = (q_1, \epsilon)$

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

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

$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$

$\}$

q_0 initial state

Z_0 initial stack

A final state $\{q_f\}$

Q.3. Construct PDA for the given CFG, ~~and test~~ ~~whether~~

$S \rightarrow 0BB$

$B \rightarrow 0S | 1S | 0$

HK127

Q. 4. Draw a PDA for the CFG given below

$S \to aSb$

$S \to a | b | \epsilon$

HK-128