AKTU

B. Tech II-Year

Gateway Classes

GATEWAY CLASSES

Semester -IV

CS IT & Allied Branches

BCS402 Theory of Automata and Formal Languages

UNIT-4 Push Down Automata & Properties of Context

Gateway Series for Engineering

- Topic Wise Entire Syllabus
- Long - Short Questions Covered
- AKTU PYQs Covered

AKTU

B. Tech II-Year

Gateway Classes

GATEWAY CLASSES

BCS402 Theory of Automata and Formal Languages

Unit-4

Introduction to Push Down Automata and Properties

Syallbus

Push Down Automata and Properties of Context Free Languages: Nondeterministic Pushdown Automata (NPDA)- Definition, Moves, A Language Accepted by NPDA, Deterministic Pushdown Automata(DPDA) and Deterministic Context free Languages(DCFL), Pushdown Automata for Context Free Languages, Context Free grammars for Pushdown Automata, Two stack Pushdown Automata, Pumping Lemma for CFL, Closure properties of CFL, Decision Problems of CFL, Programming problems based on the properties of CFLs.

AKTU PYQ's

FOR PDF NOTES DOWNLOAD APP

LINK IN DISCRIPTION

Pushdown Automata

CFL

RL

RL is a CFL subset

a a a b

$a^n b^n$ $n \geq 1$

CFG

CFL

(Deterministic CFL)

DCFL

for some

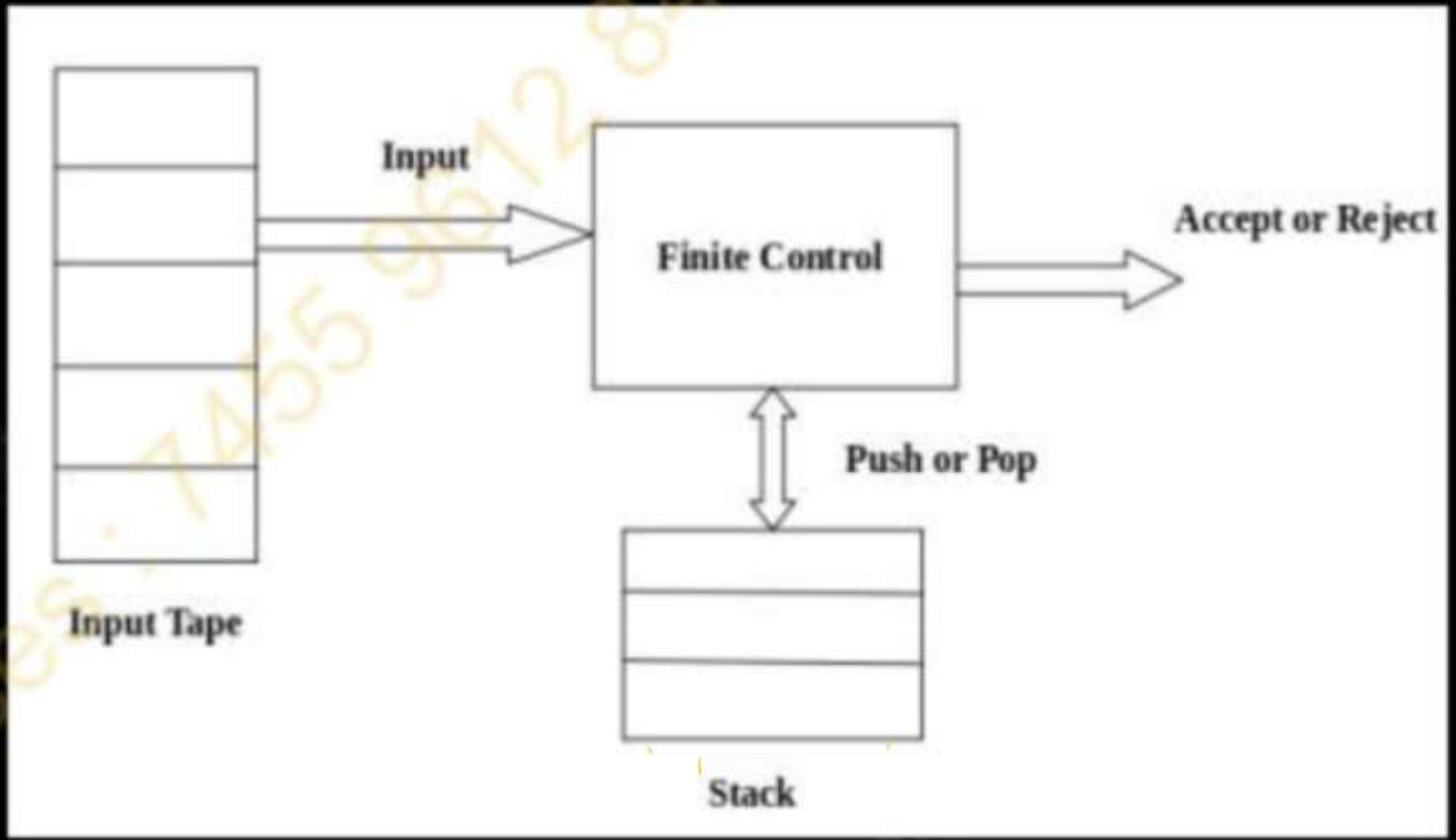
Pushdown Automata (Acceptor)

for all

P(DPDA) < P(NPDA)

Deterministic Pushdown Automata

(Non-Deterministic Pushdown Automata)



- Pushdown Automata is a finite automata with extra memory called stack It is used to recognize Context Free Languages

Pushdown Automata (AKTU)

Push down automata can be of two types-

- deterministic Pushdown automata if, there is only one move from a state on an input symbol and stack symbol.
- The non-deterministic pushdown automata can have more than one move from a state on an input symbol and stack symbol
- It is not always possible to convert non-deterministic pushdown automata to deterministic pushdown automata.

- The expressive power of non-deterministic PDA is more as compared to expressive deterministic PDA as some languages are accepted by NPDA but not by deterministic PDA
- The pushdown automata can either be implemented using acceptance by empty stack or acceptance by final state and one can be converted to another
- Deterministic finite automata(DFA) and Non-deterministic finite automata(NFA) have same power
- A PDA is more powerful than FA. Any language which can be acceptable by FA can also be acceptable by PDA.
- PDA also accepts a class of language which even cannot be accepted by FA.

Pushdown Automata

A Pushdown Automata (PDA) can be defined as

Q is the set of states (1)

Σ is the set of input symbols (2)

Γ is the set of pushdown symbols (which can be pushed and popped from stack) (3)

q_0 is the initial state (4)

Z_0 is the initial pushdown symbol (which is initially present in stack) (5)

F is the set of final states (6)

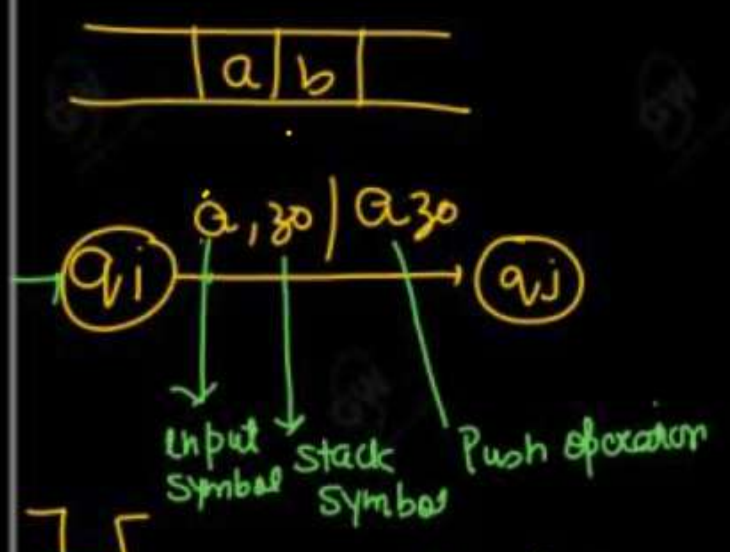
- δ is a transition function which maps Q x {Σ ∪ ∈} x Γ into Q x Γ*. (DPDA) (7)
- δ is a transition function which maps Q x {Σ ∪ ∈} x Γ into $2^{Q \times \Gamma^*}$. (NPDA) (8)

Explain

PDA	NPDA	DPDA
1–8	1–6 + 8	1–6 + 7

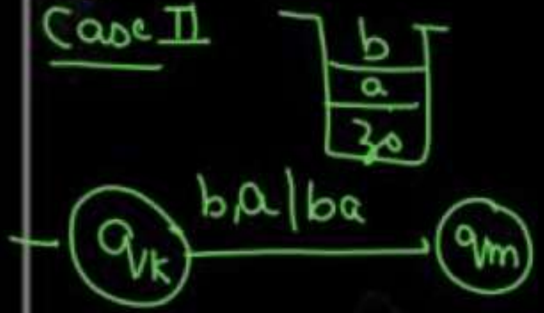
Basic operation on stack

➢ PUSH OPERATION



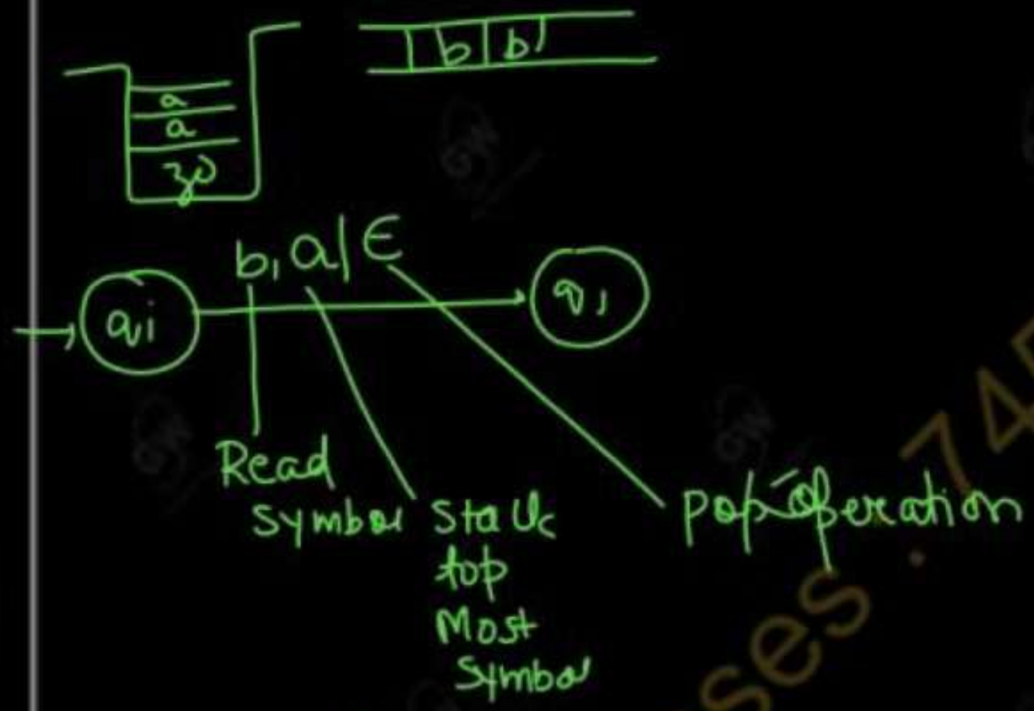
$\delta(q_i, a, z_0) = (q_j, a z_0)$

Case II



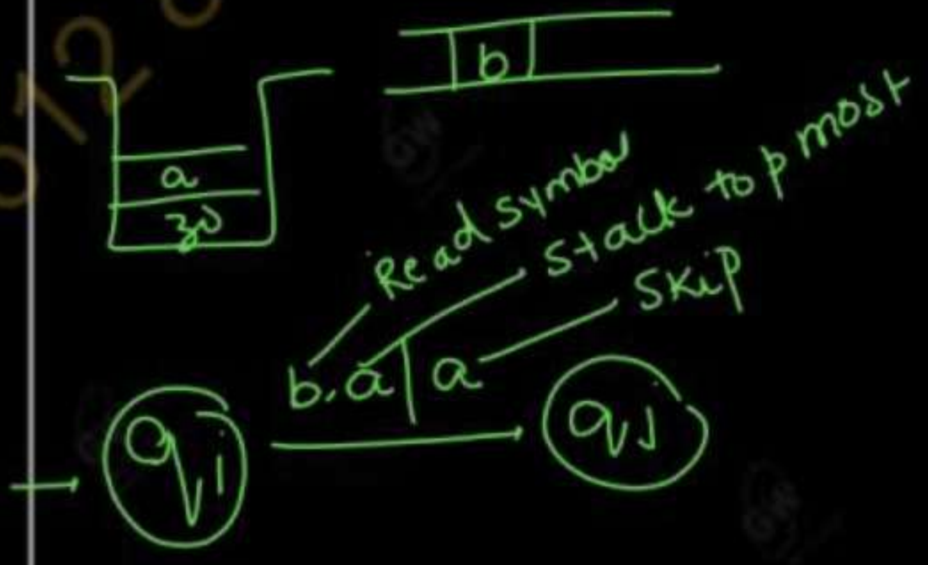
$\delta(q_k, b, a) = (q_m, ba)$

POP OPERATION



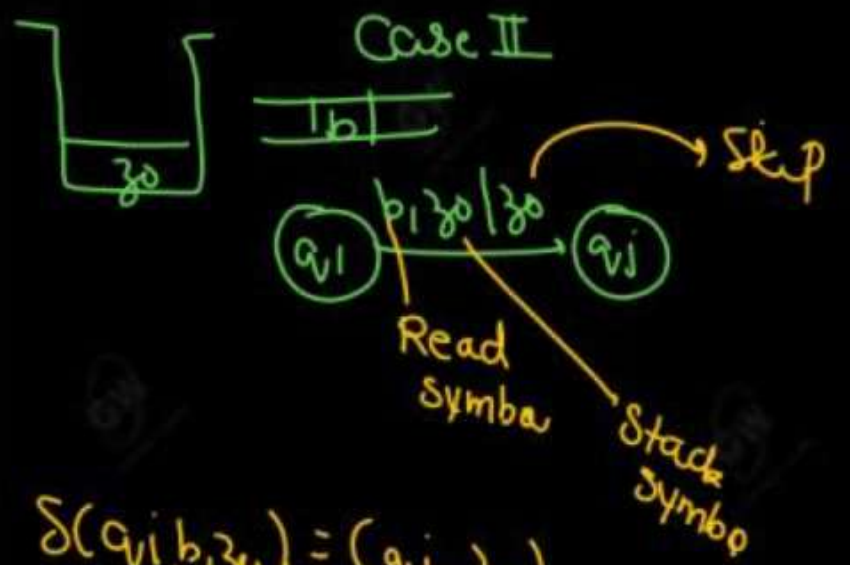
$\delta(q_i, b, a) = (q_j, \epsilon)$

SKIP OPERATION



$\delta(q_i, b, a) = (q_j, a)$

Case II



$\delta(q_i, b, z_0) = (q_j, z_0)$

Pushdown automata	finite automata
For Type-2 grammar we can design pushdown automata. CFG	For Type-3 grammar we can design finite automata. Regular Grammar
Non – Deterministic pushdown automata has more powerful than Deterministic pushdown automata.	Non-Deterministic Finite Automata has same powers as in Deterministic Finite Automata.
Not every Non-Deterministic pushdown automata is transformed into its equivalent Deterministic pushdown Automata . CFL RL	Every Non-Deterministic Finite Automata is transformed into an equivalent Deterministic Finite Automata
Context free languages can be recognized by pushdown automata.	Regular languages can be recognized by finite automata.
Pushdown automata has the additional stack for storing long sequence of alphabets.	Finite Automata doesn't has any space to store input alphabets.
It gives acceptance of input alphabets by going up to empty stack and final states.	It accepts the input alphabets by going up to final states.

Condition for deterministic Pushdown Automata

M is *deterministic* if it satisfies both the following conditions:

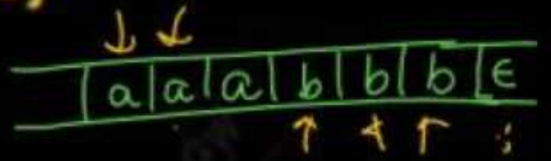
- For any $q \in Q, a \in \Sigma \cup \{\varepsilon\}, x \in \Gamma$, the set $\delta(q,a,x)$ has at most one element.
- For any $q \in Q, x \in \Gamma$, if $\delta(q,\varepsilon,x) \neq \emptyset$, then $\delta(q,a,x) = \emptyset$ for every $a \in \Sigma$.

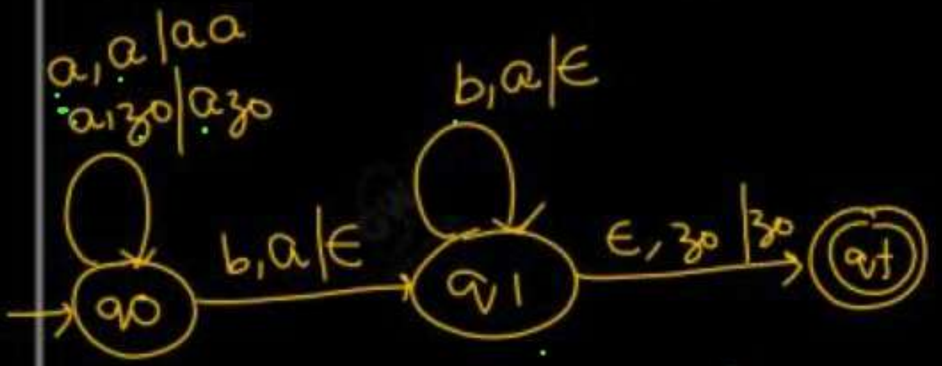
Only one operation must be performed either push, pop, skip on a particular state, Reading a particular from a tape & stack

Pushdown Automata

➢ $L=\{a^n b^n \; n>=1\}$

(DPDA)





$\delta(q_0, a, z_0) = (q_0, a z_0)$
$\delta(q_0, a, a) = (q_0, aa)$] Push

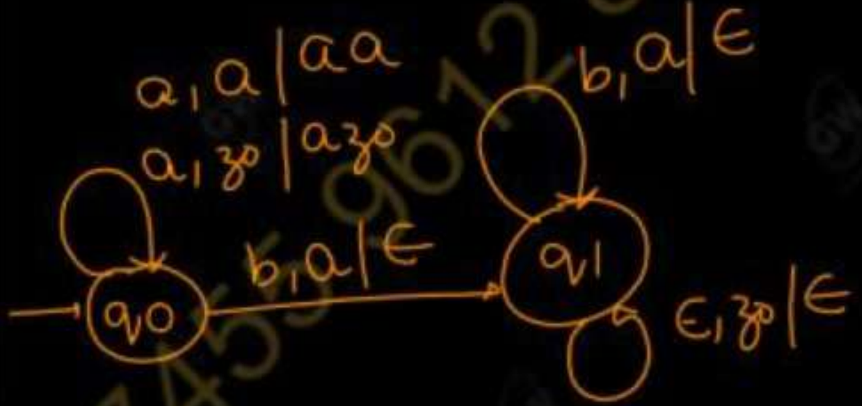
$\delta(q_0, b, a) = (q_1, \epsilon)$
$\delta(q_1, b, a) = (q_1, \epsilon)$] Pop

$\delta(q_1, \epsilon, z_0) = (q_f, z_0)$ — Stop

Acceptance by final state

$Q = \{q_0, q_1, q_f\}$
$q_0 = \{q_0\}$
$F = \{q_f\}$ $\quad Z_0 = \{z_0\}$
$\Sigma = \{a, b\}$
$\Gamma = \{z_0, a\}$
δ

Acceptance by empty stack



(Minimum state use)

$\delta(q_0, a, z_0) = (q_0, a z_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \epsilon)$
$\delta(q_1, b, a) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$

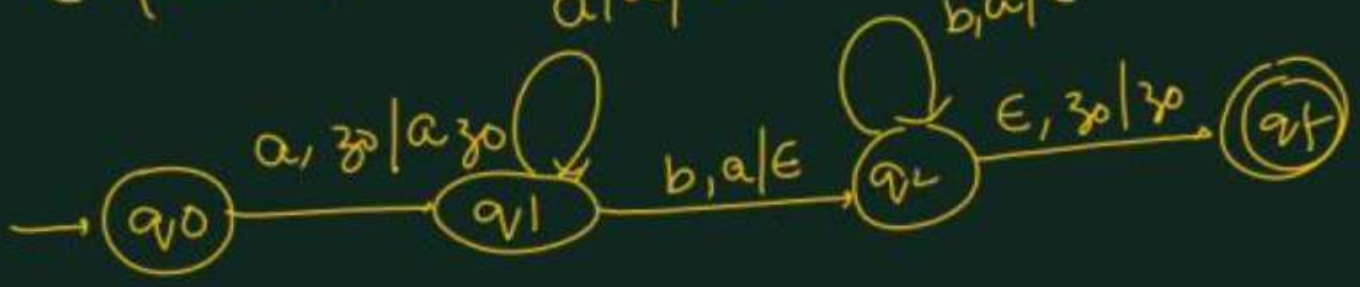
$Q = \{q_0, q_1\}$
$\Sigma = \{b, a\}$
$\Gamma = \{z_0, a\}$
$Z_0 = \{z_0\}$
$q_0 = \{q_0\}$
$F = \{\phi\}$
δ

Second Method

DPDA

$\overline{a|a|b|b|\epsilon}$

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



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

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

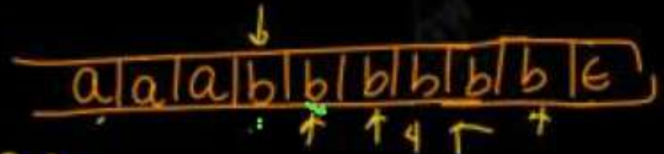
$\delta(q_1, b, a) = (q_2, \epsilon)$

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

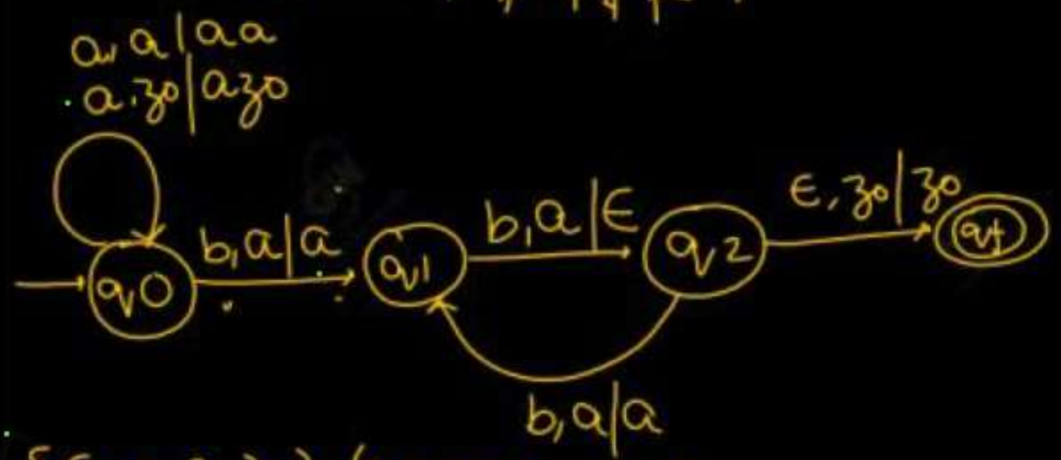
$\delta(q_2, \epsilon, z_0) = (q_f, z_0)$

➤ $L=\{a^n b^{2n}\ n>=1\}$

DPDA



Acceptance by final state



$\delta(q0, a, z_0) = (q0, az_0)$ — 1

$\delta(q0, a, a) = (q0, aa)$ — 2

$\delta(q0, b, a) = (q1, a)$ — 3

$\delta(q1, b, a) = (q2, \epsilon)$ — 4

$\delta(q2, b, a) = (q1, a)$ — 5

$\delta(q2, \epsilon, z_0) = (qf, z_0)$ — 6

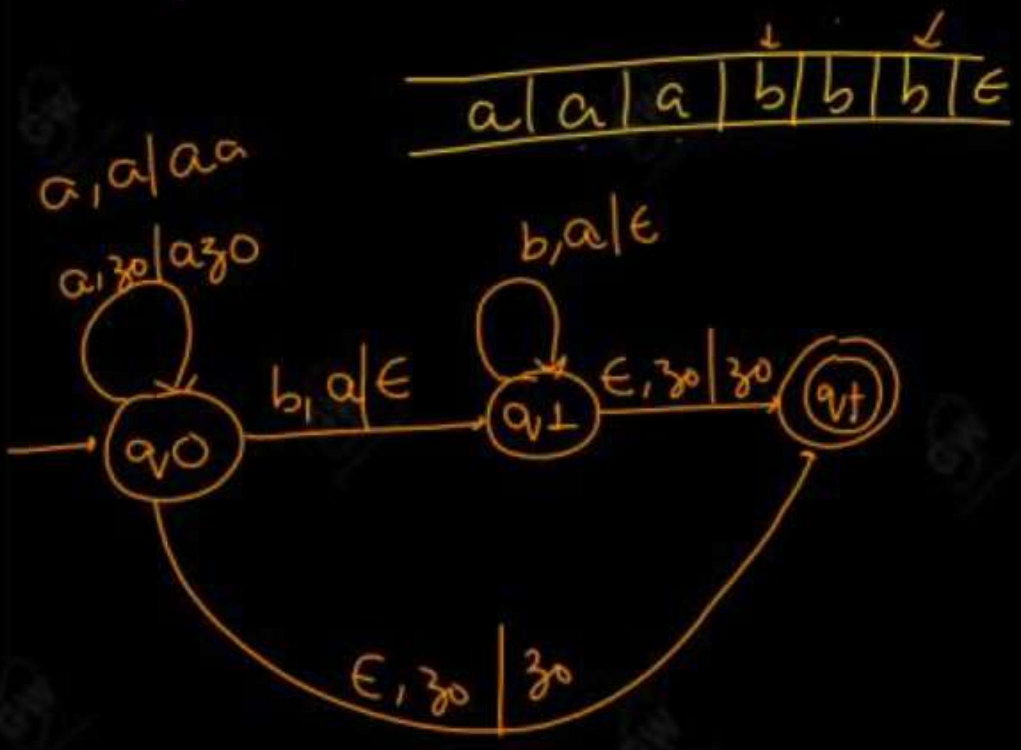
Acceptance by empty stack

1–5 Same

6 $\delta(q2, \epsilon, z_0) = (q2, \epsilon)$

- $L=\{a^n b^n \ n>=0\}$

(NPDA)



Acceptance by final state

$\delta(q_0,a,z_0)=(q_0,az_0)$
$\delta(q_0,a,a)=(q_0,aa)$
$\delta(q_0,b,a)=(q_1,\epsilon)$
$\delta(q_1,b,a)=(q_1,\epsilon)$
$\delta(q_1,\epsilon,z_0)=(q_f,z_0)$

$\delta(q_0,\epsilon,z_0)=(q_f,z_0)$

Acceptance by empty stack.

$Q=\{q_0,q_1\}$
$\Gamma=\{a,z_0\}$
$q_0=\{q_0\}$
$F=\{\phi\}$
δ
$\Sigma=\{a,b\}$

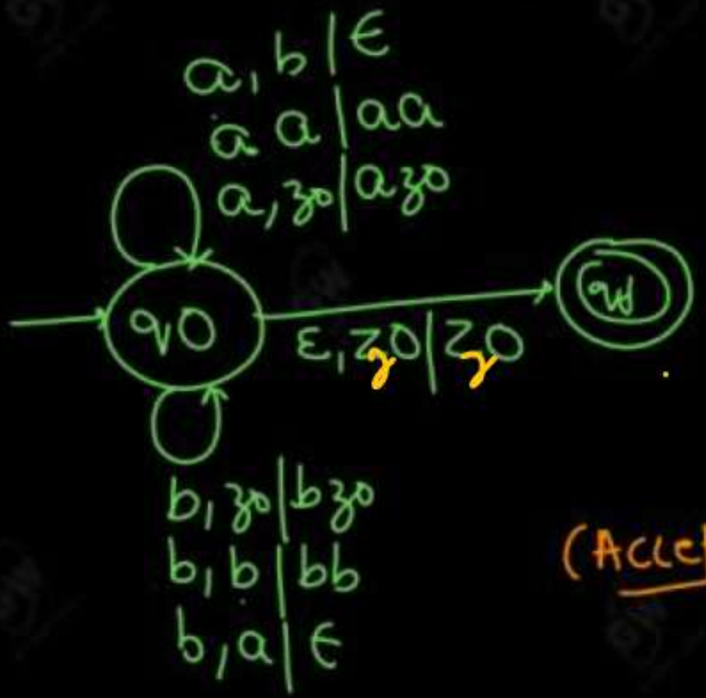
$\delta(q_0,a,z_0)=(q_0,az_0)$
$\delta(q_0,a,a)=(q_0,aa)$
$\delta(q_0,b,a)=(q_1,\varepsilon)$
$\delta(q_0,\varepsilon,z_0)=(q_1,z_0)$
$\delta(q_1,b,a)=(q_1,\varepsilon)$
$\delta(q_1,\varepsilon,z_0)=(q_1,\varepsilon)$

Pushdown Automata (AKTU PYQ)

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

➤ $L = \{w \mid n_a(w) = n_b(w)\}$

(NPDA)



$$\text{Push}\begin{cases}\delta(q_0, a, z_0) = (q_0, az_0)\\ \delta(q_0, a, a) = (q_0, aa)\end{cases}$$

$$\text{pop} \leftarrow \delta(q_0, a, b) = (q_0, \epsilon)$$

$$\text{Push}\begin{cases}\delta(q_0, b, z_0) = (q_0, bz_0)\\ \delta(q_0, b, b) = (q_0, bb)\end{cases}$$

$$\text{pop} \leftarrow \delta(q_0, b, a) = (q_0, \epsilon)$$

$$\text{Skip} \leftarrow \delta(q_0, \varepsilon, z_0) = (q_f, z_0)$$

(Acceptance by final state)

bb aa

aabb

baba

abab

$\boxed{\text{aabbababbababbaa}}$

Acceptance by empty stack

$$\text{Push}\begin{cases}\delta(q_0, a, z_0) = (q_0, az_0)\\ \delta(q_0, a, a) = (q_0, aa)\end{cases}$$

$$\text{pop} \leftarrow \delta(q_0, a, b) = (q_0, \epsilon)$$

$$\text{Push}\begin{cases}\delta(q_0, b, z_0) = (q_0, bz_0)\\ \delta(q_0, b, b) = (q_0, bb)\end{cases}$$

$$\text{pop} \leftarrow \delta(q_0, b, a) = (q_0, \epsilon)$$

$$\text{pop} \leftarrow \delta(q_0, \epsilon, z_0) = (q_0, \epsilon)$$

Power of acceptance by final state equal to power of acceptance by empty stack

Pushdown Automata (M. Imp) AKTU PYQ

{a,b}⁺ any comb of a+

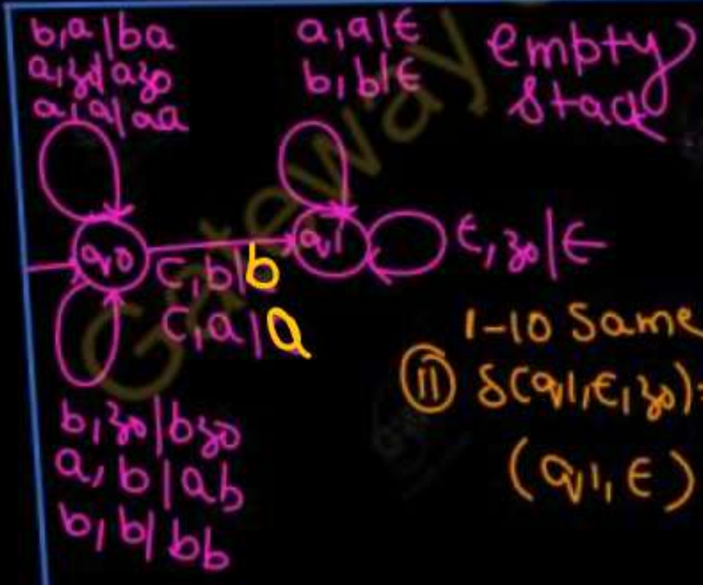
➢L={WCWʳ} W belong{a,b}

Acceptance by final state

① $\delta(q_0, a, z_0) = (q_0, az_0)$
② $\delta(q_0, a, a) = (q_0, aa)$
③ $\delta(q_0, b, a) = (q_0, ba)$
④ $\delta(q_0, b, z_0) = (q_0, bz_0)$
⑤ $\delta(q_0, a, b) = (q_0, ab)$
⑥ $\delta(q_0, b, b) = (q_0, bb)$
⑦ $\delta(q_0, c, b) = (q_1, b)$
⑧ $\delta(q_0, c, a) = (q_1, a)$
⑨ $\delta(q_1, a, a) = (q_1, \epsilon)$
⑩ $\delta(q_1, b, b) = (q_1, \epsilon)$
⑪ $\delta(q_1, \epsilon, z_0) = (q_f, z_0)$

DPDA

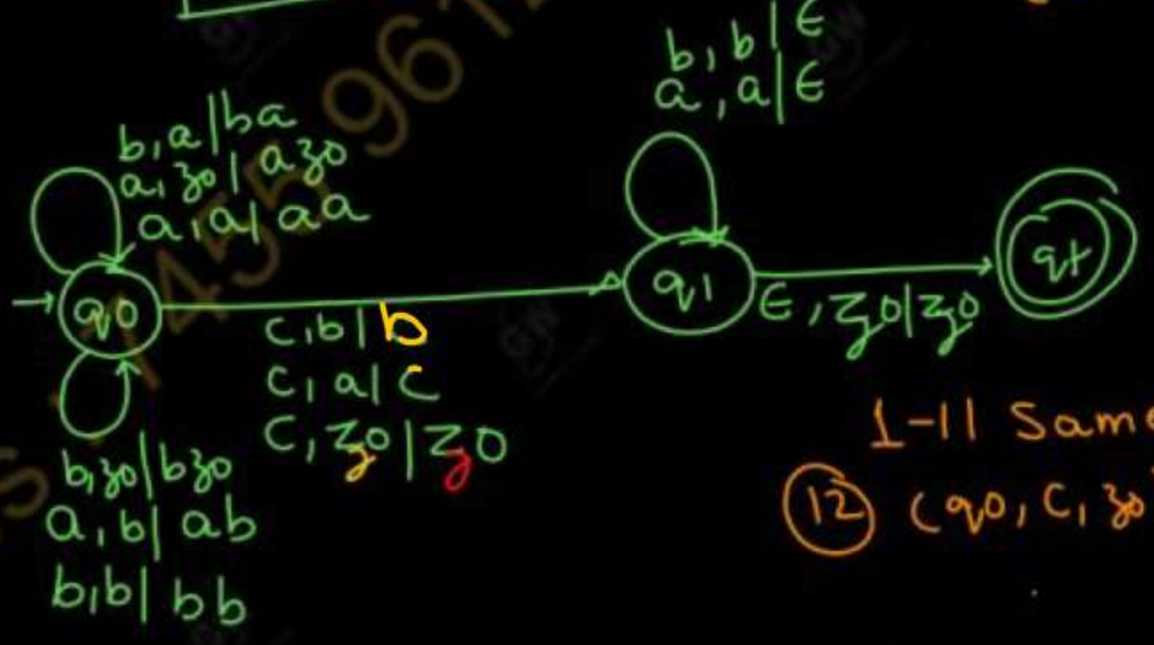
WCWʳ
aabbCbbaaε
bbaaCaabb
ababc baba
babaCabc



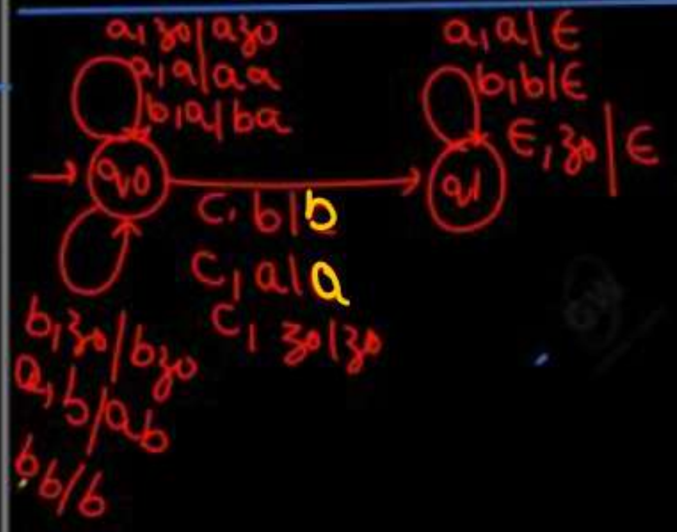
1-10 Same
⑪ $\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$

WCWʳ W belong{a,b}*

(DPDA)



1-11 Same
⑫ $(q_0, c, z_0) = (q_1, z_0)$

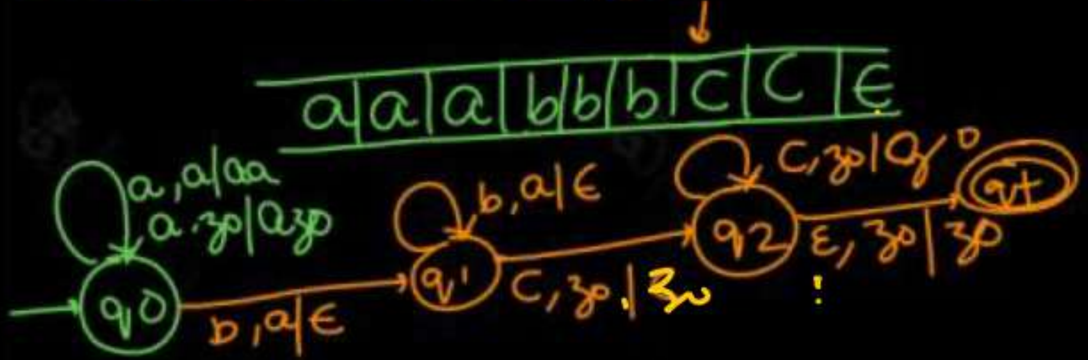


Acceptance by empty stack
(1-10) Same
11 $\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$
⑫ $(q_0, c, z_0) = (q_1, z_0)$

Pushdown Automata

GW

- $L=\{a^n b^n c^m \; n,m>=1\}$



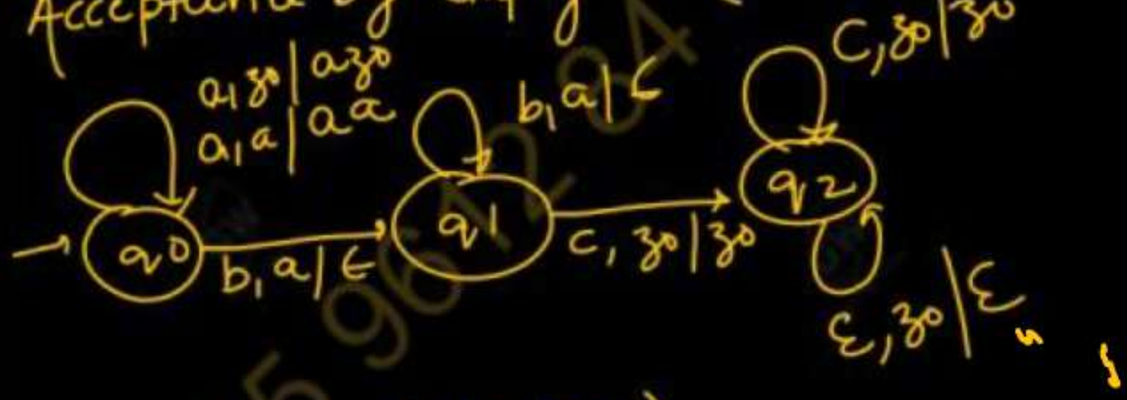
$\delta(q0, a, z_0) = (q0, az_0)$
$\delta(q0, a, a) = (q0, aa)$
$\delta(q0, b, a) = (q1, \epsilon)$
$\delta(q1, b, a) = (q1, \epsilon)$
$\delta(q1, c, z_0) = (q2, z_0)$

$\delta(q2, c, z_0) = (q2, z_0)$
$\delta(q2, \epsilon, z_0) = (qf, z_0)$

(NPDA)

(Acceptance by final state)

Acceptance by empty stack



$\delta(q0, a, z_0) = (q0, az_0)$
$\delta(q0, a, a) = (q0, aa)$
$\delta(q0, b, a) = (q1, \epsilon)$
$\delta(q1, b, a) = (q1, \epsilon)$
$\delta(q1, c, z_0) = (q2, z_0)$
$\delta(q2, c, z_0) = (q2, z_0)$
$\delta(q2, \epsilon, z_0) = (q2, \epsilon)$

- $L=\{a^n b^m c^n \mid n,m>=1\}$

a	a	a	b	b	c	c	c	ε

DPDA

$\delta(q_0, a, z_0) = (q_0, az_0)$ — 1

$\delta(q_0, a, a) = (q_0, aa)$ — 2

$\delta(q_0, b, a) = (q_1, a)$ — 3

$\delta(q_1, b, a) = (q_1, a)$ — 4

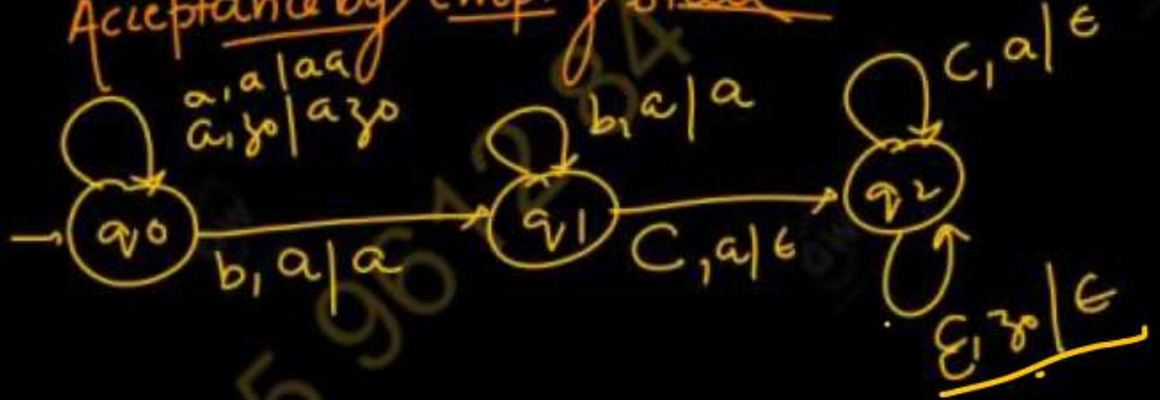
$\delta(q_1, c, a) = (q_2, \epsilon)$ — 5

$\delta(q_2, c, a) = (q_2, \epsilon)$ — 6

$\delta(q_2, \epsilon, z_0) = (q_f, z_0)$ — 7

Acceptance by final state

Acceptance by empty stack



1-6 — Same

7 $\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$

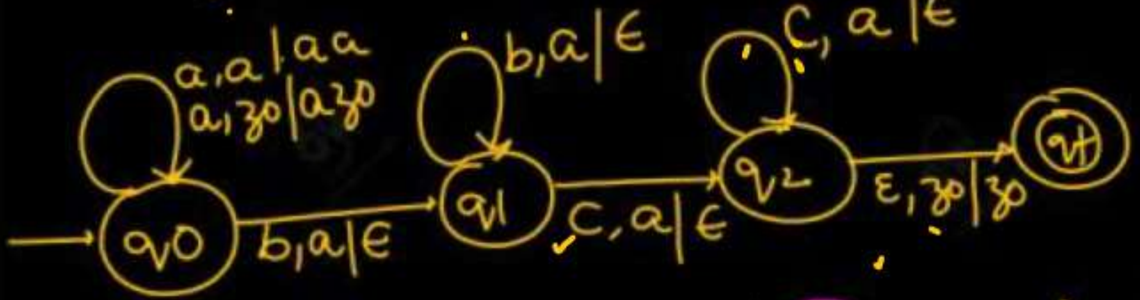
Another name <u>One Stack PDA</u>

Pushdown Automata (AKTU PYQ's)

➢ $L=\{a^{m+n}b^m c^n \mid n,m>=1\}$

a | a | a | a | a | a | b | b | c | c | c | ∈

(DPDA)



Acceptance by final state

$\delta(q0,a,z_0)=(q0,az_0)$ — 1

$\delta(q0,a,a)=(q0,aa)$ — 2

$\delta(q0,b,a)=(q1,\epsilon)$ — 3

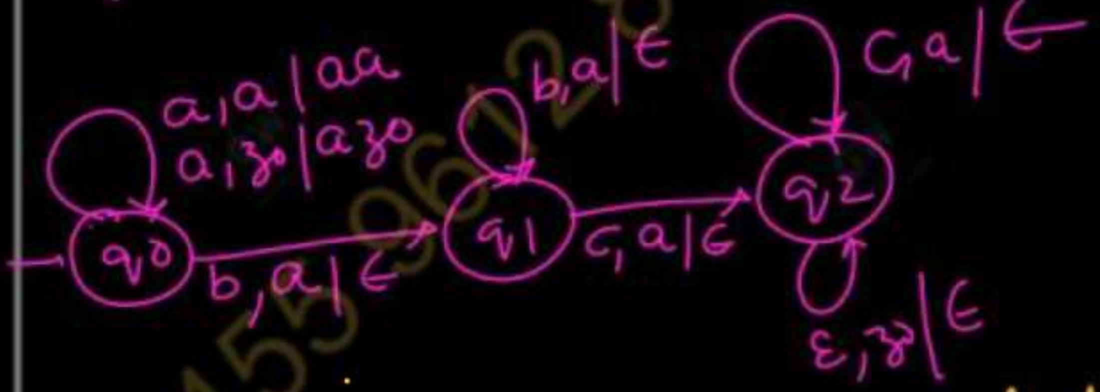
$\delta(q1,b,a)=(q1,\epsilon)$ — 4

$\delta(q1,c,a)=(q2,\epsilon)$ — 5

$\delta(q2,c,a)=(q2,\epsilon)$ — 6

$\delta(q2,\epsilon,z_0)=(q_f,z_0)$ — 7

Acceptance by empty stack



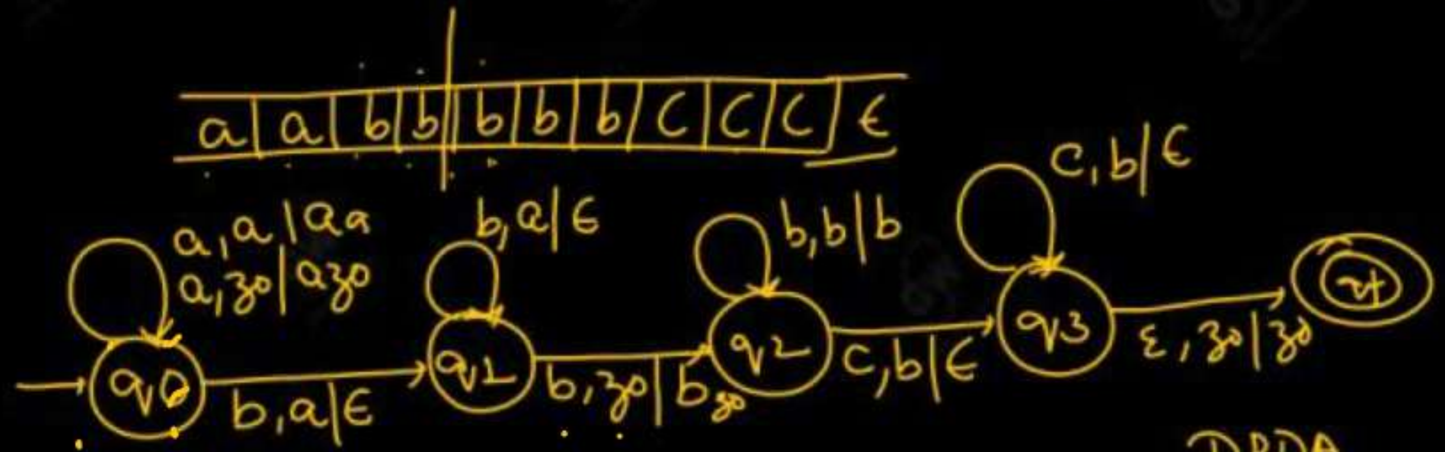
1 - 6 Same as Acceptance by final

7 $\delta(q2,\epsilon,z_0)=(q2,\epsilon)$

- $L=\{a^n b^{m+n} c^m \mid n,m>=1\}$

$a^n b^n b^m c^m$

Acceptance by final state



DPDA

$\delta(q_0, a, z_0) = (q_0, a z_0)$ — 1

$\delta(q_0, a, a) = (q_0, aa)$ — 2

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

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

$\delta(q_1, b, z_0) = (q_2, b z_0)$ — 5

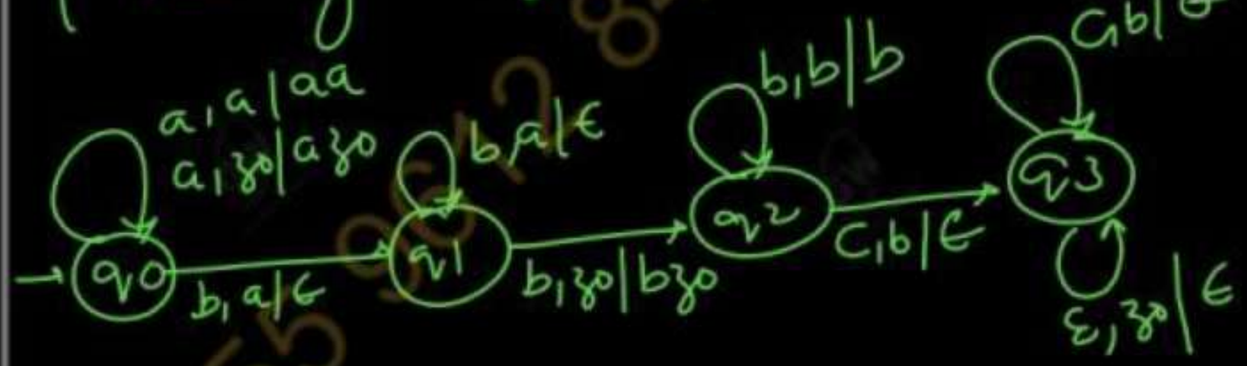
$\delta(q_2, b, b) = (q_2, bb)$ — 6

$\delta(q_2, c, b) = (q_3, \epsilon)$ — 7

$\delta(q_3, c, b) = (q_3, \epsilon)$ — 8

$\delta(q_3, \varepsilon, z_0) = (q_f, z_0)$ — 9

Acceptance by empty stack



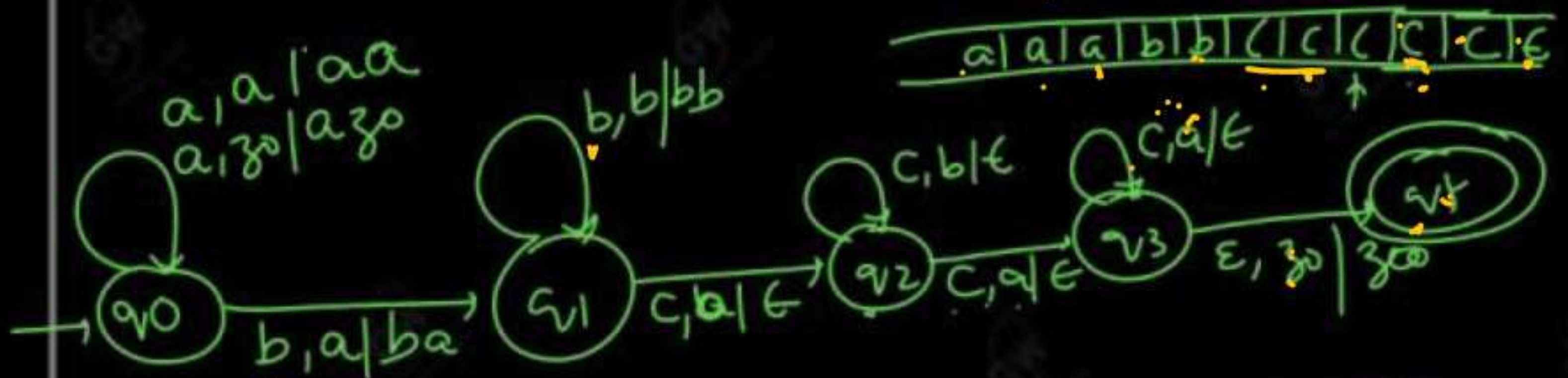
1-8 Same

$\delta(q_3, \varepsilon, z_0) = (q_3, \epsilon)$

Pushdown Automata

➤ $L=\{a^n b^m c^{m+n}\ n,m>=1\}$

$a^n b^m c^n c^m$ or $a^n b^m c^m c^n$



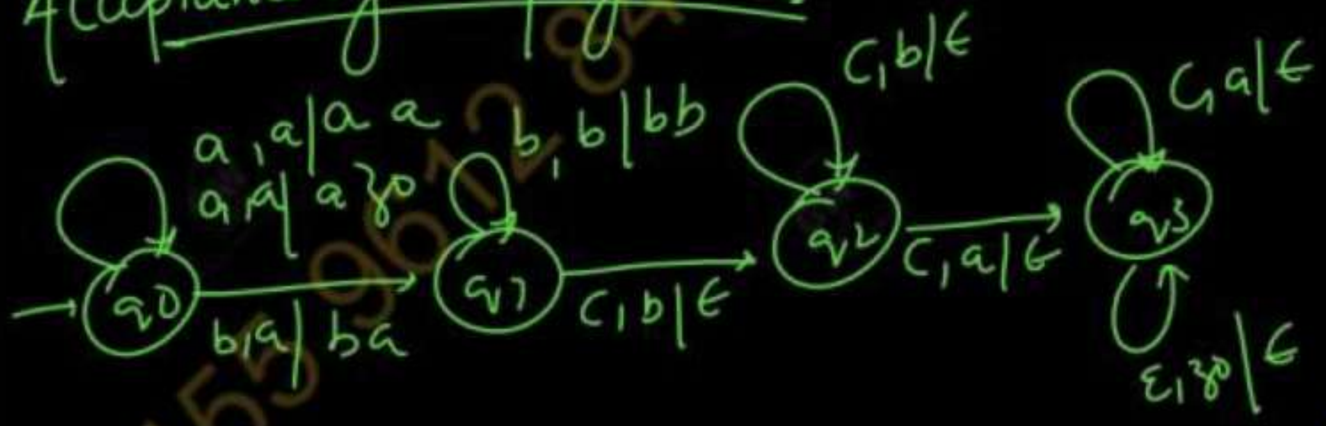
DPDA

$\delta(q0,a,z0)=(q0,az0)$ 1
$\delta(q0,a,a)=(q0,aa)$ 2
$\delta(q0,b,a)=(q0,ba)$ 3
$\delta(q1,b,b)=(q1,bb)$ 4
$\delta(q1,c,b)=(q2,\epsilon)$ 5
$\delta(q2,c,b)=(q2,\epsilon)$ 6

$\delta(q2,c,a)=(q3,\epsilon)$ 7
$\delta(q3,c,a)=(q3,\epsilon)$ 8
$\delta(q3,\varepsilon,z0)=(qf,z0)$ 9

Acceptance by final state

Acceptance by empty stack

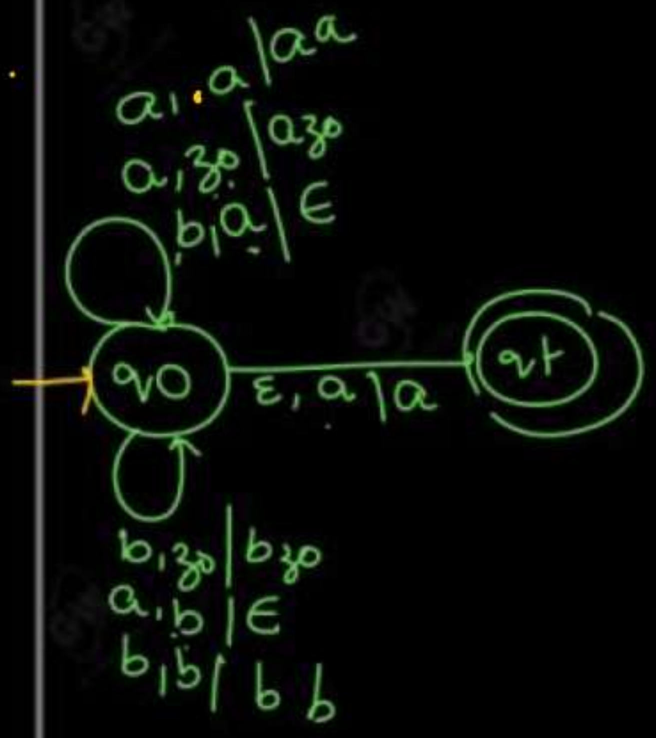


1- 8 Same

10.

$\delta(q3,\varepsilon,z0)=(q3,\epsilon)$

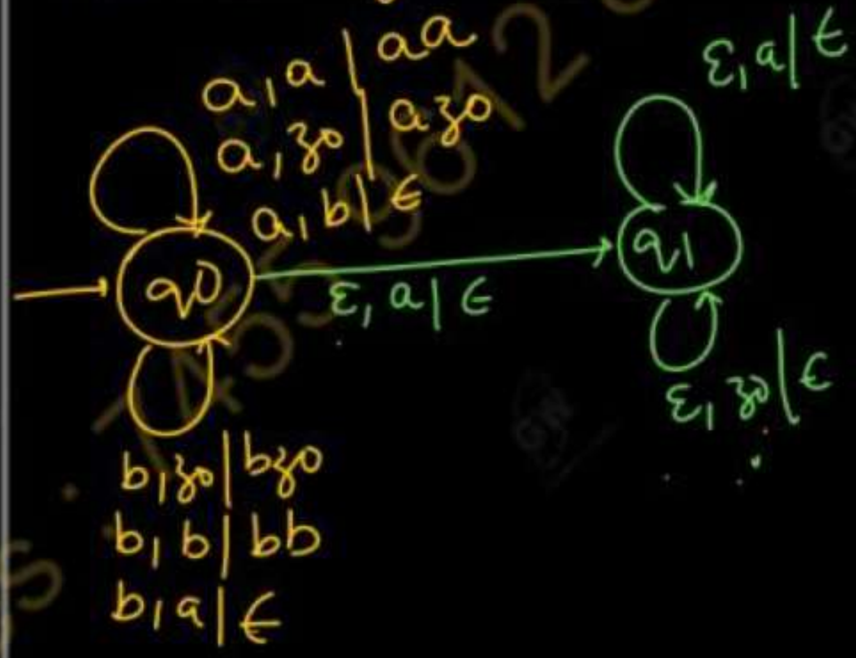
➤ $L=\{w \mid n_a(w)>n_b(w)\}$ Imp



(NPDA)
Acceptance by final state)

① $\delta(q0, a, z_0) = (q0, az_0)$
② $\delta(q0, a, a) = (q0, aa)$
③ $\delta(q0, b, a) = (q0, \epsilon)$
④ $\delta(q0, b, z_0) = (q0, bz_0)$
⑤ $\delta(q0, b, b) = (q0, bb)$
⑥ $\delta(q0, b, a) = (q0, \epsilon)$
⑦ $\delta(q0, \varepsilon, a) = (qf, a)$

Acceptance by empty stack



a	a	a	a	ε	ε

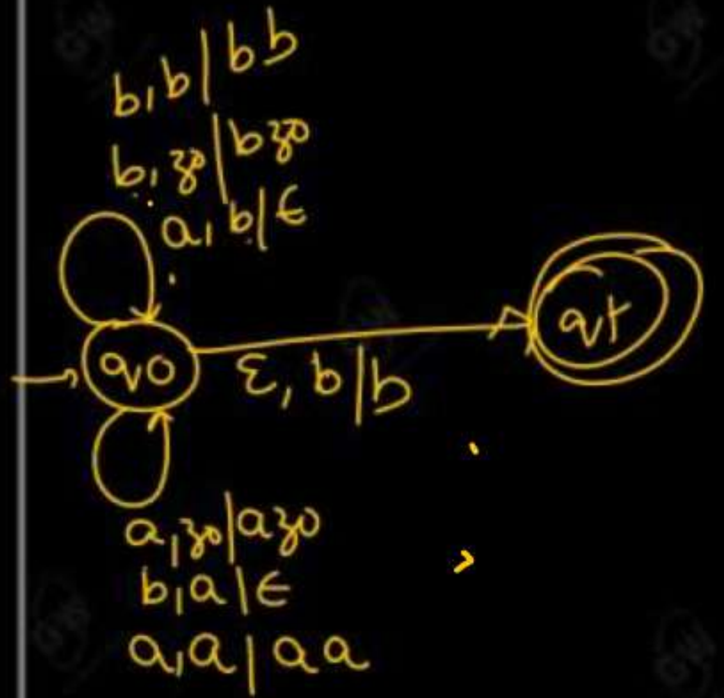
$\delta(q0, a, z_0) = (q0, az_0)$
$\delta(q0, a, a) = (q0, aa)$
$\delta(q0, a, b) = (q0, \varepsilon)$
$\delta(q0, b, z_0) = (q0, bz_0)$
$\delta(q0, b, b) = (q0, bb)$
$\delta(q0, b, a) = (q0, \varepsilon)$

$\delta(q0, \varepsilon, a) = (q1, \varepsilon)$
$\delta(q1, \varepsilon, a) = (q1, \varepsilon)$
$\delta(q1, \varepsilon, z_0) = (q1, \varepsilon)$

Pushdown Automata (AKTU PYQ's)

➤ $L=\{w \mid n_a(w)<n_b(w)\}$

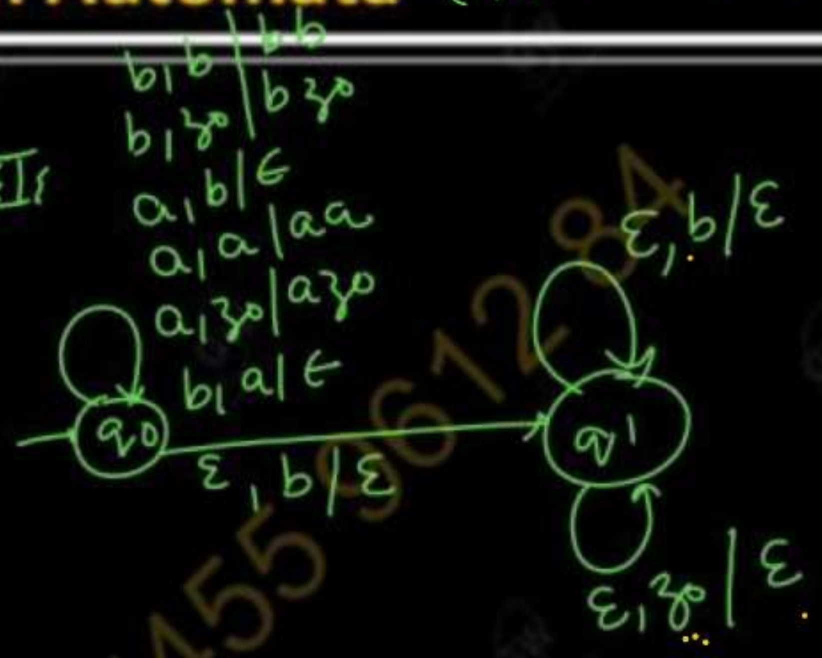
a	a	b	b	b	b	ε	ε



(NPDA)

Acceptance by final state.

$\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_0, \epsilon)$
$\delta(q_0, a, b) = (q_0, \epsilon)$
$\delta(q_0, b, z_0) = (q_0, bz_0)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, \varepsilon, b) = (q_f, b)$

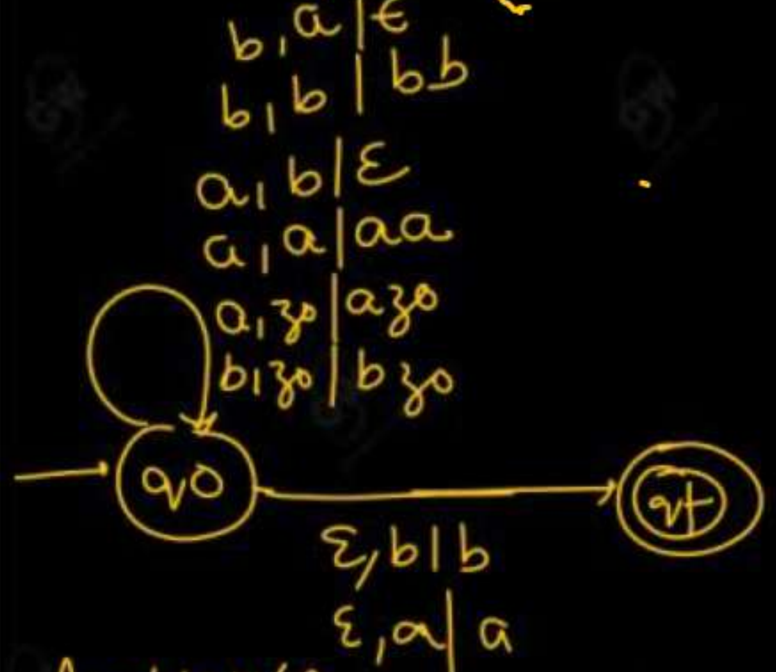


$\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, a, b) = (q_0, \epsilon)$
$\delta(q_0, b, z_0) = (q_0, bz_0)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, b, a) = (q_0, \epsilon)$

$\delta(q_0, \varepsilon, b) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, b) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$

Acceptance by empty stack

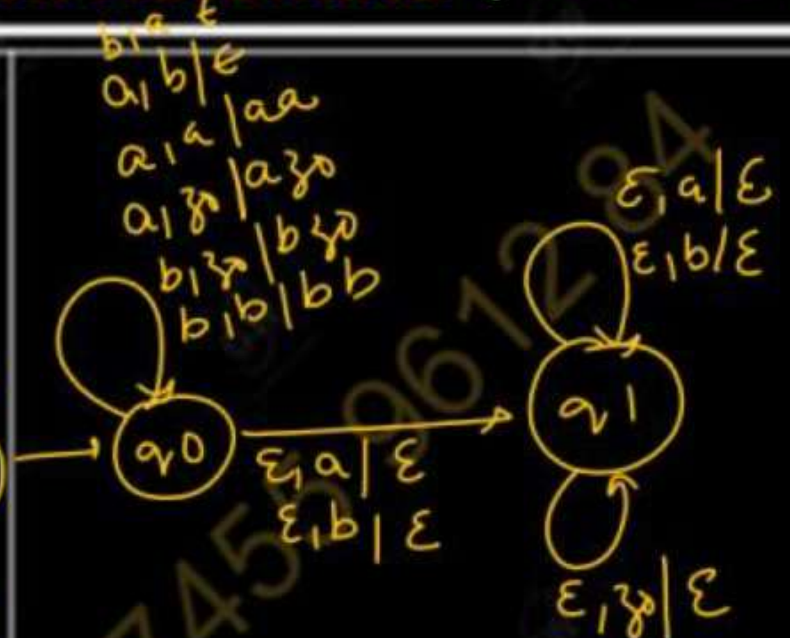
- $L=\{w \mid n_a(w) \neq n_b(w)\}$



Acceptance By final state

(NPDA)

$\delta(q_0,a,z_0)=(q_0,az_0)$
$\delta(q_0,a,a)=(q_0,aa)$
$\delta(q_0,b,z_0)=(q_0,bz_0)$
$\delta(q_0,b,b)=(q_0,bb)$
$\delta(q_0,b,a)=(q_0,\varepsilon)$
$\delta(q_0,a,b)=(q_0,\varepsilon)$
$\delta(q_0,\varepsilon,a)=(q_f,a)$
$\delta(q_0,\varepsilon,b)=(q_f,a)$



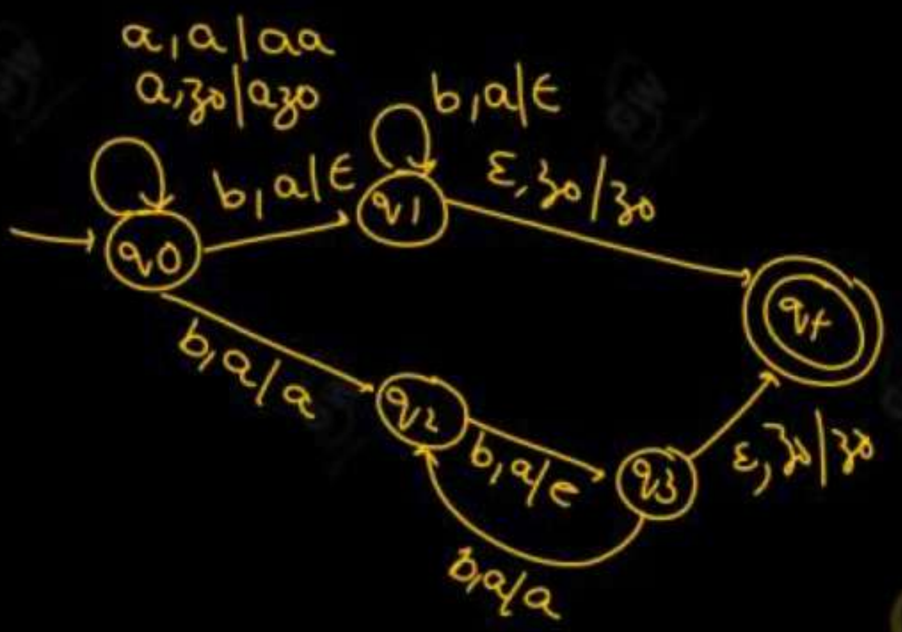
Acceptance by empty stack

$\delta(q_0,a,z_0)=(q_0,az_0)$
$\delta(q_0,a,a)=(q_0,aa)$
$\delta(q_0,b,z_0)=(q_0,bz_0)$
$\delta(q_0,b,b)=(q_0,bb)$
$\delta(q_0,a,b)=(q_0,\varepsilon)$
$\delta(q_0,b,a)=(q_0,\varepsilon)$
$\delta(q_0,\varepsilon,a)=(q_1,\varepsilon)$
$\delta(q_0,\varepsilon,b)=(q_1,\varepsilon)$
$\delta(q_1,\varepsilon,a)=(q_1,\varepsilon)$
$\delta(q_1,\varepsilon,b)=(q_1,\varepsilon)$
$\delta(q_1,\varepsilon,z_0)=(q_1,\varepsilon)$

➢ $L=\{a^n b^n \mid n>=1 \cup a^n b^{2n} \mid n>=1\}$

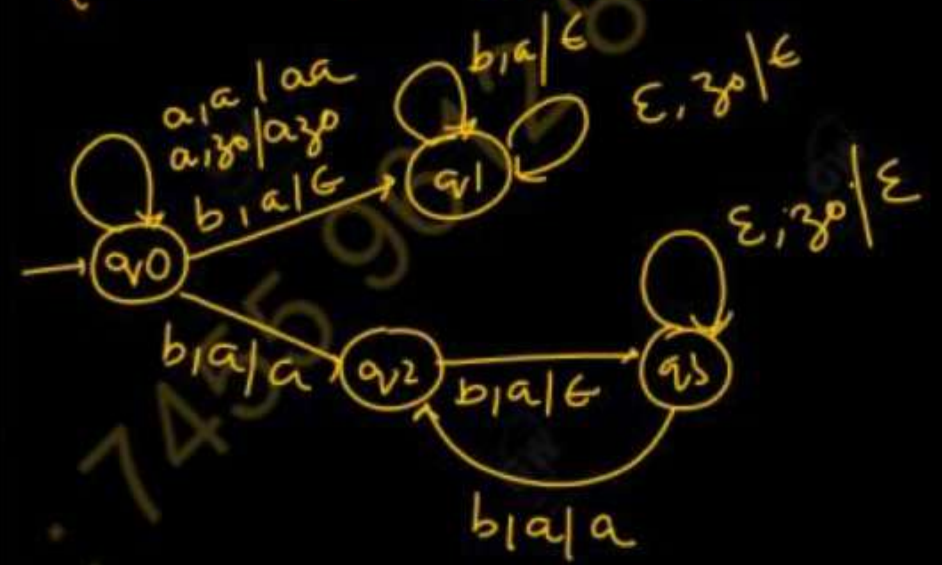
$\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \epsilon)$
$\delta(q_1, b, a) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$
$\delta(q_0, b, a) = (q_2, a)$
$\delta(q_2, b, a) = (q_3, \epsilon)$
$\delta(q_3, b, a) = (q_2, a)$
$\delta(q_3, \varepsilon, z_0) = (q_f, z_0)$

Acceptance by final state



(NPDA)

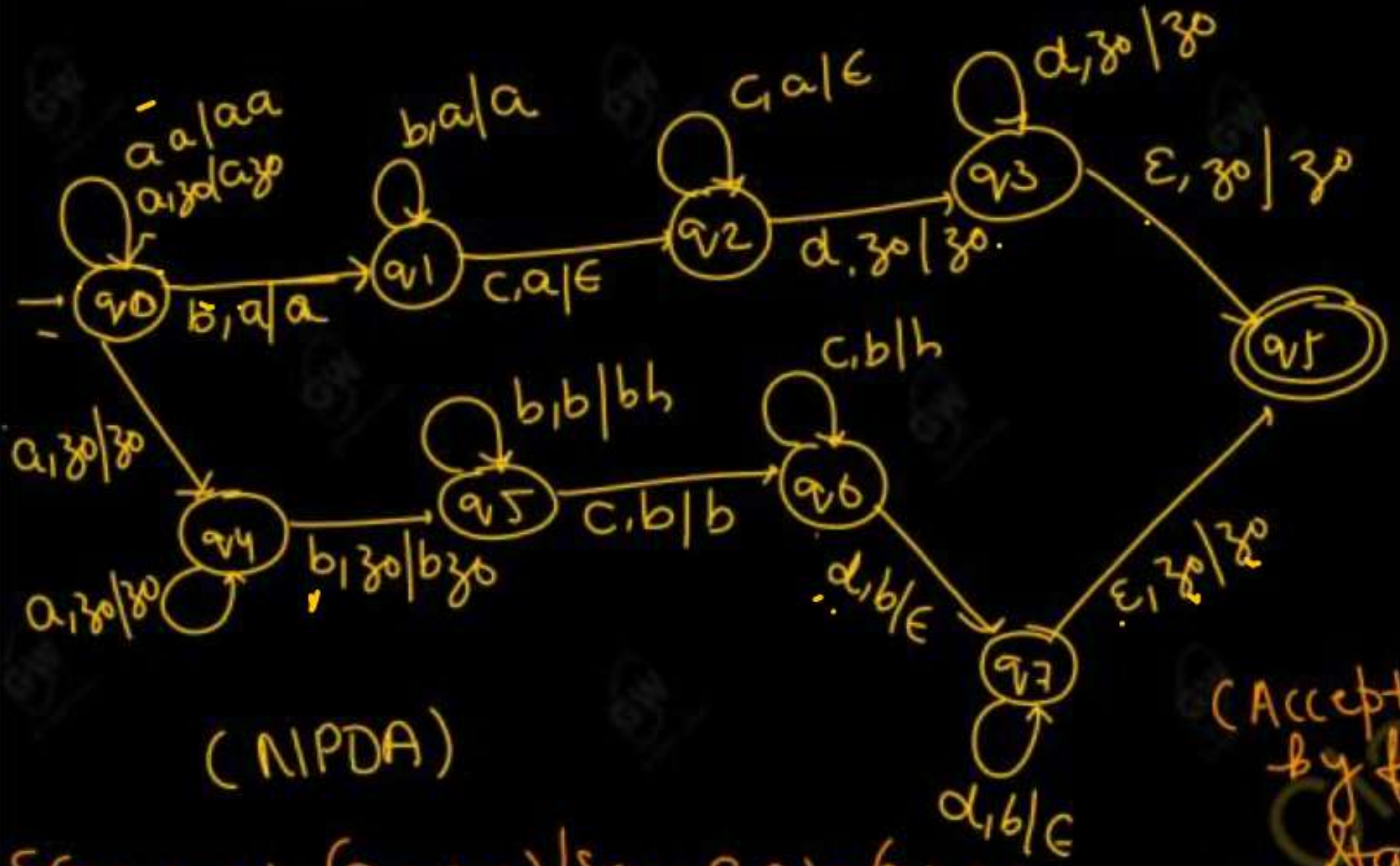
Acceptance by empty stack



$\delta(q_0, a, z_0) = (q_0, az_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \epsilon)$
$\delta(q_1, b, a) = (q_1, \epsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$
$\delta(q_0, b, a) = (q_2, a)$
$\delta(q_2, b, a) = (q_3, \varepsilon)$
$\delta(q_3, b, a) = (q_2, a)$
$\delta(q_3, \varepsilon, z_0) = (q_3, \varepsilon)$

Pushdown Automata (2020-21) AKTU PYQ

➢ $L=\{a^i b^j c^k d^l \mid i==k \text{ or } j==l\}$



(NPDA)

(Acceptance by final State)

1 $\delta(q_0,a,z_0)=(q_0,az_0)$

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

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

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

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

$\delta(q_2,c,a)=(q_2,\epsilon)$ 6

$\delta(q_2,d,z_0)=(q_3,z_0)$ 7

$\delta(q_3,d,z_0)=(q_3,z_0)$ 8

$\delta(q_3,\epsilon,z_0)=(q_f,z_0)$ 9

10 $\delta(q_0,a,z_0)=(q_4,z_0)$

11 $\delta(q_4,a,z_0)=(q_4,z_0)$

12 $\delta(q_4,b,z_0)=(q_5,bz_0)$

13 $\delta(q_5,b,b)=(q_5,bb)$

$\delta(q_5,c,b)=(q_6,b)$ 14

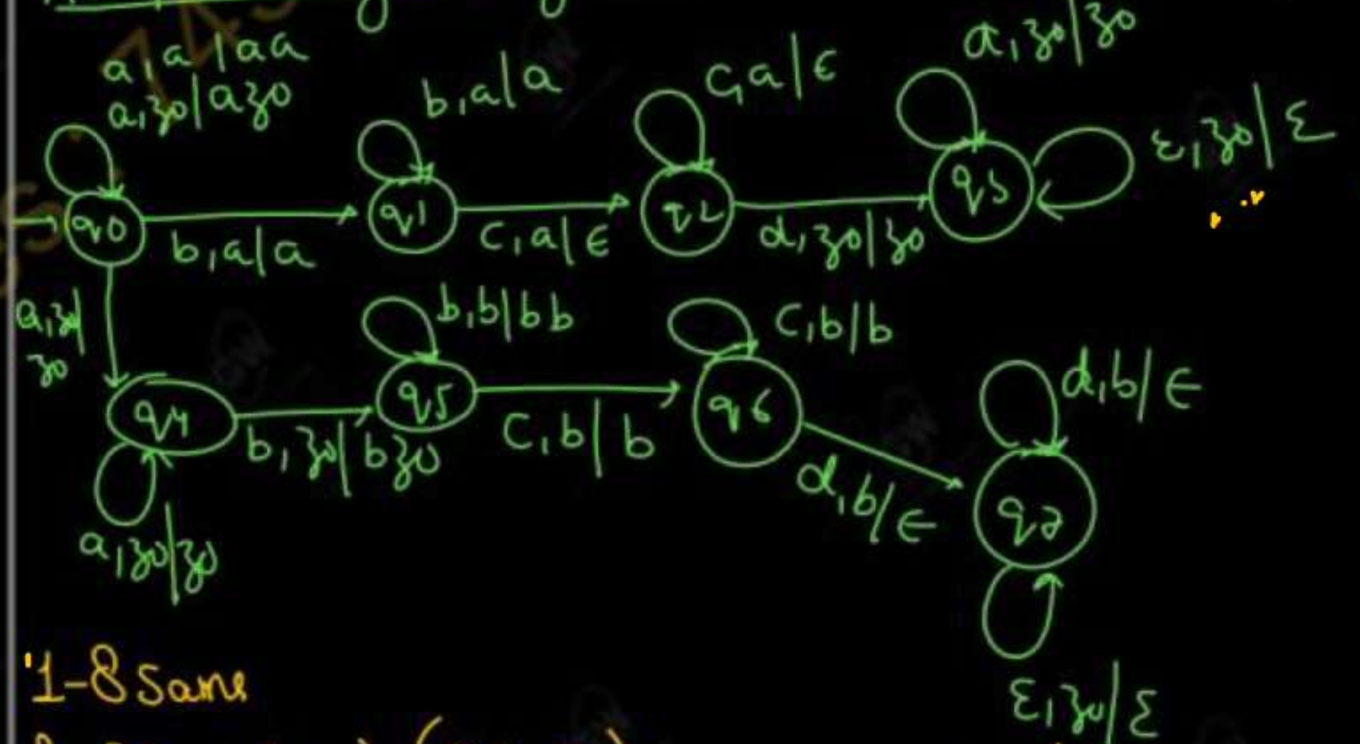
$\delta(q_6,c,b)=(q_6,b)$ 15

$\delta(q_6,d,b)=(q_7,\epsilon)$ 16

$\delta(q_7,d,b)=(q_7,\epsilon)$ 17

$\delta(q_7,\epsilon,z_0)=(q_f,z_0)$ 18

Acceptance by empty stack



1-8 Same

9 $\delta(q_3,\epsilon,z_0)=(q_3,\epsilon)$

10-17 Same

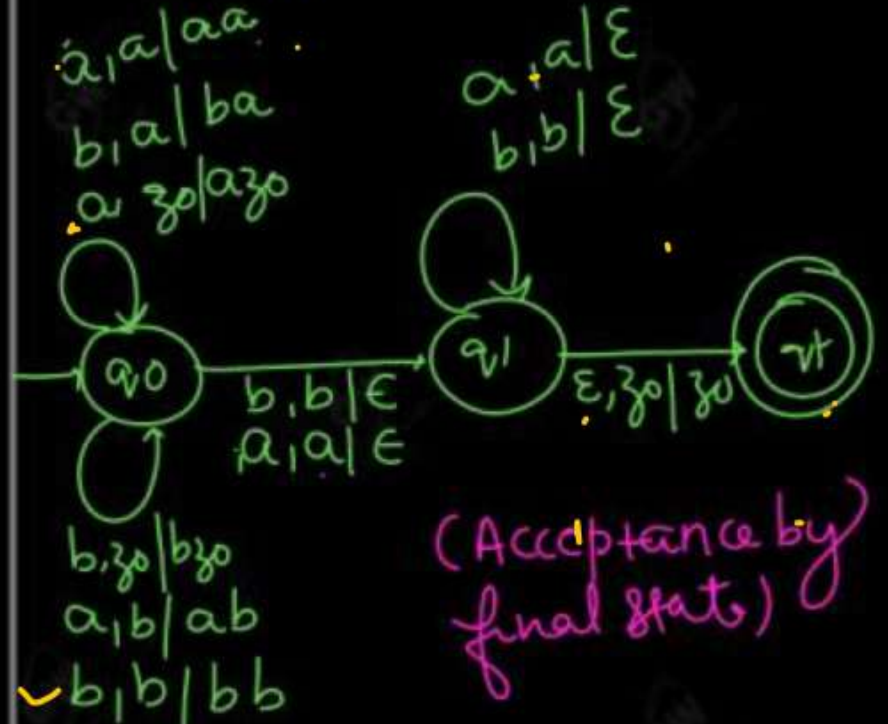
18 $\delta(q_7,\epsilon,z_0)=(q_7,\epsilon)$

Pushdown Automata

palindrome) (AKTU PYQ)

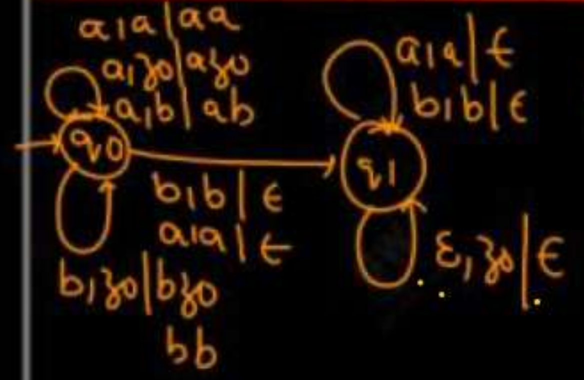
(NPDA)

➢ L={WW^R w belong to[a,b]}*



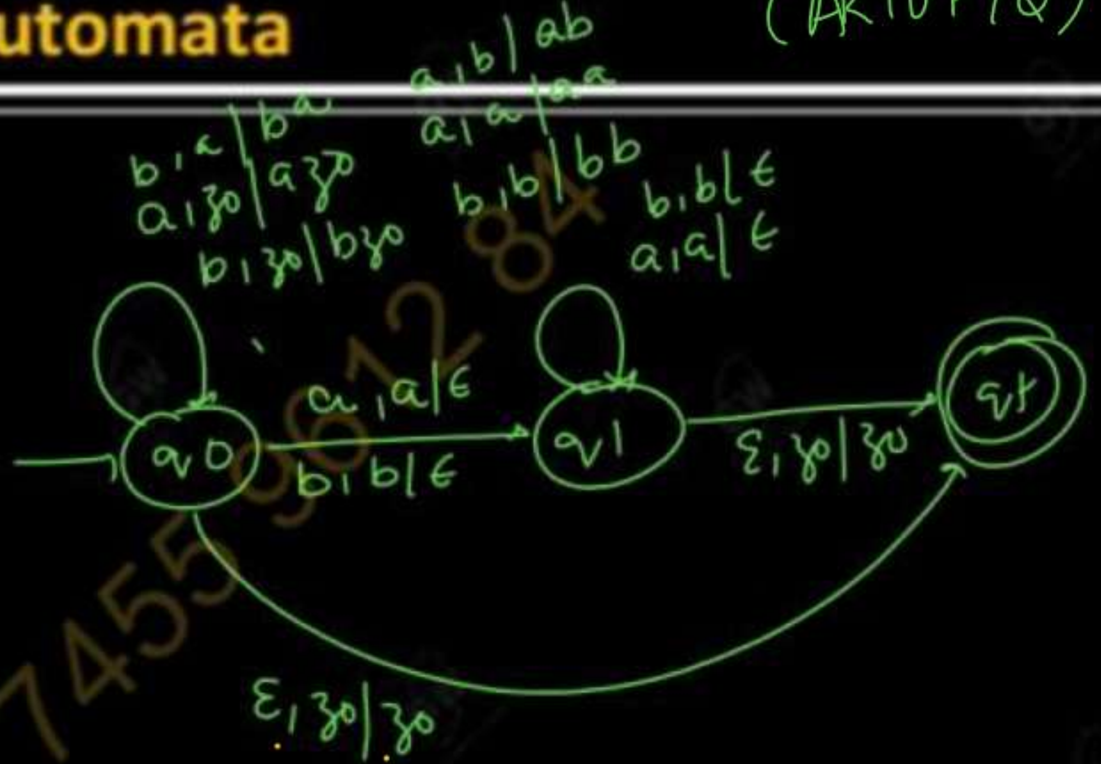
(Acceptance by final state)

$\delta(q_0,a,z_0)=(q_0,az_0)$ — 1
$\delta(q_0,a,a)=(q_0,aa)$ — 2
$\delta(q_0,b,a)=(q_0,ba)$ — 3
$\delta(q_0,b,z_0)=(q_0,bz_0)$ — 4
$\delta(q_0,b,b)=(q_0,bb)$ — 5
$\delta(q_0,a,b)=(q_0,ab)$ — 6
$\delta(q_0,a,a)=(q_1,\epsilon)$ — 7
$\delta(q_0,b,b)=(q_1,\varepsilon)$ — 8
$\delta(q_1,a,a)=(q_1,\varepsilon)$ — 9
$\delta(q_1,b,b)=(q_1,\varepsilon)$ — 10
$\delta(q_1,\varepsilon,z_0)=(q_f,z_0)$ — 11



(Acceptance by empty stack)

1-10 Same
11 $\delta(q_1,\varepsilon,z_0)=(q_1,\varepsilon)$



WW^r in ?*

$W\in\{a,b\}^*$

WW^R

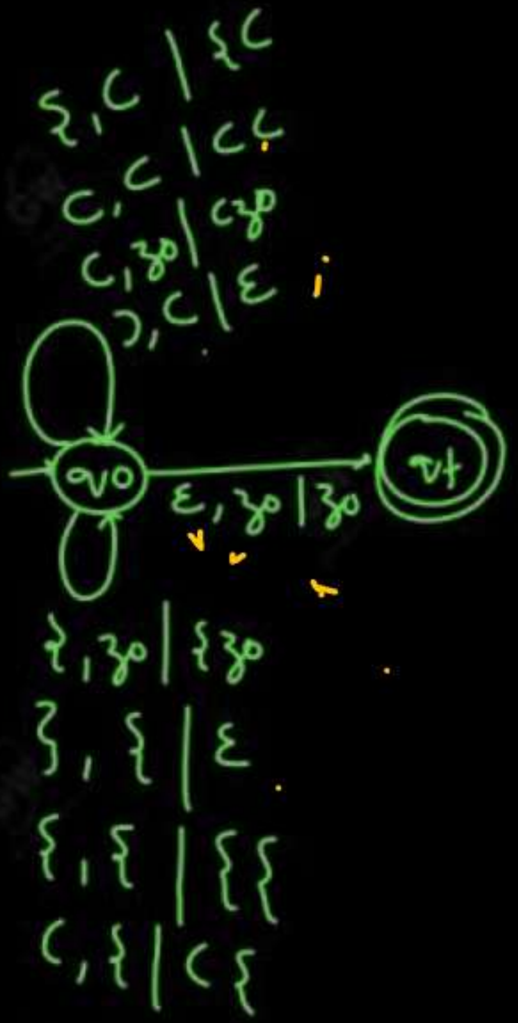
Acceptance by final state

1-11 Same

12: $\delta(q_0,\varepsilon,z_0)=(q_f,z_0)$

| | ε | |

PDA for balanced paranthesis (AKTU PYQ's)

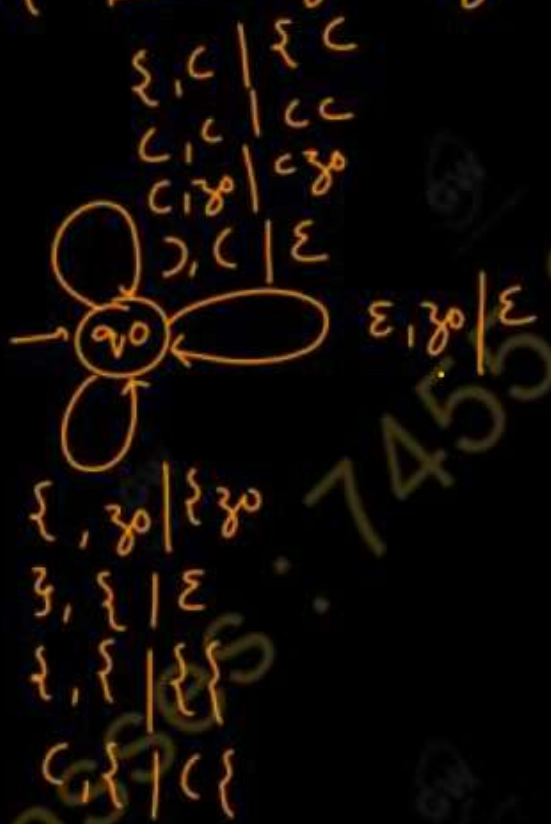


(NPDA)

Acceptance by final state

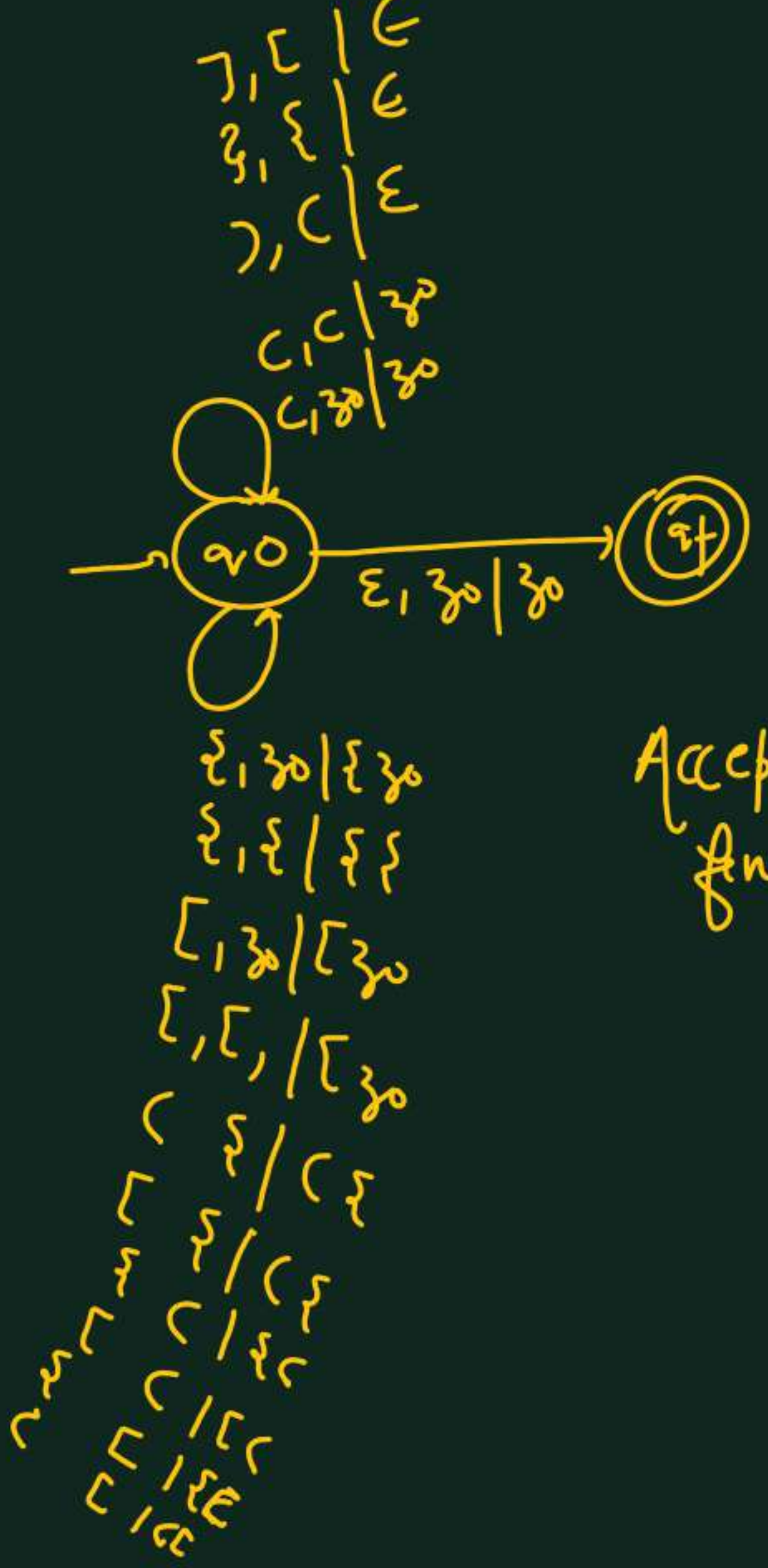
1 $\delta(q_0, (, z_0) = (q_0, (z_0)$
2 $\delta(q_0, (, () = q_0, (()$
3 $\delta(q_0,), () = (q_0, \epsilon)$
4 $\delta(q_0, \{, () = (q_0, \{()$
5 $\delta(q_0, \{, z_0) = \{q_0, \{z_0\}$
6 $\delta(q_0, \{, \{\} = \{q_0, \{\{\}$
7 $\delta(q_0\ \}, \{\} = (q_0, \varepsilon)$
8 $\delta(q_0, (, \{) = \{q_0, (\{\}$
9 $\delta(q_0, \varepsilon, z_0) = (q_f, z_0)$

Acceptance by empty stack

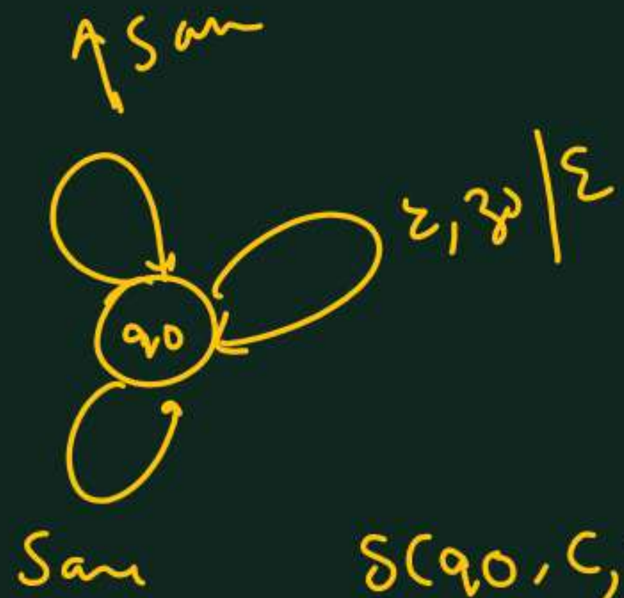


(1-8 Same)

9. $\delta(q_0, \varepsilon, z_0) = (q_0, \varepsilon)$



Acceptace by final State



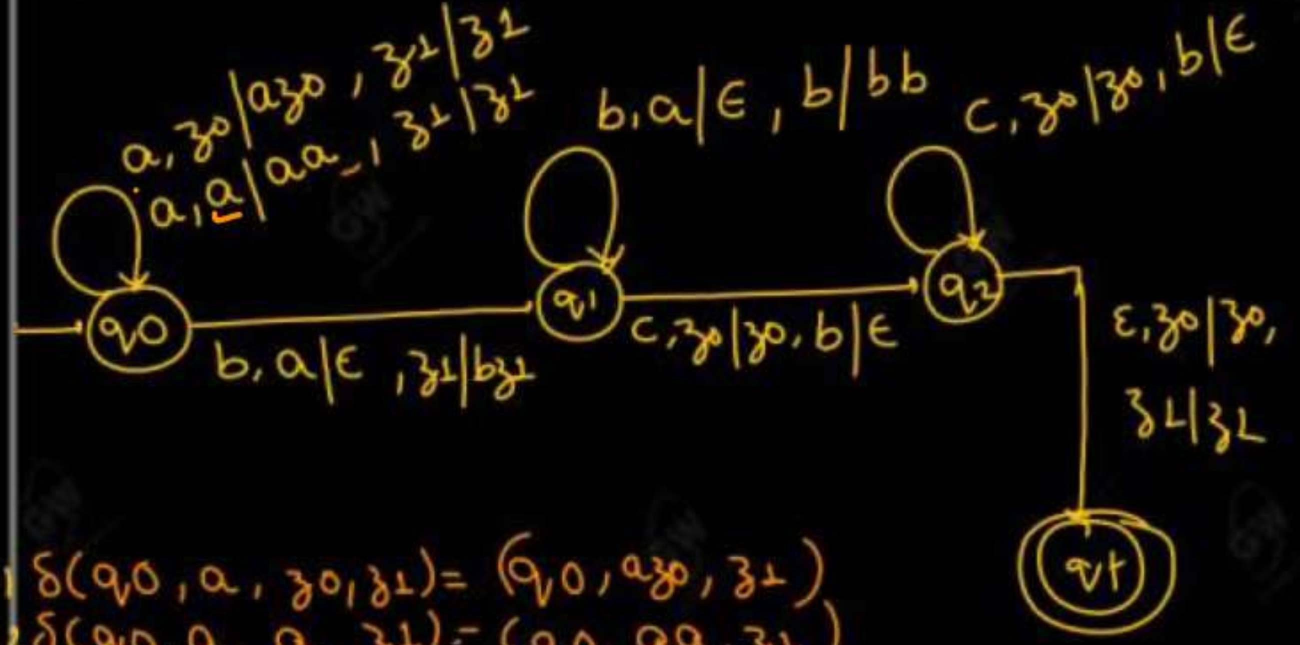
$$\delta(q0, (, z_0) = (q0, (z_0)$$
$$\delta(q0, (, () = (q0, (()$$
$$\delta(q0,), () = (q0, \varepsilon)$$

TWO STACK PDA

Imp Theory + Numerical (AKTU PYQ)

$L=\{a^n b^n c^n \; n>=1\}$

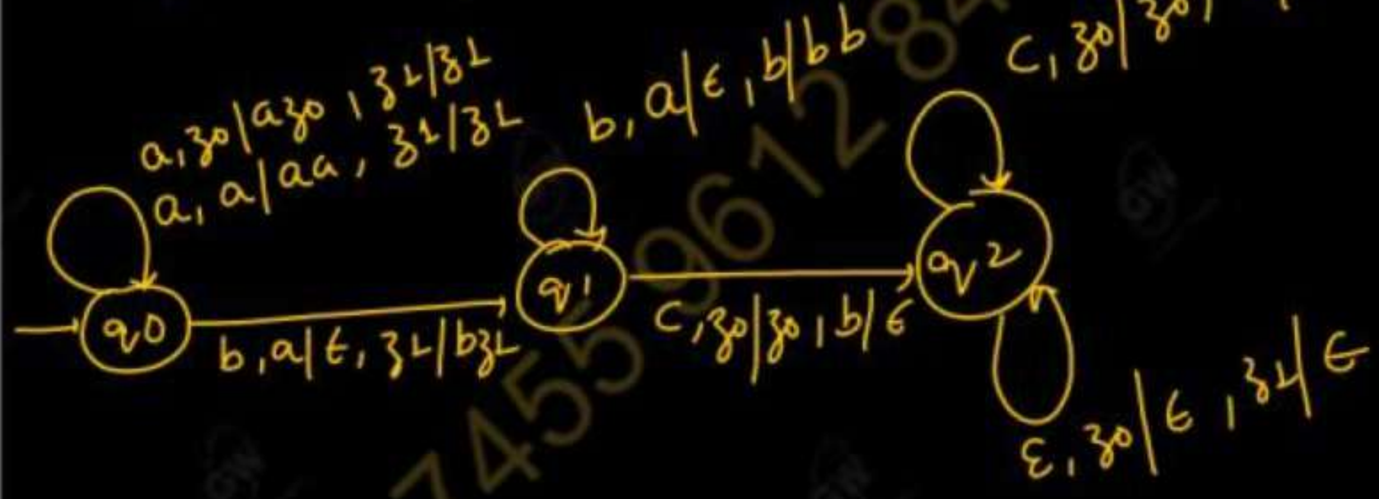
a	a	b	b	c	c	ε



1. $\delta(q_0, a, z_0, z_1) = (q_0, az_0, z_1)$
2. $\delta(q_0, a, a, z_1) = (q_0, aa, z_1)$
3. $\delta(q_0, b, a, z_1) = (q_1, \epsilon, bz_1)$
4. $\delta(q_1, b, a, b) = (q_1, \epsilon, bb)$
5. $\delta(q_1, c, z_0, b) = (q_2, z_0, \epsilon)$
6. $\delta(q_2, c, z_0, b) = (q_2, z_0, \epsilon)$
7. $\delta(q_2, \varepsilon, z_0, z_1) = (q_f, z_0, z_1)$

(Acceptance by final State)

- Acceptance by empty stack

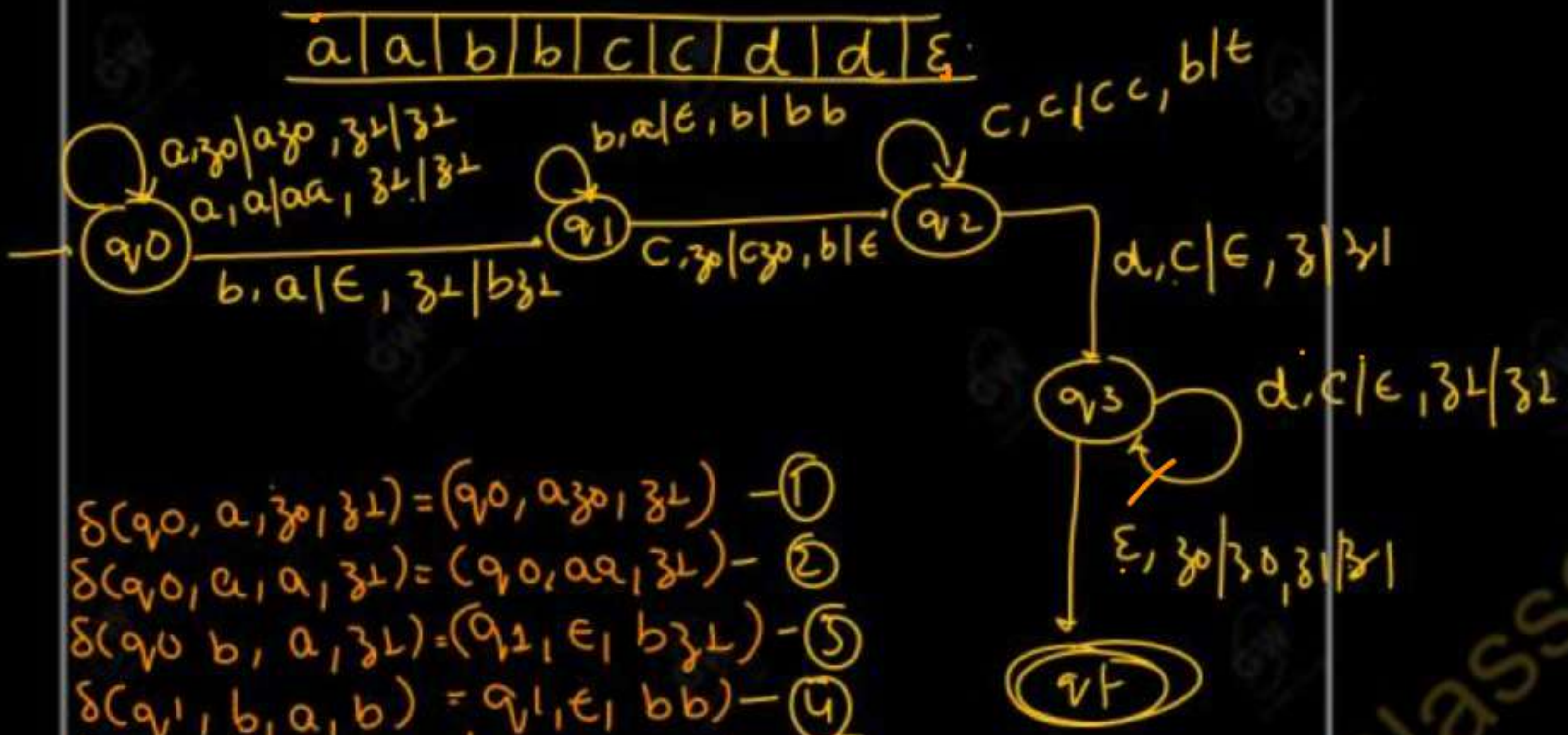


1-6 Same

7. $\delta(q_2, \varepsilon, z_0, z_1) = (q_2, \epsilon, \epsilon)$

TWO STACK PDA (AKTU PYQ)

$L=\{a^n b^n c^n d^n \mid n>=1\}$



$\delta(q_0, a, z_0, z_1) = (q_0, az_0, z_1)$ — 1

$\delta(q_0, a, a, z_1) = (q_0, aa, z_1)$ — 2

$\delta(q_0, b, a, z_1) = (q_1, \epsilon, bz_1)$ — 3

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

$\delta(q_1, c, z_0, b) = (q_2, cz_0, \epsilon)$ — 5

$\delta(q_2, c, c, b) = (q_2, cc, \epsilon)$ — 6

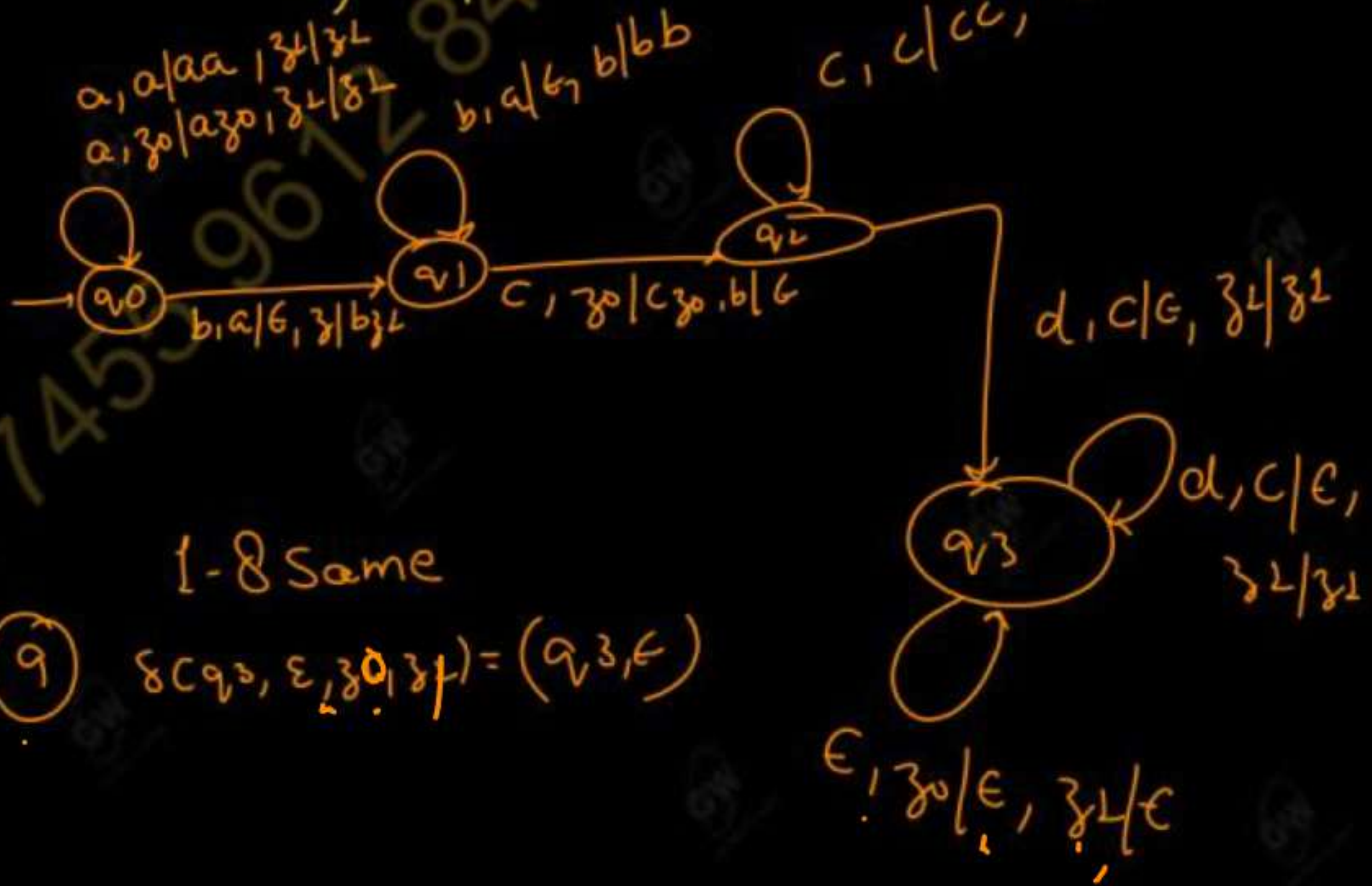
$\delta(q_2, d, c, z_1) = (q_3, \epsilon, z_1)$ — 7

$\delta(q_3, d, c, z_1) = (q_3, \epsilon, z_1)$ — 8

$\delta(q_3, \varepsilon, z_0, z_1) = (q_f, z_0, z_L)$ — 9

Final state

By empty stack



1-8 Same

9 $\delta(q_3, \varepsilon, z_0, z_1) = (q_3, \epsilon)$

TWO STACK PDA

- A Two-Stack Pushdown Automaton (Two-Stack PDA) is a theoretical computational model that extends the capabilities of a standard Pushdown Automaton (PDA) by equipping it with two stacks instead of one.
- This added stack allows the automaton to recognize a broader class of languages, specifically those within the class of Turing-recognizable languages.

Formal definition of PDA

Q is the set of states

∑is the set of input symbols

Γ is the set of pushdown symbols (which can be pushed and popped from stack)

q0 is the initial state

Z is the initial pushdown symbol (which is initially present in stack)

F is the set of final states

δ is a transition function which maps Q x {Σ ∪ ∈} x Γ x Γ into Q x Γ*x Γ *

TWO STACK PDA

Operational Description

The Two-Stack PDA reads an input symbol or an empty string (ϵ\epsilonϵ).

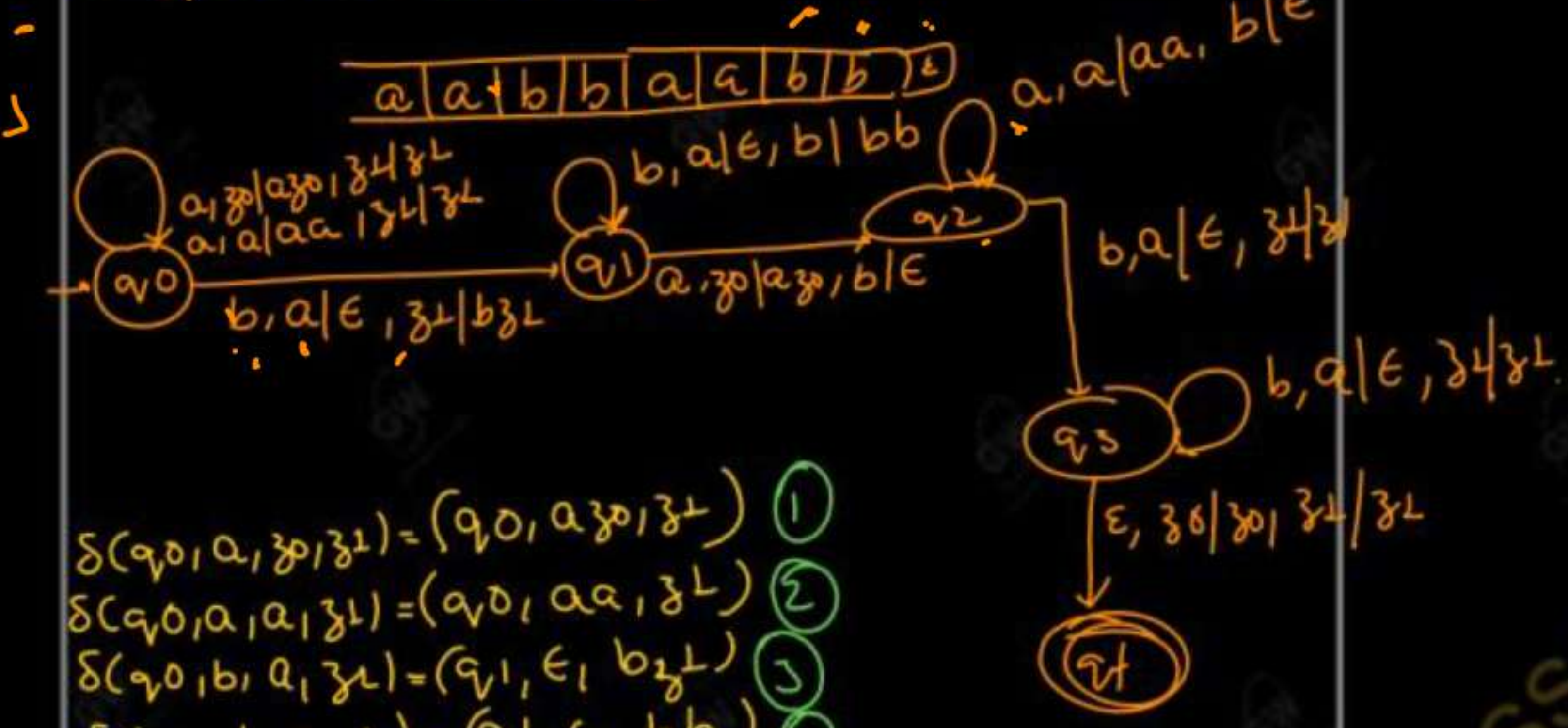
It inspects the top symbols of both stacks.

Based on the current state, the input symbol, and the top symbols of the stacks, it transitions to a new state.

During the transition, it can push or pop symbols from either or both stacks.

TWO STACK PDA

$L=\{a^n b^n a^n b^n \; n>=1\}$



$\delta(q0, a, z_0, z_1) = (q0, az_0, z_1)$ (1)

$\delta(q0, a, a, z_1) = (q0, aa, z_1)$ (2)

$\delta(q0, b, a, z_1) = (q1, \epsilon, bz_1)$ (3)

$\delta(q1, b, a, b) = (q1, \epsilon, bb)$ (4)

$\delta(q1, a, z_0, b) = (q2, az_0, \epsilon)$ (5) final State

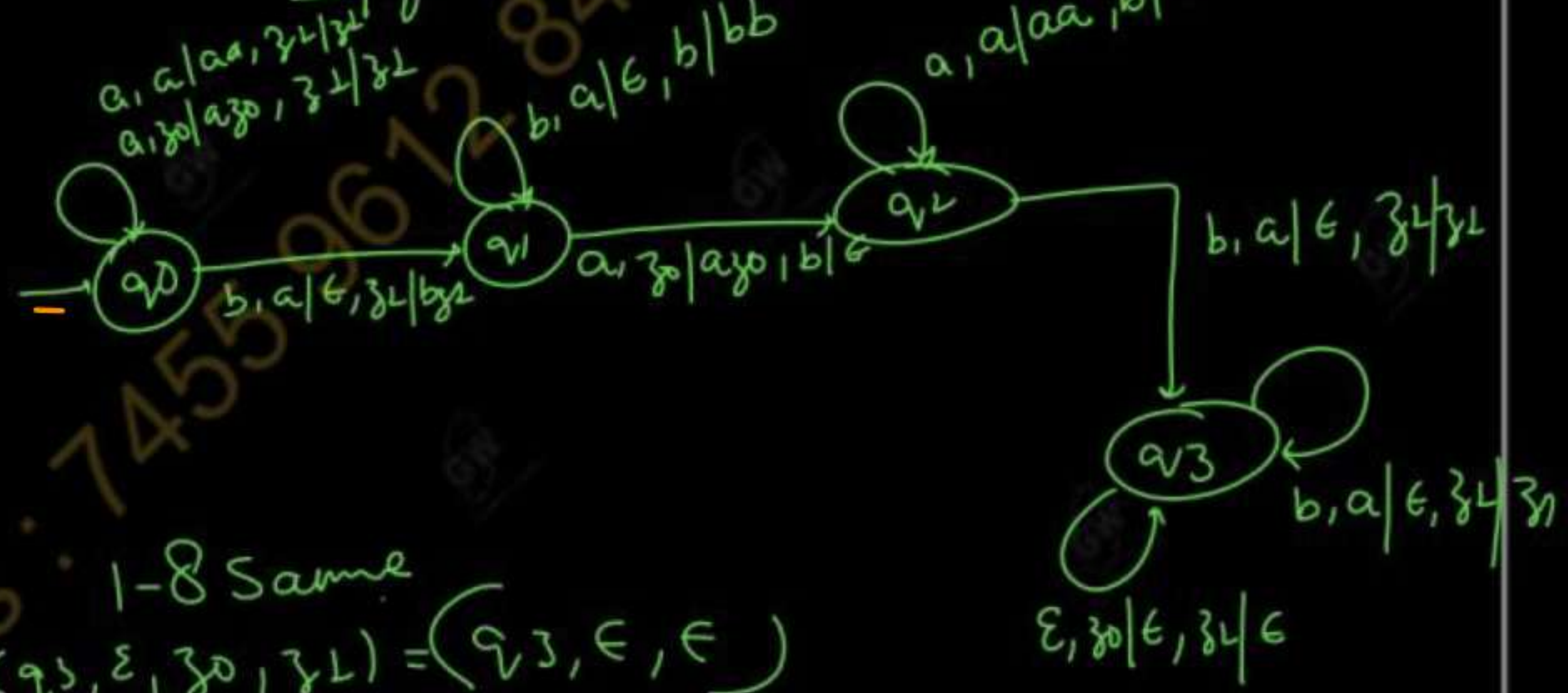
$\delta(q2, a, a, b) = (q2, aa, \epsilon)$ (6)

$\delta(q3, b, a, z_1) = (q3, \epsilon, z_1)$ (7)

$\delta(q3, b, a, z_1) = (q3, \epsilon, z_1)$ (8)

$\delta(q3, \varepsilon, z_0, z_1) = (qf, z_0, z_1)$ (9)

Empty Stack



1-8 Same

$\delta(q3, \varepsilon, z_0, z_1) = (q3, \epsilon, \epsilon)$

CFG TO PDA

- If context free grammar is not in GNF FORM
- For non-terminal symbol
- $\delta(q, \varepsilon, A) = (q, \alpha)$ $A \rightarrow \alpha$
- For each terminal symbols, add the following rule:

$\delta(q, a, a) = (q, \varepsilon)$ for every terminal symbol

Peter Linz
D. Ullman

$M = \{Q, \Sigma, \Gamma, q0, Z, F, \delta\}$
$\{q, \Sigma, (V \cup T), q, S, \Phi, \delta\}$

CFG TO PDA

S->aSb

S->ab

Test whether aaabbb

Acceptance through empty stack

For non-terminal

$\delta(q, \epsilon, S) = (q, aSb)$

$\delta(q, \epsilon, S) = (q, ab)$

For terminal

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

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

$M = Q, \Sigma, \Gamma, F, q_0, z, \delta$

$\{q\}, \{a, b\}, \{a, b, S\}, \{\phi\}, \{q\}, \{S\}, \delta$

$\delta(q, aaabbb, S) \vdash (q, aaabbb, aSb)$

$\vdash (q, aabbb, Sb)$

$\vdash (q, aabbb, aSbb)$

$\vdash (q, abbb, Sbb)$

$\vdash (q, abbb, abbb)$

$\vdash (q, bbb, bbb)$

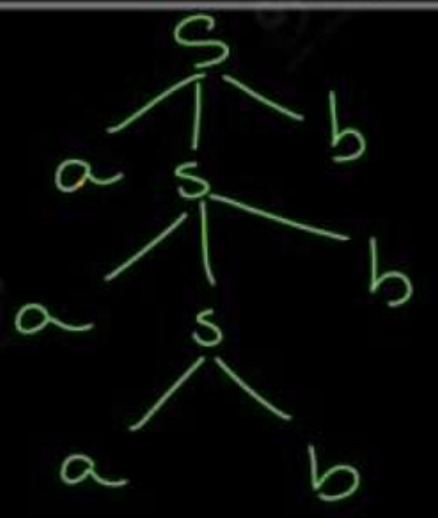
$\vdash (q, bb, bb)$

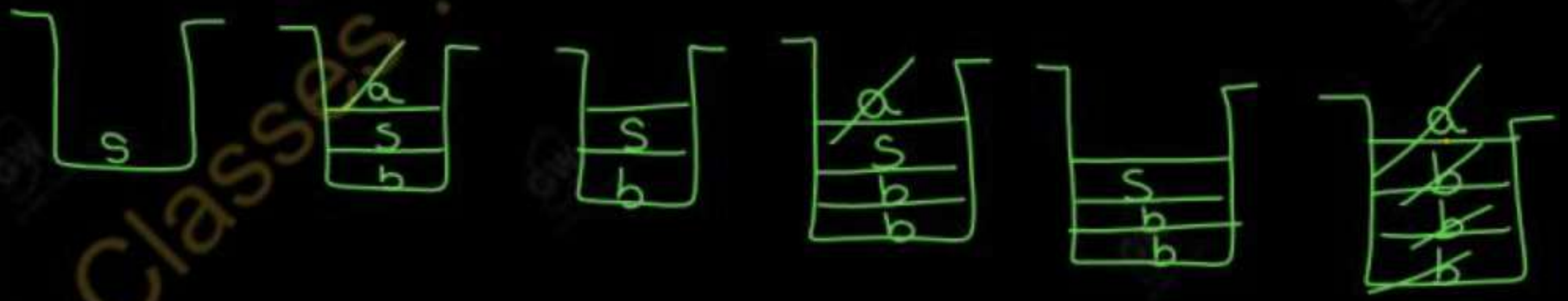
$\vdash (q, b, b)$

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

(NPDA)

~~aaabbb~~





CFG TO PDA (AKTU PYQ)

I->a/b/Ia/Ib/I0/I1

(Not in GNF)

E->I/E*E/E+E/(E)

PDA

For non-terminal I

$\delta(q,\varepsilon,I) = (q,a)$
$\delta(q,\varepsilon,I) = (q,b)$
$\delta(q,\varepsilon,I) = (q,Ia)$
$\delta(q,\varepsilon,I) = (q,Ib)$
$\delta(q,\varepsilon,I) = (q,I0)$
$\delta(q,\varepsilon,I) = (q,I1)$

For non-terminal E

$\delta(q,\varepsilon,E) = (q,I)$
$\delta(q,\varepsilon,E) = (q,E*E)$
$\delta(q,\varepsilon,E) = (q,E+E)$
$\delta(q,\varepsilon,E) = (q,(E))$

for terminal

(NPDA)

$\delta(q,a,a) = (q,\varepsilon)$
$\delta(q,b,b) = (q,\varepsilon)$
$\delta(q,0,0) = (q,\varepsilon)$
$\delta(q,1,1) = (q,\varepsilon)$
$\delta(q,*,*) = (q,\varepsilon)$
$\delta(q,+,+) = (q,\varepsilon)$
$\delta(q,(,() = (q,\varepsilon)$
$\delta(q,),)) = (q,\varepsilon)$

Acceptance by empty stack.

CFG TO PDA

- **If context free grammar is in GNF FORM**
- For non-terminal symbol s->0A
- $\delta(q, 0, S) = (q, A)$

S->0

$\delta(q, 0, S) = (q, \varepsilon)$

$M = \{Q, \Sigma, \Gamma, q0, Z, F, \delta\}$
$\{q, \Sigma, V, q, S, \Phi, \delta\}$

S->0BB

B->0S/1S/0   GNF

Test whether 010000 is a accepted or not

$\delta(q, 0, S) = (q, BB)$
$\delta(q, 0, B) = q, S)$
$\delta(q, 1, B) = (q, S)$
$\delta(q, 0, B) = (q, \varepsilon)$
$\delta(q, 0, B)$

$Q = \{q\}$ $F = \phi$
$q_0 = \{q\}$
δ $\Sigma = \{$
$\Gamma = \{S, B\}$

010000

$\delta(q, 010000, S) \vdash (q, 10000, BB)$
$\vdash (q, 0000, SB)$
$\vdash (q, 000, BBB)$
$\vdash (q, 00, BB)$
$\vdash (q, 0, B)$
$\vdash (q, \varepsilon, \varepsilon)$

CFG TO PDA

S->aA

A->aABC/bB/a

B->b

C->c

GNF

whether aaabc is accepted or not

$\delta(q, a, S) = (q, A)$
$\delta(q, a, A) = (q, ABC)$
$\delta(q, b, A) = (q, B)$
$\delta(q, a, A) = (q, \varepsilon)$
$\delta(q, b, B) = (q, \varepsilon)$
$\delta(q, c, C) = (q, \varepsilon)$

aaabc

$\delta(q, aaabc, S) \vdash (q, aabc, A)$
$\vdash (q, abc, ABC)$
$\vdash (q, bc, BC)$
$\vdash (q, c, C)$
$\vdash (q, \varepsilon, \varepsilon)$

this is accepted by Empty Stack (NPDA)

CFG TOPDA

S->AA/a

A->SA/b

aloT

GNF

test whether abbabb accepted or not

$\delta(q, \varepsilon, S) = (q, AA)$

$\delta(q, \varepsilon, S) = (q, a)$

$\delta(q, \varepsilon, A) = (q, SA)$

$\delta(q, \varepsilon, A) = (q, b)$

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

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

$\delta(q, abbabb, S) \vdash (q, abbabb, AA)$

$\vdash (q, abbabb, SAA)$

$\vdash (q, abbabb, aAA)$

$\vdash (q, bbabb, AA)$

$\vdash (q, bbabb, bA)$

$\vdash (q, babb, A)$

$\vdash (q, babb, SA)$

$\vdash (q, babb, AAA)$

$\vdash (q, babb, bAA)$

$\vdash (q, abb, \underline{AA})$

$\vdash (q, abb, SAA)$

$\vdash (q, abb, aAA)$

$\vdash (q, bb, AA)$

$\vdash (q, bb, bA)$

$\vdash (q, b, A)$

$\vdash (q, b, b)$

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

CFG TOPDA

S->0S1/A

A->1A0/S/ ε

for termenal

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

$\delta(q, 1, 1) = (q, \varepsilon)$

for Non-terminal

$\delta(q, \varepsilon, S) = (q, 0S1)$

$\delta(q, \varepsilon, S) = (q, A)$

$\delta(q, \varepsilon, A) = (q, 1A0)$

$\delta(q, \varepsilon, A) = (q, S)$

$\delta(q, \varepsilon, A) = (q, \varepsilon)$

CFG TOPDA

E->aAB/d

A->BA/d

B->Ead/c

For non-terminal

$\delta(q, \epsilon, E) = (q, aAB)$
$\delta(q, \epsilon, E) = (q, d)$
$\delta(q, \epsilon, A) = (q, BA)$
$\delta(q, \epsilon, A) = (q, d)$
$\delta(q, \epsilon, B) = (q, Ead)$
$\delta(q, \epsilon, B) = (q, c)$

for terminal

$\delta(q, a, a) = (q, \epsilon)$
$\delta(q, d, d) = (q, \epsilon)$
$\delta(q, c, c) = (q, \epsilon)$

CFG TOPDA

S->XY

X->AX/BX/a

Y->YA/YB/a

A->a

B->b

$\delta(q,\varepsilon,S) = (q, XY)$

$\delta(q,\varepsilon,X) = (q, AX)$

$\delta(q,\varepsilon,X) = (q, BX)$

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

$\delta(q,\varepsilon,Y) = (q, YA)$

$\delta(q,\varepsilon,Y) = (q, YB)$

$\delta(q,\varepsilon,Y) = (q, a)$

$\delta(q,\varepsilon,A) = (q, a)$

$\delta(q,\varepsilon,B) = (q, b)$

<u>terminal</u>

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

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

Difference

S. No	DPDA(Deterministic Pushdown Automata)	NPDA(Non-deterministic Pushdown Automata)
1.	It is less powerful than NPDA .<u>Example:</u> We can only construct DPDA for odd-length palindromes and not for even length palindromes. wcw^r abcba	It is more powerful than DPDA .<u>Example:</u> NPDA can be constructed for both even-length and odd-length palindromes. ww^r abba wcw^r waw^r wbw^r
2.	It is possible to convert every DPDA to a corresponding NPDA.	It is not possible to convert every NPDA to a corresponding DPDA.
3.	The language accepted by DPDA is a subset of the language accepted by NDPA.	The language accepted by NPDA is not a subset of the language accepted by DPDA.

Difference

S. No	DPDA(Deterministic Pushdown Automata)	NPDA(Non-deterministic Pushdown Automata)
4.	The language accepted by DPDA is called DCFL(Deterministic Context-free Language) which is a subset of NCFL(Non-deterministic Context-free Language) accepted by NPDA.	The language accepted by NPDA is called NCFL(Non-deterministic Context-free Language).
5.	There is only one state transition from one state to another state for an input symbol.	There may or maynot be more than one state transition from one state to another state for same input symbol.

Instantaneous Description (ID)

2 Marks

Instantaneous Description (ID) is an informal notation of how a PDA "computes" a input string and make a decision that string is accepted or rejected.

A ID is a triple (q, w, α), where:

1. q is the current state.
2. w is the remaining input.
3. α is the stack contents, top at the left.

$\delta(q_0, a, z_0) = (q_0, a z_0)$
$\delta(q_0, a, a) = (q_0, a a)$
$\delta(q_0, b, a) = (q_1, \varepsilon)$
$\delta(q_1, b, a) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

$\boxed{a^n b^n \; n \geq 1}$

aabb

$\delta(q_0, aabb, z_0) \vdash (q_0, abb, a z_0)$
$\vdash (q_0, bb, a a z_0)$
$\vdash (q_1, b, a z_0)$
$\vdash (q_1, \varepsilon, z_0)$
$\vdash (q_f, z_0)$

Turnstile notation

$\delta(q_0, aabb, z_0) \vdash^* (q_f, z_0)$

⊢ sign is called a "turnstile notation" and represents one move.

⊢* sign represents a sequence of moves.

Eg- $(p, b, T) \vdash (q, w, \alpha)$

This implies that while taking a transition from state p to state q, the input symbol 'b' is consumed, and the top of the stack 'T' is replaced by a new string 'α'

Equivalence of Acceptance by Final State and Empty Stack

- If L = N(P1) for some PDA P1, then there is a PDA P2 such that L = L(P2). That means the language accepted by empty stack PDA will also be accepted by final state PDA.
- If there is a language L = L (P1) for some PDA P1 then there is a PDA P2 such that L = N(P2). That means language accepted by final state PDA is also acceptable by empty stack PDA

PDA Acceptance

A language can be accepted by Pushdown automata using two approaches:

1. **Acceptance by Final State:** The PDA is said to accept its input by the final state if it enters any final state in zero or more moves after reading the entire input.

Let $P = (Q, \Sigma, \Gamma, \delta, q0, Z, F)$ be a PDA. The language acceptable by the final state can be defined as:

$$L(PDA) = \{w \mid (q0, w, Z) \vdash^* (q, \varepsilon, Z), q \in F\}$$

$\delta(q0, aabb, z0) = (q_f \vdash \varepsilon, Z_0)$

2. **Acceptance by Empty Stack:** On reading the input string from the initial configuration for some PDA, the stack of PDA gets empty.

Let $P = (Q, \Sigma, \Gamma, \delta, q0, Z, F)$ be a PDA. The language acceptable by empty stack can be defined as:

$$N(PDA) = \{w \mid (q0, w, Z) \vdash^* (p, \varepsilon, \varepsilon), q \in Q\}$$

PDA TO CFG

$M = \{ \{q0, q1\}, \{0,1\}, \{Z0, X\}, \delta, q0, Z0, \Phi \}$

1. $\delta(q0,1,Z0)=(q0,XZ0)$
2. $\delta(q0,1,X)=(q0,XX)$
3. $\delta(q0,0,X)=(q1,X)$
4. $\delta(q1,1,X)=(q1,\varepsilon)$
5. $\delta(q1,0,Z0)=(q0,Z0)$
6. $\delta(q0,\varepsilon,Z0)=(q0,\varepsilon)$

q0, q1,
X Z

[q0, Z0, q0] — A
[q0, Z0, q1] — B
[q1, Z0, q0] — C
[q1, Z0, q1] — D
[q0, X, q0] — E
[q0, X, q1] — F
[q1, X, q0] — G
[q1, X, q1] — H

STEP 1: SU[q,A,P] q,p belong Q

A BELONG τ

S->[q0,z0,p] for each p

q0 represent initial state zo initial stack symbol p means all the state use in PDA

S → [q0, Z0, q0]
S → [q0, Z0, q1]

Step 1

$\boxed{S \rightarrow A \mid B}$

PDA TO CFG

step 2: IF $\delta(q,X,A)=(P,B1B2..........BN)$

[q,A,qm+1]->X [p,B1,q2]

[q2,B2,q3].....[qm,BN,Qm+1]

[q0,Z0,q0] – A
[q0,Z0,q1] – B
[q1,Z0,q0] – C
[q1,Z0,q1] – D

[q0,X,q0] – E
[q0,X,q1] – F
[q1,X,q0] – G
[q1,X,q1] – H

1. $\delta(q0,1,Z0)=(q0,XZ0)$

[q0,Z0,q0] = 1 [q0,X q0] [q0 Z0 q0]
[q0,Z0,q0] = 1 [q0,X q1] [q1 Z0 q0]
[q0,Z0,q1] = 1 [q0,X q0] [q0 Z0 q1].
[q0,Z0,q1] = 1 [q0,X q1] [q1 Z0 q1]

A → 1EA
A → 1FC
B → 1EB
B → 1FD

A → 1EA | 1FC
B → 1EB | 1FD

PDA TO CFG

step 2: IF $\delta(q,X,A)=(P,B1B2..........BN)$

[q,A,qm+1]->X [p,B1,q2]

[q2,B2,q3].....[qm,BN,Qm+1]

[q0,Z0,q0] - A
[q0,Z0,q1] - B
[q1,Z0,q0] - C
[q1,Z0,q1] - D
[q0,X,q0] E
[q0,X,q1] F
[q1,X,q0] G
[q1,X,q1] H

2. $\delta(q0,1,X)=(q0,XX)$

[q0,X,q0] → 1[q0,X,q0] [q0X q0]
[q0,X,q0] → 1[q0,X,q1] [q1X q0]
[q0,X,q1] → 1[q0,X,q0] [q0X q1]
[q0,X,q1] → 1[q0,X,q1] [q1,X q1]

E → 1EE
E → 1FG
F → 1EF
F → 1FH

E → 1EE | 1FG
F → 1EF | 1FH

PDA TO CFG

step 2: IF $\delta(q,X,A)=(P,B1B2..........BN)$

$[q,A,qm+1] \rightarrow X\ [p,B1,q2]$

$[q2,B2,q3].....[qm,BN,Qm+1]$

3. $\delta(q0,0,X)=(q1,X)$

$[q0,X,q0] \rightarrow 0\ [q1,X,q0]$

$[q0,X,q1] \rightarrow 0\ [q1,X,q1]$

$\boxed{\begin{array}{l} E \rightarrow 0G \\ F \rightarrow 0H \end{array}}$

5 $\delta(q1,0,Z0)=(q0,Z0)$

$[q1,Z0,q0] \rightarrow 0\ [q0,Z0,q0]$

$[q1,Z0,q1] \rightarrow 0\ [q0,Z0,q1]$

$\boxed{\begin{array}{l} C \rightarrow 0\ A \\ D \rightarrow 0\ B \end{array}}$

PDA TO CFG

step 3: IF $\delta(q,X,A)=(P, \varepsilon)$

$[q,A,p] \rightarrow X$

$S \rightarrow A|B$
$A \rightarrow 1EA|1FC|\varepsilon$
$B \rightarrow 1EB|1FD$
$E \rightarrow 1EE|1FG|0G$
$F \rightarrow 1EF|1FH|0H$
$C \rightarrow 0A$
$D \rightarrow 0B$
$H \rightarrow 1$

CFG

G is useless symbol

$S \rightarrow A|B$
$A \rightarrow 1EA|1FC|\varepsilon$
$B \rightarrow 1EB|1FD$
$E \rightarrow 1EE$
$F \rightarrow 1EF|1FH|0H$
$C \rightarrow 0A$
$D \rightarrow 0B$
$H \rightarrow 1$

E will not terminate with terminal

$S \rightarrow A|B$
$A \rightarrow 1FC|\varepsilon$
$B \rightarrow 1FD$
$F \rightarrow 1FH|0H$
$C \rightarrow 0A$
$D \rightarrow 0B$
$H \rightarrow 1$

$S \rightarrow A$
$A \rightarrow 1FC|\varepsilon$
$F \rightarrow 1FH|0H$
$C \rightarrow 0A$
$H \rightarrow 1$

4. $\delta(q1,1,X)= (q1, \varepsilon)$

$[q1, X, q1] \rightarrow 1$

$\boxed{H \rightarrow 1}$

6. $\delta(q0, \varepsilon, Z0)=(q0, \varepsilon)$

$[q0, Z0, q0] \rightarrow \varepsilon$

$\boxed{A \rightarrow \varepsilon}$

Final CFG

$S \rightarrow A$
$A \rightarrow 1FC|\varepsilon$
$F \rightarrow 1FH|0H$
$C \rightarrow 0A$
$H \rightarrow 1$

PDA TO CFG

$M = \{\{q0,q1\}, \{a,b\}, \{X,Z0\}, \delta, q0, Z0, \Phi\}$

1. $\delta(q0,a,Z0)=(q0,XZ0)$
2. $\delta(q0,a,X)=(q0,XX)$
3. $\delta(q0,b,X)=(q1,\varepsilon)$
4. $\delta(q1,\varepsilon,X)=(q1,\varepsilon)$
5. $\delta(q1,b,x)=(q1,\varepsilon)$
6. $(q1,\varepsilon,Z0)=(q1,\varepsilon)$

[q0, Z0, q0] — A
[q0, Z0, q1] — B
[q1, Z0, q0] — C
[q1, Z0, q1] — D
[q0, X, q0] — E
[q0, X, q1] — F
[q1, X, q0] — G
[q1, X, q1] — H

Step 1
S → [q0, Z0, q0]
S → [q0, Z0, q1]

$\boxed{S \to A|B}$

1. $\delta(q0, a, Z0) = (q0, XZ0)$

[q0, Z0, q0] → a [q0, X, q0] [q0, Z0, q0] A → a E A
[q0, Z0, q0] → a [q0, X, q1] [q1, Z0, q0] A → a F C
[q0, Z0, q1] → a [q0, X, q0] [q0, Z0, q1] B → a E B
[q0, Z0, q1] → a [q0, X, q1] [q1, Z0, q1] B → a F D

$\boxed{\begin{array}{l} A \to aEA|aFC \\ B \to aEB|aFD \end{array}}$

2. $\delta(q0, a, X) = (q0, XX)$

[q0, X, q0] → a [q0, X, q0] [q0, X, q0] E → a E E
[q0, X, q0] → a [q0, X, q1] [q1, X, q0] E → a F G
[q0, X, q1] → a [q0, X, q0] [q0, X, q1] F → a E F
[q0, X, q1] → a [q0, X, q1] [q1, X, q1] F → a F H

E → aEE | aFG
F → aEF | aFH

3. $\delta(q0, b, X) = (q1, \varepsilon)$

[q0, X, q1] → b

F → b

4. $\delta(q1, \varepsilon, X) = (q1, \varepsilon)$

[q1, X, q1] → ε

H → ε

6. $\delta(q1, \varepsilon, Z0) = (q1, \varepsilon)$

[q1, Z0, q1] → ε

D → ε

5. $\delta(q1, b, X) = (q1, \varepsilon)$

[q1, X, q1] → b

H → b

CFL TO CFG THEN TO PDA

Construct a CFG G corresponding to the following context free language then construct PDA Corresponding to G

$L=\{0^n 1^{2n} \mid n>=1\}$

$S \to 0S11 \mid 011$

$V = \{S\}$

$P = \{S \to 0S11, \; S \to 011\}$

$T = \{0, 1\}$

$S = \{S\}$

PDA

$\delta(q, \varepsilon, S) = (q, \underline{0S11})$

$\delta(q, \varepsilon, S) = (q, \underline{011})$

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

$\delta(q, 1, 1) = (q, \varepsilon)$

$\{ \underset{Q}{\;} \} \; \underset{\Sigma}{\{0,1\}} \; \underset{\Gamma}{\{0,1,S\}} \; \underset{Z}{\{S\}}, \delta$

001111 – this string is accepted or not

$\delta(q, 001111, S) \vdash (q, 001111, 0S11)$

$\vdash (q, 01111, S11)$

$(q, 01111, 01111)$

$(q, 1111, 1111)$

$(q, 111, 111)$

$(q, 11, 11)$

$(q, 1, 1)$

$(q, \varepsilon, \varepsilon)$

Acceptance by empty stack

CFL TO CFG THEN TO PDA

Construct a CFG G corresponding to the following context free language then construct PDA Corresponding to G

$L=\{a^n b^m c^k \; n=m \text{ or } m<=k\}$

$S \to TC \mid AR$
$T \to aTb \mid \varepsilon$
$C \to cC \mid \varepsilon$
$A \to aA \mid \varepsilon$
$R \to bRc \mid C$

$\delta(q, \varepsilon, S) = (q, TC)$
$\delta(q, \varepsilon, S) = (q, AR)$
$\delta(q, \varepsilon, T) = (q, aTb)$
$\delta(q, \varepsilon, T) = (q, \varepsilon)$
$\delta(q, \varepsilon, C) = (q, cC)$
$\delta(q, \varepsilon, C) = (q, \varepsilon)$
$\delta(q, \varepsilon, A) = (q, aA)$
$\delta(q, \varepsilon, A) = (q, \varepsilon)$
$\delta(q, \varepsilon, R) = (q, bRC)$
$\delta(q, \varepsilon, R) = (q, C)$

$\delta(q, a, a) = (q, \varepsilon)$
$\delta(q, b, b) = (q, \varepsilon)$
$\delta(q, c, c) = (q, \varepsilon)$

$Q = \{q\}$
$\Sigma = \{a, b, c\}$
$q_0 = q$
$F = \phi$
δ
$\Gamma = \{a, b, c, S, T, C, A, R\}$

$L = \{a^n b^n \; n \geq 1\}$
PDA ⟶ CFG 7Marks

Decision properties of CFL

There are four decision properties of CFL

Emptiness problem

Finiteness problem

Membership problem

➢Emptiness Problem

Remove all the useless symbol

S->XY

X->AA/AX

A->a

B->BY

Y->b

X→c

S→XY
X→AA|AX
A→a
Y→b
X→c

S
XY
AAY
aac

S
Ax
ac

S->aSb/Sb/Sa

S
aSb Sb Sa

Empty Grammar
No language is Generated by this Grammar

Membership problem

Check whether a particular string is generated by CFG

CYK algorithm

CYK is applicable on CNF only

S->AB

A->BB/a

B->AB/b

Check whether a string abbb is a valid member of following CFG

	4	3	2	1
1	S,B	A	S,B	A
2	S,B	A	B	
3	A	B		
4	B			

12 23 34
a b b b b b.
11 22 22 33 33 44
A B B B B B
A

13
123
(12),(3) (12) (33) (S,B) (B) SB ϕ
(1) (23) (11 (23) A A ϕ BB → A

(2,4)
(2 3 4)
(22) (3,4) B A ϕ
(2,3) (44) A B S,B

(14)
(1234) AS ϕ
(11) (24) = A (S,B) AB (S,B)
(12) (34) = (S,B) (A) SA ϕ
BA ϕ
(13) (44) = AB (S,B)

Finiteness problem

1 simplify the grammar

2Convert it into CNF Chomsky Normal form

3. Construct the CNF graph

4. If the graph contain any loop or cycle then the grammar generates infinite language

$S \to AB$

$A \to BC/a$

$B \to CC/b$

$C \to a$

1. No unit production $A \to B$ $B, A \in V$
2. epsilon production - No $A \to \varepsilon$
3. useless Symbol - NO
4. CNF $A \to BC$, $A \to a$ (Already in CNF)

So this finite Lag

5.

(No-Cycle loop is available)

Thank You