

3/3/18

UNIT-5

TURING MACHINE

Turing machine contains 7 tuples ${}^{TM} \times (Q, \Sigma, \Gamma, \delta, q_0, F, B)$

where

Q = set of states

Σ = set of input symbols

Γ = set of input tape symbols

δ = Transition function

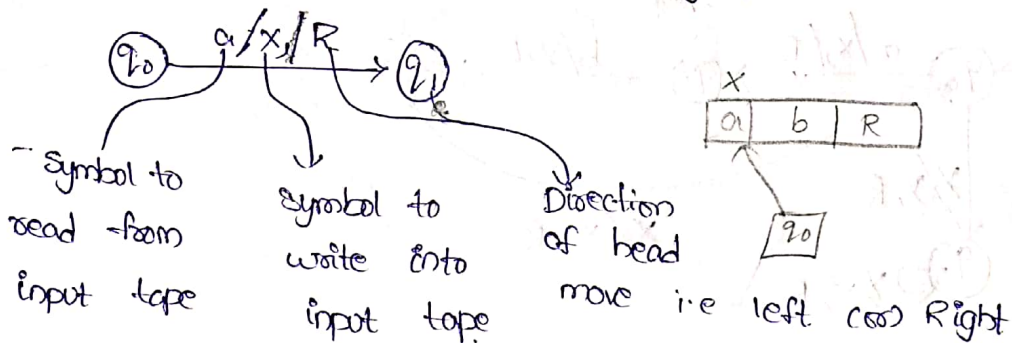
q_0 = initial state

F = final state

B = Blank symbol.

where δ can be defined as $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L/R\}$

The Transition Diagram:- The transition function can be represented in the form of graphical notation.



1) Design a Turing machine for $L = 01^*0$

$L = 01^*0$

$0(1^*)0$

$L = \{00, 010, 0110, 01110, \dots\}$

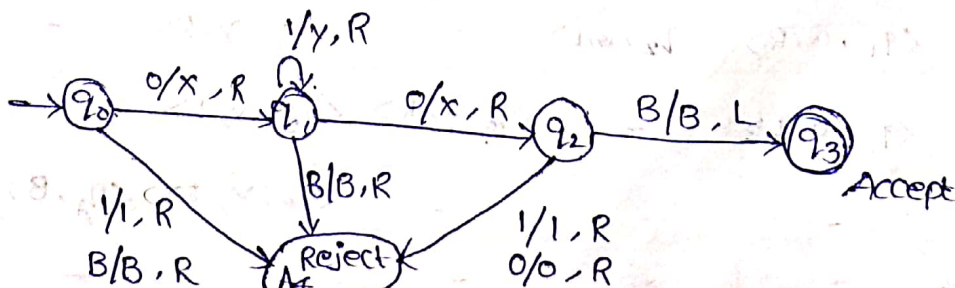
$\begin{matrix} x & x & B \\ 0 & 0 & B \end{matrix}$

$\begin{matrix} x & y & x & B \\ 0 & 1 & 0 & B \end{matrix}$

$\begin{matrix} x & y & y & x & B \\ 0 & 1 & 1 & 1 & B \end{matrix}$

$\begin{matrix} 0 & 1 & 1 & 0 & B \end{matrix}$

$x \quad y \quad y \quad x \quad B$

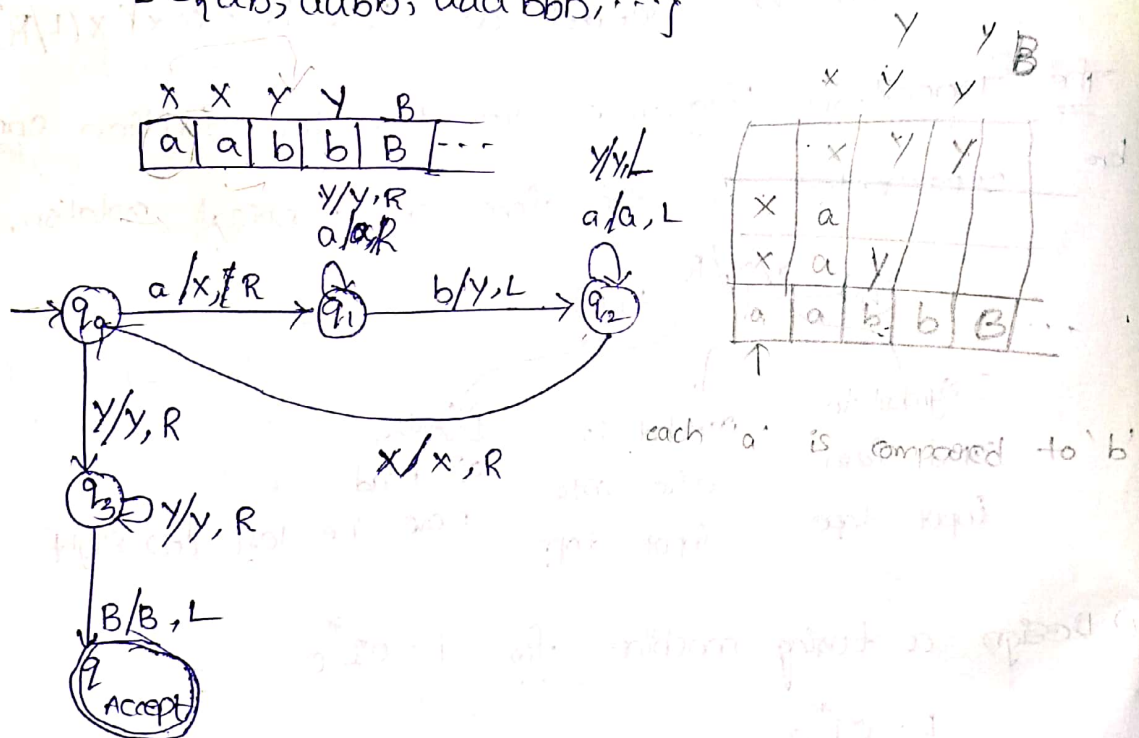


Tape Symbol State	0	1	x	y	B
q_0	$\langle q_1, x, R \rangle$	$\langle q_4, 1, R \rangle$	-	-	$\langle q_4, B, R \rangle$
q_1	$\langle q_2, x, R \rangle$	$\langle q_1, y, R \rangle$	-	-	$\langle q_4, B, R \rangle$
q_2	$\langle q_4, 0, R \rangle$	$\langle q_4, 1, R \rangle$	-	-	$\langle q_3, B, L \rangle$
q_3	-	-	-	-	-
q_4	-	-	-	-	-

*
**

2) Design a Turing Machine for language $L = \{a^n b^n / n \geq 1\}$

$L = \{ab, aabb, aaabbb, \dots\}$

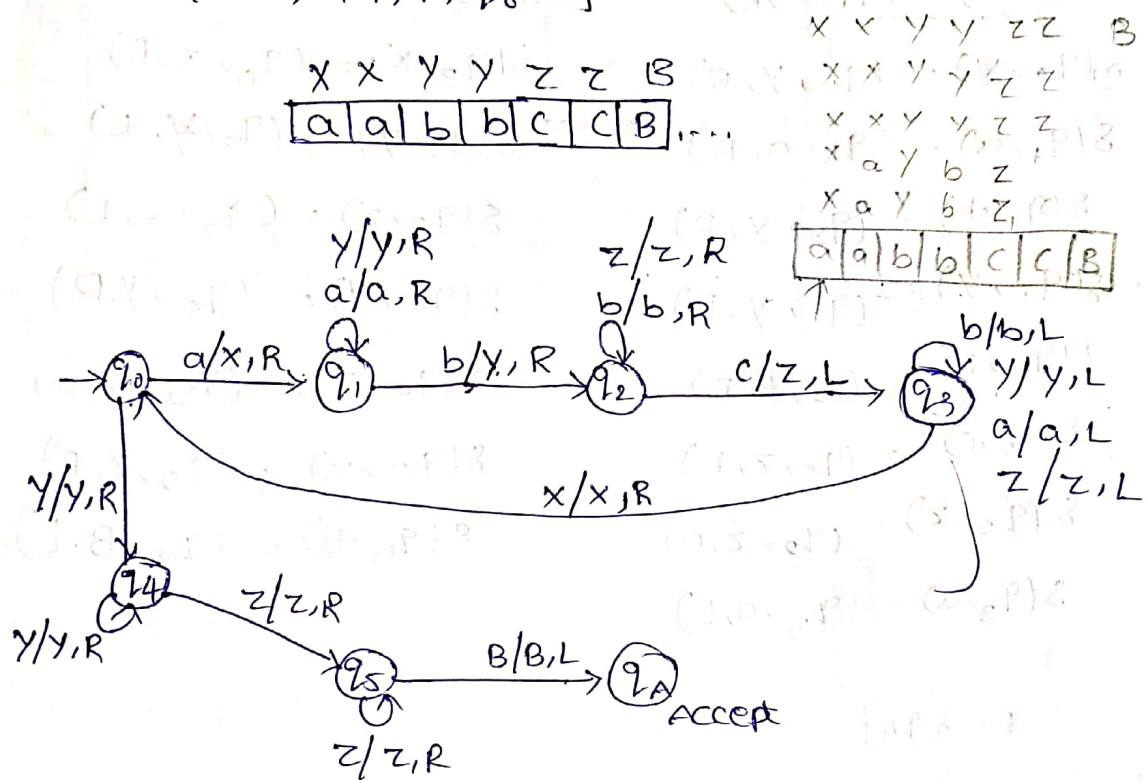


Tape Symbol State	a	b	x	y	B
$\rightarrow q_0$	$\langle q_1, x, R \rangle$	-	-	$\langle q_3, y, R \rangle$	-
q_1	$\langle q_1, a, R \rangle$	$\langle q_2, b, L \rangle$	-	$\langle q_1, y, R \rangle$	-
q_2	$\langle q_2, a, L \rangle$	-	$\langle q_0, x, R \rangle$	$\langle q_2, y, L \rangle$	-
q_3	-	-	-	$\langle q_3, y, R \rangle$	$\langle q_4, B, L \rangle$
* q_{Accept}	-	-	-	-	-

8
Design Turing machine for $L = \{a^n b^n c^n / n \geq 1\}$

$L = \{abc, aabbcc, aaabbbccc, \dots\}$

$TM = \{Q, \Sigma, \Gamma, \delta, q_0, B\}$



Tape symbol / state	a	b	c	x	y	z	B
$\rightarrow q_0$	$\langle q_1, x, R \rangle$	-	-	-	$\langle q_4, y, R \rangle$	-	-
q_1	$\langle q_1, a, R \rangle$	$\langle q_2, y, R \rangle$	-	-	$\langle q_1, y, R \rangle$	-	-
q_2	-	$\langle q_2, b, R \rangle$	$\langle q_3, z, L \rangle$	-	-	$\langle q_2, z, R \rangle$	-
q_3	$\langle q_3, a, L \rangle$	$\langle q_3, b, L \rangle$	-	$\langle q_0, x, R \rangle$	$\langle q_3, y, L \rangle$	$\langle q_3, z, L \rangle$	-
q_4	-	-	-	-	$\langle q_4, y, R \rangle$	$\langle q_5, z, R \rangle$	-
q_5	-	-	-	-	-	$\langle q_5, z, R \rangle$	$\langle q_A, B, L \rangle$
q_A	-	-	-	-	-	-	-

TM: $M = \{Q, \Sigma, \Gamma, \delta, q_0, B, F\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_A\}$

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

$\Gamma = \{a, b, c, x, y, z, B\}$

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

$\delta(q_3, b) = (q_3, b, L)$

$\delta(q_0, y) = (q_4, y, R)$

$\delta(q_3, x) = (q_0, x, R)$

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

$\delta(q_3, y) = (q_3, y, L)$

$\delta(q_1, b) = (q_2, y, R)$

$\delta(q_3, z) = (q_3, z, L)$

$\delta(q_1, y) = (q_1, y, R)$

$\delta(q_4, y) = (q_4, y, R)$

$\delta(q_2, b) = (q_2, b, R)$

$\delta(q_4, z) = (q_5, z, R)$

$\delta(q_2, c) = (q_3, z, L)$

$\delta(q_5, z) = (q_5, z, R)$

$\delta(q_2, z) = (q_2, z, R)$

$\delta(q_5, B) = (q_A, B, L)$

$\delta(q_3, a) = (q_3, a, L)$

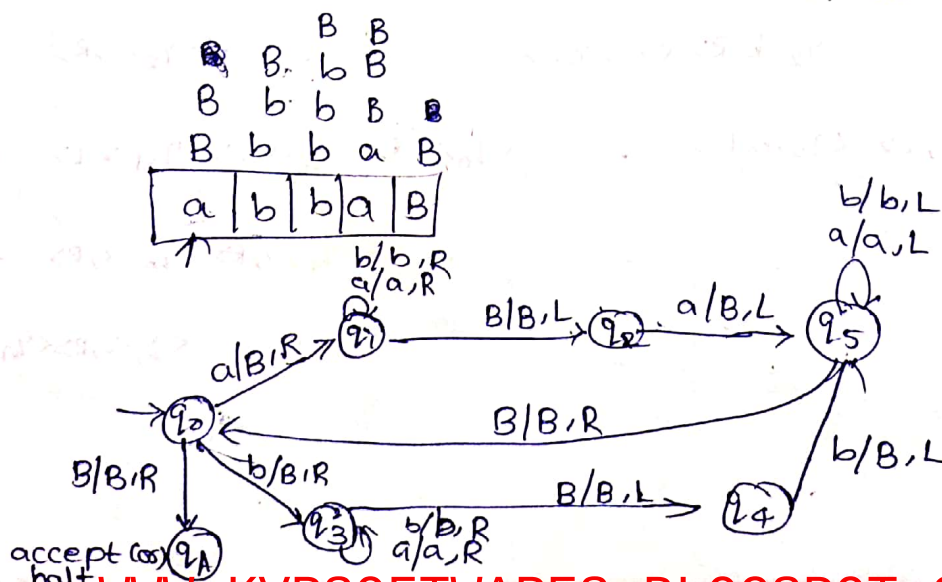
$F = \{q_A\}$

4) Design Turing Machine for $L = \{ww^R \mid w \in (a, b)^*\}$

Given $L = \{ww^R \mid w \in (a, b)^*\}$

It is a even length palindrome

$L = \{aa, bb, abba, baab, abbbba, abaabba, \dots\}$



Tape Symbol State	a	b	B
$\rightarrow q_0$	$\langle q_1, B, R \rangle$	$\langle q_3, B, R \rangle$	$\langle q_A, B, R \rangle$
q_1	$\langle q_1, a, R \rangle$	$\langle q_1, b, R \rangle$	$\langle q_2, B, L \rangle$
q_2	$\langle q_5, B, L \rangle$	-	-
q_3	$\langle q_3, a, R \rangle$	$\langle q_3, b, R \rangle$	$\langle q_4, B, L \rangle$
q_4	-	$\langle q_5, B, L \rangle$	-
q_5	$\langle q_5, a, L \rangle$	$\langle q_5, b, L \rangle$	$\langle q_0, B, R \rangle$
* q_A	-	-	-

1D abba

q_0 a b b a B
 $\vdash$ B q_1 b b a B
 $\vdash$ B b q_1 b a B
 $\vdash$ B b b q_1 a B
 $\vdash$ B b b a q_1 B
 $\vdash$ B b b q_2 a B
 $\vdash$ B b q_5 b B B
 $\vdash$ B q_5 b b B B
 $\vdash$ q_5 B b b B B
 $\vdash$ B q_0 b b B B
 $\vdash$ B B q_3 b B B
 $\vdash$ B B b q_3 B B
 $\vdash$ B B q_4 b B B
 $\vdash$ B q_5 B B B B
 $\vdash$ B B q_0 B B B
 $\vdash$ B B B q_A B B
 Accept

5) Design Turing Machine for 'parity Counter' that outputs '0' (or) '1' depending on w

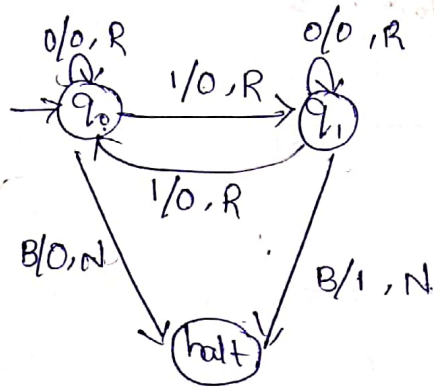
1's odd - 1

1's even - 0

0 0 0 1
0 1 0 B

0 0 0 0 0 0 0
0 1 1 0 1 1 B

Transition Table



odd 1's 010

→ q₀ 0 1 0 B

→ q₀ 1 0 B

→ q₀ 0 B

→ q₁ B

→ 1 halt

even 1's 1010

→ q₀ 1 0 1 0 B

→ q₁ 0 1 0 B

→ q₁ 1 0 B

→ q₀ 0 B

→ q₀ B

→ halt

Design TM for 2's complement

I/p 10010

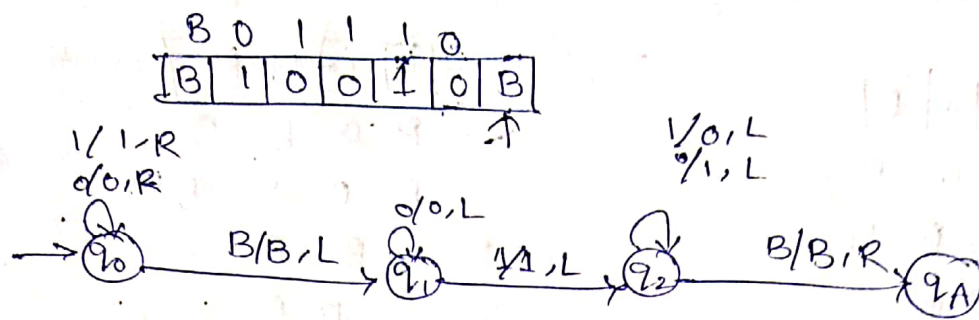
1's 01101

+1
2's 01110

MSB
B 1 0 0 1 0 B
LSB

accept
halt

First of all we have to move LSB then upto



$I_D: 810010B$

$H B q_0 10010B$

$H B 1 q_0 0010B$

$H B 1 0 q_0 010B$

$H B 1 0 0 q_0 10B$

$H B 1 0 0 1 q_0 0B$

$H B 1 0 0 1 0 q_0 B$

$H B 1 0 0 1 q_1 0B$

$H B 1 0 0 q_1 1 0B$

$H B 1 0 q_2 0 1 0B$

$H B 1 q_2 0 1 1 0B$

$H B q_2 1 1 1 0B$

$H B B 0 1 1 0B$

$H q_2 B 0 1 1 0B$

$H B q_2 0 1 1 0B$

(Addition of two integers. Design TM)
 Design Turing Machine for to accept set of all the
 palindromes.

Sol:-

a	b	a	B
---	---	---	---

B b a B

B b B B

B B B

halt

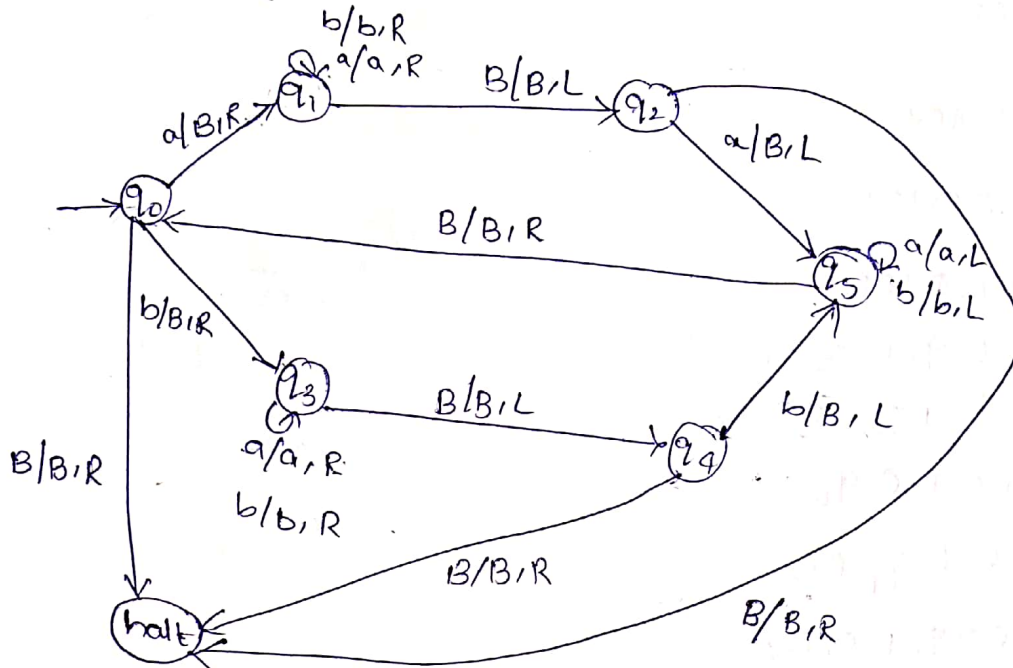
b	a	b	B
---	---	---	---

B a b B

B a B B

B B B

B B
 halt



bab

⊢ q₀ babB

⊢ B q₃ abB

⊢ B a q₃ bB

⊢ B a b q₃ B

⊢ B a q₄ bB

⊢ B q₅ a B B

⊢ q₅ B a B B

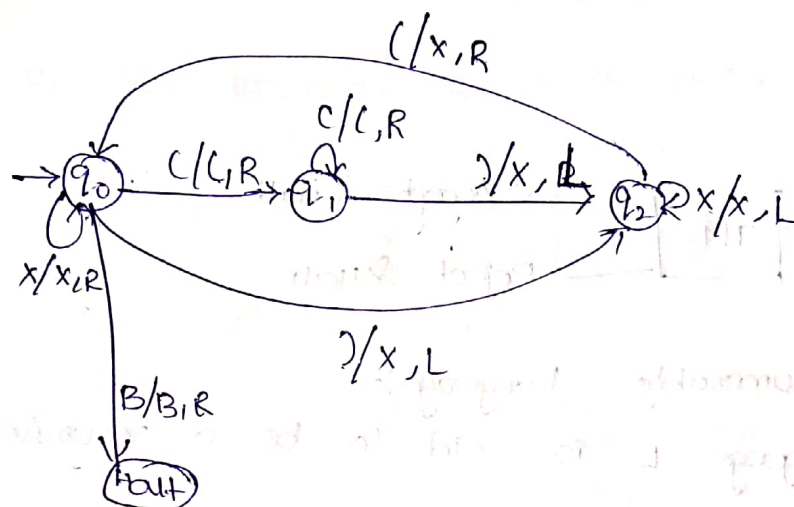
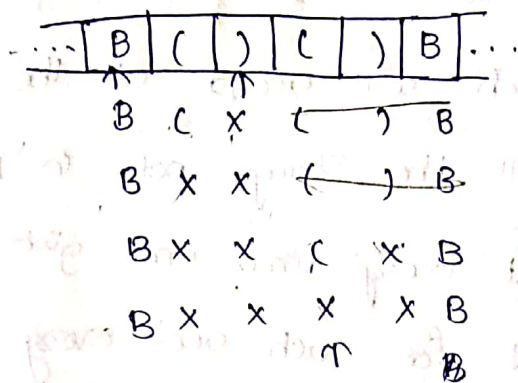
⊢ B q₀ a b B

⊢ B B q₁ B B

⊢ B q₂ B B B

⊢ B B q₄ B B

Design Turing Machine for parenthesis checking



$\vdash q_0 ((()) B$

$(q_1 ()) B$

$((q_1)) B$

$(((q_2 x)) B$

$(q_2 (x)) B$

$(x q_0 x) B$

$(x x q_0) B$

$(x x q_0 x x B$

$(q_2 x x x B$

$q_2 (x x x B$

$x q_0 x x x B$

$x x q_0 x x B$

$x x x q_0 x B$

$x x x x q_0 B$

$x x x x B \vdash$

Types of Grammars - Chomsky Hierarchy:

Linguist Noam Chomsky defined a hierarchy of languages in terms of complexity. This four level hierarchy, called the Chomsky hierarchy, corresponds to four classes of machines.

The Chomsky hierarchy classifies grammars according to form of their productions into the following four levels.

- (1) Type 0 grammars - unrestricted grammar
- (2) Type 1 grammars - context sensitive grammar
- (3) Type 2 grammars - context free grammar
- (4) Type 3 grammars - regular grammar.

(1) Type - 0 grammars - Unrestricted Grammars (URG)

These grammars include all formal grammars. In URGs, all the productions are of the form $\alpha \rightarrow \beta$, where α and β may have any number of terminals and non-terminals, i.e., no restrictions on either side of productions.

Every grammar is included in it if it has at least one non-terminal on the left hand side.

Ex:-
 $AA \rightarrow abCB$
 $AA \rightarrow bAA$
 $bA \rightarrow a$
 $S \rightarrow aAb|C$

or ~~(AA) $\rightarrow$ (abCB)~~

They generate exactly all languages that can be recognized by a Turing machine. The language that is recognized by a Turing machine is defined as set of all the strings on which it halts. These languages

Scanned by CamScanner

are also known as the recursively enumerable languages. ⁽²⁾

(2) Type 1 grammar - Context Sensitive Grammar: (CSG)

These grammars define the context-sensitive languages.

In Context Sensitive grammar, all the productions are of the form $\alpha \rightarrow \beta$, where length of α is less than or equal to the length β , i.e. $|\alpha| \leq |\beta|$. α and β may have any number of terminals and non-terminals.

These languages are exactly all the languages that can be recognized by linear bound automata.

(2)

Ex:- $|\alpha| \leq |\beta| \quad \alpha \rightarrow \beta$

$aAbcD \rightarrow abcDbcD$

(3) Type 2 Grammar - Context-free Grammar (CFG)

These grammars define the context-free languages.

These are defined by rules of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$ where $|\alpha| = 1$ and α is non-terminal and β is a string of terminals and non-terminals.

We can replace α by β regardless of where it appears.

Hence the name context free grammar.

These languages are exactly those languages that can be recognized by a pushdown automaton.

CFG Context free languages defines the syntax of all programming languages.

Ex:- $\begin{aligned} &S \rightarrow aS \mid Sa \mid a \\ &S \rightarrow aAA \mid bBB \mid \epsilon \end{aligned}$

Scanned by CamScanner

(4) Type 3 grammars - regular grammars:

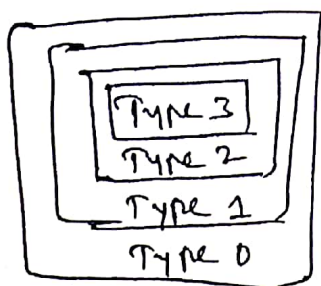
These grammars generate the regular languages. Such a grammar restricts its rules to a single non-terminal on the LHS. The RHS consists of either a single terminal or string of terminals with single non-terminal on left or right end.

Ex: $A \rightarrow aA|a$ — right linear grammar $\alpha \rightarrow \beta, \alpha \in \{V\}^+, \beta \in V\{T\}^+$
 $A \rightarrow Aa|a$ — left linear grammar.

* Every regular language is context free, every context-free language is context-sensitive and every context-sensitive language is recursively enumerable.

Table: Chomsky's hierarchy

<u>Grammar</u>	<u>Language</u>	<u>Automaton</u>	<u>production rules</u>
Type 0	Recursively enumerable	Turing machine	$\alpha \rightarrow \beta$ no restrictions on α, β α should have at least one non-terminal
Type 1	Context-sensitive	Linear bounded Automate	$\alpha \rightarrow \beta$ $ \alpha \leq \beta $
Type 2	Context-free	Push down automata	$\alpha \rightarrow \beta$ $ \alpha = 1$
Type 3	Regular	Finite state automaton	$\alpha \rightarrow \beta,$ $\alpha \in \{V\}^+, \beta \in V\{T\}^+$ $\beta = T^+ \{V\}$ $= T^+$



Chomsky hierarchy of grammars

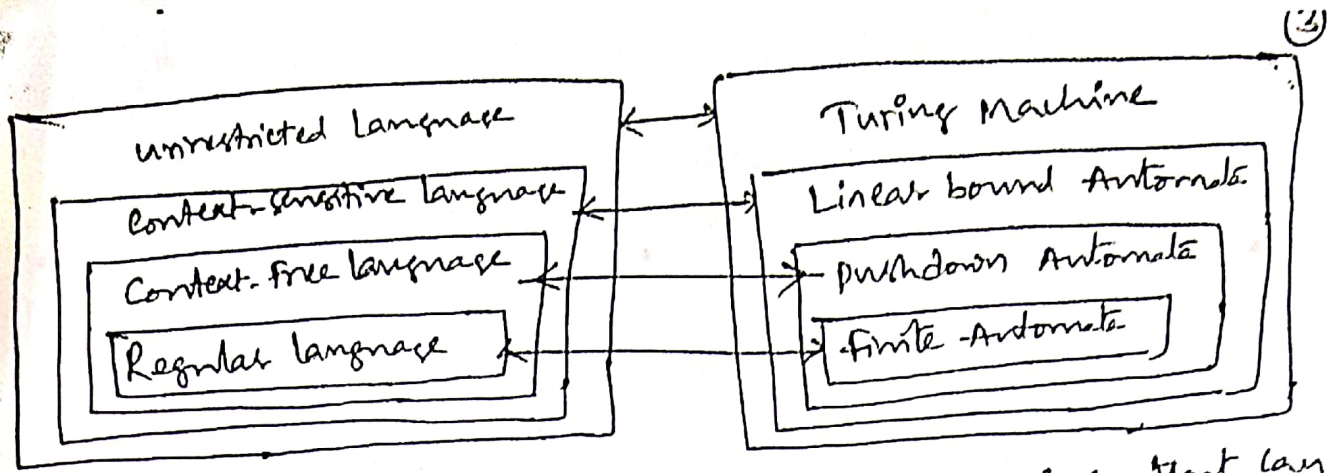


fig: The hierarchy of languages and the machine that can recognize the same is shown above fig.

- Every RL is context free, every CFL is context-sensitive and every CSL is unrestricted. So the family of regular language can be recognized by any machine.
- CFLs are recognized by pushdown automata, linear bounded automata and Turing machines.
- CSLs are recognized by Linear bounded automata and Turing machines.
- Unrestricted languages are recognized by only Turing machines.

(4)

Push Down Automata (PDA)

①

PDA - FA + Stack
memory element

PDA = $(Q, \Sigma, \delta, q_0, Z_0, F, \Gamma)$
(T.M.)

Q = finite set of states

Σ = input symbols

δ = Transition function

q_0 = initial state

Z_0 = Bottom of the stack

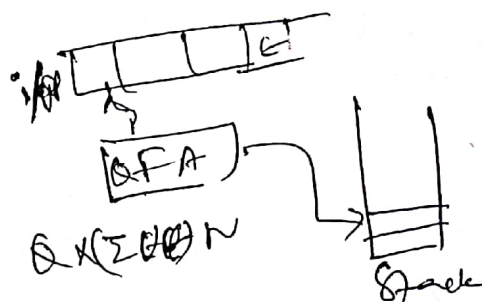
F = set of final states

Γ = stack alphabet.

PDA
↓
DPDA NPDA

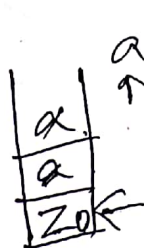
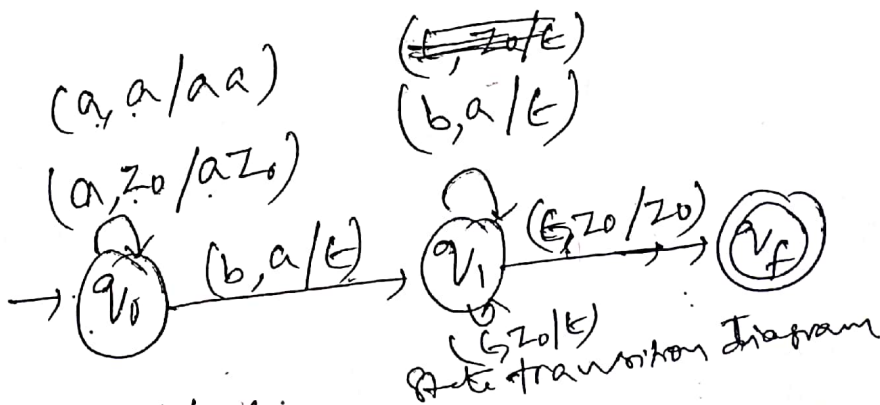
DPDA: $\delta: Q \times \underbrace{\{\Sigma \cup \epsilon\}}_{\text{input}} \times \underbrace{N}_{\text{top}} \rightarrow \underbrace{Q \times \Gamma}_{\text{top}}^*$

NPDA: $\delta: Q \times \underbrace{\{\Sigma \cup \epsilon\}}_{\text{input}} \times \underbrace{N}_{\text{top}} \rightarrow \underbrace{2}_{= (Q \times \Gamma)^*}$



Ex: $a^n b^n | n \geq 1$

a a b b ϵ
↑ ↑ ↑ ↑ ↑



DPDA

transitions
function

final state

Empty stack

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

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

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

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

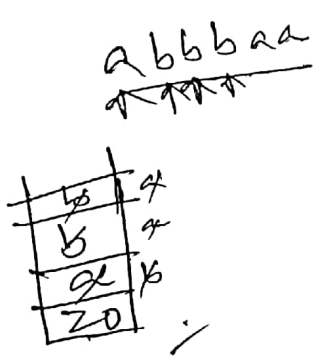
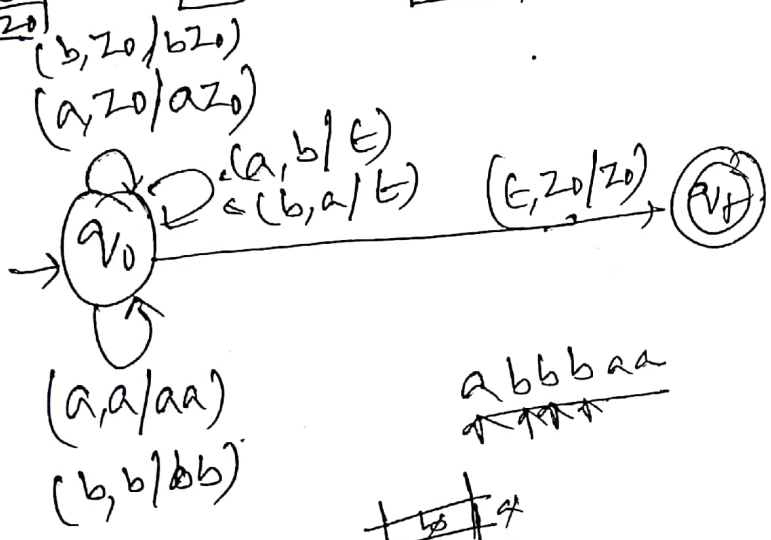
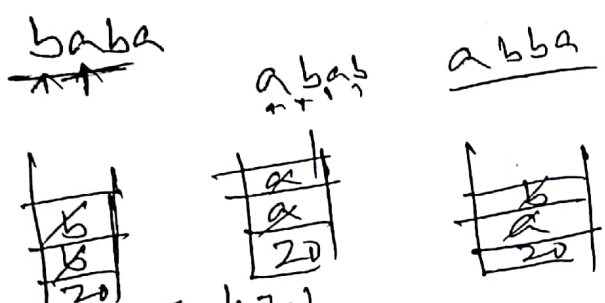
$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0) \text{ or } (q_1, \epsilon)$

accept by
final state

Acceptance by
empty stack

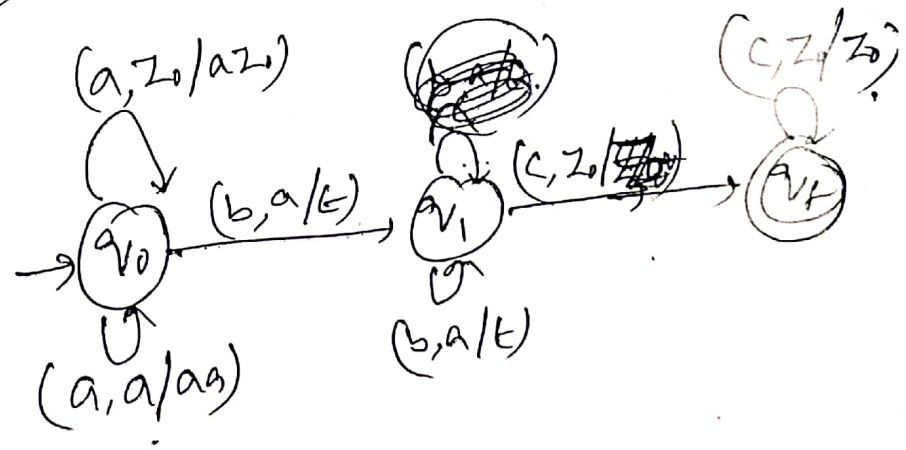
$|w|/n_a(w) = n_b(w)$ [number of a's must be even],
DPDA

Ex: a b b b a a
 a a b b b a b a
 a b a b



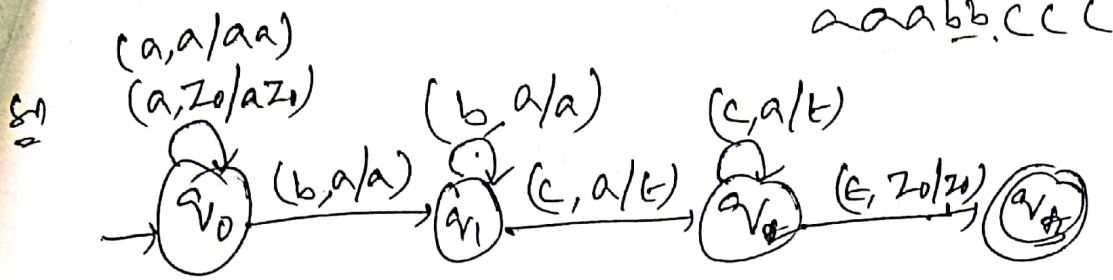
$a^n b^n c^m \mid n, m \geq 1$

DPDA
 $a^n \rightarrow \text{push}$
 $b^n \rightarrow \text{pop}$
 $c^m \rightarrow \text{in final state}$

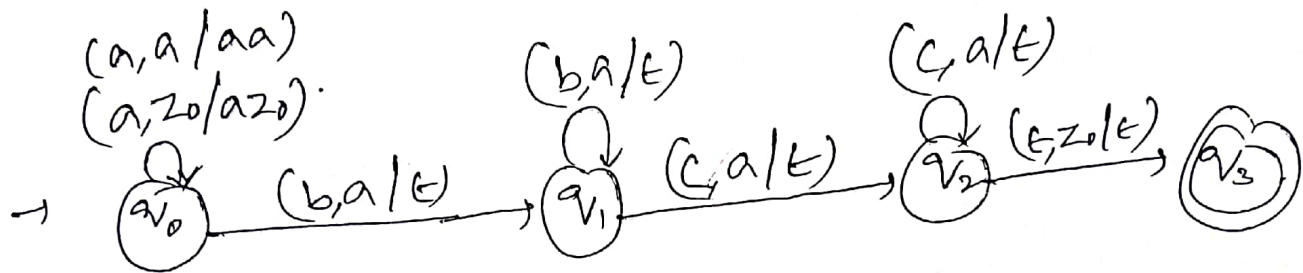


$a^n b^m c^n \mid n, m \geq 1$ don't want cont ≥ 2

(2)

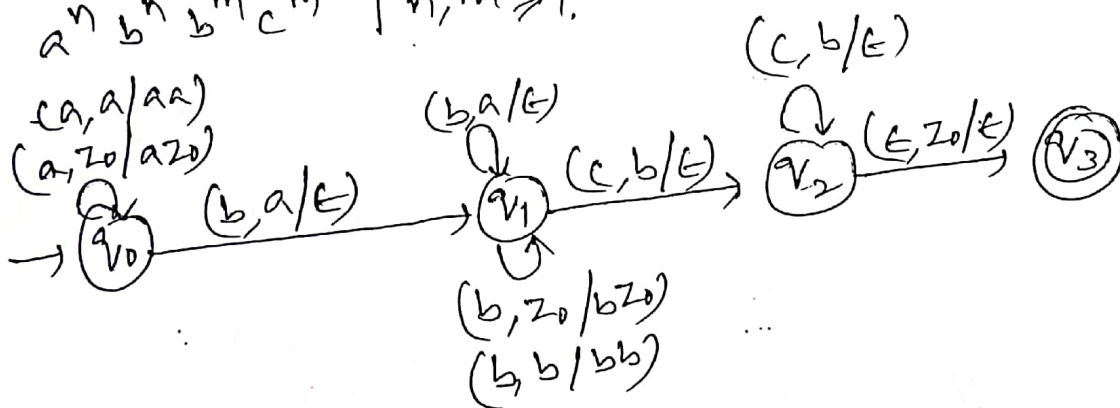


$a^{m+n} b^m c^n \mid m, n \geq 1$

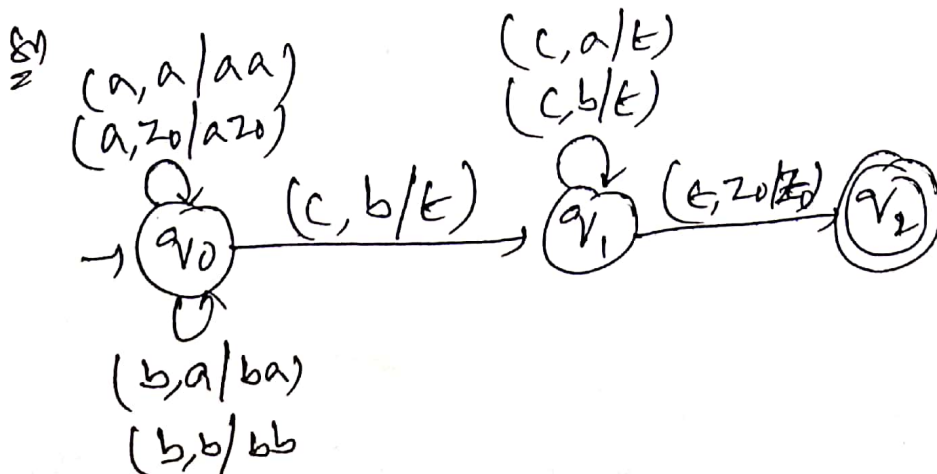


$a^n b^{m+n} c^m \mid n, m \geq 1$

Ex) $a^n b^n b^m c^m \mid n, m \geq 1$



$a^n b^m c^{n+m} \mid n, m \geq 1$



$a^n b^n c^m d^m \mid n, m \geq 1$

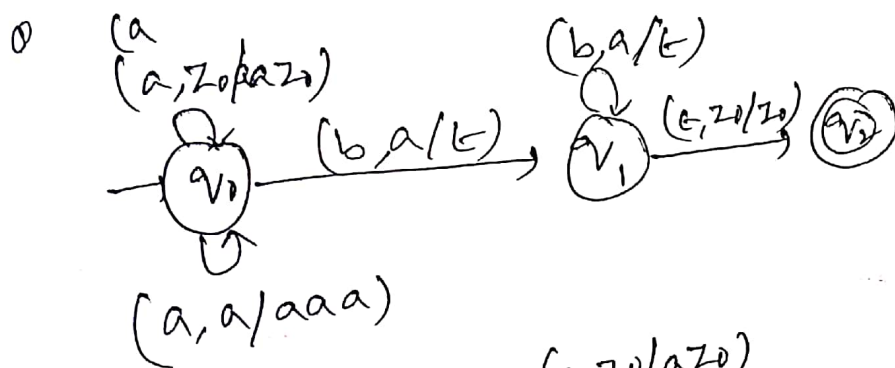
$a^n b^m c^m d^n \mid n, m \geq 1$

$a^n b^m c^n d^m \mid n, m \geq 1$ X is not CF
not PDA

$a^n 2^n \mid n \geq 1$

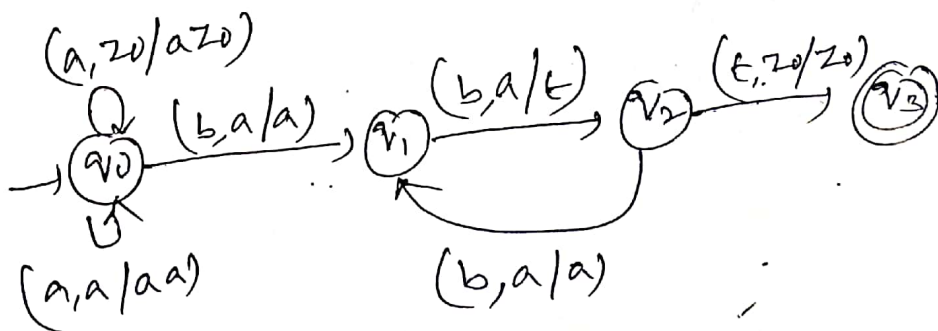
$a^n b^{2n} \mid n \geq 1$

1. $abb, aabbbb, aaabbbbb, \dots$
two solutions $\leftrightarrow$ for every a - push two a's.
for every b - pop two b's

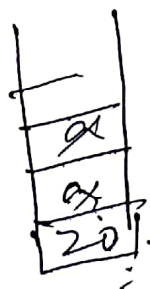


$bb, bbb, \dots$

for every b



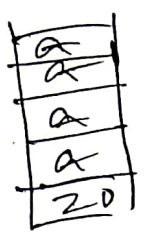
$aaabbbb$



Ex: $a^n b^n c^n \mid n \geq 1$ X not a PDA

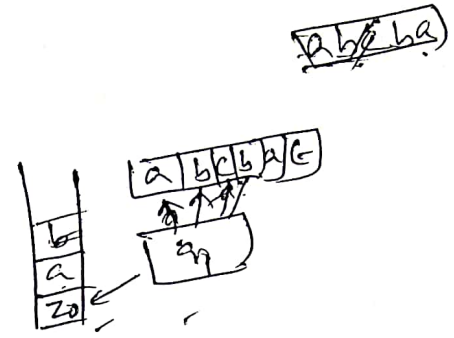
one possibility

X $aa\ bb\ cc$
 $\uparrow\ \uparrow\ \uparrow\ \uparrow\ \uparrow\ \uparrow$

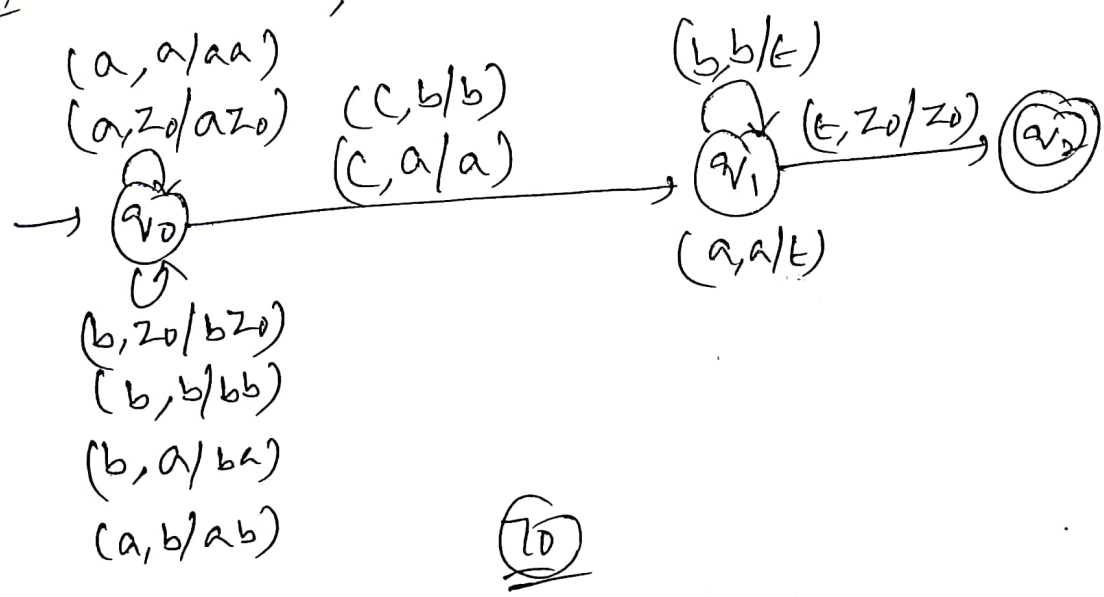


for every a - two a's push

Ex $WCW^R \mid w \in (a,b)^+$



Ex $abcba, abbcbba$

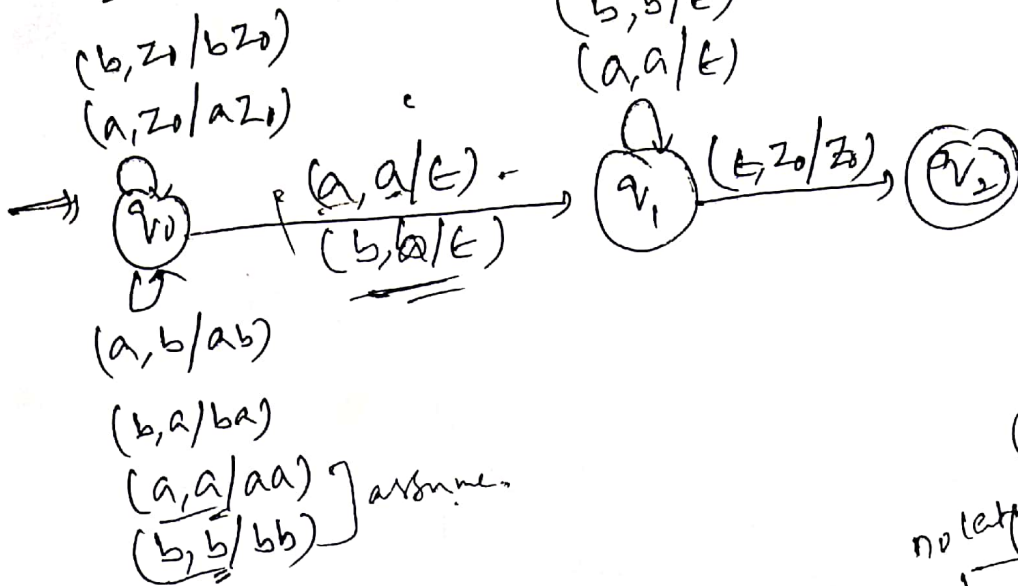


Ex $WW^R \mid w \in (a+b)^+$ whenever

Ex $\frac{abaaba}{w \quad w^R}$



NDPDA $ww^R \mid w \in (a+b)^+$



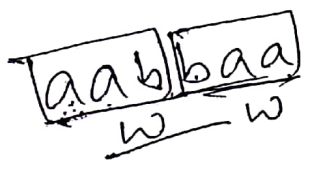
abba
Instantaneous Description

~~aa a~~
~~Instantaneous Description~~
 $\delta(q_0, aaaa, z_0)$
 $\delta(q_0, aaa, az_0)$

no letter / centre
 (q_0, aa, aaz_0) (q_1, aa, z_0) X

no letter / centre
 (q_0, a, aaz_0) (q_1, a, az_0)
no letter / centre
 (q_0, ϵ, aaz_0) (q_0, ϵ, az_0) X
 (q_1, ϵ, z_0)
 (q_2, z_0) ✓

~~abab~~



baa aab



~~a a~~ / ~~a a~~
 $w \mid \dots w^R$

NDPAA

Center has pop come
center not come push



Change $\rightarrow$ getting $\rightarrow$ centre

a b / b a