FA + stack

PDA programming -

Page No.
Date :

PDA - Machine $M(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
- 7 tupled

- Q : finite set of states
- Σ : input alphabet
- Γ : stack alphabet (Γ can be same as Σ or may not be.)
- δ : transition function
- q_0 : start state
- Z_0 : Start stack symbol / Empty stack symbol
- F : set of final states

When stack is empty Z will be at top of stack.

$Z_0 \in \Gamma$

F can be empty.

PDA programming -
1. final state Method
2. Empty stack Method
   - Both have same power
   - Both are interconvertable
   - Both can represent any CFL

δ is partial function i.e in DPDA i.e you need not to specific move for each combination ie why DC allowed but in DFA, δ is total function.

Empty stack Method - No final state is specified
if $F = \phi$
- it may be empty stack method of a language ϕ

δ for PDA -

DPDA - $\delta(q_0, a, b) = \cancel{\delta_0}\ (q_1, ab)$ → top of stack (push a)

- current state
- input symbol
- top of stack
- Next state after inputting a

δ of q_0 =

- Both DPDA & NPDA allow ϵ-moves.

EXCELLENT

Γ* - infinite set of finite sized strings.

Page No.
Date :

$\delta_{Q \times (\Sigma \cup \epsilon) \times \Gamma^*} \rightarrow \delta_{Q \times \Gamma^*}$ (ε- move spontaneous jump without reading I/P symbol)

$\therefore \delta_{Q \times \Sigma \times \Gamma^*} \rightarrow \delta_{Q \times \Gamma^*}$ (as you can put any no of symbol)

NPDA - $\delta(q_0, a, b) = \{(q_1, ab), (q_2, \epsilon)\}$

NPDA can do nondeterministic step by making two copies of the machine.

push a pop b

$\therefore \delta_{Q \times (\Sigma \cup \epsilon) \times \Gamma} \rightarrow 2^{Q \times \Gamma^*}$

Γ* will allow infinite subsets as

$\{[a^n b^n] \cup [a^n b^{2n}] \cup [a^n b^{3n}] \cup \ldots\}$

∴ it is written as

$\delta_{Q \times (\Sigma \cup \epsilon) \times \Gamma} \rightarrow$ finite subsets of $Q \times \Gamma^*$

or

$(\text{finite}) 2^{Q \times \Gamma^*}$

Infinite sets are not valid as PDA can only make finite no of copies.

[$\delta(q_0, a, b) = (q_1, a^*)$] - invalid because it must have some limit

[$\delta(q_0, \epsilon, b) = (q_1, a)$] - Valid as Null Moves are allowed.

	DPDA	NPDA
1.	Choices are not allowed	choices are allowed.
2.	Dead configuration is allowed	DC allowed.
3.	ε moves allowed with restriction.	ε- moves allowed freely.

Γ^* - infinite set of finite sized strings.

Page No.
Date : / /

$\delta_{Q \times (\Sigma \cup \epsilon) \times \Gamma^*} \rightarrow \mathcal{P}_{Q \times \Gamma^*}$ (ϵ- move spontaneous jump without reading I/P symbol)

$\therefore \delta_{Q \times \Sigma \times \Gamma^*} \rightarrow \mathcal{P}_{Q \times \Gamma^*}$ (as you can put any no of symbol)

NPDA - $\delta(q_0, a, b) = \{(q_1, ab), (q_2, \epsilon)\}$

NPDA can do non deteriministic step by making two copies of the machine.

↓ push a ↓ pop b

$\therefore \delta_{Q \times (\Sigma \cup \epsilon) \times \Gamma} \rightarrow 2^{Q \times \Gamma^*}$

Γ^* will allow infinite subsets as

$\{[a^n b^n] \cup [a^n b^{2n}] \cup [a^n b^{3n}] \cup \ldots\}$

$\therefore$ it is written as

$\delta_{Q \times (\Sigma \cup \epsilon) \times \Gamma} \rightarrow$ finite subsets of $Q \times \Gamma^*$

or

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

Infinite sets are not valid as PDA can only make finite no of copies.

[$\delta(q_0, a^*, b) = (q_1, a^*)$] - invalid because it must have some limit

[$\delta(q_0, \epsilon, b) = (q_1, a)$] - Valid as Null Moves are allowed.

	DPDA	NPDA
1.	Choices are not allowed	choices are allowed.
2.	Dead configuration is allowed	DC allowed
3.	ϵ moves allowed with restriction.	ϵ- moves allowed freely.

EXCELLENT

In NFA, ε moves are not allowed as it always create choice but due to stack it is possible to avoid choice by using ε move in DPDA

Page No.
Date :

- DPDA, ε moves are allowed in such a way it does not create choices.

$\therefore \quad \delta(q, \varepsilon, a) \neq \phi$

then $\quad \delta(q, c, a) = \phi \quad \forall c \in \Sigma$

i.e if null move is specified for a state with specific symbol then you cannot do with input symbol otherwise choices will be created.

Ex - $\begin{cases} \delta(q, \varepsilon, a) = (q', a) \\ \delta(q, a, a) = (q'', aa) \end{cases}$

↘ choice created Hence it is not a DPDA because DPDA cannot create two copies together.

- the combination of $Q \times \Sigma \times \Gamma$ is not specified in δ, they are considered as ϕ which lead to Dead configuration.

Command -

- PUSH $\quad \delta(q_0, a, b) = (q_1, ab)$
- POP $\quad \delta(q_0, a, b) = (q_1, \varepsilon)$
- DoNothing (SKIP) $\quad \delta(q_0, a, b) = \{\delta(q_0, b)$ or $(q_1, b)\}$
- REPLACE → $\delta(q_0$

$[\delta(q_0, a, b) = (q_1, c)]$

any no of symbol can be pushed and only one symbol can be poped at a time

in skip, change of state can occur but no operation with stack

- FA can be simulated by PDA as

$[\delta(q_0, a) = q_1$
$\therefore \delta(q_0, a, Z) = (q_1, Z)]$

↓ Start stack symbol

- SKIP is used to simulate FA by PDA

EXCELLENT

• Conversion of CFG to PDA using GNF uses Replace command.

Page No.
Date : /

• $\{c^* a^n b^n \mid n \geq 1\}$
SKIP c like

$\delta(q_0, c, Z) = (q_0, Z)$
$\delta(q_0, a, Z) = (q_0, aZ)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \epsilon)$
$\delta(q_1, b, a) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, Z) = (q_f, Z)$

Every PDA can be programmed in three states only (Maximum states)

Ques - $(a^n b^n \mid n \geq 0)$

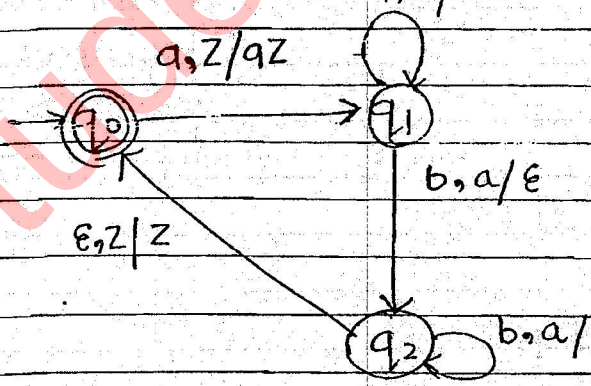
~~$\delta(q_0, a, Z) = ($~~

String rejection in DPDA is done in two ways -
• Non final state
• Dead configuration

• if a language accept ϵ, make start state as final state, if possible

push 1st b ← $\delta(q_0, a, Z) = (q_1, aZ)$ $F = \{q_0\}$
push all a ← $\delta(q_1, a, a) = (q_1, aa)$ $q_0 = q_0$
$\delta(q_1, b, a) = (q_2, \epsilon)$ → pop first b
$\delta(q_2, b, a) = (q_2, \epsilon)$ → pop all b
$\delta(q_2, \epsilon, Z) = (q_0, Z)$ → final state

3 ways to program
• state table
• Diagram
• Transition function
I/P stack Push to stack
q, Z / qZ or q, Z, qZ
If more than 1 comment
q, Z, qZ ; b, Z, bZ



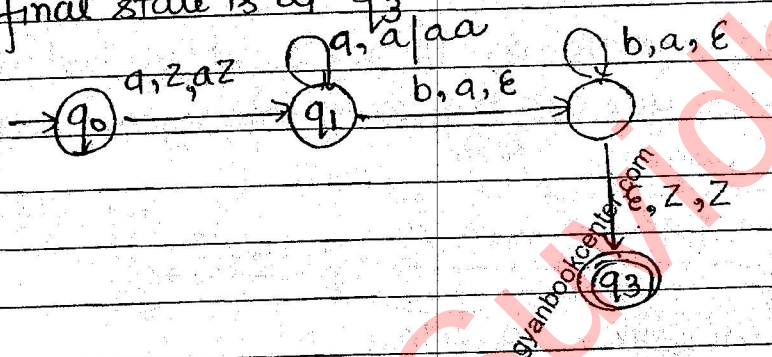
In final state Method, need not to empty the stack but in empty stack method, stack must be emptied at the end of fn i.e for acceptance of string stack must be emptied.

EXCELLENT

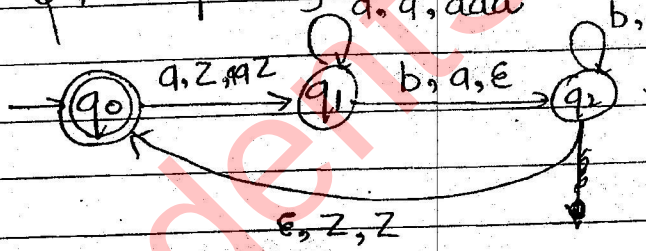
I $\{a^m b^n \mid m, n \geq 0\}$

In final state - a will be there in stack
In empty state - you have to remove a from stack for its acceptance.

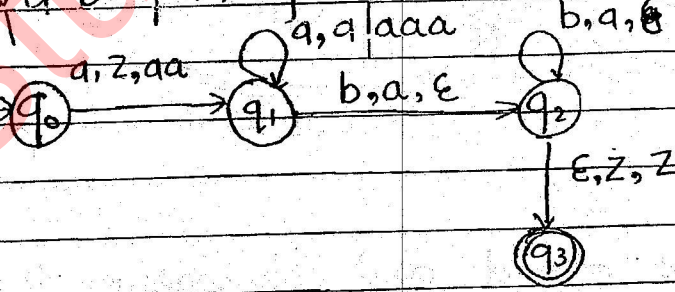
Ques - $\{a^m b^n \mid n \geq 1\}$
then acceptance will not be q_0
final state is at q_3



Ques - $\{a^n b^{2n} \mid n \geq 0\}$

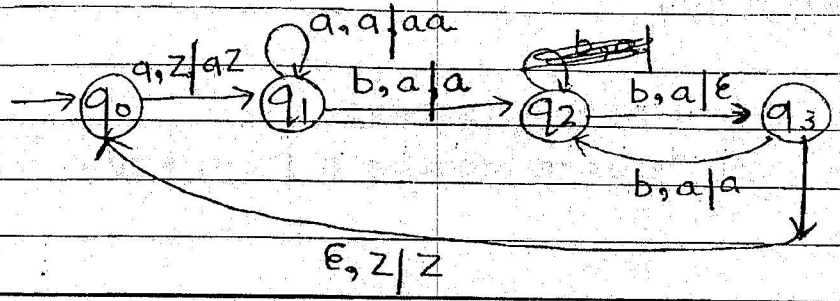


Ques - $\{a^n b^{2n} \mid n \geq 1\}$



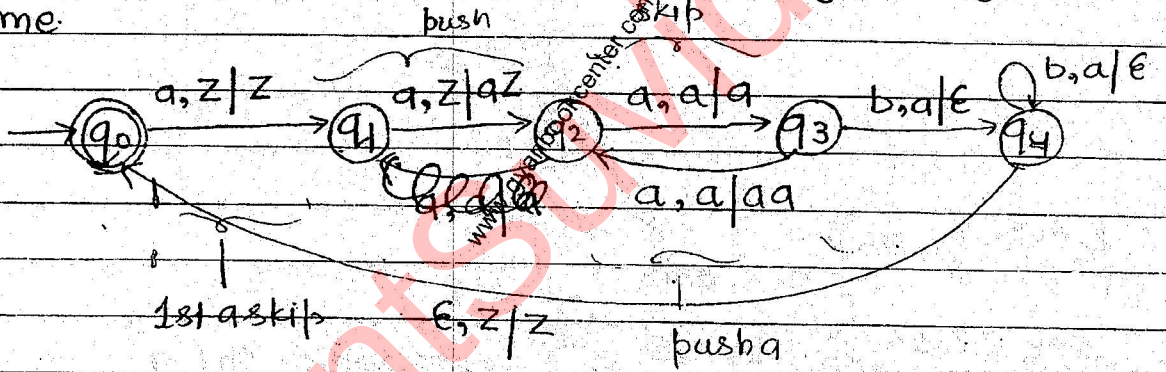
Ques - $\{a^n b^{2n} \mid n \geq 1\}$
↓ for popping b pop for 1 b & skip for another

$a^n b^{2n}$ - push 1 a for 1 a & pop 2 b's for 1 a

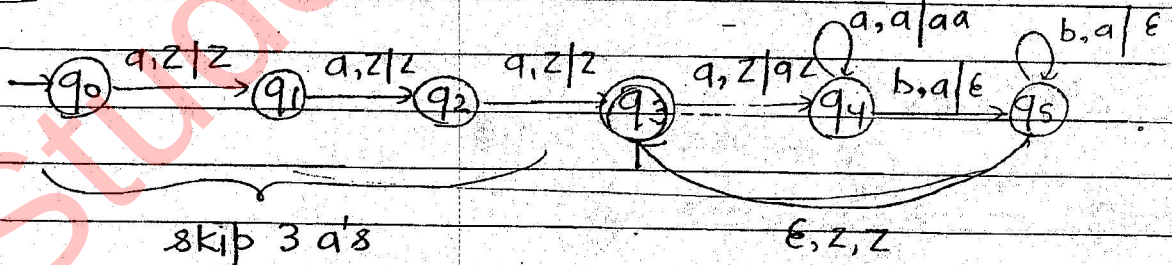


to see for machine
- check for the condition of both a & b.

Ques - $\{a^{2n} b^n\}$ only 1 program as it has condn of poping because you can pop only one symbol at a time



Ques - $a^{2n+3} b^n = aaa a^{2n} b^n$

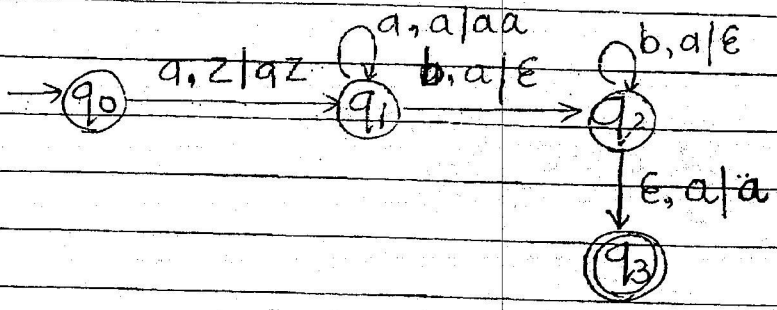


$a^{2n} b^{n+3}$ → skip 3 b's in mid and also consider the case for b^3 when n=0

EXCELLENT

Page No.
Date

Ques - $\{a^m b^n \mid m > n\}$, $m, n \geq 0$



Ques - $\{a^m b^n \mid m < n\}$

last comment -

$\delta(q_2, b, z) = (q_3, z)$ it will go to final state if aabbbb

at 3rd b it go to q_f at another b it will go to dead so you have to write

$\delta(q_f, b, z) = (q_f, z)$

$\delta(q_f, \epsilon, z) = (q_f, z)$

Ques - $\{a^m b^n \mid m \geq n\}$

$\delta(q_2, \epsilon, a) = (q_f, z)$ → for $a > b$

$\delta(q_2, \epsilon, z) = (q_f, z)$ → $a = b$

$\{a^m b^n \mid m \leq n\}$

$\delta(q_2, b, z) = (q_f, z)$

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

$\{a^m b^n \mid m \neq n\}$

(for $m < n$ & $m > n$)

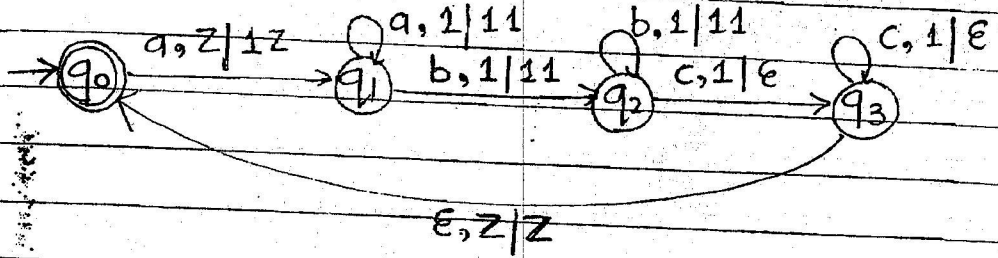
Combine them.

for a you can push 1 also

Page No.
Date : /

Ques - $\{a^m b^n c^{m+n} \mid m, n \geq 0\}$

for a push = 1
b push 1
c pop 1

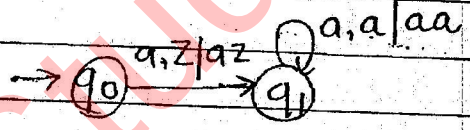


Ques - $\{a^{m+n} b^m c^n \mid m, n \geq 1\}$

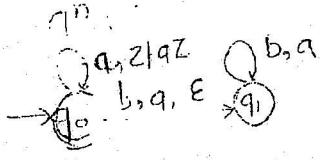
a - push 1
b, c - pop 1

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

~~p = m+n~~ a → push 1
b - pop until you reach end of stack.
if reach end just push 1
c → pop 1



EXCELLENT



Page No.
Date

$\{a^n b^n\} \cup \{a^n b^{2n}\}$

$\delta(q_0, \epsilon, z) = \{(q_1, z)\ (q_2, z)\}$

↓ Now q_1 will care for $a^n b^n$

↓ It can do for $a^n b^{2n}$

if it is having any split, it might be an union.

- this thing is allowed only in NPDA

$\{a^n b^m \mid m \le n \le 2m\}$

$S \to aSb \mid aSbb$

then

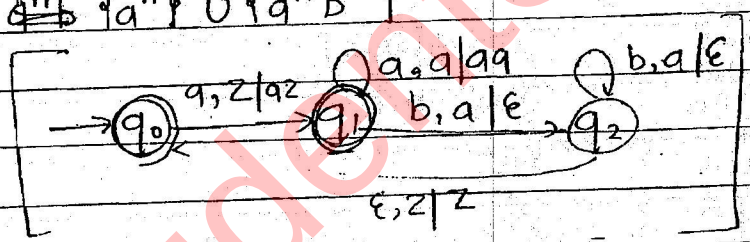
$\delta(q_0, a, z) = \{(q_i, az), (q_i, aaz)\}$

it can have any one required.

for $n \le m \le 2n$

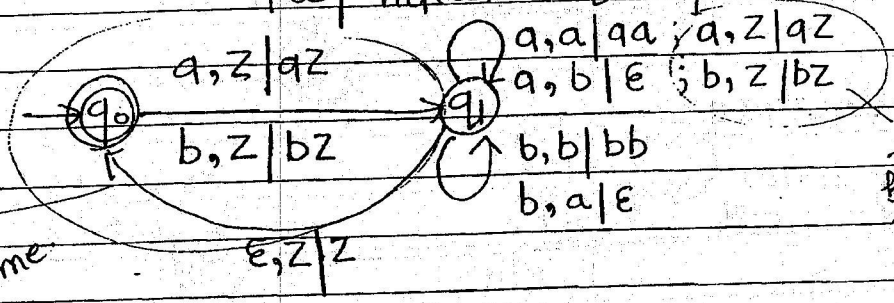
$\delta(q_0, a, z) = \{(q_1, bz), (q_1, bbz)\}$

$\{a^n\} \cup \{a^n b^n\}$



II family $- n_a = n_b$

$\{\omega \mid n_a(\omega) = n_b(\omega)\}$



take care for 1st a or b as any thing can come

because stack can become in mid also.

EXCELLENT

III family - Palindrome

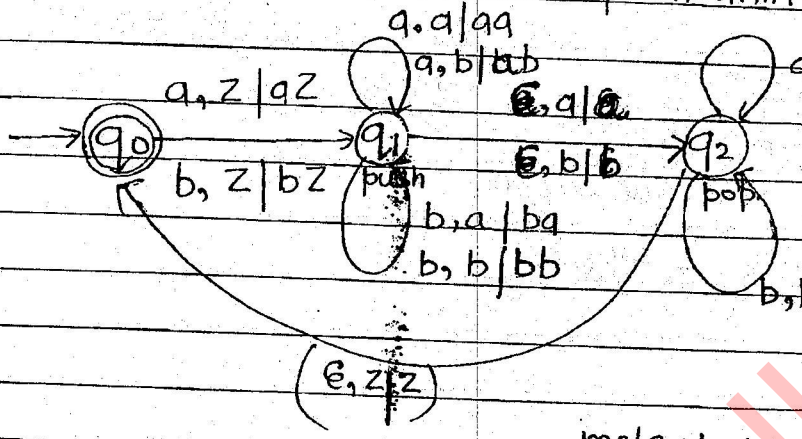
→ $\{ww^R \mid w \in (a,b)^*\}$

• NPDA job as push to pop is not clear.

aabbaa

push until you reach mid

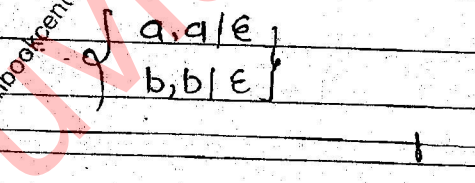
ε - even palindrome



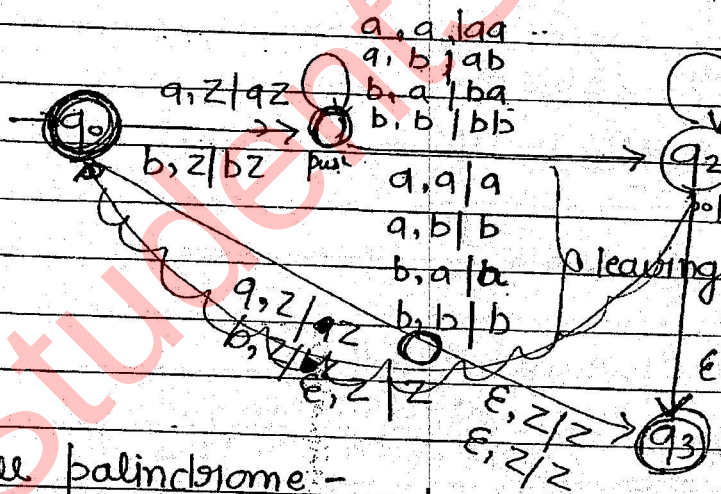
one copy will push & one will always pop

instead of writing null move, i can also write -

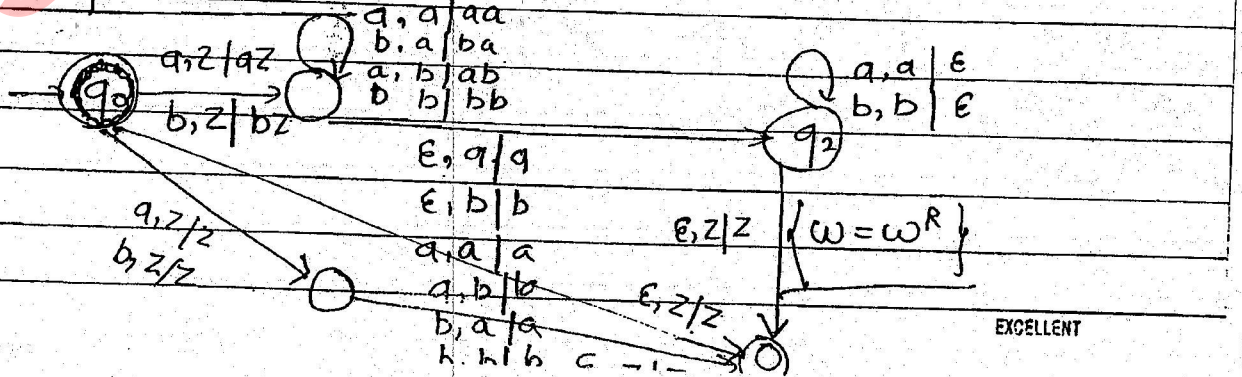
$\{a,a|\varepsilon$
$b,b|\varepsilon\}$



$\{w x w^R \mid w \in (a,b)^*\}$



for all palindrome -

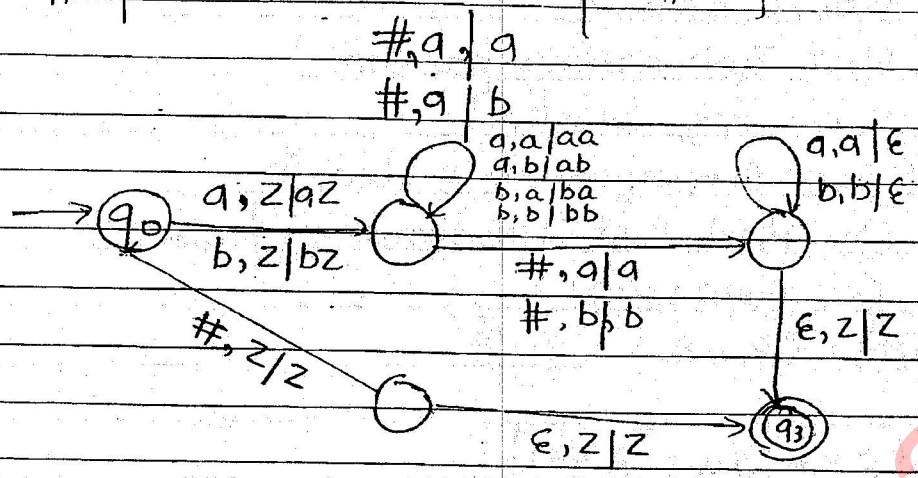


EXCELLENT

final state in wxw^R is maintained by making a copy for the single length strings

Page No.
Date

for # palindrome $\{w\#w^R\}$



GODEL, TURING, POST

Turing Machine, REC, RE, Not RE

Turing Machine - 1. Defmition
2. Machine to Language
3. L - M ? $\{a^n b^n c^n \mid n \geq 0\}$
4. Turing Machine as a transducer
$f(x,y) = x+y$
5. Church turning thesis
Imp 6. Variation of Turing Machine

"No Logical Machine can have more power than turing machine."

RE & REC language - • Definition of REC & RE.
• facts related REC & RE.
• Enumeration Procedure (EP) property
• Membership Algorithm (MA) property
• Lexigraphical ordering property
• Closure properties
• L, $\bar{L}$ theorem
• Examples of REC, RE, Not RE

RE but not REC - L_d, c.f.g. (self embedding)