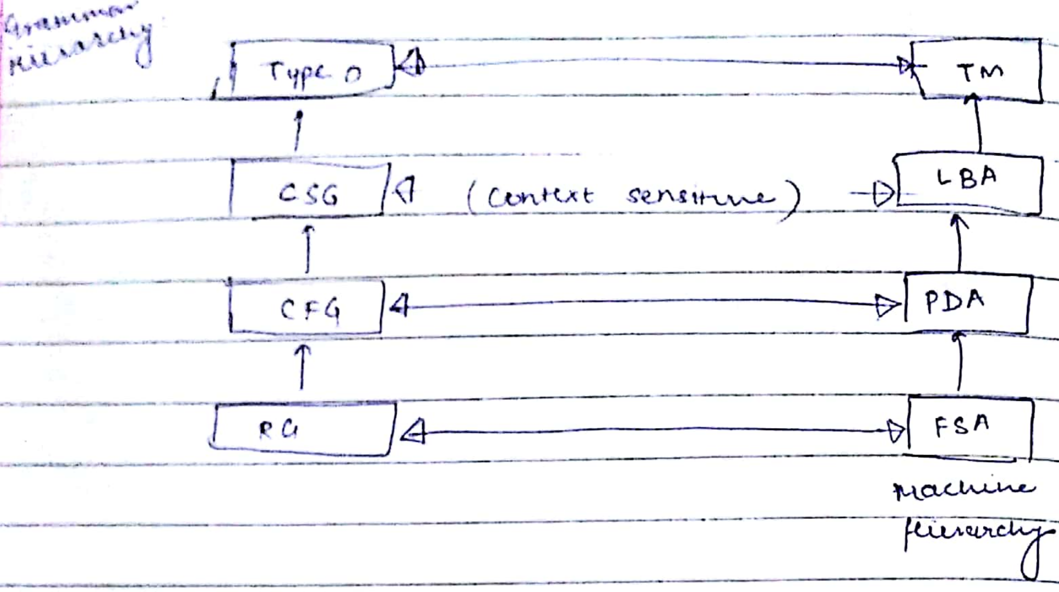
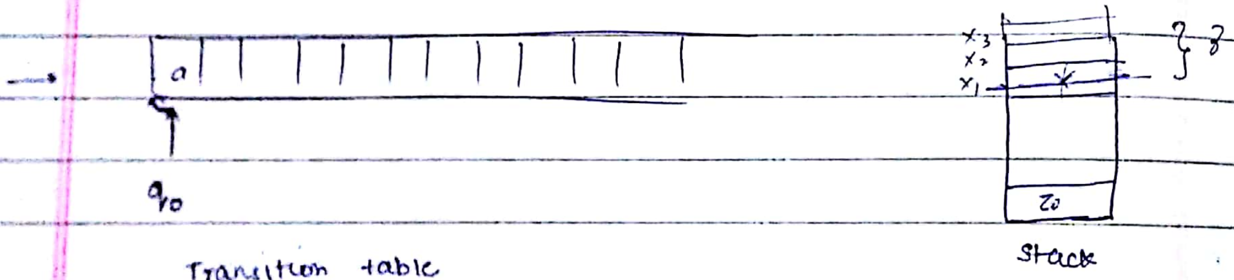
DATE: 5/3/18
PAGE NO.:

PUSHDOWN AUTOMATA



Chomskian Heirarchy of languages



not only sees 'a' (i/p symbol) but also top of stack (symbol at top of stack)

Transition :

FSA : $\delta(q, a) = q'$

PDA : $\delta(q, a, x) = (q', \gamma)$

x can be replaced by sequence of symbols called γ $\gamma = x_1 x_2 x_3 \ldots x_k$

can be empty (ϵ)

{ x is deleted
x is popped up }

Teacher's Signature

PAGE NO.:

* Automata : Finite representation of an infinite no. of things.

$a^n b^n$: not accepted by FA :
↳ don't have memory. don't know count of 'a' occured
↳ can't move to b till a is completed.

↓

Possible in case of Pushdown Automata.
→ can store a in stack & whenever get a 'b', pop out 1 'a' out of the stack.

* Everything FA is a PDA (without any use of stack)

$\{a^n b^n ; n \geq 1\}$: accepted by PDA
$\{a^n b^n c^n ; n \geq 1\}$: not " " "

Q Every finite subset of Non-Regular set is regular.
→ Regular
Yes.

Q Let L be a RL. What can you say about L^R ?
Yes

Q. Union of RL is RL
Yes

Q. Infinite Union of Regular set is Regular
No.

12/3/18

PDA

Introdⁿ to Automata Theory — Hopcroft, Ullman, Motwani

DFA = NDFA = ε - NDFA (Can convert from either one to anyone)

Deterministic PDA : DPDA < NPDA (Power)

PDA : (Q, Σ, Γ, δ, q_0, Z_0, F) [Γ: stack; Z_0: starting stack alphabet]

- Q → finite set of all states
- Σ → finite set of i/p symbols / alphabets
- Γ → finite set of stack alphabets
- q_0 → Initial state
- Z_0 → start stack symbol
- F → set of all final states
- δ → $\delta : Q \times \Sigma \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ $\left[2^{Q \times \Gamma^*} \right]$ set of all subsets of $Q \times \Gamma^*$; Γ^* ↓ string of symbols over γ

NPDA : (go to multiple states ⇒ $2^{\sim}$) (subset)

2 ways of accepting automata :

→ Acceptance by final states

→ Acceptance by Empty Store : Whenever stack is empty & you've read ~~all~~ the whole i/p string.

ID of a PDA : (q, w, x) / (q, w, γ)
↓ instantaneous descrpⁿ

q → current state; w → i/p string; x → top of stack

δ : $(q, aw', x) \vdash (q', w', x')$
⇓
$\delta(q, a, x)$ includes (q', α) — includes because it's non-deterministic

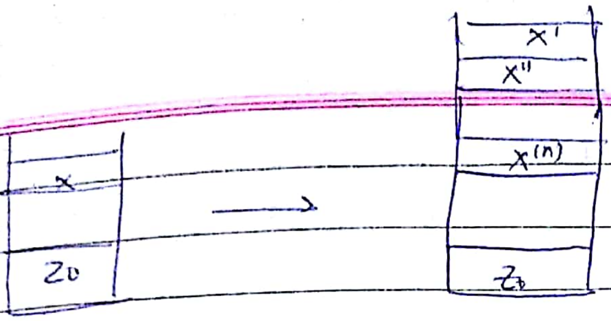
α = string of symbols ⇒ $\alpha = x' x'' x''' \ldots x^{(n)}$ (x' going to be top of stack)

q goes to q'

aw' goes to w' ⇒ no need to show

x

Teacher's Signature



$\alpha \in \Gamma^*$

if $\alpha = \Lambda \Rightarrow$ just popping out X

$\alpha = X_1 \Rightarrow$ replacing X by X_1

① Acceptance by final state :

(q_0, w, Z_0) — starting ID

$w = a^n b^n$

$L = \{ w \mid w \in \Sigma^* \text{ and } (q_0, w, Z_0) \vdash^* (q_f, \epsilon, X)$ where $q_f \in F$, $X \in \Gamma \}$

can ~~be~~ be any symbol in stack

② Acceptance by Empty Store :

$L = \{ w \mid w \in \Sigma^* \text{ and } (q_0, w, Z_0) \vdash^* (q, \epsilon, \epsilon) \}$

0110

→ $L = \{ ww^R \mid w \in \{0,1\}^* \}$

$\delta(q_0, 0, Z_0)$ includes $(q_0, 0Z_0)$ ⇒ Pushing 0 in Z_0

need to store 0 because have to check pattern

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

Teacher's Signature

at any point of time, it may say that w is finished. Now check for $w^R \Rightarrow$ go to new state q_1 & no change in the stack

to accept null string

$\delta(q_0, \epsilon, Z_0)$ includes (q_1, Z_0) → Non-deterministic move

final state

$\delta(q_0, 0, 0)$ " $(q_0, 00)$ (→ tos)

$\delta(q_0, 0, 1)$ " $(q_0, 01)$ (→ tos)

$\delta(q_0, 1, 0)$ " $(q_0, 10)$ (→ tos)

$\delta(q_0, 1, 1)$ " $(q_0, 11)$ (→ tos)

$\delta(q_0, \epsilon, 0|1)$ " $(q_1, 0|1)$

at any time it may say w is finished so reach to q_1

Reverse: (w^R)

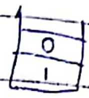
$\delta(q_1, 0, 0)$ " (q_1, ϵ) → pushing ϵ in stack

$\delta(q_1, 1, 1)$ " (q_1, ϵ)

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

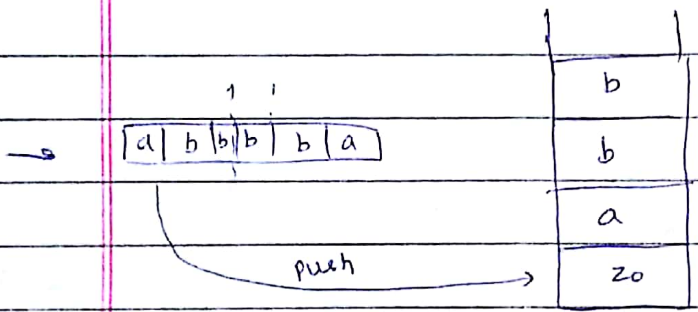
↓ final state accept it

1001



→ got w in stack

If have 0, tos will also be 0



after this, pop &

q_0 : for reading w

when checking for w^R : go to q_1

From q_1 : check if string & tos are same. If so, pop the tos.

→ If a string is accepted by PDA, there will be atleast 1 path to reach final state.

→ At any moment, it has equal probability to either remain on same state or move to next state

$Q = \{q_1, q_0, q_2\}$ $\Sigma = \{0,1\}$

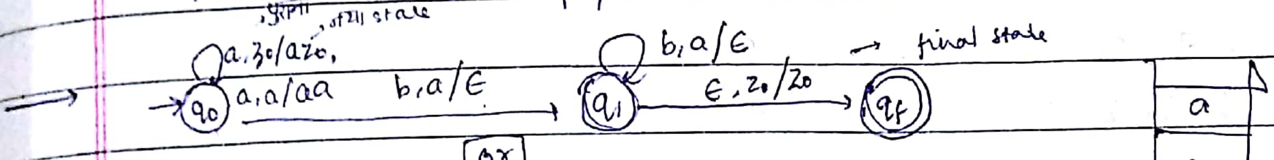
$F = \{q_2\}$ $\Gamma = \{Z_0, 0, 1\}$

Eg. $L = \{a^n b^n \mid n \geq 1\}$ → here, we can find when a will ~~start~~ ~~end~~ & b will start (unlike earlier eg).

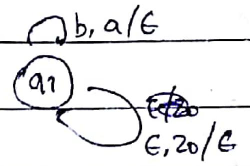
→ Order dhyaan rakhna h ⇒ b aane pe state change

when reads a → push

when reads b → pop 1 a with 1 b



Or



⇒ Acceptance by Empty stack

$\delta(q_0, a, z_0) \Rightarrow \delta(q_0, az_0)$

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

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

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

~~$\delta(q_1, \epsilon, z_0)$~~

$\delta(q_1, \epsilon, z_0) \to \delta(q_f, \epsilon)$ $\delta(q_1, \epsilon, z_0) \to (q_f, z_0)$

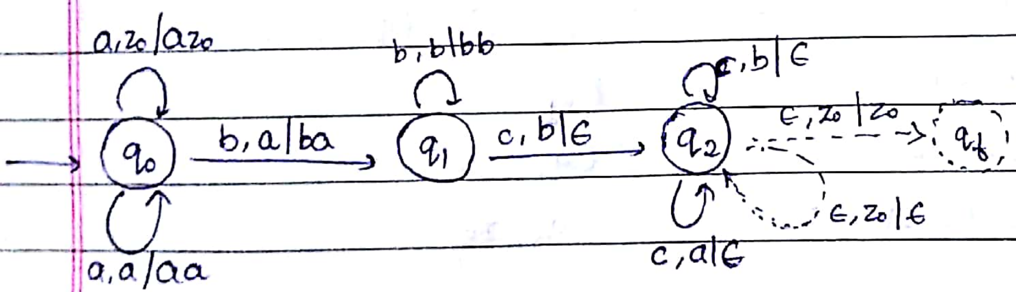
1.) $L = \{a^i b^j c^k \mid i, j, k \geq 1, i+j = k\}$

2.) $L = \{a^i b^j c^k \mid i, j, k \geq 1, i+k = j\}$

* 3.) $L = \{a^m b^n \mid m, n \geq 1, m < n \leq 2m\}$

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

1.) $L = \{a^i b^j c^k \mid i, j, k \geq 1, i+j = k\}$



$\delta(q_0, a, z_0) \to \delta(q_0, az_0)$ $\delta(q_2, c, b) \to \delta(q_2, \epsilon)$

$\delta(q_0, a, a) \to (q_0, aa)$ $\delta(q_2, c, a) \to (q_2, \epsilon)$

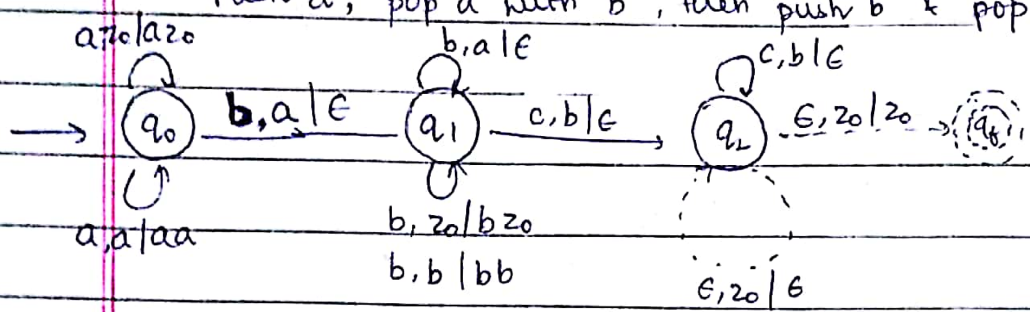
$\delta(q_0, b, a) \to (q_1, ba)$

$\delta(q_1, b, b) \to (q_1, bb)$ $\delta(q_2, \epsilon, z_0) \to (q_2, \epsilon, \epsilon)$ $\delta(q_2, \epsilon, z_0) \to (q_f, z_0)$

$\delta(q_1, c, b) \to (q_2, \epsilon)$

2) $L = \{a^i b^j c^k \mid i,j,k \geq 1,\ i+k=j\}$

Push a, pop a with b, then push b & pop b with c



$\delta(q_0, a, z_0)$ includes (q_0, az_0)

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

$\delta(q_0, b, a)$ " (q_1, ϵ)

$\delta(q_1, b, a)$ " (q_1, ϵ)

$\delta(q_1, b, z_0)$ " (q_1, bz_0)

$\delta(q_1, b, b)$ " (q_1, bb)

$\delta(q_1, c, b)$ " (q_2, ϵ)

$\delta(q_2, c, b)$ " (q_2, ϵ)

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

3.) Push a, Read b ⇒ pop a or read 2 b & pop 1 a.

$\delta(q_0, a, z_0)$ includes (q_0, az_0)

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

1st choice $\delta(q_0, b, a)$ " $\{(q_1, \epsilon), (q_2, a)\}$

pop a ↳ won't pop a

$\delta(q_1, b, a)$ " $\{(q_1, \epsilon), (q_2, a)\}$

$\delta(q_2, b, a)$ " (q_1, ϵ)

$\delta(q_1, \epsilon, z_0)$ " (q_f, z_0)

$\delta(q_1, \epsilon, z_0)$ " (q_f, z_0)

$q_1 \rightarrow$ for every b : popping a

$q_2 \rightarrow$ skipping for 1 b

Scanned by CamScanner

1) Acceptance by Final states $\equiv$ 2) Acceptance by Empty store

→ showing

$q_f \rightarrow q_{em}$

a) $1 \Rightarrow 2$

let PDA $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

with acceptance by final state

(want to construct ~~&~~ acceptance by empty store for L)

let $L = L(A)$

We need to construct A' (new PDA) s.t.

$L = L(A')$ [acceptance by empty store]

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

→ Once we go to f.s., we can empty the stack for

$F = \{q_1, q_2, \ldots q_n\}$

δ' should include all transitions of δ in add^n to that we've the following]

$\forall q_i \in F$ $\delta'(q_i, \epsilon, X)$ includes (q_{empty}, X) : go to state q_{empty} & empty all the stack

$\delta'(q_{empty}, \epsilon, X)$ includes (q_{empty}, ϵ) → stack empty ⇓ Acceptance by Empty store

By construction, we see that $Q' = Q \cup \{q_{empty}\}$

$L(A') = L(A)$

b.) $2 \Rightarrow 1$

We have : $B = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \phi)$ ← final state tak pahuchega hi nhi

B is accepting by empty store.

let $L = L(B)$

We need to construct B' (new PDA) s.t.

$L = L(B')$

$\Gamma' = \Gamma \cup \{z_0'\}$ q_0

$Q' = Q \cup \{q_0', q_{final}\}$ q_0'

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

B' accepts by final state

→ when stack becomes empty, make it reach to F.S.

when start processing for B', put z_0' in stack
z_0' is initial symbol

z_0'

$\delta'(q_0', \epsilon, z_0')$ includes $\delta(q_0, Z_0 Z_0')$

from q_0, we can apply all transitions of B.

$\delta(q, \epsilon, z_0')$ includes (q_{final}, z_0') $\underline{F = q_{final}}$

$q \in Q'$ ⇓

B ka stack is empty → go to final state

* z_0' included to check when B ka stack becomes empty.

* Moreover δ' includes all transitions of δ.

→ By construction,

→ $L(B) = L(B')$

Eg. $L = \{ w \in \{a,b\}^* \mid n_a(w) = n_b(w) \}$

$L = \{ w \in \{a,b\}^* \mid n_a(w) = n_b(w)$ → Dyck set (Dyck language)
for any prefix w' of w
$n_a(w') \geq n_b(w') \}$

→ ( ) ( ) ( ) : well formed parenthesis
a b a b a b

(( ))) : not " "
a a b b b

))) (((: " " "
b b b a a a

no. of open brackets $\geq$ no. of closed brackets
~~in prefix of~~
string

Teacher's Signature

PDA ban skta h → CFL h
Otherwise, nhi h

→ whenever see a → push in stack
b → pop out of stack

CFG

|

CFL —subset→ DCFL (jiska DPDA ban jaaye)

↓

PDA

for some CFL, we can draw DPDA

for all CFL, we can draw NPDA

DPDA ⊂ NPDA (* Every NPDA can't be converted into DPDA)

$\boxed{PDA = FA + Stack}$ → ~~used because~~ it is 0 address (add. don't 'ne to be specified for del^n / ins^n, always happens on top of stack)

Transition func^n

1) DPDA : $\delta : Q \times \Sigma \times \Gamma$ (toS) $\rightarrow Q \times \Gamma^*$

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

Basic Op^{ns} on Stack :-

① PUSH :

| a | b |

$(q_i) \xrightarrow{a, Z_0 / aZ_0} (q_j)$ push kiya h

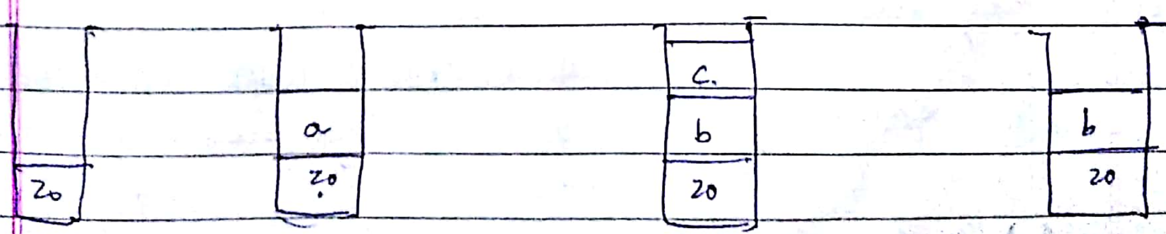
$(Z_0 \rightarrow aZ_0)$

② POP

| a |

$(q_i) \xrightarrow{a, c / \epsilon} (q_j)$

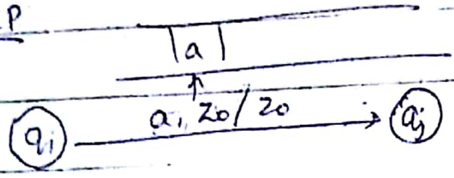
sirf tos dikhega, c pop kiya, naya status q_j pe pata chalega

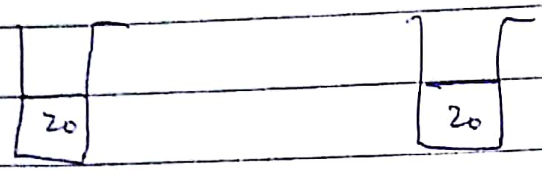


$\delta(q_i, a, Z_0) \rightarrow (q_j, aZ_0)$

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

③ SKIP





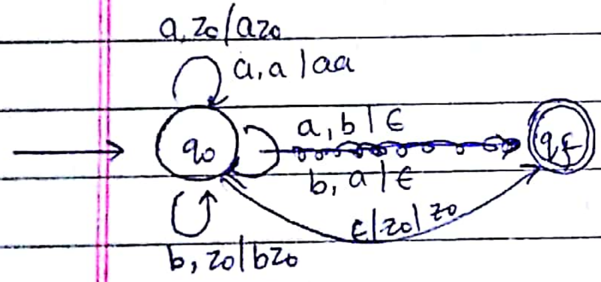
$\delta(q_i, a, Z_0) = (q_j, Z_0)$

Eg. $L = \{ w \in \{a,b\}^* \mid n_a(w) = n_b(w) \}$

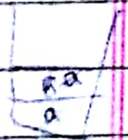
→ i/p 'a' : a will pop out b, else, push a

→ i/p 'b' : b will pop out a, else, push b.

aababb

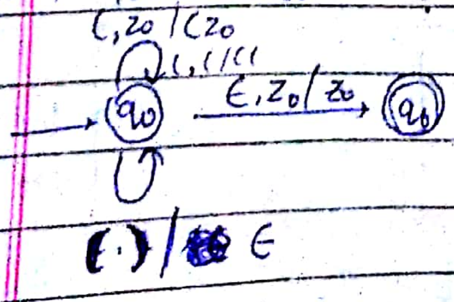


(q_0, a, Z_0)	includes	(q_0, aZ_0)
(q_0, a, a)	"	(q_0, aa)
(q_0, b, Z_0)	"	(q_0, bZ_0)
(q_0, b, b)	"	(q_0, bb)
(q_0, a, b)	"	(q_0, ϵ)
(q_0, b, a)	"	(q_0, ϵ)
(q_0, ϵ, Z_0)	"	(q_f, Z_0)



eg → P $(q_0, \epsilon, Z_0) \longrightarrow (q_0, \epsilon)$

Eg. Paranthesis :



$(q_0, (, Z_0)$	includes	$(q_0, (Z_0)$
$(q_0, (, ()$	includes	$(q_0, (()$
$(q_0, (,))$	"	(q_0, ϵ)
(q_0, ϵ, Z_0)	"	(q_f, Z_0)

Teacher's Signature

in $\{ww^R\}$, $\{a^m b^n : m < n \le 2m\}$: non-deterministic PDA

Can't be accepted by DPDA

DATE: / /
PAGE NO.:

Deterministic PDA (DPDA)

PDA = $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

in which

(1) $\forall q \in Q$, $a \in \{\Sigma \cup \epsilon\}$ $x \in \Gamma$

$\delta(q, a, x)$ contains atmost a unique pair (q, α)

(2) If $\delta(q, \epsilon, x)$ is non-empty, then → can't read ϵ & a at same time

$\delta(q, a, x)$ should be empty & vice versa

→ $\{wcw^R\}$: whenever c is there, we know w is over

⇓ accepted

it'll be a DPDA

* $L(DPDA) \subset L(PDA)$

wcw^R ww^R

$a^n b^m$ $n \le m \le 2n$

Any deterministic PDA can be called as non-deterministic also.

Equivalence of PDA and CFG

1.) CFG ⇒ PDA

2.) PDA ⇒ CFG

1) $\boxed{\text{CFG} \Rightarrow \text{PDA}}$ P : $A \rightarrow \alpha$, $A \in V$, $\alpha \in (V \cup T)^*$

Given a CFG, $G = (V, T, P, S)$

Construct an eq. PDA A s.t. $L(A) = L(G)$

each is in sentential form

$S \xrightarrow{\text{derives}} \alpha A \beta$. $\alpha, \beta \in (V \cup T)^*$

$\rightarrow \alpha \gamma \beta$

$\rightarrow \alpha B \gamma' \beta$ → producing sentencial form by applying rules in grammar

$\rightarrow \alpha \gamma'' \gamma' \beta$

⋮

$\rightarrow$ aabbaa...

A B

→ We'll construct PDA which accepts by empty store

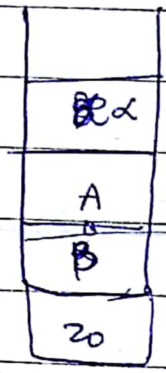
* we use CFG with ~~right~~ left most derivation.

$S \rightarrow \alpha AB\beta$

α

Pop NT & push its rule.

Pop T.

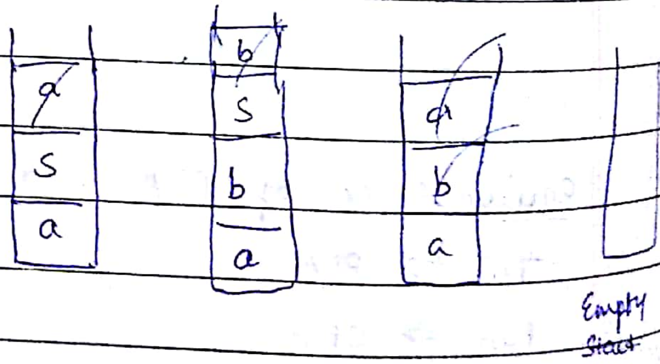


31/3/18

$S \rightarrow aSa$

$S \rightarrow bSb$

$S \rightarrow a$



| a | b | a | b | a |

(make sure i/p symbol & tos are same)

→ It'll accept by PDA.

→ No proofs in exam

F.S. not needed

$P = (\{q\}, T, V \cup T, \delta, q, S, \phi)$

δ transition :-

1) $\forall$ non-terminals (variables) $A \in V$

$\delta(q, \epsilon, A) = \{(q, \beta) \mid A \to \beta$ is a prod^n rule in $P\}$

$\Downarrow$

ϵ move

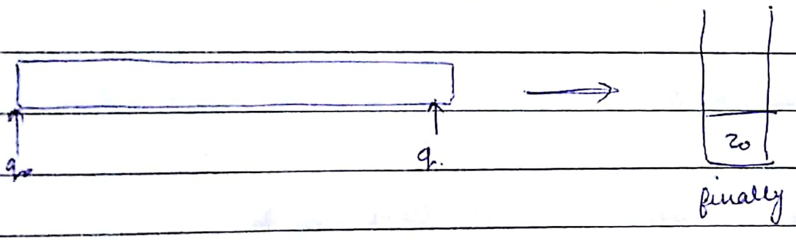
2) $\forall$ terminals $a \in T$

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

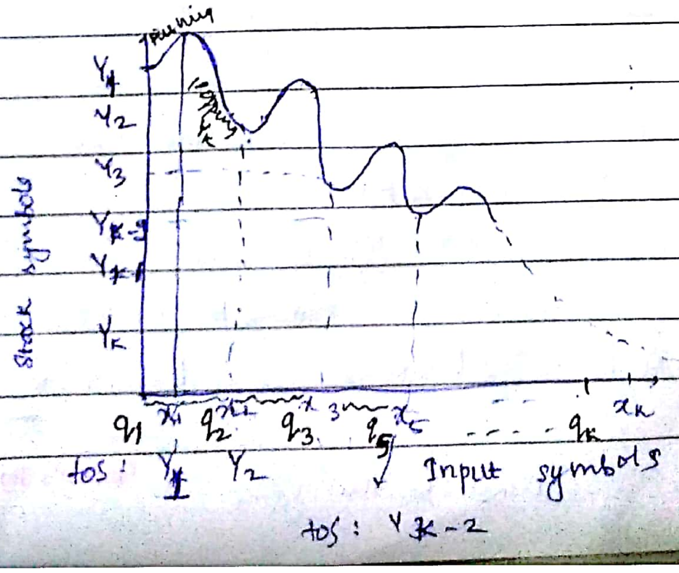
No. of rules $\equiv V + T$

2) $\boxed{\text{PDA} \Rightarrow \text{CFG}}$ (PDA : acceptance by empty store)

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



Graph :



: pushing & popping out of stack

atlast, pop out all the elements in stack

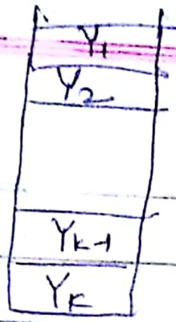
Teacher's Signature

b/w $q_1 - q_2$ read x_1

$q_2 - q_3$: read x_2

⋮

$q_k - q_{k+1}$: read x_k



→ If I want to pop Y_k ⇒ should we popped out $Y_1, Y_2, \ldots Y_{k-1}$ symbols already

From q_1 to q_2 : we are popping Y_1

" state : $[q_1 Y_1 q_2]$

likewise from q_2 to q_3 : $[q_2 Y_2 q_3]$ ⇒ popping each symbol while reading some input symbol

→ $[pXq]$: p is a state

q is a state

$[pXq]$ ~~x~~ is a NT to be popped out.

$[q_0 Z_0 q]$: q_0 to q. In meanwhile, pop Z_0 : Empty stack

⟹ $G = (V, \Sigma, R, S)$

V consists of

(1) Special symbol S → Start symbol

(2) All symbols of the form $[pXq]$

where $p, q \in Q$ and $X \in T$

Production rules of P

(1) $\forall$ states $p \in Q$ $S \to [q_0 Z_0 p]$ → can be any state

(2) let $\delta(q, a, X)$ contains $(r, Y_1 Y_2 Y_3 \ldots Y_k)$

pushing k symbols

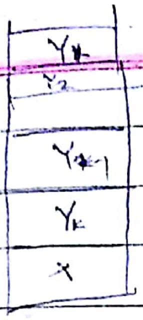
$$[qXr_k] \to a\, [rY_1 r_1]\, [r_1 Y_2 r_2] \ldots [r_{k-1} Y_k r_k]$$

need not be the same

Teacher's Signature

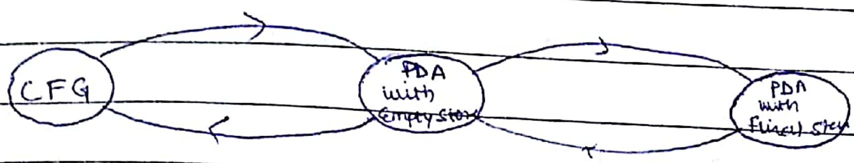
→ Book → Read Proof

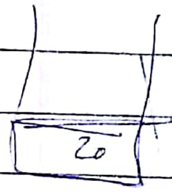
DATE: / /
PAGE NO.:



⇒ Pop out all $Y_1, Y_2, Y_3, \ldots Y_k$

23-03-18





Eg. PDA : $\underline{\underline{i^n e^{n+1}}}$ → This is what we need

Eg. ~~If-then-else~~

$e, Z|\epsilon$ i → if   pushing Z when if comes

$i, z|zz$ e → else

~~state~~ : By empty stack

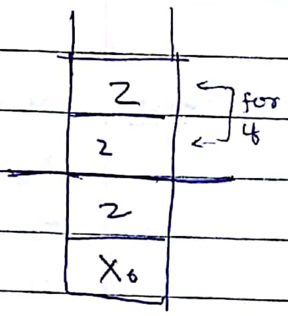
start → q

~~———~~

$e, Z|\epsilon$

$i, Z|ZZ$

start → p —ϵ, $X_0 | ZX_0$→ q —$\epsilon, X_0|\epsilon$→ q_f ✗



⇒ No. of e = No. of i + 1 ⇒ violates rule

PDA ⇒ CFG :1   δ for PDA :

δ(q, e, Z) has (q, ε)

δ(q, i, Z) has (q, ZZ)

n rules for all $q \in Q$ ← (1) $S \rightarrow [qZq]$ ↑A

~~(2)~~ ~~tq~~ ← 2nd Prodn for this

n^2 rules (2) $[qZq] \rightarrow i\,[qZq]\,[qZq]$ { have a more Z to be popped out }

(3) $[qZq] \rightarrow e$

already popping it out , don't have to capture anything else

→ $S \rightarrow A$ $[q_1 Z q_2] \rightarrow i\,[q_1 Z q_3]\,[q_3 Z q_4]$

$A \rightarrow iAA$

$A \rightarrow e$ ~~may vary from 1 to n~~

⇒ n^2 rules

Teacher's Signature

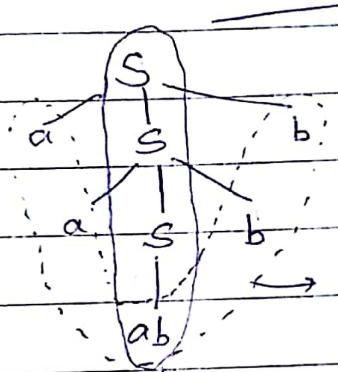
Chomsky : $A \to BC$: 1 NT by only 1
$A \to a$ 1 " " "

DATE: / /
PAGE NO.:

CFG

$S \to aSb$ → $a^n b^n$
$S \to ab$

→ Parse Tree



↔ yield of the tree (a^3b^3)

↳ longest path

Theorem : Suppose we have a parse tree according to Chomsky Normal Form. $G = (V, T, P, S)$ and suppose that the yield of the tree is a terminal string w. If the length of the longest path is n, then

$$|w| \leq 2^{n-1}$$