Fact: It has only two classes. R and $\Sigma^* - R$.

Proof:

① Right congruence:

$x \equiv_R y \Rightarrow xa \equiv_R ya \; (a \in \Sigma)$ To prove.

(obvious)

② refining R

To prove

$x \equiv_R y \Rightarrow (x \in R \Leftrightarrow y \in R)$

Put, $a = \varepsilon$

To prove: $\equiv_R$ is the coarsest relation on R, (regular or not regular)

i.e. we have to prove

Claim: $\equiv$ on Σ^* which is right congruence and refines R then

$\equiv$ refines $\equiv_R$.

$x \equiv y \Rightarrow x \equiv_R y$

Proof:

$x \equiv y \Rightarrow xa \equiv ya \quad (a \in \Sigma)$ [Congruent prop. of $\equiv$]

$\Rightarrow xab \equiv yab \quad (b \in \Sigma)$

$\vdots$

$\Rightarrow \forall z \in \Sigma^* \; xz \equiv yz$

$\Rightarrow \forall z \in \Sigma^* \; (xz \in R \Leftrightarrow yz \in R)$

[refining prop. of $\equiv$)

$\Rightarrow \quad x \equiv_R y$

i.e. $\equiv$ ~~def~~ refines $\equiv_R$

Theorem: (Myhill-Nerode Theorem)

Let $R \subseteq \Sigma^*$.

The following statements are equivalent.

a) R is regular

b) there exists a Myhill-Nerode relation for R

c) the relation $\equiv_R$ is of finite index.

Proof:

($a \Rightarrow b \Rightarrow c \Rightarrow a$) To be proved.

$a \Rightarrow b$ (Already done)

$b \Rightarrow c$ (Since there exists a relation $\equiv$ which is a Myhil-Nerode relation which by defn will be of finite index and since $\equiv$ refines $\equiv_R$ $\Rightarrow$ ~~Obviously~~ $\equiv_R$ is finite.)

$c \Rightarrow a$ $\quad \equiv_R$ is a Myhil-Nerode relation

$\Rightarrow$ DFA can be formed from R

$\Rightarrow$ R is regular.

Lets use the theorem on DFAs

$M = (Q, \Sigma, \delta, s, F)$ ~ DFA

$p, q \in Q$

$p \approx q \overset{def.}{=} \forall z,\ \hat{\delta}(p, z) \in F \Leftrightarrow \hat{\delta}(q, z) \in F$

for R $\approx$ is an identity relation. (i.e. minimised DFA)

R ↓ lang. accepted by DFA.

coarsest relation on R

By defn:

$$x \equiv_R y \Leftrightarrow \forall z \in \Sigma^* (xz \in R \Leftrightarrow yz \in R)$$

$$\Leftrightarrow \forall z \in \Sigma^* \left(\hat{\delta}(s, xz) \in F \Leftrightarrow \hat{\delta}(s, yz) \in F \right)$$

$$\Leftrightarrow \forall z \in \Sigma^* \left(\hat{\delta}(\hat{\delta}(s, x), z) \in F \Leftrightarrow \hat{\delta}(\hat{\delta}(s, y), z) \in F \right)$$

$\cancel{\Leftrightarrow \hat{\delta}(p, z)}$

$$\Leftrightarrow \hat{\delta}(s, x) \approx \hat{\delta}(s, y)$$

identity relation

$$\Leftrightarrow \hat{\delta}(s, x) = \hat{\delta}(s, y)$$

$$\Leftrightarrow x \equiv_M y$$

Context-free languages

<u>generates</u> → <u>accept</u>

Context free grammar (CFG)

push down automata (PDA)

[this is non-deterministic]

any grammar (not necessarily CFG) : $G = (V, T, S, P)$

V : set of finite variables (capitals)

T : set of finite terminals (small)

S : Start symbol.

P : Production rules.

↳ $\alpha \longrightarrow \beta$ (replace α with β)

$\alpha, \beta \in (V, T)^*$

α has at least one variable from V

Type 3 grammar → Regular grammars

Just a fact

$\alpha \rightarrow \beta$

$\alpha \in V$ [i.e. α has only one variable]

right linear

$\beta \in T$ or $\beta \in TV$

left linear

$\beta \in V$ or $\beta \in VT$

[regular grammars generate reg. language]

Context-Free grammar (Type 2 grammar)

$\alpha \in V$, $\beta \in \{VT\}^*$

Eg: $L = \{a^n b^n, \ n \geq 1\}$ | This is a CFG, $\alpha \in V$ β is anything

&

<u>Production rules</u>

$S \longrightarrow aSb$

$S \longrightarrow ab$

Leftmost derivation, rightmost derivation of grammars.

$L(G)$: language accepted by grammar G

$= \{ w \in T^* \mid S \xRightarrow{*} w \}$ (0 or more states)

<u>Definition</u>: A lang. L is context-free language if $\exists$ CFG G s.t.

$L(G) = L$

Sentential Forms:

$$S \xrightarrow{*} \gamma \leftarrow \text{sentential forms}$$

$$\gamma \in \{V + T\}^*$$

Parse Trees:

Yield of a parse tree:

Ambiguous grammars:

A CFG $G = (V, T, S, P)$ is said to be ambiguous iff there exists a string in T^* that has more than one parse tree.

Unambiguous gramma:

Push-Down-Automata : (ε-NFA + Stack)

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

Q : set of finite states

Σ : finite alphabet

Γ : finite stack symbol

q_0 : start state

Z_0 : Beginning ~~state of stack~~ symbol in stack;
a special symbol in stack.

F : final states.

$$\delta(q, a, \overset{\text{topmost stack symbol}}{X}) = (P, \gamma)$$

$q \in Q$

$a \in \Sigma$ or, ε

$X \in \Gamma$

case 1: $\gamma = \varepsilon \Rightarrow$ pop X

case 2: $\gamma = X \Rightarrow$ no change

case 3: $\underbrace{\gamma = YZ}_{\gamma \text{ is a string}} \Rightarrow$ replace X with Z and push Y

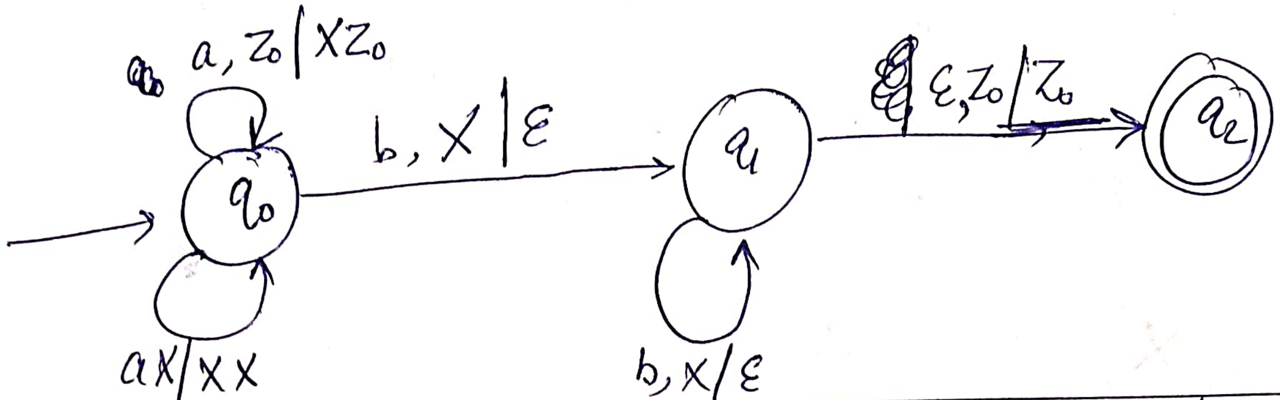
YZ is an example only $(Y, Z \in \Gamma)$

Transition from q, a, X to p, γ

[Note:
Case 5: pop X and push the stack symbols from right to left]

Eg: $\{a^n b^n \mid n \geq 1\}$ (Not regular)

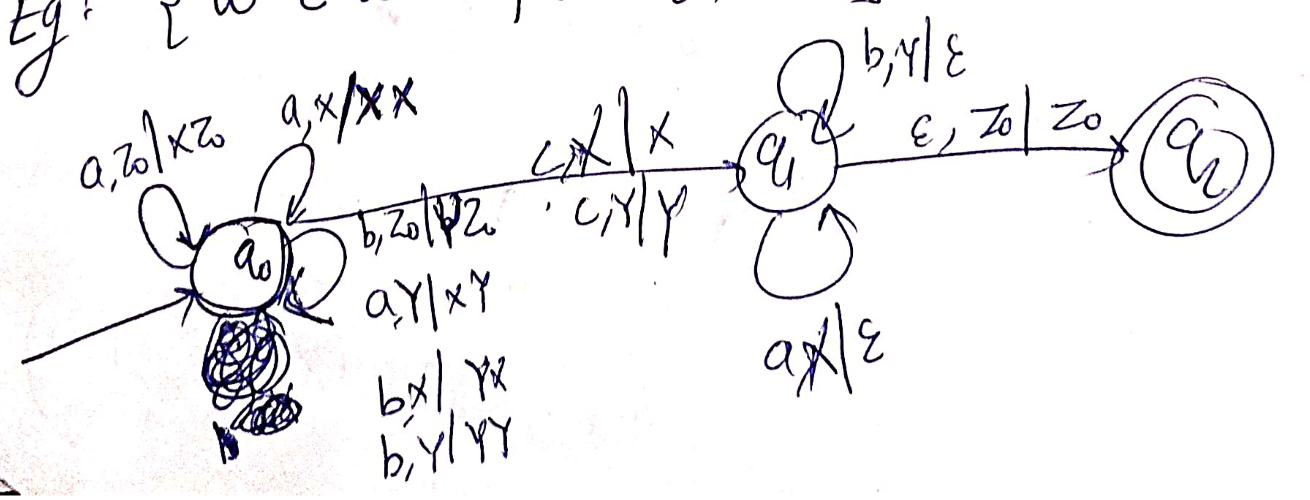
defining a PDA for this language.



X : for a

Note: X keeps the count of number of A's present.

Eg: $\{w c w^r \mid w \in \{a,b\}^{+}\}$ ($+$: empty string not allowed)



Eg

Qw

If $*$ was there inst of $+$, add a transition, $c, Z_0 | Z_0$ from q_0 to q_1

$*$: empty string allowed

Instantaneous Description :

ID : (q, w, γ) $\begin{pmatrix} q \in Q \\ w \in \Sigma^* \\ \gamma \in \Gamma^+ \end{pmatrix}$

(Example : using previous PDA)

$(q_0, abcba, Z_0) \longrightarrow (q_0, bcba, XZ_0) \longrightarrow (q_0, cba, Y\cancel{X}Z_0)$

$\downarrow$

(q, ba, YXZ_0)

Deterministic PDA :

we define a PDA $P = (Q, \Sigma, T, \delta, q_0, Z_0, F)$ to be deterministic (DPDA) if and only if the following conditions are met.

1) $\delta(q, a X)$ has atmost one member for any q in Q, a in Σ or $a = \varepsilon$ & $X \in \Gamma$

2) If $\delta(q, a, X)$ is non-empty for some $a \in \Sigma$, then $\delta(q, \varepsilon, X)$ must be empty.

~~violating this would make PDA non-deterministic since~~

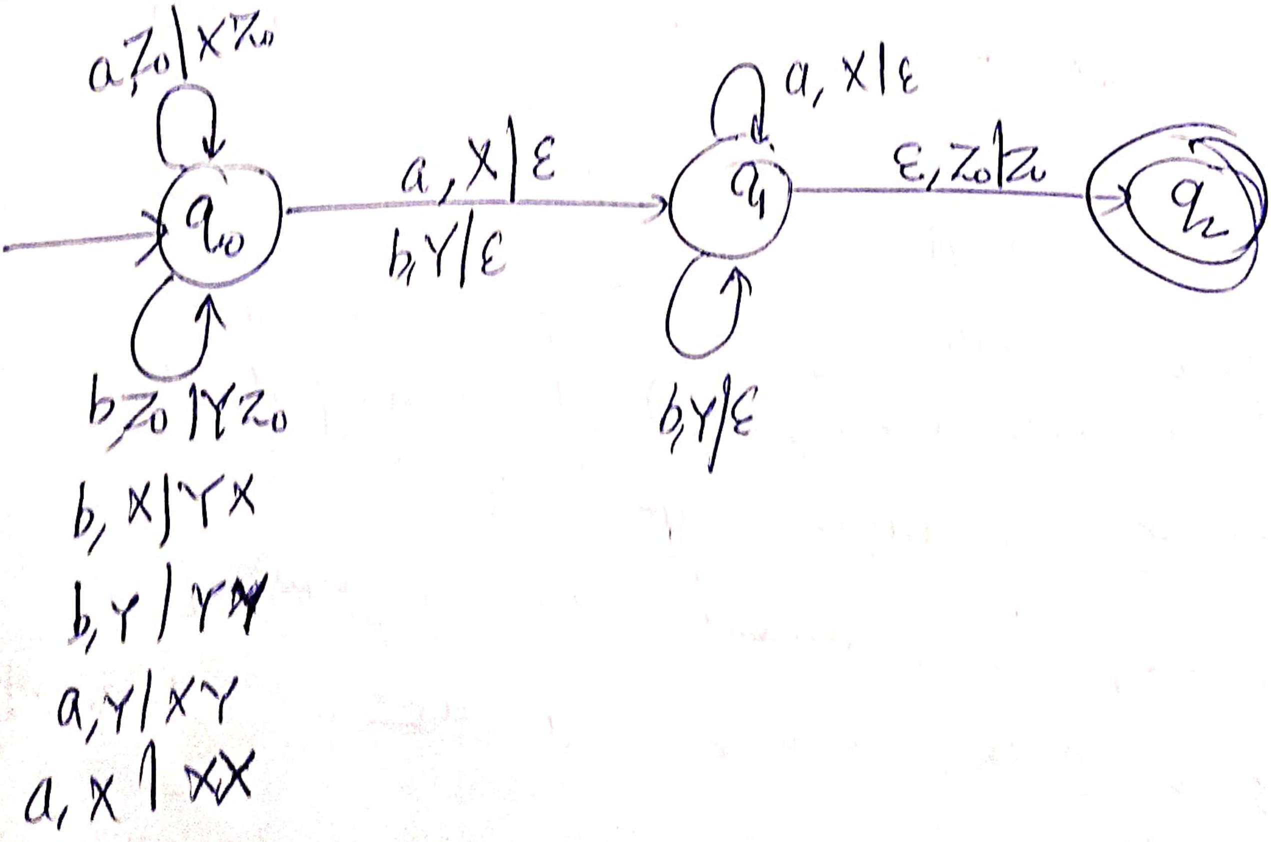
$\{wcw^r \mid w \in \{a,b\}^*\}$ is DCFL

$\{ww^r \mid w \in -\}$ is ND-CFL

ND-CFL

$\{ww^r \mid w \in \{a,b\}^*\}$

At each state, have two transitions, one for assuming this is the centre ~~as~~ and the other



Instantaneous Description (ID) of this PDA.

$(q_0, aabbbbaa, Z_0) \rightarrow (q_0, abbbbaa, XZ_0)$

not came $\swarrow$ $\searrow$ centre has came

$(q_0, bbbbaa, XXZ_0)$ $(q_1, bbbbaa, Z_0)$

$\swarrow \quad \searrow$ $\downarrow$

$\vdots$ stuck

Lang. accepted by PDA

① **Acceptance by final states**

$P = \cancel{(Q, q_0, F)}\ (Q, q_0, F, \Gamma, Z_0, \Sigma)$

$$L(P) = \{ w \in \Sigma^* \mid (q_0, w, Z_0) \xrightarrow{*} (q, \varepsilon, \gamma),\ q \in F,\ \gamma \in \Gamma \}$$

zero or more steps

② **Acceptance by empty stack**

$$N(P) = \{ w \in \Sigma^* \mid (q, w, Z_0) \xrightarrow{*} (q, \epsilon, \varepsilon),\ q \in Q \}$$

For a given P

$N(P) \neq L(P)$ (may not be same)

---

given P, $L(P) = L$

we can construct a diff. PDA P'

s.t. $N(P') = L$

[reverse is also possible]

<u>Theorem</u>: If $L = N(P_N)$ for some PDA.

$P_N = (Q, \Sigma, \Gamma, \delta_N, q_0, Z_0)$

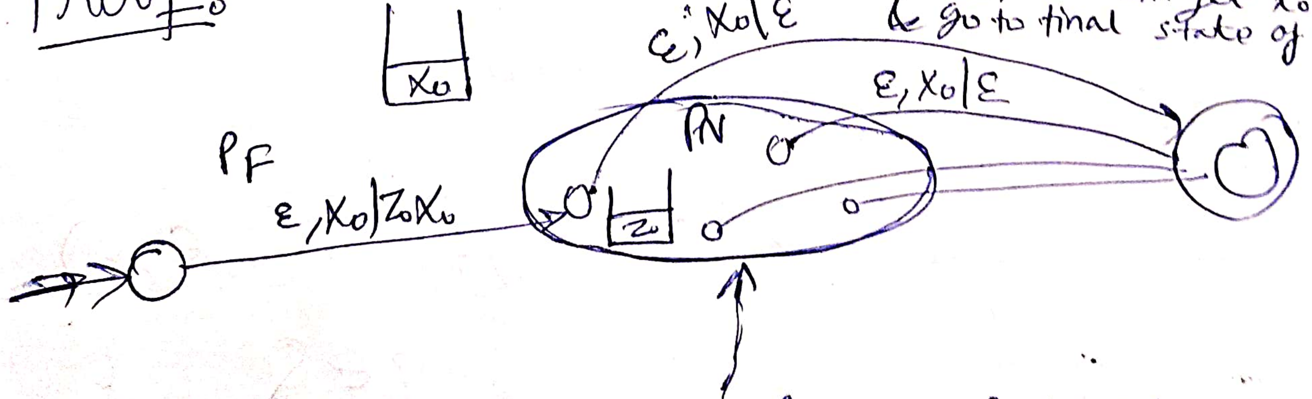
then there is a PDA P_F such that

$L = L(P_F)$

[X_0 : sp. symbol of PF, Z_0 : sp. symbol of P_N]

<u>Proof</u>:

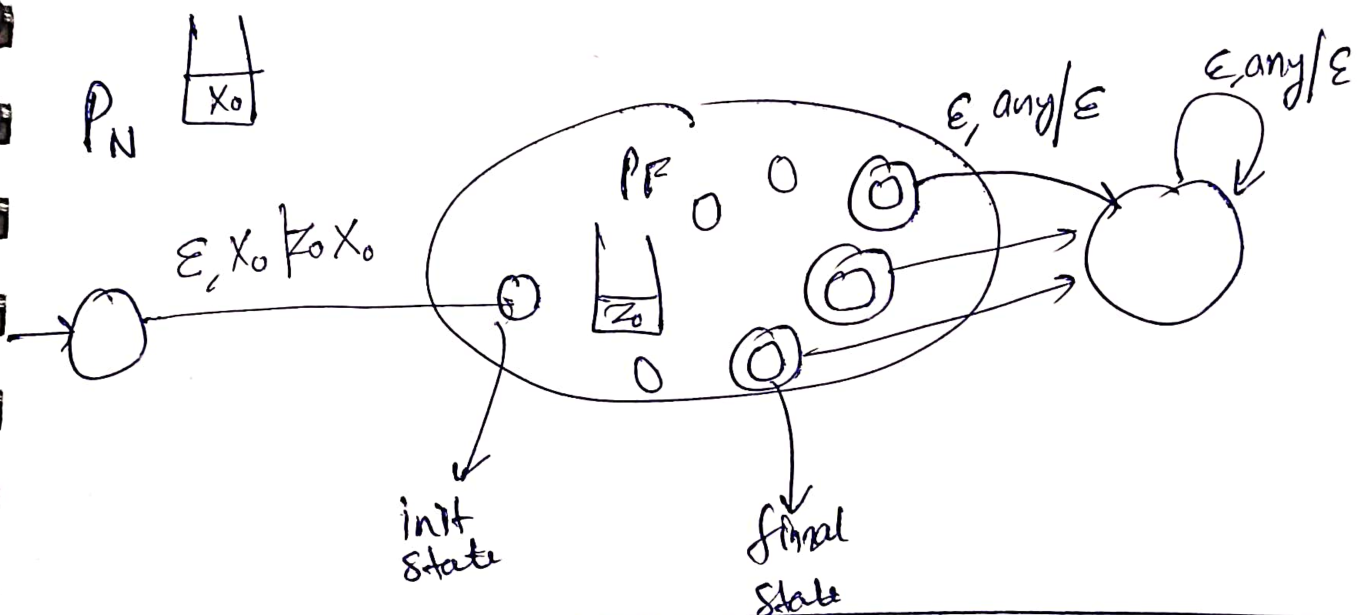
{As soon as P_N pops Z_0, we will get X_0, hence go to final state of P_F}



[some states in P_N, no final states needed since its accepts by empty stack]

Theorem:

Let $L = L(P_F)$ for some PDA $P_F = (Q, \Sigma, \Gamma, \delta_F, q_0, Z_0, F)$. Then there is a PDA such that $L = N(P_N)$.



To prove: $\nexists$ No DPDA for $\{ww^r \mid w \in \{0,1\}^*\}$

Proof: Suppose Not:

$\Rightarrow$ A DPDA P exists.