KTU NOTES

The learning companion.

KTU STUDY MATERIALS | SYLLABUS | LIVE NOTIFICATIONS | SOLVED QUESTION PAPERS

17/1/22

Module - 3

Myhill - Nerode Relations and context Free Grammars

Context Free Grammar

→ V, T, P, S

→ P of the form $\alpha \longrightarrow \beta$

α → single non terminal, $\alpha \in V$

$\beta \in (V \cup T)^*$

A context free grammar is defined as a quadruple (V, T, P, S).

V - set of non-terminals or variables.

T - set of terminals

P - set of production rules

S - start signal.

The production rules are defined as $\alpha \longrightarrow \beta$ where α is a single non-terminal. (α is a number of p)

β can be any combination of termind and non-termind

β is a member of $(V \cup T)^*$.

Note - :

* Every regular grammar is also a CFG.

Derivation of Strings in CFG

→ Leftmost Derivation
→ Rightmost Derivation

$S \longrightarrow AB \mid \varepsilon$
$A \longrightarrow aB$
$B \longrightarrow Sb$

Leftmost Derivation - :

$S \longrightarrow AB$
$S \longrightarrow aBB$
$S \longrightarrow aSbB$
$S \longrightarrow a\varepsilon bB$
$S \longrightarrow a\varepsilon bSb$
$S \longrightarrow a\varepsilon b\varepsilon b$
$S \longrightarrow abb$

Rightmost Derivation - :

$S \longrightarrow AB$
$S \longrightarrow ASb$
$S \longrightarrow A\varepsilon b$
$S \longrightarrow aB\varepsilon b$
$S \longrightarrow aSb\varepsilon b$
$S \longrightarrow a\varepsilon b\varepsilon b \rightarrow abb$

Q $E \longrightarrow E + E$
$E \longrightarrow E - E$
$E \longrightarrow a/b$

Derive 'a - b + a' L & R

Ans) Left - :

$E \longrightarrow E + E$
$E \longrightarrow E - E + E$
$E \longrightarrow a - E + E$
$E \longrightarrow a - b + E$
$E \longrightarrow a - b + a$

Right - :

$E \longrightarrow E - E$
$E \longrightarrow E - E + E$
$E \longrightarrow E - E + a$
$E \longrightarrow E - b + a$
$E \longrightarrow a - b + a$

Q Construct a CFG for a^* → R.E ⇒ RL

RG ≤ CFG

Ans a^*

$L = \{ \varepsilon, a, aa, \ldots \}$

$S \longrightarrow \varepsilon$

$S \longrightarrow aS \longrightarrow$ $S \longrightarrow \varepsilon | aS$

a^+

$L = \{a^*, aa, aaa, \ldots\}$

$S \longrightarrow aS | a$

Q Construct a CFG for $(0+1)^*$

Ans. $(0+1)(0+1)(0+1)\ldots$

$S \longrightarrow 0|1|0S|1S$

Q Construct a CFG for

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

$\{abb, aabbbb \ldots\}$

$S \longrightarrow aSbb/abb$

Q Construct a CFG for palindrome {0. given by $\{0,1\}^*$

Ans $L = \{\varepsilon, 0, 1, 00, 11, 000, 010, 111, 0000, 1001 \ldots\}$

$S \longrightarrow 0S0|1S1|0|1|\varepsilon$

Q Construct a CFG for $0^*|1^*$.

Ans $S \longrightarrow A|B$

$A \longrightarrow 0A|\varepsilon$ $\quad S \longrightarrow A|A$

$B \longrightarrow 1B|\varepsilon$ $\quad A \longrightarrow 0S|1S|\varepsilon$

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

Ans $S \longrightarrow aSb|ab$

Derivation tree or Parse tree

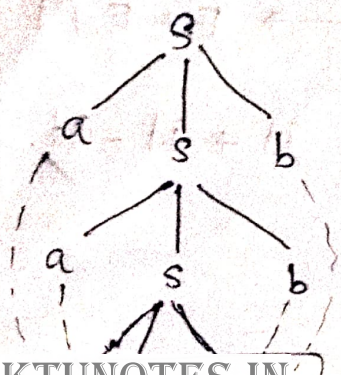
It is a graphical representation for the derivation of a given string from the production rules for a given CFG. Parse tree contains following properties -:

1) The root node is always a node indicating start symbol
2) Derivation is read from left to right
3) The leaf node is always terminal symbol
4) The interior node are always non-terminals.

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

$S \longrightarrow aSb|ab$

eg. $\omega = aaabbb$



aaabbb

Q Draw the parse tree for

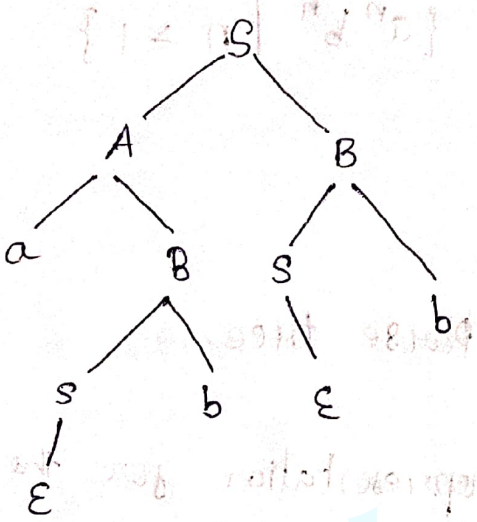
"abb"

$S \longrightarrow AB \mid \varepsilon$

$A \longrightarrow aB$

$B \longrightarrow Sb$

Ans



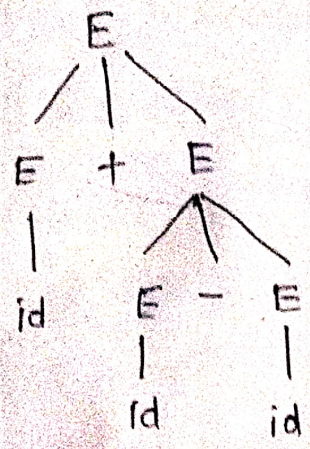
Ambiguity

A grammar is said to be ambigous if there exist more than one leftmost derivation or more than one right most derivation or more than one parse tree for a given string. If grammar is not ambigous then it is unambiguos.

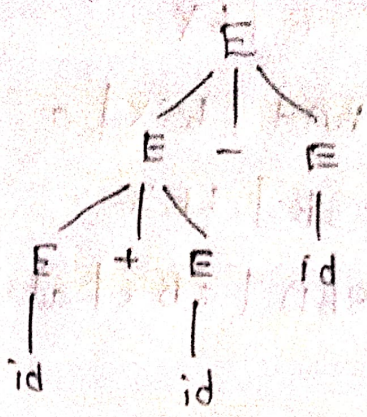
eg. $E \longrightarrow E+E \mid E-E \mid id$

String id + id - id.

(1)



(2)



* IMP **Simplification of CFG**

* Removal of useless symbols

* Removal of Unit productions

* Removal of Null Productions

A symbol Y in a CFG is useful iff

* $Y \xRightarrow{*} w$, where $w \in L(G)$ and $w \in T^*$, that is Y leads to a string of terminals. Then Y is called generating.

* If there is a derivation $S \xrightarrow{*} \alpha Y \beta \xRightarrow{*} w$, $w \in L(G)$ for some α and β, then Y is reachable.

(1) Removal of useless symbol

$S \xrightarrow{*} AB \mid a$

$A \rightarrow b$

B useless symbol

$S \rightarrow a$

$A \rightarrow b^*$

A is useless

Q $S \longrightarrow aB \mid bX$

$A \longrightarrow BAd \mid bSX \mid a$

$B \longrightarrow aSB \mid bBX$

$X \longrightarrow SBD \mid aBx \mid ad$

Ans $S \longrightarrow aB \mid bX$

$B \longrightarrow aSB \mid bBX$

$X \longrightarrow aBx \mid ad$

$S \longrightarrow bX$

$X \longrightarrow ad$

Q $A \longrightarrow xyz \mid Xyzz$

$X \longrightarrow Xz \mid xYz$

$Y \longrightarrow yYy \mid Xz$

$Z \longrightarrow zy \mid z$

Ans

Y x

Z x

X x

$A \longrightarrow xyz \mid Xyzz$

$\Rightarrow A \longrightarrow xyz$

Unit Production –:

$\alpha \longrightarrow \beta$ where both α and β are single non-terminal.

(2) Removal of Unit Productions –:

$A \longrightarrow B \Rightarrow$ unit production

$B \longrightarrow a$

Replace $A \longrightarrow B$

$A \longrightarrow a$

eg. CFG –:

$S \longrightarrow AB$

$A \longrightarrow a$

$B \longrightarrow C \mid b$

$C \longrightarrow D$

$D \longrightarrow E$

$E \longrightarrow a$

Unit productions in grammar are –:

$B \longrightarrow C$

$C \longrightarrow D$

$D \longrightarrow E \Rightarrow D \longrightarrow a$

$\therefore C \longrightarrow a$

$B \longrightarrow a$

$\Rightarrow$

$S \longrightarrow AB$

$A \longrightarrow a$

$B \longrightarrow a \mid b$

$C \longrightarrow a$ (unreachable $\therefore$ useless)

$D \longrightarrow a$ (")

$E \longrightarrow a$ (")

After removing useless symbols $\Rightarrow$

$S \longrightarrow AB$

$A \longrightarrow a$

$B \longrightarrow a \mid b$

Q $S \longrightarrow A \mid bb$

$A \longrightarrow B \mid b$

$B \longrightarrow S \mid a$

Ans Unit productions are :-

$S \longrightarrow A$

$A \longrightarrow B$

$B \longrightarrow S$

$\Rightarrow$ $S \longrightarrow b$ ($S \rightarrow A$, $A \rightarrow b$)

$S \longrightarrow a$ ($S \rightarrow A$, $A \rightarrow B$, $B \rightarrow a$)

$A \longrightarrow a$ ($A \rightarrow B$, $B \rightarrow a$)

$A \longrightarrow bb$ ($A \rightarrow B$, $B \rightarrow S$, $S \rightarrow bb$)

$B \longrightarrow bb$ ($B \rightarrow S$, $S \rightarrow bb$)

$B \longrightarrow b$ ($B \rightarrow S$, $S \rightarrow A$, $A \rightarrow b$)

$\Rightarrow$ $S \longrightarrow a \mid b \mid bb$ (bb was in the start)

$A \longrightarrow a \mid bb \mid b$

$B \longrightarrow b \mid bb \mid a$

(3) Removal of Null productions

Null productions / ε-productions of the form $A \longrightarrow \varepsilon$, where A is non terminal.

A non-terminal N can be called nullable if there is a production $N \longrightarrow \varepsilon$ or There is a derivation that starts at N and leads to ε; $N \overset{*}{\Longrightarrow} \varepsilon$.

If $A \longrightarrow \varepsilon$, then we look for all productions whose RHS contains A and replace each occurrence of A in each of these productions.

Eg. $S \longrightarrow aA$

$A \longrightarrow b/\varepsilon$

$A \longrightarrow \varepsilon$

$S \longrightarrow aA \Rightarrow a.\varepsilon \Rightarrow a$

$\Rightarrow$

$S \longrightarrow aA/a$

$A \longrightarrow b$ (remove ε production)

Q Remove null production

$S \longrightarrow ABAC$

$A \longrightarrow aA/\varepsilon$

$B \longrightarrow bB/\varepsilon$

$C \longrightarrow c$

$A \longrightarrow \varepsilon$

$B \longrightarrow \varepsilon$

Look for occurrence of A in RHS :

$S \longrightarrow ABAC/BAC/ABC/BC$ (Replace A with ε)

$A \longrightarrow aA/a$ $\quad C \longrightarrow c$ $\quad B \longrightarrow bB/\varepsilon$

Look for occurence of B in RHS -: (in new grammer)

$S \longrightarrow ABAC/BAC/ABC/BC/AAC/AC/C$

$A \longrightarrow aA/a$

$C \longrightarrow c$

$B \longrightarrow bB/$

Chomsky Normal Form (CNF)

→ Impose some restrictions on the way productions are written.

→ A CFG is in CNF if all the productions in the CFG is of the form

$$A \longrightarrow BC \quad \text{where} \quad A, B, C \in V$$

or

$$A \longrightarrow a \qquad a \in T$$

All productions such as $\langle NT \rangle \longrightarrow \langle NT \rangle \langle NT \rangle$ should be of the form. $\langle NT \rangle \longrightarrow \langle T \rangle$

Steps to convert a CFG to CNF

① If the start symbol S occurs on the R.H.S, create a new start symbol S' and add a new production $S' \longrightarrow S$

② Remove all null productions

③ Remove all unit productions

④ Replace each production $A \longrightarrow B_1 B_2 \dots B_n$ where $n > 2$, with $A \longrightarrow B_1 C$ where $C \longrightarrow B_2 \dots B_n$.

Repeat this step for all productions having 2 or more symbols on R.H.S

⑤ If the RHS of the production is of the form $A \longrightarrow aB$, where 'a' is a terminal and A & B are non-terminals, then the productions is replaced by $A \longrightarrow XB$ and $X \longrightarrow a$.

Repeat this step for every production which is of the form $A \longrightarrow aB$

Q Convert to CNF

$S \longrightarrow ASA \mid aB$

$A \longrightarrow B \mid S$

$B \longrightarrow b \mid \varepsilon$

Ans)

1. $S' \longrightarrow S$

$S \longrightarrow ASA \mid aB$

$A \longrightarrow B \mid S$

$B \longrightarrow b \mid \varepsilon$

2. Remove null productions

① $B \longrightarrow \varepsilon$

$S' \longrightarrow S$
$S \longrightarrow ASA \mid aB$
$A \longrightarrow B \mid S$
$B \longrightarrow b$

$\Rightarrow$

$S' \longrightarrow S$
$S \longrightarrow ASA \mid aB \mid a$
$A \longrightarrow B \mid S \mid \varepsilon$
$B \longrightarrow b$

$\Rightarrow$ ② $A \longrightarrow \varepsilon$

$S' \longrightarrow S$
$S \longrightarrow ASA \mid aB \mid a \mid SA \mid AS \mid S$
$A \longrightarrow B \mid S$
$B \longrightarrow b$

3. Remove all unit productions

$S' \longrightarrow S$
$S \longrightarrow S \times$
$A \longrightarrow S$
$A \longrightarrow B$

$\Rightarrow$

$S' \longrightarrow S$
$S \longrightarrow ASA \mid aB \mid a \mid AS \mid SA$
$A \longrightarrow B \mid S$
$B \longrightarrow b$

$\Rightarrow$

$S' \longrightarrow ASA \mid aB \mid a \mid AS \mid SA$
$S \longrightarrow ASA \mid aB \mid a \mid AS \mid SA$
$A \longrightarrow b \mid ASA \mid aB \mid a \mid AS \mid SA$

4. $S' \longrightarrow AX \mid aB \mid a \mid AS \mid SA$ → Already in CNF no need of changing

$S \longrightarrow AX \mid aB \mid a \mid AS \mid SA$

$A \longrightarrow b \mid AX \mid aB \mid a \mid AS \mid SA$

$B \longrightarrow b$

$X \longrightarrow SA$

$A \longrightarrow B_1 B_2 \cdots B_n$

↓

No production should be of this form

5. $S' \longrightarrow AX \mid YB \mid a \mid AS \mid SA$

$S \longrightarrow AX \mid YB \mid a \mid AS \mid SA$

$A \longrightarrow b \mid AX \mid YB \mid a \mid AS \mid SA$

$B \longrightarrow b$

$X \longrightarrow SA$

$Y \longrightarrow a$

(aB was not in CNF)

Greibach Normal Form (GNF)

A CFG $G = (V, T, P, S)$ is in Greibach Normal Form if each rule / production in P is of the form,

$A \longrightarrow xbT^*$

$lhs(\gamma) \in V$

$rhs(\gamma) = a\alpha \; ; \; a \in T \quad \alpha \in V^*$

or $A \longrightarrow aB_1, \ldots B_n$ where $A, B, \ldots B_n \in$ Non terminals (V)

$A \longrightarrow a \quad a \in T$

LHS - single non-terminal

RHS - Terminal followed by n non-terminals ($n \geq 0$)

Steps

1. Remove the null & unit productions
2. Check if the given CFG is in CNF, else convert to CNF
3. Remove any left recursion.
4. Convert the resultant production to GNF form.

Q $S \longrightarrow xB \mid AA$

$A \longrightarrow a \mid SA$

$B \longrightarrow b$

$x \longrightarrow a$

Ans)

No unit and null productions

CNF

~~Production~~ already in CNF, hence no need to convert.

No left recursion currently.

$A \rightarrow a$, $B \rightarrow b$, $X \rightarrow a$ are in CNF, convert others.

$S \longrightarrow XB \Rightarrow$

Substitute $X \rightarrow a$ in $S \longrightarrow XB$

$S \longrightarrow aB \mid AA$

$A \longrightarrow a \mid SA$

$B \longrightarrow b$

$X \longrightarrow a$

$A \longrightarrow SA \Rightarrow$

Substitute $S \rightarrow aB \mid AA$ in $A \rightarrow SA$

$S \longrightarrow aB \mid AA$

$A \longrightarrow a \mid aBA \mid AAA$

$B \longrightarrow b$

$X \longrightarrow a$

$\Rightarrow$

$S \longrightarrow aB \mid AA$

$A \longrightarrow a \mid aBA \mid ac \mid aBAC$

$C \longrightarrow AAC \mid AA$

$B \longrightarrow b$

$X \longrightarrow a$

Left recursion

$A \rightarrow AAB$

$A \rightarrow AB$

$A \rightarrow Aa$

$A \rightarrow (aB/AA)A$

$A \rightarrow aBA / AAA$

$A \rightarrow AAA$

left recursion,

New symbol C

$C \rightarrow AAC / \varepsilon$

$A \rightarrow aC / aBAC$

Remove ε

$C \rightarrow AAC / AA$

$A \rightarrow a / aBA / aC / aBAC$

$\Rightarrow$

$S \rightarrow aB \mid aA \mid aBAA \mid aCA \mid aBACA$

$A \rightarrow a \mid aBA \mid aC \mid aBAC$

$C \rightarrow aAC \mid aBAAC \mid aCAC \mid aBACAC$

$C \rightarrow aA \mid aBAA \mid aCA \mid aBACA$

$B \rightarrow b$

$X \rightarrow a$

$S \rightarrow AA$

$S \rightarrow (a \mid aBA \mid aC \mid aBAC)A$

$\cdot\ aA \mid aBAA \mid aCA \mid aBACA$

Q Convert to GNF

$S \longrightarrow CA$

$S \longrightarrow BB$

$B \longrightarrow b$

$B \longrightarrow SB$

$A \longrightarrow a$

Ans $S \longrightarrow CA \mid BB$

$B \longrightarrow SB \mid b$

$A \longrightarrow a$

$C \longrightarrow b$

Substituting $C \longrightarrow b$ in $S \longrightarrow CA$

$S \longrightarrow bA \mid BB$

$B \longrightarrow SB \mid b$

$A \longrightarrow a$

$C \longrightarrow b$

$B \longrightarrow SB$

$S \longrightarrow bA \mid BB$

Substitute $S \longrightarrow bA \mid BB$ in $B \longrightarrow SB$

$S \longrightarrow bA \mid BB$

$B \longrightarrow bAB \mid BBB \mid b$

$A \longrightarrow a$

$C \longrightarrow b$

Removing left recursion,

$B \longrightarrow bAB \mid BBB \mid b$

Introducing new non terminal X,

$X \longrightarrow BBX \mid BB$

$B \longrightarrow bAB \mid b \mid bABX \mid bX$

$S \rightarrow bA \mid \tilde{B}B$

$B \rightarrow bAB \mid b \mid bABX \mid bX$

$X \rightarrow \overset{\times}{B}BX \mid \overset{\times}{B}B$

$A \rightarrow a$

$C \rightarrow b$

$\Rightarrow$

$S \rightarrow bA \mid bABB \mid bB \mid bABXB \mid bXB$

$B \rightarrow bAB \mid b \mid bABX \mid bX$

$X \rightarrow bABBX \mid bBX \mid bABXBX \mid bXBX$

$X \rightarrow bABB \mid bB \mid bABXB \mid bXB$

$A \rightarrow a$

$C \rightarrow b$

---

* Correctness of CFG

Checking if the language generated by CFG is CFL

If L is some language and G is a CFG, then to prove that $L = L(G)$.

For this check the following:-

1) Every string ω generated by G is in L.
2) Every string ω in L can be generated by G.

eg.

Let $L = \{0^i 1^j \mid 2i \leq j \leq 3j\}$

Grammar for this,

$$S \longrightarrow 0S11 \mid 0S111 \mid \varepsilon$$

zero

Claim: (this is the language generated by grammar)

If string ω is generated by G, then $\omega \in L$

Proof: (Proof by induction on the number of steps taken to generate thus ω)

Base case: derivation of ω with single step.

$S \xrightarrow{*} \omega$ $\quad S \xrightarrow{1} \omega$, $\quad \omega = \varepsilon$

$L = \{0^i 1^j \mid 2i \leq j \leq 3j\}$

$i = j = 0$

$0^0 1^0 = \varepsilon$

$L = \{\varepsilon, \ldots\}$

Base case stands

* Inductive hypothesis: Assume that for every derivation $S \xrightarrow{*} \omega$ with $n \geq 1$ steps, $\omega \in L$.

* Inductive step: prove that every derivation with $n+1$ steps generate string in L.

Let $S \xrightarrow{*} \omega$ be a derivation with $n+1$ steps and $n+1 > 1$, then first step in derivation cannot be $S \longrightarrow \varepsilon$ (it will halt)

So the derivation starts from ① $S \longrightarrow 0S11$ or ② $S \longrightarrow 0S111$

In the first case, w_1 must be of the form $0w_1 11$ where w_1 is a string derived from S in the remaining 'n' steps..

$w_1 \in L$ or $w_1 = 0^i 1^j$

$w = 0\,0^i 1^j 11 = 0^{i+1} 1^{j+2} \in L$

$\therefore w \in L$

$\therefore 2i \le j \le 3i$

$2i+2 \le j+2 \le 3i+2$

$2(i+1) \le j+2 \le 3(i+1)$

Claim II

Every string $w \in L$, can be generated by G.

Proof

$L = \{\varepsilon, 011, 0111, 001111, 0011111, \ldots\}$

$S \rightarrow 0S11 \mid 0S111 \mid \varepsilon$

$L = \{0^i 1^j \mid 2i \le j \le 3j$

We will do induction on the number of 0's in w

$n \ge 0$, L contains exactly strings $0^n 1^{2n}, 0^n 1^{3n+1}, \ldots$

For any given $n > 0$ and any given k, $0 \le k \le n$, the string $w = 0^n 1^{2n+k}$ can be generated by G.

* Base case : $n = 1$

$0 \le k \le n$ or $0 \le k \le 1$, $k = 0, 1$

If $k = 0$, $\omega = 0^n 1^{2n+k} = 0^1 1^{2+0} = 011$

$S \longrightarrow 0S11 \longrightarrow 011 \longrightarrow$ for $k = 0$, it is valid

If $k = 1$, $\omega = 0^n 1^{2n+k} = 0^1 1^{2+1} = 0111$

$S \longrightarrow 0S111 \longrightarrow 0111 \longrightarrow$ valid

* Inductive hypothesis :

Assume that the claim is true for $n = i$, that is $0 \le k \le i$, the string $\omega = 0^i 1^{2i+k} \in L$ can be generated by G.

* Inductive step :

Prove the claim for $n = i+1$. Let

$\omega = 0^{i+1} 1^{2(i+1)+k}$ with $0 \le k \le i+1$

(i) $0 \le k \le i$, $\omega = 0^{i+1} 1^{2(i+1)+k}$

$= 0^{i+1} 1^{2i+2+k}$

$= 00^i 1^{2i+k} 11$

Inductive hypothesis says $0^i 1^{2i+k} \in L$ can be generated by G

$S \xrightarrow{*} 0^i 2^{2i+k}$ and $\therefore$

$S \xrightarrow{*} 0S11 \longrightarrow 00^i 1^{2i+k} 11 = \omega \in L$ //

(iii) $k = i+1$

$$w = 0^{i+1}1^{2(i+1)+i+1} = 0^{i+1}1^{3i+3}$$

$$= 00^{i}1^{3i}111$$

$$S \xrightarrow{*} 0^{i}3^{i}$$

$$S \longrightarrow 0S111 \longrightarrow 00^{i}1^{3i}111$$ true

Hence the CFG, G is correct for L.

* Myhill Nerode Relations

Equivalence Relations

→ It is a binary relation that is reflexive, symmetric and transitive

eg. $X = \{a, b, c\}$, $R = \{(a,a), (b,b), (c,c), (b,c), (c,b)\}$

1. Reflexive → aRa for all $a \in X$
2. Symmetric → $aRb \rightarrow bRa \quad \forall a, b \in X$
3. Transitive → $aRb \cap bRc \rightarrow aRc \quad \forall a, b, c \in X$

∴ X is equivalence relation. $a \equiv b$, etc.

Myhill - Nerode relations :-

Let $R \subseteq \Sigma^*$ be a regular set, and let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for R with no inaccessible states (minimal). Then the autom[illegible] valence relation

$\equiv_m$ on Σ^* defined by

$$x \equiv_m y \overset{\text{def}}{\iff} \hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$$

Myhill - Nerode Relation should be :-

1) Equivalence relation

2) H is a RIGHT CONGRUENCE

for any $x, y \in \Sigma^*$ and $a \in \Sigma$

$$x \equiv_m y \Rightarrow xa \equiv_m ya$$

eg. $x \equiv_m y$ then $\hat{\delta}(q_0, xa) = \hat{\delta}(\hat{\delta}(q_0, x), a)$

$= \delta(\hat{\delta}(q_0, y), a)$

$= \hat{\delta}(q_0, ya)$

3) If refines R for any $(x, y) \in \Sigma^*$

$$x \equiv_m y \Rightarrow (x \in R \iff y \in R)$$

4) If is of finite index

ie. it has only finite no. of equivalence classes.

$\{ x \in \Sigma^* \mid \hat{\delta}(s, x) = q \}$

corresponding to each state q of m

Myhill Nerode Theorem

The following three statements are equivalent:

1) The set $L \subseteq \Sigma^*$ is accepted by some FSA
2) L is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.
3) L is equivalence relation R_L be defined by $x R_L y$ iff for all z in Σ^* xz is in L exactly when yz is in L. Then R_L is of finite ~~number~~ index.

* A language is regular iff $\equiv_L$ partitions Σ^* into finitely many equivalence classes. If $\equiv_L$ partitions Σ^* into n equivalence classes then the minimal DFA recognizing L has exactly n states.

Applications of Myhill Nerode Theorem:

* Minimization of FA
* To prove the languages are regular or not.

$L = \{a^n b^n \mid n \geq 0\}$ $\Sigma = \{a, b\}$

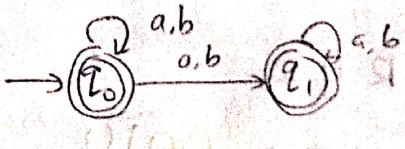
$x = a^k$ $y = a^i$ $a^n \mid n \geq 0$ are regular

$k \neq i$ $z = b^k$ $b^n \mid n \geq 0$

$a^k b^k$ will satisfy but $a^i b^k$ will not satisfy the criteria : $i \neq k$

$x R y \rightarrow xz \; R \; yz$

DFA - :



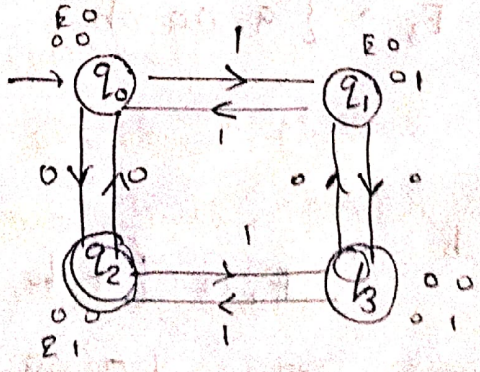
will not be finite index class.

∴ Language is not regular.

Q Convert to equivalence classes

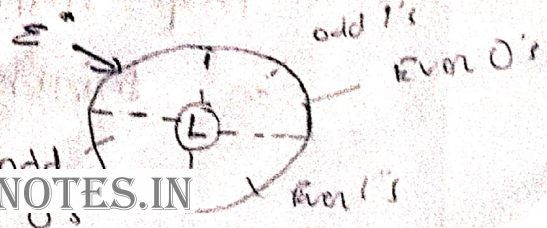
$L = \{\omega \in \{0,1\}^* \mid \omega$ has odd number of 0^s and even number of $1^s\}$

Ans First convert to minimal DFA



No. of states in the min. DFA = No. of equivalence classes

= 4



$q_0 = \{\varepsilon, 0011, 000011, 00, 11, 00001111 \ldots\} = E_1$

$q_1 = \{1, 001, 00001, 00111, \ldots\} = E_2$

$q_2 = \{0, 110, 11110, 11000, \ldots\} = E_3$

$q_3 = \{01, 0001, 0111, \ldots\} = E_4$

Select arbitrary $z \in \Sigma^*$

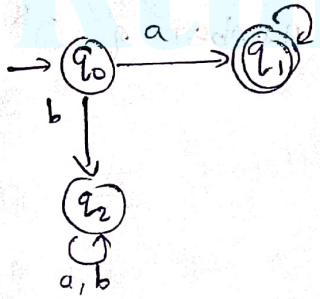
$z = 0$

$x R y \Rightarrow xz R yz$ $\quad x, y \in E_2$

eg. $xz = \underline{10}$ $\quad yz = \underline{0010}$

$\hookrightarrow E_4$ $\quad \hookrightarrow E_4$

Q $L = \{$ strings starting with 'a' $\}$ $\quad \Sigma = \{a, b\}$



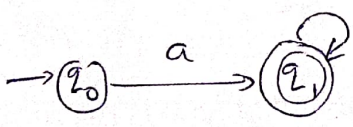
$\Rightarrow$ 3 eq. classes.

$E_1 = \{\varepsilon\}$

$E_2 = \{b, ba, bb, \ldots\}$

$E_3 = \{a, aa, ab, \ldots\}$

or



$E_1 = \{b, ba, bb \ldots\}$

$E_2 = \{a, aa, ab, \ldots\}$

Trap state need not be considered in Minimal DFA.