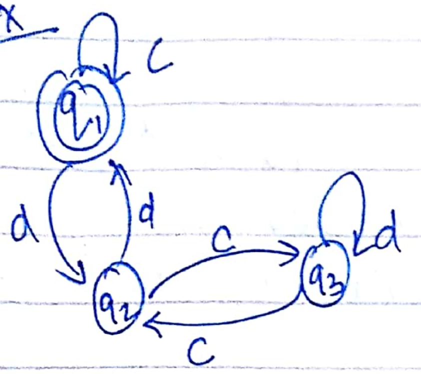
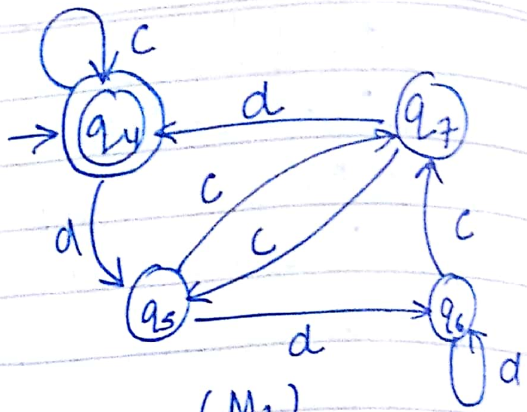
Ex





(M_2)

M_1	c	d
$(q_1 q_4)$	$(q_1 q_4)$	$(q_2\, q_5)$
$(q_2\, q_5)$	$(q_3\, q_7)$	$(q_1,\, q_6)$ → final, non final Stop here also both are non equivalent

till Now Mid Sem

Date 14/Mar/18

* Context free grammer   Unit 4

$G = (V_N, \Sigma, P, S)$

$\underbrace{A \rightarrow \alpha}_{\text{Type 2 production}}$, where $A \in V_N$ & $\alpha \in (V_N \cup \Sigma)^*$

Type 2 - grammer, or CFG context free grammer.

, Language generated by CFG is CFL

* Derivation Tree of a CFG

Defn :- A derivation for a CFG, $G = (V_N, \Sigma, P, S)$ is a tree satisfying the following conditions.

(i) Every Vertex has a label which is a Variable or terminal or $\wedge$

(ii) The root has label S

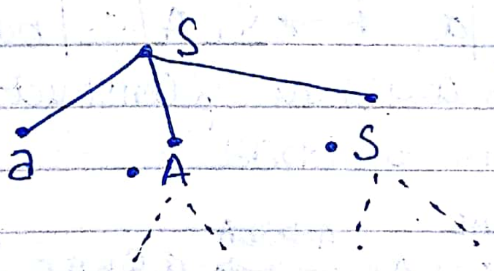
(iii) The label of an internal Vertex is a Variable

(iv) if n is a Vertex with label A has K Child nodes

Say $n_1 n_2 \ldots n_K$ which are labelled with $X_1, X_2 \ldots X_K$ Resp. then $A \to X_1 X_2 \ldots X_K$ is a production in P

(V) A Vertex is a leaf if its label any Terminal Symbol Say $a \in \Sigma$ or $\wedge$

$S \to aAS$



Derivation Tree

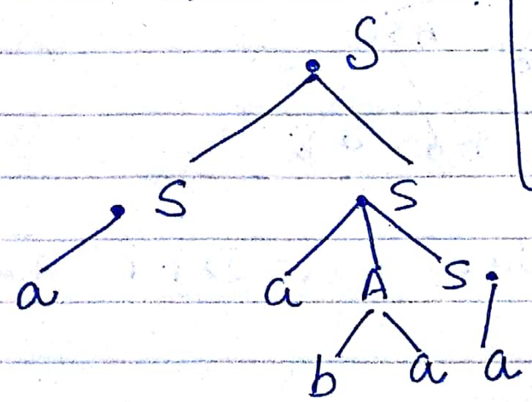
Example

$G = (\{S, A\}, \{a, b\}, P, S)$

Where P Contains $(S \to SS / a / aAS,\ A \to ba)$

Construct Derivn Tree?

$\Rightarrow$



aabaa

$S \to SS \xrightarrow{S \to a} aS \xrightarrow{S \to aAS} aaAS \xrightarrow{A \to ba} aabaS \xrightarrow{S \to a} aabaa$

$aabaa \subset L(G)$

Concatenation of leaf nodes from Left to Right is yield of tree (D-T)

aabaa

Defn

A Subtree of a Derivation tree is a tree
(i) Whose root node is some vertex v of T
(ii) whose vertices are the descendents of V together with their labels
(iii) whose edges are those connecting the descendents of v.

A-tree: Subtree of D.T. whose root node is labelled with label A

Q

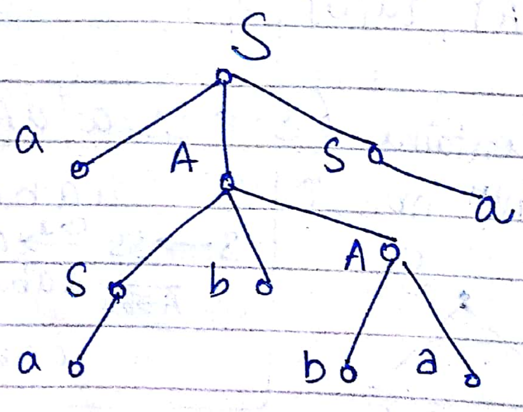
Suppose $G(V_N, \Sigma, P, S)$

$$S \to aAS \mid a, \quad A \to SbA \mid SS \mid ba$$

Show that $S \overset{*}{\underset{G}{\Rightarrow}} aabbaa$ & Construct a derivn tree whose yield is aabbaa.

Sol

(i) $S \xrightarrow{S \to aAS} aAS \xrightarrow{A \to SbA} aSbAS \xrightarrow{S \to a} aabAS$
$\xrightarrow{A \to ba} aabbaS \xrightarrow{S \to a} aabbaa$



$\Rightarrow$ Yield of above DT is aabbaa

Other way possible

(ii) $S \xrightarrow{S\to aAS} aAS \xrightarrow{S\to a} aAa \xrightarrow{A\to SbA} aSbAa \xrightarrow{A\to ba}$
$aSbbaa \xrightarrow{S\to a} aabbaa$

(iii) $S \xrightarrow{S\to aAS} aAS \xrightarrow{A\to SbA} aSbAS \xrightarrow{S\to a} aSbAa \xrightarrow{S\to a}$
$aabAa \xrightarrow{A\to ba} aabbaa$

$S \overset{*}{\Rightarrow} w$ is a left most derivation if we always replace leftmost variable in obtaining w, derivn
(i) is its example

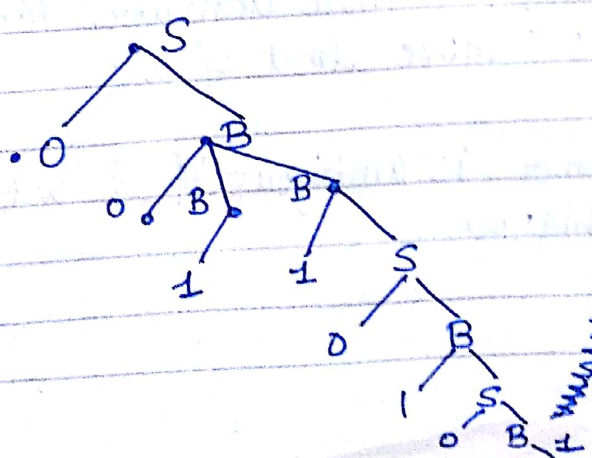
(ii) is Rightmost derivation

(iii) is a mixture, not a particular type.

Ex. $S \to 0B \mid 1A$, $A \to 0 \mid 0S \mid 1AA$, $B \to 1 \mid 1S \mid 0BB$

for the String 00110101 find left most derivn, Rightmost derivn, & derivn tree.

Sol
(i) $S \xrightarrow{S\to 0B} 0B \xrightarrow{B\to 0BB} 00BB \xrightarrow{B\to 1} 001B \xrightarrow{B\to 1S} 0011S$
$\xrightarrow{S\to 0B} 00110B \xrightarrow{B\to 1S} 001101S \xrightarrow{S\to 0B} 0011010B$
$\xrightarrow{B\to 1} 00110101$ [L.M.D]



(ii) R.M.D

$S \xrightarrow{S\to 0B} 0B \xrightarrow{B\to 0BB} 00BB \xrightarrow{B\to 1S} 00B1S \xrightarrow{S\to 0B} 00B10B$

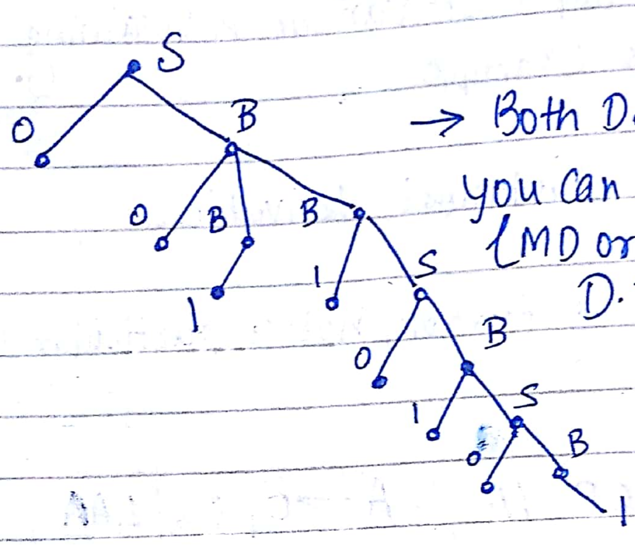
$\xrightarrow{B\to 1S} 00B101S \xrightarrow{S\to 0B} 00B1010B$

$\downarrow B\to 1$

$00B10101$

$\downarrow B\to 1$

00110101



→ Both D.T are Same

you can use any derivn (LMD or RMD) for Constructing D.T.

Date :- 19/Mar/18

* Ambiguity in CFG

→ A terminal String $w \in L(G)$ is ambiguous if $\exists$ two or more derivation tree of w (or 2 or more lmd of w)

→ A Grammar is ambigous if $\exists$ $w \in L(G)$ which is ambiguous.

__Example__

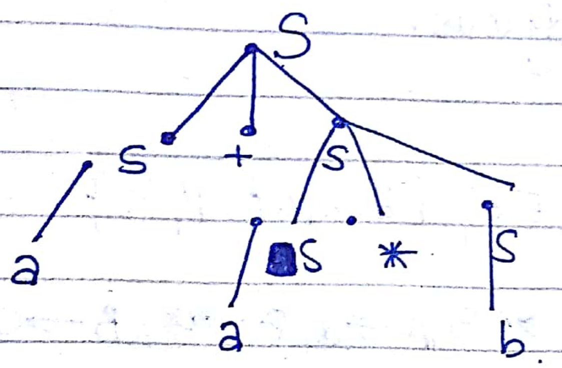
$G = (\{S\}, \{a, b, +, *\}, P, S)$ where P consists of

$$S \to S+S \mid S*S \mid a \mid b$$

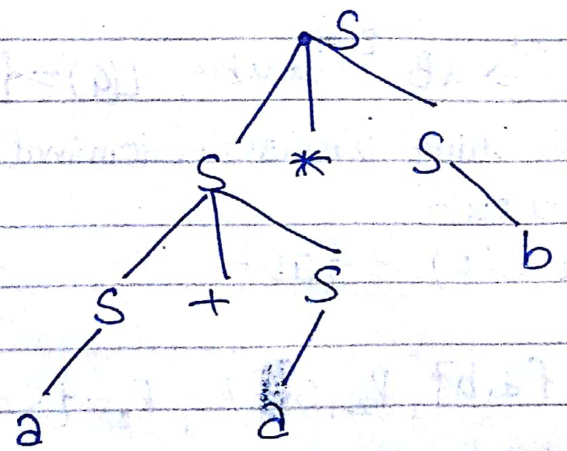
let $w = a+a*b$

⇒ to find the ambiguity of w using LMD

① $S \to S+S \xrightarrow{S\to a} a+S \xrightarrow{S\to S*S} a+S*S \xrightarrow{S\to a} a+a*S \xrightarrow{S\to b} a+a*b$



② $S \to S*S \xrightarrow{S\to S+S} S+S*S \xrightarrow{S\to a} a+S*S \xrightarrow{S\to a} a+a*S \xrightarrow{S\to b} a+a*b$



hence w is ambigous

hence grammer G is ambigous.

Scanned by CamScanner

abba $Sbs \rightarrow Sbsbc$ ababa

$Sbs \rightarrow Sbsbs$

Example

$S \rightarrow Sbs \mid a$ · Show that G is ambiguous.?

Sol

Take $w = ababa$.

$S \rightarrow Sbs \xrightarrow{S \rightarrow a} abs \xrightarrow{S \rightarrow Sbs} abSbs \xrightarrow{S \rightarrow a} ababs \xrightarrow{S \rightarrow a} ababa$

$S \rightarrow Sbs \xrightarrow{S \rightarrow Sbs} Sbsbs \xrightarrow{S \rightarrow a} absbas \xrightarrow{S \rightarrow a} ababs \xrightarrow{S \rightarrow a} ababa$

→ hence 2 different derivn tree exist for w
hence G is ambiguous.

Q $G_1 = \{\{S, A, B, C, E\}, \{a, b, c\}, P, S\}$
Where

$P = \{S \rightarrow AB, A \rightarrow a, B \rightarrow b, B \rightarrow C, E \rightarrow c/\Lambda\}$

find $L(G_1)$?

Sol $L(G_1) = \{w \in \Sigma^* : S \stackrel{*}{\Rightarrow} w\}$

$S \rightarrow AB \xrightarrow{A \rightarrow a} aB \xrightarrow{B \rightarrow b} ab$, $L(G) = \{ab,$

$S \rightarrow AB$ → No String consists of terminal can be found out

hence $L(G) = \{ab\}$

$G_2 = (\{S, A, B\}, \{a, b\}, P_2, S)$, $P_2 = \{S \rightarrow AB, A \rightarrow a, B \rightarrow b\}$

$L(G_2)$ = Same as above $\{ab\}$

→ Both are equivalent Grammer $G_1 \simeq G_2$
Since $L(G_1) = L(G_2)$

* Simplification of CFG

Theorem
if G be a CFG s.t. $L(G) \neq \phi$ we can find an equivalent grammer G' s.t. each Variable in G' derives some terminal strings

Construction of G'
Proof → Read from Book. (we can leave this proof)
→ Suppose $G' = (V_N', \Sigma, P', S)$
Construction of V_N' :
we define $W_i \subseteq V_N$ by recursion

$W_1 = \{A \in V_N \mid \exists$ a production $A \rightarrow w,\ w \in \Sigma^*\}$
$W_{i+1} \overset{W_i \cup}{=} \{A \in V_N \mid \exists$ some production $A \rightarrow \alpha,$
$\alpha \in (\Sigma \cup W_i)^* \}$
$\therefore W_2 = W_1 \cup \{A \in V_N \mid \exists$ some prodn $A \rightarrow \alpha,$
$\alpha \in (\Sigma \cup W_1)^* \}$

$\therefore W_i \subseteq W_{i+1}$
at $W_{K+1} = W_K$ for some k $V_N' = W_K$
Stop the production.

Construction of P'

$P' = \{A \rightarrow \alpha,\ \alpha \in (V_N' \cup \Sigma)^*,\ A \in V_N' \}$

Scanned by CamScanner

Example $G = (V_N, \Sigma, P, S)$

$S \to AB$, $A \to a$, $B \to b$, $B \to C$, $E \to c$

$\to$ find an equivalent grammer G'. S.t. Every Variable in G', derives some String of terminals.

Step 1 (Construction of V_N')

$W_1 = \{A, B, E\}$
$W_2 = W_1 \cup \{S\} = \{A, B, E, S\}$
$W_3 = \{S, A, B, E\} = W_2$
Stop.
$V_N' = \{S, A, B, E\}$

Step 2 P' (Construction of P')

$P' = \{S \to AB, A \to a, B \to b, E \to c\}$
↳ Every production derives String of terminals

Theorem 2

for every CFG, $G = (V_N, \Sigma, P, S)$ we can construct equivalent grammer S.t
$G' = (V_N', \Sigma', P', S)$ S.t every symbol in $(V_N' \cup \Sigma')$ appears in some Sentiontial form.

Construction of G'

$W_1 = \{S\}$
$W_{i+1} = W_i \cup \{X \in V_N \cup \Sigma \mid \exists$ a production $A \to \alpha$ with $A \in W_i$ & α Containing the symbol $X\}$

$\therefore W_i \subseteq W_{i+1}$

at Some stage

$W_{k+1} = W_k$ for Some k

$\Rightarrow V_N' = V_N \cap W_k$

$\Rightarrow \Sigma' = \Sigma \cap W_k$

$\Rightarrow P' = \{A \to \alpha,\ A \in W_k,\ \alpha \in (V_N' \cup \Sigma')^* \}$

Example

$G = (\{S, A, B, E\}, \{a, b, c\}, P, S)$

$S \to AB,\ A \to a,\ B \to b,\ E \to c$

$W_1 = \{S\}$

$W_2 = \{S\} \cup \{A, B\} = \{S, A, B\}$

$W_3 = W_2 \cup \{a, b\} = \{S, A, B, a, b\}$

$W_4 = W_3$

$V_N' = V_N \cap W_k \quad \{S, A, B\}$

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

$P' = \{S \to AB,\ A \to a,\ B \to b\}$

$\to$ Equivalent Grammer.

Date 21/Mar/18

Theorem * elimination of null productions

In a CFG, a variable $A \in V_N$ is a Nullable Variable if $A \overset{*}{\Rightarrow} \Lambda$

if $G = (V_N, \Sigma, P, S)$ is a CFG, then we can find a CFG G_1 having no null productions s.t.

$L(G_1) = L(G) - \{\Lambda\}$

Construction

Step 1

Construction of the set of nullable variable

$W_1 = \{ A \in V_N : A \to \Lambda \text{ is in } P \}$

$W_{i+1} = W_i \cup \{ A' \in V_N : \exists\ A' \to \alpha \text{ with } \alpha \in W_i^* \}$

Obviously, $W_i \subseteq W_{i+1}$. Since V_N is finite at some stage. $W_{k+1} = W_k$ for some k

let $W = W_k$

→ set of nullable variable

Construction of Set of productions

(i) any production whose R.H.S does not have any nullable variable is included in P'

(ii) The productions are obtained either by not erasing any nullable variable on the right hand side of $A \to X_1 X_2 \dots X_k$ or by erasing some or all nullable variables provided some symbol appears on the R.H.S after erasing.

Example : $S \to aS / AB$, $A \to \Lambda$, $B \to \Lambda$, [redacted] $D \to b$

Construct a grammar without NULL productions?

Step 1. $W_1 = \{A, B\}$

$W_2 = \{A, B\} \cup \{S\} = \{A, B, S\}$

$W_3 = \{S, A, B\} \cup \{\ \} = W_2$

$W = \{S, A, B\}$

Construction of set of productions

(i) $D \to b$ is included in P'

(ii) $S \to AB$ gives $S \to AB$ in P', $S \to B$, $S \to A$

$S \to aS$ gives $S \to aS$, $S \to a$

* elimination of Unit Production

G: $S \to A$, $A \to B$, $B \to C$, $C \to d$

$L(G) = \{d\}$

So if we eliminate above prodⁿs & make $S \to d$ then it will solve our purpose

Theorem : If G is a CFG, we can find a CFG G_1 which has no null production or unit production

Step 1 : eliminate null production to a grammar say G', Now In G' we eliminate unit production to get a grammar G_1

Step 2 : Construction of $W_i(A)$ $\forall A \in V_N$

Scanned by CamScanner

$W_0(A) = \{A\}$

$W_{i+1}(A) = W_i(A) \cup \{ B \in V_N : C \to B \text{ is in } P \text{ with } C \in W_i(A) \}$

$W_i \subseteq W_{i+1}$

at Some Stage $W_{K+1}(A) = W_K(A)$ let

$W(A) = W_K(A)$

↳ set of Variables derivable from A

Step 3 : Construction of A-productions

The A- productions in G_1 are either

(i) the non unit production in G'

(ii) $A \to \alpha$ whenever $B \to \alpha$ is in G with $B \in W(A)$ and $\alpha \notin V_N$

* Reduced Grammer

Grammer $G(V_N, \Sigma, P, S)$ [redacted] is a reduced Grammer, If every Variable derives Some String of terminals & every Symbol of $(V_N \cup \Sigma)$ appears in Some Sentential form

$G \xrightarrow{\text{find } G_1 \text{ in which every variable derives some string of terminals}} G_1 = (V_N', \Sigma, P', S) \xrightarrow{\text{every Symbol } (V_N' \cup \Sigma) \text{ appears in some Sentiantial form.}} G_2 \longrightarrow$ reduced Grammer.

~~Find~~

→ If G is CFG we can Construct a equivalent Grammer G' which is reduced and has no null or no unit production

Step 1 eliminate Null production to get G_1

Step 2 eliminate unit production from G_1 to get G_2

Step 3 find the reduced Grammer wrt. G_2

Ex. Find a Reduced grammar equivalent to grammar whose productions
$S \to AB / CA$, $B \to BC / AB$, $A \to a$, $C \to aB / b$

Sol^n

Step 1. ~~Eliminate NULL production~~.
Find G_1 in which every variable derives some string of terminals.

Use Theorem 1. $G \to G_1 \to G_2$.

$G_1 = (V_N', \Sigma, P', S)$

$W_1 = \{A, C\}$
$W_2 = \{A, C\} \cup \{S\} = \{A, C, S\}$
$W_3 = \{A, C, S\}$

$V_N' = \{S, A, C\}$

Production

$P' = \{S \to CA, A \to a, C \to b\}$

$G_1 = (V_N', \Sigma, P', S)$

Step 2. Find G_2 every symbol $(V_N' \cup \Sigma)$ appears in some sentential form

Use Theorem 2

$W_1 = \{S\}$
$W_2 = \{S, A, C\}$
$W_3 = \{S, A, C, a, b\}$
$W_4 = W_3$

Scanned by CamScanner

$V_N'' = \{S, A, C\}$
$\Sigma'' = \{a, b\}$
$P'' = \{S \to AC, A \to a, C \to b\}$

$G'' = (V_N'', \Sigma'', P'', S)$

Date: 26/Mar/18

Normal forms of CFG

$A \to \alpha, A \in V_N, \alpha \in (V_N \cup \Sigma)^*$

1) Chomsky normal form (CNF)
2) Greibach Normal form (GNF)

(CNF)

Chomsky Normal form

$A \to a, a \in \Sigma$ or $A \to BC, A, B, C \in V_N$
if $\Lambda \in L(G)$
$S \to \Lambda$ allowed provided R.H.S of any production does not contain Symbol S

Reduction of CFG into CNF ← * Important ↳ steps

Step 1 (Elimination of null & unit productions) first we eliminate null productions and then for resulting grammer, eliminate unit production let the grammer thus obtained be $G = (V_N, \Sigma, P, S)$

Step 2 (elimination of Terminal on R.H.S)

Define $G_1 = (V_N', \Sigma, P_1, S)$ where P_1 & V_N' are constructed as follows:

(i) all the productions in P of the form $A \to a$ or $A \to BC$ are included in P_1. All the Variables in V_N are included in V_N'

(ii) Consider $A \to X_1 X_2 \cdots X_n$ with some Terminals on R.H.S. If X_i is a terminal say a_i add a new Variable C_{a_i} to V_N' and $C_{a_i} \to a_i$ to P_1. In Production $A \to X_1 X_2 \cdots X_n$ every Terminal on R.HS is replaced by the corresponding new Variable and the Variables on the R.H.S are retained so the resulting production is added to P_1. Thus we get $G_1 = (V_N', \Sigma, P_1, S)$

Step 3 (Restricting the number of Variables on R.H.S)

for any prod. in P_1 the R.H.S Consist of either a Single Terminal or 2 or more Variable So we define say $G_2 (V_N'', \Sigma, P_2, S)$ as follows:—

(i) All productions in P_1 are added to P_2 if they are in the required form, all the variables in V_N' are added to V_N'' and then

(ii) Consider $A \to A_1 A_2 \cdots A_m$, $m \geq 3$ Now Introduce new production $A \to A_1 C_1$ where $C_1 \to A_2 A_3 \cdots A_m$ then $C_1 \to A_2 C_2$ where $C_2 \to A_3 A_4 \cdots A_m$ and these new Variable C_i's are added to V_N''

Thus we G_2 in CNF

Example Reduce the grammer to CNF

$S \to aAD$, $A \to aB / bAB$

, $B \to b$, $D \to d$

Sol

Step 1 elimimate of Null & unit prodn

there are no null & unit prodn.

$G = (V_N, \Sigma, P, S)$

Step 2 eliminate Terminal on RHS

Define $G_1 = (V_N', \Sigma, P_1, S)$

(i) $B \to b$ & $D \to d$ are included in P_1

(ii) $S \to aAD$ Introduce new variable wrt the Terminal a say C_a then $C_a \to a$ will be included in P_1 & C_a in V_N' Now we have

$S \to C_aAD$ in P_1

$A \to aB$ gives $A \to C_aB$ in P_1

$A \to bAB$, Introduce " " "

" " say C_b then $C_b \to b$ " "

" " P_1 & C_b " V_N' Now we have

$A \to C_bAB$

$G_1 = (V_N', \Sigma, P_1, S)$

Where $V_N' = \{S, A, B, D, C_a, C_b\}$

$P_1 = \{B \to b, D \to b, C_a \to a, S \to C_aAD$

$, A \to C_aB, A \to C_bAB, C_b \to b\}$

Step 3 here we define $G_2 = (V_N'', \Sigma, P_2, S)$

(Restriction of no. of Variable on R.H.S)

(i) $B \to b$, $D \to d$, $C_a \to a$, $A \to C_aB$, $C_b \to b$ are Included in P_2

Now

$S \to C_aAD$ We Introduce $S \to C_aC_1$

where $C_1 \to AD$ in P_2

$A \to C_bAB$ we Introduce $A \to C_bC_2$

$C_2 \to AB$ in P_2

$V_N'' = \{S, A, B, D, C_a, C_b, C_1, C_2\}$

$P_2 = \{B \to b, D \to d, C_a \to a, A \to C_aB, C_b \to b, S \to C_aC_1, C_1 \to AD, A \to C_bC_2, C_2 \to AB\}$

Now this G_2 is in CNF.

Question Reduce in to CNF

$S \to \sim S / [S \supset S] / p / q$

(S being only variable)

Negation as terminal

$S \to \sim S$

$S \to [S \supset S]$

$S \to p$

$S \to q$

Sol **Step 1** eliminate NULL & Unit prodn.

→ No One is present.

Step 2 Define

$G_1 = (V_N', \Sigma, P_1, S)$

(i) $S \to p$ & $S \to q$ are Included in P_1

(ii) $S \to \sim S$, $C_\sim \to \sim$ in P_1

Now $S \to C_\sim S$ in P_1

$S \rightarrow [S \supset S]$ Here $C_{[} \rightarrow [$ in P_1

then $C_{\supset} \rightarrow \supset$ in P_1 then $C_{]} \rightarrow]$ in P_1

$S \rightarrow C_{[} S C_{\supset} S C_{]}$ in P_1

$P_1 = \{ S \rightarrow p, S \rightarrow q, C_{\sim} \rightarrow \sim, C_{[} \rightarrow [, C_{\supset} \rightarrow \supset, C_{]} \rightarrow], S \rightarrow C_{[} S C_{\supset} S C_{]} \}$

$V_N' = \{ S, C_{\sim}, C_{[}, C_{]}, C_{\supset} \}$

Step 3

$S \rightarrow C_{[} \underline{S C_{\supset} S C_{]}}$

↳ $S \rightarrow C_{[} C_1$, $C_1 \rightarrow \underline{S C_{\supset} S C_{]}}$

$C_1 \rightarrow S C_2$ where $C_2 \rightarrow \underline{C_{\supset} S C_{]}}$

$C_2 \rightarrow C_{\supset} C_3$, $C_3 \rightarrow S C_{]}$

$P_2 = \{ S \rightarrow p, S \rightarrow q, C_{\sim} \rightarrow \sim, C_{[} \rightarrow [, C_{\supset} \rightarrow \supset,$
$C_{]} \rightarrow],$ $S \rightarrow C_{[} C_1, C_1 \rightarrow S C_2,$
$C_2 \rightarrow C_{\supset} C_3, C_3 \rightarrow S C_{]} \}$

$V_N'' = \{ S, C_{\sim}, C_{[}, C_{]}, C_{\supset}, C_1, C_2, C_3 \}$

* Do GNF yourself

Date : 4/April/18

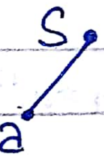
* Pumping lemma for CFL

lemma : if G be a CFG in CNF and T be a derivation tree in G if the length of the longest path in T is less than equal to K, then the yield of T is of length less than or equal to 2^{K-1}

Proof We proove this result by PMI

Suppose the length of the longest path in T is 1

Step base



yield of T = a

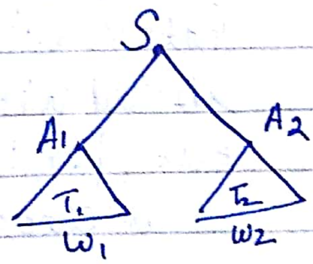
$|\text{yield}| = 1 = 2^{1-1} = 2^0 = 1$

Result is True for $K=1$

Induction Step

Suppose the result is true for $(K-1)$ $(K>1)$ i.e length of the longest path $= K-1$, $|\text{yield}| = 2^{K-1-1} = 2^{K-2}$

Suppose the length of the longest path in T be K



$|w_1| = 2^{K-2}$

$|w_2| = 2^{K-2}$

$w \to$ is yield of T

$w = w_1 w_2$

$|w| = |w_1| + |w_2| = 2^{K-2} + 2^{K-2} = 2 \cdot 2^{K-2} = 2^{K-1}$

So Result is true for (K), (K>1)

→ let L be a CFL. Then we can find a natural no. such that ① every $z \in L$ with $|z| \geq n$ can be written as $uvwxy$ for some strings u, v, w, x, y

② $|vx| \geq 1$

③ $|vwx| \leq n$

④ $uv^k w x^k y \in L \;\forall\, k \geq 0$

Proof Suppose $G = (V_N, \Sigma, P, S)$ be a CFG in CNF generating the CFL L

let $|V_N| = m$

let $n = 2^m$ → We will prove that $n = 2^m$ is the required natural no.

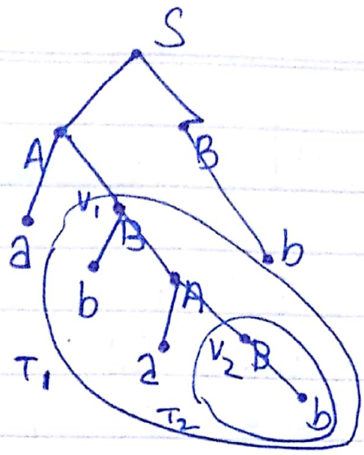
→ Suppose $z \in L$, $|z| \geq 2^m$ construct the Derivn tree of z say it is 'T'

→ Therefore the length of the longest path in T (say Γ) is greater than or equal to $(m+1)$

no. of vertices in Γ is $(m+2)$ and only last vertex will be a leaf and all other $(m+1)$ vertices are labelled with Variables

But $|V_N| = m$ therefore atleast one label is repeated So we choose the repeated label as follows

Start with the leaf of Γ and Travel along Γ upwards Now we stop when some label, say B is repeated (Among several repeated labels, B is the first) let V_1 & V_2 be the vertices with label B, V_1 being nearer to the root, Now the portion of the path from V_1 to the leaf has only one label, namely B, which is repeated



Γ: $S \to A \to B \to A \to B \to b$

T

$z = ababb$

Let T_1 & T_2 be the Subtree with v_1 & v_2 as root and z_1, w as yields Respectively.

T_1

$z_1 = bab$

$w = b$

$z = u z_1 y$

$z_1 = v w x$

→ So $z = uvwxy$

→ $|vx| \geq 1$, $|vwx| \leq n$ (obviously)

→

$S \overset{*}{\Rightarrow} uBy$, $B \overset{*}{\Rightarrow} vBx$, $B \overset{*}{\Rightarrow} w$

$S \overset{*}{\Rightarrow} uBy \underset{B \overset{*}{\Rightarrow} w}{\overset{*}{\Rightarrow}} uwy = uv^0wx^0y \in L \quad \forall K=0$

Now

$S \overset{*}{\Rightarrow} uBy \overset{*}{\Rightarrow} uvBxy \overset{*}{\Rightarrow} uvwxy \in L \quad K=1$

$\Downarrow *$

$uvvBxxy = uv^2Bx^2y \in L, \; K=2$

and so on.

→ This proves pumping lemma.

Scanned by CamScanner

Date 5/Apr/18

→ Show that $L = \{a^p \mid p \text{ is a prime}\}$ is not a CFL

Soln

Step 1 Suppose L is a CFL. let n be the natural no. Obtained by P. lemma let
$z \in L$, $z = a^p$, p is a prime greater than n

$z = uvwxy$

$uv^0wx^0y = uwy \in L$

$|uwy| = q$ where q is a prime

Suppose $|vx| = r$

$|uv^q wx^q y| = q + qr = q(1+r)$

= which is not a prime

$uv^q wx^q y \notin L$

∴ L is not a Context free language.

Decision Algorithm for CFL

1) Algorithm for deciding whether a CFL is empty.

2) " " " " is finite or non finite

1) Algorithm.

We can apply the construction of the theorm in which we find the equivalent grammer s.t. every Variable in which derives some terminals string for getting $V_N' = W_k$

L is non Empty iff $S \in W_k$

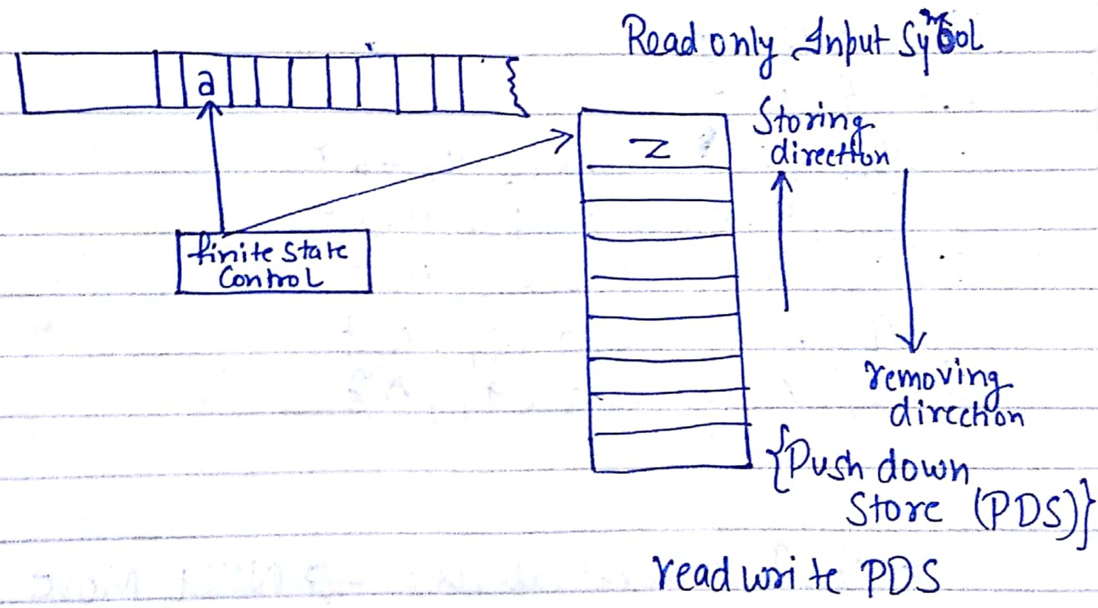
2) Algorithm

We construct a reduced CFG in CNF generating L. Now we draw a directed graph whose vertices are variable in G and if $A \to BC$ is a production, there are directed edges from A to B and $A \to C$, L is finite iff the directed graph has no cycles.

Unit 5

Pushdown Automata (pda/PDA)

→



PDA (Def^n)

A pda consists of:

i) a finite non empty set of states denoted by Q

ii) " " " " " input symbol denoted by Σ,

(iii) " " " " " " Pushdown symbol denoted by Γ

(iv) q_0 is the initial State
(v) initial Symbol of PDS denoted by Z_0
(vi) a set of final States $F \subseteq Q$
(vii) a Transition function δ from $Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma$ to the set of finite subset of $(Q \times \Gamma^*)$

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

$Q = \{q_0\, q_1\, q_f\}$
$\Sigma = \{a, b\}$,
$\Gamma = \{a, Z_0\}$
$F = \{q_f\}$
$\delta(q_0, a, Z_0) = \{(q_0, a Z_0)\}$
$\delta(q_0, a, a) = \{(q_0, aa)\}$
$\delta(q_0, b, a) = \{(q_1, \Lambda)\}$
$\delta(q_1, b, a) = \{(q_1, \Lambda)\}$
$\delta(q_1, \Lambda, Z_0) = \{(q_1, \Lambda)\}$

$\rightarrow \delta(q, \Lambda, Z) = (q', \alpha) \Rightarrow$ NULL Move
if Z is empty machine will hault.

<u>Defn</u>

(Instantaneous description) [ID]
let $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a pda. An ID is (q, x, α), where $q \in Q$, $x \in \Sigma^*$, $\alpha \in \Gamma^*$

$(q, a_1 a_2 \ldots a_n, Z_1 Z_2 \ldots Z_m)$

Defn Let A be pda. A move relation, denoted by $\vdash$, between ID's is defined as

$$(q, a_1a_2 \ldots a_n, z_1z_2 \ldots z_m) \vdash (q', a_2a_3 \ldots a_n, \beta z_2 \ldots z_m)$$

iff $\delta(q, a_1, z_1)$ contains (q', β)

$$(q, x, \alpha) \vdash^{n} (q', y, \beta)$$

$$(q, x, \alpha) \vdash (q_1, x_1, \alpha_1) \vdash (q_2, x_2, \alpha_2) \vdash \ldots \vdash (q', y, \beta)$$

$(q_0, x, Z_0) \longrightarrow$ Initial ID

if $(q_1, x, \alpha) \vdash^{*} (q_2, \Lambda, \beta)$ then $\forall\, y \in \Sigma^*$
$(q_1, xy, \alpha) \vdash^{*} (q_2, y, \beta)$

$$x = a_1 a_2 \ldots a_n$$

$$(q_1, a_1a_2 \ldots a_n, z_1z_2 \ldots z_m) \vdash (q_2, a_2 \ldots a_n, \beta' z_2 \ldots z_m) \vdash \ldots \vdash (q, \Lambda, \beta)$$

$$(q_1, a_1a_2 \ldots a_nb_1b_2 \ldots b_k, z_1z_2 \ldots z_m) \vdash (q, b_1b_2 \ldots b_k, \beta)$$

$$(q_1, xy, \alpha) \vdash^{*} (q, y, \beta)$$

Date: 9/Apr/18

Ex. 1

$A = \{\{q_0, q_1, q_f\}, \{a, b, c\}, \{a, b, Z_0\}, \delta, q_0, Z_0, q_f)$

where

$\delta(q_0, a, Z_0) = \{(q_0, aZ_0)\}$, $\delta(q_0, b, Z_0) = \{(q_0, bZ_0)\}$

$\delta(q_0, a, a) = \{(q_0, aa)\}$, $\delta(q_0, b, a) = \{(q_0, ba)\}$

$\delta(q_0, a, b) = \{(q_0, ab)\}$, $\delta(q_0, b, b) = \{(q_0, bb)\}$

$\delta(q_0, c, a) = \{(q_1, a)\}$, $\delta(q_0, c, b) = \{(q_1, b)\}$

, $\delta(q_0, c, Z_0) = \{(q_1, Z_0)\}$

$\delta(q_1, a, a) = \delta(q_1, b, b) = \{(q_1, \Lambda)\}$

$\delta(q_1, \Lambda, Z_0) = \{(q_f, Z_0)\}$

$w = bacab$

$(q_0, w, Z_0) = (q_0, bacab, Z_0) \vdash (q_0, acab, bZ_0)$

$\vdash (q_0, cab, abZ_0) \vdash (q_1, ab, abZ_0)$

$\vdash (q_1, b, bZ_0) \vdash (q_1, \Lambda, Z_0) \vdash (q_f, \Lambda, Z_0)$

hence w is accepted by PDA

take $w' = abcbb$

$(q_0, w, Z_0) = (q_0, abcbb, Z_0) \vdash (q_0, bcbb, aZ_0) \vdash$

$(q_0, cbb, baZ_0) \vdash (q_1, bb, baZ_0) \vdash (q_1, b, aZ_0)$

We cannot move further Stop here

$(q_0, abcbb, Z_0) \overset{*}{\vdash} (q_1, b, aZ_0)$

machine doesnot reach to final state.

Acceptance by PDA

1) Acceptance by final State
2> Acceptance by null State

1) Suppose A is any PDA
$(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be any Pda
then
$$T(A) = \{ w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q_f, \Lambda, \alpha),$$
$$\text{for } \alpha \in \Gamma^*, q_f \in F \}$$

2> $N(A) = \{ w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \Lambda, \Lambda)$ for
Some $q \in Q \}$

Suppose $L = \{ w c w^T \mid w \in \{a, b\}^* \}$
Show that $\boxed{L = T(A)}$
Proof :- $L \subseteq T(A)$ and $T(A) \subseteq L$ or $L^c \subseteq [T(A)]^c$
let $x \in L$ i.e $x = w c w^T$
$(q_0, w c w^T, Z_0) \vdash (q_0, \ldots c w^T, b Z_0) \vdash$ ---
$\vdash - (q_0, c w^T, w^T Z_0) \vdash (q_1, w^T, w^T Z_0)$
$\vdash (q_f, \Lambda, Z_0)$

$x \in L$ then $L \subseteq T(A)$

Now Show $L^c \subseteq [T(A)]^c$

let $x \in L^c \Rightarrow x \notin L$
Case (i) x does not Contain the Symbol c
then machine won't reach to final state
So $x \notin T(A)$
$\Rightarrow x \in [T(A)]^c \rightarrow L^c \subseteq [T(A)]^c$

Case (ii) let

$x = w_1 c w_2 , \quad w_2 \neq w_1^T$

$(q_0, w_1 c w_2, Z_0) \vdash^* (q_0, c w_2, w_1^T Z_0) \vdash^*$
$(q_1, w_2, w_1^T Z_0) \rightarrow$ machine haults
since $w_2 \neq w^T$

$x \notin T(A)$
i.e $x \in [T(A)]^c$

$L^c \subseteq [T(A)]^c$

Proved

Now suppose
we add one more $\delta(q_f, \Lambda, Z_0) = \{(q_f, \Lambda)\}$
to Ex ①

$(q_0, wcw^T, Z_0) \vdash^* (q_f, \Lambda, Z_0) \vdash (q_f, \Lambda, \Lambda)$
$wcw^T \in N(A)$ is accepted by NULL Store.

by Similar Step we can prove
$L = N(A)$

Date 11/Apr/18

Theorem →Imp

let $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a pda accepting L by null store,
we can find a pda $B = (Q', \Sigma, \Gamma', \delta_B, q_0', Z_0', F')$
which accepts L by final state i.e $L = N(A) = T(B)$

Pf

let us define B as follows
q_0' is a new state (not in Q) which is the

Initial state of B

$F' = \{q_f\}$ with q_f is a new state (not in Q)

$Q' = Q \cup \{q_0', q_f\}$

q_0' is a new pds in B which is initial pushdown symbol in B

$\Gamma' = \Gamma \cup \{Z_0'\}$, we define δ_B by the following rules.

R_1 : $\delta_B(q_0', \Lambda, Z_0') = \{(q_0, Z_0 Z_0')\}$

R_2 : $\delta_B(q, a, Z) = \delta(q_0, a, Z)\ \forall (q, a, Z) \in Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma$

R_3 : $\delta_B(q, \Lambda, Z_0') = \{(q_f, \Lambda)\}\ \forall q \in Q$

To Show that

$$N(A) = T(B)$$

(I) $N(A) \subseteq T(B)$

Let $w \in N(A)$ i.e. $(q_0, w, Z_0) \overset{*}{\vdash} (q, \Lambda, \Lambda)$ for some $q \in Q$ (given)

We have to show that $w \in T(B)$ i.e $(q_0', w, Z_0') \overset{*}{\vdash} (q_f, \Lambda, \alpha)$, $\alpha \in \Gamma'^*$

$(q_0', w, Z_0') \overset{R_1}{\vdash} (q_0, w, Z_0 Z_0') \vdash$
$\vdash (q_f, \Lambda, \Lambda)$ i.e $w \in T(B)$

$(q_0', \Lambda, Z_0') \underset{B}{\overset{R_1}{\vdash}} (q_0, \Lambda, Z_0 Z_0')$

$(q_0', w, Z_0') \underset{B}{\vdash} (q_0, w, Z_0 Z_0')$

$\vdash (q, \Lambda, \Lambda Z_0') \vdash (q, \Lambda, Z_0') \vdash$

(q_f, Λ, Λ)

Since $\therefore (q_0, xy, Z_0) \overset{*}{\vdash} (q, \Lambda y, \alpha)$

$\therefore w \in T(B)$

$\boxed{\text{i.e } N(A) \subseteq T(B)}$

(II) $T(B) \subseteq N(A)$

let $w \in T(B)$ i.e $(q_0', w, z_0') \vdash^{*} (q_f, \Lambda, \alpha)$,

$\alpha \in \Gamma'^{*}$

To show that

$w \in N(A)$ i.e. $(q_0, w, z_0) \vdash^{*} (q, \Lambda, \Lambda)$

for some $q \in Q$

Example

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

$Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$

$\Gamma = \{a, z_0\}$

Show that $N(A) = \{a^n b^n \mid n \geq 1\}$ also

Construct B s.t $T(B) = N(A)$ δ is defined by

$R_1 : \delta(q_0, a, z_0) = \{(q_0, az_0)\}$

$R_2 : \delta(q_0, a, a) = \{(q_0, aa)\}$

$R_3 : \delta(q_0, b, a) = \{(q_1, \Lambda)\}$

$R_4 : \delta(q_1, b, a) = \{(q_1, \Lambda)\}$

$R_5 : (q_1, \Lambda, z_0) = \{(q_1, \Lambda)\}$

(i) find $N(A)$

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

(I) To show that

$L \subseteq N(A)$

let $w \in L$, $w = a^n b^n$

$(q_0, w, z_0) = (q_0, a^n b^n, z_0) \vdash^{R_1} (q_0, a^{n-1} b^n, az_0)$

$\vdash^{*} (q_0, b^n, a^n z_0) \vdash (q_1, b^{n-1}, a^{n-1} z_0)$

$\vdash^{*} (q_1, \Lambda, z_0) \vdash (q_1, \Lambda, \Lambda)$

$\therefore w \in N(A)$ $\quad q_1 \in Q$

II) $N(A) \subseteq L$

let $w \in N(A)$ i.e. $(q_0, w, z_0) \vdash^{*} (q, \wedge, \wedge)$
for some $q \in Q$

(q_0, w, z_0) where $w = ax\ldots\ldots$

Choice ①

$(q_0, ax\ldots, z_0) \vdash (q_0, x\ldots, az_0)$
$= (q_0, by\ldots, az_0) \vdash (q_1, y\ldots, z_0)$

y: a or b

↳ machine is haulting when $y = a$ or b

Choice ②

$(q_0, ax\ldots, z_0) \vdash (q_0, x\ldots, az_0) =$
$(q_0, ay\ldots, az_0) \vdash (q_0, \overset{b}{y}\ldots, aaz_0) \vdash$
$(q_1, \underset{b}{y_1}, az_0) \vdash (q_1, y_2, z_0)$

as the machine encounter b it delete a from pds

Utimately we have z_0 in pds.

so

$N(A) \subseteq L$

(iii) construct B s.t $T(B) = N(A)$

$B = (Q', \Sigma, \Gamma', \delta_B, q_0', z_0', F = \{q_f\})$

q_0' is a new state (not in A) is the initial state of B

$Q' = Q \cup \{q_0', q_f\}$

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

Rules

① $\delta_B (q_0', \wedge, z_0') = \{(q_0, z_0 z_0')\}$

② $\delta_B (q, a, z) = \delta (q, a, z) \; \forall \; (q, a, z) \in (Q \times \Sigma \cup \{\wedge\} \times \Gamma)$

③ $\delta_B (q, \wedge, z_0') = \{(q_f, \wedge)\} \; \forall q \in Q$

hence

$\delta_B(q_0', \Lambda, z_0') = \{(q_0, z_0 z_0')\}$ from Rule ①

$\delta_B(q_0, a, z_0) = \{(q_0, a z_0)\}$

$\delta_B(q_0, a, a) = \{(q_0, aa)\}$ " Rule ②

$\delta_B(q_0, b, a) = \{(q_1, \Lambda)\}$

$\delta_B(q_1, b, a) = \{(q_1, \Lambda)\}$

$\delta_B(q_1, \Lambda, z_0) = \{(q_1, \Lambda)\}$

$\delta_B(q_0, \Lambda, z_0') = \{(q_F, \Lambda)\}$ " Rule ③

$\delta_B(q_1, \Lambda, z_0') = \{(q_f, \Lambda)\}$

Date 12/Apr/18

Imp

Theorem: if $A = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ accepts L by final State, we can find a pda B accepting L by null State

i.e $L = T(A) = N(B)$

Pf: let us consider B as follows

q_0' is a new state (not in Q) known as initial state of B

d is a new state (not in Q) — dead state

z_0' is a new initial PDS in B

$Q' = Q \cup \{q_0', \alpha\}$

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

δ_B is defined as follows

R_1: $\delta_B(q_0', \Lambda, z_0') = \{(q_0, z_0 z_0')\}$

R_2: $\delta_B(q, a, z) = \delta(q, a, z)\ \forall q \in Q,\ a \in \Sigma,\ z \in \Gamma$

R_3: $\delta_B(q, \Lambda, z) = \delta(q, \Lambda, z) \cup \{(d, \Lambda, z) \cup \{(d, \Lambda)\}\ \forall z \in \Gamma',\ \boxed{q \in F}$

R_4: $\delta_B(d, \Lambda, z) = \{(d, \Lambda)\}\ \forall z \in \Gamma'$

Scanned by CamScanner

$T(A) \subseteq N(B)$

Let $w \in T(A)$ i.e $(q_0, w, Z_0) \vdash^{*} (q, \Lambda, \alpha), q \in F, \alpha \in \Gamma^{*}$

To show that

$w \in N(B)$ i.e $(q_0', w, Z_0') \vdash^{*} (q, \Lambda, \Lambda), q \in Q$

$(q_0', w, Z_0') \vdash^{B} (q_0, w, Z_0 Z_0') \vdash (q, \Lambda, \alpha Z_0')$ $(\because w \in T(A))$

$\vdash (d, \Lambda, Z_0') \vdash (d, \Lambda, \Lambda)$

$w \in N(B)$

→ In the same manner we can show

$N(B) \subseteq T(A)$

Theorem: if L is a CFL then we can find a pda A accepting L by empty store

i.e $L = N(A)$

Sol Pf

L is a CFL

Suppose $G = (V_N, \Sigma, P, S)$ is a CFG s.t.

$L = L(G)$

Construct A as follows

$A = (\{q\}, \Sigma, \Gamma = V_N \cup \Sigma, \delta, q, S, \phi)$

δ is defined by

$R_1 : \delta(q, \Lambda, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$

$R_2 : \delta(q, a, a) = \{(q, \Lambda)\} \forall a \in \Sigma$

Now

Show that $L = N(A)$

$q \in F$

first we'll prove that $L(G) \subseteq N(A)$

Before proving this

we prove One result

if $S \overset{*}{\Rightarrow} UA\alpha$ by left most derivation
then $(q, uv, S) \overset{*}{\vdash} (q, v, A\alpha)$ $\forall\, v \in \Sigma^*$
$U \in \Sigma^*$
$A \in V_N, \alpha \in (V_N \cup \Sigma)^*$

let $S \overset{*}{\Rightarrow} UA\alpha$, $u = \Lambda, A = S, \alpha = \Lambda$ . .

$(q, uv, S) \vdash (q, v, S)$ $A \in V_N$, $\alpha \in (V_N \cup \Sigma)^*$

If
$S \overset{1}{\Rightarrow} UA\alpha$, $(q, uv, S) \overset{R_1}{\vdash} (q, uv, UA\alpha)$
$\overset{R_2}{\vdash} (q, v, A\alpha)$

<u>Induction step</u> : Suppose the result is true for n steps
Suppose $S \overset{n+1}{\Rightarrow} UA\alpha$
$S \overset{n}{\Rightarrow} U_1 A_1 \alpha_1 \Rightarrow UA\alpha$, when $A_1 \Rightarrow U_2 A \alpha_2$ in P
$\downarrow$
$u_1 u_2 A \alpha_2 \alpha_1$ $\quad$ $u = u_1 u_2$, $\alpha = \alpha_2 \alpha_1$

$(q, u_1 u_2 v, S) \overset{*}{\vdash} (q, u_2 v, A_1 \alpha_1)$ ——— (1)

By R_1, $(q, \Lambda, A_1) \vdash (q, \Lambda, u_2 A \alpha_2)$

$(q, u_2 v, A_1) \vdash (q, u_2 v, u_2 A \alpha_2)$
$(q, u_2 v, A_1 \alpha_1) \vdash (q, u_2 v, u_2^{A} \alpha_2 \alpha_1)$

$(1) \Rightarrow (q, uv, S) \vdash (q, u_2 v, A_1 \alpha_1) \vdash (q, u_2 v, u_2 A \alpha_2 \alpha_1)$
$\vdash (q, v, A \alpha_2 \alpha_1) = (q, v, A\alpha)$

$\rightarrow$ So Result is true for $n+1$ also

let $w \in L(G)$ i.e $S \overset{*}{\Rightarrow} w$

$S \overset{*}{\Rightarrow} uAv \Rightarrow uu'v = w \quad (A \to u' \text{ in } P)$

$(q, w, S) = (q, uu'v, S) \vdash (q, u'v, Av)$
$\vdash (q, u'v, u'v) \vdash (q, v, v) \vdash (q, \Lambda, \Lambda)$

$\boxed{\therefore \; w \in N(A)}$

$\boxed{\text{thus } L(G) \subseteq N(A)}$

→ Now To show that $N(A) \subseteq L(G)$

If $(q, uv, S) \vdash^{*} (q, v, \alpha)$ then $S \overset{*}{\Rightarrow} u\alpha$

by PMI
base: $(q, uv, S) \vdash^{0} (q, v, \alpha)$, $u = \Lambda$, $S = \alpha$, $S \overset{*}{\Rightarrow} u\alpha$

→ Suppose the result is true for n steps.

→ let $(q, uv, S) \vdash^{n+1} (q, v, \alpha)$

(i) $(q, uv, S) \vdash^{n} (q, v, A\alpha_2) \vdash (q, v, \alpha_1\alpha_2)$
$= (q, v, \alpha)$

By Induction hypothesis, $S \overset{*}{\Rightarrow} uA\alpha_2 \Rightarrow u\alpha_1\alpha_2 = u\alpha$

(ii) $(q, uv, S) \vdash^{n} (q, av, a\alpha) \vdash (q, v, \alpha)$

$u = u'a$ for some $u' \in \Sigma^*$

$uv = \underbrace{\ldots\ldots\ldots}_{v'} av$

$(q, u'av, S) \vdash^{n} (q, av, a\alpha)$

$S \overset{*}{\Rightarrow} u'a\alpha = u\alpha$

let $w \in N(A)$ i.e $(q, w, S) \vdash^{*} (q, \Lambda, \Lambda)$

$(q, w\Lambda, S) \vdash^{*} (q, \Lambda, \Lambda)$

$S \overset{*}{\Rightarrow} w\Lambda$

$S \overset{*}{\Rightarrow} w$

$\boxed{\begin{array}{l} \therefore \; w \in L(G) \\ N(A) \subseteq L(G) \end{array}}$

Last theorem left

Exercise

Construct a pda 'A' equivalent to the following CFG
$S \to 0BB$, $B \to 0S \mid 1S \mid 0$
Test whether 010^4 is in $N(A)$

$\Rightarrow$ $A = (\underbrace{\{q\}}_{Q}, \underbrace{\{0,1\}}_{\Sigma}, \underbrace{\{S, B, 0, 1\}}_{\Gamma}, \delta, q, S, \phi)$

$\delta(q, \Lambda, A) = \{(q, \alpha)$ if $A \to \alpha$ in $P\}$
$\delta(q, a, a) = \{(q, \Lambda)\} \forall a \in \Sigma^*$

Rule 1

$\Rightarrow$ R_1 $\delta(q, \Lambda, S) = \{(q, 0BB)\}$
R_2 $\delta(q, \Lambda, B) = \{(q, 0S), (q, 1S), (q, 0)\}$

Rule 2

R_3 $\delta(q, 0, 0) = \{(q, \Lambda)\}$
R_4 $\delta(q, 1, 1) = \{(q, \Lambda)\}$

Test 010^4 is in $N(A)$

$$(q, 010^4, S) \vdash (q, 010^4, 0BB)$$
$$\vdash (q, 10^4, BB) \vdash (q, 10^4, 1SB)$$
$$\vdash (q, 0^4, SB) \vdash (q, 0^4, 0BBB)$$
$$\vdash (q, 000, BBB) \vdash^* (q, \Lambda, \Lambda)$$

hence

$010^4 \in N(A)$