Unit - 2

12/2/2018

REGULAR EXPRESSION

* Notation to describe the Regular language.
* Algebraic expression.
* Notation involve with alphabet, parenthesis and operator
  $*, \bullet, +$

* E is Regular expression, $L(E)$
* <u>Definition</u>: Regular expression recursively as follows

<u>Basis</u>: ϕ is Regular expression, denoting language $\{\}$.

ie., $L(\phi) = \{\}$ or ϕ

* ϵ is Regular expression, denoting the Language $\{\epsilon\}$

ie., $L(\epsilon) = \{\epsilon\}$

* a is a Regular expression, $a \in \Sigma$, denoting the language

$L(a) = \{a\}$

<u>Induction</u>: If E and F are Regular Expression denoting language $L(E)$ and $L(F)$ respectively.

* E + F is a Regular expression, denoting union of $L(E)$ and $L(F)$.

ie., $L(E+F) = L(E) \cup L(F)$

* E.F is a RE, denoting concatenation of $L(E)$ followed by $L(F)$.

ie., $L(EF) = L(E)L(F)$

* E^* is a RE, denoting the closure of $L(E)$.

$L(E^*) = (L(E))^*$

Scanned by CamScanner

Priority	RE	L(RE)
① ( )	1) ϕ	$L(\phi) = \{\}$ or $\phi \to 0 \to$
② *	2) $L(E) = \{\epsilon\}$	$0 \to \{0\}$
③ .	3) $L(a) = (a)$	$\{a\} = ab$.
④ +		

$L(a+b) = L(a) + L(b)$ $\to \bigcirc \xrightarrow{a;b} \bigcirc$

$L(a+b) = L(a+) = L(a) L$

$L(ab) = \{a\}\{b\} = a \cdot r^{+}$ $= \{ab\}$ $\bigcirc \xrightarrow{a} \bigcirc \xrightarrow{b} \bigcirc$

$= \{a+b\}$

$L(a^*) = [(L(a) + L(b)) + 0]$ $\to a \to b$ Elements

$= [L(a)]^* = (a)^* = (a)^* = \{\epsilon, a, aa, \ldots\}$ $\to \bigcirc$ a

$= \to$ repeat

$(a+b)^* = L[(a+b)]^*$

Write a regular expression for the following statements

1) A set of a's and b's ending with ab, $\Rightarrow (a+b)^* ab$

2) set of string of a's and b's starts with ab. $\Rightarrow ab(a+b)^*$

3) set of remaing with language with substring ab.

$\Rightarrow (a+b)^* ab (a+b)^*$

Problems

Write Regular expression for the following languages.

1) $L = \{ |W| \bmod 3 = 0 \mid W \in \{a,b\}^* \}$

2) set of strings of a's and b's with no consecutive pair of a's.

3) set of strings of a's and b's with atleast one consecutive pair of a's.

4) set of strings of a's & b's with atmost one consecutive pair of a's.

5) set of strings of a's & b's with even length.

6) set of strings of a's & b's with odd length.

② $L = \{ a^n b^m \mid n \geq 3, m \leq 3 \}$

③ $L = \{ a^n b^{2m} \mid n, m \geq 1 \}$

$\cdot \rightarrow$ AND

$+ \rightarrow$ OR

①

1) $((a+b)\cdot(a+b)\cdot(a+b))^*$

2) ~~$(b^* a b^*)^*$~~ $(ab+b)^*(a+\epsilon)$

3) $(a+b)^* a a^* (a+b)^*$

4) ~~$(a+b)^* b a a^* b (b+a)^*$~~ 4) $(ab+b)^*(aa+\epsilon)(ba+b)^*$ $m \geq 2$

5) $((a+b)\cdot(a+b))^*$

6) $((a+b)\cdot(a+b))^* \cdot (a+b)$

③ $a \cdot a^* \cdot (b \cdot b)^* \cdot b \cdot b$

②

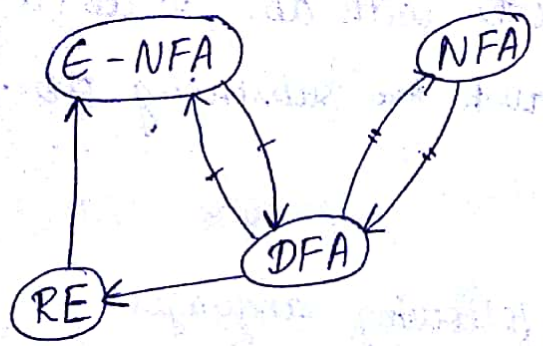
$a \cdot a \cdot a \cdot a^* \cdot (\epsilon + b + bb + bbb)$

Scanned by CamScanner

*→ Zero or more

+→ one or more

Equivalence b/n Regular Expression & Finite Automata



RE to ϵ-NFA

Theorem: Every language defined by Regular expression, is also defined by the Finite Automata.

Proof: Consider $L = L(R)$ for some Regular language R, then show $L = L(E)$ with some ϵ-NFA with following condition.

1) Only one accepting state
2) no arc into the initial state
3) no arc out of accepting state

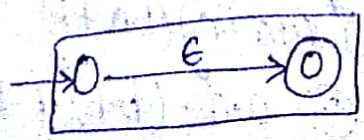
Proof is by using structural Induction of R.E. ie., definition of R.E.

Basis: * ϕ is a R.E, denoting the language $\{\ \}$

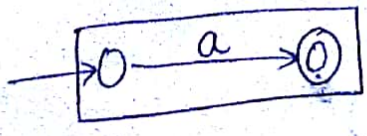
ie., $L(\phi) = \{\ \}$

* ϵ in R.E, denoting language $\{\epsilon\}$

ie., $L(E) = \{\epsilon\}$



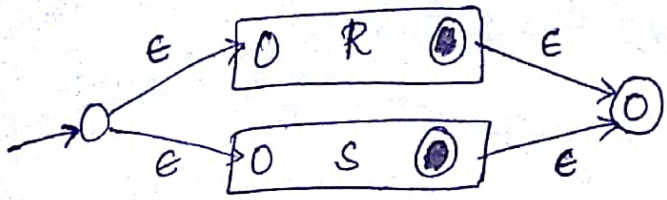
* 'a' is a R.E, $a \in \Sigma$ ie., $L(a) = \{a\}$



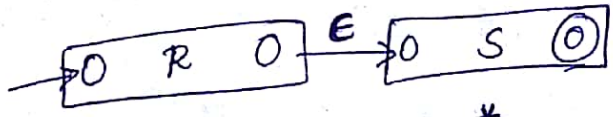
Scanned by CamScanner

Induction: If R and S are Regular Expression, denoting language L(R) and L(S)

* R+S is a R.E, denoting language L(R) using L(S).

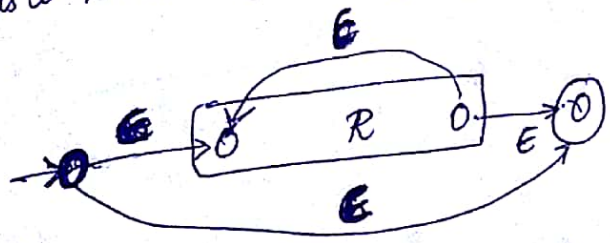


* R·S is a R.E, concatenation of L(R) and L(S)



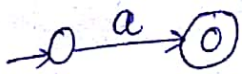
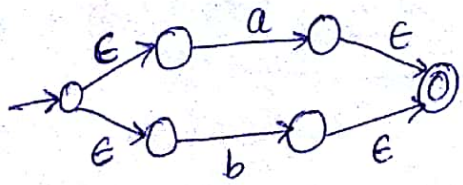
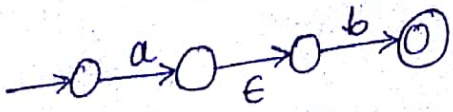
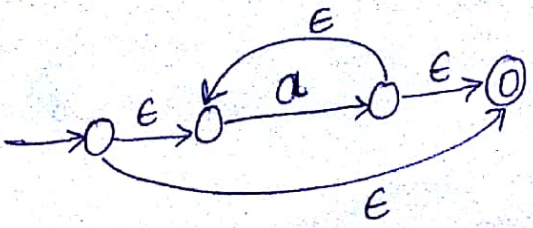
$L(R \cdot S) = L(R) \cdot L(S)$

* R^* is a R.E, $L(R^*) = (L(R))^*$



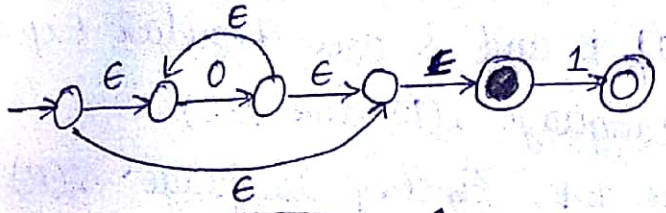
Hence, followed all assumptions & proof.

RE.

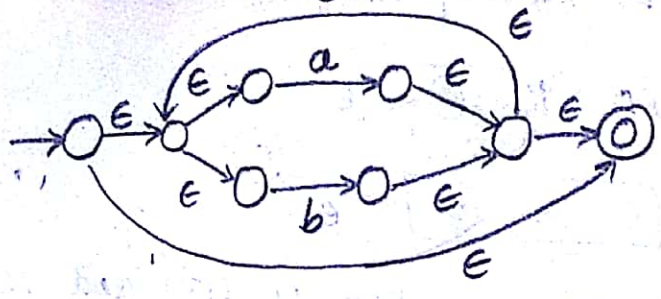
RE	ϵ-NFA
ϕ	
ϵ	
a	
$a+b$	
$a \cdot b$	
a^*	

Scanned by CamScanner

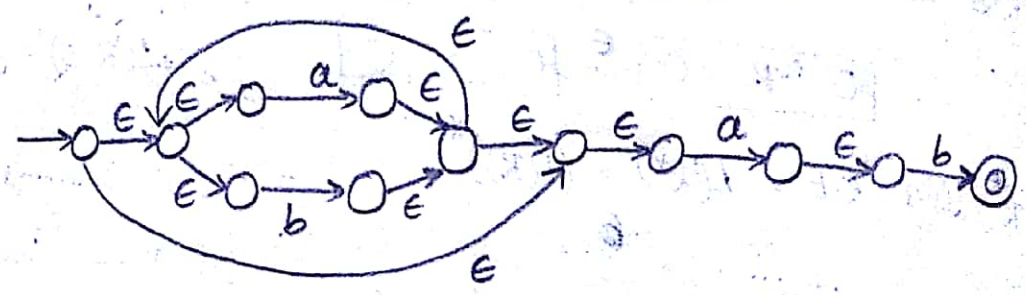
0^*1



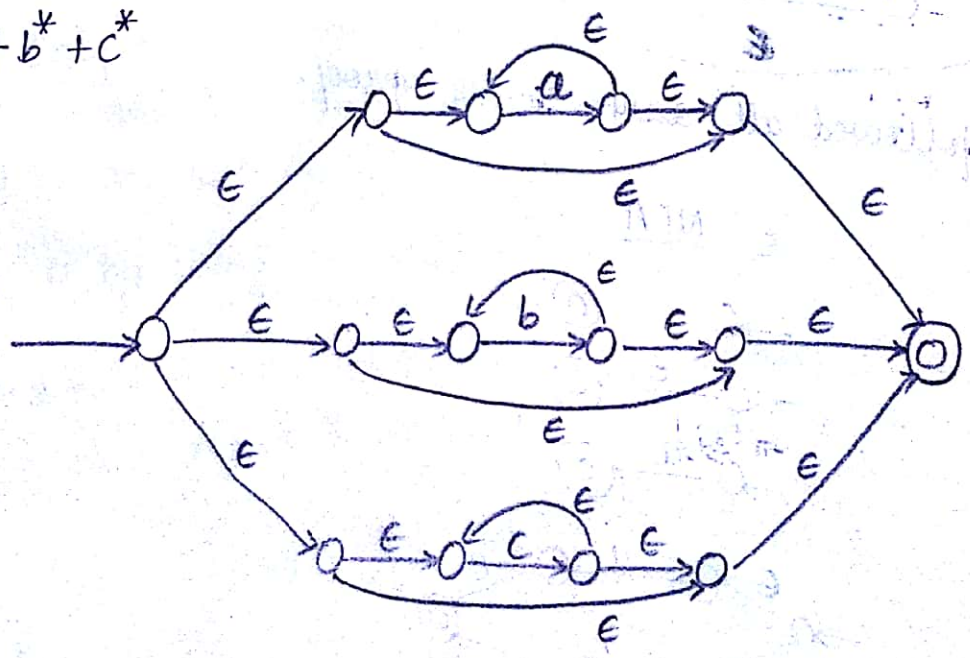
$(a+b)^*$



$(a+b)^*ab$



$a^*+b^*+c^*$



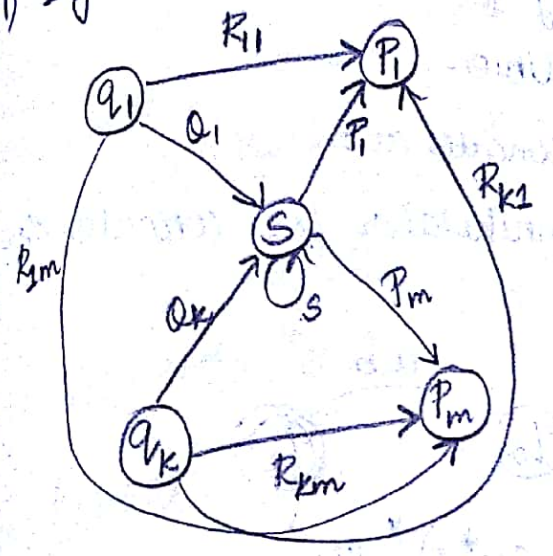
Scanned by CamScanner

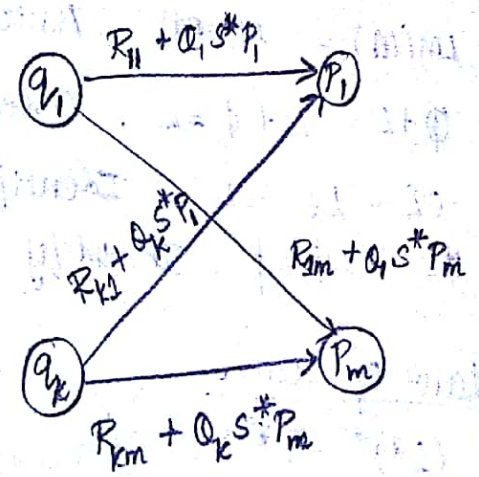
Converting DFA's to R.E

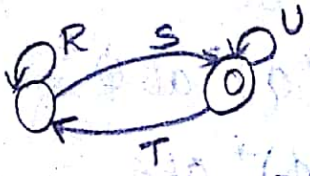
2 methods

1) By eliminating state
2) Kleen's Theorem

1) By eliminating state





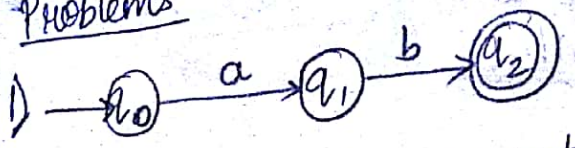
① 

$\rightarrow (R + SU^*T)^* SU^*$

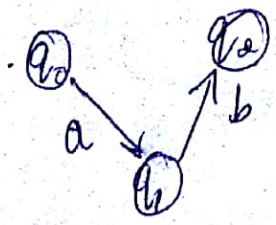
②

$\rightarrow R^*$ [Initial & final states are same]

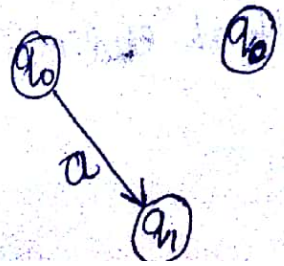
Problems

1) 

$S = \phi + a\phi^* b$ ~~$= \phi + \phi\phi^*\phi$~~

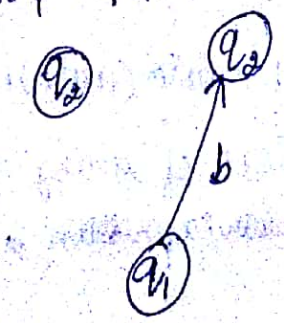


$R = \phi + a\phi^*\phi$



$T = \phi + \phi\phi^*\phi$

$U = \phi + \phi\phi^* b$



Scanned by CamScanner

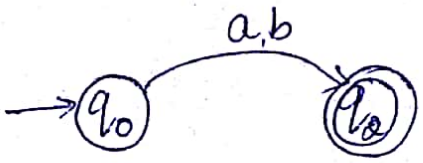
Laws of Algebra

Two languages L and M

1) $L+M = M+L$, Commutative law for $\cup$
2) $(L+M)+M = L+(M+M)$, Associative law for $\cup$
3) $LM(M) = L(MM)$, Associative only for concatanation
4) $\phi+L = L+\phi = L$, Indentify for Union
5) $\epsilon L = L\epsilon = L$, Identify for concatenation
6) $\phi L = L\phi = \phi$, Indentify and annihilator for concatenation.

Laws of Closure

1) $(L^*)^* = L^*$
2) $\phi^* = \epsilon$
3) $\epsilon^* = \epsilon$
4) $L^+ = LL^* = L^*L$
5) $L^* = L^+ + \epsilon$
6) $L^? = L+\epsilon$



$(R+SU^*T)^*SU^*$

$= (\phi + ab\phi^*\phi)^* ab\phi^*$

$= (\phi + ab\epsilon\phi)^* ab\epsilon$

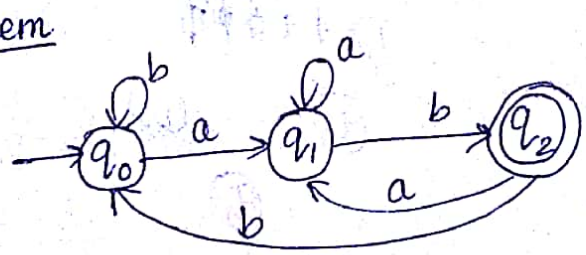
$= (\phi+\phi)^* ab\epsilon$

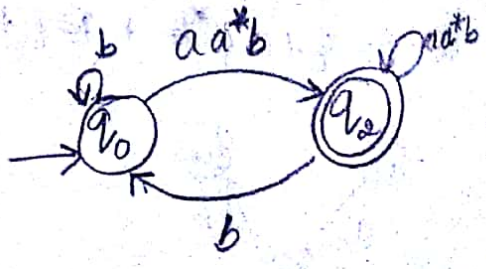
$= \epsilon ab\epsilon$

$= ab$

Problem

1)





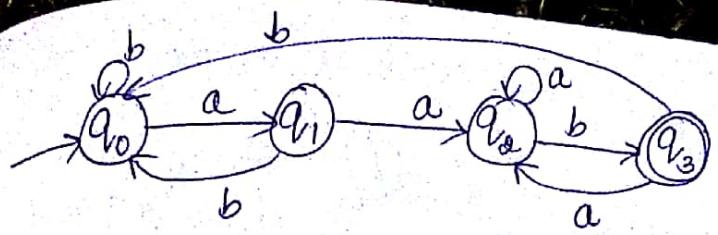
$(R+SU^*T)^*SU^*$

$(b+aa^*b(aa^*b)b)^* aa^*b(aa^*b)^* = \epsilon a\epsilon b\epsilon = ab$

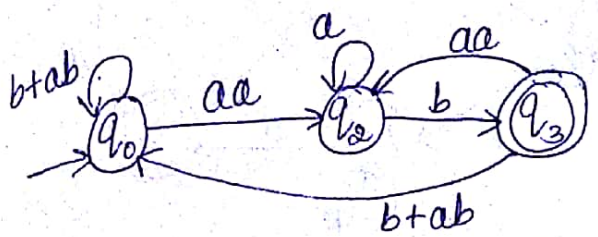
$L = \{$set of string of a's & b's ends with ab ending$\}$

regular expression $\rightarrow$ $R.E = (a+b)^*ab$

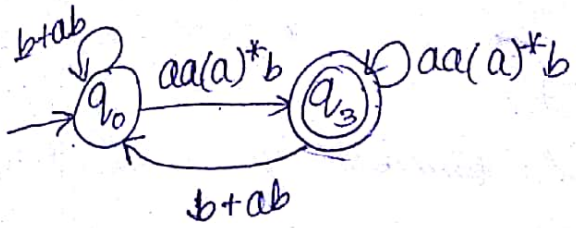
Scanned by CamScanner

2) 

, Eliminate q_1 for 1st figure

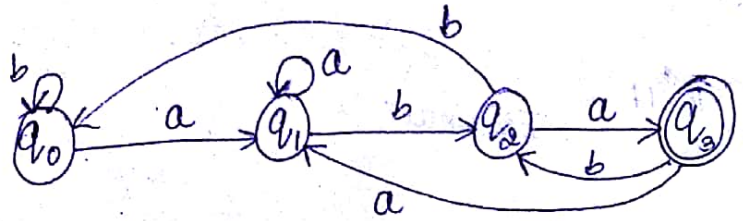


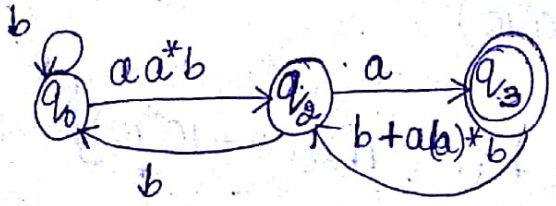
, Eliminate q_2 from 2nd figure

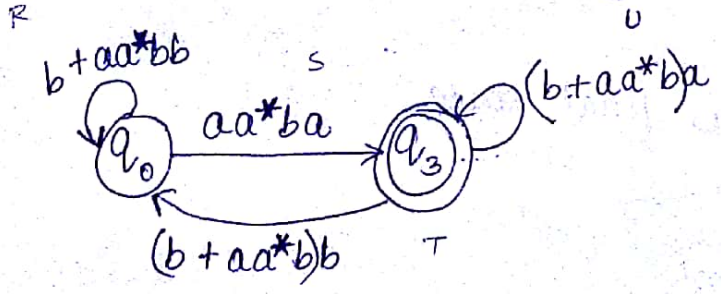


$(R+SU^*T)^*SU^*$

$[(b+ab) + aa(a)^*b\,(aa(a)^*b)^*(b+ab)]^*\,aa(a)^*b\,(aa(a)^*b)^*$

3) 





$(R+SU^*T)^*SU^*$

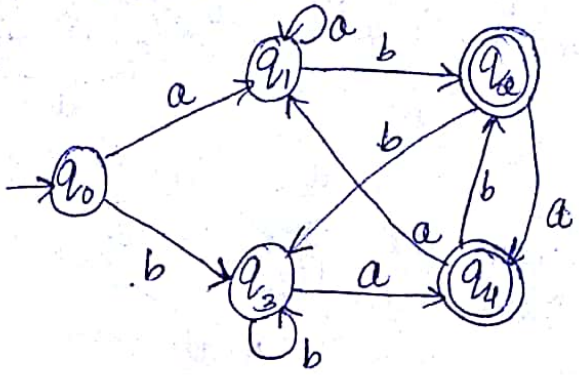
$[(b+aa^*bb) + (aa^*ba)[(b+aa^*b)a]^*[(b+aa^*b)b]]\,(aa^*ba)[(b+aa^*b)a]^*$

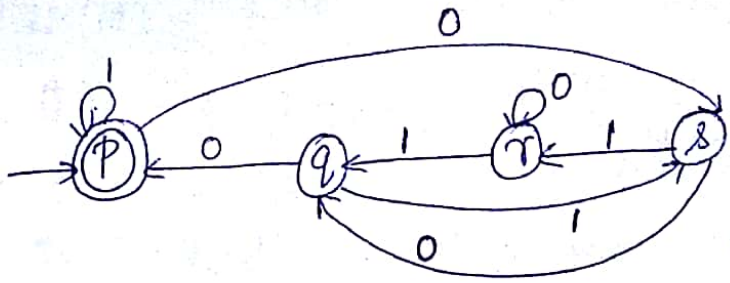
Scanned by CamScanner

4)

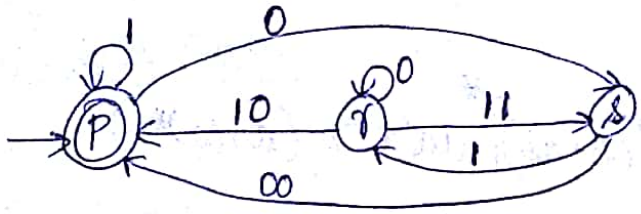
	0	1
$\rightarrow *p$	s	p
q	p	s
r	r	q
s	q	r

5) Ends with ab or ba.

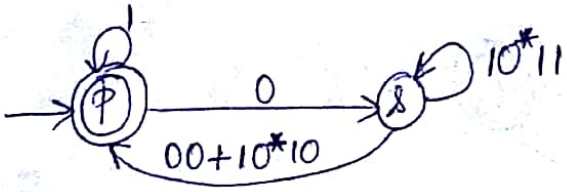




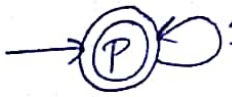
Eliminate 'q'



Eliminate 'r'



Eliminate 's'

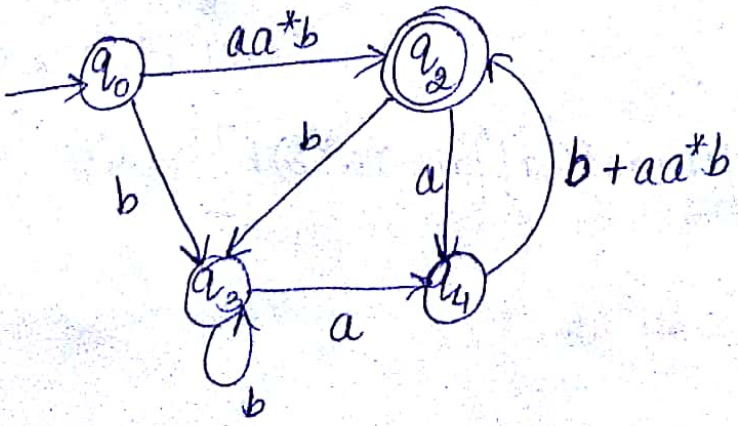


$1+(010^*11)(00+10^*10)$

$(R+SU^*T)^*SU^* = [1+(010^*11)(00+10^*10)]^*$

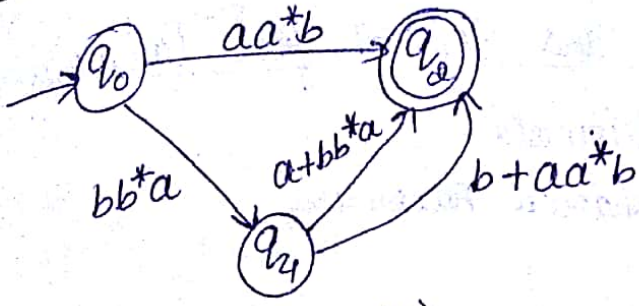
5) 1st consider q_2 as final state

Eliminate q_1.

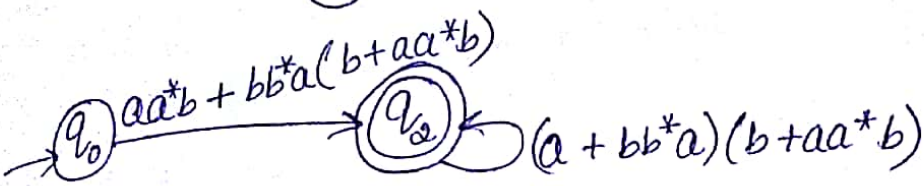


Eliminate q_3

Scanned by CamScanner



Eliminate q_4

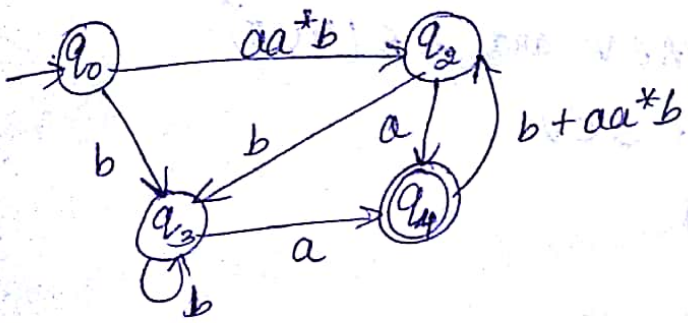


$(R+SU^*T)^*SU^*$

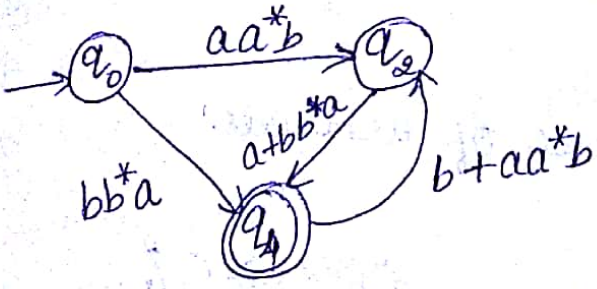
$$r_1 = [\phi + [aa^*b + bb^*a(b+aa^*b)][a + (bb^*a)(b+aa^*b)]^*\phi]^*$$
$$[aa^*b + (bb^*a)(b+aa^*b)][(a+bb^*a)(b+aa^*b)]^*$$

If q_4 is considered as final state,

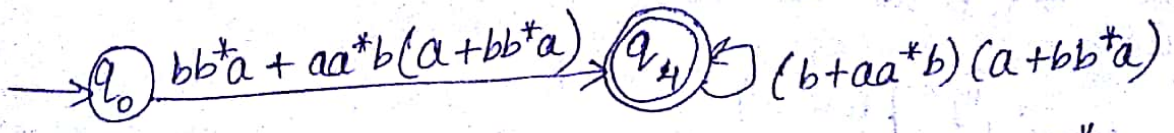
Eliminate q_1.



Eliminate q_3



Eliminate q_2



$$r_2 = [\phi + [bb^*a + aa^*b(a+bb^*a)][(b+aa^*b)(a+bb^*a)]^*\phi]^*$$
$$[bb^*a + aa^*b(a+bb^*a)][(b+aa^*b)(a+bb^*a)]^*$$

Scanned by CamScanner

Context free Grammar and Context free Language

Regular Language → Finite Automata
Context free Language → Pushdown Automata

Applications

* Parser
* XML , DFD

Context free Grammar

Defined with 4 tuples

$G = (V, T, P, S)$

where, V = finite set of variable / non-terminal
T = finite set of terminals
P = Set of rules or productives
$A \rightarrow \alpha$, where $A \in V$ and $\alpha \in (V \cup T)^*$
S = Start variable , $S \in V$.

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

where, $P = \{ S \rightarrow \epsilon$
$S \rightarrow aSb \}$

NOTE:
String can only be sequence of terminals.

$S \rightarrow \epsilon$
$S \rightarrow aSb$ $\Rightarrow \{ \epsilon, ab, aabb, aaabbb, --- a^n b^n \mid n \geq 0 \}$

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

* Write CFG for $L = \{ a^n b^n \mid n \geq 1 \}$

$S \rightarrow ab$
$S \rightarrow aSb$ $\{ ab, aabb, aaabbb, --- \}$

Scanned by CamScanner

2) $L = \{a^n b^{n+1} \mid n \geq 0\}$

$S \to b$
$S \to aSb$ $\{b, abb, aabbb, ----\}$

3) $L = \{a^{2n} \mid n \geq 0\}$

$S \to \epsilon$
$S \to aSa$ $\{aa, \epsilon, aaaa, ----\}$

4) $L = \{a^{2n+1} \mid n \geq 0\}$

$S \to a$
$S \to aSa$ $\{a, aaa, aaaaa, ----\}$

<u>Derivation</u>

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

$P = \{ S \to aAB \to ①$
$A \to bBb \to ②$
$B \to A / \epsilon$ ③ ④

<u>abbbb</u>

<u>Left most Derivation</u>

$S \underset{1}{\Rightarrow} a\underline{A}B \underset{2}{\Rightarrow} ab\underline{B}bB \underset{4}{\Rightarrow} ab\epsilon bB \Rightarrow abb\underline{B} \underset{3}{\Rightarrow} abb\underline{A} \underset{2}{\Rightarrow} abbb\underline{B}b \underset{4}{\Rightarrow}$
$abbb\epsilon b \underset{4}{\Rightarrow} abbbb$

<u>Derivation</u>: The process of obtaining a string from the start variable of G by applying one or more production is called Derivation.

'$\Rightarrow$' represent derivation process.

① Left most Derivation
② Right most Derivation

<u>For abbbb</u>

<u>Right most Derivation</u>

$S \underset{1}{\Rightarrow} aA\underline{B} \underset{3}{\Rightarrow} aA\underline{A} \underset{2}{\Rightarrow} aAb\underline{B}b \underset{4}{\Rightarrow} aAb\epsilon b \Rightarrow a\underline{A}bb \underset{2}{\Rightarrow} ab\overset{\epsilon}{\underline{B}}bbb \underset{4}{\Rightarrow} abbbb$

* Top down Parser → Left most derivation 'C'
* Bottom up Parser → Right most derivation 'C++'

Scanned by CamScanner

Left most Derivation

A derivation is said to be left most, if in each step, the left most variable in the sentencial form, is replaced by its production.

$$S \overset{*}{\underset{lm}{\Longrightarrow}} abbbb$$

Right most derivation

A derivation is said to be right most, if in each step, the right most variable in the sentencial form, is replaced by its production.

$$S \overset{*}{\underset{rm}{\Longrightarrow}} abbbb$$

Write CFG for the following

1) $L = \{ ww^R \mid w \in \{a,b\}^* \}$ $\qquad w^R \rightarrow$ Reverse of w.

2) $L = \{ 0^n w w^R 1^n \mid n \geq 1, w \in \{a,b\}^* \}$

3) $L = \{ |w| \bmod 3 = 0, w \in \{a,b\}^* \}$

4) $L = \{ a^m b^m c^n \mid m \geq 1, n \geq 0 \}$

Answers

1) $S \rightarrow \epsilon$ $\qquad \{\epsilon, aa, bb, abba, baab, \ldots\}$ $\qquad$ asa, aεa, aa

$S \rightarrow aSa$

$S \rightarrow bSb$ $\qquad S \rightarrow \epsilon / asa / bSb$

2) ~~$(S \rightarrow \epsilon / 0S1$~~ $\qquad \{01, 0aa1, 0bb1, 0abba1, 0baab1, \ldots\}$

~~$\rightarrow \epsilon / asa / bSb)$~~ $\qquad$ 0abba

$S \rightarrow 0S1$

$A \rightarrow \epsilon / 0A1 / B$

$B \rightarrow aAa / bAb$

Scanned by CamScanner

2) {ε, aaa, bbb, aab, aba, baa, bba, bab, abb, aaabbb, bbbaaa ---}

$S \to \epsilon / aaas / bbbs / aabs / abas / baas / bbas / babs / abbs$

$S \to \epsilon / aAb$
$A \to bSa$

baaba
babsba
bba

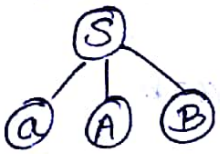
4) {ab, abc, aabbc, abcc, aabbcc, ----}

$S \to aAc$
$A \to bSb / B$
$B \to aSa / bSb / cSc$

Derivation tree

It is an ordered tree, in which nodes are labelled with left side of productions in which the children of a node represent its corresponding right side.

$S \to aAB$
$A \to bBb$
$B \to \epsilon / A$



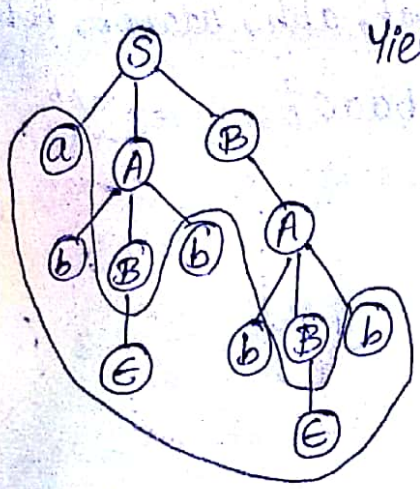
Formal definition

Let $G = (V, T, S, P)$ be a CFG, an ordered tree is a derivation tree with the following conditions.

1. The root is labelled with 'S'.
2. Every leaf has a label from $T \cup \{\epsilon\}$
3. Every interior node has a label from V (variable).
4. If we have a vertex labelled $A \in V$, and if children are labelled $a_1, a_2, \ldots a_n$, then P must contain a production of the form $A \to a_1 a_2 \ldots a_n$
5. The leaf labelled 'ε' has no siblings, ie., a vertex with a child labelled 'ε' can have no other children.

Scanned by CamScanner

Yield of tree = abbbb



Problems

1) $S \rightarrow AB$

$A \rightarrow aaA / \epsilon$

$B \rightarrow Bb / \epsilon$

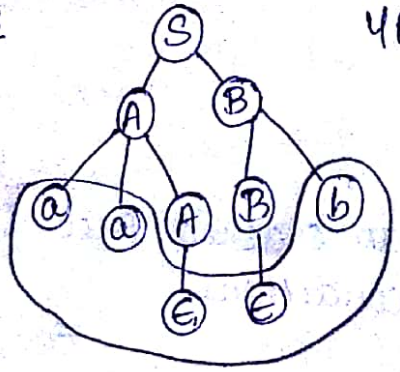
Write LMD, RMD, DT for input "aab"

LMD: $S \Rightarrow AB \Rightarrow aaAB \Rightarrow aa\epsilon B \Rightarrow aaB \Rightarrow aaBb \Rightarrow aa\epsilon b \Rightarrow aab$

RMD: $S \Rightarrow AB \Rightarrow ABb \Rightarrow A\epsilon b \Rightarrow Ab \Rightarrow aaAb \Rightarrow aa\epsilon b \Rightarrow aab$

DT:

Yield of tree = aab



2) $E \rightarrow +EE / *EE / -EE / x / y$. LMD, RMD, DT for i/p $+*-xyxy$

LMD: $E \Rightarrow +EE \Rightarrow +*EEE \Rightarrow +*-EEEE \Rightarrow +*-xEEE \Rightarrow +*-xyEE$
$\Rightarrow +*-xyxE \Rightarrow +*-xyxy$

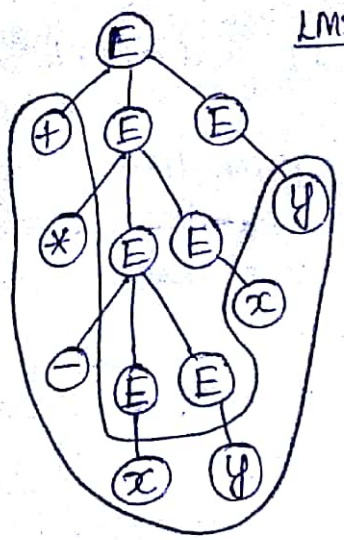
RMD: $E \Rightarrow +EE \Rightarrow +Ey \Rightarrow +*EEy \Rightarrow +*Exy \Rightarrow +*-EExy \Rightarrow$
$+*-Eyxy \Rightarrow +*-xyxy$

DT:

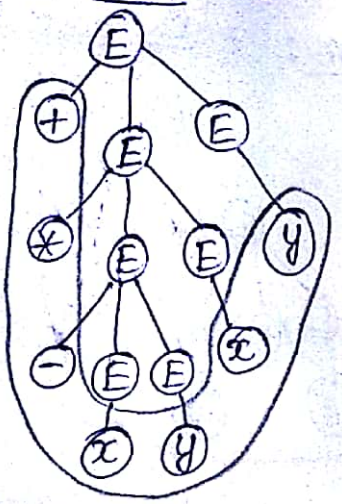
Yield of tree = $+*-xyxy$

Scanned by CamScanner

LMD



RMD



Obtain CFG for following RE

1) $(a+b)^* ab$
2) $(ab)^* a (a+b)^*$
3) $(a+b)^*$

Answers

1) $G = \{ \underset{\uparrow ab,}{aab}, bab, abab, aaab, bbab \ldots \}$

$S \rightarrow Aab$

$A \rightarrow \epsilon \mid aA \mid bA$

2) $G = \{ a, abaa, abab, ababaaa, \ldots \}$

$S \rightarrow AaB$

$A \rightarrow \epsilon \mid abA$

$B \rightarrow \epsilon \mid aB \mid bB$

3) $G = \{ \epsilon, a, b, aa, bb, ab, \ldots \}$

$S \rightarrow \epsilon \mid Aa \mid Ab$

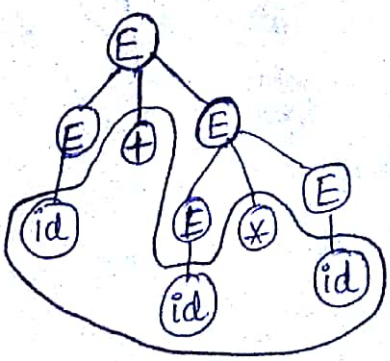
$A \rightarrow S$

Ambiguous Grammer

Eg) $E \rightarrow \overset{①}{E+E} \mid \overset{②}{E*E} \mid \overset{③}{id}$

Generate, $\underset{1}{id} + \underset{2}{id} * \underset{3}{id}$

LMD: $E \overset{①}{\Rightarrow} E+E \overset{③}{\Rightarrow} id+E \overset{②}{\Rightarrow} id+E*E \overset{③}{\Rightarrow} id+E*E \overset{③}{\Rightarrow} id+id*id$

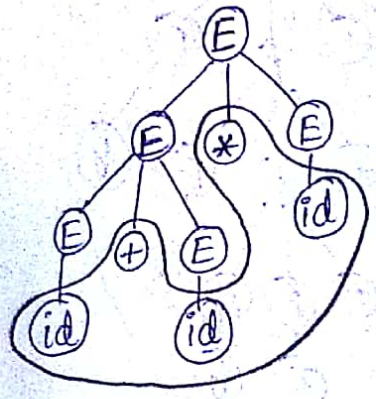


$1+2*3 = 7$

multiplication first

Scanned by CamScanner

LMD: $E \overset{2}{\Rightarrow} E*E \overset{1}{\Rightarrow} E+E*E \overset{3}{\Rightarrow} id+E*E \overset{3}{\Rightarrow} id+id*E \overset{3}{\Rightarrow} id+id*id$



$1+2*3 = 9$ Addition first

Ambiguous Grammer

Let CFG, $G = (V, T, P, S)$ is ambiguous, if there is atleast one string W in T^*, for which we can find two different Parsetree, each with root labelled as an yield W.

Eg: 2) Check whether following grammer is ambiguous or not.

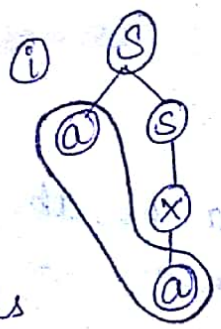
$S \rightarrow aS / X$ "String aa"

$X \rightarrow aX / a$

(i) $S \underset{lm}{\Rightarrow} aS \Rightarrow aX \Rightarrow aa$

(ii) $S \underset{lm}{\Rightarrow} X \Rightarrow aX \Rightarrow aa$

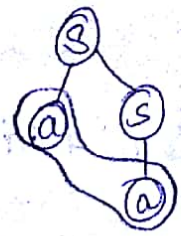
(i)



(ii)

The given grammer is ambigous as it generates one or more trees.

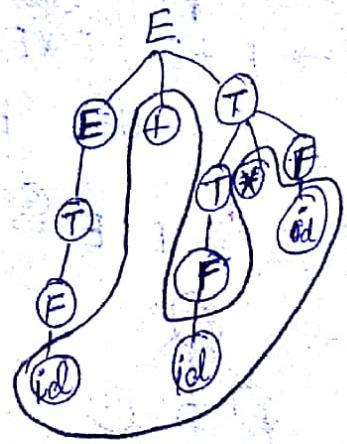
$S \rightarrow a / \cancel{aa} / aS$



$E \rightarrow E+T / T$

$T \rightarrow T*F / F$

$F \rightarrow id$



Scanned by CamScanner

Declaration Statement CFG

Data type variables_list Datatype → int, float, double, char

$D \to TV;$

$T \to int / float / double / char$

$V \to id / id.V$

1) Find LMD, RMD and DT for the string aaabbabbba using CFG.

$S \to aB / bA$

$A \to aS / bAA / a$

$B \to bS / aBB / b$

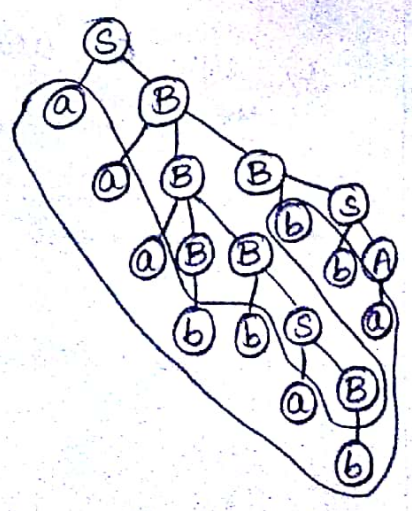
LMD: $S \overset{lm}{\Rightarrow} aS \Rightarrow aaS \Rightarrow aaaS \overset{aaa}{\Rightarrow} bA \Rightarrow aaabbAA \Rightarrow aaabbaA \Rightarrow$
$aaabbabAA \Rightarrow aaabbabbAAA \Rightarrow aaabbabbbAAAA \Rightarrow$

LMD: $S \overset{lm}{\Rightarrow} aB \Rightarrow aaBB \Rightarrow aaaBBB \Rightarrow aaabBB \Rightarrow aaabbSB \Rightarrow$
$aaabbaBB \Rightarrow aaabbabB \Rightarrow aaabbabbS \Rightarrow aaabbabbbA \Rightarrow$
$aaabbabbba$

RMD: $S \overset{rm}{\Rightarrow} aB \Rightarrow aaBB \Rightarrow aaBb \Rightarrow aaaBBb \Rightarrow aaaB$

$S \overset{rm}{\Rightarrow} aB \Rightarrow aaBB \Rightarrow aaBbS \Rightarrow aaBbaB \Rightarrow aaBbabS \Rightarrow aaBbabbA$
$\Rightarrow aaBbabbbA \Rightarrow aaBbabbba \Rightarrow aaaBBbabbba \Rightarrow aaaBbbabbb a$

✓ $S \overset{rm}{\Rightarrow} aB \Rightarrow aaBB \Rightarrow aaBbS \Rightarrow aaBbbA \Rightarrow aaBbba \Rightarrow aaaBBbba$
$\Rightarrow aaaBbbba \Rightarrow aaabSbbba \Rightarrow aaabbAbbba \Rightarrow aaabbabbba$



2) Is the following CFG is ambiguous?

① $S \to AB / aaB$

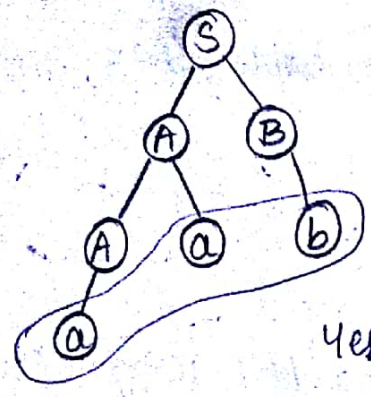
$A \to a / Aa$

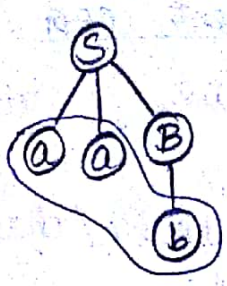
$B \to b$

$S \overset{lm}{\Rightarrow} AB \Rightarrow AaB \Rightarrow aaB \Rightarrow aab$

$S \overset{lm}{\Rightarrow} aaB \Rightarrow aab$

Scanned by CamScanner





Text ①

Yes, it is ambiguous

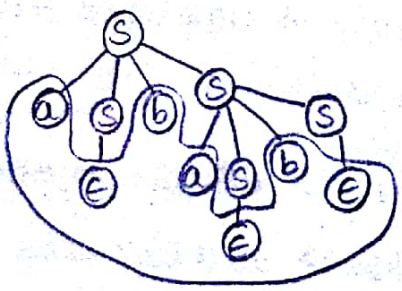
b) $S \to aSbS \mid bSaS \mid \epsilon$

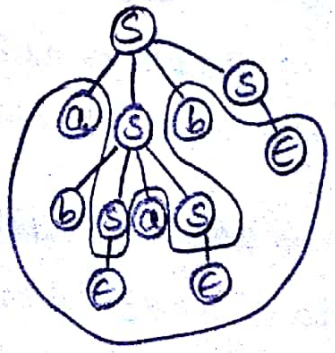
$S \overset{lm}{\Rightarrow} aSbS \Rightarrow abSaSbS \Rightarrow ab\epsilon aSbS \Rightarrow ab\epsilon a\epsilon bS \Rightarrow$
$ababS \Rightarrow ababaSaS \Rightarrow ababa\epsilon aS \Rightarrow ababaaS \Rightarrow$
$aababaa\epsilon \Rightarrow ababaa$

$S \overset{lm}{\Rightarrow} aSbS \Rightarrow$ ~~aSbaSbS~~ $abSaSbS \Rightarrow ab\epsilon aSbS \Rightarrow abaSbS \Rightarrow$
$ababSaSbS \Rightarrow abab$

✓ $S \overset{lm}{\Rightarrow} aSbS \Rightarrow a\epsilon bS \Rightarrow abS \Rightarrow abaSbS \Rightarrow aba\epsilon bS \Rightarrow$
$aba\epsilon b\epsilon \Rightarrow abab$

✓ $S \overset{lm}{\Rightarrow} aSbS \Rightarrow abSaSbS \Rightarrow ab\epsilon aSbS \Rightarrow aba\epsilon bS \Rightarrow abab\epsilon \Rightarrow$ aba





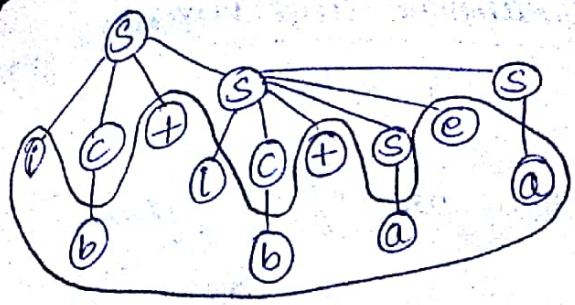
c) $ib + ib + aea$ Using $S \to ic+S \mid ic + SeS \mid a$
$C \to b$

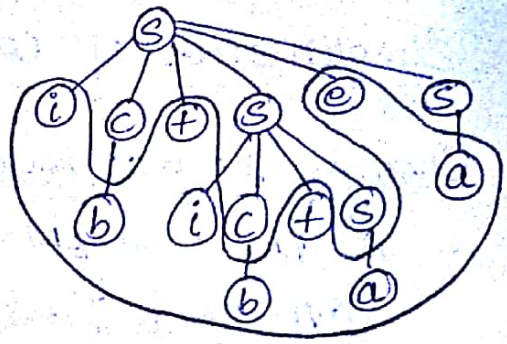
$S \overset{lm}{\Rightarrow} ic + S \Rightarrow$

$ic + ic + SeS \Rightarrow ic + ic + Sea \Rightarrow ic + ic + aea \Rightarrow ib + ib + aea$

$S \overset{lm}{\Rightarrow} ic + SeS \Rightarrow ic + ic + SeS \Rightarrow ic + ic + Sea \Rightarrow ib + ib + aea$

Scanned by CamScanner





2) i) $L = \{a^n b^m c^k \mid k = n+m,\ n \geq 0,\ m \geq 0\}$

$P = \{ S \to \epsilon / B$
$A \to bAc \mid bc \mid \epsilon$
$B \to aBc \mid ac / A / \epsilon \}$

$S \to E / abc$
$aAbAc$
$A \to a/b/c$
$aAbAcA$
$A \to a/bc$

ii) $L = \{a^n b^m c^k \mid n + 2m = k$

$P = \{ S \to \epsilon / B$
$A \to bAcc \mid bcc / \epsilon$
$B \to aBc \mid ac / A / \epsilon \}$

iii) $L = \{a^n b^m c^k \mid n = m$ (or) $m \leq k\}$

$P = \{ S \to \epsilon / A$
$A \to aAbB / S$
$B \to Bc / \epsilon \}$

vi) $L = \{xy \mid x \in \{a,b\}^*$ and $y \in \{ab$ or $ba\}\}$

$P = \{ S \to SB / \epsilon / as / bs$
$B \to ab / ba \}$

v) $L = \{0^m 1^m 2^n \mid m \geq 1,\ n \geq 0\}$

$P = \{ S \to 0A1B$
$A \to 0A1 / \epsilon$
$B \to 2B / \epsilon \}$

vi) $L = \{a^i b^j \mid i \neq j,\ i, j \geq 0\}$

Scanned by CamScanner

Simplification of Context free Grammar and normal forms

Chomsky Normal form

A CFG is in CNF, if all productions are of the form.

$A \to BC$] ⇒ 2 variable

(or)

$A \to a$] ⇒ one terminal

where A, B, C are in V
'a' is in T.

Eg: The grammar,

$S \to AS / a$
$A \to SA / b$ is in chomsky normal form.

The grammar,

$S \to AS / ASS$
$A \to SA / bb$ is not in chomsky normal form

Simplifying CFG (Preliminary steps)

i) Eliminate useless symbols and production.
ii) Eliminate ε productions. (Eg:- $A \to \epsilon$)
iii) Eliminate unit productions. (Eg:- $A \to B$)

(i) EUS

The variables or terminals that do not appear in any derivation of terminal string from the start variable.

Def:- [A symbol X is useful for a grammar $G = (V, T, P, S)$ if there is some derivation of the form.]

$S \xrightarrow{*} \alpha X \beta \xrightarrow{*} w$, where w is in T^*.

Part of Definition:

If X is not useful, then it is useless

Scanned by CamScanner

Elimination steps

1) X is generating if $X \xrightarrow{*} w$ for some terminal string.
2) X is reachable, if there is a derivation
$S \xrightarrow{*} \alpha X \beta$ for some α and β

⇒ A symbol that is useful will be both generating and reachable.

⇒ First eliminate the signals that are not generative, then the ones that are not reachable.

Eg. ⇒ $S \to AB/a$ } useful
$A \to b$
$C \to c$ ⇒ not reachable (useless) but generating
$B \to d$ } ful

⇒ $S \to AB/a$
$A \to b/C$
$C \to D$
$B \to d$

Here C, D are not generating

Eliminate useless productions form

① $S \to a/aA/B$
$A \to aB/a$
$B \to Aa$
$C \to cCD$
$D \to ddd$

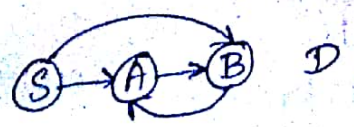
$G_1 = (V_1, T_1, P_1, S)$

P_1	V_1	T_1
$A \to a$	{A}	{a}
$D \to ddd$	{A, D}	{a, d}
$S \to a$	{A, D, S}	{ " }
$S \to aA$	{A, D, S}	{ " }
$B \to Aa$	{S, A, D, B}	{ " }
$S \to B$	{ " }	{ " }
$A \to aB$	{ " }	{ " }

(Generating)

$G_2 = (V_2, T_2, P_2, S)$

P_2	V_2	T_2
$S \to a/aA/B$	{SABD}	{a}
$A \to a/aB$	"	"
$B \to Aa$	"	"



(Reachable)

Scanned by CamScanner

(ii) $S \to aA / bA$
$A \to aA / a$
$B \to bB$
$D \to ab / Ea$
$E \to ac / d$

$G_1 \Rightarrow (V_1, T_1, P_1, S)$

P_1	V_1	T_1
$A \to a$	$\{A\}$	$\{a\}$
$D \to ab$	$\{A, D\}$	$\{a, b\}$
$E \to ac/d$	$\{A, D, E\}$	$\{a, b, c, d\}$
$S \to aA/bA$	$\{A, S, D, E\}$	$\{\}$
$A \to aA$	$\{\}$	$\{\}$
$D \to Ea$	$\{\}$	$\{\}$

$G_2 \Rightarrow (V_2, T_2, P_2, S)$

P_2	V_2	T_2
$S \to aA/bA$	$\{S, A\}$	$\{a, b\}$
$A \to a/aA$	$\{S, A\}$	$\{a, b\}$

S → A B D E

Steps

Generality

First add only the productions that have only terminal, then according to the variable set, add the productions with those variables also.

Reachable

Then start from starting variable & continue.

Eliminate ϵ - productions

Definition:- * Any production of a context free grammar of the form, $A \to \epsilon$ is called ϵ - production.

* Any variable A for which the derivation $A \xrightarrow{*} \epsilon$ is possible is called Nullable.

Computing Nullable Variable (V_N)

Let $G = (V, T, P, S)$ be a CFG. Find all the nullable variable of G by the following iterative algorithm,

Basic: If $A \to \epsilon$ is a production of G, then A is nullable.

Induction: If there is production $B \to C_1 C_2 \cdots C_k$, where each C_i is nullable then B is nullable.

Scanned by CamScanner

Theorem

Let G be any CFG with ϵ not in $L(G)$. Then, there exists an equivalent grammar G_1, having no ϵ productions.

Proof We first find the set V_n of all nullable variables of G, using the following steps,

1) Find all production $A \to \epsilon$, put A into V_n.
2) Repeat the following steps until no other variables are added to V_n ....

For all productions $B \to A_1 A_2 \cdots A_k$, where $A_1 A_2 \cdots A_k$ are in V_N, put B into V_n.

Eg: 1) Eliminate ϵ - production

a) $S \to AaB \mid aaB$

$A \to \epsilon$

$B \to bbA \mid \epsilon$

$\left.\begin{array}{l} A \to \epsilon \\ B \to \epsilon \end{array}\right\} \Rightarrow A, B$ are nullable variables

P_1	V_N
$A \to \epsilon$	$\{A\}$
$B \to \epsilon$	$\{A, B\}$

P_2

$S \to aB \mid Aa \mid a \mid aa \mid AaB \mid aaB$

$B \to bbA \mid bb$

2) $S \to ABaC$

$A \to BC$

$B \to b \mid \epsilon$

$C \to D \mid \epsilon$

$D \to d$

P_1

$\left.\begin{array}{l} A \to \epsilon\epsilon \\ B \to \epsilon \\ C \to \epsilon \end{array}\right\} \Rightarrow A, B, C$ are nullable

P_2

$S \to Bac \mid Aac \mid ABa \mid ABac \mid ac \mid a \mid Aa \mid Ba$

$B \to b$

$C \to D$

$D \to d$

Scanned by CamScanner

3) $S \to BAAB$

$A \to 0A2 \mid 2A0 \mid \epsilon$

$B \to AB \mid 1B \mid \epsilon$

P_1	V_n
$S \to \epsilon\epsilon\epsilon\epsilon$	$\{S\}$
$A \to \epsilon$	$\{S, A\}$
$B \to \epsilon$	$\{S, A, B\}$

$S, A, B \to$ Nullable Variables

P_2

$S \to BAAB \mid AAB \mid BAB \mid BAA \mid AB \mid BA \mid AA \mid BB \mid B \mid A$

$A \to 0A2 \mid 02 \mid 2A0 \mid 20$

$B \to AB \mid 1B \mid 1 \mid A \mid B$

Elimination of Unit Productions

Def:- Any production of a Context free grammar of the form $A \to B$, where $A, B \in V$ is called Unit-Production.

Basis & Induction

Problems:-

1) $S \to Aa \mid B$

$B \to A \mid bb$

$A \to a \mid bc \mid B$

Unit productions	Non unit productions
$S \to B$	$S \to Aa$
$B \to A$	$B \to bb$
$A \to B$	$A \to a \mid bc$

$S \overset{*}{\Rightarrow} B$ (S B)

$S \Rightarrow A$ (S A)

$B \overset{*}{\Rightarrow} A$ (BA)

$A \overset{*}{\Rightarrow} B$ (AB)

$\xrightarrow{(\because \text{ by applying } B \to A)}$

$S \to bb$

$S \to a \mid bc$

$B \to a \mid bc$

$A \to bb$

Merge both

$S \to Aa \mid bb \mid a \mid bc$

$B \to bb \mid a \mid bc$

$A \to a \mid bc \mid bb$

Steps

① Write Unit & non-unit productions separately.

② Write unit pairs like (SB) and replace B with non-unit terms

③ Merge Non-unit productions & newly generated non-unit productions.

Scanned by CamScanner

1) $S \to AB$
$A \to a$
$B \to C \mid b$
$C \to D$
$D \to E \mid bC$
$E \to d \mid Ab$

Unit prod

$B \to C$
$C \to D$
$D \to E$

Non-unit prod

$S \to AB$
$A \to a$
$B \to b$
$D \to bC$
$E \to d / Ab$

$B \to C$ (BC)
$B \to D$ (BD)
$B \to E$ (BE)
$C \to D$ (CD)
$C \to E$ (CE)
$D \to E$ (DE)

$B \to C \to D \to E$

$B \to bC$ as $C \to D$
$B \to bC$

$B \to d \mid Ab$
$C \to d \mid Ab$
$D \to d \mid Ab$

$S \to AB$
$A \to a$
$B \to b \mid d \mid Ab$
$C \to d \mid Ab$
$D \to d \mid Ab \mid bC$
$E \to d \mid Ab$

* G
$\downarrow$
ϵ production
$\downarrow$
G_1
$\downarrow$
Unit production
$\downarrow$
G_2
$\downarrow$
Useless production
$\downarrow$
G_3

1) $S \to aA \mid aBB$
$A \to aaA \mid \epsilon$
$B \to bB \mid bbC$
$C \to B$

P: $A \to \epsilon$

V_n: $\{A\}$

G_1
$S \to a \mid aBB \mid aA$
$A \to aa \mid aaA$
$B \to bB \mid bbC$
$C \to B$

Scanned by CamScanner

1)

Unit production

$C \to B$

Non-unit production

$S \to aBB \mid aA \mid a$
$A \to aa \mid aaA$
$B \to bB \mid bbC$
$C \to B$

G_2

$S \to aBB \mid aA \mid a$
$A \to aa \mid aaA$
$B \to bB \mid bbC$
$C \to bB \mid bbC$

Useless production

Generating

P_1	V_1	T_1
$S \to a$	$\{S\}$	$\{a\}$
$A \to aa$	$\{S, A\}$	$\{a\}$
$S \to aA$	$\{S, A\}$	$\{a\}$
$A \to aaA$	$\{S, A\}$	$\{a\}$

Reachable

P_2	V_2	T_2
$S \to aA \mid a$	$\{S, A\}$	$\{a\}$
$A \to aa \mid aaA$	$\{S, A\}$	$\{a\}$

$S \longrightarrow A$

2) $S \to abAB$
$A \to bAB \mid \epsilon$
$B \to BAa \mid A \mid \epsilon$

ϵ - Production

$A \to \epsilon$
$B \to \epsilon$

G_1

$S \to abAB \mid abB \mid abA \mid ab$
$A \to bAB \mid bA \mid bB \mid b$
$B \to BAa \mid Aa \mid Ba \mid a \mid A$

Unit-production

$B \to A$

Non-Unit production

$S \to abAB \mid abB \mid abA \mid ab$
$A \to bAB \mid bA \mid bB \mid b$
$B \to BAa \mid Aa \mid Ba \mid a \mid A$

G_2

$S \to abAB \mid abB \mid abA \mid ab$
$A \to bAB \mid bA \mid bB \mid b$
$B \to BAa \mid Aa \mid Ba \mid a \mid bAB \mid bA \mid bB \mid b$

Scanned by CamScanner

Problem

1) $S \to 0A \mid 1B$
$A \to 0AA \mid 1S \mid 1$
$B \to 1BB \mid 0S \mid 0$

Step I

$S \to S_1A \mid S_2B$
$S_1 \to 0$
$S_2 \to 1$
$A \to S_1AA \mid S_2S \mid 1$
$B \to S_2BB \mid S_1S \mid 0$

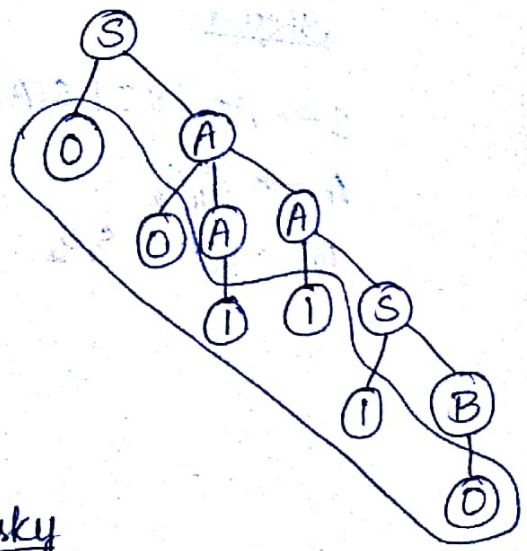
Step 2

$S \to S_1A \mid S_2B$
$S_1 \to 0$
$S_2 \to 1$
$A \to S_1A_1 \mid S_2S \mid 1$
$A_1 \to AA$
$B \to S_2B_1 \mid S_1S \mid 0$
$B_1 \to BB$

Get string 001110

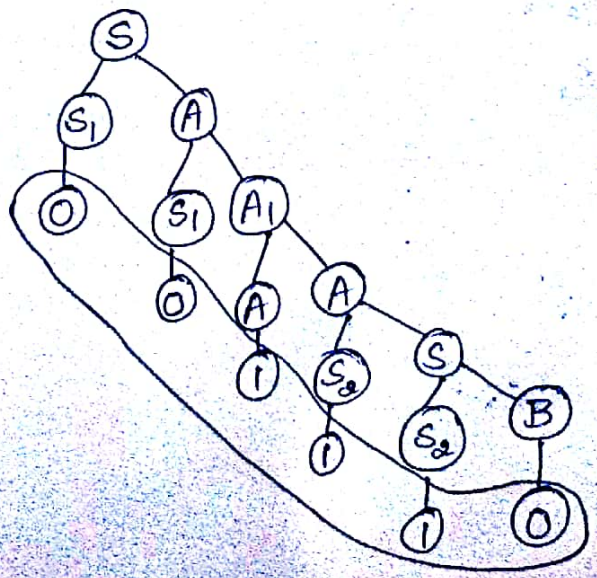
CFG

LMD $\Rightarrow$ $S \underset{lm}{\Rightarrow} 0A \Rightarrow 00AA \Rightarrow 0011S \Rightarrow 00111B \Rightarrow 001110$



Chomsky

$S \Rightarrow S_1A \Rightarrow 0A \Rightarrow 0S_1A_1 \Rightarrow 00A_1 \Rightarrow 00AA \Rightarrow 001A \Rightarrow 001S_2S \Rightarrow$
$0011S \Rightarrow 0011S_2B \Rightarrow 00111B \Rightarrow 001110$



Scanned by CamScanner

Q) $S \to AB \mid aB$
$A \to aab \mid \epsilon$
$B \to bbA$

P	V_n	
$A \to \epsilon$	$\{A\}$	$\to$ Nullable

$S \to AB \mid aB \mid B$
$A \to aab$
$B \to bbA \mid bb$

<u>Unit production</u>

$S \to B$

<u>Non-unit production</u>

$S \to AB \mid aB \mid B$
$A \to aab$
$B \to bbA \mid bb$

$S \to AB \mid aB \mid bbA \mid bb$
$A \to aab$
$B \to bbA \mid bb$

<u>Chomsky</u>

<u>Step 1</u>

$S \to AB \mid S_1B \mid S_2S_2A \mid S_2S_2$
$S_1 \to a$
$S_2 \to b$
$S \to AB \mid S_1B \mid S_3A \mid S_2S_2$
$S_3 \to S_2S_2$
$A \to S_1S_1S_2$
$A \to S_4S_2$
$S_4 \to S_1S_1$
$B \to S_2S_2A \mid S_2S_2$
$B \to S_3A \mid S_2S_2$

<u>Step 2</u>

$S \to AB \mid S_1B \mid S_3A \mid S_2S_2$
$A \to S_4S_2$
$B \to S_3A \mid S_2S_2$

Scanned by CamScanner