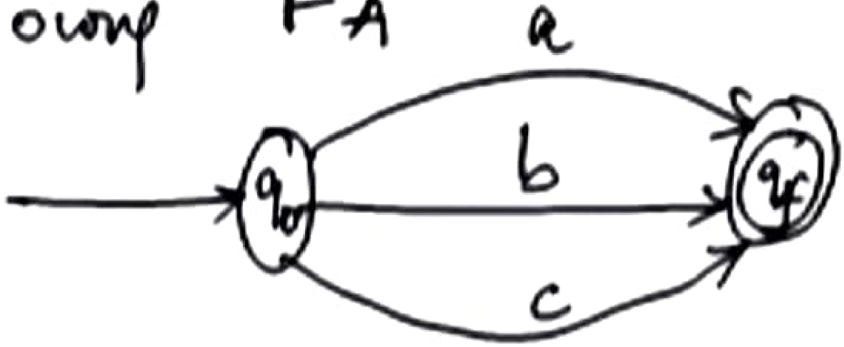
Q?:

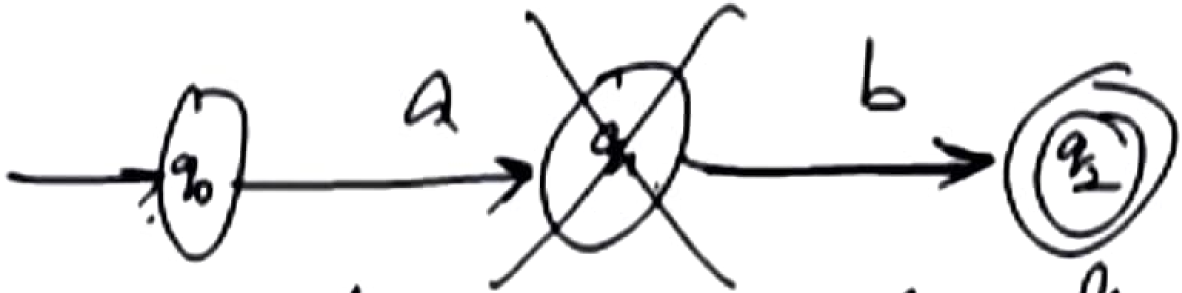
(1) Construct the equivalent RE for the following FA



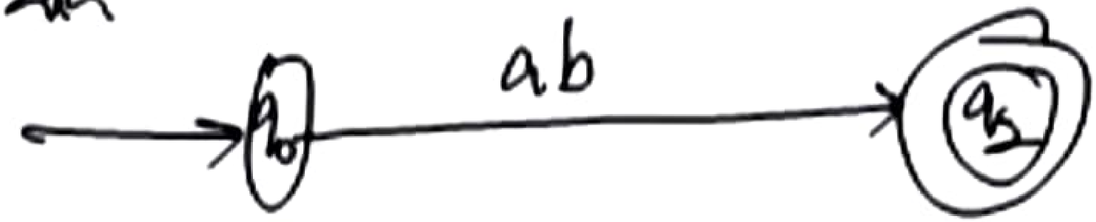
Union

$\Rightarrow$ $\underline{(a+b+c)}$

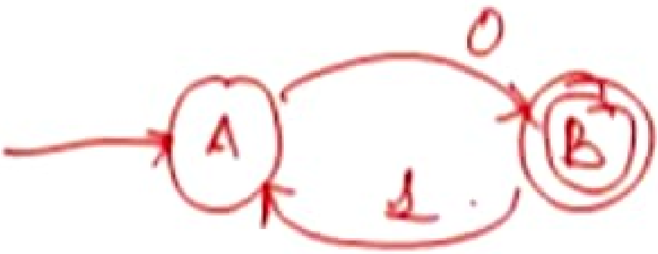
(2)



eliminate i/s if there is a path

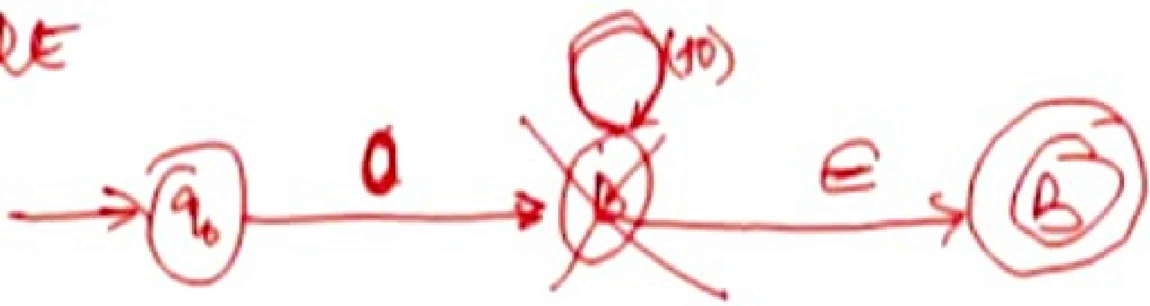


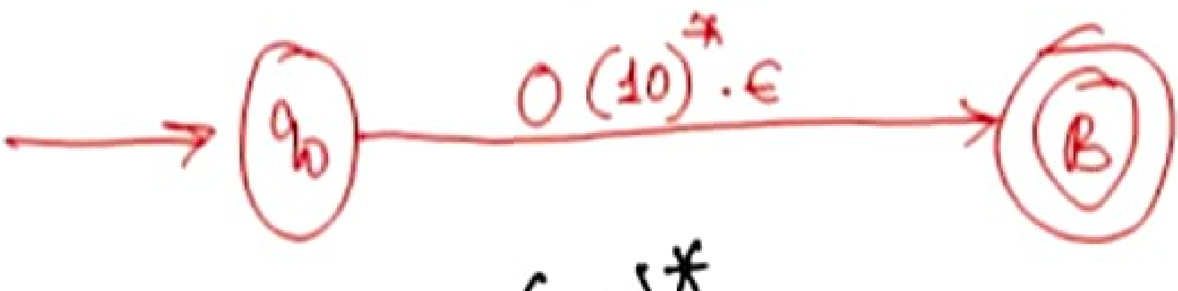
$=$ $\underline{ab}$ Concatenation



The Start State has the incoming edge.
Final State has an o/g edge.
hence apply the (1) & (2)

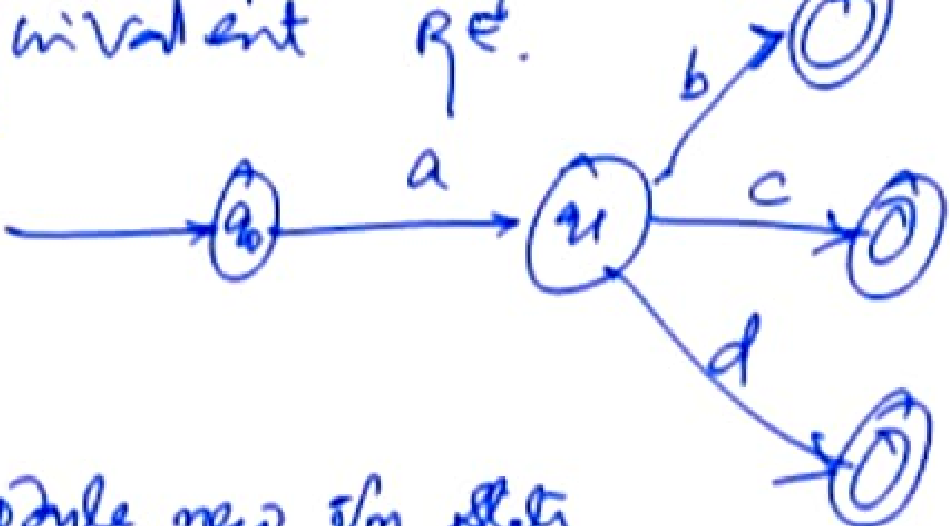
eliminate the i/m States to get
RE



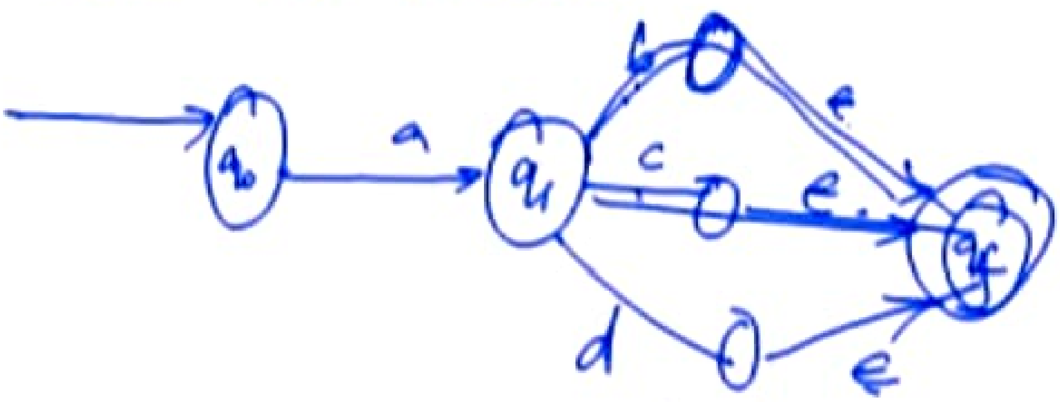


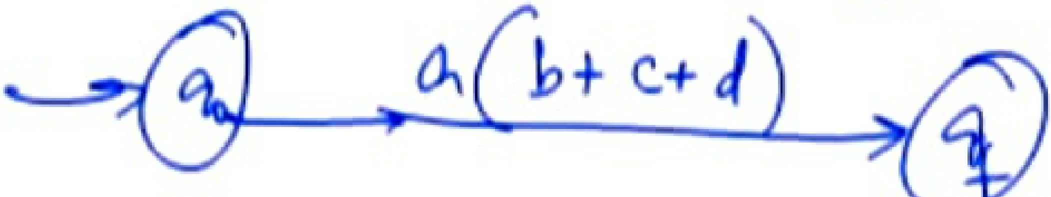
$\Rightarrow$ $\underline{0(10)^*}$.

Q. For the following FA write its equivalent RE.



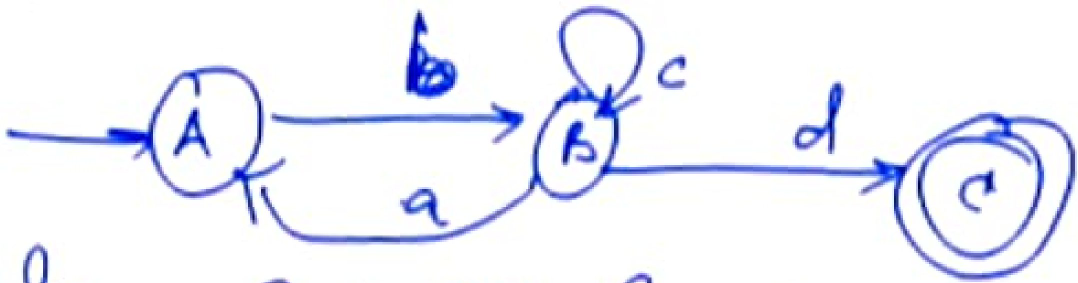
introduce new f/n state



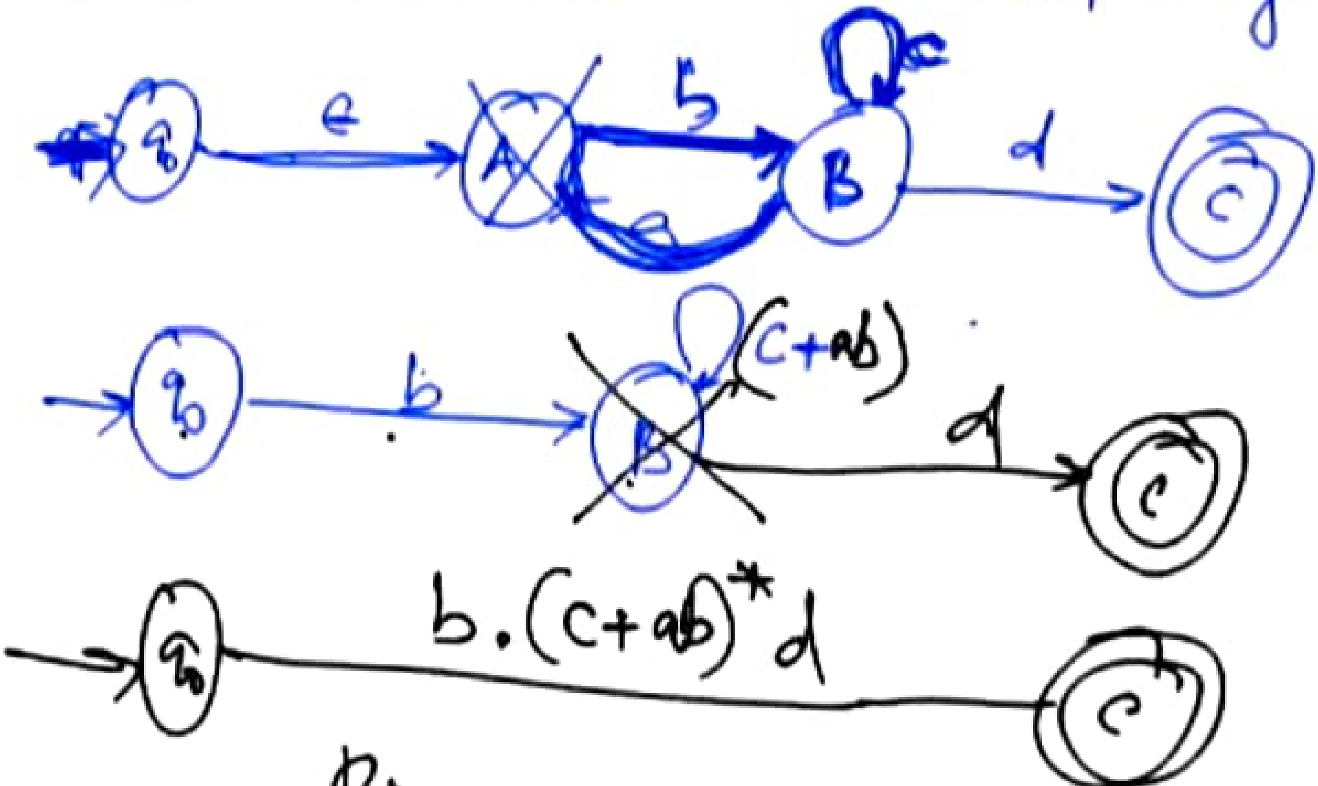


→ a(b+c+d)

Q. For the following FA, write its equivalent RE.



Soln: Start state has incoming edge.



RE

b. (c+

$S \rightarrow ABC$ — (1)
$A \rightarrow a$ — (2)
$B \rightarrow b$ — (3)
$C \rightarrow \epsilon$ — (4)

What is the string generated by the grammar?

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

$V = \{S, A, B, C\}$

$T = \{a, b\}$

$P = \{S \rightarrow ABC;\ A \rightarrow a;\ B \rightarrow b;\ C \rightarrow \epsilon\}$

S = S is Start Symbol.

To generate the string, apply the derivation.

$S \rightarrow ABC$ (1)
$\rightarrow aBC$ (2)
$\rightarrow abC$ (3)
$\rightarrow ab.\epsilon$ (4)
$\rightarrow ab$ string w, generated by Grammar G.

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

Notations:

(1) Non-terminals are represented by upper case letters. Non-terminals are also called as 'VARIABLES' (V)

(2) Terminals are represented by lower case letters, digits and special symbols (T)

Terminals cannot be replaced.

(3) Production: (Rules) (P)

Always indicated by using $\longrightarrow$

$$\underset{NT}{A} \rightarrow \underset{T}{a}\underset{NT}{B}$$

RHS of each production will be sequence of Terminals and/or non-terminals

$\boxed{A \rightarrow \alpha} \rightarrow$ General form of production

$A \in V$ and $\underline{\alpha \in (V \cup T)^*}$

i.e., $(V+T)^*$

Date : 19.10.2020:

1. Write the grammar to generate "signed integers".

Soln: Integer : 'a number'

Defn: . A digit is a number
. number followed by digits
. Digit followed by number

Ex: +28, −35, ϵ50ϵ

$S \to + / - / \epsilon$

SD
SN

16 product

$S \to +$
$S \to -$
$S \to \epsilon$
$N \to D$
$D \to 0/1/2/3/4/5/6/7/8/9$
$N \to ND/DN$ -
$\boxed{I \to SN}$ -

10

$SI \to SN$
$\to +N$
$\to +D$
$\to +5$

Q. what is the language generated by the following grammar?

G: $E \to E+E$ — (1)
$E \to E*E$ — (2)
$E \to a$ — (3)
$E \to b$ — (4)
$E \to c$ — (5)

Soln: G is given, Need to write $L(G)$.

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

$V = \{E\}$

$T = \{a, b, c, +, *\}$

$P = \{E \to E+E \mid E*E \mid a \mid b \mid c\}$

E is the start symbol.

$E \to \underline{E}+E$ — (1)
$\to \underline{E}*\underline{E}+E$ — (2)
$\to a*\underline{E}+E$ — (3)
$\to a*b+\underline{E}$ — (4)
$\to \boxed{a*b+c}$ — (5)

This grammar is to generate arithmetic exprns with a, b & c as operands and + & * as operators

Q. Write grammar to generate the

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

Sol:

$L = \{ \epsilon, ab, ba, abaabb \ldots \}$

$S \to \epsilon$ — ①

$S \to aSb$ — ②

$S \to bSa$ — ③

$S \to SS$ — ④

aabb

$b\,ab\,a$

baba

ba

ab

abaabb

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

$V = \{ S \}$

$T = \{ a, b \}$

$P = \{ S \to aSb \mid bSa \mid \epsilon \}$

S is Start Symbol

~~$S \to aSb$ — ②~~

~~ex → abSab — ③~~

abaabb

$S \to aSb$ — ②

$\to aSSb$ — ④

$\to abSaSb$ — ③

$\to abaSb$ — ①

$\to abaaSbb$ — ②

$\to \boxed{abaabb}$ ①

$P = \{$

$I \rightarrow SN$ — (1)

$N \rightarrow ND \mid DN \mid D$

$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$S \rightarrow + \mid - \mid \epsilon \ \}$

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

$V = \{N, D, S\}$ $T = \{0, 1, 2, 3 \ldots 9, +, -\}$

Start Symbol: Start symbol I is start symbol

To show the derivation for -193

$$
\begin{aligned}
I &\rightarrow SN \\
&\rightarrow -N \\
&\rightarrow -ND \\
&\rightarrow -NDD \\
&\rightarrow -DDD \\
&\rightarrow -1DD \\
&\rightarrow -19D \\
&\boxed{\rightarrow -193}
\end{aligned}
$$

Replace only one non terminal at each step.

+26

$$
\begin{aligned}
I &\rightarrow SN \\
&\rightarrow +N \\
&\rightarrow +ND \\
&\rightarrow +DD \\
&\rightarrow +2D \\
&\boxed{\rightarrow +26}
\end{aligned}
$$

Date: 20.10.2020

Recap: How to write Grammar or How to generate the strings (language) from G.
$L(G)$

Q. What is the language generated by the following grammar G.

$$\left.\begin{aligned} S &\to 0A \mid \epsilon \\ A &\to 1S \end{aligned}\right\} G$$

$V = \{S, A\}$
$T = \{0, 1\}$
S is start symbol

$L(G) = ?$

Solution: To find $L(G)$, apply the derivation process.

$S \to 0A$
$\to 01S$
$\to 0101$
$\to 0101S$
$\to 01010A$
$\to 010101S$
$\vdots$
$\to (01)^3$
$\vdots$
$(01)^n$

ϵ

$L(G) = \{(01)^n : n \geq 0\}$

$n = 0$ ✓
$S \to \epsilon$
$n = 1$ 01
$S \to 0A$
$\to 01S$
$\to 01$ ✓

Q. Write grammar to generate balanced paranthesis.

Sol:

$((a+b)*c)$ ✓ $\quad$ $(a+b)*c)$ ✗

To write the possible balanced paranthesis

ϵ ✓
( ) ✓
[ ] ✓
{ } ✓

$S \rightarrow \epsilon$ — ①
$S \rightarrow (S)$ — ②
$S \rightarrow [S]$ — ③
$S \rightarrow \{S\}$ — ④
$S \rightarrow SS$. — ⑤

( ) ( )

$S \rightarrow SS$
$\rightarrow (S)S$
$\rightarrow ( )S$
$\rightarrow ( )\epsilon$
$\rightarrow ( )$

Similarly all valid possibilities can be generated

$V = \{S\}$

$T = \{ (,), [,], \{, \} \}$

S — is Start Symbol

Ex: [( ) ( )]

$S \rightarrow [S]$ — ③
$\rightarrow [SS]$ — ⑤
$\rightarrow [(S)S]$ — ②

Left Most Derivation - LMD
Right Most Derivation - RMD
Derivation Tree - D-Tree.

Consider the following Example.

$E \to E+E$ - (1)
$E \to E-E$ - (2)
$E \to E*E$ - (3)
$E \to E/E$ - (4)
$E \to a$ - (5)

also consider $\omega = a+a*a$

LMD.

$E \to E+E$ - (1)
$\to a+E$ - (5)
$\to a+E*E$ - (3)
$\to a+a*E$ - (5)
$\boxed{\to a+a*a}$ → (5)

left most n/t at each step.

RMD:

$E \to E+E$ - (1)
$\to E+E*E$ - (3)
$\to E+E*a$ - (5)
$\to E+a*a$ - (5)
$\boxed{\to a+a*a}$ - (5)

right most nonterminal at each step of derivation.

$E \to E+E$ — (1)

$E \to E-E$ — (2)

$E \to E*E$ — (3)

$E \to E/E$ — (4)

$E \to a$ — (5)

$w = a+a*a$

LMD

$E \to \underline{E}+E$

$\to a+\underline{E}$

$\to a+\underline{E}*E$

$\to a+a*\underline{E}$

$\to a+a*a$

RMD

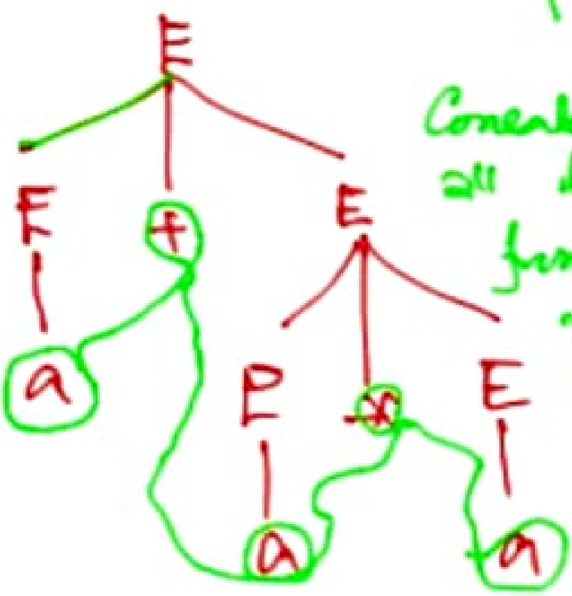
$E \to E+\underline{E}$

$\to E+E*\underline{E}$

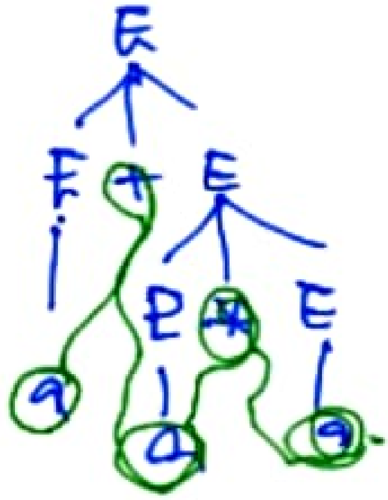
$\to E+\underline{E}*\underline{a}$

$\to \underline{E}+a*a$

$\to a+a*a$



Concatenate all the leaves from L to R yields the string w



D 205
206
182
181

Derivation Tree : D-tree

Tree is a non linear, hierarchical data structure which has special node called root and children node.

, Graph

Every tree is a graph

"A tree is an acyclic digraph"

directed downwards

Hasse diagram

DMS →

The process of derivations (either LMD or RMD) can be expressed in the form of tree which is called D tree

RMD

$w =$ ibtibtaea

$S \to iCtS$ — (1)

$S \to iCtSeS$ — (2)

$S \to a$ — (3)

$C \to b$ — (4)

Right Most Derivation

$S \to iCtS$ — (1)

$\to iCtiCtSeS$ — (2)

$\to iCtiCtSea$ — (3)

$\to iCtiCtaea$ — (3)

$\to iCtibtaea$ — (4)

$\to \boxed{ibtibtaea}$ — (4)

Q. Consider the following grammar.

$S \to S+S$ — (1)

$S \to S*S$ — (2)

$S \to a$ — (3)

Also consider $\boxed{w = a+a*a}$

It is possible to derive the same string w from two different LMDs.

LMD 1	LMD 2
$S \to \underline{S}+S$ — (1)	$S \to \underline{S}*S \to$ (2)
$\to a+\underline{S}$ — (3)	$\to \underline{S}+S*S$ — (1)
$\to a+\underline{S}*S$ — (2)	$\to a+\underline{S}*S$ — (3)
$\to a+a*\underline{S}$ — (3)	$\to a+a*\underline{S}$ — (3)
$\to \boxed{a+a*a}$	$\to \boxed{a+a*a}$ — (3)
$\boxed{1\ 3\ 2\ 3}$	2 1 3 3 3

Same string $w = a+a*a$

Same method. LMD

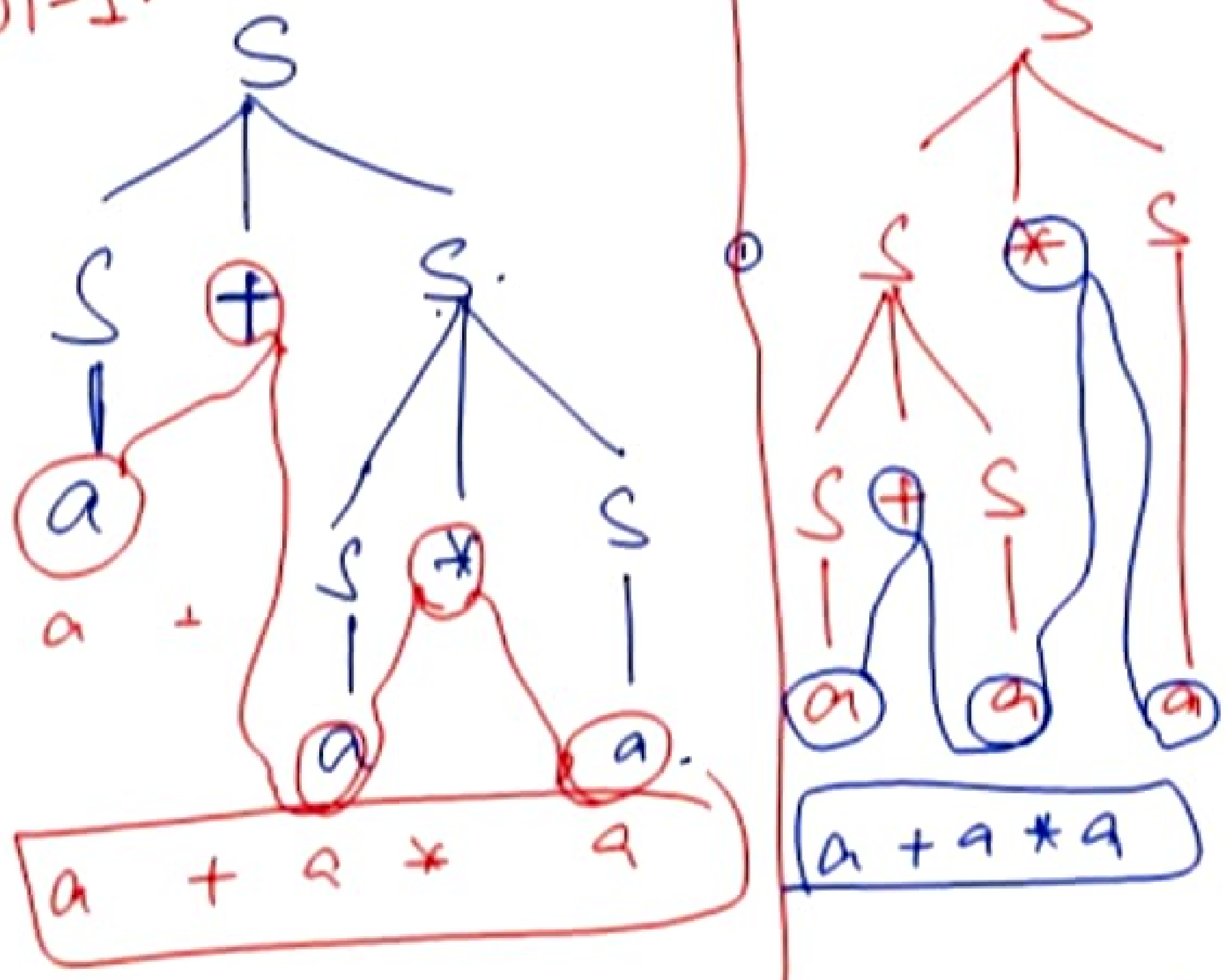
Two derivations are DIFFERENT

Derivation trees

LMD-1, $\boxed{1323}$

2 1 3 3 3

DT-1



Conclusion: There are two different d-trees for the same string w under the same method [LMD], hence the G is an ambiguous grammar

Q. Verify whether the following G is ambiguous.

$S \to iCtS \;(1) \;/\; iCtSeS \;(2) \;/\; a \;(3)$

$C \to b \;(4)$

$\boxed{w = ibtibtaea}$

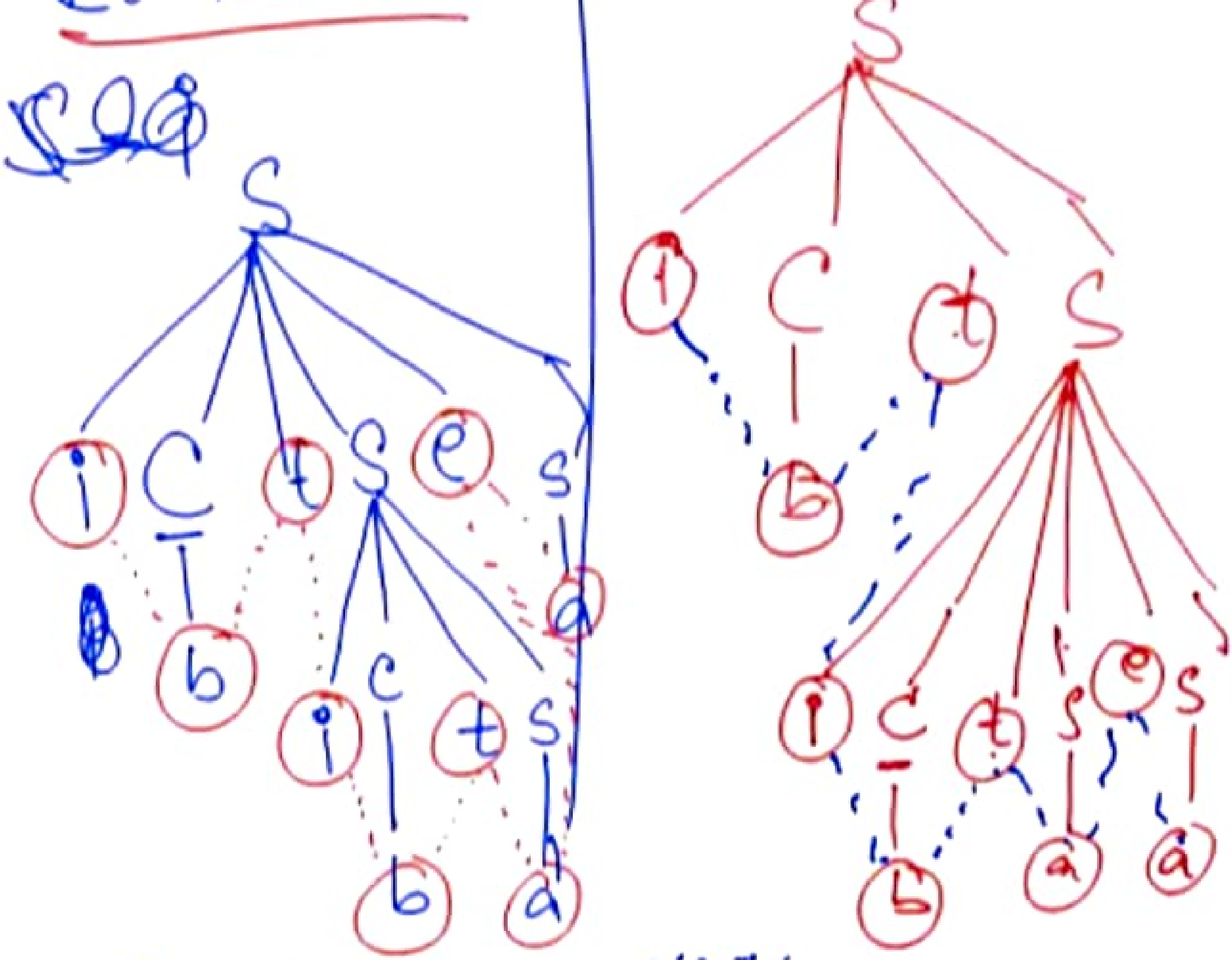
Soln:

LMD-1	LMD-2
$S \to iCtSeS$ — 2	$S \to iCtS$ — 1
$\to ibtSeS$ — 4	$\to ibtS$ — 4
$\to ibtiCtSeS$ 1	$\to ibtiCtSeS$ — 2
$\to ibtibtSeS$ — 4	$\to ibtibtSeS$ — 4
$\to ibtibtaeS$ — 3	$\to ibtibtaeS$ — 3
$\to ibtibtaea$ — 3	$\to ibtibtaea$ — 3
2 4 1 4 3 3	1 4 2 4 3 3

$241433 \neq 142433$

To write two different derivation trees

LMD - 1 | LMD - 2



String is same = w = ibtibtaea
Method is same = LMD
Two different derivations (d-trees)
Hence the G is ambiguous

Consider

$$S \rightarrow \underset{1}{aSbS} \mid \underset{2}{bSaS} \mid \underset{3}{\epsilon}$$

abab ✓

LMD-1

$S \rightarrow bSaS$ - 2

$\rightarrow baSbSaS$ - 1

$\rightarrow babSaS$ - 3

$\rightarrow babaS$ 3

$\rightarrow baba$ 3

2, 1, 3, 3, 3.

LMD-2

$S \rightarrow bSaS$ - 2

$\rightarrow baS$ - 3

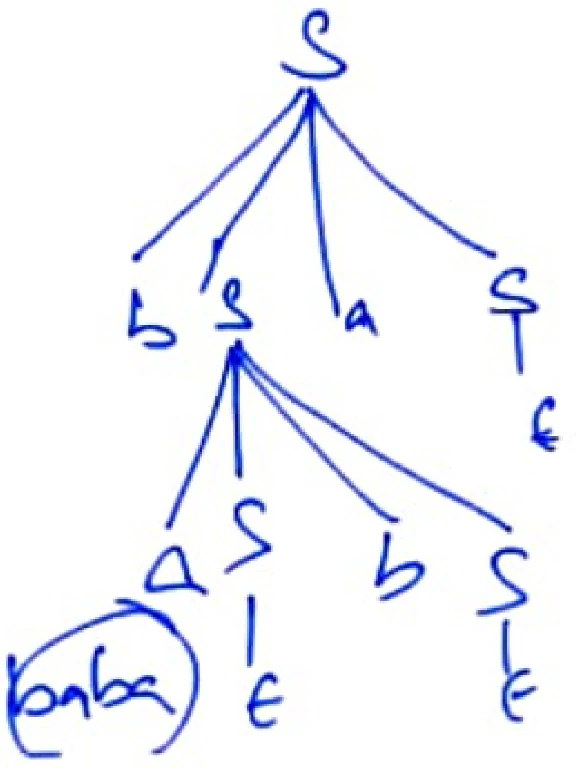
$\rightarrow babSaS$ - 2

$\rightarrow babaS$ - 3

$\rightarrow baba$ - 3

2, 3, 2, 3, 3

Dt-1



Dt-2