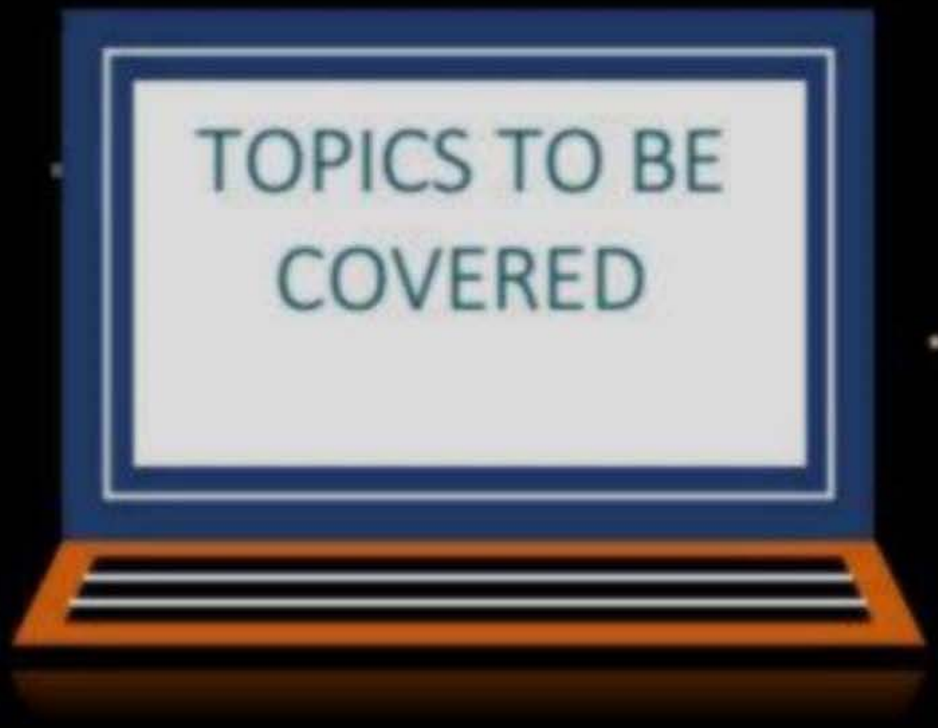
CS & IT ENGINEERING

Compiler Design

Syntax Directed Translations

Lecture No. 2

By- DEVA Sir



SDTs

- Definitions
- Evaluations

SDTs Definitions :

1. L-attributed Grammar
2. S-attributed Grammar

L-attributed SDT

① Computation depends on parent / left siblings / children.

② Translation can be anywhere

$S \rightarrow ABC$

$\{A.x = S.y;$
$B.x = A.x + S.x;$
$C.x = A.x + 1;$
$S.x = B.x * 2\}$

S-attributed SDT

① computation depends on only children [Synthesized attributes only]

② Every Translation should appear only at end of production.

$E \rightarrow AB \{E.x = A.x + B.y\}$

Attribute type → Inherited
Attribute type → Synthesized

SDT type (Definition) → L-attributed Grammar
SDT type (Definition) → S-attributed Grammar

Find Type/Definition of SDT.

① $S \to a \quad \{ S.x = a.val \}$ ⟹ S-attributed Grammar (also, L-attributed)

② $S \to Aa \quad \{ A.x = S.y \}$

$S \to Bb \quad \{ S.y = B.x \}$

$A \to a \quad \{ A.x = a.val \}$

$B \to b \quad \{ B.y = b.val \}$

⟹ L-attributed but not S-attributed

x is neither inherited nor synthesized attribute

y is synthesized attribute

(3)

$E \rightarrow E_1 + E_2 \quad \{ E.val = E_1.val + E_2.val \}$

$E \rightarrow a \quad \{ E.val = a.val \}$

↳ S-attributed Grammar
(L-attributed)

↳ val is <u>synthesized</u> attribute

(4)

$S \rightarrow AB \quad \{A.x = B.x\}$ (Right Sibling)

$A \rightarrow a \quad \{\ \}$

$B \rightarrow b \quad \{B.x = b.val\}$

⇒ Not L-attributed SDT

(neither L-attributed nor S-attributed SDT)

(5)

$D \rightarrow TL$	$\{ L.type = T.type \}$ (T.type: Left sibling)
$T \rightarrow int$	$\{ T.type = int \}$ (int: child)
$T \rightarrow float$	$\{ T.type = float \}$ (float: child)
$L \rightarrow id$	$\{ \}$

This SDT is L-attributed but not S-attributed

⑥ $S \rightarrow \{ S.x = \underbrace{a.val}_{\text{child}} \}\ a$

translation is not at end

⇒ Not S-attributed

⇒ It is L-attributed

Attributes not computed

⑦ $E \rightarrow a \ \{ print(a.val) \}$ ⇒ S-attributed SDT (L-attributed SDT)

⑧ $E \rightarrow \{ print(a.val) \}\ a$ ⇒ only L-attributed SDT

(9)

'i' is inherited attribute

'S' is Synthesized attribute

$A \rightarrow PQ$

$P.i = A.i + 2$; (parent)

$Q.i = P.i + A.i$; (left sibling, parent)

$A.S = P.S + Q.S$; (children)

⇒ only L-attributed Definition

(10)

i is Inherited

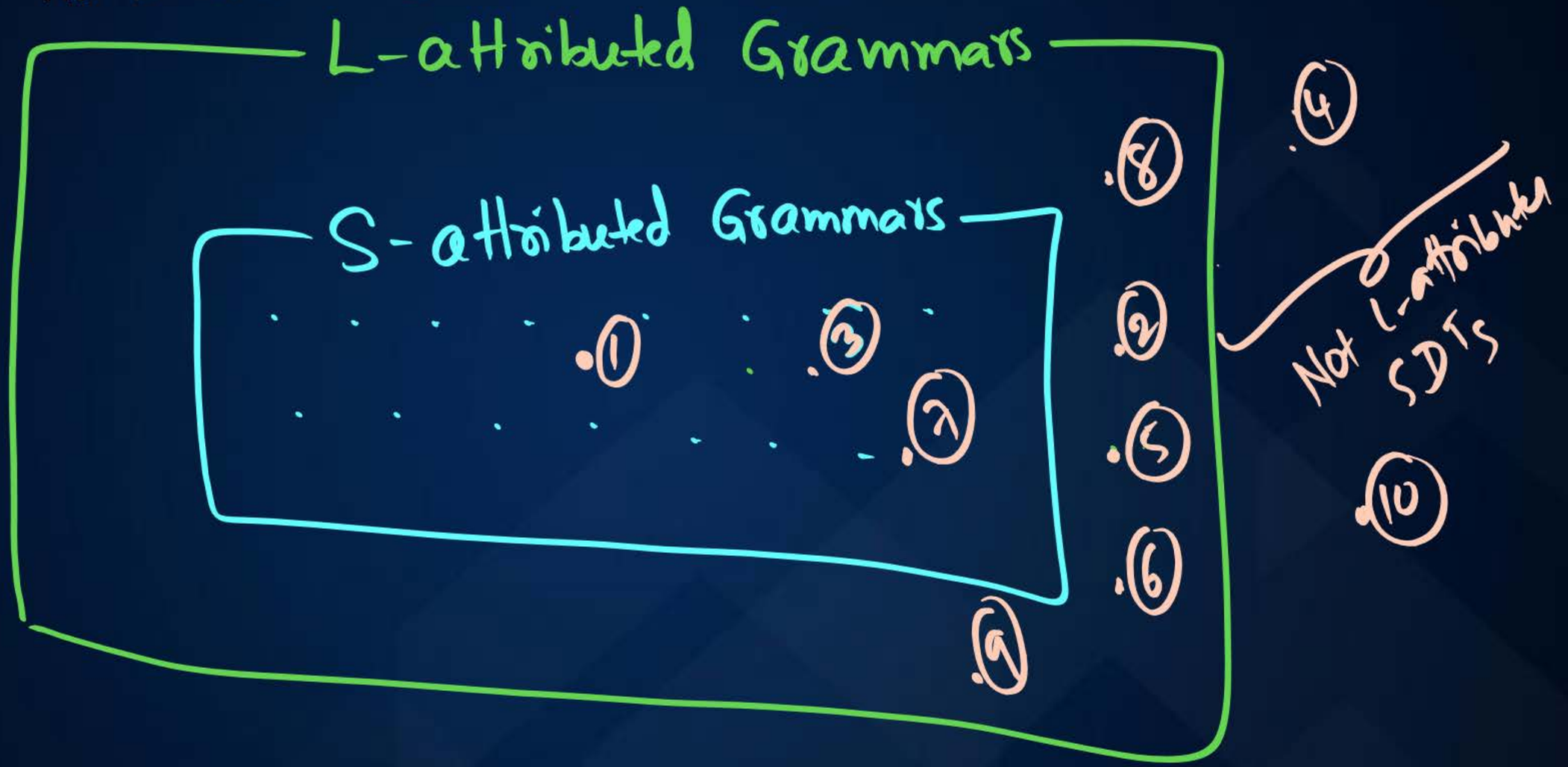
$A \rightarrow XY$

$X.i = A.i + Y.S$; (Parent, Right sibling)

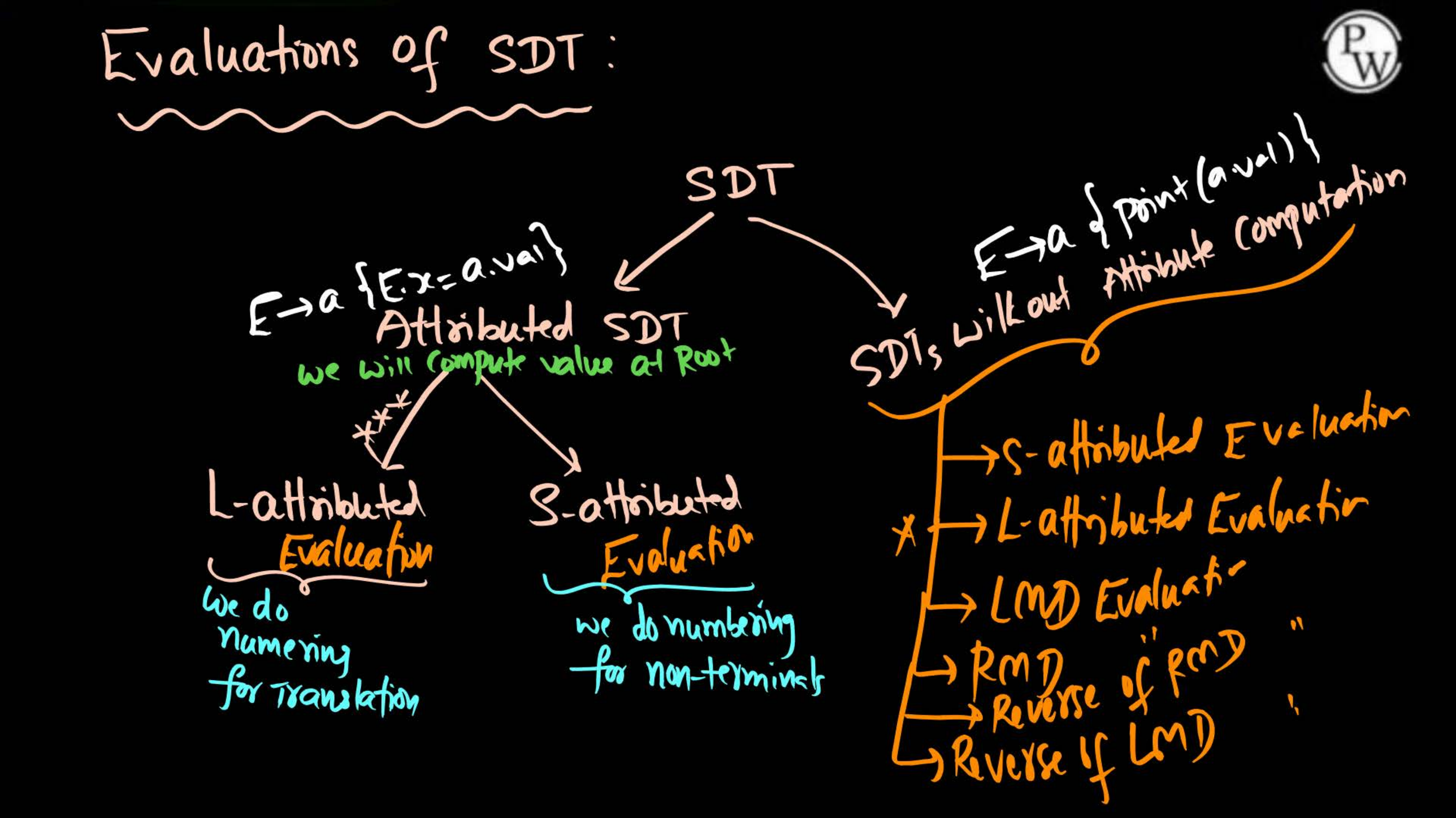
$Y.i = X.S + A.i$; (left sib, parent)

⇒ Not L-attributed (Not S-attributed)

Every S-attributed grammar is L-attributed.
Every not L-attributed grammar is not S-attributed



Evaluations of SDT :



① $E \rightarrow E_1 + T \quad \{ E.val = E_1.val * T.val \}$

$E \rightarrow a \quad \{ E.val = a.val \}$

$T \rightarrow b * T_1 \quad \{ T.val = b - T_1.val \}$

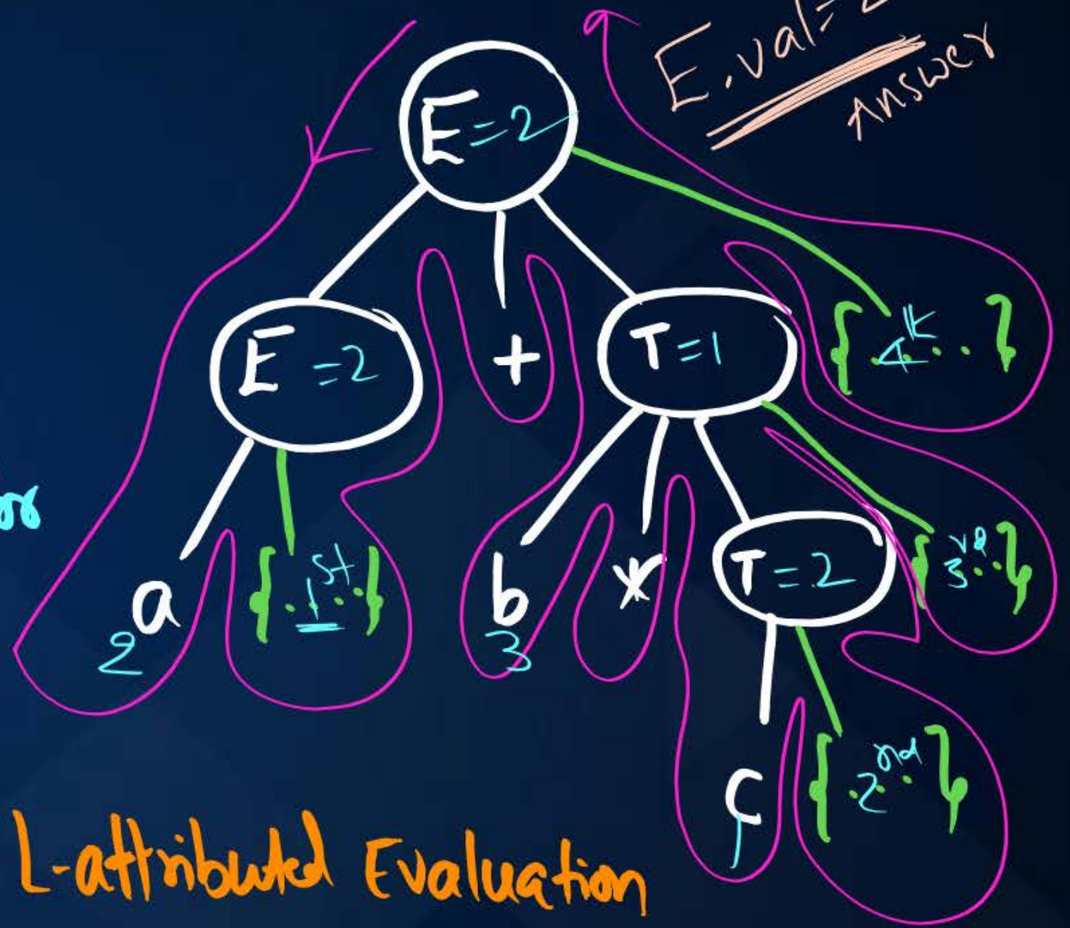
$T \rightarrow c \quad \{ T.val = c.val + 1 \}$

Find attribute value at the Root for input 2+3*1

Note: This SDT is both L-attributed & S-attributed.

⇓

Answer = 2



⇒ Using L-attributed Evaluation

Depth Left and Follow Inorder

(All Translations evaluate from Left to Right)

① $E \rightarrow E_1 + T \quad \{ E.val = E_1.val * T.val \}$

$E \rightarrow a \quad \{ E.val = a.val \}$

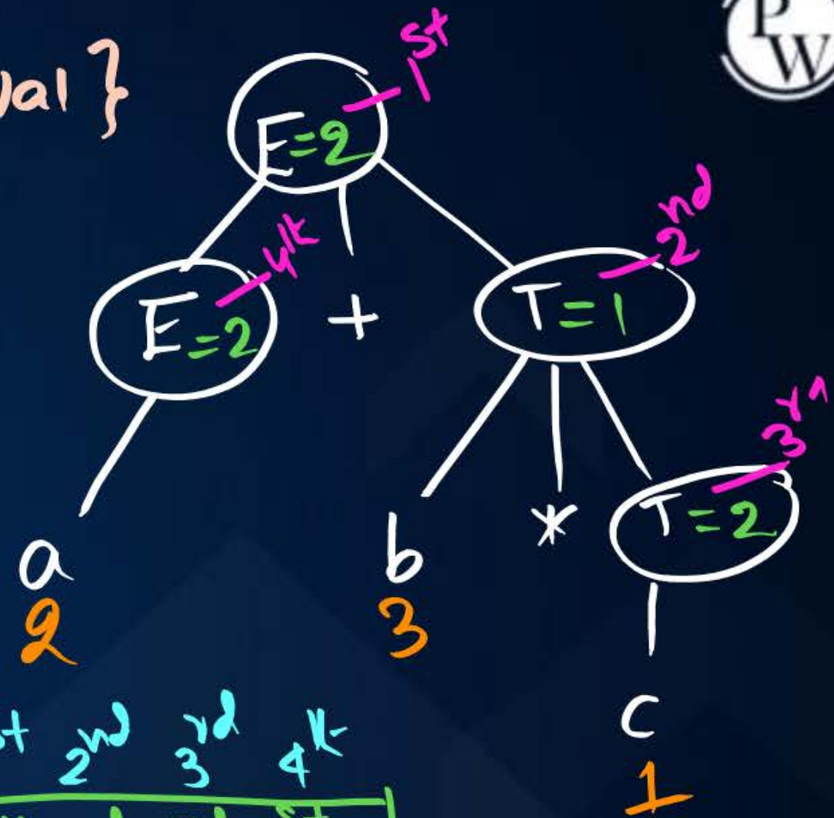
$T \rightarrow b * T_1 \quad \{ T.val = b - T_1.val \}$

$T \rightarrow c \quad \{ T.val = c.val + 1 \}$

Find attribute value at the Root for input 2+3*1 :

Note: This SDT is both L-attributed & S-attributed.

∴ Answer = 2

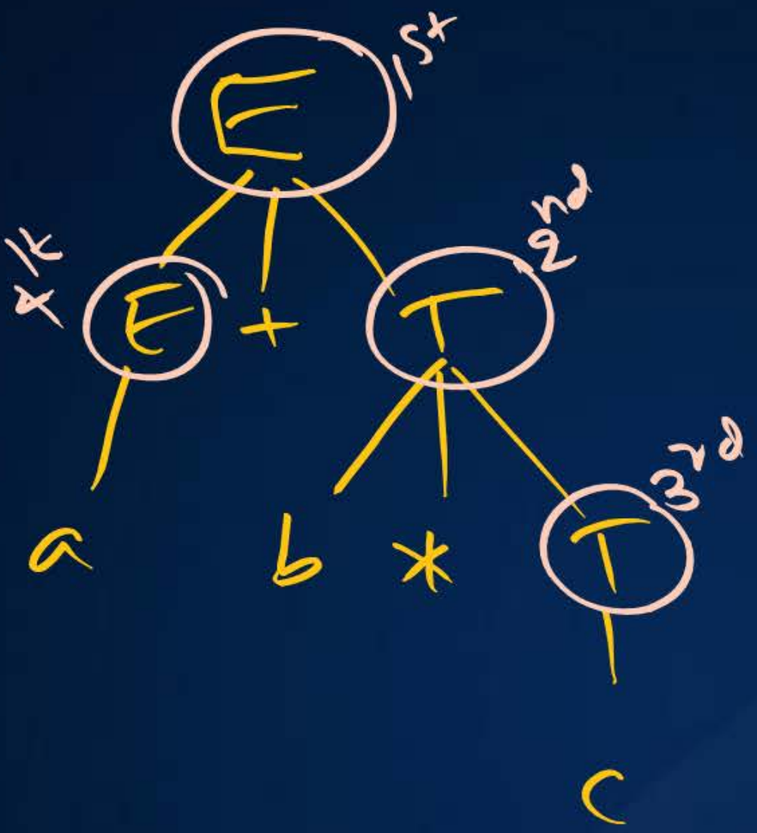


RMD order: 1st 2nd 3rd 4th

Reverse of RMD: 4th 3rd 2nd 1st

⇒ Using S-attributed Evaluation (Bottom up parsing Approach) (Reverse of RMD)

RMD numbering:



E

↓

E + T

↓

E + b * T

↓

E + b * c

↓

a + b * c

Note:

⇒ If no evaluation technique mentioned then "Use L-attributed"

(2)

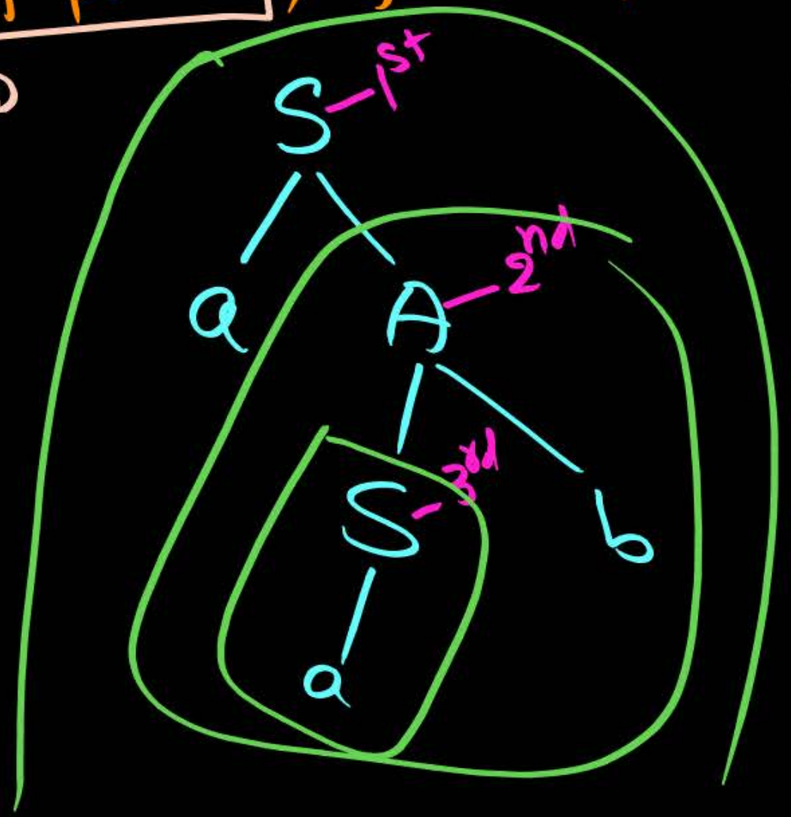
$S \rightarrow aA$ {print 1}

$S \rightarrow a$ {print 2}

$A \rightarrow Sb$ {print 3}

Using **Bottom up parser**, find o/p for "aab"

= Reverse of RMD



RMD order: 1st 2nd 3rd

Reverse of RMD: 3rd 2nd 1st

o/p: 2 3 1

(3)

$S \rightarrow aA$ {print '+'}

$S \rightarrow a$ {print '*'}

$A \rightarrow Sb$ {print '-'}

Find o/p for aab using

I) Not mentioned any evaluation

II) Using L-attributed evaluation

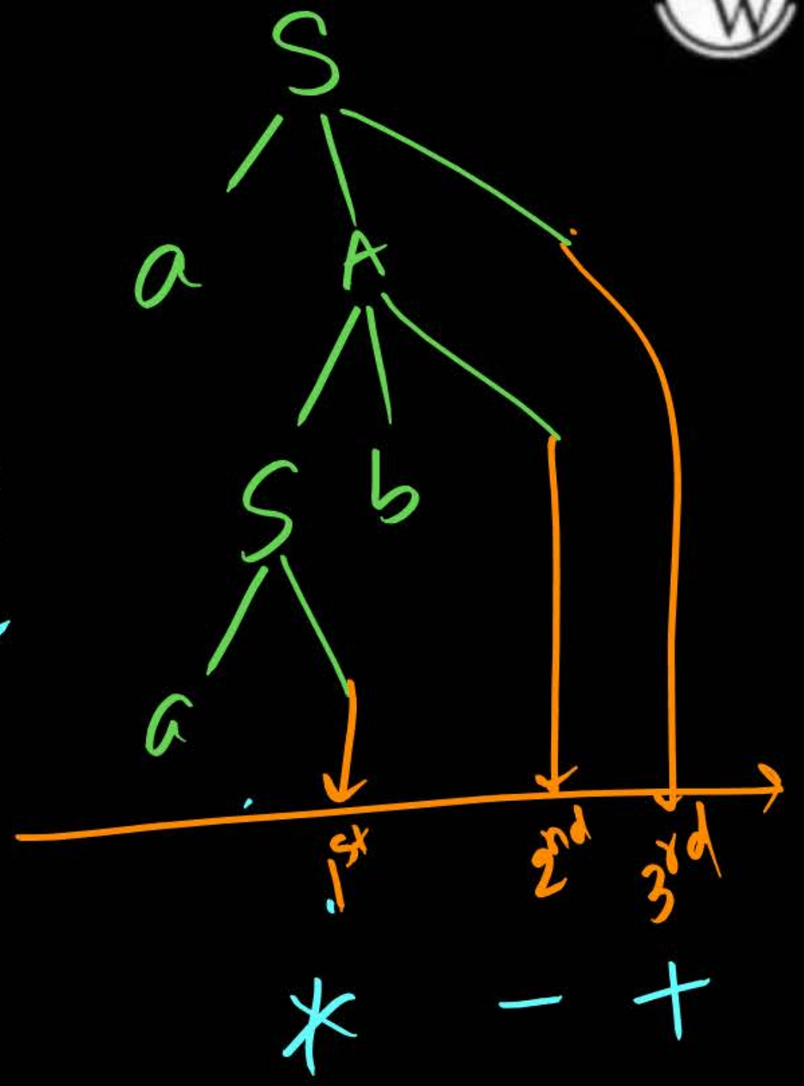
III) Using S-attributed / Bottom up parsing / LR parser / SR parser / LALR / SLR / CLR

same o/p * - +

IV) Using Top-down parsing / LL(1) / predictive parsing

V) Using RMD

VI) Using Reverse of LMD

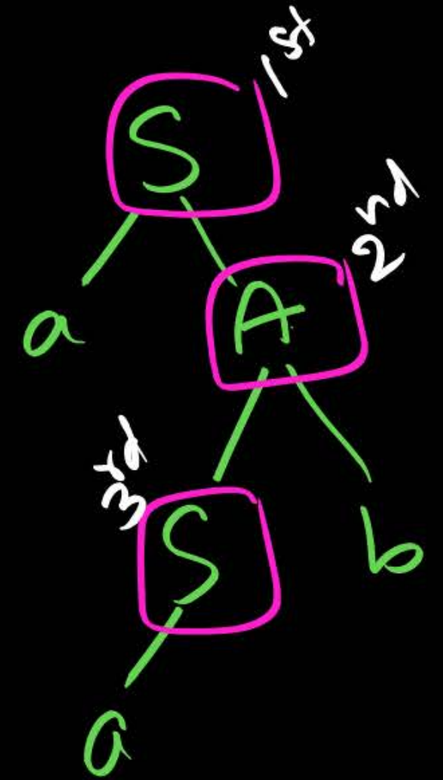


(3)

$S \to aA$ {print '+'}

$S \to a$ {print '*'}

$A \to Sb$ {print '-'}



Find o/p for **aab** using

I) Not mentioned any evaluation

II) Using L-attributed evaluation

III) Using S-attributed / Bottom up parsing / LR parser / SR parser / LALR / SLR / CLR → same o/p → * - +

IV) Using top-down parsing / LL(1) / predictive parsing

V) Using RMD ⇒ + − *

VI) Using Reverse of LMD

RMD: 1st 2nd 3rd

\+ − *

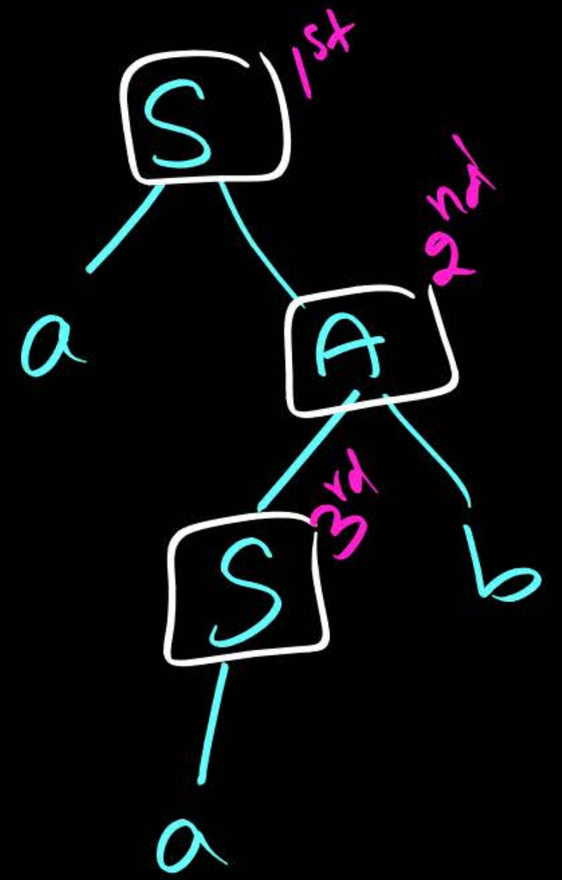
Reverse of RMD: * − +

(3)

$S \to aA$ {print '+'}

$S \to a$ {print '*'}

$A \to Sb$ {print '-'}



Find o/p for **aab** using

I) Not mentioned any evaluation.

II) Using L-attributed evaluation

III) Using S-attributed / Bottom up parsing / LR parser / SR parser / LALR / SLR / CLR

IV) Using top-down parsing / LL(1) / predictive parsing ⇒ + - *

V) Using RMD

VI) Using Reverse of LMD ⇒ * - +

LMD: 1st + 2nd - 3rd *

Reverse of LMD: * - +

(4)

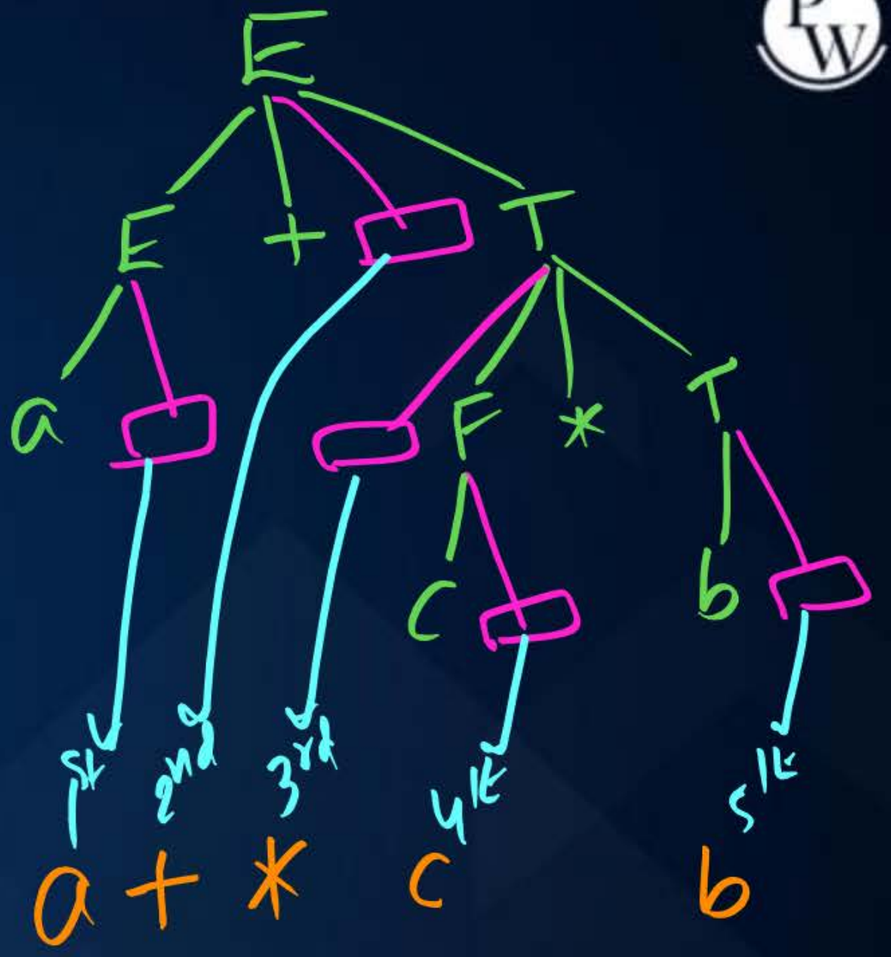
$E \rightarrow E + \{print +\} T$

$E \rightarrow a \{print\ a\}$

$T \rightarrow \{print\ *\} F * T$

$T \rightarrow b \{print\ b\}$

$F \rightarrow c \{print\ c\}$



Find o/p for Input = a+c*b using:

* I) L-attributed evaluation ⇒ a+*cb
II) Bottom up parsing
III) top-down parsing

(4)

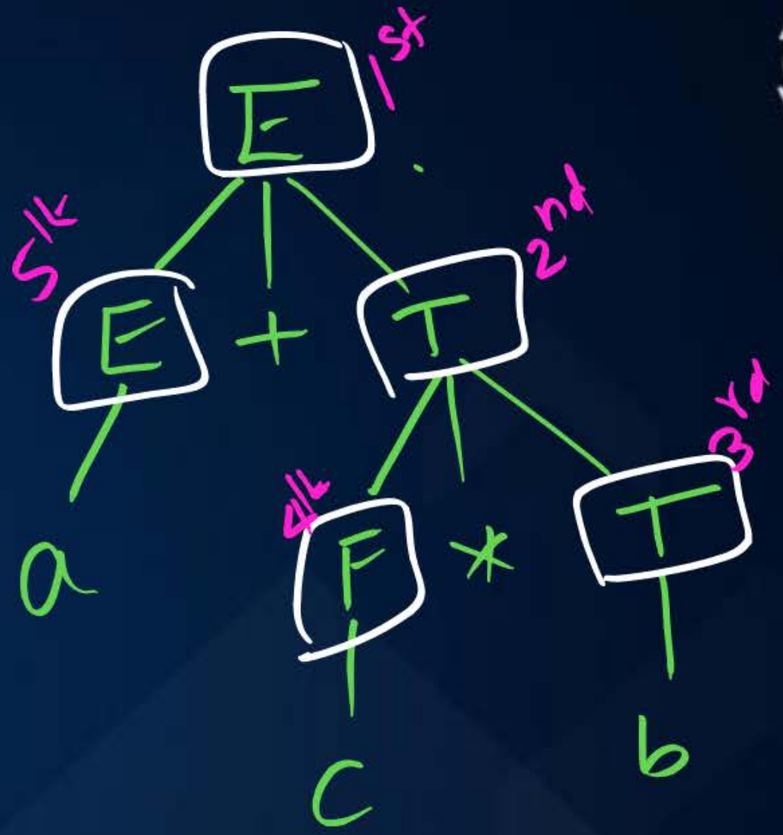
$E \rightarrow E + \{print +\} T$

$E \rightarrow a \{print\ a\}$

$T \rightarrow \{print\ *\} F * T$

$T \rightarrow b \{print\ b\}$

$F \rightarrow c \{print\ c\}$



Find o/p for Input = a+c*b. using:

* I) L-attributed evaluation
II) Bottom up parsing ⇨ acb*+
III) Top-down parsing

Reverse of RMD: 5th 4th 3rd 2nd 1st

a c b * +

(4)

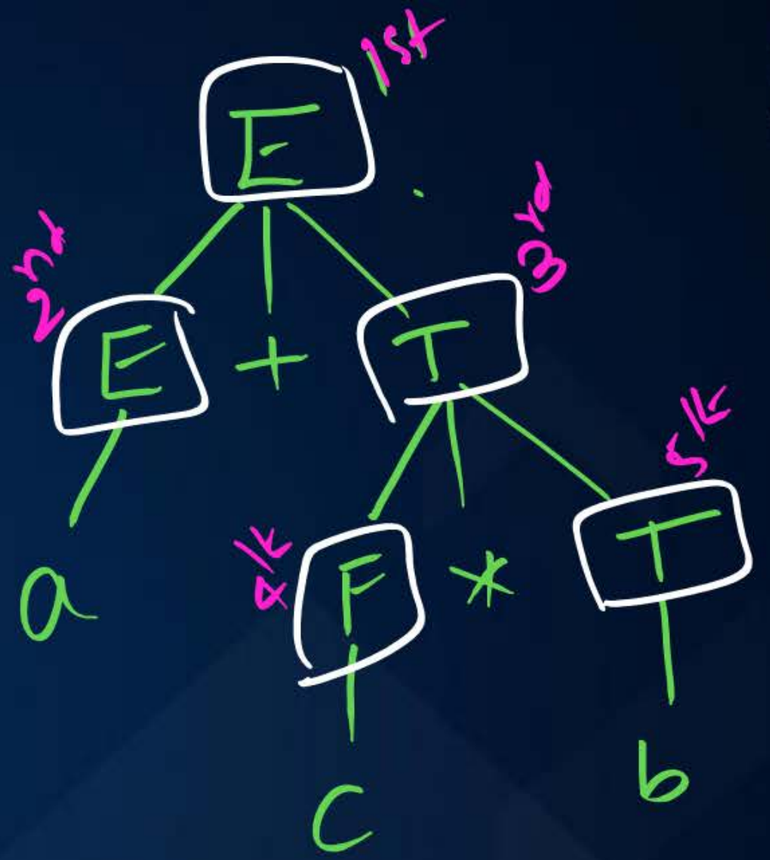
$E \rightarrow E + \{print +\} T$

$E \rightarrow a \{print\ a\}$

$T \rightarrow \{print\ *\} F * T$

$T \rightarrow b \{print\ b\}$

$F \rightarrow c \{print\ c\}$



Find o/p for Input = a+c*b. using:

* I) L-attributed evaluation
II) Bottom up parsing
III) Top-down parsing $\Rightarrow +a*cb$

LMD:

1st	2nd	3rd	4th	5th
+	a	*	c	b

$E \rightarrow \{print\ +\}\ E + T$

$E \rightarrow \{print\ a\}\ a$

$T \rightarrow \{print\ *\}\ F * T$

$T \rightarrow b\ \{print\ b\}$

$F \rightarrow c\ \{print\ c\}$

Find o/p for Input = a+c*b. using:

* I) L-attributed evaluation
II) Bottom up parsing
III) Top-down parsing

Summary

- Attributes
  - Inherited Attribute
  - synthesized ,,
- SDTs Definitions
  - L-attributed Definition
  - S-attributed Definition
- SDT Evaluations
  - Attributed ⟹ L-attributed / S-attributed
  - Without attribute Computation } or) Follow what is mentioned


THANK YOU
GW
SOLDIERS !