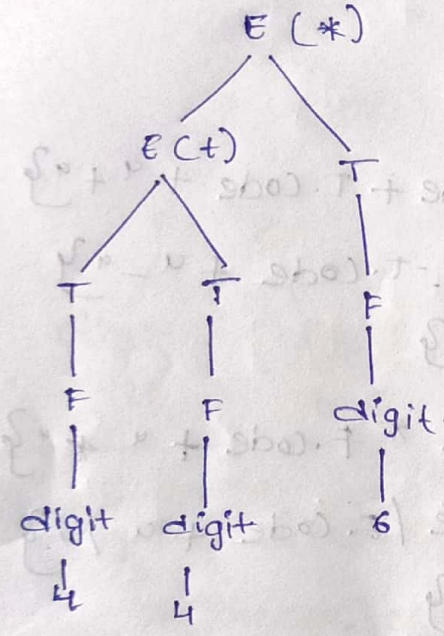
Date (01-08-23)

Analytical Question - 8

S:No 33
praveen ganesh

1. SDD for simple desk. parse tree for expression (6+4) * (4+3)n.

(A) parse Tree :-



SDD :- $S \to E$N

$E \to E+T \mid E-T \mid T$

$T \to T*F \mid T \mid F \mid F$

$F \to (E) \mid digit.$

$S \to E \{ n.val = E.val \} \{ print\ E.val \}$

$E \to E+T \{ E.val = E1.val + T.val \}$

$E \to E-T \to \{ E.val = E.val - T.val \}$

$E \to T \Rightarrow \{ E.val = T.val \}$

$T \to T*F \{ T.val = T.val * F.val \}$

$T \to T/F \{ T.val = T.val / F.val \}$

$T \to F \{ T.val = F.val \}$

$F \to (E) \{ F.val = E.val \}$

$F \to digit \{ F.val = digital \}$

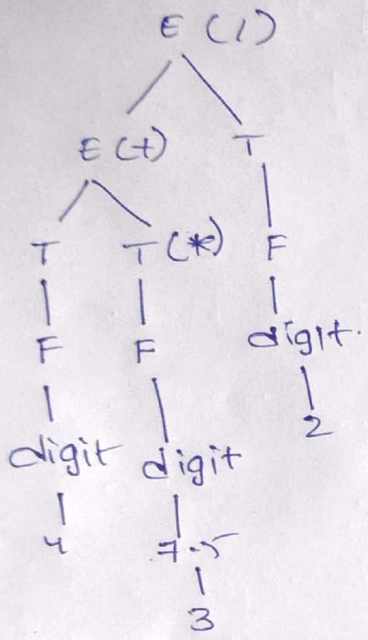
2. parse Tree according to syntax directed.

(4 + 7.5 * 3) / 2.

S:No = 33

R. praveen ganesh.

(A) parse Tree :-



1. Evaluate (7.5 * 3) = 22.5
2. Evaluate (4 + 22.5) = 26.5
3. Evaluate (26.5 / 2) = 13.25

Result = 13.25.

3. SDD for operators + and * variable.

x * (3 * x + x * x).

(A) S → E { print E.derivation }

E → E + T { E.derivation = E1.derivation + T.derivation }

E → T { E.derivation = T.derivation }

T → T * F { T.derviation = T1.derviation * F.derivation }

| F { T.derivation = F.derivation }

F → x { F.value = x.value ; F.drivation = 0 }

(E) { F.value = E.value ; F.derivation = E.derivation = 0 }

(4) Syntax - directed.

S.No = 33

R. praveenganesh.

S → E\n

E → E+T | E-T | T

T → T*F | T/F | F

F → (E) | digit

N → ;

(A) S → E { S.code = E.code }

E → E+T { E.code = E.code + T.code + "+" }

E → E-T { E.code = E.code - T.code + "-" }

E → T { E.code = T.code }

T → T*F { T.code = T.code * F.code + "*" }

T → T/F { T.code = T.code / F.code + "/" }

T → F { T.code = F.code }

F → (E) { F.code = ~~digit~~ E.code + "" }

F → digit { F.code = digit.lexeme }

N → ; { N.code = ";" }

"2 * 3 + 4 " LR parsing.

Grammar:

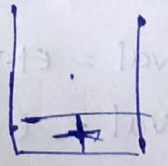
S → E\n

E → E+T | E-T | T

T → T*F | T/F | F

F → (E) | digit

N → ;



2 3 * 4 +