SDT

Advantages.

(1) SDT for evaluation of expression

$E \to E_1 + T$ { E.val = E_1.val + T.val } — (1)

$E \to T$ { E.val = T.val } — (2)

$T \to T_1 * F$ { T.val = T_1.val * F.val } — (3)

$T \to F$ { T.val = F.val } — (4)

$F \to num$ { F.val = num.lval } — (5)

input 2+3*4

$E \to E_1 + T$

$\to T + T$

$\to F + T$

$\to F + T_1 * F$

$\to F + f * f$

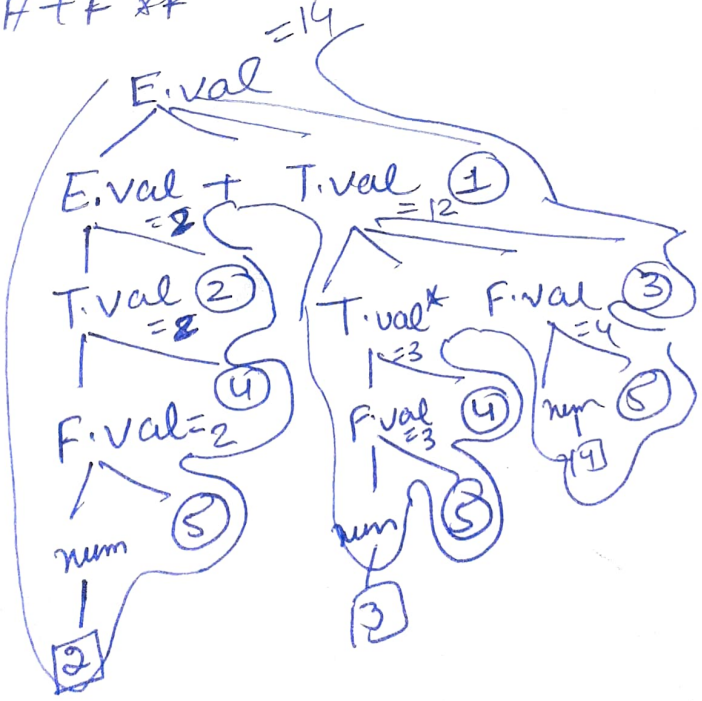
$E \to E_1 + T$

$\to E_1 + T * F$

$\to E_1 + F * F$

$\to T + f * f$

$\to f + f * f$



(2) SDT to convert infix expression to postfix expression

$E \to E+T$ { printf("+"); } — (1)

$E \to T$ { } — (2)

$T \to T*F$ { printf("*"); } — (3)

$T \to F$ { } — (4)

$F \to num$ { printf(num.lval); }

input 2 + 3 * 4

convert postfix (2 + 34 *)

(234 * +)

Top-down app.

$E \to E+T$

$\to T+T$

$\to F+T$

$\to F+T*F$

$\to F+F*F$

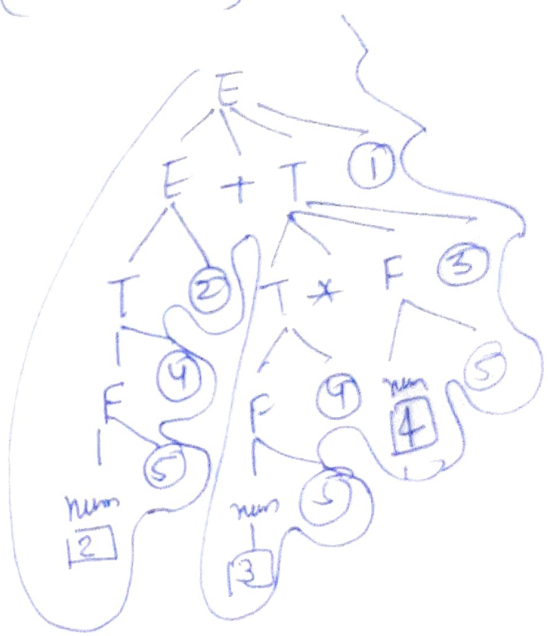
$E \to E+T$

$E+T*F$

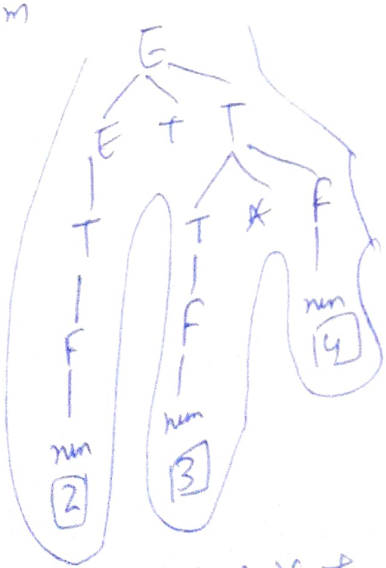
$E+F*F$

$F+F*F$

Bottom



234 * +



2 3 4 * +

(9-2)

Question what is the o/p if we carry out the following SDT to the given input string

$S \rightarrow xxW$ {print(1);} —①

$S \rightarrow y$ {print(2);} —②

$W \rightarrow Sz$ {print(3)} —③

Input string xxxxyzz

$S \rightarrow xxW$

$\rightarrow xxSz$

$\rightarrow xxxxWz$

$\rightarrow xxxxSzz$

$\rightarrow xxxxyzz$

o/p

23131

Shweta Talwar