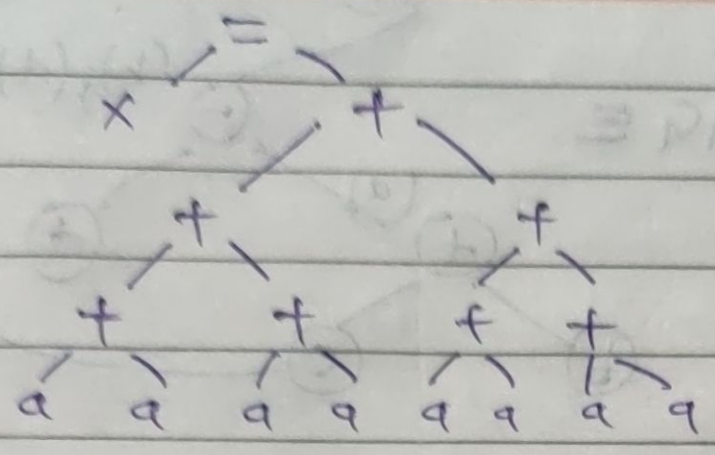
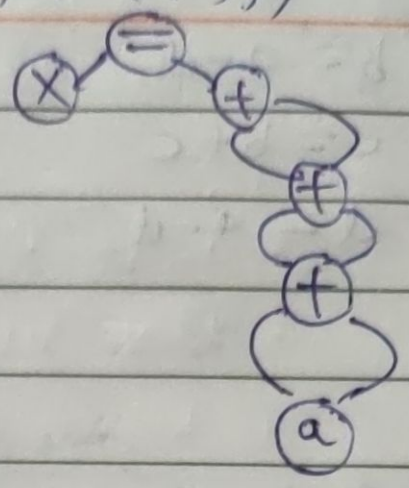
Constructing DAG

(i) $x = (((a+a)+(a+a))+((a+a)+(a+a)))$

⇒



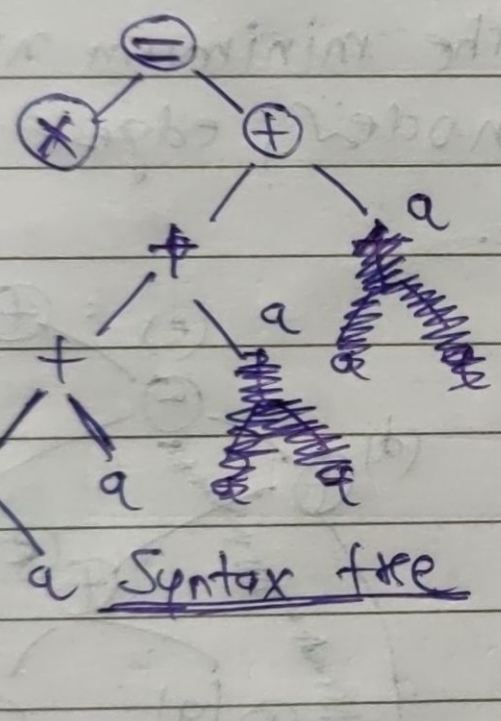
Syntax tree



DAG : 6 nodes and 8 edges

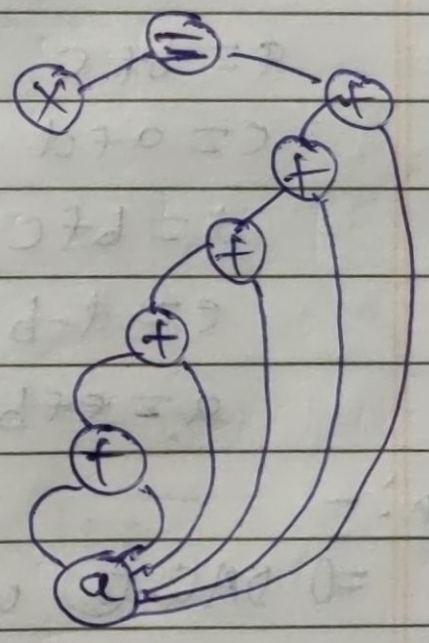
(ii) $x = (a+a+a+a+a+a)$

⇒



Syntax tree

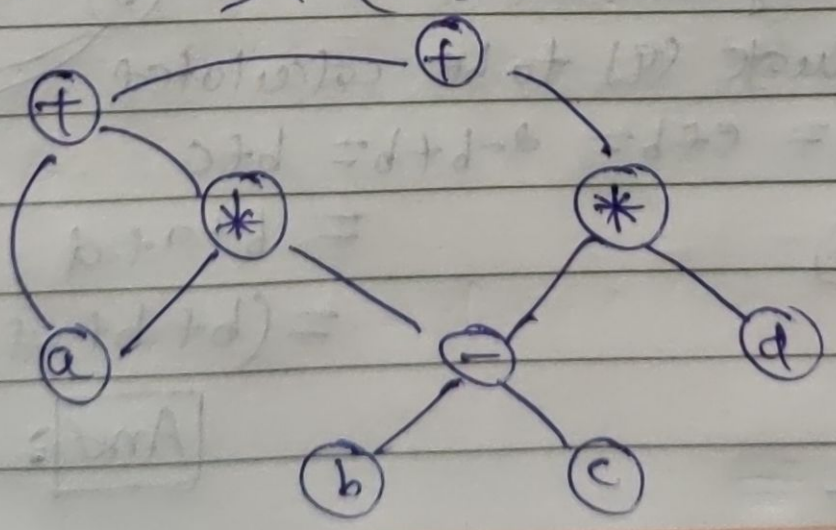
← Remember left → associativity. ∴ left recursion will be prefered.



DAG

(iii) $a + a*(b-c) + (b-c)*d$

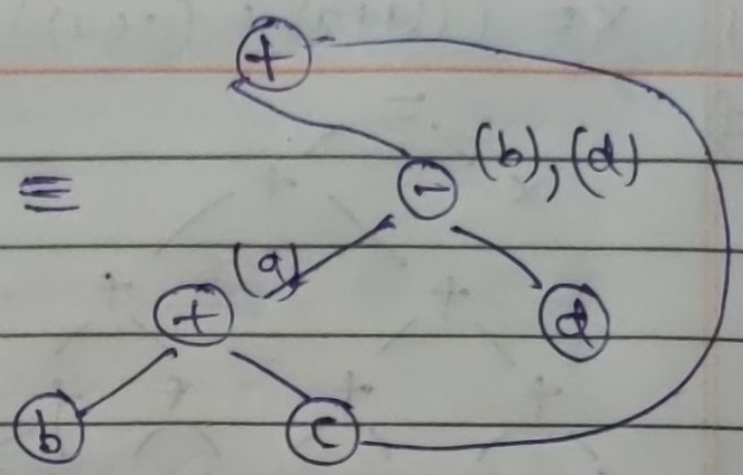
⇒



≡ DAG

(iv) $a = b + c$
$b = a - d$
$c = b + c$
$d = a - d$

DAG ≡



⇒ As (a-d) remains same, (b) and (d) are both labels for one node.

(v) $a = b + c$
$c = a + d$
$d = b + c$
$e = d - b$
$a = e + b$

Find the minimum number of nodes, edges in DAG.

Solution:-

⇒ DAG-(I) will be the original DAG.

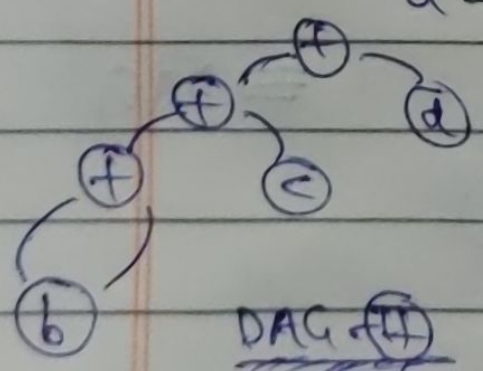
⇒ But we want the DAG having min. no of nodes and edges.

⇒ Evaluate (a) to be calculated.

$a = e + b = d - b + b = b + c$
$= b + a + d$
$= (b + b + c + d)$

Ans: 6 nodes + 6 edges

DAG-(I)



DAG-(II)

(vi)

$a = b * c$

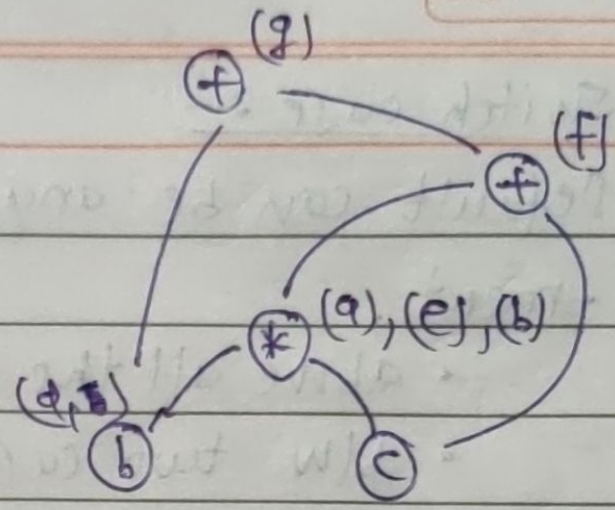
$d = b$

$e = d \times c$

$b = e$

$f = b + c$

$g = f + d$



Ans

(vii)

$T_1 = 4 * I_0$

$T_2 = a[T_1]$

$T_3 = 4 * I_0$

$T_4 = b[T_3]$

$T_5 = T_2 * T_4$

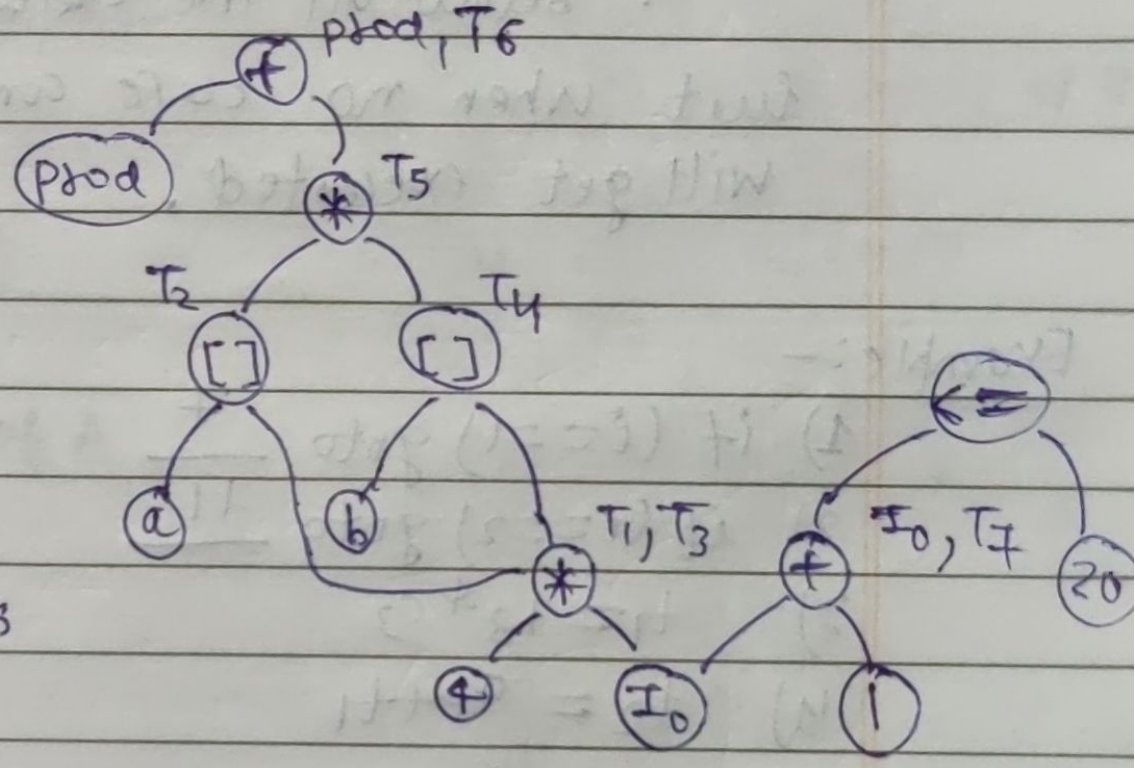
$T_6 = prod + T_5$

$prod = T_6$

$T_7 = I_0 + 1$

$I_0 = T_7$

if $I_0 <= 20$ goto 1



DAG

⇒ In DAG construction, remove as much redundancy as possible.