### Tree:

Tree is a connected graph with no cycle

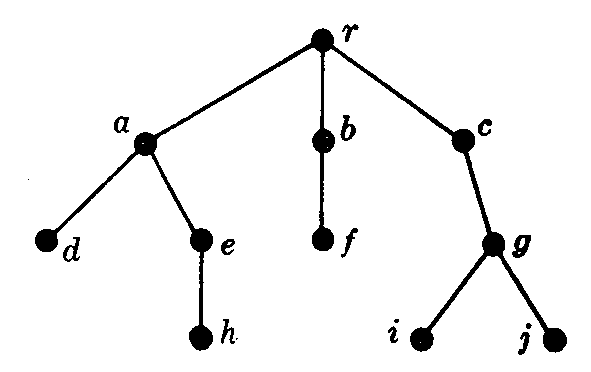
Theorem:

Let G be graph with more than one vertex. Then the following are equivalence:

* 1. G is a tree.
  2. G is cycle-free with (n-1) edges.
  3. G is connected and has (n-1) edges. (i.e: if any edge is deleted then the resulting graph is not connected)

### Rooted tree:

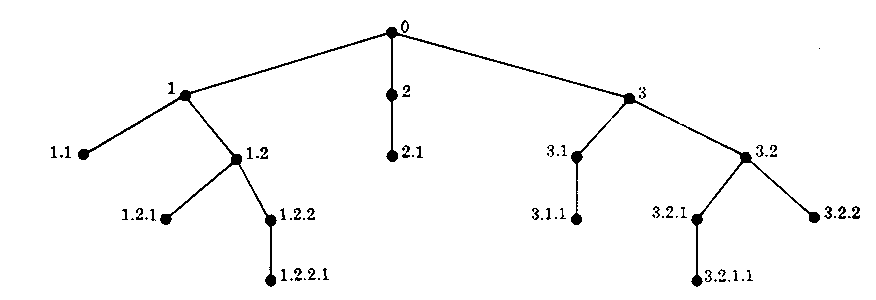
A rooted tree R consists of a tree graph together with vertex r called the root of the tree.



Height or depth: The number of levels of a tree

Leaves: The vertices of the tree that have no child (vertices with degree one)

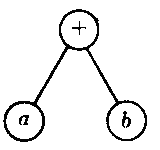
Order Rooted Tree (ORT): Whenever draw the digraph of a tree, we assume some ordering at each level, by arranging children from left to right.



Degree of tree: The largest number of children in the vertices of the tree Binary tree : every vertex has at most 2 children

Any algebraic expression involving binary operations +, -, , ÷ can be represented by an order rooted tree (ORT)

the binary rooted tree for a+b is :



The variable in the expression a & b appear as leaves and the operations appear as the other vertices.

#### Polish notation:

The polish notation form of an algebraic expression represents the expression unambiguously with out the need for parentheses

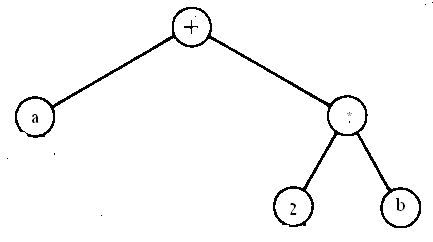
1) a + b (infix)

2) + a b (prefix)

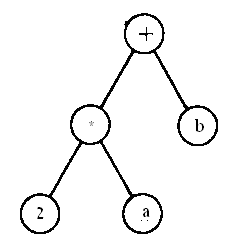
3) a b + (postfix)

example 1: infix polish notation is : a + b prefix polish notation : + a b

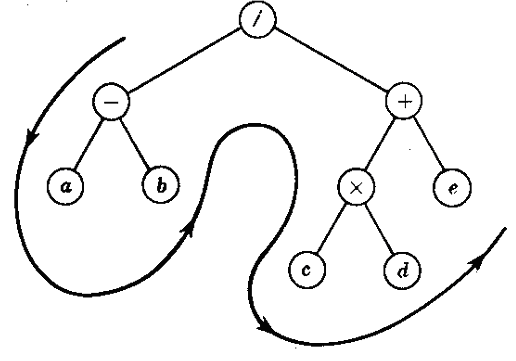
example 2: infix polish notation is : a + 2 \* b prefix polish notation : + a \* 2 b



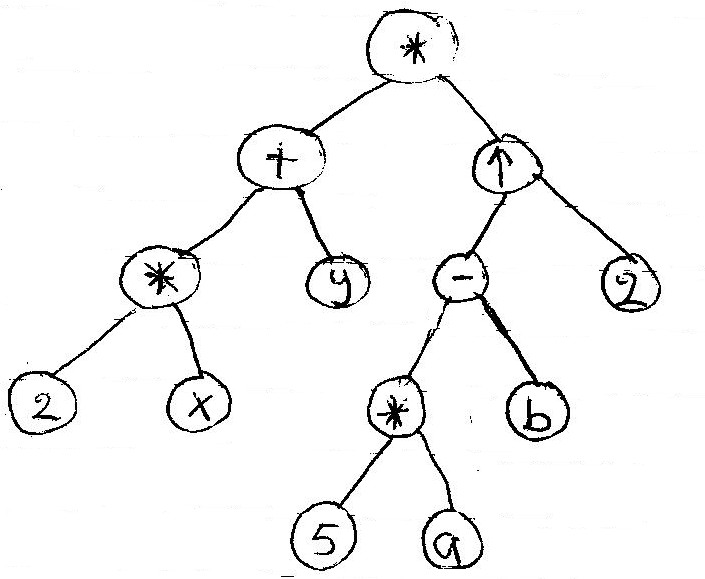
example 3: infix polish notation is : 2 \* a + b prefix polish notation : + \* 2 a b



example 4: infix polish notation is : (a – b) / (c \* d ) + e) prefix polish notation : / - a b + \* c d e



example 5: infix polish notation is : (2 \* x + y) (5 \* a – b )^2 prefix polish notation : \* + \* 2 x y ^ - \* 5 a b 2



example 6: infix polish notation is : (a + 2 \* b) ( 2 \* a + b^2) prefix polish notation : \* + a \* 2 b + \* 2 a ^ b 2

