Trees

Aim

To introduce the idea of a special kind of graph called a tree.

Learning Outcomes

At the end of this section you will:

- Know what a tree is,
- Know how to represent simple algebraic expressions using trees.

Tree: A rooted *tree* is an acyclic, connected graph with one node designated as the *root* of the tree.

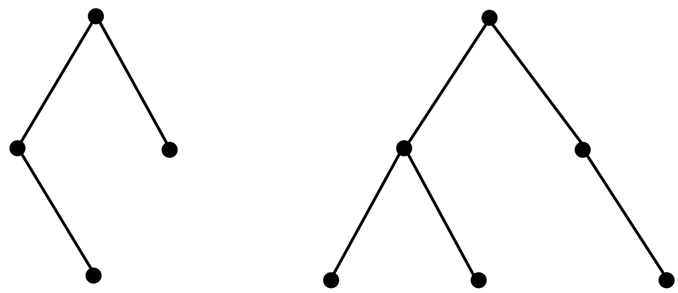

Figure 1: Two examples of trees

A tree can also be defined recursively. A single vertex(node) is a tree (with that vertex as its root). If $T_1, T_2, ..., T_t$ are disjoint trees with roots $r_1, r_2, ..., r_t$, the graph formed by attaching a new vertex r by a single edge to each $r_1, r_2, ..., r_t$ is a tree with root r. The roots $r_1, r_2, ..., r_t$ are *children* of r, and r is a *parent* of $r_1, r_2, ..., r_t$ (just like a family tree).

Depth of a node: The *depth of a node* in a tree is the length of the path from the root to the node; the root itself has depth zero.

A node with no children is called a *leaf* of the tree; all non-leaves are *internal nodes*.

Depth of a tree: The *depth(height) of a tree* is the maximum depth of any node in the tree; in other words, it is the length of the longest path from the root to any node.

Binary tree: A *binary tree* is a tree where each node has at most two children. A *full binary tree* is a tree where all nodes have exactly two children and all leaves are at the same depth.

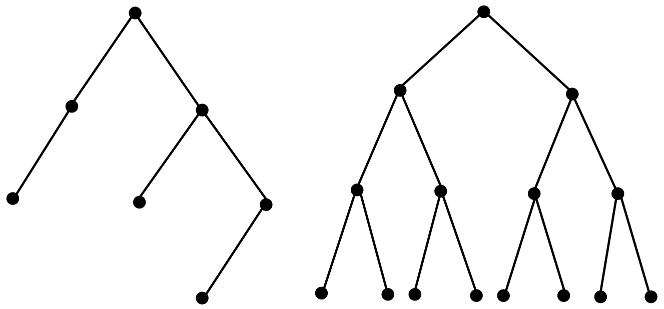

Figure 2: A binary tree on the left and a full binary tree of height 3 on the right.

A tree with n vertices has $n-1$ edges.

Algebraic expressions involving binary operations can be represented by labeled binary trees. The leaves are labeled as operands, and the internal nodes are labeled as binary operations. For any internal node, the binary operation of its label is performed on the expressions associated with its left and right subtrees. The binary tree below represents the algebraic expression $(2+x)-(3*y)$.

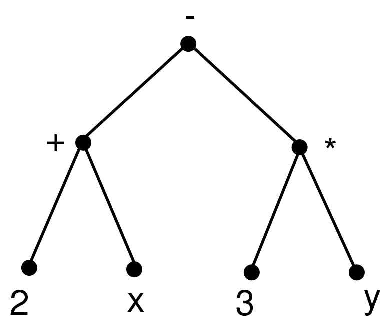

Figure 3: A tree representing $(2+x)-(3*y)$.

Related Reading

Gersting, J.L. 2007. *Mathematical Structures For Computer Science.* W.H. Freeman and Company.