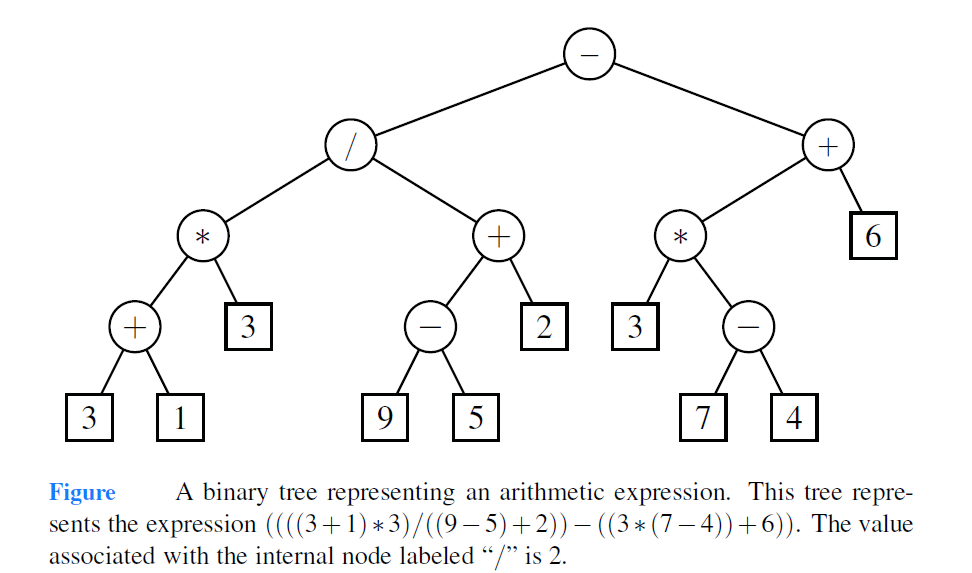
1. An arithmetic expression can be represented by a binary tree whose leaves are associated with variables or constants, and whose internal nodes are associated with one of the operators +, −, ∗, and /, as demonstrated in Figure. Each node in such a tree has a value associated with it.

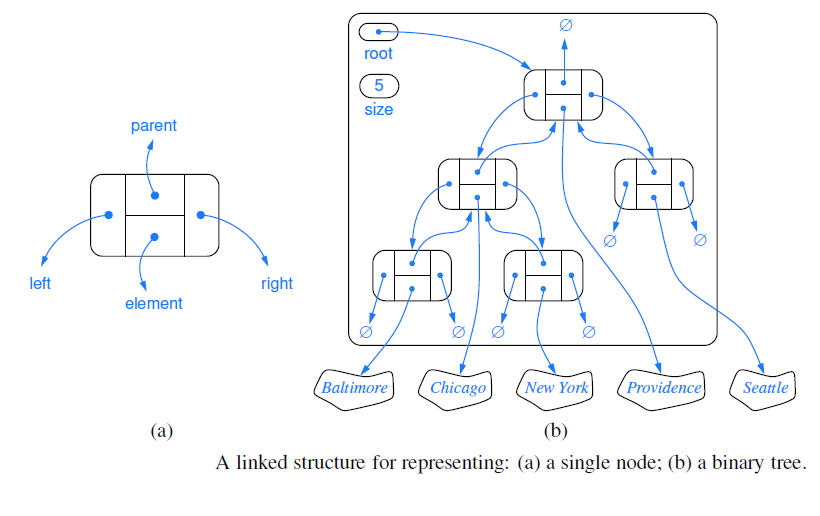
• If a node is leaf, then its value is that of its variable or constant.

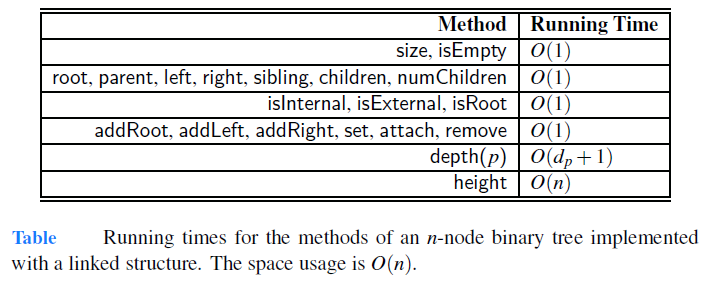
• If a node is internal, then its value is defined by applying its operation to the values of its children.

A typical arithmetic expression tree is a proper binary tree, since each operator +, −, ∗, and / takes exactly two operands. Of course, if we were to allow unary operators, like negation (−), as in “−*x*,” then we could have an improper binary tree.











An alternative representation of a binary tree *T* is based on a way of numbering the positions of *T*. For every position *p* of *T*, let *f* (*p*) be the integer defined as follows.



• If *p* is the root of *T*, then *f* (*p*) = 0.

• If *p* is the left child of position *q*, then *f* (*p*) = 2 *f* (*q*)+1.

• If *p* is the right child of position *q*, then *f* (*p*) = 2 *f* (*q*)+2.