

## Abstract syntax tree(AST)

Abstract Syntax Tree (AST) is a tree representation of the source code. Since the tree describes the hierarchy of each node, it makes simple to understand the relation between each node. For example, we have two mathematical expressions: 1)  $10 - [(5 / 4) + 1]$  and 2)  $[10 - (5 / 4)] + 1$ . Both expressions have the same numbers, but the way of using parentheses and brackets are different. Therefore, figure 1 shows the syntax trees generated by both expressions (left side — expression 1, and right side — expression 2).

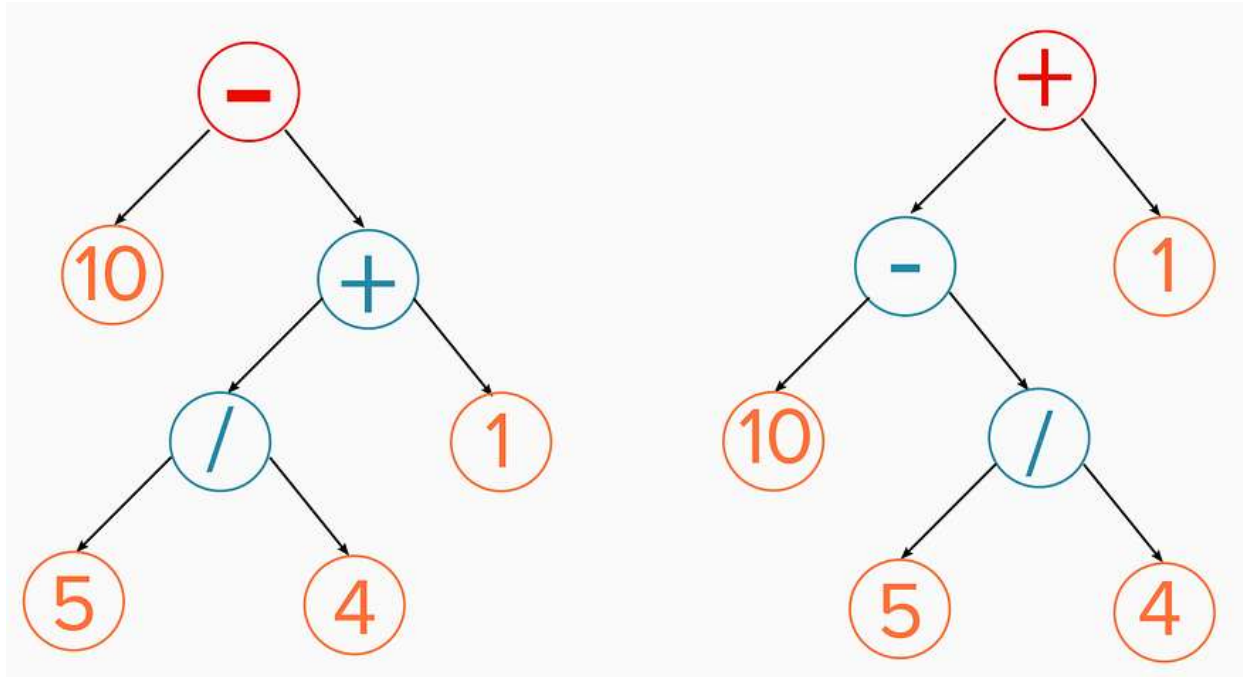


Figure 1. Syntax tree of expression 1 and expression 2, respectively.

The syntax tree is composed by nodes and edges, formally we can denote it as  $T = \langle N, E \rangle$ , where  $N$  is the list of nodes, and  $E$  is the list of edges. The nodes contain the data, and the edges indicate the hierarchy among the nodes. In the nodes of a tree we can identify the root node (red node in figure 1), and the leaf nodes (orange outline nodes). The root node has the highest hierarchy in the tree, and the leaf nodes are the last nodes in the tree.

### References: -

- [https://medium.com/@jessica\\_lopez/basic-understanding-of-abstract-syntax-tree-ast-d40ff911c3bf#:~:text=Abstract%20Syntax%20Tree%20\(AST\)%20is,5%20%2F%204\)%5D%20%2B%201.](https://medium.com/@jessica_lopez/basic-understanding-of-abstract-syntax-tree-ast-d40ff911c3bf#:~:text=Abstract%20Syntax%20Tree%20(AST)%20is,5%20%2F%204)%5D%20%2B%201.)