

Tree Traversal Algorithms

Among common traversal methods are:

Post-order

Pre-Order

In-Order

Level-Order

Polish notation – Pre-order

Infix notation – In-order

reverse Polish notation – Post-order

Reverse Polish Notation

Reverse Polish Notation is a way of expressing arithmetic expressions that avoids the use of brackets to define priorities for evaluation of operators. In ordinary notation, one might write

$$(3 + 5) * (7 - 2)$$

and the brackets tell us that we have to add 3 to 5, then subtract 2 from 7, and multiply the two results together. In RPN, the numbers and operators are listed one after another, and an operator always acts on the most recent numbers in the list. The numbers can be thought of as forming a stack, like a pile of plates. The most recent number goes on the top of the stack. An operator takes the appropriate number of arguments from the top of the stack and replaces them by the result of the operation.

In this notation the above expression would be

$$3 5 + 7 2 - *$$

Reading from left to right, this is interpreted as follows:

- Push 3 onto the stack.
- Push 5 onto the stack. The stack now contains (3, 5).
- Apply the + operation: take the top two numbers off the stack, add them together, and put the result back on the stack. The stack now contains just the number 8.
- Push 7 onto the stack.
- Push 2 onto the stack. It now contains (8, 7, 2).
- Apply the - operation: take the top two numbers off the stack, subtract the top one from the one below, and put the result back on the stack. The stack now contains (8, 5).
- Apply the * operation: take the top two numbers off the stack, multiply them together, and put the result back on the stack. The stack now contains just the number 40.

Polish Notation was devised by the Polish philosopher and mathematician Jan Lucasiewicz (1878-1956) for use in symbolic logic. In his notation, the operators preceded their arguments, so that the expression above would be written as

$$* + 3 5 - 7 2$$

The 'reversed' form has however been found more convenient from a computational point of view.