Rearranging the nodes

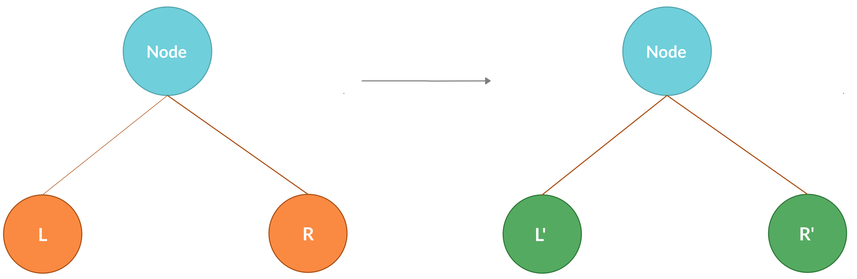
Base Case:

(No children)

Legend:

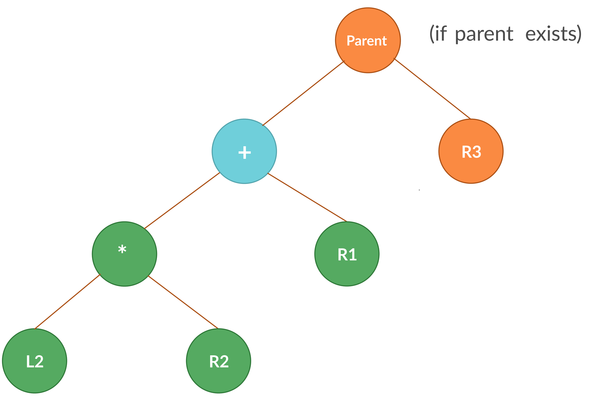
Step 1:

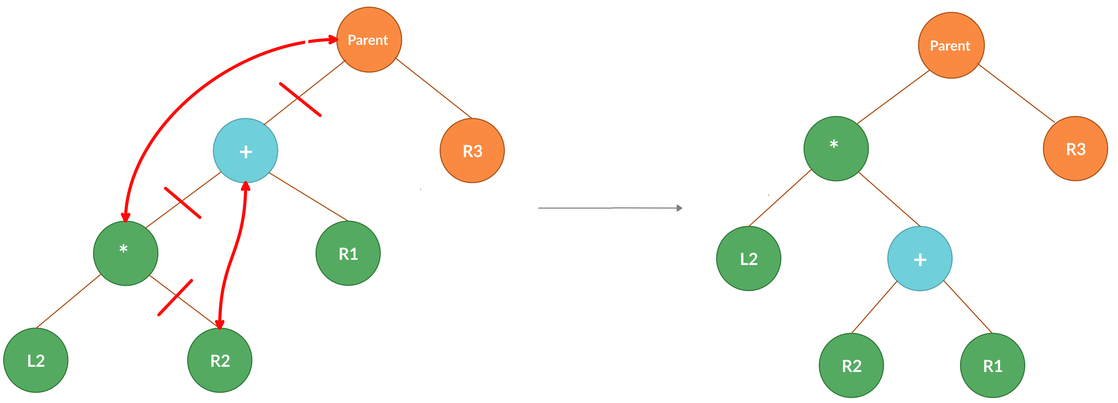
Rearrange the subtrees rooted at the children of the current node. The resulting subtrees may or may not be the same as before.



Step 2:

Get the **left** child of the current node (the left child may be different after Step 1). If the value of the current node is '+' and the value of the left child is '*', that means that the multiplication will be evaluated before the addition. Rearrange the nodes so that addition happens before multiplication. This check does not have to be performed for the right child since a right child is only created for a number or an expression within parentheses (which has to be evaluated first anyways).





This preserves the ordering of the elements in the expression and allows addition to happen before multiplication. The sub-expression is L2 * R2 + R1. Without rearrangement, L2 * R2 gets evaluated first. After the rearrangement, R2 + R1 gets evaluated first.

If the current node is a right child of the parent, the rearrangement changes by a little.