Section V.9: Recursive Tree Traversals, Verification by Induction

In data structures courses, we learn three basic traversals of a binary tree $T$. These are called PreOrder, InOrder, and PostOrder. They are very easy to code using recursion, and structural induction is a straightforward way to verify correctness of the recursive code.

Recursion is done on subtrees $T'$ of $T$, including the empty subtree. We let $N$ (node) denote the root of a subtree of $T'$, $L$ be the subtree of $T'$ whose root is the left child of $N$, and $R$ be the subtree of $T'$ whose root is the right child of $N$. We call $L$ the left subtree of $T'$ and $R$ the right subtree of $T'$.

PreOrder, InOrder, and PostOrder are denoted NLR, LNR, and LRN respectively, according as to whether the root ($N$) is visited first (Pre), in the middle (In) or last (Post). All three of these traversals are done recursively on the subtrees of $T$. The base case of the recursion addresses the empty subtree, for which nothing is done.

What follows is the recursive pseudocode for PreOrder, and verification of its correctness by structural induction. Similar pseudocode and proof of correctness can be done for InOrder and PostOrder.

**Algorithm PreOrder**

**Input**: $T$, a binary tree [could be empty]

**Output**: The contents of the nodes of $T$, displayed in PreOrder (which will be nothing if $T$ is empty).

**Algorithm Body:**

if $T$ is not empty then
    display the contents of the root $N$ of $T$;
    call PreOrder for the left subtree $L$ of $T$;
    call PreOrder for the right subtree $R$ of $T$
end Algorithm PreOrder

**Structural Induction Proof** that the pseudocode for PreOrder is correct:

**Proof**: We must show (1) PreOrder is correct for the empty tree and (2) if PreOrder is correct for all proper subtrees of $T$, then PreOrder is correct for tree $T$.

(1) Since the empty tree has no nodes, nothing can be displayed. Nothing is done in the pseudocode for the empty tree. Therefore, PreOrder is correct in this case.

(2) Let $T$ not be empty. Then $T$ has a root $N$, and the contents of $N$ are displayed first. By the induction assumption, the left subtree $L$ is traversed in PreOrder, i.e., in the order root node, left subtree, and right subtree. This is done at all levels. Also, by the induction assumption, the right subtree $R$ is traversed in PreOrder, i.e., in the order root node, left subtree, and right subtree. This is done at all levels. Therefore, for the entire tree $T$, the traversal is root node $N$, left subtree $L$, and right subtree $R$ at all levels. Therefore, the pseudocode is correct.
Example V.9.1: Trace the recursive PreOrder traversal for tree below.

Solution:

The tree is non-empty.

So we display 5, the contents of the root.
We call PreOrder on the left subtree, whose root is 7.
  We display 7, the content of the root.
  We call PreOrder on the left subtree, which is empty.
  We call PreOrder on the right subtree, whose root is 3
    We display 3, the content of the root.
    We call PreOrder on the left subtree, which is empty.
    We call PreOrder on the right subtree, which is empty.
We call PreOrder on the right subtree, whose root is 1.
  We display 1, the content of the root.
  We call PreOrder on the left subtree, whose root is 4
    We display 4, the content of the root.
    We call PreOrder on the left subtree, which is empty.
    We call PreOrder on the right subtree, which is empty.
  We call PreOrder on the right subtree, whose root is 8.
    We display 8, the content of the root.
    We call PreOrder on the left subtree, which is empty.
    We call PreOrder on the right subtree, which is empty.
We call PreOrder on the right subtree, which is empty.

What is displayed is the ordering 5, 7, 3, 1, 4, 8.
Exercises:

(1)  (a) Develop recursive code for \textbf{InOrder} traversal of a binary tree.
    (b) Prove by structural induction that this code is correct.
    (c) Trace the InOrder traversal of the tree in Example V.9.1, using the method illustrated in this example.

(2)  (a) Develop recursive code for \textbf{PostOrder} traversal of a binary tree.
    (b) Prove by structural induction that this code is correct.
    (c) Trace the PostOrder traversal of the tree in Example V.9.1, using the method illustrated in this example.