

A TAXONOMY OF BINARY TREE TRAVERSALS

ALFS BERZTISS

Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA

Abstract.

A survey of the literature shows that eleven binary tree traversals have been defined. We systematize this work by proposing a classification that consists of twenty-six traversals grouped into seven categories. Three generator schemas are provided that allow all of the traversals to be implemented.

CR categories: G.2.2.

Keywords and phrases: Binary tree, generator, program schema, traversal, tree.

1. Introduction.

The binary tree is a most interesting data structure, both in its own right and on account of its links to directed graphs. The directed graph serves as a model for any kind of relation, but digraph algorithms are commonly expressed in terms of implicit [21] or explicit [4] directed trees, and the latter can be converted to binary trees by a Knuth transformation [12]. This has resulted in the binary tree having been extensively studied.

However, no systematic classification of binary tree traversals appears to have been undertaken. There are \( n! \) permutations of the nodes of a binary tree of \( n \) nodes, and each such permutation represents a different traversal of the binary tree, but we are only interested in traversals that can be described so that the length of the description is independent of \( n \). In Section 2 we identify 26 such traversals, and group them into seven classes. In Section 3 the literature on binary tree traversals is surveyed. We find that eleven kinds of traversals of binary trees are either recorded in the literature, or they represent the Knuth transform equivalents of general trees that are mentioned in the literature. These eleven kinds of traversals belong to six of our seven classes.

Section 4 contains three traversal schemas that suffice to specify programs for all our traversals. Generators, such as are provided by Icon [9] are a good implementation tool for the traversals. In Section 5 we discuss the implications of our observations on the specification of data structures.

Received April 1986.
2. Traversals of trees and binary trees.

A rooted directed tree in which no more than two arcs originate from a node is a binary tree. Whenever two arcs originate from a node, one of the arcs goes to the left and the other to the right. Moreover, in a binary tree, even if just one arc originates from a node, a left or right orientation is still imposed on this arc.

Every rooted directed tree can be transformed into a binary tree by means of the following procedure, which has become known as the Knuth transformation [12]:

Denote the arcs originating at internal node $x$ in a tree in left to right order by $(x, y_1), (x, y_2), \ldots, (x, y_t)$. Assign left orientation to $(x, y_1)$, and replace the remaining arcs by $(y_1, y_2), (y_2, y_3), \ldots, (y_{t-1}, y_t)$, all with right orientation. After this has been done for all internal nodes of the tree, the result is a binary tree that contains the same number of arcs as the original tree.

Figure 1 shows a tree and Figure 2 its Knuth transform.

![Fig. 1. Rooted tree.](image1)

![Fig. 2. Knuth transform of 1.](image2)

The three classical disciplines of traversal of binary trees – **preorder**, **inorder** (or **symmetric order**), **postorder** – are well known and well understood. For the binary tree of Figure 2 they generate the following sequences:

- **pre**
  
  \[
  1 \ 2 \ 6 \ 7 \ 3 \ 8 \ 4 \ 5 \ 9 \ 10 \ 12 \ 11
  \]

- **in**
  
  \[
  6 \ 7 \ 2 \ 8 \ 3 \ 4 \ 9 \ 12 \ 10 \ 11 \ 5 \ 1
  \]

- **post**
  
  \[
  7 \ 6 \ 8 \ 12 \ 11 \ 10 \ 9 \ 5 \ 4 \ 3 \ 2 \ 1
  \]