

Chapter 10: Elementary Data Structures

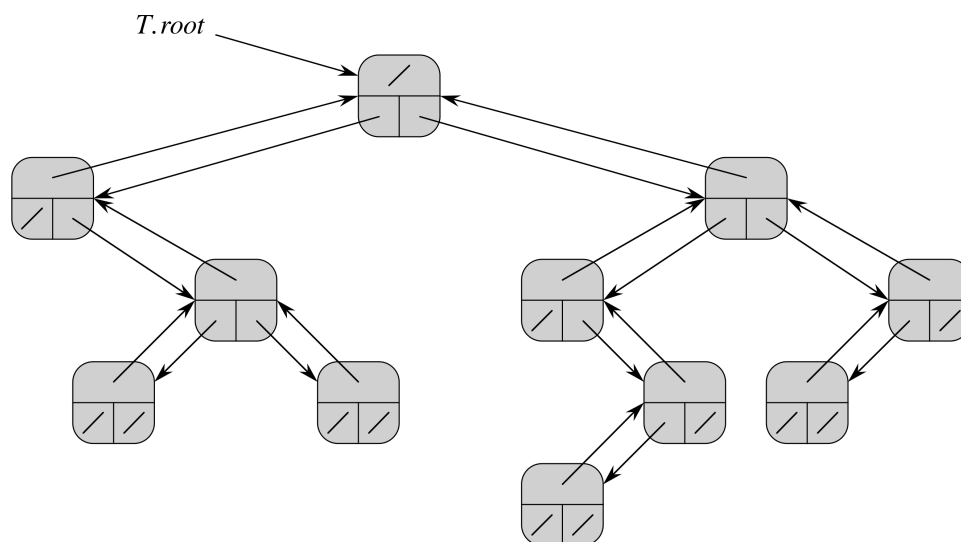
Section 10.4: Representing Rooted Trees

- Trees are composed by tree nodes.
- Each tree node has a key field and some other pointer fields pointing to other nodes. Number of pointer fields in a tree node may be different for different types of trees.
- A tree T has an attribute $\text{root}[T]$: a pointer to the root of the tree.

Binary Trees

For each node x , there are 3 pointer fields and one data field

- $x.p$ is a pointer to x 's parent.
- $x.\text{left}$ is a pointer to x 's left child.
- $x.\text{right}$ is a pointer to x 's right child.
- $x.\text{data}$ is a pointer to x 's satellite data



Rooted Tree with Bounded Branches

We can represent a general tree that has a bounded number of branches by using an array of pointers. Let the bound be r . A node in such a tree will have the following fields:

- $x.p$ is a pointer to x 's parent.
- $x.child[1..r]$ is a pointer to x 's children, up to r of them.
- $x.data$ is a pointer to x 's satellite data

In general, this would be space inefficient as most of the child pointers will be NIL. But it provides an easy and fast way to access the i child of any node.

Rooted Trees with Unbounded Branches

- Each node can have any number of children.
- Using left-child, right-sibling representation allows us to represent an arbitrary tree using only three pointers per node.
- 3 pointer fields for each node x .
 - $p[x]$ is a pointer to x 's parent.
 - $left[x]$ is a pointer to x 's left-most child.
 - $right[x]$ is a pointer to the sibling of x immediately to the right.

