Trees

Burton Rosenberg
Dept. of Math and Comp. Sci.
University of Miami

Let N be a finite set of elements called *nodes*, let $\perp$ be an element not in N. The element $\perp$ is called *bottom* by logicians, the *empty element* by mathematicians, and *null* by computer scientists.

Definition 1 (Tree) *A tree is the collection,*

$$(N, \lambda, \rho : N \to N \cup \{\perp\})$$

which satisfies the Unique Path Property.

STANDARD TERMINOLOGY: Given $n, m \in N$, if $\lambda(n) = m$, then m is said to be the *left child* of n; if $\rho(n) = m$, then m is said to be the *right child* of n. We shall see that it is not possible that for $n, n', m \in N$ that $\lambda(n) = \rho(n') = m$. That is, a node is either a left or a right child, not both. Also, there is exactly one node which is not a child of any $n \in N$. This node is called the *root*. A node $n \in N$ such that $\lambda(n) = \rho(n) = \perp$ is called a *leaf* of the tree. A leaf is a node without children.

Definition 2 (Path) *A path is a relation $\to^*$ between nodes defined as follows:*

- $n \to^0 m \iff n = m$. *This is called the empty path. It has length 0.*
- $n \to^1 m \iff \lambda(n) = m \vee \rho(n) = m$. *This path has length 1. It is also written without superscript: $n \to m$.*
- $n \to^i m \iff n \to n' \to^{i-1} m$ *for $i > 1$. This inductively defines paths of length i.*
- $\to^* = \cup_{i \geq 0} \to^i$. *That is, $\to^*$ is the transitive closure of $\to$.*

Definition 3 (Unique Path Property) *The structure (N, λ, ρ) has the unique path property if,*

$$\exists r \in N : \forall n \in N \; \exists! r \to^* n$$

This r is called a root of the tree.

Lemma 1 (Unique Root) *There is only one root in a tree.*

PROOF: Else, for two distinct $r, r' \in N$ we have booth $r \to^* r'$ and $r' \to^* r$. Hence, the path from r to r is not unique: there is the empty path and the path $r \to^* r' \to^* r$.

Lemma 2 (Unique Parents) *For any $n \in N$ with $n \neq r$ there is exactly one m such that $m \to n$. This m is said to be the parent of n. The root has no parent.*

PROOF: Distinct parents $m, m' \in N$ exist such that $m \to n$ and $m' \to n$ would imply distinct paths $r \to^* m \to n$ and $r \to^* m' \to n$. Hence any node has at most one parent. By the unique path property, if n is not the root, the path from r to n is not empty, hence there is a parent for n. However, if $m \to r$ for any $m \neq r$, then the loop $r \to^* m \to r$ would show multiple paths from r to r.

STANDARD TERMINOLOGY: Often a tree is defined as a *connected acyclic graph*. A graph is a node set and a set of edges each of which connect a pair of nodes. In our development, $n \to m$ means that the pair (n, m) is an edge. A path in a graph is a sequence of edges of the form $(n_0, n_1), (n_1, n_2), \ldots, (n_{k-1}, n_k)$. Our paths are directed versions of these paths. We are required to walk our paths consitently from parent to child. A cycle in a graph is a path which begins and ends at the same node, that is, $n_0 = n_k$, in the notation of the example path directly above. An acyclic graph is a graph without cycles. A graph is connected if there is at least one path between any two nodes in the graph.

This is equivalent to our defintion. Given a cycle in a graph which comes from our tree, assign the proper "arrows" by walking around the cycle. The Unique Parents lemma assures that either a clockwise or counterclockwise promenade will lead us fully around the cycle. Now select a node on the cycle and demonstrate mutiple paths from the root to this node, by appending to the shortest path from the root to the node additional journeys around the cycle.

Definition 4 (Descendant, Ancestor) *Given $n, m \in N$, $n \to^* m$ then m is said to be a descendant of n, and n an ancestor of m.*

Lemma 3 (Partial Order) *The relationship of descendant is a partial order.*

PROOF: The three properties of partial orders must be verified:

1. For any $n \in N$, $n \to^* n$ by definition.

2. For any $n, m \in N$, $n \to^* m$ and $m \to^* n$ implies if $n = m$. If not, there are at least two paths from the root r to n, $r \to^* n$ and $r \to^* n \to^* m \to^* n$.

3. If $n \to^* m$, and $m \to^* l$, then $n \to^* l$. This path from n to l exists: $n \to^* m \to^* l$.

We define the *left descendants* L of a node $n \in N$ to be the set of descendants of the left child of n.

$$m \in L(n) \text{ if exists } n' \in N,\ \lambda(n) = n',\ \text{and } n' \to^* m.$$

The set R of *right descendants* is defined similarly.

Let S be a set with a *Total Ordering* $\leq$. The set S labels of the nodes if there is a function $l : N \to S$.

Definition 5 *A tree has the search tree property if for all $n \in N$, for all $n_L \in L(n)$ and $n_R \in R(n)$, $l(n_L) < l(n)$ and $l(n) \leq l(n_R)$.*

Use of the Search Tree Property

We will find the node $n \in N$ in the tree with label $s \in S$, that is $l(n) = s$. We will assume that such a node exists. That is, we are not looking for something which is not at all in the tree. Let r be the root of the tree. We will use the working variable n in the search:

```
n = r ;
while ( l(n) != s )
{
    if ( l(n) < s )
       n = rightChild(n) ;
    else /* l(n) > s, of course! */
       n = leftChild(n) ;
}
```