Appendix

Graph Theory Terminology

The number of systems of terminology presently used in graph theory is equal, to a close approximation, to the number of graph theorists. Here we describe the particular terminology that we have chose to use throughout this book, though we make no claims about its superiority to any alternate choice of terminology.

A *finite graph* is a triple $G = (V, E, \phi)$, where V is a finite set of *vertices*, E is a finite set of *edges*, and ϕ is a function that assigns to each edge e a 2-element multiset of vertices. Thus, $\phi\colon E \to \left(\!\binom{V}{2}\!\right)$. If $\phi(e) = \{u, v\}$, then we think of e as *joining* the vertices u and v. We say that u and v are *adjacent* and that u and e, as well as v and e, are *incident*. If $u = v$, then e is called a *loop*. If ϕ is injective (one-to-one) and has no loops, then G is called *simple*. In this case, we may identify e with the set $\phi(e) = \{u, v\}$, sometimes written $e = uv$. In general, the function ϕ is rarely explicitly mentioned in dealing with graphs, and virtually never mentioned in the case of simple graphs.

A *walk* (called by some authors a path) of length n from vertex u to vertex v is a sequence $v_0e_1v_1e_2v_2 \cdots e_nv_n$ such that $v_i \in V$, $e_i \in E$, $v_0 = u$, $v_n = v$, and any two consecutive terms are incident. If G is simple then the sequence $v_0v_1 \cdots v_n$ of vertices suffices to determine the walk. A walk is *closed* if $v_0 = v_n$, a *trail* if the e_i's are distinct, and a *path* if the v_i's (and hence the e_i's) are distinct. If $n \geq 1$ and all the v_i's are distinct except for $v_0 = v_n$, then the walk is called a *cycle*.

A graph is *connected* if it is nonempty and any two distinct vertices are joined by a path (or walk). A graph without cycles is called a *free forest* or simply a *forest*. A connected graph without cycles is called a *free tree* (called by many authors simply a tree).

A *digraph* or *directed graph* is defined analogously to a graph, except now $\phi\colon E \to V \times V$; that is, an edge consists of an *ordered* pair (u, v) of vertices (possibly equal). The notions of walk, path, trail, cycle, and so on, carry over in a natural way to digraphs; see the beginning of Section 4.7.1 for further details.

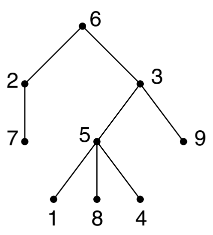

Figure A.1 A tree.

We come next to the concept of a tree. It may be defined recursively as follows. A *tree* (or *rooted tree*) T is a finite set of vertices such that:

a. One specially designated vertex is called the *root* of T, and
b. The remaining vertices (excluding the root) are partitioned into $m \geq 0$ disjoint nonempty sets $T_1, \ldots, T_m$, each of which is a tree. The trees $T_1, \ldots, T_m$ are called *subtrees* of the root.

Rather than formally defining certain terms associated with a tree, we will illustrate these terms with an example, trusting that this will make the formal definitions clear. Suppose that $T = [9]$, with root 6 and subtrees T_1, T_2. The subtree T_1 has vertices $\{2,7\}$ and root 2, while T_2 has vertices $\{1,3,4,5,8,9\}$, root 3, and subtrees T_3, T_4. The subtree T_3 has vertices $\{1,4,5,8\}$, root 5, and subtrees T_5, T_6, T_7 consisting of one vertex each, while T_4 consists of the single vertex 9. The tree T is depicted in an obvious way in Figure A.1. Note that we are drawing a tree with its root at the *top*. This is the most prevalent convention among computer scientists and combinatorialists, though many graph theorists (as well as Nature herself) would put the root at the bottom. In Figure A.1, we call vertices 2 and 3 the *children* or *successors* of vertex 6. Similarly 7 is the child of 2, 5 and 9 are the children of 3, and 1, 4, and 8 are the children of 5. We also call 2 the *parent* or *predecessor* of 7, 5 the parent of 1,4, and 8, and so on. Every vertex except the root has a unique parent. Those vertices without children are called *leaves* or *endpoints*; in Figure A.1 they are 1, 4, 7, 8, and 9.

If we take the diagram of a tree as in Figure A.1 and ignore the designation of the root (i.e., consider only the vertices and edges), then we obtain the diagram of a free tree. Conversely, given a free tree G, if we designate one of its vertices as a root, then this defines the structure of a tree T on the vertices of G. Hence a "rooted tree," meaning a free tree together with a root vertex, is conceptually identical to a tree.

A tree may also be regarded in a natural way as a poset; simply consider its diagram to be a Hasse diagram. Thus, a tree T, regarded as a poset, has a unique maximal element, namely, the root of the tree. Sometimes it is convenient to consider the dual partial ordering of T. We therefore define a *dual tree* P to be a poset such that the Hasse diagram of the dual poset P^* is the diagram of a tree.

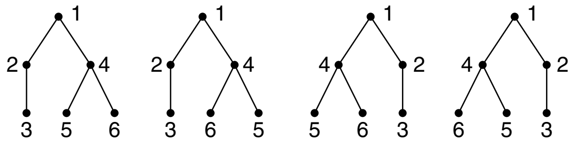

Figure A.2 Four plane trees.

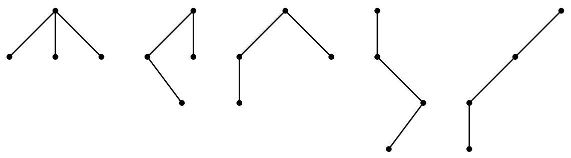
Figure A.3 Some ternary trees with four vertices.

Some important variations of trees are obtained by modifying the recursive definition. A *plane tree* or *ordered tree* is obtained by replacing (b) in the definition of a tree with:

b′. The remaining vertices (excluding the root) are put into an *ordered* partition $(T_1,\dots,T_m)$ of $m \geq 0$ pairwise disjoint, nonempty sets $T_1,\dots,T_m$, each of which is a plane tree.

To constrast the distinction between (rooted) trees and plane trees, an ordinary (rooted) tree may be referred to as an *unordered tree*. Figure A.2 shows four *different* plane trees, each of which has the same "underlying (unordered) tree." The ordering $T_1,\dots,T_m$ of the subtrees is depicted by drawing them from left-to-right in that order.

Now let $m \geq 2$. An *m-ary tree T* is obtained by replacing (a) and (b) with:

a″. Either T is empty, or else one specially designated vertex is called the *root* of T, and

b″. The remaining vertices (excluding the root) are put into a (weak) ordered partition $(T_1,\dots,T_m)$ of exactly m disjoint (*possibly empty*) sets $T_1,\dots,T_m$, each of which is an m-ary tree.

A 2-ary tree is called a *binary tree*. When drawing an m-ary tree for m small, the edges joining a vertex v to the roots of its subtrees $T_1,\dots,T_m$ are drawn at equal angles symmetric with respect to a vertical axis. Thus, an empty subtree T_i is inferred by the absence of the ith edge from v. Figure A.3 depicts 5 of the 55 nonisomorphic ternary trees with four vertices. We say that an m-ary tree is *complete* if every vertex not an endpoint has m children. In Figure A.3, only the first tree is complete.

The *length* $\ell(T)$ of a tree T is equal to its length as a poset; that is, $\ell(T)$ is the largest number ℓ for which there is a sequence $v_0, v_1, \dots, v_\ell$ of vertices such that v_i is a child of v_{i-1} for $1 \leq v_i \leq \ell$ (so v_0 is necessarily the root of T). The *complete m-ary tree of length ℓ* is the unique (up to isomorphism) complete m-ary tree with *every* maximal chain of length ℓ; it has a total of $1+m+m^2+\cdots+m^\ell$ vertices.