1.1 FUNDAMENTALS OF GRAPH THEORY

Jonathan L. Gross, Columbia University
Jay Yellen, Rollins College

Introduction

Configurations of nodes and connections occur in a great diversity of applications. They may represent physical networks, such as electrical circuits, roadways, or organic molecules. They are also used in representing less tangible interactions as might occur in ecosystems, sociological relationships, databases, or in the flow of control in a computer program.

1.1.1 Graphs and Digraphs

Any mathematical object involving points and connections between them may be called a *graph*. If all the connections are unidirectional, it is called a *digraph*. Our highly inclusive definition in this initial section of the *Handbook* permits fluent discussion of almost any particular modification of the basic model that has ever been called a graph.

Basic Terminology

DEFINITIONS

D1: A ***graph*** $G = (V, E)$ consists of two sets V and E.

- The elements of V are called ***vertices*** (or ***nodes***).
- The elements of E are called ***edges***.
- Each edge has a set of one or two vertices associated to it, which are called its ***endpoints***. An edge is said to ***join*** its endpoints.

NOTATION: The subscripted notations V_G and E_G (or $V(G)$ and $E(G)$) are used for the vertex- and edge-sets when G is not the only graph under consideration.

D2: If vertex v is an endpoint of edge e, then v is said to be ***incident*** on e, and e is incident on v.

D3: A vertex u is ***adjacent*** to vertex v if they are joined by an edge.

D4: Two adjacent vertices may be called ***neighbors***.

D5: ***Adjacent edges*** are two edges that have an endpoint in common.

D6: A ***proper edge*** is an edge that joins two distinct vertices.

D7: A ***multi-edge*** is a collection of two or more edges having identical endpoints.

D8: A ***simple adjacency*** between vertices occurs when there is exactly one edge between them.

D9: The ***edge-multiplicity*** between a pair of vertices u and v is the number of edges between them.

D10: A ***self-loop*** is an edge that joins a single endpoint to itself.

TERMINOLOGY: An alternative word for "self-loop" is "loop". This can be used in contexts in which "loop" has no other meanings.

TERMINOLOGY: In computer science, the word "graph" is commonly used either to mean a graph as defined here, or to mean a computer-represented data structure whose value is a graph.

EXAMPLE

E1: A line drawing of a graph $G = (V, E)$ is shown in Figure 1.1.1. It has vertex-set $V = \{u, v, w, x\}$ and edge-set $E = \{a, b, c, d, e, f\}$. The set $\{a, b\}$ is a multi-edge with endpoints u and v, and edge c is a self-loop.

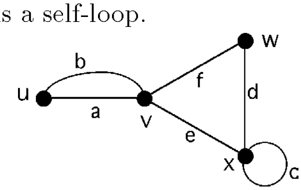

Figure 1.1.1 **A graph.**

REMARKS

R1: A graph is realized in a plane or in 3-space as a set of points, representing the vertices, and a set of curved or straight line segments, representing the edges. The curvature or length of such a line segment is irrelevant to the meaning. However, if a *direction* is indicated, that is significant.

R2: Occasionally, a graph is *parametrized* so that each edge is regarded as the homeomorphic image of the real interval $[0, 1]$ (except that for a self-loop, the endpoints 0 and 1 have the same image).

Simple Graphs

Most of theoretical graph theory is concerned with *simple* graphs. This is partly because many problems regarding general graphs can be reduced to problems about simple graphs.

DEFINITIONS

D11: A ***simple graph*** is a graph that has no self-loops or multi-edges.

D12: A ***trivial graph*** is a graph consisting of one vertex and no edges.

D13: A ***null graph*** is a graph whose vertex- and edge-sets are empty.

Edge Notation for Simple Adjacencies and for Multi-edges

NOTATION: An edge joining vertices u and v of a graph may be denoted by the juxtaposition uv if it is the only such edge. Occasionally, the ordered pair (u, v) is used in this situation, instead of uv. To avoid ambiguities when multi-edges exist, or whenever else desired, the edges of a general graph may be given their own names, as in Figure 1.1.1 above.

EXAMPLE

E2: The simple graph shown in Figure 1.1.2 has edge-set $E = \{uv, vw, vx, wx\}$.

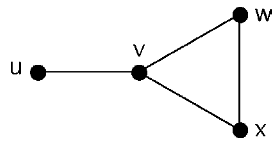

Figure 1.1.2 **A simple graph.**

General Graphs

Many applications require non-simple graphs as models. Moreover, some non-simple graphs serve an essential role in theoretical constructions, especially in constructing graph drawings (simple and non-simple) on surfaces (see Chapter 7).

TERMINOLOGY NOTE: Although the term "graph" means that self-loops and multi-edges are allowed, sometimes, for emphasis, the term ***general graph*** is used.

DEFINITIONS

D14: A ***loopless graph*** is a graph that has no self-loops. (It might have multi-edges.) Sometimes a loopless graph is referred to as a ***multigraph***.

D15: The ***dipole*** D_n is a loopless graph with two vertices and n edges joining them.

D16: The ***bouquet*** B_n is a graph with one vertex and n self-loops.

EXAMPLES

E3: The loopless graph in Figure 1.1.3 depicts the benzene molecule C_6H_6.

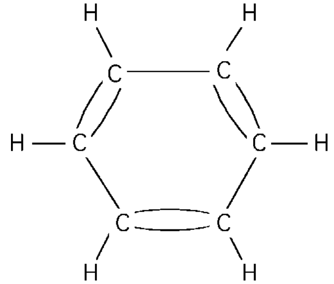

Figure 1.1.3 **Graph model for a benzene ring.**

E4: The dipole D_3 is shown in Figure 1.1.4.

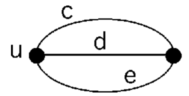

Figure 1.1.4 **The loopless graph D_3.**

E5: Two graphs with self-loops are shown in Figure 1.1.5.

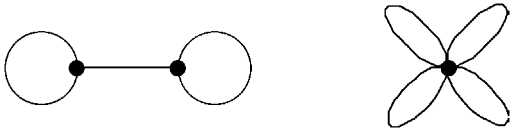

Figure 1.1.5 **The dumbbell graph and the bouquet B_4.**

Attributes

Allowing graphs to have additional attributes beyond vertices and edges enables them to serve as mathematical models for a wide variety of applications. Two of the most common additional edge attributes, both described in great detail later in the *Handbook*, are edge *direction* (e.g., Chapters 3 and 11) and edge *weight* (e.g., Chapters 4 and 11). Another common attribute (for edges or vertices) is *color*. Graph coloring is discussed in Chapter 5.

DEFINITIONS

D17: A ***vertex attribute*** is a function from the vertex-set to some set of possible attribute values.

D18: An ***edge attribute*** is a function from the edge-set to some set of possible attribute values.

Digraphs

An edge between two vertices creates a connection in two opposite senses at once. Assigning a direction makes one of these senses *forward* and the other *backward*. Viewing direction as an edge attribute is partly motivated by its impact on computer implementations of graph algorithms. Moreover, from a mathematical perspective, regarding *directed graphs* as augmented graphs makes it easier to view certain results that tend to be established separately for graphs and for digraphs as a single result that applies to both. The attribute of edge direction is developed extensively in Chapter 3 and elsewhere in this *Handbook*.

DEFINITIONS

D19: A ***directed edge*** (or **arc**) is an edge e, one of whose endpoints is designated as the ***tail***, and whose other endpoint is designated as the ***head***. They are denoted *head*(e) and *tail*(e), respectively.

TERMINOLOGY: A directed edge is said to be ***directed from*** its tail and ***directed to*** its head. (The tail and the head of a directed self-loop are the same vertex.)