Chapter 1

Introduction to Graph Theory and Graph Labelings

1.1 Basic Notation and Terminology

The notation and terminology used through this thesis is taken for the most part from Chartrand and Lesniak [6].

In this first section we will introduce some basic definitions and notation about graph theory. We begin this task by introducing what we mean by a graph. A graph G is a finite non empty set of objects, called vertices, together with a set of unordered pairs of distinct vertices of G called edges. The vertex and edge sets of G are usually denoted by $V(G)$ and $E(G)$ respectively.

If an edge $e = \{u, v\}$, then se say that e joins the vertices u and v, and u and v are said to be adjacent edges in G. From now on, and for simplicity, we will denote the edge $\{u, v\}$ by uv, whenever no ambiguity arises (the same convention will apply to loop graphs, which we define below). We say that a graph G of order p and size q is a (p, q)-graph.

It has become a tradition to describe graphs by means of diagrams in which each element of the vertex set of the graph is represented by a dot and each edge $e = uv$ is represented by a curve joining the dots that represent the vertices u and v.

For example, if we consider the graph G with

$$V(G) = \{v_1, v_2, v_3, v_4, v_5, \}$$

and

$$E(G) = \{v_1v_2, v_1v_3, v_2v_3, v_4v_5\}.$$

Then a possible diagram for this graphs is shown in Figure 1.1. However, although the diagram is the most common way of representing graphs, there

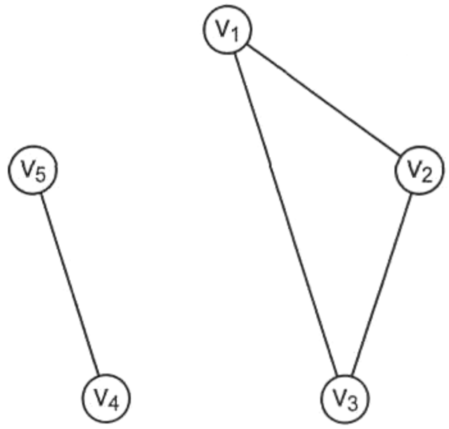

Figure 1.1

are many other ways of representing them. Another very common way is by means of the adjacency matrix. Let G be a graph with $V(G) = \{v_1, v_2, \ldots v_p\}$. Then define its $p \times p$ adjacency matrix $A = (a_{ij})$ to be

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E(G); \\ 0 & \text{if } v_i v_j \notin E(G). \end{cases}$$

For example the adjacency matrix A for the graph of Figure 1.1 is shown below.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

If some other way for representing a graph is needed, then it will be defined in situ.

Next, we will define a concept that allows to tell when two graphs are basically "equal." This is the concept of isomorphism. Two graphs are said to be isomorphic if they have the same structure, and at the most, they differ in the way their vertices and edges are labeled, or in the way they are drawn. In order to make this idea more precise, we will define two graphs G_1 and G_2 to be isomorphic if there exists a bijective function $\phi : V(G) \to V(G)$ such that $uv \in (G_1) \Leftrightarrow \phi(u)\,\phi(v) \in E(G_2)$. The function ϕ is called an isomorphism. If two graphs G_1 and G_2 are isomorphic, then we write $G_1 \cong G_2$.

A parameter that appears often when studying graphs is the degree of vertex. The degree of a vertex u of a graph G, denoted by $\deg_G u$, or simply by $\deg u$ if the graph G is clear from the context, is defined as

$$\deg_G u = |\{v \mid uv \in E(G)\}|$$

A vertex v of a graph G is called even if its degree is even and odd if its degree is odd. Also, if $\deg_G v = 0$, v is called an isolated vertex, and if $\deg_G v = 1$, it is called an end-vertex. Also, if $e = uv$ is an edge of a graph G such that either $\deg_G u = 1$ or $\deg_G v = 1$, then e is called a pendant edge of G. Let G be a graph, then a graph H is said to be a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If H is a subgraph of G then we will write $H \subseteq G$. Some types of subgraphs that appear often when studying graph theory are those obtained by the deletion of a vertex or an edge. If G is any graph with $|V(G)| \geq 2$, and $v \in V(G)$, then the subgraph $G - v$ of G is defined to be

$$V(G - v) = V(G) \setminus \{v\}$$

and

$$E(G - v) = E(G) \setminus \{e \in E(G) \mid v \in e\}.$$

Also if $e \in E(G)$ then the subgraph $G - e$ is defined to be

$$V(G - e) = V(G)$$

and

$$E(G - e) = E(G) \setminus \{e\}.$$

If u and v are non adjacent vertices in a graph G and $e = uv$, then the graph $G + e$ is defined to be

$$V(G + e) = V(G)$$

$$E(G + e) = E(G) \cup \{e\}.$$

Another important type of subgraphs are the induced subgraphs. Let G be a graph, and suppose that $U \subseteq V(G)$ is nonempty. Then the subgraph $\langle U \rangle$ of G induced by U is the graph such that

$$V(\langle U \rangle) = U$$

and

$$E(\langle U \rangle) = \{xy \in E(G) \mid x, y \in U\}.$$

Also, if $F \subseteq E(G)$ is nonempty, then the subgraph $\langle F \rangle$ of G induced by F is the graph such that

$$V(\langle F \rangle) = \{u \in V(G) \mid uv \in F \text{ for some } v \in V(G)\}$$

$$E(\langle F \rangle) = F.$$

Another important concept is the one of connectedness. Informally, we say that a graph is connected if it is possible to "travel" from any vertex of a graph to any other vertex of it, using the vertices and edges of the graph. We can make this concept more formal in the following way. A graph G is connected if given any pair of distinct vertices of G, namely u and v, there exists a sequence of vertices and edges of G of the form

$$u = x_1, x_1x_2, x_2, x_2x_3, x_3, x_3x_4, x_4, \ldots, x_{n-2}x_{n-1}, x_{n-1}, x_{n-1}x_n, x_n = v$$

and is called disconnected otherwise.

If G is a disconnected graph then we define a component of G to be a subgraph induced by a set $U \subset V(G)$ such that $\langle U \rangle$ is connected, but if $v \in V(G) \backslash U$, then $\langle U \cup \{v\} \rangle$ is disconnected. The number of components of a graph G is usually denoted by $k(G)$. A bridge e of a graph G is any element of $E(G)$ such that $k(G) < k(G - e)$.

Similarly a cut vertex v of a graph G is any element of $V(G)$ with the property that $k(G) < k(G - v)$.

In order to conclude this first section, we will introduce the very important concept of decomposition. A decomposition of a graph G is a collection $\{H_i\}$ of subgraphs of G such that $H_i = \langle E_i \rangle$ for some subset E_i of $E(G)$ and where $\{E_i\}$ is a partition of $E(G)$. If $\{H_i\}$ is a decomposition of G then we can write

$$G \cong H_1 \oplus H_2 \oplus \ldots \oplus H_{|\{H_i\}|} = \bigoplus_{i=1}^{|\{H_i\}|} H_i.$$

1.2 Directed Graphs and Loop-Graphs

Graphs where first introduced as ways of modeling situations that may take place in real life. However, although graphs work very well as models, sometimes variations to the concept of graph are needed. The goal in this section is to introduce some of these modifications. We first introduce the concept of directed graphs, also called digraphs.

A directed graph or digraph D is a finite non empty set of objects called vertices together with a (possibly empty) set of ordered pairs of distinct vertices of D called arcs or directed edges. As in graphs the vertex and edge sets are denoted by $V(D)$ and $E(D)$ respectively. Note that it is possible to represent a digraph, topologically with a drawing in which each vertex is denoted by a dot, and each ordered pair (u, v) is denoted by a curve joining vertices u and v, with an arrow pointing from vertex u to vertex v. Figure 1.2 shows a possible representation of the digraph D, defined below

$$V(D) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(D) = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_4, v_5)\}.$$

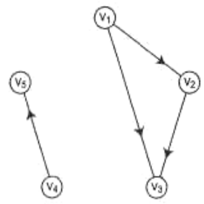

Figure 1.2

Another way of representing digraphs is using the adjacency matrix. Let D be a digraph with $V(D) = \{v_1, v_2, \ldots, v_p\}$. The $p \times p$ adjacency matrix $A = (a_{ij})$ of the digraph D is defined by the rule

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E(D); \\ 0, & \text{if } (v_i, v_j) \notin E(D). \end{cases}$$

It is worthwhile to mention that in general, the terminology used for digraphs is similar to the terminology used for graphs. However, there are some exceptions to this rule. For instance, the concept of degree of a vertex for graphs is substituted by the concepts of indegree and outdegree of a vertex in the case of digraphs. The indegree of a vertex u of a graph G, denoted by $\mathrm{in}(u)$ is defined to be

$$\mathrm{in}(u) = |\{v \mid (v, u) \in E(D)\}|.$$

The outdegree of a vertex u denoted by $\mathrm{out}(u)$, is defined to be

$$\mathrm{out}(u) = |\{v \mid (u, v) \in E(D)\}|.$$

Next, we will introduce the concept of loop-graphs.

A loop-graph L is a finite non empty set of objects called vertices, together with a set of edges consisting of subsets of the set of vertices such that each subset consists of either one vertex or two vertices. The subsets consisting of only one vertex are called loops and the subsets consisting of two elements are called edges. The set of vertices of a loop-graph L is denoted by $V(L)$ and the set of edges is denoted by $E(L)$. Also, the topological representation of a loop graph is obtained as follows. The vertices of the loop graph are represented by dots, while if uv is an edge of the loop graph, then uv is represented by a curve joining the dots that represent vertices u and v. A loop u of the loop graph is represented by a curve beginning and ending at u. Figure 1.3 shows the topological representation of the loop-graph L defined below,

$$V(L) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(L) = \{v_1v_2, v_1v_3, v_2v_3, v_3v_4, v_4v_1, v_4v_5, v_1, v_2, v_5\}$$

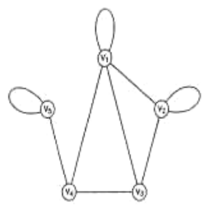

Figure 1.3

It is also worthwhile to mention that in general the terminology used for loop-graphs is similar to the terminology used for graphs. However, as in the case of digraphs, we need to pay special attention to the concept of degree.

If L is a loop-graph and $v \in V(L)$, then the degree of v in L, denoted by $\deg_L(v)$ or simply by $\deg(v)$ if the loop graph is clear, is defined as

$$\deg(v) = \begin{cases} |\{u \mid uv \in E(L)\}| & \text{if } v \notin E(L) \\ |\{u \mid uv \in E(L)\}| + 2 & \text{if } v \in E(L) \text{ and } u \neq v. \end{cases}$$

Finally we will introduce the concept of directed loop graph. A directed loop graph L_D is defined to be a set of objects called vertices together with a set of ordered pairs that are not necessarily distinct vertices of L_D. The topological representation and the adjacency matrix of a directed digraph L_D are obtained in the obvious way, and the terminology is similar to the terminology used for digraphs.

1.3 Important Types of Graphs

Throughout this thesis, we will encounter several families of graphs very often. This is why we devote this section to define these families of graphs. The first type of graph that we will discuss is the complete graph. A graph is said to be the complete graph on p vertices, and is denoted by K_p, if its order is p and its size is $q = (p\,(p-1))/2$. Figure 1.4 shows the graphs K_5, K_6, and K_7.

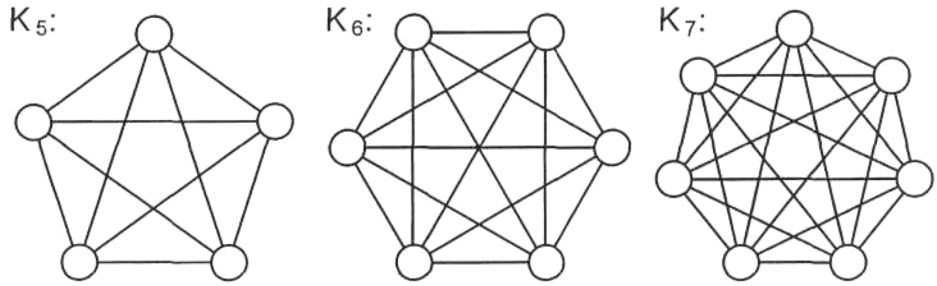

Figure 1.4

Next, we introduce the concept of bipartiteness. A graph G is said to be bipartite if it is possible to partition the set $V\,(G)$ into two sets V_1 and V_2 such that if $uv \in E\,(G)$ then $\{u, v\} \nsubseteq V_i$ for $i = 1, 2$ and is called a complete bipartite graph, denoted by $K_{|V_1|,|V_2|}$, if every vertex in V_1 is adjacent to every vertex in V_2 and vice versa. Figure 1.5 shows the graphs $K_{3,3}$ and $K_{3,5}$.

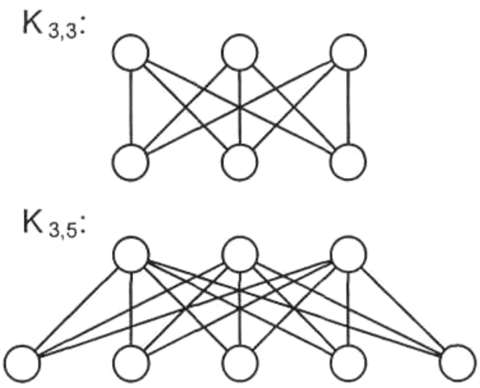

Figure 1.5

A graph G is called r-regular if $\deg v = r$, for all $v \in V\,(G)$ and a cycle is any connected 2-regular graph. Usually the cycle on n vertices is denoted by C_n. Figure 1.6 shows a 3-regular graph P and the cycle C_5.

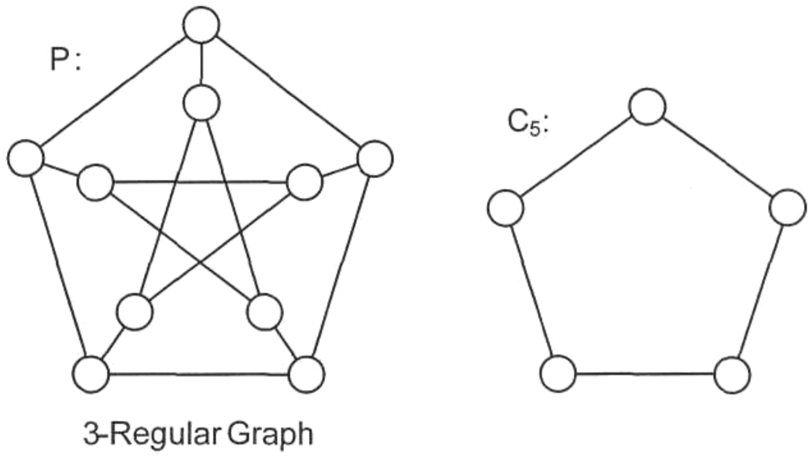

Figure 1.6

It is interesting to mention that the graph P of Figure 1.6 is a famous graph called the Petersen Graph.

An acyclic graph is a graph that does not contain any subgraph isomorphic to a cycle and a tree is a connected acyclic graph. Acyclic graphs with more than one component are called forests. A unicyclic graph is a graph that is not acyclic and there is at least an edge such that the deletion of this edge results in an acyclic graph. See Figure 1.7 for clarification.

A path is a graph obtained from the deletion of any edge of a cycle and a linear forest is a graph for which each component is a path. Usually a path with n vertices is denoted by P_n. A caterpillar is any tree for which the deletion of its end-vertices produces a path, while a lobster is any tree for which the deletion of its vertices produces a caterpillar. A star is any graph isomorphic $K_{1,n}$, while a galaxy is a graph for which each component is a star. See Figure 1.8 for examples.

To close this section, we define the complement of a graph G, denoted by $\overline{G}$, to be the graph with $V\left(\overline{G}\right) = V\left(G\right)$ and $E\left(\overline{G}\right) = \{uv \mid uv \notin E\left(G\right)\}$. For an example see Figure 1.9.

1.4 Operations on Graphs

It is a natural question to ask when two graphs are given, how is it possible to combine them in order to obtain a new graph. In fact, there exist many different ways of combining graphs, and in this section we will describe only the most common types of binary operations defined on the set of graphs. We begin by considering two graphs G_1 and G_2 with the property that $V\left(G_1\right) \cap$

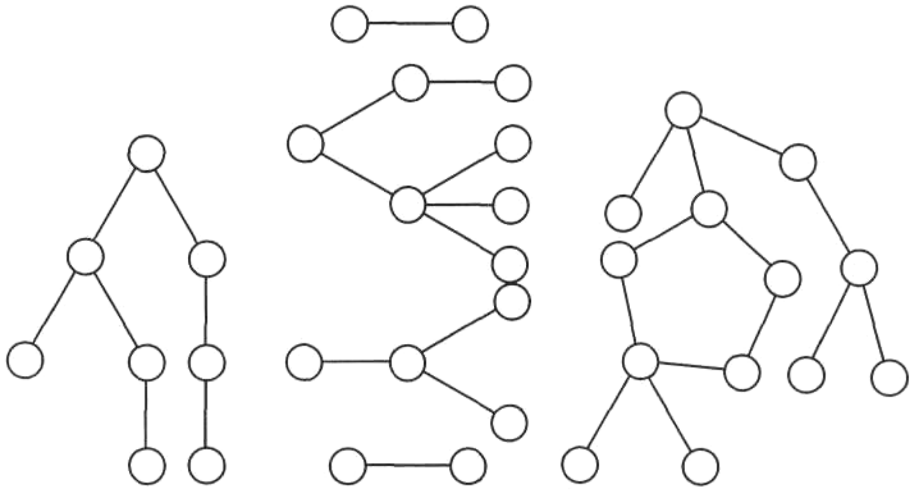

Figure 1.7

$V(G_2) = \emptyset$. The union $G = G_1 \cup G_2$ is the graph that has $V(G) = V(G_1) \cup V(G_2)$, and $E(G) = E(G_1) \cup E(G_2)$. In general if $G_1, G_2, \ldots, G_n$ are n graphs such that $V(G_i) \cap V(G_j) = \emptyset$ for $i \neq j$, we can define the union $G = G_1 \cup G_2 \cup \ldots \cup G_n = \cup_{i=1}^{n} G_i$ to be the graph G such that $V(G) = \cup_{i=1}^{n} V(G_i)$ and $E(G) = \cup_{i=1}^{n} E(G_i)$.

If a graph G consists of $n\,(\geq 2)$ disjoint copies of a given graph H, then it is possible to write G as $G = nH$.

For instance, Figures 1.10 and 1.11 show the graphs $G \cong 2K_4 \cup 2K_{1,3} \cup 3C_4 \cup C_5$ and $H \cong 4K_5$ respectively.

Next, we define the join operation of G_1 and G_2 to be the graph $G = G_1 + G_2$ such that $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2) \cup \{xy \mid x \in V(G_1), y \in V(G_2)\}$. So if we let $G_1 = K_3$ and $G_2 = P_2$, then we obtain the graph in Figure 1.12.

The Cartesian product of graphs G_1 and G_2 is the graph $G = G_1 \times G_2$ obtained in the following way.

$$V(G) = V(G_1) \times V(G_2)$$

and

$$\begin{aligned}(x_1, x_2)(y_1, y_2) \in E(G) \quad \Leftrightarrow \quad & x_1 = y_1 \text{ and } x_2y_2 \in E(G_2) \text{ or} \\ & x_2 = y_2 \text{ and } x_1y_1 \in E(G_1).\end{aligned}$$

See Figure 1.13 for an example.

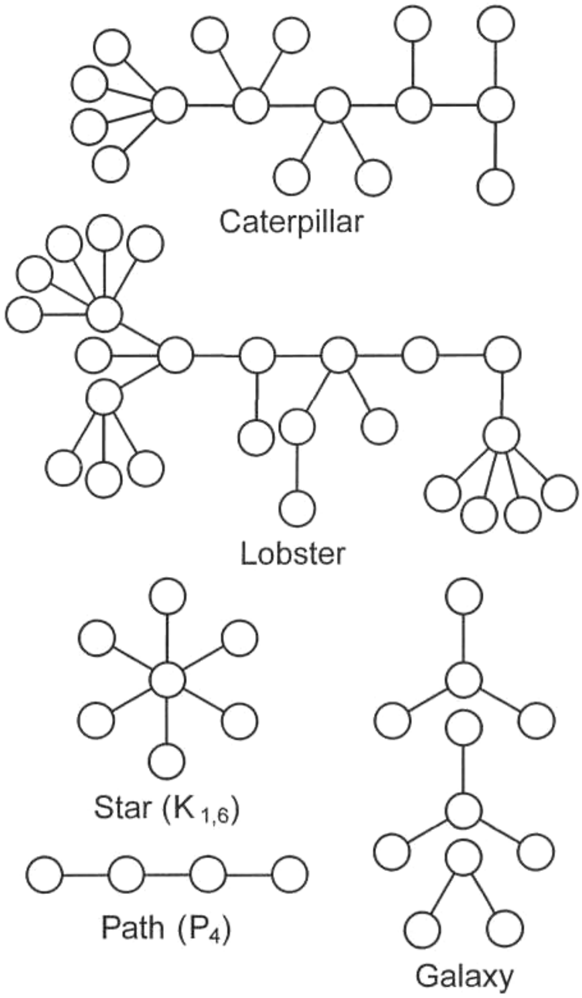

Figure 1.8

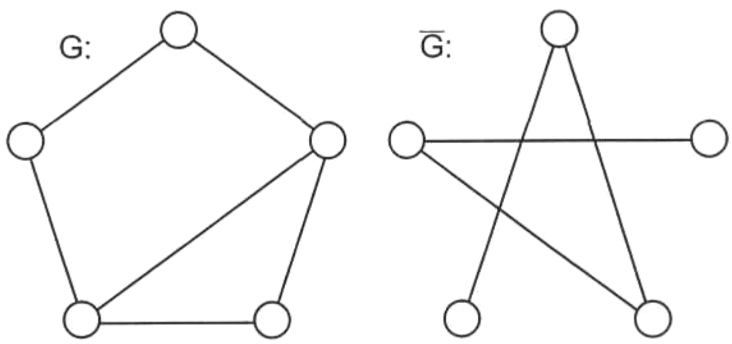

Figure 1.9

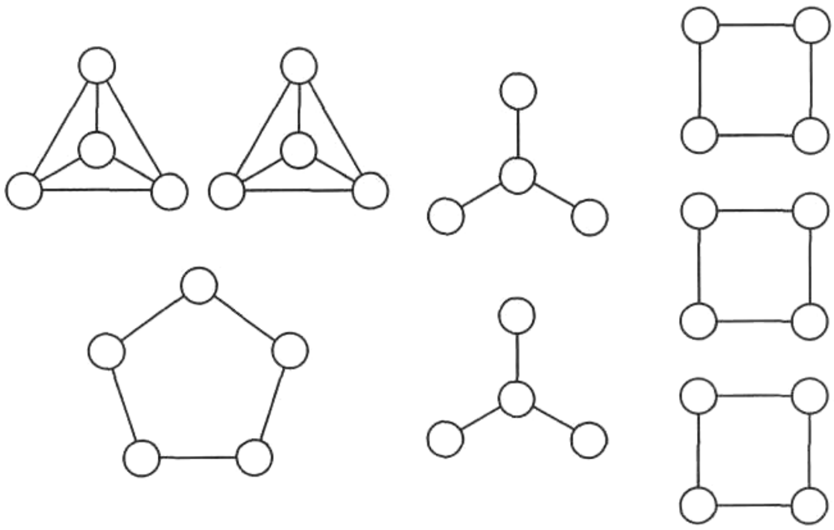

Figure 1.10: $2K_4 \cup 2K_{1,3} \cup 3C_4UC_5$.

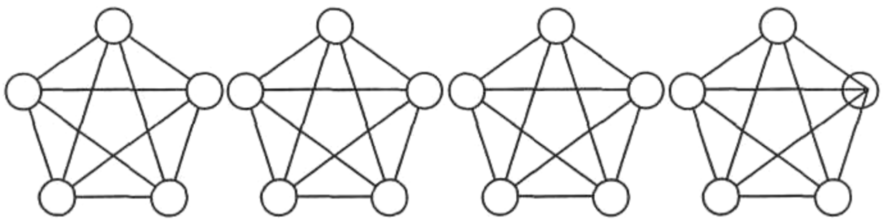

Figure 1.11: $4K_5$.

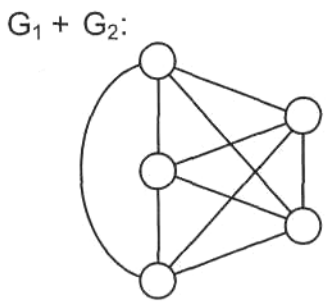

Figure 1.12

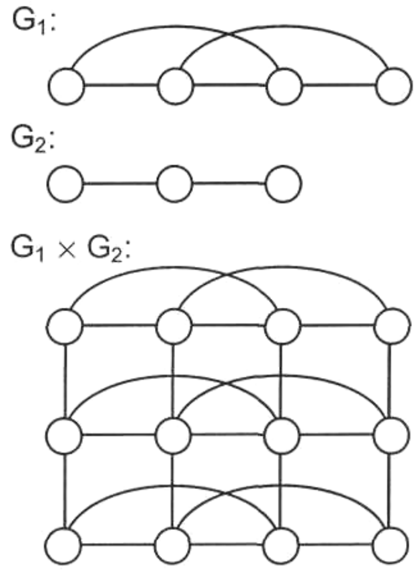

Figure 1.13

The last operation that we will discuss is the crown product of two graphs. The crown product of the graphs G_1 and G_2 is the graph $G = G_1 \odot G_2$ obtained by placing a copy of G_1 and $|V(G_1)|$ copies of G_2 and then joining each vertex of G_1 with all the vertices in one copy of G_2 in such a way that all vertices in the same copy of G_2 are joined with exactly one vertex of G_1. See Figure 1.14 for an example.

1.5 Introduction to Graph Labelings

The area of graph theory has experienced fast development during the last 60 years, and among the huge diversity of concepts that appear while studying this subject, one that has gained a lot of popularity is the concept of labelings of graphs. With more than 250 papers in the literature and a very complete dynamic survey by Joseph Gallian [18], this new branch of mathematics has caught the attention of many authors, and many new labeling results appear every year. This popularity is not only due to the mathematical challenges of graph labelings, but also, to the wide range of applications that graph labelings offer to other branches of science, as for instance x-ray crystallography, coding theory, radar, astronomy, circuit design and communication design. In fact G. S. Blomm, and S. W. Golomb studied applications of graph labelings to other branches of science, and it is possible to find part of this work in [3] and [4].

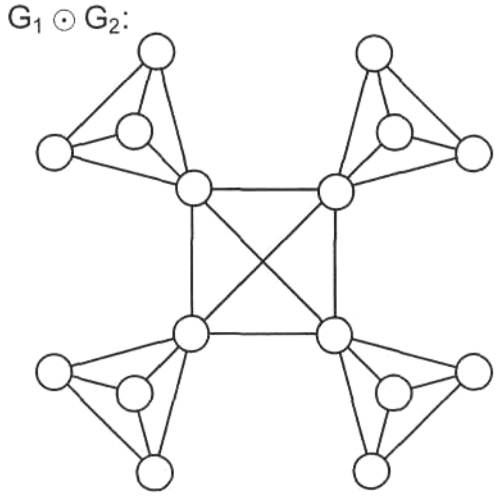

Figure 1.14

It is important to distinguish between two major classes of labelings, vertex labelings and total labelings [18]. A vertex labeling of a graph G is an assignment f of labels to the vertices of G that induces a label for each edge uv of G, depending on the labels $f(u)$ and $f(v)$. The oldest and better studied vertex labeling is the one introduced by Rosa [35] in 1967, called graceful labeling. Graceful labelings were introduced in order to provide an alternative way to attack the conjecture of Ringel [36] which states that the complete graph K_{2n+1} is decomposable into $2n+1$ subgraphs that are all isomorphic to a given tree of size n.

A function f is called a graceful labeling of a graph G with size q if f is an injective function from $V(G)$ to the set $\{0, 1, 2, \ldots, q\}$ with the property that the function $\overline{f}$ with domain $E(G)$ and range in the integers, defined by the rule $\overline{f}(uv) = |f(u) - f(v)|$, assigns different labels to the edges of G. If a graph G admits a graceful labeling, then we say that G is a graceful graph.

Although Erdős proved in an unpublished paper that almost all graphs are not graceful, many particular families of graphs have been proven to admit graceful labelings, (for more information see [18]). In particular, the most important conjecture in this direction (and probably in the whole area of graph labelings) is the one known as the Ringel-Kotzig conjecture that all trees are graceful. This conjecture has been the focus of many papers, but in spite of all efforts, no major progress has been made towards the final solution. It is worthwhile to mention that one of the main reasons this conjecture has become so popular, is because a positive answer to it implies the truth of Ringel's conjecture, mentioned before. Another well-known conjecture concerning graceful labelings is the one that states that all unicyclic graphs except C_n where $m \equiv 1$ or $2 \pmod 4$ are graceful. This one is known as Truszczynski's conjecture [39]. Truszczynski proved some particular families of unicyclic graphs of order less than or equal to q to be graceful. However, in spite of all this work, general results about Truszczynski's conjecture are non-existing.

Another important vertex labeling, that has also been the main subject of study of many papers, is the harmonious labeling. The harmonious labeling was introduced in 1980 by Graham and Sloane [22] as a possible way to study additive bases. A labeling f of the vertices of a graph G of size q is called harmonious if f is an injective function from $V(G)$ to the additive group $\mathbb{Z}_q$ such that the function $\overline{f}$ from the set $E(G)$ to $\mathbb{Z}_q$ defined by the rule $\overline{f}(uv) = f(u) + f(v) \pmod q$ assigns different labels to the edges of G. If G is a tree, then the condition that f is injective is relaxed and exactly two vertex labels are allowed to be equal. If a graph G admits a harmonious labeling, then it is said to be a harmonious graph. As in the case of graceful labelings, in the original paper, Graham and Sloane proved that almost no graphs are harmonious and also conjectured that all trees admit harmonious labelings. Since then, many different families of graphs have been proved to be harmonious, see [18], however, general results are rare, and it seems that at least in the near future the conjecture that all trees are harmonious is out of reach.

Motivated by these two types of vertex labelings, many authors have defined a large amount of different vertex labelings that Gallian [18] divides into two main groups. The first group is called variations of graceful labelings and the second one is called variations of harmonious labelings.

Among the most important labelings in the first group, are α-labelings, odd graceful labelings, graceful like labelings, cordial labelings, k-equitable labelings and hamming-graceful labelings. Among the most important labelings in the second group we find sequential and strongly C-harmonious labelings, elegant labelings and feliticious labelings. For more information on these labelings the reader is referred to [18].

The other important major class of labelings, as we said before, is the class of total labelings. A total labelings is a function from the set of vertices union the set of edges of a given graph G to a set of labels. The most important labelings of this type are k-sequential labelings, sequentially additive labelings, magic labelings and super magic labelings. Although, as we will see, super magic labeling can be thought of, as vertex labelings. In fact viewing super magic labelings as vertex labelings seem to be more useful that viewing them as total labelings. The main focus of our work will be on magic and super magic labelings to which we will pay special attention in the next chapters. If the reader is interested about other types of total labelings he is referred to [18].