Contents

CHAPTER 1

Basics of Graph Theory

> ” For one has only to look around to see ’real-world graphs’ in abundance, either in nature (trees, for example) or in the works of man (transportation networks, for example). Surely someone at some time would have passed from some real-world object, situation, or problem to the abstraction we call graphs, and graph theory would have been born.”[1]
>
> **by D. R. Fulkerson.**

Summary In this chapter a short introduction of the basic definitions from graph theory will be given. These definitions will help to follow the discussion given in rest of the document as well as for easy reference to the nomenclature used afterward.

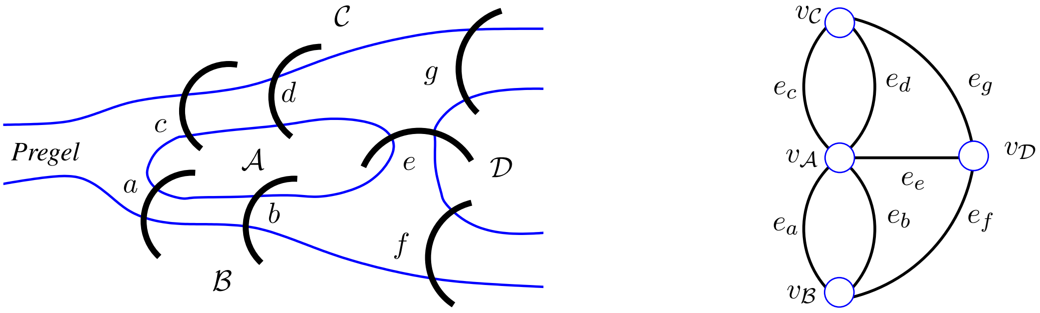
Keywords: Graph, multi-graph, vertex neighbor, edge adjacency, vertex degree, subgraphs, walk, paths, cycles, connectivity, forest, tree, vertex removal, edge removal, vertex identifying, edge contraction.

1.1 Introduction

In 1736, Leonard Euler was puzzled whether it is possible to walk across all the bridges on the river Pregel in Königsberg[2] only once and return to the starting point (see Figure 1.1a). This is how Euler stated the problem in ”Solutio problematis ad geometriam situs pertinentis.” [Euler, 1736] (an English translation of this paper can be found in [Biggs et al., 1976]):

[1] From preface to *Studies in Graph Theory*, Part II, The Mathematical Association of America, 1975.

[2] Nowadays Pregoyla in Kaliningrad.



(a) Seven bridges on the river Pregel
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ – landmasses
a, b, c, d, e, f, and g – bridges

(b) Abstracted graph
$v_{\mathcal{A}}$, $v_{\mathcal{B}}$, $v_{\mathcal{C}}$, and $v_{\mathcal{D}}$ – vertices
e_a, e_b, e_c, e_d, e_e, e_f, and e_g – edges

Figure 1.1: The seven bridges problem and the abstracted graph.

> ”In Königsberg in Prussia, there is an island $\mathcal{A}$, called *Kneiphoff*; the river (Pregel) which surrounds it is divided into two branches, as can be seen in Figure 1.1a, and these two branches are crossed by seven bridges, a, b, c, d, e, f and g. Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he would cross each bridge once and only once. I was told that some people asserted that this was impossible, while others were in doubt; but nobody would actually assert that it could be done. On the basis of the above, I have formulated the general problem: Given any configuration of the river and the branches into which it may divide, as well as any number of bridges, to determine whether or not it is possible to cross each bridge exactly once.”

In order to solve this problem, Euler in an ingenious way abstracted the bridges and the landmasses. He replaced each landmass by a dot (called vertex) and the each bridge by an arch (called edge or line) (Figure 1.1b). Euler proved that there is no solution to this problem. The Königsberg bridge problem was the first problem studied in what is later called graph theory. This problem was a starting point also for another branch in mathematics, the topology. This example shows a connection between graph theory and topology.

Unfortunately many books on graph theory have different notions for the same thing, or the same term has different meanings. The main purpose of this chapter is to collect basic notions of the graph theory in one place and to be consistent in terminology. This will help to follow the discussion given in rest of the document as well as for easy reference to the nomenclature used afterward. The definitions are compiled from the books [Diestel, 1997], [Thulasiraman and Swamy, 1992], [Harary, 1969], [Christofides, 1975] and [Bondy and Murty, 1976], therefore the citations are not repeated. Interested reader can find all these definitions and more in the above mentioned books. The erudite reader in graph theory can skip reading this chapter.

1.2 Basic Definitions

The Webster dictionary [Webster, 1913] defines graphs as having two meanings:

> **Graph**, n. (Math.)
> 1. A curve or surface, the locus of a point whose coordinates are the variables in the equation of the locus.
> 2. A diagram symbolizing a system of interrelations by spots, all distinguishable from one another and some connected by lines of the same kind.

The non-formal definition of the graph given in point 2 is the meaning used in this document. Formally, one can define graph G on sets V and E as:

Definition 1.1 (Graph) *A graph $G = (V(G), E(G), \iota_G(\cdot))$ is a pair of sets of $V(G)$ and $E(G)$ and an incidence relation $\iota_G(\cdot)$ that maps pairs of elements of $V(G)$ (not necessarily distinct) to elements of $E(G)$.*

The elements v_i of the set $V(G)$ are called vertices (or nodes, or points) of the graph G, and the elements e_j of $E(G)$ are its edges (or lines). Let an example be used to clarify the incidence relations $\iota_G(\cdot)$. Let the set of vertices of the graph G in Figure 1.1b be given by $V(G) = \{v_{\mathcal{A}}, v_{\mathcal{B}}, v_{\mathcal{C}}, v_{\mathcal{D}}\}$ and the edge set by $E(G) = \{e_a, e_b, e_c, e_d, e_f, e_g\}$. The incidence relation is defined as :

$$\begin{aligned} \iota_G(e_a) = (v_{\mathcal{A}}, v_{\mathcal{B}}),\ \iota_G(e_b) = (v_{\mathcal{A}}, v_{\mathcal{B}}),\ \iota_G(e_c) = (v_{\mathcal{A}}, v_{\mathcal{C}}), \\ \iota_G(e_d) = (v_{\mathcal{A}}, v_{\mathcal{C}}),\ \iota_G(e_e) = (v_{\mathcal{A}}, v_{\mathcal{D}}),\ \iota_G(e_f) = (v_{\mathcal{B}}, v_{\mathcal{D}}), \\ \iota_G(e_g) = (v_{\mathcal{C}}, v_{\mathcal{D}}). \end{aligned} \tag{1.1}$$

For the sake of simplicity of the notation, the incidence relation will be omitted, therefore one can write, without the fear of confusion:

$$\begin{aligned} e_a = (v_{\mathcal{A}}, v_{\mathcal{B}}),\ e_b = (v_{\mathcal{A}}, v_{\mathcal{B}}),\ e_c = (v_{\mathcal{A}}, v_{\mathcal{C}}), \\ e_d = (v_{\mathcal{A}}, v_{\mathcal{C}}),\ e_e = (v_{\mathcal{A}}, v_{\mathcal{D}}),\ e_f = (v_{\mathcal{B}}, v_{\mathcal{D}}), \\ e_g = (v_{\mathcal{C}}, v_{\mathcal{D}}). \end{aligned} \tag{1.2}$$

i.e. the graph is defined as $G = (V, E)$ without explicit mentioning of the incidence relation, even though it is always understood. The vertex set $V(G)$ and $E(G)$ are simple written as V and E. There will be no distinction between a graph and its sets, one may write a vertex $v \in G$ or $v \in V$ instead of $v \in V(G)$, an edge $e \in G$ or $e \in E$, and so on. Vertices and edges are usually represented with symbols like $v_1, v_2, ...$ and $e_1, e_2, ...$, respectively. Note that in Equation 1.2, each edge is identified with a pair of vertices. If the edges are represented with ordered pairs of vertices, than the graph G is called *directed* or *oriented*, otherwise it is called *undirected* or *non-oriented*. Two vertices connected by an edge $e_k = (v_i, v_j)$ are called *end vertices* or *ends* of e_k. In the directed graph the vertex v_i is called the *source*, and v_j the *target* vertex of edge e_k. Since the elements of edge set E are distinct, more than one edge can join the same vertices. Edges having the same end vertices are called *parallel edges*[3]. If $e_k = (v_i, v_i)$, i.e. the end vertices are the same, then e_k is called *self-loop*.

Definition 1.2 (Multigraph) *A graph G containing parallel edges and/or self-loops is a multigraph.*

[3] Also called double edges.

A graph having no parallel edges and self-loops is called *simple graph*.

The number of vertices in G is called the *order*, written as $|V|$; its number of edges is given as $|E|$. A graph of order 0 is called an *empty graph*[4], and of order 1 is simply called *trivial graph*[5]. A graph is *finite* or *infinite* based on its order. In this document all the graphs used are finite and not empty, if not otherwise stated.

The usual way to visualize graphs is by drawing a circle (or a dot) for each vertex and a line connecting these dots (as it was done by Euler), and in oriented graphs an arrow depicts the order of vertices (Figure 1.2). Just how these dots and lines are visualized is not important, what is relevant is the information which vertices are paired with which edge. In the graph in Figure 1.2, edges e_4 and e_5 are parallel edges; edge e_7 is the self loop. Note that in non-oriented graphs the order of vertices defining the edge does not matter, for example the edge $e_1 = (v_1, v_2)$ could have been defined also as $e_2 = (v_2, v_1)$ (see Figure 1.2a), whereas in oriented graphs the order of vertices defines the edge as well, i.e. $e_2 = (v_2, v_1)$; the edge (v_1, v_2) does not exist in Figure 1.2b.

Definition 1.3 (Vertex neighbors) *Two vertices v_i and v_j are neighbors or adjacent if they are the end vertices of the same edge $e_k = (v_i, v_j)$.*

Definition 1.4 (Edge adjacency) *Two edges e_i and e_j are adjacent if they have an end vertex in common, say v_k, i.e. $e_i = (v_k, v_l)$ and $e_j = (v_k, v_l)$.*

For example, in the graph in Figure 1.2a v_1 and v_2 are neighbors, since they are connected by edge $e_1 = (v_1, v_2)$; edge e_1 and e_2 are adjacent since they have vertex v_1 as a common end. If all vertices of G are pairwise neighbors, then G is *complete*. A complete graph on m vertices is written as K^m.

An edge is *incident* on its end vertices. The degree of a vertex v is defined as:

Definition 1.5 (Vertex degree) *The degree (or valency) $deg(v)$ of a vertex v is the number of edges incident on it.*

The vertex of degree 0 is called *isolated*; of degree 1 is called *pendant vertex*. Note that by Definition 1.5 a self-loop at a vertex v contributes twice in the $deg(v)$. For example in the graph of Figure 1.2a $deg(v_1) = 2$; $deg(v_3) = 4$; $deg(v_5) = 3$ and so on.

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs:

Definition 1.6 (Subgraph) *$G' = (V', E')$ is a subgraph of G ($G' \subseteq G$) if $V' \subseteq V$ and $E' \subseteq E$.*

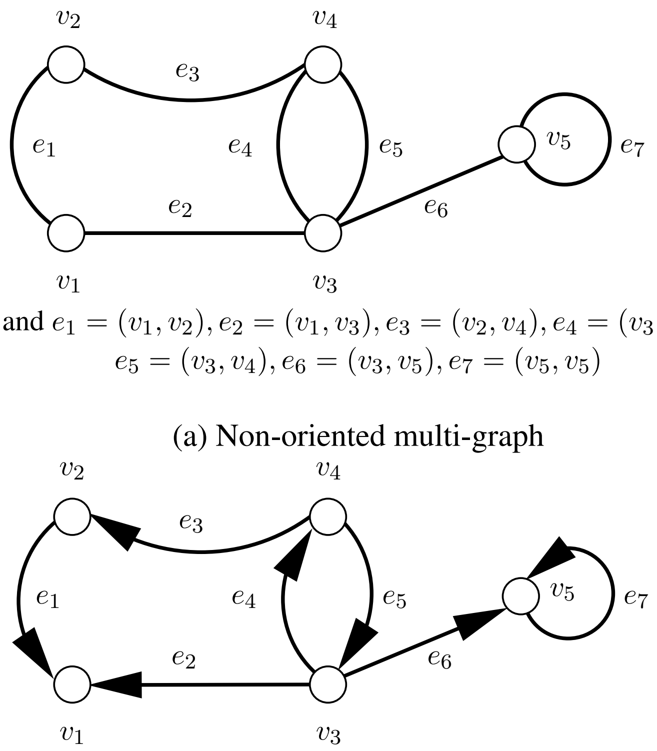
I.e. the graph G contains graph G', graph G is called also a supergraph of G ($G \supseteq G'$). If either $V' \subset V$ or $E' \subset E$, the graph G' is called a proper subgraph of G. We say sometimes that G contains G'. For example the graphs G_1 and G_2 in Figure 1.3 represent some of the subgraphs of graph G from Figure 1.2a, graph G_2 is a proper subgraph of G.

Definition 1.7 (Induced subgraph) *If $G' \subseteq G$ and G' contains all the edges of $e = (v_i, v_j) \in E$ such that $v_i, v_j \in V'$, then G' is the (vertex) induced subgraph of G and V' induces (spans) G' in G.*

[4] A graph with no vertices and hence no edges.

[5] A graph with one vertex and possibly with self-loops.

$G = (V, E) \,|\, V = \{v_1, v_2, v_3, v_4, v_5\},\ E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$



and $e_1 = (v_1, v_2), e_2 = (v_1, v_3), e_3 = (v_2, v_4), e_4 = (v_3, v_4),$
$e_5 = (v_3, v_4), e_6 = (v_3, v_5), e_7 = (v_5, v_5)$

(a) Non-oriented multi-graph

and $e_1 = (v_2, v_1), e_2 = (v_3, v_1), e_3 = (v_4, v_2), e_4 = (v_3, v_4),$
$e_5 = (v_4, v_3), e_6 = (v_3, v_5), e_7 = (v_5, v_5)$

(b) Oriented multi-graph

Figure 1.2: Non-oriented and oriented multi-graph.

The induced subgraph is usually written as $G' = G[V']$, i.e. since $V' \subset G(V)$, then $G[V']$ denotes the graph on V' whose edges are the edges of G with both ends in V'. If not otherwise stated by induced subgraph, the vertex-induced subgraph is meant. If there are no isolated vertices in G', then G' is called the *induced subgraph of G on the edge set E'* or simply *edge-induced subgraph of G*. An example of vertex-induced subgraph is given in Figure 1.3b. Finally,

Definition 1.8 (Spanning subgraph) *If $G' \subseteq G$ and V' spans all of G, i.e $V' = V$ then G' is a spanning subgraph of G.*

The subgraph in Figure 1.3a G_1 is a spanning subgraph of G since it contains all the vertices of G.

Definition 1.9 (Maximal(minimal) subgraph) *A subgraph G' of a graph G is a maximal (minimal) subgraph of G with respect to some property Π if G' has the property Π and G' is not a proper subgraph of any other subgraph of G having the property Π.*

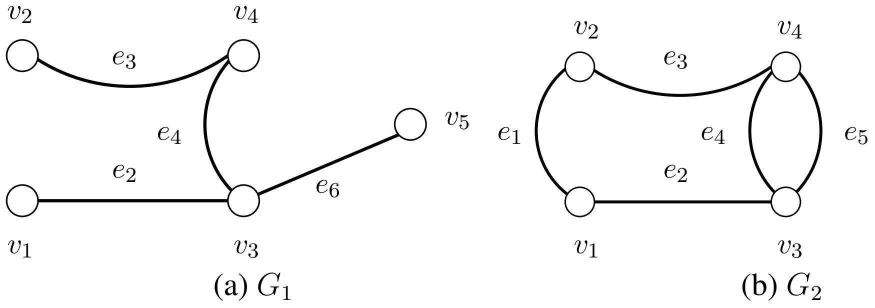


Figure 1.3: Subgraphs of graph G from Figure 1.2a.

The minimal and maximal subsets with respect to some property are defined analogously. For example in Figure 1.3b, the edge set E_2 of G_2, a vertex-induced subgraph of G, is the maximal subset of E such that the end vertices of all of its edges are in V_2 . This definition will be used later to define a component of G as a maximal connected subgraph of G, and a spanning tree of a connected G is a minimal connected spanning subgraph of G.

1.3 Paths and Cycles

Let $G = (V, E)$ be a graph with sets $V = \{v_1, v_2, \cdots\}$ and $E = \{e_1, e_2, \cdots\}$, then:

Definition 1.10 (Walk) *A walk in a graph G is a finite non-empty alternating sequence* $v_0, e_1, v_1, \cdots, v_{k-1}, e_k, v_k$ *of vertices and edges in G such that* $e_i = (v_i, v_{i+1})$ *for all* $1 \leq i \leq k$

This walk is called a $v_0 - v_k$ walk with v_0 and v_k as the terminal vertices, all other vertices are internal vertices of this walk. In a walk edges and vertices can appear more than once. If $v_0 = v_k$, the walk is *closed*, otherwise it is *open*. For the graph in Figure 1.2a an open walk could be the sequence $v_1, e_2, v_3, e_4, v_4, e_4, v_3, e_5, v_4, e_4, e_6, v_5$ and a closed walk is $v_1, e_1, v_2, e_3, v_4, e_5, v_3, e_4, v_4, e_5, e_2, v_1$.

Definition 1.11 (Trail) *A walk is a trail if all its edges are distinct.*

A trail is closed if its end vertices are the same, otherwise it is opened. By the definition the walk can contain the same vertex many times. For example the walk $v_2, e_1, e_2, v_3, e_4, v_4, e_5, v_3, e_6$ is a trail in graph shown in Figure 1.2a, even though the vertex v_3 appears twice.

Definition 1.12 (Path) *A path P is a trail where all vertices are distinct.*

A path defined thus a sequence of vertices together with a sequence of edges which allow to connect each vertex of the path to its sucessor. A simple path is defined as a sequence of vertices $v_0, v_1, v_2, \cdots, v_k$, each vertex being joint to its successor by some edge. Thus a simple path does not explicitly encode which edge allows to pass from one vertex to the next one. Note that using simple graphs two vertices are connected by at most one edge. The notion of path and simple path are thus equivalent on this graphs. This is obviously not the case with more general graphs. Note that in a multigraph a path is not uniquely defined by this nomenclature, because of possible multiple edges between two vertices. Vertices v_0 and v_k are linked by the path P, also P is called a path from v_0 to v_k (as well as between v_0 and v_k).

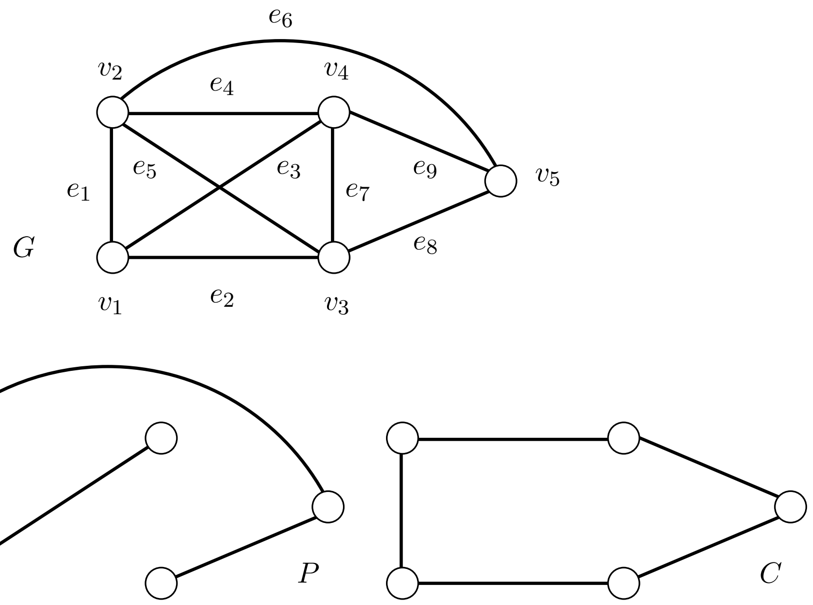


Figure 1.4: A path $P = P^4$ and a cycle $C = C^5$ in graph G.

Definition 1.13 (Path length) *The number of edges in the path is called the path length.*

The path length is denoted with P^k, where k is the number of edges in the path. An example of the path is given in Figure 1.4, and it can be written as $P = v_4, v_1, v_2, v_5, v_3$. The length of this path is 4, i.e. $P = P^4$. Note that by definition it is not necessary that a path contains all the vertices of the graph.

Analogously one defines the cycles as:

Definition 1.14 (Cycle) *A closed trail is a cycle C if all its vertices except the end vertices are distinct.*

Cycles, like paths, are denoted by the cyclic sequence of vertices $C = v_0, v_1, \cdots, v_k, v_0$. The length of the cycle is the number of edges and it is called k-cycle written as C^k. The minimum length of a cycle in a graph G is the girth $g(G)$ of G, and the maximum length of a cycle is its circumference. In Figure 1.4a cycle C^5 is shown. Note that the *girth* of graph G in Figure 1.4 is $g(G) = 3$. The distance between two vertices v and w in G denoted by $d(u, w)$, is the length of the shortest path between these vertices. The *diameter* of G, $diam(G)$ is the maximum distance between any two vertices of G.

From the above one can note the following properties of paths and cycles:

- in a path the degree of each vertex is 2, except for the end vertices for which the degree is 1,

- in a cycle the vertex degree of each vertex is 2, and

- in a path the number of edges is one less than the number of vertices; in a cycle the number of edges and of vertices are equal.

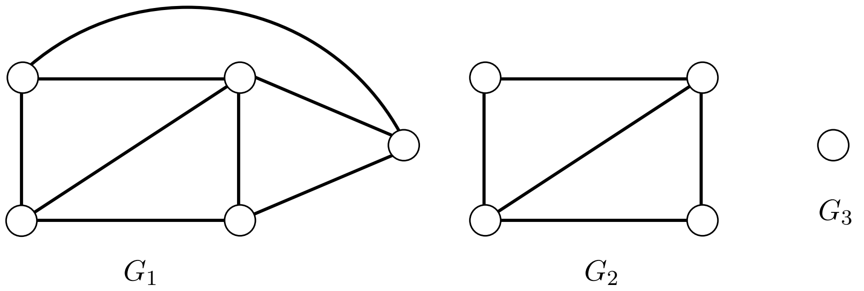


Figure 1.5: A non-connected graph G and its components G_1, G_2 and G_3.

1.4 Connectivity and Graph Components

The connectivity is an important concept in graph theory and it is one of the basic concept used in this document. Two vertices v_i and v_j are connected in a graph $G = (V, E)$ if there is a path $v_i - v_j$ in G. A vertex is connected to itself.

Definition 1.15 (Connectivity) *A non-empty graph is connected if any two vertices are joint by a path in G.*

In Figure 1.5 graphs G_1, G_2 and G_3 are connected graphs.

Let graph $G = (V, E)$ be a non-connected graph. The set partitioning is defined:

Definition 1.16 (Set partitioning) *A set V is partitioned into subsets V_1, V_2, $\cdots$, V_p if $V_1 \cup V_2 \cup \cdots \cup V_p = V$ and for all i and j, $i \neq j$ $V_i \cap V_j = \emptyset$. $\{V_1, V_2, \cdots, V_p\}$ is called a partition of V.*

Since the graph G is not connected, the vertex set V can be partitioned into subsets V_1, V_2, $\cdots$, V_p, and each vertex induced subgraph $G[V_i]$ is connected, then there exist no path between a vertex in subset V_i and a vertex in V_j, $j \neq i$.

Definition 1.17 (Component) *A maximally connected subgraph of G is called a component of graph G.*

A component of G is not a proper subgraph of any other connected subgraph of G. An isolated vertex is considered to be a component, since it is connected to itself, by definition. Note that a component is always non-empty, and that if a graph G is connected then it has only one component, i.e. itself. Figure 1.5 shows a non-connected graph G, with its components G_1, G_2 and G_3.

The following theorem is used in the Section **??** to show that after the edge removal from the cycle the graph stays connected.

Theorem 1.1 *If a graph $G = (V, E)$ is connected, then the graph remains connected after the removal of an edge e of a cycle $C \in E$, i.e. $G' = (V, E - \{e\})$ is connected.*

Proof: Suppose that removing edge e of a cycle $\mathcal{C}$ disconnects graph G' into two graphs[6] say $G^\dagger$ and $G^\ddagger$. This implies that there is no path between the vertices of $G^\dagger$ and of $G^\ddagger$. By

[6] an edge with this properties is called a bridge

definition, a cycle C is a closed trail, therefore there are always two paths joining the vertices of the cycle. Therefore there must be at least another edge e'' between vertices of $G^{\dagger}$ and $G^{\ddagger}$ if $e \in C$. This contradicts that graph G' is disconnected. $\square$

From the above theorem one can conclude that edges that disconnect a graph do not lie on any cycle.

The definition of cut and cut-set are as follows. Let $\{V_1, V_2\}$ be partitions of the vertex set V of a graph $G = (V, E)$.

Definition 1.18 (Cut) *The set $\mathcal{K}(V_1, V_2)$ of all edges having one end in one vertex partition (V_1) and the other end on the second vertex partition (V_2) is called a cut.*

Definition 1.19 (Cut-set) *A cut-set $\mathcal{K}_S$ of a connected graph G is a minimal set of edges such that its removal from G disconnects G, i.e. $G - \mathcal{K}_S$ is disconnected.*

If the induced subgraphs of G on vertex set V_1 and V_2 are connected then $\mathcal{K} = \mathcal{K}_S$. If the vertex set $V_1 = \{v\}$, the cut is denoted by $\mathcal{K}(v)$. For example the removal of the set of edges $K_1 = \{e_6, e_8, e_9\}$ from the graph shown in Figure 1.4 is a cut-set as well as a cut, since it is minimal and disconnects the graph into two connected components (by definition an isolated vertex is connected). The set of edges $K_2 = \{e_3, e_5, e_6, e_8, e_9\}$ also disconnects the graph into two components but it is not minimal since K_1 is the a proper subset of K_2 .

1.5 Trees and Forests

Trees as simple graph structure, are the most common structure used. Before the definition of the tree is given, a definition of the acyclic graph is required.

Definition 1.20 (Acyclic graph) *A graph G is acyclic if it has no cycles.*

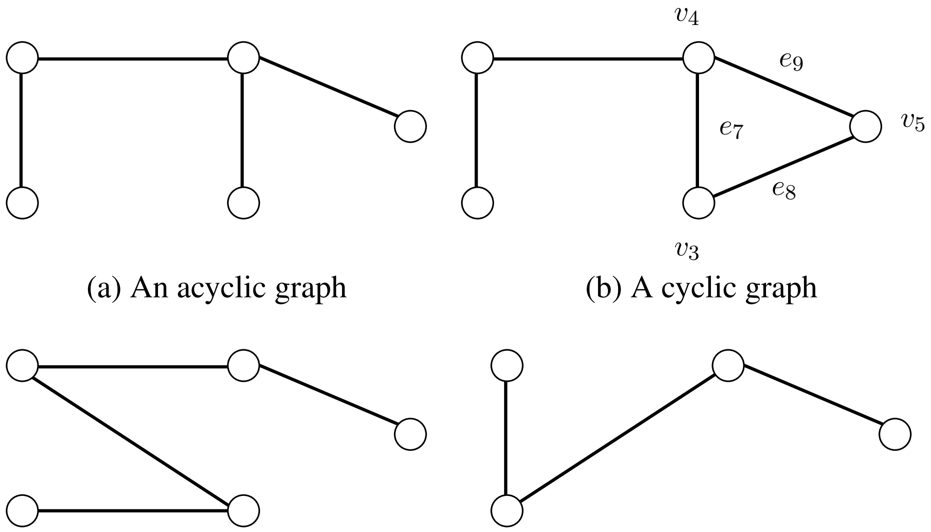
A simple example is shown in Figure 1.6a, whereas the graph under (b) in the same figure is a cyclic graph since it contains a cycle $(v_3, e_7, v_4, e_9, v_5, e_8)$.

Definition 1.21 (Tree) *A tree of graph G is a connected acyclic subgraph of G.*

The vertices of degree 1 in a tree are called *leaves*, and all edges are called branches. A nontrivial tree has at least two leaves and a branch, for example the simplest tree with two vertices joined by an edge. Note that an isolated vertex is by the definition an acyclic connected graph, therefore a tree. In Figure 1.6a, (c) and (d) examples of a trees are shown.

Definition 1.22 (Spanning tree) *Spanning tree of graph G is a tree of G containing all the vertices of G.*

Edges of the spanning tree are called *branches*. The subgraph G' containing all vertices of G and only those edges not in the spanning tree, is called *cospanning tree*, these edges are called *cords*. Note that a cospanning tree may not be connected. In Figure 1.6a and (c) are depicted two spanning trees of the graph G from Figure 1.4. An acyclic graph with k connected components is called a k-tree [Thulasiraman and Swamy, 1992]. Each connected component of a k-tree is a tree by itself. If the k-tree is a spanning subgraph of G, then it is called a spanning k-tree of G.



(a) An acyclic graph

(b) A cyclic graph

(c) A spanning tree

(d) A tree

(a) and (c) Trees of G

(b) and (d) Spanning trees G

Figure 1.6: Trees and spanning trees of the graph G from Figure 1.4.

Definition 1.23 (Forest) *A forest F of a graph G is a spanning k-tree of G, where k is the number of component of F.*

In other words a forest is a set of trees. In Figure 1.7, two examples of forest are shown, (a) with two and in (b) three components, i.e. trees, and span all the vertices of graph G. Note that the trees shown in Figure 1.6a and (c) are also forest containing only one component, the tree shown in (d) is not a forest since it does not span all the vertices of G. A forest is simply a set of trees, spanning all the vertices of G.

A connected subgraph of a tree T is called a subtree of T. If T is a tree then there is exactly one unique path between any two vertices of T. For a tree T one can also say that it is

- minimally connected, i.e. T is connected but $T - e$ is disconnected for every $e \in T$; and

- maximally acyclic, i.e. T is acyclic but $T + e$ is cyclic, for any two non-adjacent vertices $v_i, v_j \in T$ such that $e = (v_i, v_j)$.

The proof of these assertion is found in [Thulasiraman and Swamy, 1992]. Note that spanning tree and forest are synonymous if the graph has only one component.

1.6 Operations on Graphs

In this section shortly some basic binary and unary operations on graphs are described. Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. Three basic binary operation on two graphs are:

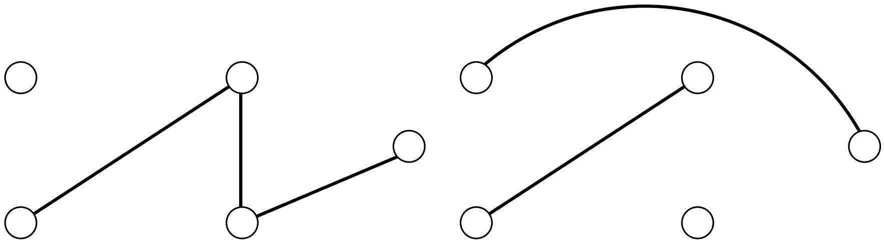

(a) Two components (b) Three components

Figure 1.7: Examples of forest of G from Figure 1.4.

Union and Intersection

The *union* of G and G' is the graph $G'' = G \cup G' = (V \cup V', E \cup E')$, i.e. the vertex set of G'' is the union of V and V', and the edge set is the union of E and E', respectively. The *intersection* of G and G' is the graph $G'' = G \cap G' = (V \cap V', E \cap E')$, i.e. the vertex set of G'' has only those vertices present in both V and V', and the edge set contains only those edges present in both E and E'', respectively. An example in Figure 1.8a of union and (b) of intersection of two graphs is given.

Symmetric Difference

The *symmetric difference*[7] between two graphs G and G', written as $G \oplus G'$, is the induced graph G'' on the edge set $E \boxplus E' = (E \setminus E') \cup (E' \setminus E)$[8], i.e. this graph has no isolated vertices and contains edges present either in G or in G' but not in both. In Figure 1.8 an example of the ring sum between two graphs is given.

The four unary operations on a graph are:

Vertex Removal

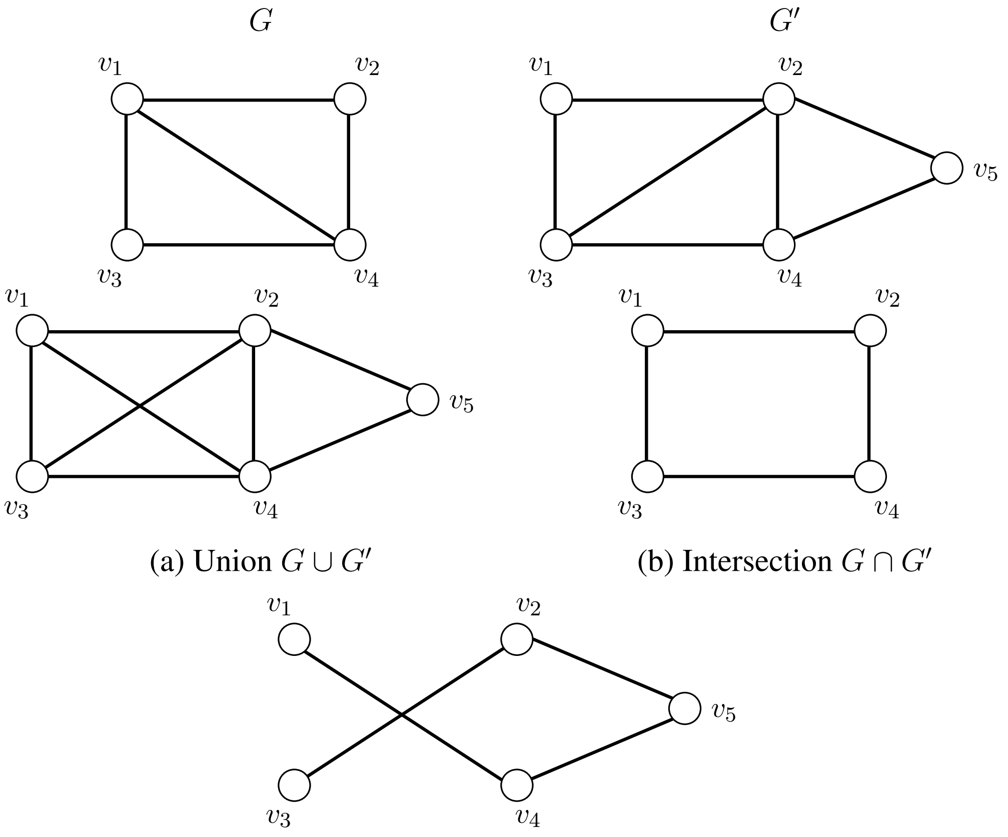
Let $v_i \in G$, then $G - v_i$ is the induced subgraph of G on the vertex set $V - v_i$; i.e. $G - v_i$ is the graph obtained after removing the vertex v_i and all the edges $e_j = (v_i, v_j)$ incident on v_i. The removal of a set of vertices from a graph is done as the removal of single vertex in succession. An example of vertex removal is shown in Figure 1.9a.

Edge Removal

Let $e \in G$, then $G - e$ is the subgraph of G after removing the edge e from E. The end vertices of the edge $e = (v_i, v_j)$ are not removed. The removal of a set of edges from a graph is done as the removal of single edge in succession. An example of edge removal is shown in Figure 1.9b.

[7] Called also ring sum.

[8] Where $\setminus$ is the set minus operation and is interpreted as removing elements from X that are in Y.



(a) Union $G \cup G'$

(b) Intersection $G \cap G'$

(c) Symmetric difference $G \oplus G'$

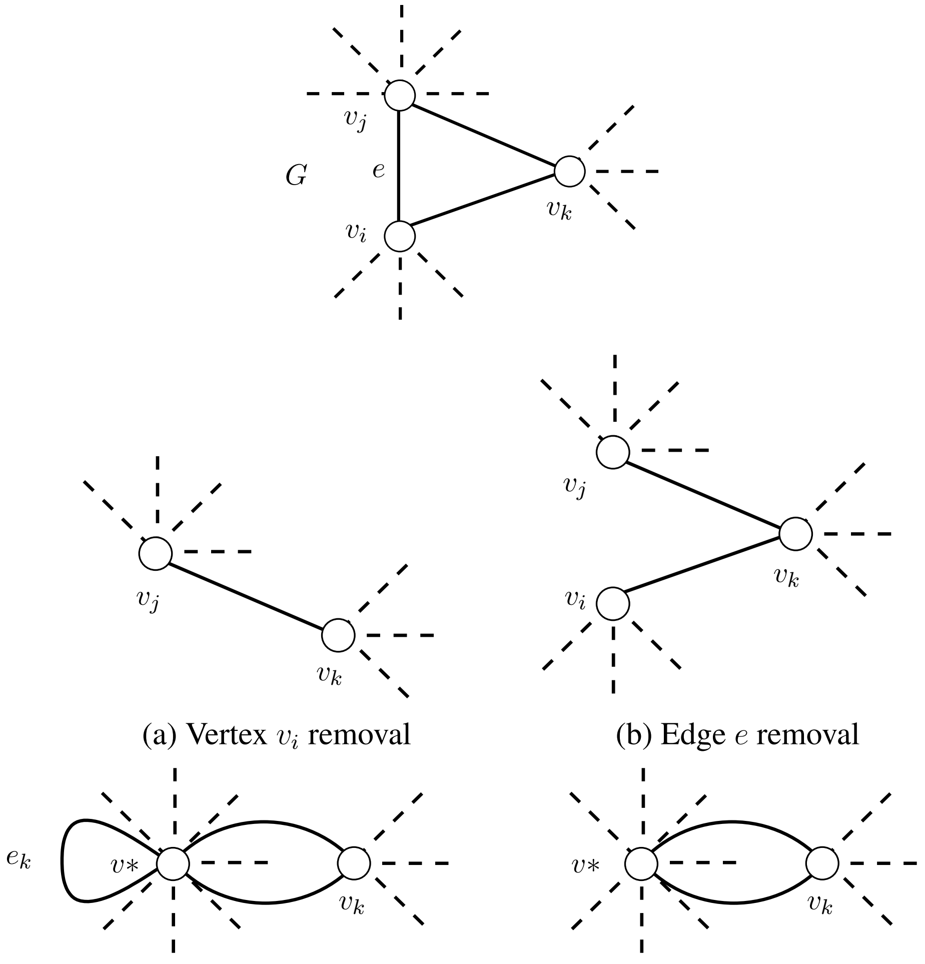
Figure 1.8: Binary graphs operations.

Vertex Identifying

Let v_i and v_j be two distinct vertices of graph G joined by the edge $e = (v_i, v_j)$. Two vertices v_i and v_j are identified if they are replaced by a new vertex $v*$ such that all the edges incident on v_i and v_j are now incident on the new vertex $v*$. An example of vertex identifying is given in Figure 1.9c.

Edge Contraction

Let $e = (v_i, v_j) \in G$ be the edge with distinct end points $v_i \neq v_j$ to be contracted. The operation of an edge contraction denotes the removing of the edge e and identifying its end vertices v_i and v_j into a new vertex $v*$. If the graph G' results from G after contracting a sequence of edges, than G is said to be *contractible* to a graph G'. Note the difference between vertex identifying and edge contraction, in Figure 1.9c and (d). Vertex identifying preserves the edge e_k, whereas edge contraction first removes this edge. In Chapter **??**, Section **??** a detailed treatment of edge contraction and edge removal in the dual graphs context is given.



(a) Vertex v_i removal

(b) Edge e removal

(c) Identifying v_i with v_j

(d) Contracting edge e

Figure 1.9: Operations on graph.

1.6.1 Homeomorphism

Let $e = (u, v)$ and $e' = (v, u')$ be the only edges incident on a vertex v. Removal of the vertex v and replacing e and e' by the edge (u, u') is called *series merger*. Adding a new vertex v on an edge (u, u') creating edges (u, v) and (v, u') is called *series insertion*.

Definition 1.24 (Isomorphic graphs) *Two graphs $G = (V, E)$ and $G' = (V', E')$ are isomorphic if there exist a bijection $\beta : V \rightarrow V'$ with $(u, u') \in E \Leftrightarrow (\beta(u), \beta(u')) \in E'$ for all $u, u' \in V$, and it is written as $G \simeq G'$.*

The β is called an isomorphism. If the graphs are identical, i.e. $G = G'$, β is called automorphism.

Definition 1.25 (Homeomorphic graphs) *Two graphs are homeomorphic if they are isomorphic and can be made isomorphic by repeated series of insertion and/or mergers.*

Note that if the graph G is planar then any homeomorphic graph to G is also planar.

1.7 Vector Spaces on Graphs

The duality concept of planar graphs is easily defined and explained using the vector spaces on graphs. The vector space on graphs is used to give a definition of dual graphs in Chapter **??**, Section **??**, Definition **??**, which is on the other hand, used to prove an important property of dual graphs with respect to the edge contraction and removal operation in Chapter **??**, Section **??**, Theorem **??**, which states that graphs during the process of dual graph contraction stay planar and are duals.

In this section the necessary definitions are given for building a vector space over a graph. Let $G = (V, E)$ be a graph with at least one edge, and let the set of all subsets of E (called also the power set of E) including $\emptyset$ be denoted by $\mathcal{E}_G$. We use 2^E as nomenclature for the power set[9]. Therefore we write $\mathcal{E}_G = 2^E \cup \emptyset = \{E_i | \forall i = 0, \ldots, 2^{|E|}\, E_i \in 2^E, i \neq j \Rightarrow E_i \neq E_j\}$. It follows the prove that $\mathcal{E}_G$ under the operation of addition $\boxplus$ and of multiplication $\boxdot$ is a vector space over the field $\mathbb{F}_2 = \{0, 1\}$ (see Appendix **??** for more details on how to build a vector space in general). Let the operation of addition $\boxplus$ be defined as the symmetric difference between two sets (see Section 1.6), and the scalar multiplication operation as:

$$1 \boxdot E_i = E_i, \tag{1.3}$$

and

$$0 \boxdot E_i = \emptyset. \tag{1.4}$$

for any $E_i \in \mathcal{E}_G$. The symmetric difference ($\boxplus$) of any two elements E_i and E_j of $\mathcal{E}_G$ is $E_k = E_i \boxplus E_j = (E_i - E_j) \cup (E_j - E_i)$ and it must be an element of the collection of all subsets of E, i.e. $E_k \in \mathcal{E}_G$, therefore $\mathcal{E}_G$ is closed under the addition operation. The associativity also holds for all E_i, E_j and E_k in $\mathcal{E}_G$ since $(E_i \boxplus E_j) \boxplus E_k = E_i \boxplus (E_j \boxplus E_k)$. Can be easily proved using some set algebra or Venn diagrams and using $(E_i - E_k) \cup (E_k - E_i) = (E_i \cup E_k) \cap (E_i^T \cup E_k^T)$, where E^T consist of all the edges not in E and it is called the complement of E. The commutative property $E_i \boxplus E_j = E_j \boxplus E_i$ is proved in the same manner. For any element in $\mathcal{E}_G$ there is an identity element

$$E_i \boxplus \emptyset = E_i, \tag{1.5}$$

and an inverse one

$$E_i \boxplus E_i = \emptyset. \tag{1.6}$$

Therefore it is proved that $\mathcal{E}_G$ is an abelian group under the addition operation $\boxplus$. Let a, b be elements in the field $\mathbb{F}_2 = \{0, 1\}$ with the additive identity element $e_{\mathbb{F}_{2\oplus}} = 0$ and multiplicative identity element $e_{\mathbb{F}_{2\odot}} = 1$. Let the addition and multiplication in field $\mathbb{F}$ be defined as modulo 2 addition ($\oplus$) and modulo 2 multiplication ($\odot$) (see Table **??** in Appendix **??** for details). E_i and E_j are elements in $\mathcal{E}_G$, and a and b are scalars from $\mathbb{F}_2$. From the definition of the additive and multiplicative operation one can prove that the other axioms needed for a vector space are satisfied:

[9] Sometimes $\mathcal{P}(E)$ is used.

1. $(a \oplus b) \boxdot E_i = (a \boxdot E_i) \boxplus (b \boxdot E_i)$,
2. $a \oplus (E_i \boxplus E_j) = (a \boxdot E_i) \boxplus (a \boxdot E_j)$,
3. $(a \odot b) \boxdot E_i = a \boxdot (b \boxdot E_i)$, and
4. $e_{\mathbb{F}_{2\odot}} \boxdot E_i = 1 \boxdot E_i = E_i$

All necessary axioms needed for a vector space are proved, hence $\mathcal{E}_G$ is a vector space over the field $\mathbb{F}_2$, more precisely it is an *edge space*. If the set $E = \{e_1, e_2, \ldots, e_n\}$ than the vector set $\mathcal{B} = \{\{e_1\}, \{e_2\}, \ldots, \{e_n\}\}$ constitutes the basis of $\mathcal{E}_G$ and this space has the dimension $n = |E|$. Every element of $\mathcal{E}_G$ can be expressed as the linear combination of elements of $\mathcal{B}$ with scalars from $\mathbb{F}_2$. The element of the edge space can be interpreted as functions of the form $E \rightarrow \mathbb{F}_2$. In an analogous way a vertex space $\mathcal{V}_G$ over the field $\mathbb{F}_2$ can be build as a vector space of all function $V \rightarrow \mathbb{F}_2$, $\mathcal{V}_G$ can be considered as the power set over V, the sum is defined as the (vertex) symmetric difference. The zero vector in $\mathcal{V}_G$ is the empty (vertex) set, and inverse of $V_i \in \mathcal{V}_G$ is the V_i itself. If the set $V = \{v_1, v_2, \ldots, v_m\}$, then set $\{\{v_1\}, \{v_2\}, \ldots, \{v_m\}\}$ is the basis of vertex space $\mathcal{V}_G$, hence its dimension is $m = |V|$. The discussion continues only on the edge vector space afterward and the terms vector space and edge space are interchanging.

From the definition of the symmetric difference operators in Section 1.6, the symmetric difference between two edge-induced subgraphs is the same as the symmetric difference of their edge sets, it follows that the set of all edge-induced subgraphs of G is a vector space over $\mathbb{F}_2$ if the operations of scalar multiplication is defined as:

$$1 \boxdot G_i = G_i, \tag{1.7}$$

and

$$0 \boxdot G_i = \emptyset. \tag{1.8}$$

where $\emptyset$ represents the empty graph.

The duals of the plane graph are easily defined using concept of cycle and cut subspaces of $\mathcal{E}_G$. Let the cycle subspace $\mathcal{C}_G$ represent the set of all cycles (including the empty graph) in G; and the cut subspace $\mathcal{K}_G$ represent the set of all cuts (including the empty graph) in $G = (V, E)$. We show, that $\mathcal{C}_G$ and $\mathcal{K}_G$ are subspaces in $\mathcal{E}_G$.

Proposition 1.1 *The set of all cycles $\mathcal{C}_G$, is a subspace of the vector space $\mathcal{E}_G$ of G.*

Proof: The proof is due to [Thulasiraman and Swamy, 1992]. The idea of the proof is as follows. $\mathcal{C}_G$ is subspace of $\mathcal{E}_G$, if one prove that for all $C, C' \in \mathcal{C}$ also $C \boxplus C'$ is in $\mathcal{C}_G$. From the definition of the cycles, every vertex is of degree 2, therefore $\mathcal{C}_G$ may be considered as the edge-induced subgraphs of G, in which all vertices are of degree even. Let C and C' be in $\mathcal{C}_G$, be edge-induced subgraphs with the degree of all their vertices even. Let $C'' = C \boxplus C'$. In other words we should prove that every vertex in C'' is of even degree. Consider a vertex $v \in C''$. This vertex must be present in at least one of subgraphs C and C'. Let X, X' and X'' denote the set of edges incident to v in C, C' and C'', respectively, and $deg(v_X)$, $deg(v_{X'})$ and $deg(v_{X''})$ the degree of v in C, C' and C''. From above $deg(v_X)$ and $deg(v_{X'})$ are even (and one of them may be zero), and hence $deg(v_{X''})$ is nonzero. Since $C'' = C \boxplus C''$ follows that $X'' = X \boxplus X'$. Hence $deg(v_{X''}) = deg(v_X) + deg(v_{X'}) - 2deg(X \cap X')$, where $deg(X \cap X')$ is the contribution

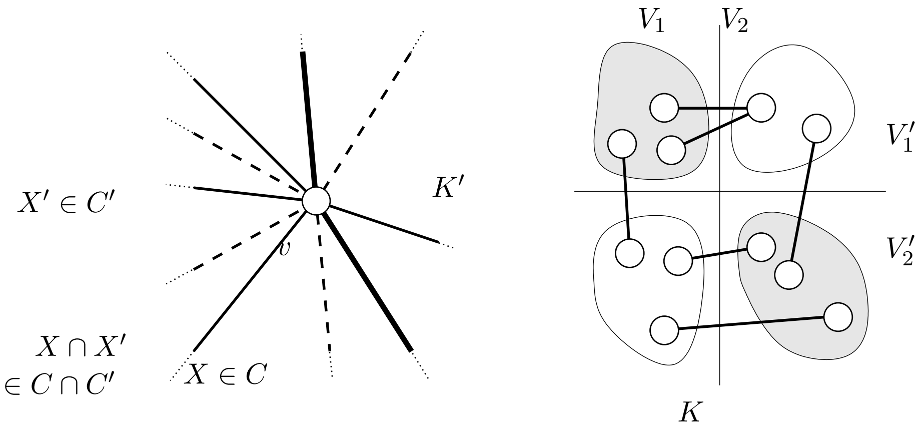


Figure 1.10: Cycle vertex in $C \boxplus C'$ and cut edges in $K \boxplus K'$.

to the degree of edges in $X \cap X'$ (see Figure 1.10a) and note that these edges are counted twice first in $deg(v_X)$ and then $deg(v_{X'})$). $deg(v_{X''})$ is even since $deg(v_X)$ and $deg(v_{X'})$ are even i.e. the degree of v in C'' is even. Since this is $\forall v \in C''$, it follows that C'' is in $\mathcal{C}_G$. □

Proposition 1.2 *The set of all cuts $\mathcal{K}_G$, is a subspace of the vector space $\mathcal{E}_G$ of G.*

Proof: The proof is due to [Diestel, 1997]. To prove that $\mathcal{K}_G$ is subspace of $\mathcal{E}_G$, one must prove that for all $K, K' \in \mathcal{K}_G$ also $K \boxplus K'$ is in $\mathcal{K}_G$. Since $K \boxplus K = \emptyset \in \mathcal{C}$, and $K \boxplus \emptyset = K \in \mathcal{C}$ (by the definition of symmetric difference), it is assumed that K and K' are non-empty and distinct. Let $\{V_1, V_2\}$ and $\{V_1', V_2'\}$ be the partition of the set V. Note that $V_1 \cup V_2 = V_1' \cup V_2' = V$. Then $K \boxplus K'$ consists of all the edge that cross one of these partitions but not the other (see Figure 1.10). These are precisely the edges between $(V_1 \cap V_1') \cup (V_2 \cap V_2')$ and $(V_1 \cap V_2') \cup (V_2 \cap V_1')$, and by $K \neq K'$ these two sets form another partition of V. Hence $K \boxplus K' \in \mathcal{K}_G$ and $\mathcal{K}_G$ is a subspace of $\mathcal{E}_G$. □

Corollary 1.1 *The space $\mathcal{K}_G$ is created by cuts of form $\mathcal{K}(v)$.*

Proof: Due to [Diestel, 1997]. Every edge $e = (v_1, v_2) \in E$ lies in two cuts $\mathcal{K}(v_1)$ and $\mathcal{K}(v_2)$, thus every partition $\{V_1, V_2\}$ satisfies $\mathcal{K}_G = \sum_{v \in V_1} \mathcal{K}(v)$, where the sum is over the operator $\boxplus$. □

Let the concept of (vector) edge space be clarified with a simple example of the graph $G = (V, e)$ given in Figure 1.11. For the given edge set E, the set power is:

$$\begin{aligned}\mathcal{E}_G = \{&\emptyset, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\},\\ &\{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_1, e_5\}, \{e_2, e_3\}, \{e_2, e_4\}, \{e_2, e_5\}, \{e_3, e_4\}, \{e_3, e_5\}, \{e_4, e_5\},\\ &\{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_1, e_2, e_5\}, \{e_1, e_3, e_4\}, \{e_1, e_3, e_5\}, \{e_1, e_4, e_5\}, \{e_2, e_3, e_4\},\\ &\{e_2, e_3, e_5\}, \{e_2, e_4, e_5\}, \{e_3, e_4, e_5\},\\ &\{e_1, e_2, e_3, e_4\}, \{e_1, e_2, e_3, e_5\}, \{e_1, e_2, e_4, e_5\}, \{e_1, e_3, e_4, e_5\}, \{e_2, e_3, e_4, e_5\},\\ &\{e_1, e_2, e_3, e_4, e_5\}\}.\end{aligned}$$

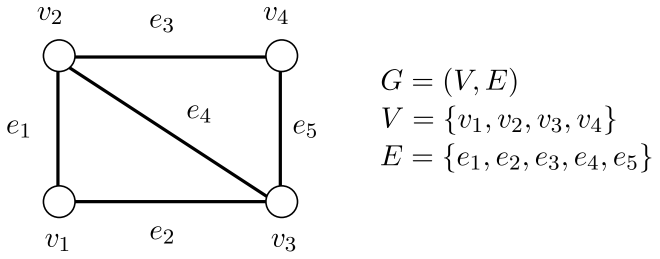

v_2
e_3
v_4
e_1
e_4
e_5
v_1
e_2
v_3
$G = (V, E)$
$V = \{v_1, v_2, v_3, v_4\}$
$E = \{e_1, e_2, e_3, e_4, e_5\}$

Figure 1.11: Edge space on G is $\mathcal{E}_G : E \to \mathbb{F}_2 = \{0, 1\}$.

Altogether $\mathcal{E}_G$ contains $2^{|E|} = 2^5 = 32$ subsets. Let the vector addition $\boxplus$ and scalar multiplication $\boxdot$, as well as the identity and inverse elements for the $\boxplus$ and $\boxdot$ respectively, be defined as described above in this section. It is easily shown that $\mathcal{E}_G$ is an edge space over the field $\mathbb{F}_2$. The set $\mathcal{B} = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$ is the basis of the edge space, understood this vectors are linearly independent. This space has the dimension 5. Every element of the set $\mathcal{E}_G$ can be represented as the linear combination of the elements of $\mathcal{B}$, e.g.

$$\{e_1, e_2, e_5\} = (1 \boxdot \{e_1\}) \boxplus (1 \boxdot \{e_2\}) \boxplus (0 \boxdot \{e_2\}) \boxplus (0 \boxdot \{e_3\}) \boxplus (0 \boxdot \{e_4\}) \boxplus (1 \boxdot \{e_5\}) =$$

based on the definition of the scalar multiplication $\boxdot$, it follows:

$$= \{e_1\} \boxplus \{e_2\} \boxplus \emptyset \boxplus \emptyset \boxplus \emptyset \boxplus \{e_5\} = \{e_1, e_2, e_3\}.$$

The set of all cycles for the graph G given is

$$\mathcal{C}_G = \{\emptyset, \{e_1, e_2, e_4\}, \{e_3, e_4, e_5\}, \{e_1, e_2, e_3, e_5\}\},$$

with the base $\mathcal{B}_\mathcal{C} = \{\{e_1, e_2, e_4\}, \{e_3, e_4, e_5\}\}$, and the set of all cuts in G is

$$\mathcal{K}_G = \{\emptyset, \{e_3, e_5\}, \{e_1, e_2\}, \{e_1, e_4, e_5\}, \{e_1, e_3, e_4\}, \{e_3, e_4, e_2\}, \{e_2, e_4, e_5\}, \{e_1, e_2, e_3, e_5\}\}.$$

with the base $\mathcal{B}_\mathcal{K} = \{\{e_3, e_5\}, \{e_1, e_2\}, \{e_1, e_4, e_5\}\}$. One can easily prove that all elements of $\mathcal{C}_G$ can be described as a linear combination of vectors $\mathcal{B}_\mathcal{C}$ with scalar from $\mathbb{F}_2$, the same hold for $\mathcal{K}_G$ and $\mathcal{B}_\mathcal{K}$, respectively. E.g. a cycle $C_1 = \{e_1, e_2, e_3, e_5\} = 1 \boxdot \{e_1, e_2, e_4\} \boxplus 1 \boxdot \{e_3, e_5, e_4\} = \{e_1, e_2\} \boxplus \{e_3, e_5\}$, and a cut $K_1 = \{e_1, e_3, e_4\} = 1 \boxdot \{e_3, e_5\} \boxplus 1 \boxdot \{e_1, e_4, e_5\} = \{e_3, e_5\} \boxplus \{e_1, e_4, e_5\}$. It is shown in Appendix **??** that the set of n-vectors over $\mathbb{F}_2$, where such that each vector's element in $\mathbb{F}_2$ form also a vector space. For the enumeration of the edges E as in Figure 1.11, one writes e.g. instead of $\{e_1, e_2, e_3\}$ an 5-vector of the form $(1, 1, 1, 0, 0)$. 1 is put if the edge is present in the set and 0 otherwise. For example the set of all cycles can be written as

$$\mathcal{C}_G = \{(0, 0, 0, 0, 0), (1, 1, 0, 1, 0), (0, 0, 1, 1, 1), (1, 1, 1, 0, 1)\},$$

This form of presentation of graphs is easier to manipulate in computers, even though for humans it is hard to follow if the number of edges is large, since the vectors of the form $(\cdot, \cdot, \ldots, \cdot)$ are cumbersome.

Bibliography

[Biggs et al., 1976] Biggs, N. L., Lloyd, E. K., and Wilson, R. J. (1976). *Graph Theory 1736-1936.* Clarendon Press, Oxford. 1

[Bondy and Murty, 1976] Bondy, J. A. and Murty, U. S. R. (1976). *Graph Theory with Applications, 5th Ed.* Elevier Science Publishing Co. Inc. 2

[Christofides, 1975] Christofides, N. (1975). *Graph Theory - An Algorithmic Approach.* Academic Press, New York, London, San Francisco. 2

[Diestel, 1997] Diestel, R. (1997). *Graph Theory.* Springer, New York. 2, 16

[Euler, 1736] Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis (The solution of a problem relating to the geometry of position). *Commentarii academiae scientiarum Petropolitanae*, 8:128–140. reprinted in Opera Omnia: Series 1, Volume 7, pp. 1 - 10, 1766. 1

[Harary, 1969] Harary, F. (1969). *Graph Theory*. Addison Wesley. 2

[Thulasiraman and Swamy, 1992] Thulasiraman, K. and Swamy, M. N. S. (1992). *Graphs: Theory and Algorithms.* Wiley-Interscience. 2, 9, 10, 15

[Webster, 1913] Webster, M. (1913). *Webster's Revised Unabridged Dictionary.* Merriam Webster Inc. 2

List of Symbols and Abbreviations

If not stated otherwise in the text the symbol and abbreviations applies throughout the document.

Symbols

E	edge set
e	edge
$G = (V, E)$	graph
T	tree
V	vertex set
v	vertex
$\lvert \cdot \rvert$	set cardinality

Index

The first page number is usually, but not always, the primary reference to the indexed topic.