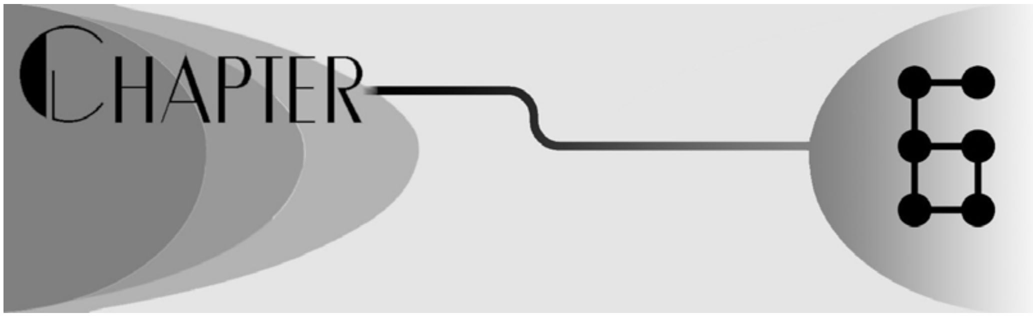

Edge Cuts and Connectivity

Chapter Goals

Define edge cuts and discuss their basic properties.

Examine edge and vertex connectivity issues.

Define separability and blocks of a graph.

Discuss the structure of 1- and 2-connected graphs.

Explore 1- and 2-isomorphisms for graphs.

6.1 Introduction

This chapter deals with a number of connectivity issues. How can we gauge the level of connectivity in a graph. One example is the number of edges or vertices we need to delete before the graph breaks into pieces. The discussion begins with a description of edge cuts in §6.2. In §6.3 we show how to generate edge cuts. §6.4 relates fundamental cycles and edge cuts. We cover the notions of edge and vertex connectivity in §6.5 and examine some practical applications of these concepts. §6.6 addresses the ideas of separability and blocks. §6.7 and §6.8 present more general isomorphisms.

6.2 Edge Cuts

In Chapter 5 we studied spanning trees, a minimal type of connected subgraph of a connected graph G. In this section we study *edge cuts* of a graph. This is another type of subgraph of a connected graph. The removal of an edge cut results in a disconnected graph. This concept is made precise in the following definition.

Definition 6.1
Let G be a connected graph. An edge cut *or simply a* cut *S of G is a set $S \subseteq E(G)$ such that the graph $G - S$ satisfies the following:*

1. *It is disconnected.*

2. *$G - S'$ is connected for any proper set $S' \subset S$.*

✍ **Remark 6.2**

1. *An edge cut S is a set of edges. The induced subgraph $G[S]$ of G consisting of the edges of S together with their endvertices is uniquely determined by S and vice versa. Hence, we sometimes talk about and view an edge cut as a subgraph of G.*

2. *If e is a cut-edge, then the set consisting of e alone is an edge cut. By Theorem 4.7 every edge of a tree is a cut-edge, so every edge of a tree forms an edge cut.*

The following example helps clarify the notion of edge cut.

☞ **Example 6.3**
In Figure 6.1 the set of edges $\{a, c, d, f\}$ is an edge cut. Some other edge cuts in this graph are $\{a, b, g\}$, $\{a, b, e, f\}$, and $\{d, h, f\}$. The set $\{i\}$ is also an edge cut. The set of edges $\{a, c, h, d\}$ is not *an edge cut because one of its proper subsets $\{a, c, h\}$ is an edge cut.*

To emphasize the fact that no proper subset of an edge cut can be an edge cut, some authors refer to an edge cut as a *minimal edge cut*. This stresses the fact that the removal of any proper subset of an edge cut does not disconnect the graph.

There is another way of looking at an edge cut. If one partitions the vertices of a connected graph G into two subsets, an edge cut is the set of edges that join

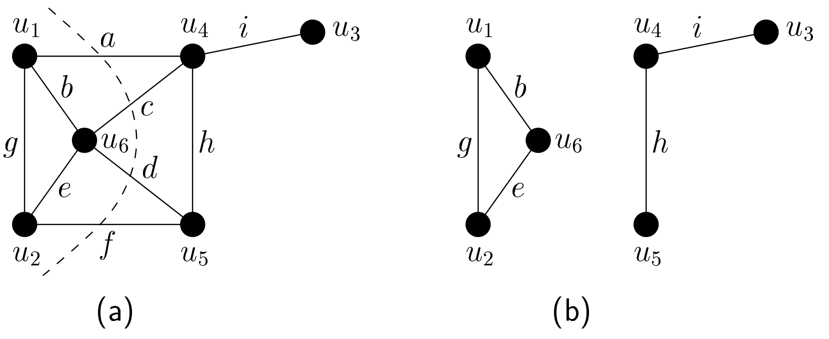

Figure 6.1: (a) Original graph. (b) Removal of the edge cut $\{a, c, d, f\}$ from the graph "cuts" it into two.

any two vertices in separate sections of the partition. Removal of these edges from G destroys all paths between these two sets of vertices. For example, in Figure 6.1(a) the edge cut $\{a, c, d, f\}$ disconnects vertex set $\{u_1, u_2, u_6\}$ from vertex set $\{u_3, u_4, u_5\}$.

Edge cuts are of great importance in studying properties of communication and transportation networks. The following example illustrates one application.

☞ **Example 6.4**

Suppose the six vertices in Figure 6.1(a) represent cities and the edges represent connection by telephone lines. We want to find out if there are any weak spots in the network that need strengthening by means of additional telephone lines. In order to do so, we need to look at all edge cuts of the graph. The edge cut with the smallest number of edges indicates the most vulnerable area. In Figure 6.1(a) the city represented by u_3 can be severed from the rest of the network by the destruction of just one edge, the cut-edge i. By adding more telephone lines to city u_3 some redundancy can be added to the system.

We shall present some more applications of edge cuts later, but first we need some basic properties.

Theorem 6.5

Let G be a connected graph.

1. *Every edge cut in G contains a branch of every spanning tree of G.*
2. *Conversely, every minimal set of edges containing a branch of each spanning tree of G is an edge cut of G.*

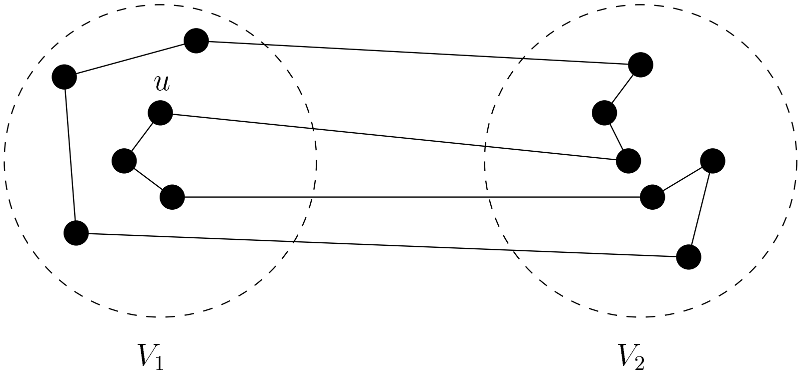

Figure 6.2: Circuit and an edge cut in G.

PROOF: Let S be an edge cut and T a spanning tree of G. If $S \cap E(T) = \emptyset$ then the edges of $G - S$ contain $E(T)$ and hence T as a subgraph. Therefore, $G - S$ is connected, contradicting the fact that S is an edge cut.

To prove the second statement, let Q be a minimal set of edges that contains a branch of each spanning tree of G. Since $G - Q$ has the same set of vertices as G and contains no spanning tree of G, it must be disconnected. On the other hand, if $e \in Q$ then by minimality of Q, there is a spanning tree T of G with no branches in $Q \setminus \{e\}$. This means that the graph $G - (Q \setminus \{e\})$ contains T as a subgraph and is therefore connected. By definition, we have that Q is an edge cut. □

We conclude this section with a theorem relating circuits to edge cuts.

Theorem 6.6
Every circuit has an even number of edges in common with any edge cut.

PROOF: Consider an edge cut S in a graph G (see Figure 6.2). Let the removal of S partition $V(G)$ into two disjoint subsets V_1 and V_2. Let C be a circuit in G. The initial and final vertex u of C is contained in either V_1 or V_2, say $u \in V_1$. Starting from u and traversing along C, we go equally often from V_1 to V_2 along an edge in C as from V_2 to V_1, since we start and end at u. Because every edge with one endvertex in V_1 and one in V_2 is contained in S, the theorem follows. □

6.3 Generating All Edge Cuts in a Graph

In Example 6.4 it was pointed out how edge cuts are used to identify weak spots in a communication network. For this purpose we list all edge cuts of the corresponding graph and find which ones have the smallest number of edges. It is apparent even from Example 6.4 that there is a large number of edge cuts in general. Hence, we must develop a systematic method to generate all relevant edge cuts.

Not only is a spanning tree is essential in defining a set of fundamental cycles, it is also essential in defining a set of *fundamental edge cuts*. It will be beneficial for you to look for the parallelism between circuits and edge cuts. We start with the following observation.

Observation 6.7
Let G be a connected graph.

1. *Let S be an edge cut of G. Then there is an unique partition*

$$V(G) = V_1(S) \cup V_2(S)$$

   of the vertex set of G such that $G[V_1]$ and $G[V_2]$ are both connected graphs.

2. *Likewise, let $V(G) = V_1 \cup V_2$ be a partition such that $G[V_1]$ and $G[V_2]$ are connected. Then there is a corresponding cut set consisting of all of the edges of G with one endvertex in V_1 and the other in V_2. We denote this cut set by $S(V_1, V_2)$.*

With this correspondence between the partitions of $V(G)$ and the edge cuts of G, we can put forth the following definition.

Definition 6.8
Let G be a connected graph. Let T be a spanning tree of G and b a branch of T. By Theorem 4.7 the subgraph $T - b$ of G has exactly two components. Therefore, it determines an unique partition $V(G) = V_1 \cup V_2$ and hence the corresponding edge cut $S = S(V_1, V_2)$ of G. Such an edge cut is called a fundamental edge cut *of G with respect to T. We denote the fundamental edge cut for a branch b by S_b.*

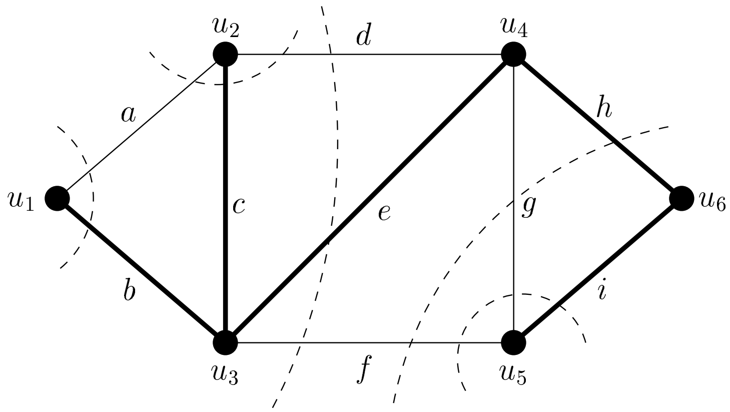

Figure 6.3: Fundamental edge cuts of a graph with respect to a fixed spanning tree shown in bold.

➤ **Note 6.9**

If G is a connected graph and T is a spanning tree of G, then by Theorem 6.5 every edge cut of G will contain at least one branch of T. An edge cut S that contains exactly one branch of T is a fundamental edge cut of G with respect to T.

☞ **Example 6.10**

Figure 6.3 shows a spanning tree T in thick lines. All five of the fundamental edge cuts with respect to T are shown as broken lines "cutting" through each edge cut. The edge cuts determined by the branches b, c, e, h, and i are as follows:

$$\begin{aligned} S_b &= \{a,b\}, \\ S_c &= \{a,c,d\}, \\ S_e &= \{d,e,f\}, \\ S_h &= \{f,g,h\}, \text{ and} \\ S_i &= \{f,g,i\}. \end{aligned}$$

Next we show how to generate a new edge cut of a graph from two given ones.

Theorem 6.11

Let G be a connected graph. Let S_1 and S_2 be two different edge cuts of G. Then the symmetric difference $S_1 \triangle S_2$ is either a new edge cut of G, or a disjoint union of edge cuts.

Proof: We sketch the proof. Let S_1 and S_2 be two edge cuts in a given connected graph G. Let $V(G) = V_1 \cup V_2$ be the unique partitioning of the vertex set of G with respect to S_1. Let $V(G) = V_3 \cup V_4$ be the unique partitioning with respect to S_2. The top two graphs in Figure 6.4 illustrate the situation.

Clearly, we have

$$\begin{aligned} V_1 \cup V_2 = V(G) \quad &\text{and} \quad V_1 \cap V_2 = \emptyset, \text{ and} \\ V_3 \cup V_4 = V(G) \quad &\text{and} \quad V_3 \cap V_4 = \emptyset. \end{aligned}$$

Now put

$$\begin{aligned} V_5 &= (V_1 \cap V_4) \cup (V_2 \cap V_3) \text{ and} \\ V_6 &= (V_1 \cap V_3) \cup (V_2 \cap V_4), \end{aligned}$$

and observe that the symmetric difference $S_1 \triangle S_2$ can only contain edges with one endvertex in V_5 and the other in V_6. Also, we note that all such edges are contained in $S_1 \triangle S_2$ (see Figure 6.4). Since

$$V_5 \cup V_6 = V(G) \text{ and } V_5 \cap V_6 = \emptyset,$$

the set of edges $S_1 \triangle S_2$ gives rise to a partitioning of $V(G)$. We can conclude that if both $G[V_5]$ and $G[V_6]$ are connected subgraphs of G, then $S_1 \triangle S_2$ is an edge cut. Otherwise, it is a disjoint union of edge cuts. □

☞ **Example 6.12**

In Figure 6.3 consider the symmetric difference of the following three pairs of edge cuts:

$$\begin{aligned} \{d, e, f\} \triangle \{f, g, h\} &= \{d, e, g, h\}, \qquad (6.1) \\ \{a, b\} \triangle \{b, c, e, f\} &= \{a, c, e, f\}, \text{ and} \qquad (6.2) \\ \{d, e, g, h\} \triangle \{f, g, i\} &= \{d, e, f, h, i\} \\ &= \{d, e, f\} \cup \{h, i\}. \qquad (6.3) \end{aligned}$$

We see that Equations (6.1) and (6.2) give new edge cuts, and Equation (6.3) provides a disjoint union of two edge cuts.

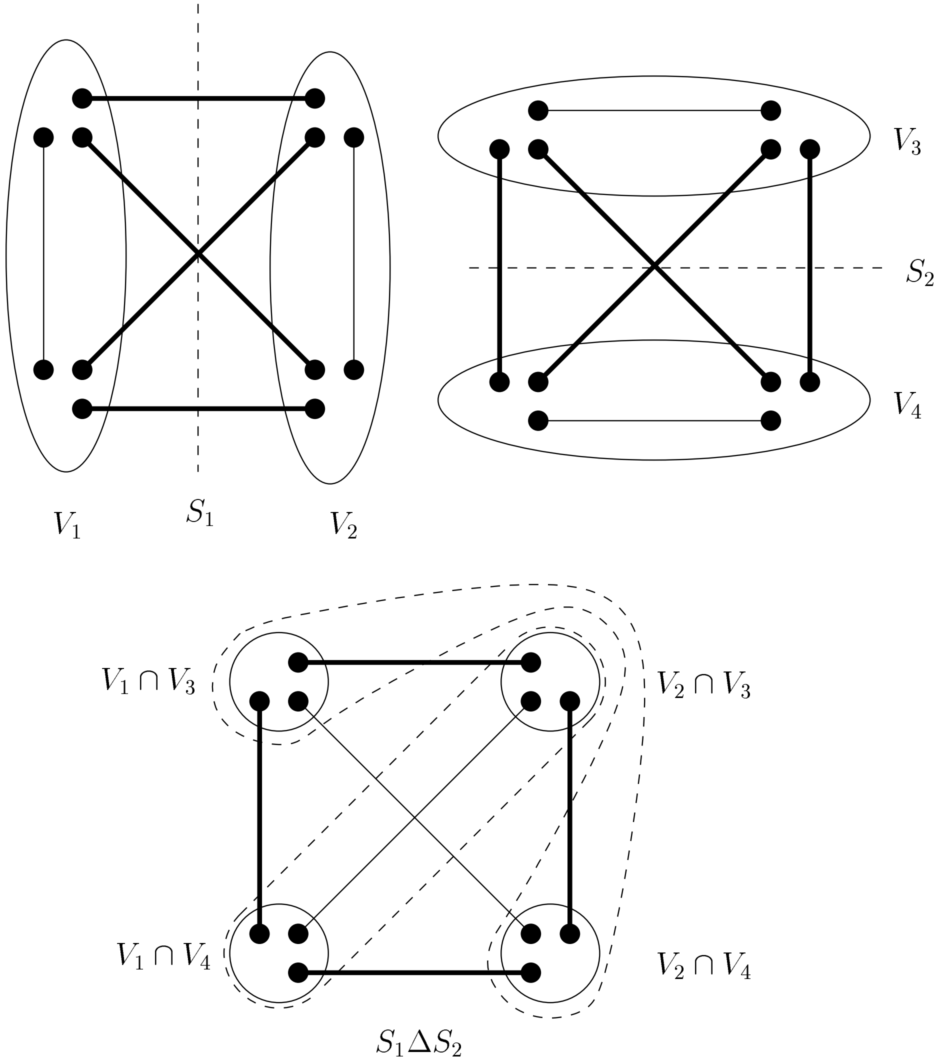

Figure 6.4: Two edge cuts and their partitionings.

We now have a method of generating additional edge cuts from a number of given edge cuts. Obviously, we cannot start with any two edge cuts in a given graph and hope to obtain all of its edge cuts by this method. So, it is natural to ask how can we describe a minimal set of edge cuts from which every edge cut of G is obtained by taking symmetric differences? Theorem 6.13 provides the answer, but the proof is in Chapter 9. Recall that the operation of symmetric difference on sets is associative.

Theorem 6.13
Let G be a connected graph and T a spanning tree of G. If $\mathcal{F}(T) = \{S_1, \ldots, S_k\}$ is the set of all fundamental edge cuts of G with respect to T, then

$$\{S_1 \vartriangle S_2 \vartriangle \cdots \vartriangle S_k\}$$

contains all edge cuts of G.

6.4 Fundamental Cycles and Edge Cuts

In this section we further explore the connection between fundamental cycles and fundamental edge cuts with respect to a fixed spanning tree of a given graph.

Theorem 6.14
Let G be a connected graph and T a fixed spanning tree of G. For a chord c of G with respect to T, let Γ_c be the fundamental cycle of G determined by c. For a branch b of G with respect to T, let S_b be the fundamental edge cut determined by b. Then

$$c \in S_b \textit{ if and only if } b \in \Gamma_c.$$

PROOF: Let c be a fixed chord of G with respect to T. If $b \in \Gamma_c$, then the branch b is also contained in the fundamental edge cut S_b. S_b determines a partition with vertices in V_1 or V_2. Let b' be another branch of T. Clearly, b' has both its endvertices in V_1 or both in V_2. Hence, b' cannot be contained in S_b (see Figure 6.5). By Theorem 6.6 Γ_c and S_b have an even number of edges in common. So besides b, the edge c is the only other possible edge common to both Γ_c and S_b. This holds for any branch b of G, which is contained in Γ_c, with respect to T. So, $c \in S_B$ if $b \in \Gamma_c$.

On the other hand, if $b \notin \Gamma_c$ then the vertices of Γ_c are all in V_1 or are all contained in V_2. Hence, the edge c is not contained in S_b. This whole argument holds for any chord c, so we have the theorem. □

The following example illustrates Theorem 6.14.

☞ **Example 6.15**
Consider the spanning tree with edges $\{b, c, e, h, i\}$ shown in thick lines in Figure 6.3. The fundamental cycle made by chord f has edges $\{f, e, h, i\}$. The

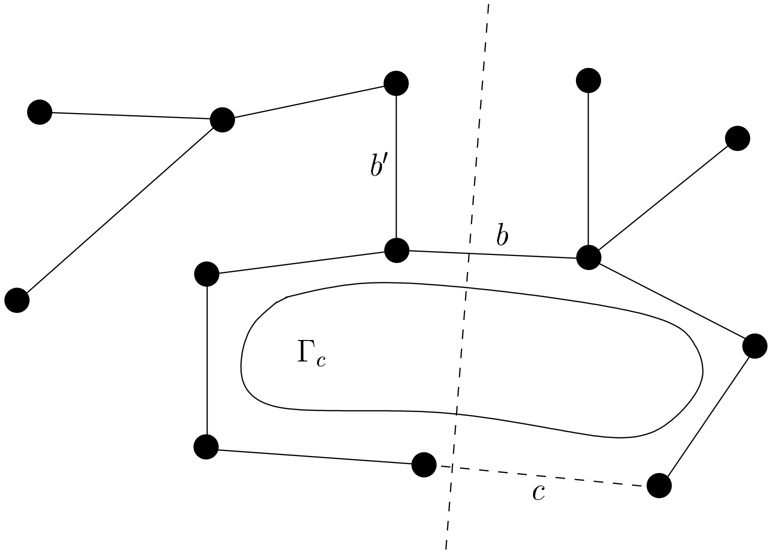

Figure 6.5: A spanning tree, a chord, and two branches.

three fundamental edge cuts determined by the three branches e*,* h*, and* i *are* $\{d, e, f\}$*,* $\{f, g, h\}$*, and* $\{f, g, i\}$*, respectively. Note that the chord* f *occurs in each of these three fundamental edge cuts and there is no other fundamental edge cut that contains* f*.*

Consider the branch e *of the spanning tree shown in Figure 6.3 with edge set* $\{b, c, e, h, i\}$*. The fundamental edge cut determined by* e *is* $\{e, d, f\}$*. The fundamental cycles determined by chords* d *and* f *are* $\{d, c, e\}$ *and* $\{f, e, h, i\}$*, respectively. The branch* e *is contained in both of these fundamental cycles, and none of the remaining fundamental cycles contains branch* e*.*

6.5 Connectivity

If a connected graph has a cut-edge, then we can disconnect the graph with the removal of one edge. We say that the *edge connectivity* of the graph is one. Clearly, this is not restricted to connected graphs. This observation motivates the following definition.

Definition 6.16
Let G *be a given graph. The smallest number of edges whose removal disconnects*

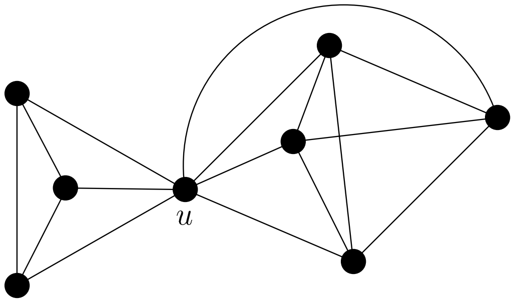

Figure 6.6: Vertex u is a cut-vertex.

G is called the edge connectivity *of G, denoted $\kappa'(G)$. We say that G is k*-edge-connected *if k is less than or equal to $\kappa'(G)$.*

Remark 6.17

Note the following about edge connectivity:

1. *If G is a disconnected graph, then $\kappa'(G)$ equals zero.*
2. *If G is connected then $\kappa'(G)$ is the size of the smallest edge cut of G.*
3. *If G is a graph and G' is the graph we get from G by removing all of its loops, then*

$$\kappa'(G) = \kappa'(G').$$

The next example illustrates edge connectivity.

Example 6.18

The edge connectivity of any tree is one. The edge connectivities of the graphs in Figures 6.1(a), 6.3, and 6.6 are one, two, and three, respectively.

If a connected graph has a cut-vertex, then the minimum number of vertices whose removal disconnects the graph is one. In Figure 6.6 the vertex u is a cut-vertex of the graph. We can generalize connectivity for vertices as well.

Definition 6.19

Let G be a graph. The minimum number of vertices whose removal disconnects G or creates a graph with a single vertex is called the connectivity *of G, denoted $\kappa(G)$. If k is less than or equal to $\kappa(G)$, then G is said to be k*-connected.

✎ Remark 6.20

A couple of observations about vertex connectivity are worth noting.

1. *If G is a disconnected graph, then $\kappa(G)$ equals zero.*
2. *For a graph G, let G' be the graph obtained from G by removing all the loops from G and changing all multiple edges to single ones. In this case we have*
$$\kappa(G) = \kappa(G').$$
3. *The set of vertices removed in order to disconnect the graph or reduce it to a single vertex is called a* vertex cut. *It is instructive to compare and contrast the notion of vertex cut with that of edge cut (see Definition 6.1).*

The next example illustrates vertex connectivity.

☞ Example 6.21

For natural numbers m and n greater than or equal to one, we have the following:

1. *For the null graph N_n, we have $\kappa(N_n)$ equals zero.*
2. *For any tree T on two or more vertices, we have $\kappa(T)$ equals one.*
3. *For any cycle C_n on three or more vertices, we have $\kappa(C_n)$ equals two.*
4. *For the complete graph K_n, we have $\kappa(K_n)$ equals $n - 1$.*
5. *For the complete bipartite graph K_{mn}, we have $\kappa(K_{mn}) = \min(\{m, n\})$. (Can you see why?)*

We will use this terminology in the following applications in allocations of stations to be connected.

☞ Example 6.22

Suppose we are given n stations that are to be connected by means of e telephone lines, bridges, railroads, tunnels, or highways, where e is greater than or equal to $n - 1$. What is the best *way to connect the stations? By best we mean that the network should be as invulnerable to the destruction of individual stations and individual lines as possible. In other words, we need to construct a graph with n vertices and e edges that has the maximum possible vertex and edge connectivity.*

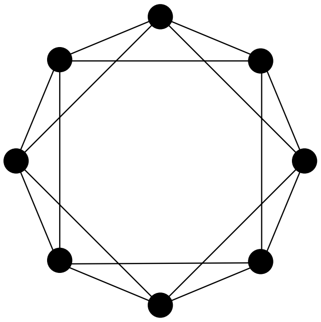

Figure 6.7: Graph with 8 vertices and 16 edges representing a network of stations.

The graph in Figure 6.6 has n equal to 8 and e equal to 16. It has vertex connectivity of one and edge connectivity of three.

Another graph having 8 vertices and 16 edges is shown in Figure 6.7. It can easily be seen that the vertex connectivity as well as the edge connectivity of this graph is four. Consequently, even after any three stations are bombed, or any three connections destroyed, the remaining stations can continue to "communicate" with each other. Thus the network of Figure 6.7 is better connected, and hence less vulnerable, than the network of Figure 6.6.

What is the highest vertex and edge connectivity we can achieve for a graph with n vertices and e edges? The next two theorems constitute the answer to this question. First, we need a definition.

Definition 6.23
For any graph G let

$$\begin{aligned}\Delta(G) &= \max(\{d_G(u) \mid u \in V(G)\}), \text{ and}\\ \delta(G) &= \min(\{d_G(u) \mid u \in V(G)\}),\end{aligned}$$

be the maximum degree and minimum degree, respectively of a vertex in G. When unambiguous, we write Δ (δ) instead of $\Delta(G)$ (respectively, $\delta(G)$).

Theorem 6.24
Let G be a graph. We have that

$$\kappa(G) \leq \kappa'(G) \leq \delta(G).$$

PROOF: Suppose u is a vertex with $d_G(u)$ equal to $\delta(G)$. We can isolate u by removing all of the edges having u as one endvertex. There are at most $d_G(u)$ edges to remove; there are exactly $d_G(u)$ if G has no loops. Hence, we have that $\kappa'(G)$ is less than or equal to $\delta(G)$.

For the other inequality, we can assume G is connected. Let S be an edge cut consisting of $\kappa'(G)$ edges. Let $V(G) = V_1 \cup V_2$ be the corresponding partition and let $n_i = |V_i|$. We can assume n_i is greater than or equal to two since otherwise the theorem follows easily. By the first part of this theorem, we have

$$\kappa'(G) \leq \delta(G) \leq \Delta(G) \leq |V(G)| - 1 \leq n_1 + n_2 - 1.$$

Since $(n_1 - 1)(n_2 - 1) > 0$, we have that $\kappa'(G)$ is less than $n_1 n_2$. Hence, there are two vertices $u_1 \in V_1$ and $u_2 \in V_2$ that are not neighbors. For each edge $\{u, v\} \in S$ pick either u or v so that your choice is not equal to u_1 nor u_2. Let W denote this set of vertices. The size of W is less than or equal to $\kappa'(G) = |S|$. The removal of W eliminates all edges in S from G. Furthermore, the remaining graph consists of at least two components—one containing u_1 and the other u_2. Therefore,

$$\kappa(G) \leq \kappa'(G).$$

This completes the proof of the theorem. □

➤ Note 6.25

The bounds in Theorem 6.24 can be reached. They can take any positive integer value. If n is greater than or equal to one, we have

$$\kappa(K_n) = \kappa'(K_n) = \delta(K_n) = n - 1.$$

The following theorem shows how invulnerable a graph on n vertices and e edges can be, where n and e are given fixed numbers.

Theorem 6.26

Let G be a graph on n vertices having e greater than or equal to $n - 1$ edges. In this case we have

$$\kappa(G) \leq \kappa'(G) \leq \left\lfloor \frac{2e}{n} \right\rfloor.$$

PROOF: By the Hand Shaking Theorem (Theorem 1.24), we have

$$n\delta(G) \leq \sum_{u \in V(G)} d_G(u) = 2e.$$

So, $\delta(G) \leq 2e/n$, and the inequality follows from Theorem 6.24. □

The following example illustrates an interesting class of regular graphs having maximum possible connectivity.

☞ **Example 6.27**
Assume n and e are such that e is greater than or equal to $n-1$, k equals $\lfloor 2e/n \rfloor$, and k is an even number less than or equal to $n-1$. We can construct a k-regular graph H on n vertices by connecting each vertex u on the cycle C_n to all vertices of distance $k/2$ or less from u in C_n. This graph is sometimes called the k-regular Harary graph *on n vertices due to Frank Harary. The graph shown in Figure 6.7 is a 4-regular Harary graph on eight vertices.*

One can argue that the graph resulting from removing any $k-1$ vertices from H is connected. Therefore, $\kappa(H)$ is greater than or equal to k. In fact, $\kappa(H)$ equals k (see Exercise 9). This shows that the upper bound in Theorem 6.26 can be reached in this case. Similar constructions can be developed when k is odd (see Exercise 10).

6.6 Separability

This section focuses on graphs having connectivity one or two. They are of special interest since their structures are relatively simple to describe. We begin with an important definition.

Definition 6.28
Let G be a connected graph on three or more vertices.

1. *G is said to be* separable *if it has at least one cut-vertex. Otherwise, G is* nonseparable.

2. *A maximal nonseparable connected subgraph in G is called a* block *of G.*

It makes sense to talk about the blocks of any graph G even if G is not connected. We just consider each component at a time; a block of a component of G is then a block of G.

The next example illustrates Definition 6.28.

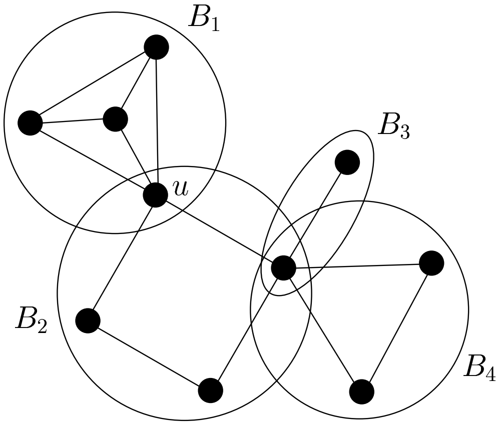

Figure 6.8: A connected graph with four blocks.

☞ **Example 6.29**

In Figure 6.1(a) the vertex u_4 is a cut-vertex, so the graph is separable. The blocks of this graph are $G[\{u_3, u_4\}]$, consisting of just an edge and its endvertices, and $G[\{u_1, u_2, u_4, u_5, u_6\}]$. (The Harary graph in Figure 6.7 is nonseparable and hence a block itself.) In Figure 6.8 the vertex u is a cut-vertex, and the graph is separable with four blocks B_1, B_2, B_3, and B_4. In a tree every vertex with degree greater than one is a cut-vertex (see Exercise 18). So trees with more than two vertices are separable.

The next lemma is needed in the proof of Theorem 6.32.

Lemma 6.30

If B_1 and B_2 are two distinct blocks in a graph G, then $B_1 \cap B_2$ is either empty or consists of a single vertex.

Proof: Assume that B_1 and B_2 have two distinct vertices u and v in common. We argue that $B_1 \cup B_2$ is nonseparable as follows.

Removing a vertex $w \notin \{u, v\}$ from one of the blocks, say B_1, will leave a subgraph $(B_1 - w) \cup B_2$ of $B_1 \cup B_2$. Since B_1 is a block it is connected, and so is B_2. If $x \in V(B_1 - w)$ and $y \in V(B_2)$, then there are paths from x to u in $B_1 - w$, and from u to y in B_2, respectively. Hence, $(B_1 - w) \cup B_2$ is connected. Likewise, removing either u or v, say u, from $B_1 \cup B_2$ will yield a subgraph $(B_1 - u) \cup (B_2 - u)$ of $B_1 \cup B_2$. Every vertex in $V(B_1 - u)$ can be connected to every vertex in $V(B_2 - u)$ via the vertex v. So, in any case, the removal of a single vertex from $B_1 \cup B_2$ yields a connected graph. Therefore, $B_1 \cup B_2$, which properly contains both B_1 and B_2, is nonseparable. This contradicts the fact that B_1 and B_2 are blocks of G. □

Definition 6.31
Let G be a connected graph. Let $C(G) \subseteq V(G)$ be the set of G's cut-vertices. Let $\mathcal{B}(G)$ be the set of blocks of G. The block-cutpoint graph *is the graph with vertex set $C(G) \cup \mathcal{B}(G)$ and edge set*

$$\{\{u, B\} : u \in C(G), B \in \mathcal{B}(G), \text{ and } u \in V(B)\}.$$

It is denoted by BC(G)*.*

We note that *BC*(G) is a bipartite graph with vertex partition $C(G) \cup \mathcal{B}(G)$. Moreover, we have the following.

Theorem 6.32
For a connected graph G, the block-cutpoint graph BC(G) *is a tree. In addition, all the leaves of* BC(G) *are contained in $\mathcal{B}(G)$.*

Proof: If G is a connected graph, then clearly *BC*(G) is also connected. Assume that we have a simple cycle $(u_1, B_1, u_2, B_2, \ldots, u_m, B_m, u_1)$ in *BC*(G). By Lemma 6.30, we have that m is greater than or equal to three. Also, there is at least one cut-vertex u_i of G, i greater than or equal to two, that is not contained in B_1. Since $G' = B_2 \cup \cdots \cup B_m$ is a connected graph, there is a path P from u_2 to u_m contained in G'. Now the subgraph $B_1 \cup P$ in G is clearly nonseparable and strictly contains both B_1 and the vertex u_i. This contradicts the fact that B_1 is a block of G.

Since each cut-vertex is contained in at least two blocks, every cut-vertex must be an internal vertex of *BC*(G). Therefore, all the leaves in *BC*(G) are contained in $\mathcal{B}(G)$. □

Remark 6.33
If G is disconnected then BC(G) *is a forest with the same number of components as G. If G has components $G_1, \ldots, G_k$, then*

$$\mathrm{BC}(G) = \mathrm{BC}(G_1) \cup \cdots \cup \mathrm{BC}(G_k).$$

By Theorem 4.4 we obtain the following.

Corollary 6.34
Every connected separable graph has at least two blocks, where each contains exactly one cut-vertex of G.

The previous results give us a way to reduce our consideration to the blocks of a given separable graph. In particular, if we can demonstrate a property for each block of given graph G, then we can use induction on the number of blocks of G using of Corollary 6.34. But in order to follow this course of action, we need some description of the blocks.

By definition, a block of a connected graph G has no cut-vertices. Therefore, it is 2-connected. Likewise, each maximal 2-connected subgraph of G is a block of G. Hence, describing the blocks of G is the same as describing 2-connected graphs in general.

Theorem 6.35
For a connected graph G on three or more vertices the following statements are equivalent:

1. *G is 2-connected.*
2. *For every pair of distinct vertices, there is a cycle in G passing through both of them.*
3. *For every pair of distinct vertices, there are two vertex disjoint paths in G connecting them.*

Proof: Clearly, we have that statement 3 is true if and only if statement 2 is true. For the other equivalences, we proceed as follows:

▷ $2 \Rightarrow 1$:

Let x, y, and u be distinct vertices of G, and γ a simple cycle through all three vertices. Then one of the paths from x to y will not contain the vertex u. This holds for arbitrary x and y, so u is not a cut-vertex. Hence, G is 2-connected.

▷ $1 \Rightarrow 2$:

Let x and y be vertices of G. Let

$$U_x = \{u \in V(G) : \text{ there is a cycle through } x \text{ and } u \text{ in } G\}.$$

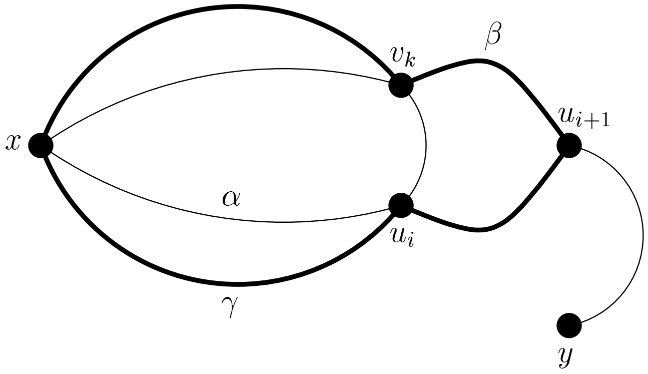

Figure 6.9: A cycle through x and u_{i+1}.

Since G is 2-connected on three or more vertices, it cannot have any cut-edges. Otherwise, one of the endvertices would be a cut-vertex. But any edge e, which is not a cut-edge, must be contained in a cycle in G. This is because there is an alternative path from one endvertex of e to the other that does not include e. Hence, all the neighbors of x are contained in U_x.

Assume now that $y \notin U_x$. We will derive a contradiction. Let $\alpha = (u_0, e_1, u_1, \ldots, e_a, u_a)$ be a path from $u_0 = x$ to $u_a = y$. Let i be the largest index such that $u_i \in U_x$. From above, we have i is greater than or equal to one and $u_i \neq x$. Since G is 2-connected, $G - u_i$ is connected. So, there is a path $\beta = (v_0, f_1, v_1, \ldots, f_b, v_b)$ from $v_0 = x$ to $v_b = u_{i+1}$ that does not contain u_i. Let γ be a cycle through x and u_i. Let k be the largest index such that v_k is on γ. At this point we construct a cycle though x and u_{i+1} as follows (see Figure 6.9).

Let p_1 be the path from x to v_k along the path of γ that does not contain u_i. If k equals zero, then p_1 consists of just one vertex. Let p_2 be the path from v_k to u_{i+1} along β. Let p_3 be the path (u_{i+1}, e_{i+1}, u_i). Finally, we let p_4 be the path from u_i to x along the path of γ that does not contain v_k. We now have a cycle

$$\gamma' = p_1 p_2 p_3 p_4$$

through both x and u_{i+1}, contradicting that $u_{i+1} \notin U_x$.

□

This theorem can be generalized to k-connected graphs as follows.

Theorem 6.36
A connected graph G is k-connected if and only if every pair of vertices in G are joined by k or more paths which, besides from their endvertices, are vertex-disjoint.

A similar description holds for k-edge-connected graphs.

Theorem 6.37
A connected graph G is k-edge-connected if and only if every pair of vertices in G is joined by k or more edge-disjoint paths.

For the proofs of Theorems 6.36 and 6.37, we refer the reader to [15, Chapter 5] or [31, Section 4.2]. You are encouraged to look up these two proofs. Although the proofs themselves are not too involved, they do require a bit of new terminology that we will not need in this book.

6.7 1-Isomorphism

As mentioned earlier, when considering properties of graphs, one can often reduce the study of a graph to that of its blocks. More precisely, a certain property can hold for a given graph if and only if it holds for each of its blocks. Hence, connected graphs that have the same set of blocks have a lot in common. We explore such graphs in this section, beginning with a simple example.

☞ **Example 6.38**
The graph G in Figure 6.10 has three cut-vertices a, b, and c and five blocks B_1, B_2, B_3, B_4, and B_5.

The graph G' in Figure 6.11 has only two cut-vertices a' and b'. It has five blocks B'_1, B'_2, B'_3, B'_4, and B'_5, where each block B'_i is isomorphic to the block B_i of G.

We see that G is connected with 11 vertices and G' is disconnected with 12 vertices. Although the graphs G and G' have the same blocks, the graphs are far from being isomorphic.

The following definition introduces a new notion of isomorphism to capture the similarities between G and G'.

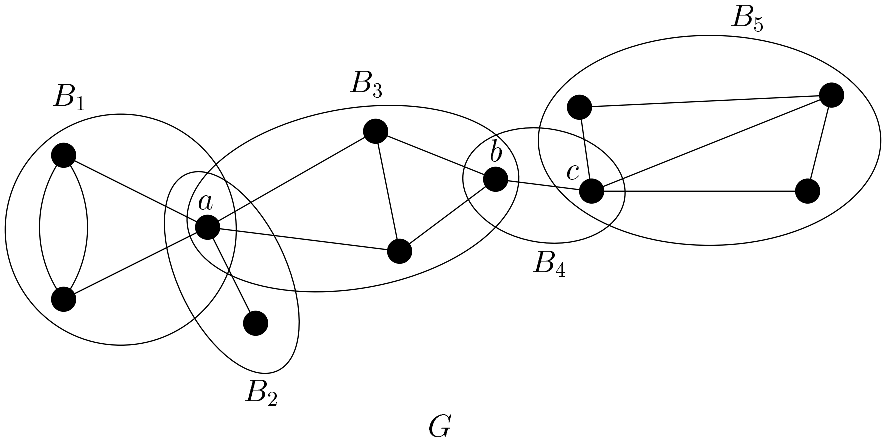

Figure 6.10: G has three cut-vertices and five blocks.

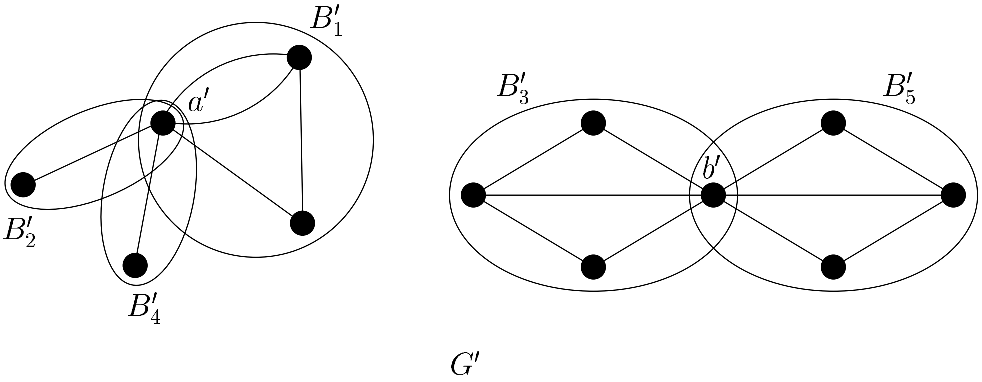

Figure 6.11: G' has the same set of blocks as G.

Definition 6.39
Let G and G' be graphs. Let $\mathcal{B}(G)$ and $\mathcal{B}(G')$ be their respective sets of blocks. If there is a bijection

$$\beta : \mathcal{B}(G) \to \mathcal{B}(G')$$

such that B and $\beta(B)$ are isomorphic graphs for each $B \in \mathcal{B}(G)$, then G and G' are said to be 1-isomorphic.

The graphs G and G' in Figures 6.10 and 6.11, respectively are 1-isomorphic since $B_i \cong B'_i$ for each $i \in \{1, 2, 3, 4, 5\}$.

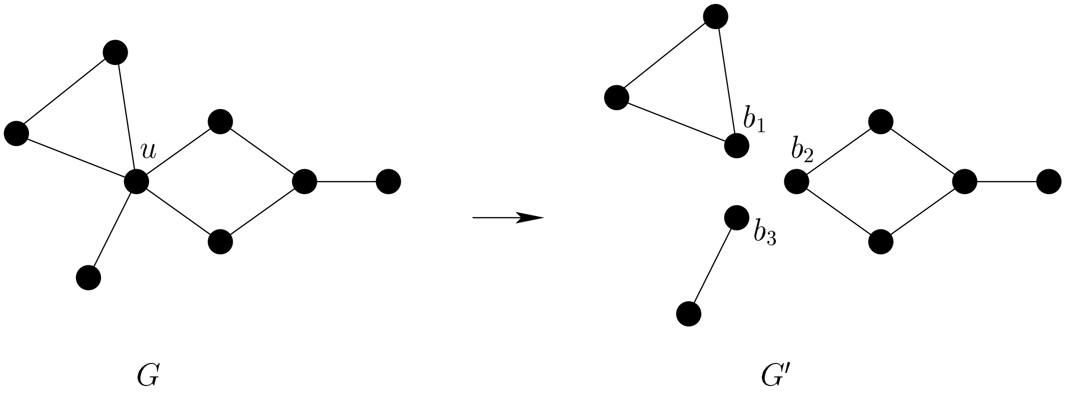

Figure 6.12: G' derived from G by a 1-split of the cut-vertex u.

Consider the following operation called a *1-split*. For a graph G, let u be a cut-vertex of G. Let $B_1, \ldots, B_m$ be the set of blocks containing u.

▷ Introduce new vertices $b_1, \ldots, b_m$.

▷ In $G - u$ connect each vertex b_i to the neighbors of u in B_i while preserving edge multiplicity. This results in a new graph G'.

The graph G' has one less cut-vertex than G. However, it is 1-isomorphic to G. The 1-split pulls the blocks connected by u apart and "splits" u into m copies such that each block B_i has a copy of u (see Figure 6.12).

By repeating the 1-split operation, we have the following observation.

Observation 6.40
Two graphs are 1-isomorphic if and only if they can become isomorphic under repeated 1-split operations.

Remark 6.41
Two isomorphic graphs are 1-isomorphic to each other. It is also apparent that two nonseparable graphs are 1-isomorphic if and only if they are isomorphic.

Observe that under the 1-split operation, the number of edges stays the same, $|E(G)| = |E(G')|$. Also, the number of components is increased by $m-1$ and the number of vertices is increased by $m-1$. So, from Observation 6.40, we get the following.

Theorem 6.42
Suppose G and G' are two 1-isomorphic graphs with c and c' components, respectively. Then we have

$$\begin{aligned} |E(G)| &= |E(G')| \textit{ and} \\ |V(G)| - c &= |V(G')| - c'. \end{aligned}$$

6.8 2-Isomorphism

In Section 6.7 we generalized the concept of isomorphism by introducing 1-isomorphism. The concept of 1-isomorphism for nonseparable graphs is the same as isomorphism. However, for separable graphs, 1-isomorphism is different from isomorphism. We can generalize this concept further to broaden its scope for 2-connected graphs as follows.

Consider the following *2-split-glue* operation. Let G be a 2-connected graph. Let u and v be a pair of vertices whose removal from G leaves the remaining graph disconnected. That is, $G - \{u, v\} = G_1 \cup G_2$ is a disjoint union. We modify $G - \{u, v\}$ as follows:

▷ Introduce four new vertices: u_1 and u_2 corresponding to u, and v_1 and v_2 corresponding to v.

▷ Preserve the edge multiplicity of G by connecting u_1 to the former neighbors of u in G_1 and by connecting u_2 to the former neighbors of u in G_2. In the same manner, connect v_1 to the former neighbors of v in G_1 and connect v_2 to the former neighbors of v in G_2.

▷ If u and v were connected in G with m edges, connect u_1 and v_1 with m_1 edges, and connect v_1 and v_2 with m_2 edges, where $m_1, m_2 \geq 0$ and $m_1 + m_2 = m$. Any choice of m_1 and m_2 is fine. At this point we have two disjoint graphs G_1' and G_2'.

▷ Connect u_1 and v_2 with an edge e_{12}, and connect u_2 and v_1 with an edge e_{21} to obtain the connected graph G'.

▷ Form the contraction $(G' \cdot e_{12}) \cdot e_{21}$ to obtain the graph G''.

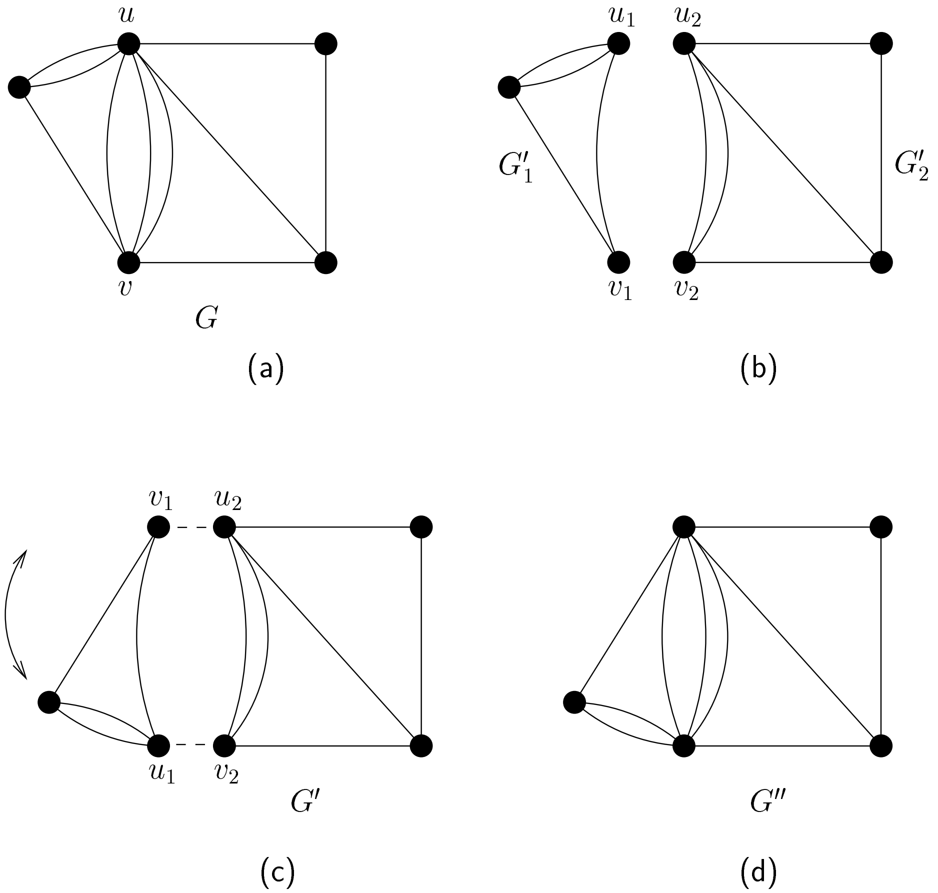

Figure 6.13: G'' obtained from G by a 2-split-glue operation.

Roughly speaking, the 2-split-glue operation "splits" the vertex u into u_1 and u_2 and the vertex v into v_1 and v_2 in such a way that G is divided into two disjoint graphs G'_1 and G'_2. The vertices u_1 and v_1 go with G'_1, and u_2 and v_2 with G'_2. Now the graphs G'_1 and G'_2 are joined by merging u_1 with v_2 and u_2 with v_1. Edges whose endvertices were u and v in G could have gone with G'_1 or G'_2, without affecting the final graph G''. In Figure 6.13 we have taken m_1 equal to one and m_2 equal to two.

Observation 6.43

If G is 2-connected and G'' is obtained from G by a 2-split-glue operation, then G'' is also 2-connected.

PROOF: A cut-vertex in G'' is also a cut-vertex in G. □

The following definition makes precise another more general form of isomorphism.

Definition 6.44
Two graphs are said to be 2-isomorphic *if and only if they can become isomorphic under repeated operations—either 1-split or 2-split-glue.*

From Definitions 6.39 and 6.44 it follows immediately that isomorphic graphs are always 1-isomorphic, and 1-isomorphic graphs are always 2-isomorphic. But 2-isomorphic graphs are not necessarily 1-isomorphic, and 1-isomorphic graphs are not necessarily isomorphic. However, for graphs with connectivity three or more, isomorphism, 1-isomorphism, and 2-isomorphism are synonymous.

Under a 2-split-glue operation, the number of vertices, edges, and components are fixed. Hence, we have the following.

Theorem 6.45
Suppose G and G' are two 2-isomorphic graphs with c and c' components, respectively. Then we have

$$\begin{aligned} |E(G)| &= |E(G')| \text{ and} \\ |V(G)| - c &= |V(G')| - c'. \end{aligned}$$

There is an alternative description of when two graphs are 2-isomorphic in terms of their *cycle correspondence*.

Definition 6.46
Let G and G' be two graphs. Let $\mathcal{C}(G)$ and $\mathcal{C}(G')$ be the sets of their cycles, respectively. The graphs are said to have a cycle correspondence, *if there are bijections*

$$\begin{aligned} \gamma : E(G) &\rightarrow E(G') \text{ and} \\ \tilde{\gamma} : \mathcal{C}(G) &\rightarrow \mathcal{C}(G'), \end{aligned}$$

such that $E(\tilde{\gamma}(C)) = \gamma(E(C))$ for all cycles $C \in \mathcal{C}(G)$. That is, the edges in $\tilde{\gamma}(C)$ are formed by the images of the edges of C under γ.

In a separable graph G every cycle is confined to a particular block (see Exercise 22). So, every cycle in G retains its edges as G undergoes a 1-split operation. Hence, 1-isomorphic graphs have a cycle correspondence. The following theorem is considered one of the most important descriptions of 2-isomorphic graphs.

Theorem 6.47 (Whitney's)
Two graphs are 2-isomorphic if and only if they have cycle correspondence.

Proof: We will prove the "only if" part. Let us consider what happens to a cycle in a graph G when it undergoes a 2-split-glue operation. We use the notation given in the description of the 2-split-glue operation. A cycle $C \in \mathcal{C}(G)$ will fall in one of three categories:

1. C is comprised of edges all in G_1',
2. C is comprised of edges all in G_2', or
3. C is comprised of edges from both G_1' and G_2'.

In cases 1 and 2, C is unaffected by the 2-split-glue operation. In case 3, the cycle C must include both vertices u and v. After the 2-split-glue operation, the resulting cycle C'' has the same set of edges as C except that the order of the edges in C between vertices u and v in exactly one of the parts, G_1' or G_2', which constituted an edge of C, has been reversed. Thus every cycle in a graph undergoing 2-split-glue operation retains its original edges. Therefore, 2-isomorphic graphs have a cycle correspondence.

The "if" part, which we omit, is more involved. The reader is referred to Whitney's original paper [32] for a proof. □

The following example makes these ideas concrete.

☞ **Example 6.48**
In Figure 6.14 we have two 2-isomorphic, 2-connected graphs G and G''. There is a bijection between $E(G)$ and $E(G'')$ given by $\gamma(e_i) = e_i''$ for $i \in \{1, 2, 3, 4, 5, 6, 7\}$. For the cycle $C = (e_1, e_2, e_3, e_4, e_5)$, we have a corresponding cycle $C'' = (e_1'', e_2'', e_3'', e_5'', e_4'')$. Notice that the order of the indices of the last two edges has been reversed.

A valid definition of $\tilde{\gamma} : \mathcal{C}(G) \to \mathcal{C}(G'')$ can be given by the following rule:

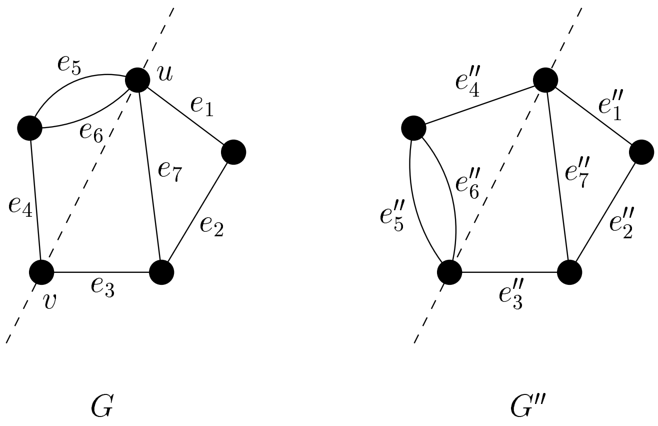

Figure 6.14: The order of one edge of the cycle C has been reversed in C''.

1. *Let $C \in \mathcal{C}(G)$ be specified by $C = (e_{i_1}, \ldots, e_{i_k})$, where k is less than or equal to five (since neither G nor G'' has any cycle of length more than five) and $\{i_1, \ldots, i_k\} \subseteq \{1, 2, 3, 4, 5, 6, 7\}$ is distinct.*

2. *Form the k-tuple $(e''_{i_1}, \ldots, e''_{i_k})$.*

3. *Reverse the order of those possible two e_{i_l}'s that have $i_l \in \{4, 5, 6\}$ and obtain the cycle $C'' = (e''_{j_1}, \ldots, e''_{j_k})$.*

4. *Define $\tilde{\gamma}(C) = C''$.*

As we shall observe later, the ideas of 2-isomorphism and cycle correspondence play important roles in the theory of networks in general and in the duality of graphs. As we have seen, 2-isomorphism is very much related to the notion of 2-connected graphs. The description of general k-connected graphs becomes more and more cumbersome. A nice description of 3-connected graphs can be found in [9, Section 3.2]. At this point we have reduced certain problems for graphs to the classification shown in Figure 6.15.

6.9 Exercises

1. List all of the edge cuts separating the vertex pair u_2 and u_3 in the graph shown in Figure 6.1(a).