

AUTOMORPHISM GROUPS OF SIMPLE GRAPHS

LUKE RODRIGUEZ

ABSTRACT

Group and graph theory both provide interesting and meaningful ways of examining relationships between elements of a given set. In this paper we investigate connections between the two. This investigation begins with automorphism groups of common graphs and an introduction of Frucht's Theorem, followed by an in-depth examination of the automorphism groups of generalized Petersen graphs and cubic Hamiltonian graphs in LCF notation. In conclusion, we examine how Frucht's Theorem applies to the specific case of cubic Hamiltonian graphs.

1. GROUP THEORY FUNDAMENTALS

In the field of Abstract Algebra, one of the fundamental concepts is that of combining particular sets with operations on their elements and studying the resulting behavior. This allows us to consider a set in a more complete way than if we were just considering their elements themselves outside of the context in which they interact. For example, we might be interested in how the set of all integers modulo m behaves under addition, or how the elements of a more abstract set S behave under some set of permutations defined on the elements of S . Both of these are examples of a *group*.

Definition 1.1. *A group is a set S together with an operation \circ such that:*

- (1) *For all $a, b \in S$, the element $a \circ b = c$ has the property that $c \in S$.*
- (2) *For all $a, b, c \in S$, the equality $a \circ (b \circ c) = (a \circ b) \circ c$ holds.*
- (3) *There exists an element $e \in S$ such that $a \circ e = e \circ a = a$ for all $a \in S$.*
- (4) *For all $a \in S$ there exists an element $b \in S$ such that $a \circ b = b \circ a = e$.*

Note that under this definition there are sets together with operations that do not form groups, for example the set of integers under division. Now that we have the notion of a group, it is natural to begin to classify these groups and organize them into groups of similar structure. One common group is S_n , the Symmetric Group consisting of all possible permutations of n elements. Another is D_n , the set of symmetries of an n -gon, also known as the Dihedral Group on n elements. See Gallian's text for full definitions of both of these groups[6].

In examining the specific example of the group D_5 , the group of symmetries of the pentagon, we notice that there are two basic kinds of elements, rotations and flips. In particular, we have 5 distinct rotations that form a group R by themselves, as well as 5 distinct flips. It is only natural then to try to write the group somehow as a product of two other groups. There is one additional difficulty here, which is that the flips do not actually form a group by themselves, as can be verified by composing any two flips together as well as the fact that there is no identity element. If we consider instead the group F of two elements formed by the identity and one flip (say the one over the axis through vertex 0), we notice that $R = \{R_0, R_{72}, R_{144}, R_{216}, R_{288}\}$ and $F = \{R_0, f\}$, and

Date: May 17, 2014.

Thanks to Professor Barry Balof for his guidance and encouragement throughout this project as well as to Professor Albert Schueller for his feedback on formatting and stylistic choices. I would not have been able to complete this project without their generously given time.

hence $|R||F| = 10 = |D_5|$. This seems to suggest some sort of equality between D_5 and these two groups considered together. However, we need a systematic way to define products of groups. The first method is called the *external direct product*, and is defined as follows:

Definition 1.2. *Let G_1, G_2, \dots, G_n be a finite collection of groups. The external direct product of these groups, written as $G_1 \times G_2 \times \dots \times G_n$ is the set of all n -tuples for which the i th component is an element of G_i , and the operation is componentwise [6].*

Using Definition 1.2, one might think that $D_5 \cong R \times F$. To begin, we take as an example the symmetry of the square obtained by first operating by R_{72} and then f . To express the group as a direct product, we would like to be able to represent this state in the form (R_{72}, f) . However, if we return to Definition 1.2, we note that the operation, in this case composition of rotations and flips, should be componentwise, meaning that it should not matter whether we rotate or flip our pentagon first. A quick check shows us that this is not true, and order does in fact matter. Therefore we need another way to formalize a product of groups. Before we get there, however, we will need to formalize the notion of group *homomorphisms*, *isomorphisms*, and *automorphisms*, which we will do with definitions taken from Steinberger [7].

Definition 1.3. *Let G and G' be groups. A homomorphism from G to G' is a function $f : G \rightarrow G'$ such that $f(x \circ y) = f(x) \circ f(y)$ for all $x, y \in G$.*

Definition 1.4. *An isomorphism is a homomorphism that is one-to-one and onto.*

Definition 1.5. *An automorphism of a group G is an isomorphism from G to itself. We denote the set of all automorphisms of G as $\text{Aut}(G)$*

Note that in Definition 1.3, the \circ operator refers to the operation defined as part of the group G when inside the argument of f , but then changes to the operation defined as part of the group G' on the other side of the equality. These operations are context dependent, they could be two entirely different operations depending on the nature of the two groups involved.

Theorem 1.6. *The set of all group automorphisms of G forms a group under function composition.*

Proof. We will show that the elements of $\text{Aut}(G)$ satisfy the four conditions laid out in Definition 1.1.

- (1) Let α, β be any two automorphisms in $\text{Aut}(G)$. Since both α and β are automorphisms, they permute the elements of G . It follows that the combination of the two will still permute the elements of G , and thus the resulting permutation is an automorphism. It follows that $\alpha \circ \beta \in \text{Aut}(G)$.
- (2) Since function composition is associative, this point follows by assumption.
- (3) The identity automorphism i is the permutation defined by $i(g) = g$ for all $g \in G$. Given any automorphism $\alpha \in \text{Aut}(G)$, we find that $(\alpha \circ i)(g) = \alpha(i(g)) = \alpha(g)$, and that $(i \circ \alpha)(g) = i(\alpha(g)) = \alpha(g)$ for all $g \in G$.
- (4) Let us consider a generic automorphism $\alpha \in \text{Aut}(G)$. The size of $\text{Aut}(G)$ is finite, since we can have at most $n!$ permutations on a set of n elements. Thus $\alpha \circ \alpha \circ \dots \circ \alpha = \alpha^j$ must eventually equal the identity permutation for some positive integer j , otherwise $\text{Aut}(G)$ would be infinite. Let $\beta = \alpha^{j-1}$. It follows that $\beta \circ \alpha = \alpha \circ \beta = \alpha^j = i$.

It follows that $\text{Aut}(G)$ is a group under function composition. □

With this notion of an automorphism group we can define the second method of defining a product of two groups. This second method, called a *semidirect product*, is a generalized form of the external direct product that allows us to have the groups whose product we are taking interact with each other. More detail on both this definition and the related concepts can be found in Steinberger [7]

Definition 1.7. Let K and H be groups, and let $\alpha : K \rightarrow \text{Aut}(H)$ be a homomorphism. The semidirect product of H and K with respect to α is the set $H \times K$ with the binary operation

$$(h_1, k_1) \circ (h_2, k_2) = (h_1 \circ \alpha(k_1)(h_2), k_1 \circ k_2)$$

and is written $H \rtimes_{\alpha} K$.

This definition seems very abstract and complex at first, but we can see how it is simply a generalized version of Definition 1.2 by proving the following Corollary:

Corollary 1.8. Given two groups H and K , the external direct product $H \times K$ is isomorphic to the semidirect product $H \rtimes_i K$, where i is the identity permutation on H .

Proof. Let i be the identity permutation on H , and define our homomorphism α by $\alpha(k) = i$ for all $k \in K$. Then by Definition 1.7, we see that taking the product any two elements of $H \rtimes_{\alpha} K$ is equivalent to the set $H \times K$ with the binary operation

$$\begin{aligned} (h_1, k_1) \circ (h_2, k_2) &= (h_1 \circ \alpha(k_1)(h_2), k_1 \circ k_2) \\ &= (h_1 \circ i(h_2), k_1 \circ k_2) \\ &= (h_1 \circ h_2, k_1 \circ k_2) \end{aligned}$$

By examining Definition 1.2, we see that this is precisely the external direct product of the two groups. \square

Now we can formalize the structure of D_5 using the semidirect product defined in Definition 1.7. To do so, we must first find some isomorphism γ of our group of rotations R other than the identity, as Lemma 1.8 tells us that this would simply be $R \times F$ which we know is not isomorphic to D_5 . In particular, we state that γ defined by $\gamma(R_m) = R_m^{-1} = R_{360-m}$ for all $R_i \in R$ is an isomorphism of R , a proof of which is trivial. We know from the behavior of the different elements of our group D_5 that whether or not we flip makes a difference for the subsequent rotations, and this leads us to believe that the homomorphism $\alpha : F \rightarrow \text{Aut}(R)$ will be defined by $\alpha(R_0) = i$ and $\alpha(f) = \gamma$. It follows that our guess for the structure of D_5 is $R \rtimes_{\alpha} F$. To see if this solves our former problem, we would like to show that two rotations with a flip in between is different than two rotations followed by a flip. Using Definition 1.7 we find that

$$(R_{72}, f) \circ (R_{72}, R_0) = (R_{72} \circ \alpha(f)(R_{72}), f \circ R_0) = (R_{72} \circ \gamma(R_{72}), f) = (R_{72} \circ R_{288}, f) = (R_0, f)$$

while

$$(R_{72}, 0) \circ (R_{72}, f) = (R_{72} \circ \alpha(R_0)(R_{72}), R_0 \circ f) = (R_{72} \circ i(R_{72}), f) = (R_{72} \circ R_{72}, f) = (R_{144}, f).$$

Hence, we find that the order does matter, and therefore that this semidirect product seems to be a good representation of the structure of D_5 . We will not present a formal proof of the isomorphism between the two in this paper, but the interested reader can read further on the subject in Gallian's algebra text [6].

2. GRAPH THEORY FUNDAMENTALS

Another natural question to ask when given a set of elements is how they are related to each other. For example, given a set of cities, we might want to know which of them are directly connected by roads. Similarly, we might want to know who is friends with whom in a given group of people. Both of these sets lend themselves naturally to representation in the form of a *graph*.

Definition 2.1. A graph is a collection of vertices connected by edges. Formally, we represent a graph G as a set of vertices $V_G = \{u_0, u_1, \dots, u_{n-1}\}$ together with a set of edges of the form $E_G = \{\{u_i, u_j\}, \dots, \{u_k, u_l\}\}$ where i, j, k, l are integers in the range $[0, n - 1]$.

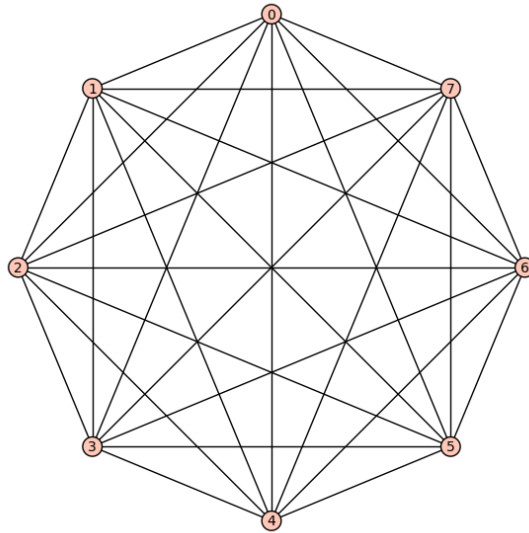


FIGURE 1. The complete graph on 8 vertices K_8

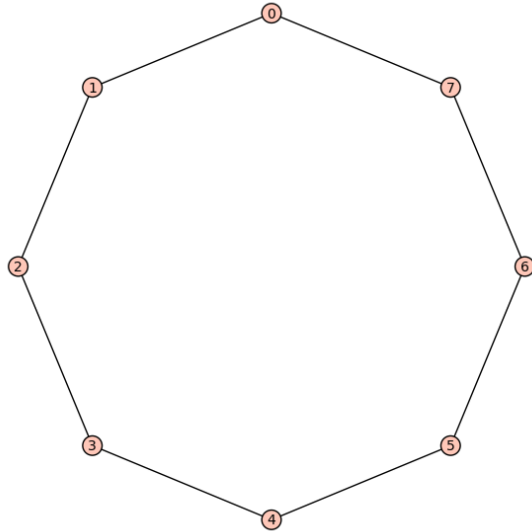
In our first example the set V_G would be the set of cities, and $\{u_i, u_j\} \in E_G$ if the cities represented by u_i and u_j were connected by a road. In the second example, V_G would be the set of people and $\{u_i, u_j\} \in E_G$ if the people represented by u_i and u_j were friends (this is, of course, assuming that friendship is a symmetric relation).

While these examples are very similar, there are some important potential differences. Suppose that we are building a graph to represent cities and the roads between them because we want to know the shortest route to drive between any two cities. Then in this case it would be important to know how long each of the roads are. We could then assign a *weight* to any given edge of our graph. Alternately, suppose we want to know how many ways there are to get from one city to another. Then it would be important to know if there were more than one direct road between two cities, or even if there was a scenic loop that went from one city back to itself. To faithfully represent this in our graph, we would need to allow there to be multiple edges between vertices in addition to edges between a vertex and itself. In contrast, we consider the case of group of people and their friendships. It clearly does not make sense to assign a quantifier to the friendship between any two people, nor does it make sense to have two people be friends twice over or someone be friends with themselves. Therefore in this case we would like to only allow our graph to have edges between two distinct vertices, and no more than one edge per vertex pair. Such a graph is said to be *simple*. In this paper we will only be considering simple graphs. For more information on other kinds of graphs, see Bogart [2] or Bona [3].

One useful concept that helps us discuss graphs is as follows: the *degree* of a vertex is equal to the number of edges incident at that vertex. Therefore, to find the degree of a given vertex $v \in V_G$, we must simply count the number of edges $e \in E_G$ that have v as one of their two entries. Knowing the degrees of vertices present in the graph will help to compare graphs to each other.

Just as with groups, there are a few graphs that are common enough to have their own standard notation. The three that we will be using in this paper are;

- (1) The Complete Graph on n vertices K_n in which every vertex is connected to all $n - 1$ other vertices (Figure 1).
- (2) The Path Graph on n vertices P_n consists of n vertices $v_0, v_1, v_2, \dots, v_{n-1}$ and $n - 2$ edges, where $\{v_i, v_{i+1}\} = e_i \in E_{P_n}$ for all $0 \leq i \leq n - 2$ (Figure 2).

FIGURE 2. The path graph on 8 vertices P_8 FIGURE 3. The cycle graph on 8 vertices C_8

- (3) The Cycle Graph C_n consists of the path graph P_n , but with the extra edge $\{v_{n-1}, v_0\}$ (Figure 3).

We still need more terminology to be able to investigate graphs more fully. First, given a graph G and an edge $e \in G$, the graph obtained by removing e from G is represented by $G \setminus e$. We also will discuss the difference between adjacent and non-adjacent edges, where two edges in a graph G are said to be *adjacent* if they share a vertex. Additionally, we will often discuss the idea of a *subgraph* of a given graph G . Any subset of V_G together with edges from E_G that use only vertices from this subset form a subgraph. Thus, $G \setminus e$ is a subgraph of G , as it uses the trivial subset of V_G that is the whole set and all but one edge from E_G . An *induced subgraph* of G is a subgraph formed by removing any number of vertices from V_G as well as all edges that include those vertices. Additionally, the *complement* of a graph G is denoted \bar{G} and represents the graph formed by taking the vertex set V_G along with all edges not in E_G . Thus, the complement of the complete graph is the empty graph, and vice-versa. Another common concept in Graph Theory is that of a *Hamiltonian* graph. A Hamiltonian graph contains a cycle such that every vertex is visited exactly once, with the exception of the starting and ending vertex which must be the same in order for it to truly be a cycle. Such a cycle is called a Hamiltonian Cycle. For example, the cycle graph C_n can easily be seen to be Hamiltonian, as the edges form a Hamiltonian cycle. However, the path graph P_n is not Hamiltonian, as there are no Hamiltonian cycles. It is worth noting that any Hamiltonian graph can be naturally represented by putting its vertices in the form of a cycle, and then adding any other edges that might be present through the interior of the figure.

It is only natural to want use some of our tools from our study of Abstract Algebra to understand these graphs better. We start by refining the idea of a group automorphism defined in Definition

1.5 to apply to graphs instead. In the case of a group automorphism, we needed a permutation that was one-to-one and onto that had the additional property of preserving the structure of the group under the operation \circ associated with it. In the case of a graph, we have no operation to preserve, but we would like to maintain the information provided by the graph through the isomorphism, therefore preserving the relationship between two vertices that are connected by an edge. This leads us to the following definition:

Definition 2.2. *A graph automorphism of G is a permutation ϕ on the set of vertices V_G that satisfies the property that $\{u_i, u_j\} \in E_G$ if and only if $\{\phi(u_i), \phi(u_j)\} \in E_G$.*

Now that we have a definition of a graph automorphism that parallels Definition 1.5, it not a stretch to wonder if the set of all graph automorphisms on a particular graph G forms a group, as was found with group automorphisms in Theorem 1.6. This is in fact the case.

Theorem 2.3. *The set $\text{Aut}(G)$ of all graph automorphisms of a graph G forms a group under function composition.*

The proof of this fact follows that of Theorem 1.6 almost precisely. We also note that a graph and its complement are very similar in structure. This leads us to believe that there might be some sort of relationship between their automorphism groups. This turns out to be the case, as outlined in the next theorem.

Theorem 2.4. *Given any graph G , $\text{Aut}(G) = \text{Aut}(\bar{G})$.*

Proof. We will proceed by showing set inclusion in both directions. First suppose that we have some permutation $\sigma \in \text{Aut}(G)$, and an edge $e \notin E_G$. By the definition of the complement of a graph, it follows that $e \in E_{\bar{G}}$. Similarly, we know from the definition of a graph automorphism that $\sigma(e) \notin E_G$, and hence we find that $\sigma(e) \in E_{\bar{G}}$. Thus we have shown that σ is an automorphism of \bar{G} , and thus $\sigma \in \text{Aut}(\bar{G})$. Note that G is isomorphic to the complement of \bar{G} , so we can simply interchange G and \bar{G} and find that if $\tau \in \text{Aut}(\bar{G})$, then $\tau \in \text{Aut}(G)$. Thus we have shown that the two automorphism groups are equal. \square

3. EDGE AUTOMORPHISM GROUPS AND LINE GRAPHS

After establishing the concept of a graph, we quickly turned to the idea of graph automorphisms that permuted the vertices of a particular graph G . However, that this is not the only way we could approach automorphisms of a particular graph. We could just as easily have decided that we would permute the edges instead of the vertices, and define our automorphism in this way. This leads us to a second kind of graph automorphism, the *edge automorphism*. This concept, along with the others discussed in this section, is explored in more detail in chapters 9 and 10 of Behzad, Chartrand, and Lesniak-Foster's book [1].

Definition 3.1. *An edge automorphism is a permutation ϕ on the set of edges E_G that satisfies the property that e_1, e_2 are adjacent if and only if $\phi(e_1), \phi(e_2)$ are also adjacent.*

These automorphisms also form a group under function composition, which will be denoted $\text{Aut}_E(G)$.

However, we notice that this concept of an edge automorphism is closely related to that of a graph automorphism given in Definition 2.2. In fact, any graph automorphism will induce some particular edge automorphism. We call this an *induced edge automorphism*, and the set of all such automorphisms will be represented by $\text{Aut}_I(G)$. It is clear that $\text{Aut}_I(G) \subseteq \text{Aut}_E(G)$, but it is natural to wonder if there is in fact equality in all cases. Figure 5 gives three examples of graphs for which this is not the case. Each of these graphs has an edge automorphism that is not induced by any graph automorphism. The automorphisms are

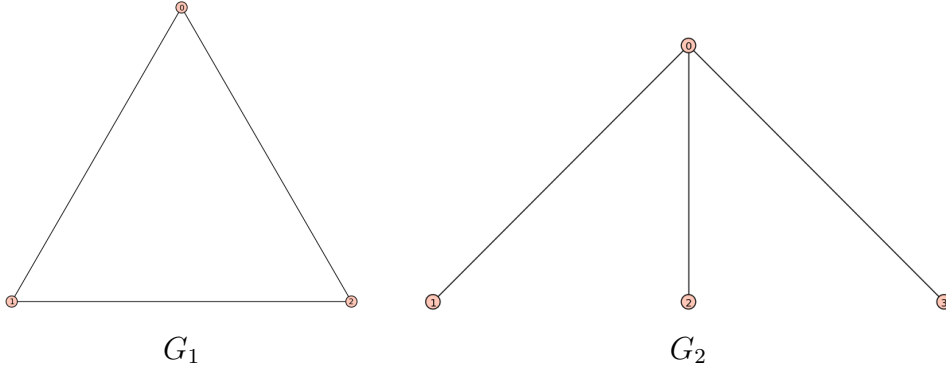


FIGURE 4. Two graphs G_1 and G_2 that have the same edge automorphism group, but are non-isomorphic [1].

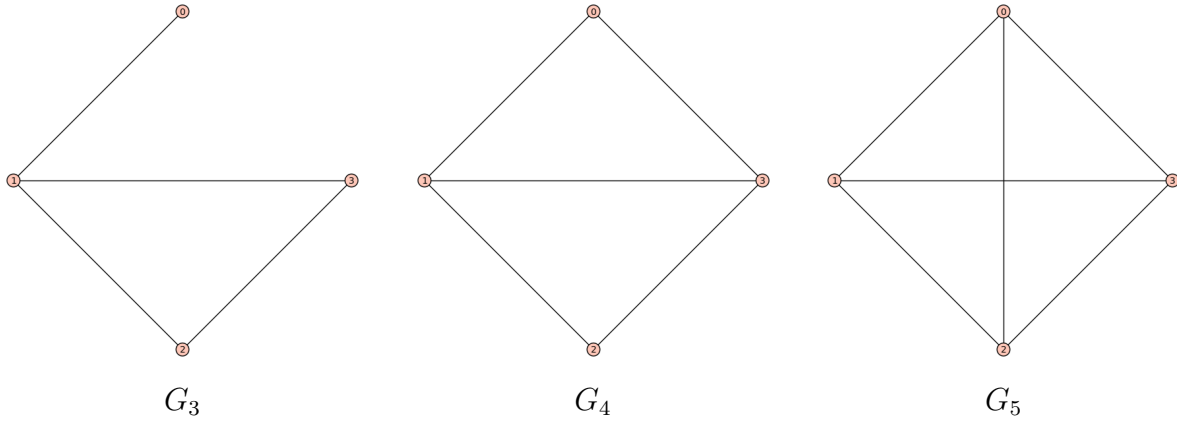


FIGURE 5. Three graphs G_1 , G_2 and G_3 that have edge-automorphisms not induced by any automorphism [1].

$$\phi_3 = \begin{pmatrix} \{v_0, v_1\} & \{v_1, v_2\} & \{v_1, v_3\} & \{v_2, v_3\} \\ \{v_2, v_3\} & \{v_1, v_2\} & \{v_1, v_3\} & \{v_0, v_1\} \end{pmatrix}$$

$$\phi_4 = \begin{pmatrix} \{v_0, v_1\} & \{v_1, v_2\} & \{v_1, v_3\} & \{v_2, v_3\} & \{v_0, v_3\} \\ \{v_1, v_2\} & \{v_2, v_3\} & \{v_1, v_3\} & \{v_0, v_3\} & \{v_0, v_1\} \end{pmatrix}$$

$$\phi_5 = \begin{pmatrix} \{v_0, v_1\} & \{v_1, v_2\} & \{v_1, v_3\} & \{v_2, v_3\} & \{v_0, v_3\} & \{v_0, v_2\} \\ \{v_2, v_3\} & \{v_1, v_2\} & \{v_1, v_3\} & \{v_0, v_1\} & \{v_0, v_3\} & \{v_0, v_2\} \end{pmatrix}$$

We can also approach these groups by examining the automorphism group of a graph $\text{Aut}(G)$ alongside the induced edge automorphism group $\text{Aut}_I(G)$.

Theorem 3.2. *Let G be a non-trivial connected graph. Then $\text{Aut}(G) \cong \text{Aut}_I(G)$ if and only if $G \not\cong K_2$.*

Proof. We begin by proving the converse of the first implication, namely that $\text{Aut}(K_2) \not\cong \text{Aut}_I(K_2)$. This follows from the fact that $\text{Aut}(K_2) \cong S_2$, while $\text{Aut}_I(K_2) \cong S_1$.

Next we assume that G is a connected graph on at least 3 vertices, noting that G must then have at least two edges. We define a mapping $\phi : \text{Aut}(G) \rightarrow \text{Aut}_I(G)$ such that $\phi(\alpha) = \alpha_I$, where α_I is the edge automorphism induced by α . We wish to show that ϕ is onto, one-to-one, and operation preserving.

- Our mapping ϕ must be onto, as this is precisely how we constructed $\text{Aut}_I(G)$ in Definition 3.1.
- Let $\alpha, \beta \in \text{Aut}(G)$, where $\alpha \neq \beta$. Since α and β are not equal, there must be a vertex $v \in V_G$ such that $\alpha(v) \neq \beta(v)$, and let u be a vertex adjacent to v . If $\alpha(u) \neq \beta(v)$ or $\alpha(v) \neq \beta(u)$, then we have found that, for the edge $e = \{u, v\}$, the induced automorphisms α_I and β_I cannot be equal, as $\alpha_I(e) \neq \beta_I(e)$. Instead we assume that $\alpha(u) = \beta(v)$ and $\alpha(v) = \beta(u)$. Since G has at least three vertices, then there exists another vertex w such that $w \notin \{u, v\}$ and w is adjacent to either u or v . We suppose without loss of generality that $e = \{u, w\} \in E(G)$, and note that $\alpha_I(e) \neq \beta_I(e)$. Thus we have shown that ϕ is one-to-one in all cases.
- Let $e = \{u, v\}$, and define

$$\beta(u) = u', \quad \beta(v) = v', \quad \alpha(u') = u'', \quad \alpha(v') = v''.$$

It follows that

$$\phi(\alpha\beta)(e) = \phi(\alpha\beta)(\{u, v\}) = \alpha_I\beta_I(\{u, v\}) = \{(\alpha\beta)(u), (\alpha\beta)(v)\} = \{\alpha(u'), \alpha(v')\} = \{u'', v''\}$$

and also that

$$\phi(\alpha)\phi(\beta)(e) = \phi(\alpha)\phi(\beta)(\{u, v\}) = \phi(\alpha)(\{\beta(u), \beta(v)\}) = \phi(\alpha)(\{u', v'\}) = \{\alpha(u'), \alpha(v')\} = \{u'', v''\}.$$

Hence we have shown that ϕ is operation preserving.

It follows from the fact that ϕ is onto, one-to-one, and operation preserving that ϕ is in fact an isomorphism, and thus that $\text{Aut}(G) \cong \text{Aut}_I(G)$. \square

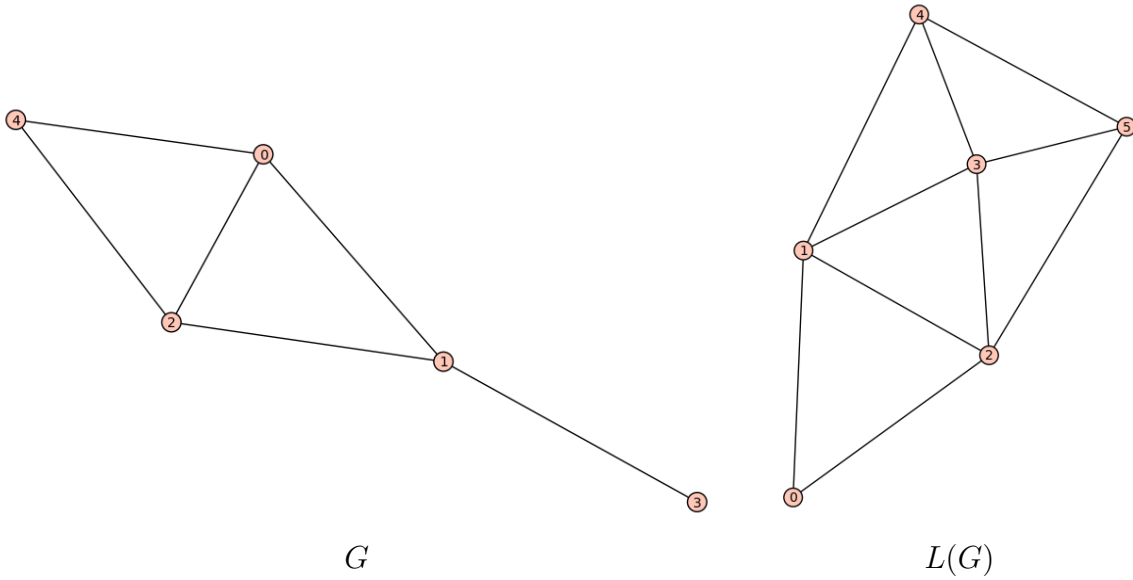
Now that we have established the relationship between the automorphism group and induced edge automorphism group of a graph, we would like to investigate how the edge automorphism group fits in. This turns out to be a more complicated process, and the details can be found in Chapter 9 of *Graphs and Digraphs* [1]. We have noted that the graphs G_3, G_4 , and G_5 in Figure 5 are such that they have edge automorphisms that are not included in the group of induced edge automorphisms. It turns out that these are the only connected graphs on three or more vertices that have this property. Thus we conclude that if G is a connected graph on 3 or more vertices not isomorphic to G_3, G_4 , or G_5 , then $\text{Aut}(G) = \text{Aut}_I(G) = \text{Aut}_E(G)$.

Given a graph G , we can create a new graph $L(G)$ by mapping each edge $e \in E_G$ to a vertex $v \in V_{L(G)}$, where any two vertices in $L(G)$ are connected if the corresponding edges are adjacent in G . This graph $L(G)$ is called the *line graph* of G , see Figure 6 for an example.

Corollary 3.3. *Given a connected graph G on 3 or more vertices, the groups $\text{Aut}(G)$ and $\text{Aut}(L(G))$ are isomorphic if G is not isomorphic to G_3, G_4 , or G_5 (Figure 5).*

Proof. By the properties of the line graph, there exists an isomorphism between the sets E_G and $V_{L(G)}$, and therefore there must also be an isomorphism between $\text{Aut}_E(G)$ and $\text{Aut}(L(G))$. By previous results, we know that $\text{Aut}(G) \cong \text{Aut}_E(G) \cong \text{Aut}(L(G))$, showing that the two automorphism groups must be isomorphic. \square

This provides us with a way of generating many non-isomorphic graphs that all have the same automorphism group, which suggests that there are a very large number of graphs of this sort. We also note that the equality between $\text{Aut}(G)$ and $\text{Aut}_E(G)$ in most gives us flexibility in terms of how we discuss the automorphism group of a graph. It allows us to define automorphisms in terms of edges and vertices interchangeably, which will become very useful in the following sections as we explore different examples of automorphism groups.

FIGURE 6. A graph G and its associated line graph $L(G)$.

4. EXAMPLES OF AUTOMORPHISM GROUPS OF CONNECTED GRAPHS

In this section we will determine the automorphism groups of certain classes of graphs in order to demonstrate that there is a relationship between the structure of a graph and its automorphism group, but we will also show that this correspondence is not unique. We begin with the classes of graphs first presented in Section 2.

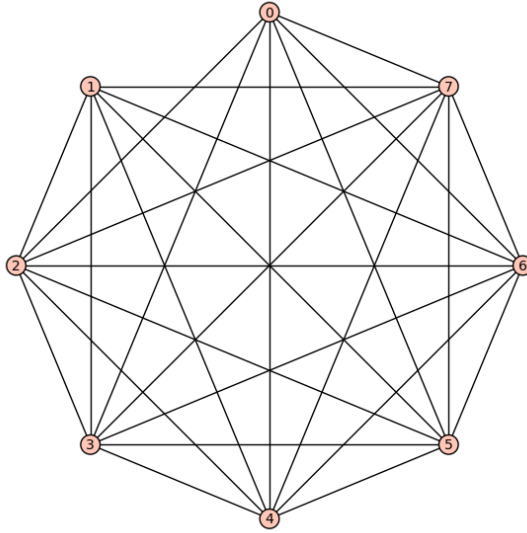
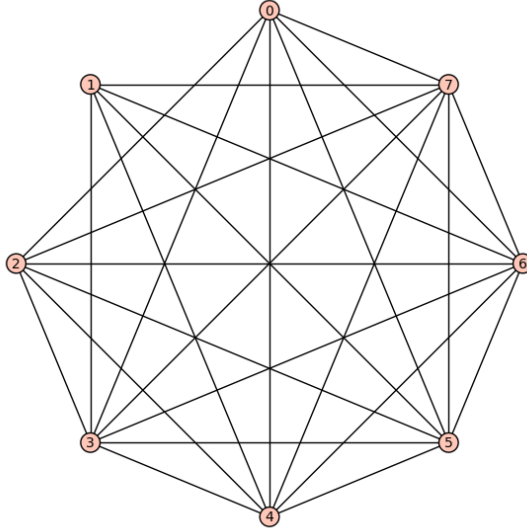
Theorem 4.1. *The automorphism group of the complete graph on n vertices $\text{Aut}(K_n)$ is isomorphic to S_n .*

Proof. The complete graph on n vertices K_n consists of n vertices connected to each of the other $n - 1$ vertices by edges. Thus an automorphism of K_n can send each vertex to any of the others, and furthermore this does not place any restriction on where any of the other $n - 1$ vertices are mapped, as they are all mutually connected. Therefore the automorphism group must have size $n(n - 1)(n - 2) \dots (2)(1) = n!$, and in particular is isomorphic to S_n . \square

Although this is a fairly simple case, we can imagine complicating it by removing edges from the graph, as shown in Figures 7, 8 and 9. This would allow us to determine the automorphism group of any graph simply by the number of edges by which it differs from the complete graph on the same number of vertices. The following theorems deal with graphs generated in such a way.

Theorem 4.2. *The automorphism group of the complete graph on n vertices with any single edge removed is isomorphic to $S_2 \times S_{n-2}$.*

Proof. Let $G = K_n \setminus e$, where e is any edge of K_n . It follows that G consists of a pair of vertices u and v which both have degree $n - 2$ along with $n - 2$ vertices all of degree $n - 1$. Any automorphism of the graph must permute each of these two sets independently of the other, so the automorphism group in general must be the direct product of two permutation groups. It is clear that the only options for the set of two vertices is to either fix or swap the two, so this portion of the direct product is isomorphic to S_2 . On the other hand, the other $n - 2$ vertices all are connected to each other, so this portion of the direct product must be isomorphic to the automorphism group of K_{n-2} , which we know by Theorem 4.1 to be S_{n-2} . It follows that the automorphism group of G is isomorphic to $S_2 \times S_{n-2}$. \square

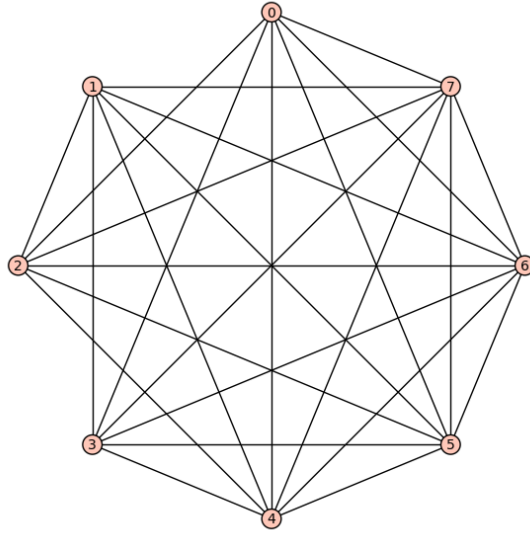
FIGURE 7. K_8 with one edge removedFIGURE 8. K_8 with two adjacent edges removed

Theorem 4.3. *The automorphism group of the complete graph on $n \geq 4$ vertices with two adjacent edges removed is isomorphic to $S_1 \times S_2 \times S_{n-3}$.*

Proof. Let u_0, u_1, u_2 be three vertices in K_n with $n \geq 3$, and define $e_0 = \{u_0, u_1\}$ and $e_1 = \{u_0, u_2\}$. Then $G = K_n \setminus \{e_0, e_1\}$. We note then that u_0 has degree $n - 3$, both u_1 and u_2 have degree $n - 2$, and all other $n - 3$ vertices have degree $n - 1$. By the same reasoning as above, the isomorphism graph of G must be isomorphic to $S_2 \times S_{n-3}$. \square

Theorem 4.4. *The automorphism group of the complete graph on $n \geq 4$ vertices with two non-adjacent edges removed is isomorphic to $D_4 \times S_{n-4}$.*

Proof. Let u_0, u_1, u_2, u_3 be four vertices in K_n with $n \geq 4$, and define $e_0 = \{u_0, u_1\}$ and $e_1 = \{u_2, u_3\}$. Then $G = K_n \setminus \{e_0, e_1\}$. As before, we will formulate the automorphism group of this graph by

FIGURE 9. K_8 with two non-adjacent edges removed

examining the 4 vertices of degree $n - 2$, and computing the direct product of this group with the automorphism group of the other $n - 4$ vertices. We note that the vertices u_0, u_1, u_2, u_3 all have the same degree, but there are 2 edges missing of the 6 possible, e_0 and e_1 . Let H be the induced subgraph of G on these four vertices. These edges are mutually non-adjacent, so E can be represented as the four edges and vertices of a square. It follows then by definition that $\text{Aut}(E) = D_4$, as D_4 is precisely the group of symmetries of the square. Hence the automorphism group of G is isomorphic to $D_4 \times S_{n-4}$. \square

This process could be repeated infinitely in order to classify the groups of all graphs, as then all one would have to do would be to classify exactly how many edges would have to be added to a particular graph before it was isomorphic to the complete graph. However, this would be very impractical, as there are many non-isomorphic ways to remove m edges from the complete graph for small m . We saw that there are two such ways to remove $m = 2$ edges, but even for $m = 3$ we note that we could remove three non-adjacent edges, two adjacent edges and one non-adjacent, three adjacent edges that are all mutually adjacent, or three adjacent edges such that two are mutually non-adjacent. Thus the number of cases that we would have to consider quickly becomes prohibitive. Next we will examine the Cycle and Path graphs, and see how we can approach the problem of finding their automorphism groups without having to consider them as a reduction of the Complete Graph.

Theorem 4.5. *The automorphism group of the Cycle Graph on n vertices $\text{Aut}(C_n)$ is isomorphic to D_n .*

Proof. We note that the most natural representation for C_n is exactly the same as that of a regular n -gon. Thus any automorphism of C_n must also be one of the set of symmetries of the n -gon, which is defined to be D_n . Conversely, no permutation of C_n can be an automorphism without being a symmetry of the n -gon, as we know that a graph automorphism must preserve the underlying structure of a graph, so any such automorphism would necessarily be an element of D_n . It follows that $\text{Aut}(C_n) \cong D_n$. \square

Theorem 4.6. *The automorphism group $\text{Aut}(P_n) \cong S_2$ for all $n \geq 2$.*

Proof. We will proceed by induction. Since there are only two vertices in P_n connected by a single edge, we find that we can either exchange them or leave them fixed, and that in either case we

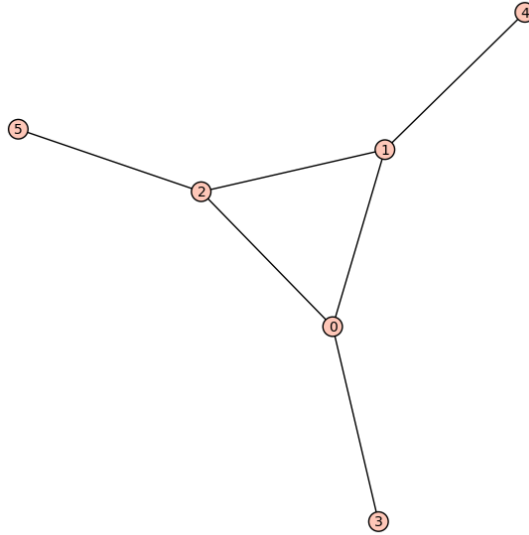


FIGURE 10. K_3 with three pendant vertices attached

preserve the edge between them. Therefore $\text{Aut}(P_2) \cong S_2$. Next suppose that $\text{Aut}(P_{k-1}) \cong S_2$ for $k \geq 3$. We note that P_k is a connected graph on three or more vertices, and therefore $\text{Aut}(P_k) \cong \text{Aut}(L(P_k))$ by Corollary 3.3. We then note that P_k consists of $k-1$ edges that can be thought of as two “ends” only adjacent to one other edge and $k-3$ edges adjacent to two other edges. Therefore $L(P_k) \cong P_{k-1}$, and in particular $\text{Aut}(P_k) \cong \text{Aut}(P_{k-1}) \cong S_2$.

It follows that $\text{Aut}(P_n) \cong S_2$ for all $n \geq 2$. \square

One interesting implication of this result is that there are an infinite number of connected graphs with automorphism groups isomorphic to S_2 . It is natural then to wonder what groups have this property, and we turn to the following theorem to help investigate this fact.

Theorem 4.7. *Given a graph G on n vertices, there exists a graph H on $2n$ vertices such that $\text{Aut}(H) \cong \text{Aut}(G)$ for $n \geq 3$.*

Proof. We will begin by constructing the graph H and then proceed to prove the isomorphism between the two automorphism groups.

Suppose G is a connected graph with vertex set $V_G = \{v_1, v_2, \dots, v_n\}$ for some $n \geq 3$. We define $V_H = V_G \cup \{u_1, u_2, \dots, u_n\}$ and $E_H = E_G \cup \{\{v_1, u_1\}, \{v_2, u_2\}, \dots, \{v_n, u_n\}\}$, and note that H is a graph on $2n$ vertices. We also note that each vertex u_i is connected only to one edge, as there are no edges that include u_i in E_G and only one such edge in $\{\{v_1, u_1\}, \{v_2, u_2\}, \dots, \{v_n, u_n\}\}$. Similarly, we note that every vertex v_i is connected to at least two edges, as there must be an edge including v_i in E_G by the fact that G is connected, and v_i appears exactly once in $\{\{v_1, u_1\}, \{v_2, u_2\}, \dots, \{v_n, u_n\}\}$ (an example of this construction is given in Figure 10 for $G = K_3$). Thus we find that any automorphism of the graph H must fix the vertices $\{v_1, v_2, \dots, v_n\}$ and $\{u_1, u_2, \dots, u_n\}$ setwise, as no vertex of degree one can be mapped to a vertex of degree two or greater and vice-versa. Define a mapping $\tau : \text{Aut}(H) \rightarrow \text{Aut}(G)$ as follows: given any $\phi \in \text{Aut}(H)$, $\tau(\phi) = \rho$ where $\phi(v_i) = \rho(v_i) = v_j$ for all vertices $v_i \in V_G$. We wish to show that our mapping τ is a bijection.

- (1) Let ϕ_1 and ϕ_2 be two automorphisms of H such that $\phi_1 = \phi_2$. It follows that $\phi_1(v_i) = \phi_2(v_i) = v_j$, so $\tau(\phi_1)(v_i) = \rho_1(v_i) = v_j$ and $\tau(\phi_2)(v_i) = \rho_2(v_i) = v_j$ and thus $\rho_1 = \rho_2$. Thus we find that τ is well-defined.
- (2) Let ρ_1 and ρ_2 be two automorphisms of G such that $\rho_1 = \rho_2$. We know that since G is a connected graph on at least three vertices, H is a connected graph on at least six

vertices and thus is not isomorphic to any of our graphs with edge automorphisms not induced by vertex automorphisms in Figure 5. Thus we can consider the behavior of our two automorphisms and their pre-images ϕ_1 and ϕ_2 defined such that $\tau(\phi_1) = \rho_1$ and $\tau(\phi_2) = \rho_2$ in terms of their behavior on the edges of G and H . In particular, we find that $\phi_1(v_i) = \phi_2(v_i)$ for all vertices $v_i \in V_G$ by the definition of our mapping τ . However, we have also noted that we must preserve edges under this automorphism, so in particular $\phi_1(\{v_i, u_i\}) = \{\phi_1(v_i), \phi_1(u_i)\} = \{v_j, \phi_1(u_j)\}$. We also have noted that any automorphism of H must fix the vertices $\{u_1, u_2, \dots, u_n\}$ setwise, so in particular $\phi_1(u_i) = u_k$ for some value $1 \leq k \leq n$. The only edges involving vertices u_i are of the form $\{v_i, u_i\}$, so it follows that $\phi_1(u_i) = u_j$. We can follow an identical argument to conclude that $\phi_2(u_i) = u_j$, and thus we have found that $\phi_1 = \phi_2$. It follows that τ is one-to-one.

- (3) Let $\rho \in \text{Aut}(G)$. We define an automorphism ϕ by $\phi(v_i) = \rho(v_i) = v_j$ and $\phi(u_i) = u_j$. We note that this automorphism fixes our two components set-wise and also preserves all of our edges of H . Thus $\phi \in \text{Aut}(H)$, and it follows that $\tau(\phi) = \rho$. Hence τ is onto.
- (4) Let ϕ_1 and ϕ_2 be two automorphisms of H . Since $\text{Aut}(H)$ is a group under function composition, it follows that $\phi_2\phi_1(w) = \phi_3(w)$ for all $w \in V_H$, where $\phi_3 \in G$. It follows that if $\phi_1(v_i) = v_j$ and $\phi_2(v_j) = v_k$, then $\phi_3(v_i) = v_k$. By the definition of our mapping τ we find that on the one hand $\tau(\phi_2\phi_1)(v_i) = \tau(\phi_3)(v_i) = \rho_3(v_i) = v_k$, while on the other $\tau(\phi_2\phi_1)(v_i) = \tau(\phi_2)\tau(\phi_1)(v_i) = \rho_2\rho_1(v_i) = \rho_2(v_j) = v_k$. Thus we have found that our mapping τ is operation preserving.

It follows that τ is a bijection, and thus that $\text{Aut}(H) \cong \text{Aut}(G)$.

If G is not a connected graph, then by definition the graph \bar{G} connected and $\text{Aut}(\bar{G}) \cong \text{Aut}(G)$. We can then use the same process defined above to construct a graph H using \bar{G} , and find that $\text{Aut}(H) \cong \text{Aut}(\bar{G}) \cong \text{Aut}(G)$. Thus we have found such a graph independent of the connectivity of G . \square

Corollary 4.8. *Given a group $H \cong \text{Aut}(G)$ for some graph G , there exist an infinite number of non-isomorphic graphs whose automorphism groups are isomorphic to H .*

Proof. For the group S_2 , the result follows directly from Theorem 4.6. For any group of size three or greater, it must be the automorphism group of some graph G on at least three vertices by a result known as Frucht's Theorem. Thus the result follows directly from Theorem 4.7. \square

Frucht's Theorem is a deep result in Algebraic Graph Theory first proven by Robert Frucht in 1939. His proof uses directed graphs that are then converted into simple undirected graphs via the use of what he called "arrows" constructed from undirected vertices and edges (see [4] for his original paper in German, which includes an illustration of the concept of arrows). A stronger version of the theorem is that there are an infinite number of graphs with a given automorphism group, which is precisely the statement of Corollary 4.8. However, the method of proof outlined here using Theorem 4.7 is different than the standard method presented by Frucht, in which he simply made the argument that there are an infinite number of non-isomorphic arrows, and thus an infinite number of non-isomorphic graphs that could be generated from a given directed graph by his method. Theorem 4.7 provides us with more insight into the generality of this result, however, as it expands the class of non-isomorphic graphs generated to have a certain automorphism group beyond the basic underlying structure of the directed graph from which they came. It then also follows that we could use Corollary 3.3 in most cases to find even more such graphs by taking the line graph, which alters their underlying structure even further from the original digraph. These tools then allow us to imagine that we can construct many qualitatively different graphs with a given automorphism group.

5. GROUPS OF GENERALIZED PETERSEN GRAPHS

Generalized Petersen graphs are a form of graph that show up quite often in applications of Graph Theory and often serve as counter-examples to what seem to be theorems. These are compelling reasons to wish to classify the automorphism groups of these graphs. The canonical example of this kind of graph shown in Figure 11 is often referred to simply as the Petersen graph, and consists of a five-cycle connected to five interior vertices, which are connected to every other interior vertex. This is written as $P(5, 2)$, where the 5 represents the number of vertices in each cycle, and the 2 represents the number of vertices to travel before connecting the interior vertices. This translates easily into a general form of the Petersen graph written $P(n, k)$. In general, we restrict the value of k by $2 \leq 2k \leq n$, as $P(n, n - m) \cong P(n, m)$ for all integers m . In this section we will show results about this particular kind of graph by following closely the methods used in Frucht, Graver, and Watkins' paper [5].

The first observation we can make about a Petersen graph in general is that if we subdivide the vertex set $V_{P(n,k)}$ into a set $U = \{u_0, u_1, \dots, u_{n-1}\}$ of vertices in the outer cycle and a set $V = \{v_0, v_1, \dots, v_{n-1}\}$ of inner vertices, then we can divide the edges into three categories. First we have the outer edges of the form $e_O = \{u_i, u_{i+1}\}$, inner edges of the form $e_I = \{v_i, v_{i+k}\}$, and spokes of the form $e_S = \{u_i, v_i\}$. Note that all addition performed in the subscripts of our vertices will be modulo n . These three types of edges partition the edge-set of the graph $E_{P(n,k)}$.

With this notation in mind, we can imagine two different kinds of permutations that are automorphisms of this graph fairly easily. The first is a rotation by $\frac{360^\circ}{n}$, and the second is a flip. This is analogous to the case of the dihedral group discussed in Section 1. We can define each of these in terms of how they act on our inner and outer vertices.

First we examine the rotation permutation ρ . We note that it acts exactly the same on inner and outer vertices, so $\rho(u_i) = u_{i+1}$ and $\rho(v_i) = v_{i+1}$. We can quickly check to make sure that this permutation preserves all three of our types of edges as follows:

$$\begin{aligned}\rho(e_O) &= \rho(\{u_i, u_{i+1}\}) = \{\rho(u_i), \rho(u_{i+1})\} = \{u_{i+1}, u_{i+2}\} \in E_{P(n,k)} \\ \rho(e_I) &= \rho(\{v_i, v_{i+k}\}) = \{\rho(v_i), \rho(v_{i+k})\} = \{v_{i+1}, v_{(i+1)+k}\} \in E_{P(n,k)} \\ \rho(e_S) &= \rho(\{u_i, v_i\}) = \{\rho(u_i), \rho(v_i)\} = \{u_{i+1}, v_{i+1}\} \in E_{P(n,k)}.\end{aligned}$$

Note that we know that the permutation ρ acts in this way on the edges by the fact that there is an isomorphism between the vertex and edge automorphism groups as discussed in Section 3. Additionally, this permutation preserves edge type, so outer edges map to outer edges, etc.

Next we examine the permutation analogous to flipping the graph, ϕ . This operation still preserves the exterior or interior nature of the vertices, so we note that $\phi(u_i) = u_{n-i}$ and $\phi(v_i) = v_{n-i}$. To check the preservation of edges, we note that

$$\begin{aligned}\phi(e_O) &= \phi(\{u_i, u_{i+1}\}) = \{\phi(u_i), \phi(u_{i+1})\} = \{u_{n-i}, u_{n-(i+1)}\} \in E_{P(n,k)} \\ \phi(e_I) &= \phi(\{v_i, v_{i+k}\}) = \{\phi(v_i), \phi(v_{i+k})\} = \{v_{n-i}, v_{n-(i+k)}\} \in E_{P(n,k)} \\ \phi(e_S) &= \phi(\{u_i, v_i\}) = \{\phi(u_i), \phi(v_i)\} = \{u_{n-i}, v_{n-i}\} \in E_{P(n,k)}.\end{aligned}$$

Again we find that this automorphism preserves the edge type.

Thus it follows that the dihedral group $D_n \subseteq \text{Aut}(P(n, k))$, as D_n is the group formed by the generators ρ and ϕ given above. Furthermore, this group represents all possible automorphisms of the general Petersen graph that preserve the type of edge. This can be seen by the fact that this is the full automorphism group of $C(n)$, as previously discussed, and therefore admits all possible permutations of these edges among themselves, along with the fact that any permutation of the outer edges uniquely determines the permutations of the inner ones.

Theorem 5.1. *If the pair (n, k) is not equal to $(4, 1)$, $(5, 2)$, $(8, 3)$, $(10, 2)$, $(10, 3)$, $(12, 5)$, or $(24, 5)$, then all automorphisms in $\text{Aut}(P(n, k))$ fix the set of edges that are spokes.*

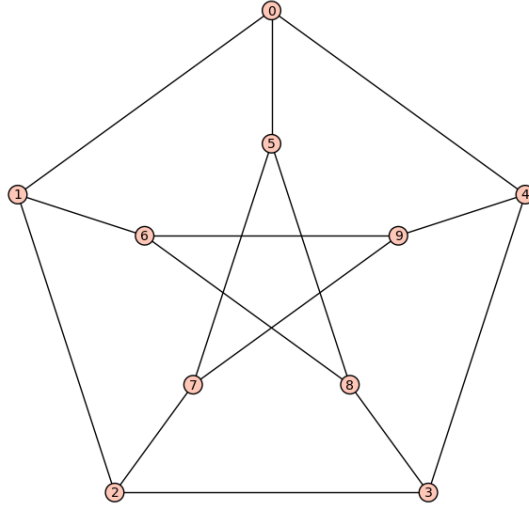


FIGURE 11. The Petersen Graph $P(5, 2)$

Consider the graph $P(5, 2)$ shown in Figure 11. This graph contains an automorphism τ such that $\tau = (3, 7)(4, 5)(8, 9)$ in cycle notation. We can find the associated edge automorphism by defining $\tau(\{u, v\}) = \{\tau(u), \tau(v)\}$, so in cycle notation

$$\tau = (\{0, 4\}, \{0, 5\})(\{2, 3\}, \{2, 7\})(\{6, 8\}, \{6, 9\})(\{8, 5\}, \{9, 4\})(\{3, 4\}, \{5, 7\})(\{3, 8\}, \{9, 7\}).$$

We note that the edges $\{0, 5\}$, $\{2, 7\}$, $\{9, 4\}$, and $\{3, 8\}$ are all spokes, but none of these are mapped to another spoke under τ . Thus we find that Theorem 5.1 does not apply for $P(5, 2)$ as expected. For a full proof of this fact for all 7 cases and the sufficiency of these 7, see Frucht's paper [5].

We have fully described the automorphisms of a given Petersen graph that fix all edges by type, and laid out the cases in which a graph automorphism can permute the spokes to either inner or outer edges. Thus we wish to investigate the cases in which we can find a permutation that exchanges inner and outer edges, as we have exhausted all other possibilities. Now suppose we define an automorphism α that exchanges inner and outer vertices. In order to fix the set of spokes, it must be the case that $\alpha(u_i) = v_{mi}$ and $\alpha(v_i) = u_{mi}$ for some integer m . However, this permutation also must have the effect of mapping the edge $\{u_i, u_{i+1}\}$ to an inner edge of the form $\{v_j, v_{j+k}\}$. Thus we find that $m = k$. However, this is not a valid permutation for all Petersen graphs. We note that if take α operating on an inner edge, we get that $\alpha(\{v_i, v_{i+k}\}) = \{u_{ki}, u_{ki+k^2}\}$. We remember that all of our arithmetic is modulo n , so it follows that α is an automorphism of $P(n, k)$ if and only if $k^2 = \pm 1$, as this is the only case in which α operates properly on the edges of $P(n, k)$. This leads us to our next statement about generalized Petersen graphs.

Lemma 5.2. *If G is a generalized Petersen graph of the form $P(n, k)$ where (n, k) is not one of the exceptional cases listed in Theorem 5.1 and $k^2 \not\equiv \pm 1 \pmod{n}$, then $\text{Aut}(G) \cong D_n$.*

Thus we are left with only the automorphism groups of generalized Petersen graphs for which $k^2 \equiv \pm 1 \pmod{n}$ to classify. We observe that if $k^2 \equiv 1 \pmod{n}$, then α^2 is simply the identity, as

$$\alpha(\alpha(\{u_i, u_{i+1}\})) = \alpha(\{v_{ki}, v_{ki+k}\}) = \{u_i, u_{i+1}\}$$

and

$$\alpha(\alpha(\{v_i, v_{i+k}\})) = \alpha(\{u_{ki}, u_{ki+1}\}) = \{v_i, v_{i+k}\}.$$

Note that we have already discussed the fact that this permutation fixes the set of spokes. However, we also observe that if $k^2 = -1 \pmod{n}$, then α^2 is not equal to the identity, as

$$\alpha(\alpha(\{u_i, u_{i+1}\})) = \alpha(\{v_{ki}, v_{ki+k}\}) = \{u_{-i}, u_{-i-1}\}$$

and

$$\alpha(\alpha(\{v_i, v_{i+k}\})) = \alpha(\{u_{ki}, u_{ki-1}\}) = \{v_{-i}, v_{-i-k}\}.$$

Thus we must consider these two cases separately, as the addition of the automorphism α yields different group structures in both cases.

Theorem 5.3. *If not isomorphic to one of the exceptional cases outlined in Theorem 5.1, the automorphism group of a generalized Petersen graph on n vertices is isomorphic to one of the three following groups:*

- D_n
- $D_n \rtimes Z_2$
- $Z_n \rtimes Z_4$

Proof. If $k^2 \neq \pm 1 \pmod{n}$, the automorphism group must be isomorphic to D_n by Lemma 5.2.

If $k^2 = 1 \pmod{n}$, we have found that both ϕ and α return the identity when applied twice. We have also noted previously that the permutations ρ and ϕ together form the dihedral group D_n . Since α is not an element of that group, it follows that the inclusion of α will generate a group of the form $D_n \rtimes_{\beta} Z_2$, where $\beta : Z_2 \rightarrow \text{Aut}(D_n)$ is some homomorphism. This could be investigated further, but for the purposes of this proof we only need the underlying structure of the group.

Next we examine the case in which $k^2 = -1 \pmod{n}$. In particular, we have noted that α^2 is not equal to the identity, and that $\alpha^2(\{u_i, u_{i+1}\}) = \{u_{-i}, u_{-i-1}\}$ and $\alpha^2(\{v_i, v_{i+1}\}) = \{v_{-i}, v_{-i-1}\}$. However, we remember that this was precisely our definition of ϕ to begin with, and it follows that $\alpha^4 = \phi^2$ returns the identity. Therefore, we find that we can describe the structure of this group entirely using only two generating permutations, ρ and α , and hence the automorphism group of graphs of this form is isomorphic to $Z_n \rtimes_{\beta} Z_4$ for some $\beta : Z_4 \rightarrow \text{Aut}(Z_n)$. □

An important thing to note is that the homomorphisms β described for both of the cases outlined in the proof of Theorem 5.3 are uniquely determined by the behaviour of our three generating permutations when operated upon each other, but their exact form was not important to the argument of the previous proof. This in turn means that there are in fact only three possible automorphism groups of generalized Petersen graphs on n vertices up to isomorphism, and we could not have two different homomorphisms to generate two non-isomorphic groups both of the form $D_n \rtimes Z_2$.

6. CUBIC HAMILTONIAN GRAPHS

In the previous section, the generalized Petersen graphs that we examined all had a high degree of symmetry. Specifically, every vertex of a generalized Petersen graph has degree three, no matter what the n and k parameters are. Such a graph is called a cubic graph. In this section we will examine cubic graphs more generally. We will specifically look at graphs that contain a cycle that visits every vertex exactly once, with the exception of the start and end vertex. If a graph has such a cycle, it is said to be Hamiltonian. If a given graph G on n vertices is both cubic and Hamiltonian, then there exists a very natural way of representing it. We can simply arrange the n vertices in the form of the cycle graph C_n , and subsequently add the third edge from each vertex passing through the middle of the circle to its other endpoint. This is method of representing such graphs that has its own notation, called Lederberg-Coexter-Frucht (LCF) notation after the three who developed it.

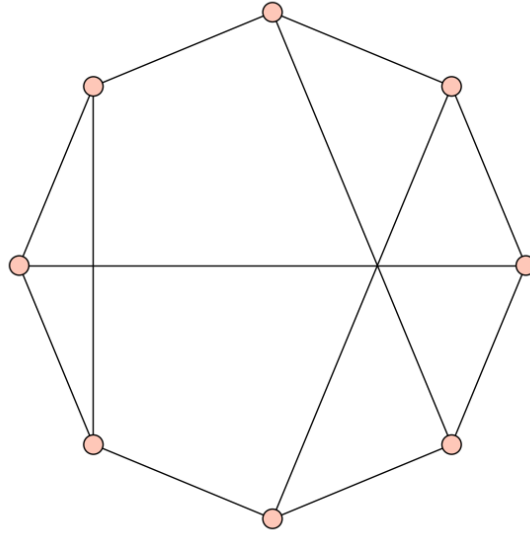


FIGURE 12. The cubic Hamiltonian graph with LCF representation $[3,3,4,-3,-3,2,4,-2]$.

6.1. LCF notation. Since the edges of the Hamiltonian cycle have all been determined in advance, LCF notation serves specifically to specify the interior edges. Starting at the designated first vertex, which by convention is the uppermost one, we proceed clockwise around the figure, assigning each interior edge as we go. This is done by indicating how far away the vertex is that forms the other endpoint of a given edge. For example, Figure 12 shows the graph whose LCF representation is $[3, 3, 4, -3, -3, 2, 4, -2]$. This is interpreted as follows:

- (1) Begin by constructing the graph C_8 , as there are 8 integer entries.
- (2) Next construct an edge between the first and fourth vertices
- (3) Continue constructing edges between the vertices i and $i + j$, where i is the index of the current entry and j is the value of that entry.

It is important to note that this notation is in some sense redundant, as we “construct” each edge twice. However, since we are dealing with simple graphs, it is simply assumed that we do not in fact intend for these to be multiple edges. The repetition is necessary for the convenience of the notation, as we otherwise would have to come up with some sort of cumbersome technique for indicating which of the vertices our different entries corresponded to. However, there are some instances in which it is natural to want to truncate the notation for the sake of efficiency, examples of which follow and are shown in Figure 13. The first is when we have the same pattern of entries repeated multiple times, we represent this as a the repeated element raised to an exponent indicating the number of times it should be repeated. Therefore, instead of writing $[4, 4, 4, 4, 4, 4, 4, 4]$, Lemma we would write $[4]^8$. We also often will have LCF patterns that are a particular sequence of integers followed by the same sequence reversed and negated, as with the LCF graph $[2, 3, -2, 3, -3, 2, -3, -2]$. In this case, we use a semicolon and dash to indicate this symmetry, so this graph would instead be written as $[2, 3, -2, 3; -]$. What follows next is an important, although fairly trivial, theorem of cubic Hamiltonian graphs.

Theorem 6.1. *A cubic graph G is Hamiltonian if and only if it has a representation in LCF notation.*

It should be clear that any graph with an LCF representation is in fact a cubic Hamiltonian graph, as we can simply use the outer cycle (note that there may be other Hamiltonian cycles, but we are guaranteed at least the one). It does not take too much more consideration to note that

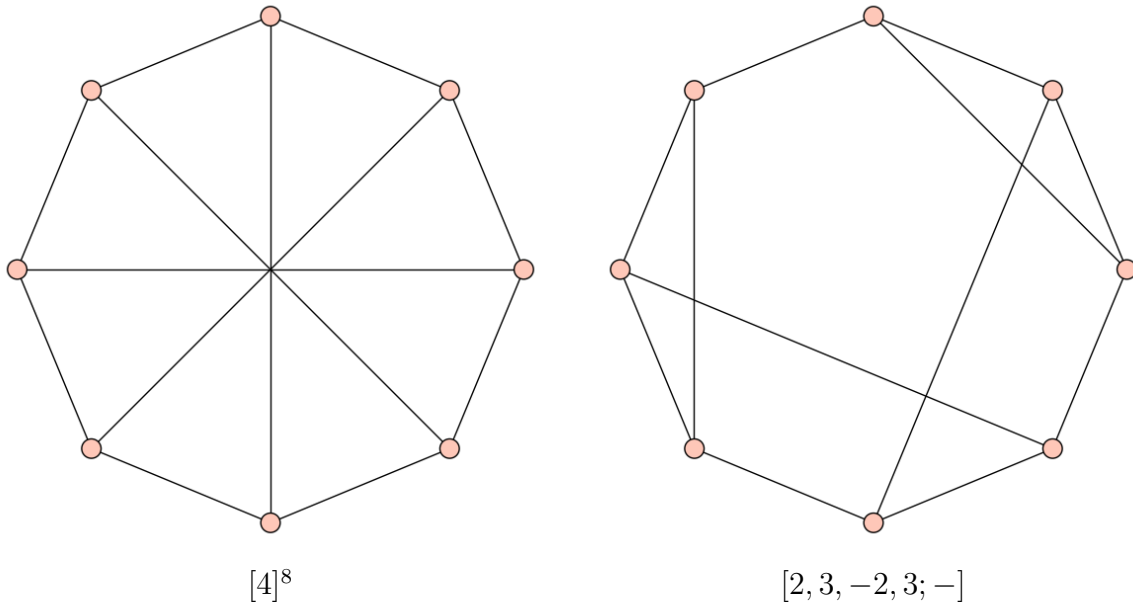


FIGURE 13. Two graphs and their efficient LCF notations..

the converse is also true, and therefore any cubic Hamiltonian graph has an LCF representation. It is also important to note that these graphs will always be on an even number of vertices. This is a result of the fact that if we had an odd number of vertices in a cubic graph, the sum of the degrees of the vertices would be $3(2n+1) = 6n+3$ for some integer n . However, this sum of degrees should also be precisely twice the number of edges, but this clearly cannot be the case as it is an odd number. Thus we find that any cubic graph must be on an even number of vertices.

6.2. Connection to Generalized Petersen Graphs. Now that we have established a convenient notation for cubic Hamiltonian graphs, we would like to see how it applies to graphs that we have already encountered. In particular, we note that our formulation of a generalized Petersen graph $P(n, k)$ is cubic no matter what parameters are chosen for n and k , and also is on an even number of vertices (namely $2n$). We would like to know whether or not we can represent all such graphs in LCF notation. It is fairly easy to see that there is a Hamiltonian cycle in any graph of the form $P(n, 1)$, as we can simply traverse the outer loop until we have reached the $n-1$ vertex, cross to the inside loop, and trace that loop in the opposite direction to return to our original vertex. Thus at least some generalized Petersen graphs are Hamiltonian.

However, this is not always the case. Take as an example the graph $P(5, 2)$ shown in Figure 11. We note that the minimum length of any cycle in $P(5, 2)$ (called the *girth* of the graph) is 5, so there are no cycles of length 3 or 4. Now let us suppose that $P(5, 2)$ were Hamiltonian. If this were the case, then it must have a representation in LCF notation by Theorem 6.1. Since it is a graph on 10 vertices, we find that our possible entries in one particular slot of the LCF notation are the integers -4 through 5 . However, if a particular entry is the integer k , we can easily construct a cycle of length $k+1$ by combining the edge represented by that entry with the edges of the outer Hamiltonian cycle that connect the two endpoints of the edge, of which there are k . Thus the LCF representation of our graph $P(n, k)$ must only include entries of $-4, 4$, and 5 . Through a bit of trial and error, we find that the only three LCF graphs on 10 vertices that meet this criteria are $[5, -4, 4, -4, 4]^2$, $[5, 5, 5, -4, 4]^2$, and $[5]^10$, shown in Figure 14. We will examine whether or not any of these could be isomorphic to $P(5, 2)$ by attempting to find cycles of length 4. In the first and second graphs, we find that we can simply pick a $-4, 4$ edge pair. The endpoints of both of these edges must be adjacent on the external cycle as there are only 10 vertices in the graph. Thus we

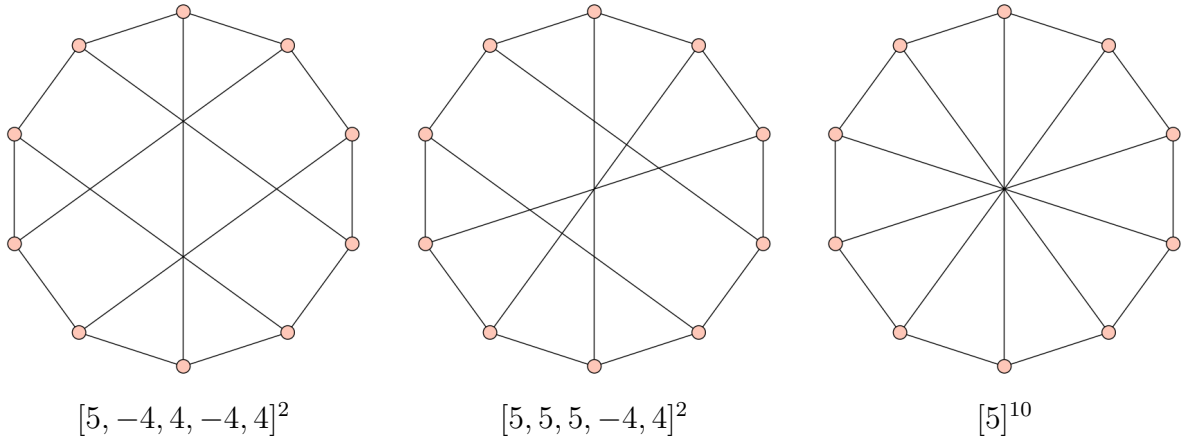


FIGURE 14. The three LCF representations of cubic Hamiltonian graphs on 10 vertices that include no integers whose absolute value is less than 4.

can find a cycle using only 4 edges, and thus neither of these graphs can be isomorphic to $P(5, 2)$. In the last case, we can start at vertex v_0 , follow the internal edge to v_5 , then proceed to v_6 . We are guaranteed that there is an edge from v_6 to v_1 by the way we have constructed the graph. Since v_0 and v_1 are connected by an edge in the external cycle, we find that there is a cycle involving only 4 edges, and thus that it is not isomorphic to $P(5, 2)$. Hence we find that $P(5, 2)$ is not Hamiltonian and has no *LCF* representation.

Unfortunately, there is no easy characterization for when a generalized Petersen graph is Hamiltonian. Therefore our discussion of the automorphism groups of LCF graphs must start from its own foundation.

6.3. Automorphism Groups of Small LCF Graphs. We will begin this investigation by examining all LCF graphs on a small number of vertices and seeing if any patterns emerge. The mathematical computing program Sage will be used as an aid. Conveniently, Sage has a command `graphs.LCFGraph(n , [a], b)` that generates an LCF graph on n vertices with LCF representation $[a]^b$ (it does not support the semicolon notation introduced above for symmetric codes). After generating the graph, we use the `.automorphism_group()` command to generate the automorphism group, at which point we can use the `.order()` and `.is_isomorphic()` commands to investigate the automorphism group further. This was used to check possible structures of the automorphism groups of the graphs.

The first cubic graph is on four vertices, and we can fairly quickly see that it is unique. In LCF notation, the only possible representation is $[2]^4$, as this is the only way to connect the vertices of a square such that they all are of degree three. We know this graph to be isomorphic to K_4 , so it follows by previous results that the automorphism group is isomorphic to S_4 .

Next we examine LCF graphs on six vertices. We note that any LCF representation on this number of vertices must include at least one entry of value 3, as there is no way to connect all 6 vertices using only 2. If we begin with a value of 3, we are forced to choose either 3 or -2 for the second value, as 2 would terminate in the same vertex as the 3 edge. If we choose -2 , then this forces our third value to be 2, resulting in the graph $[3, -2, 2]^2$. This graph can be viewed as two K_3 graphs whose vertices are connected to one vertex of the other. Therefore one any automorphism of K_3 is an automorphism of this graph, as we can simply permute the two different K_3 graphs with the same automorphism. Alternately, we could swap each of the vertices with the vertex adjacent to it in the other K_3 , independently of any other automorphism. Therefore we find that the automorphism group is isomorphic to $S_3 \times S_2$. Next, if we choose 3 for the second value, then

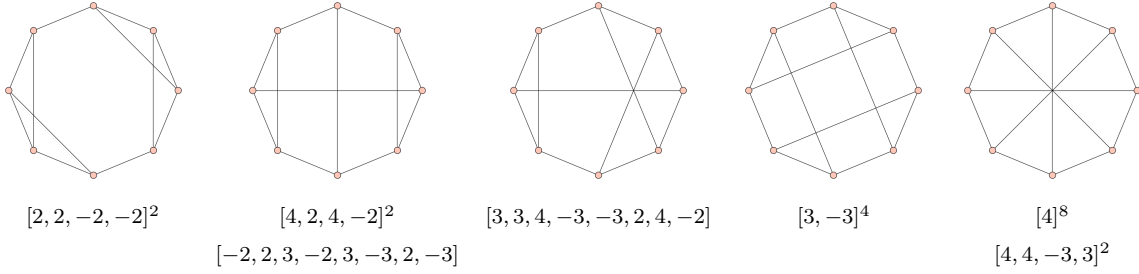


FIGURE 15. The five non-isomorphic cubic Hamiltonian graphs on eight vertices and their LCF representations.

our third value is determined to be 3, resulting in the graph $[3]^6$. This graph can be represented in a similar way as last time, with two sets of three vertices, all of which are connected to the three in the other set. Therefore I can permute any of the three vertices in each set with each other independent of the permutations of the other set, resulting in a structure of $S_3 \times S_3$. However, we can also switch the two sets of vertices with each other, but as with the discussion of the Dihedral Group earlier, this flip does not commute with the permutations previously mentioned. Thus the automorphism group of this graph has a structure $(S_3 \times S_3) \rtimes S_2$.

Since there are more LCF graphs on eight vertices, we will not walk through the logic used to determine all possible LCF representations. However, the process used is similar to that used above in the case of six vertices. We find that there are seven distinct LCF representations, but only five non-isomorphic graphs. In order to show that two distinct LCF representations result in an isomorphic graph, it suffices to find a Hamiltonian cycle in the graph other than the exterior one and re-plot the graph with that cycle as the exterior cycle. We know this new graph to be isomorphic to the original one as we did not add or remove any vertices or edges, and thus if the new graph has a different LCF representation, the two must in fact be isomorphic. Starting with the LCF representation $[4, -2, 4, 2]^2$, we find a Hamiltonian cycle of the form $(v_1, v_5, v_6, v_4, v_3, v_7, v_8, v_2, v_1)$. By plotting the graph with this cycle on as the exterior edges, we find that the new representation of this graph is $[-2, 2, 3, -2, 3, -3, 2, -3]$. Thus these two representations are isomorphic. Similarly, the graph $[4]^8$ has a Hamiltonian cycle $(v_1, v_5, v_6, v_7, v_8, v_4, v_3, v_2, v_1)$. Plotting the graph with this as the external cycle we find a new graph with LCF notation $[4, 4, -3, 3]^2$, and thus the two notations yield isomorphic graphs.

Next we can investigate the automorphism groups of each of the five non-isomorphic LCF graphs on eight vertices shown in Figure 15.

- (1) $[2, 2, -2, -2]^2$: We first note that a rotation by 180° is an automorphism of this graph. Note that there are two induced subgraphs each on four vertices isomorphic to K_4 with one edge removed. If we consider the two vertices of degree three in each of these induced subgraphs, we note that they can be freely exchanged. If we combine the rotation by 180° with one of these two permutations, we end up with an automorphism of degree four. This together with the permutation of the other two vertices forms our subgroup isomorphic to D_4 . We can also flip the graph on the axis that separates the two induced subgraphs mentioned above, and thus we have found our whole group structure: $D_4 \times S_2$.
- (2) $[4, 2, 4, -2]^2$ or $[-2, 2, 3, -2, 3, -3, 2, -3]$: Given the representation shown in Figure 15, there are two flips that come immediately to attention, both along the axes that run through the edges of length four. We also note that there is a rotation by 180° that is also an automorphism, but we can express this as the combination of both of our flips previously defined. These are in fact all of the automorphisms of this particular graph, so the automorphism group structure is $S_2 \times S_2$.

- (3) $[3, 3, 4, -3, -3, 2, 4, -2]$: To investigate this graph, we begin by looking at the induced subgraph on three vertices that forms a K_3 on the left hand side of the representation in Figure 15. We know that the automorphism group of this subgraph is S_3 by previous results, and note that any automorphism of those three uniquely determines where the three other vertices to which they are connected are sent. This leaves us with only two more vertices to consider. Since they are both connected to the same three other vertices, we can exchange the two while preserving the graph structure. This yields an overall automorphism group structure of $S_3 \times S_2$.
- (4) $[3, -3]^4$: If we consider the subgraph formed by choosing every other vertex as we progress around the cycle, we note that we have four vertices all of which are mutually non-adjacent. Thus the permutation group generated by examining this subgraph is isomorphic to S_4 . We note that any automorphism of these four vertices will determine where the other four are sent by symmetry. However, we could also exchange our two sets of four mutually non-adjacent vertices, as there was nothing that distinguished our starting vertex in the first step. Thus we find that our automorphism group structure is $S_4 \times S_2$.
- (5) $[4]^8$ or $[4, 4, -3, 3]^2$: The high level of symmetry here is reminiscent of the structure of the Dihedral group. In fact, we can define a rotation by 45° and a flip that together form exactly that structure. These are in fact the only automorphisms of this graph, as any other automorphism must permute the vertices in such a way to determine a new Hamiltonian cycle on the exterior that still preserves the graph structure, and it can be easily verified that no such cycle exists (see the proof of Theorem 6.3 for a proof of the general result). Hence, the automorphism group structure is D_8 .

6.4. Frucht's Theorem. Back in Corollary 4.8 we established that there are an infinite number of non-isomorphic graphs with a given automorphism group, and shortly thereafter discussed its implications relating to Frucht's Theorem. However, there are extensions to the theorem that strengthen the assumptions. Specifically, there are certain restrictions that can be enforced on each graph that still result in family of graphs that contain automorphism groups isomorphic to any given group. One example of such a family of graphs is any collection of k -regular graphs, where $k \geq 3$. One natural extension to this question is to ask whether it holds true for k -regular Hamiltonian graphs, or more specifically applicable to our situation, the family of LCF graphs. This is formally stated as follows:

Conjecture 6.2. *Given any group H , there exists a cubic Hamiltonian graph G such that $\text{Aut}(G) \cong H$.*

What follows in this section is an investigation of the truth of Conjecture 6.2.

So far in our investigation we have seen only groups of even order, which seems to make intuitive sense with the highly symmetric nature of the external cycle common to all LCF graphs. This encourages us to believe that the automorphism groups are in fact generally non-trivial, but this turns out to be false.

In order to construct a LCF graph with a trivial automorphism group, we must make sure that we construct it so as to eliminate all possible symmetry. Ideally we could do this by splitting the external cycle into two sets of $\frac{n}{2} + 1$ and $\frac{n}{2} - 1$ vertices, respectively, and then ensure that there is no symmetry within either of the two sets. We know from our earlier investigation that there is no graph with a trivial automorphism group on 8 or fewer vertices, and through trial and error we can establish that we cannot use this method to find such a group on 10 vertices, and therefore we examine the 12 vertex case. To begin we wish to divide the graph into unequal parts, and do so by assigning a value of 5 to our first entry. Now we have 2 more vertices in one set than the other, and we will connect them using an edge of length two. Now we have the same number of vertices without interior edges in both sets, and we simply connect these across in a natural fashion.

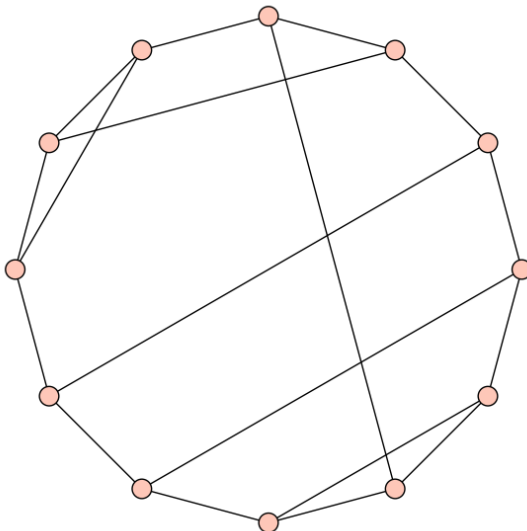


FIGURE 16. The LCF graph $[5, -3, 6, 4, 2, -5, -2, -4, -6, 2, 3, -2]$ on 12 vertices with a trivial automorphism group.

This process yields the graph $[5, -3, 6, 4, 2, -5, -2, -4, -6, 2, 3, -2]$ shown in Figure 16. We note that this graph has no rotational symmetry and no symmetry about any particular axis. Therefore the only automorphism we could possibly have would come from finding a Hamiltonian cycle not represented by the exterior cycle of the graph. In this particular case such a cycle does exist, but the graph that results from plotting this along the exterior is not isomorphic to our original one. We can then use this graph on 12 vertices as a basis to construct a cubic Hamiltonian graph on n vertices with a trivial automorphism group for any even $n \geq 12$. In general, the LCF representation will have the form $[\frac{n}{2} - 1, -3, n - 6, n - 7, \dots, 4, 2, -(\frac{n}{2} - 1), 2, 4, \dots, -n + 7, -n + 6, 2, 3, -2]$. A quick check shows us that this matches the graph on 12 vertices discussed previously when $n = 12$, and has a trivial automorphism group by the same reasoning as before but extended to a general number of vertices.

It turns out that we can also take advantage of similarities in the formulation of a graph to make general claims about their automorphism groups. In the theorem that follows, we establish and prove one such generalization.

Theorem 6.3. *The graph with LCF notation $[\frac{n}{2}]^n$ has an automorphism group isomorphic to D_n for all $n = 2m$, where $m \geq 4$.*

Proof. First we consider the cases where $m \leq 3$. For $m = 1$, we have a graph on only two vertices which cannot be a simple cubic graph, as there is only one possible edge. In the cases where $m = 2$ and $m = 3$, this graph corresponds to examples seen earlier in this section, where the automorphism groups were shown to be isomorphic to S_4 and $S_3 \times S_3 \times S_2$, respectively. Next we turn to the case where $m \geq 4$, and thus $n \geq 8$. Note that the case where $n = 8$ is shown in Figure 15. We can clearly see that this construction results in a symmetric structure that allows for all of the rotations and flips that form the group D_n , so all that remains is to show that there are no other graph automorphisms. We note that any such automorphism would have to change the structure of the external cycle, which is equivalent to saying that one of the internal edges must be mapped to an exterior edge. Suppose that ϕ is such a permutation. Without loss of generality, we can assume that the internal edge $\{1, \frac{n}{2}\}$ is exchanged with the external edge $\{1, 2\}$, noting that the edge $\{0, 1\}$ is still an external edge under ϕ . However, we know that the vertex 2 must still be adjacent to the vertices 1, 3, and $\frac{n}{2} + 1$ under ϕ , and thus the images of these vertices must be adjacent to $\frac{n}{2} + 1$. It

follows that either $\phi(3) = \frac{n}{2} - 1$ or $\phi(\frac{n}{2} + 1) = \frac{n}{2} - 1$, and thus ϕ produces an edge either of the form $\{0, 3\}$ or $\{0, \frac{n}{2} + 1\}$, neither of which are edges in our original graph. Thus we have found that any permutation that does not maintain the structure of the external cycle is not an automorphism of our graph. Since there are no other allowed automorphisms, we conclude that the automorphism group of each graph of the form $[\frac{n}{2}]^n$ for $n = 2m$, where $m \geq 4$, is isomorphic to D_n . \square

6.5. Partial Results and Further Questions. We have successfully managed to find a general result concerning a particular form of LCF graph. This suggests that there are perhaps other generalizations to be found, but it is hard to formulate them given the nature of these kinds of graphs. LCF notation, while extremely convenient, is often hard to extend to larger numbers of vertices and therefore difficult to predict the structures of higher-order graphs that could be created. We have accomplished one goal, but this is far from proving or disproving the results of Frucht's Theorem as applied to cubic Hamiltonian graphs.

There are two questions that promise to either support the extension of Frucht's Theorem to cubic Hamiltonian graphs or be a natural source of counterexamples. The first is if it is possible to create an LCF graph with a non-trivial automorphism group of odd order. It seems possible that some degree of symmetry is forced by the existence of the Hamiltonian cycle that requires the existence of an even-order automorphism along with any odd one, forcing the group containing them to be of even order, but the existence of such a graph would disprove this intuition. If this result could be proven, it would serve to disprove this particular restriction of Frucht's Theorem. The other question is the possibility of creating a LCF graph with an automorphism group isomorphic to D_n for some odd n , as this was not covered in Theorem 6.3. This would have similar implications to the first.

APPENDIX A. SAGE

This appendix serves as a quick guide to some of the Sage functions that were most useful in investigating the automorphism groups of graphs. The first two subsections introduce functions specific to group theory and graph theory respectively, while the third demonstrates some examples of how they were used in this investigation.

A.1. Group Theory. Sage contains many constructors and tools relating to group theory, but only a few will be investigated here. The dihedral group D_n and the symmetric group S_n have their own sage commands, and we can also form a group as the direct product of an arbitrary number of others, as shown in Listing A.1.

```
G = DihedralGroup(n);
H = SymmetricGroup(n);
I = direct_product_permgroups([SymmetricGroup(2), DihedralGroup(4)])
```

Listing A.1: Three different examples of group generators

For any given group, there exist a whole family of functions. that help investigate its structure and properties. A few of these pertinent to this investigation are shown in Listing ??, but the Sage documentation contains many more such functions.

A.2. Graph Theory. Much like groups, Sage has many default constructors for graphs. The ones used most commonly for this project are shown in Listing A.3, but there are many more constructors to be found in the documentation. Note that the inputs for each of the graphs are more exactly of the form presented in general in the body of this paper.


```
G = DihedralGroup(n);
H = SymmetricGroup(n);
#tests isomorphism
G.is_isomorphic(H);
#returns the order of the graph
G.order();
#returns the generators of the graph in permutation notation
G.gens();
```

Listing A.2: Three different examples of group functions

```
G = graphs.CompleteGraph(n);
H = graphs.CycleGraph(n);
I = graphs.PathGraph(n);
J = graphs.GeneralizedPetersenGraph(n,k);
K = graphs.LCFGraph(n, [n/2], n);
#creates a group A isomorphic to the automorphism group of G
A = G.automorphism_group();
```

Listing A.3: Five different examples of graph generators

```
G = graphs.LCFGraph(8, [3, 2], 4);
G.plot();
```

Listing A.4: A graph generated using the LCF function that is not cubic, whose output is shown in Figure 17

One important note is that the Generalized Petersen Graph constructor has error-checking, meaning that it will return an error if your parameters violate the rule that $k \leq n/2$. Unfortunately, the LCF constructor does not have this kind of error built in. If the notation that the user inputs is not a true LCF graph, Sage will draw the graph anyway without any warning that it the resultant graph is no longer a cubic Hamiltonian graph. For example, if the user inputs the code shown in Listing A.4, Sage will run properly and treat G as a graph without any complaint, even though some of the vertices will have degree five. Another quirk of the LCF constructor is that it generates graphs that are precisely the mirror of what we would expect given the canonical LCF representation. This is because Sage proceeds labelling vertices counter-clockwise, while the LCF notation specifically states that we proceed clockwise. This difference has no effect on the structure of the graph, so commands like the one shown in Listing A.3 to determine the automorphism group are unaffected. For this reason, all LCF graphs displayed in figures in this paper have no vertex labels, as the images were inverted to match standard notation and this would have also inverted the numerals.

A.3. **Examples.** What follows in Listings A.5 and A.6 are two different examples of Sage code used in this investigation.

REFERENCES

- [1] Behzad, Mehdi, Gary Chartrand, and Linda Lesniak-Foster. *Graphs and Digraphs*. Boston, MA: Prindle, Weber and Schmidt, 1979. Print.
- [2] Bogart, Kenneth Paul. *Introductory Combinatorics*. Boston: Pitman, 1983. Print.
- [3] Bona, Miklos. *Introduction to Enumerative Combinatorics*. Boston: McGraw-Hill Higher Education, 2007. Print.


```
G = graphs.LCFGraph(12, [-5,2,-3,-2,6,4,2,5,-2,-4,6,3],1);
I = G.automorphism_group();
#outputs both the graph and the order of the automorphism group
G.plot(); I.order();
```

Listing A.5: Method of generating the LCF graph with a trivial automorphism group shown in Figure 16

```
#i takes on values from 0 to 99
for i in range(100):
    a = 4+2*i;
    G = graphs.LCFGraph(a, [a/2],a);
    I = G.automorphism_group();
    #returns this truth value in a separate row for each input i
    I.is_isomorphic(DihedralGroup(a));
```

Listing A.6: A test of Theorem 6.3 for the first 100 possible n values

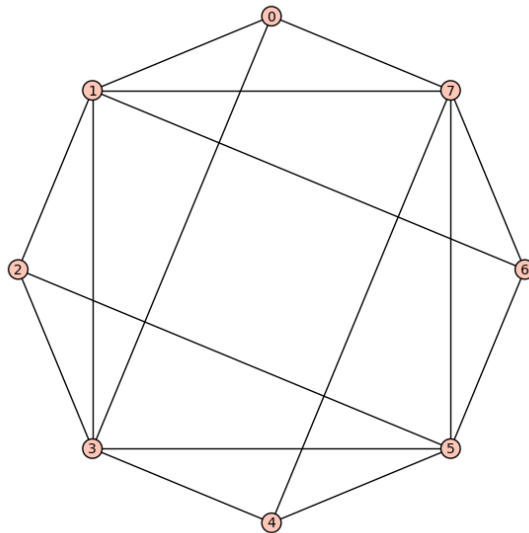


FIGURE 17. The graph generated by the code in Listing A.4.

- [4] Frucht, Roberto “Herstellung von Graphen mit vorgegebener abstrakter Gruppe” *Compositio Mathematica*, Tome 6 (1939). p. 239-250. http://archive.numdam.org/ARCHIVE/CM/CM_1939__6_/CM_1939__6__239_0/CM_1939__6__239_0.pdf
- [5] Frucht, Roberto, Jack E. Graver, and Mark E. Watkins. “The Groups of the Generalized Petersen Graphs.” *Mathematical Proceedings of the Cambridge Philosophical Society* 70.02 (1971): 211. Print.
- [6] Gallian, Joseph A. *Contemporary Abstract Algebra*. Boston: Houghton Mifflin, 2002. Print.
- [7] Steinberger, Mark. *Algebra*. Boston: PWS, 1994. Print.