Exploring Graph Isomorphism, Equivalence Classes and Automorphisms through Abstract Algebra

Shramana Dey, JRF in Computer Science - MIU *

April 21, 2024

Abstract ***Graph theory serves as a rich domain for exploring abstract algebraic concepts, particularly concerning graph isomorphism, equivalence classes, and automorphisms. This write-up delves into the fundamental principles of abstract algebra applied to graph theory, elucidating the intricate interplay between these concepts. It elucidates the significance of identifying structurally equivalent graphs, presents algorithms for isomorphism detection, and discusses the notion of equivalence classes as sets of indistinguishable graphs. Furthermore, it explores automorphisms as isomorphisms within a graph, emphasizing their role in preserving graph properties and symmetry. By integrating abstract algebraic principles, this concise exploration provides insights into the interplay between algebraic structures and graph theory, contributing to a deeper understanding of complex graph structures.***

1 Introduction

The fundamental question of graph isomorphism is determining whether two graphs are structurally identical. Graph isomorphism is not merely a matter of visual similarity but encompasses a rigorous mathematical equivalence. This introduction sets the stage for our exploration of graph isomorphism, highlighting its importance in various domains such as network analysis, computer science, and chemistry. Through an overview of key concepts and challenges in graph isomorphism, we pave the way for a deeper examination of its connections with abstract algebra, equivalence classes, and automorphisms.
It is fascinating to note that, graphs can be analyzed using algebraic structures. For instance, the adjacency matrix of a graph represents the graph's connections between vertices, and it can be manipulated using matrix algebra i.e. the set of matrices closed under some operation. Similarly, the incidence matrix describes the relationships between vertices and edges, providing another avenue for algebraic analysis.
Graph isomorphism, the notion that two graphs are structurally identical, forms an equivalence relation. Equivalence classes emerge, partitioning the set of all graphs into distinct classes based on isomorphism. This concept resonates with algebraic notions of

*Course Instructor: Sujata Ghosh

equivalence classes, fostering a deeper understanding of structural similarities between graphs. Group theory, a cornerstone of abstract algebra, finds application in studying graph symmetries. The set of all automorphisms of a graph forms a group under composition, known as the automorphism group. Understanding the structure of this group sheds light on the symmetrical properties of the graph.
The connection between graphs and abstract algebra runs deep, with algebraic structures providing powerful tools for analyzing and understanding graph properties and relationships.

2 Graph Isomorphism

Given a graph $G(V, E)$ and $G'(V'.E')$, G and G' are said to be isomorphic if they satisfy the following necessary conditions:

- They must have the same number of vertices. $|V| = |V'|$
- They must have the same number of edges. $|E| = |E'|$
- They must have equal number of vertices of a given degree.

Consider an example:

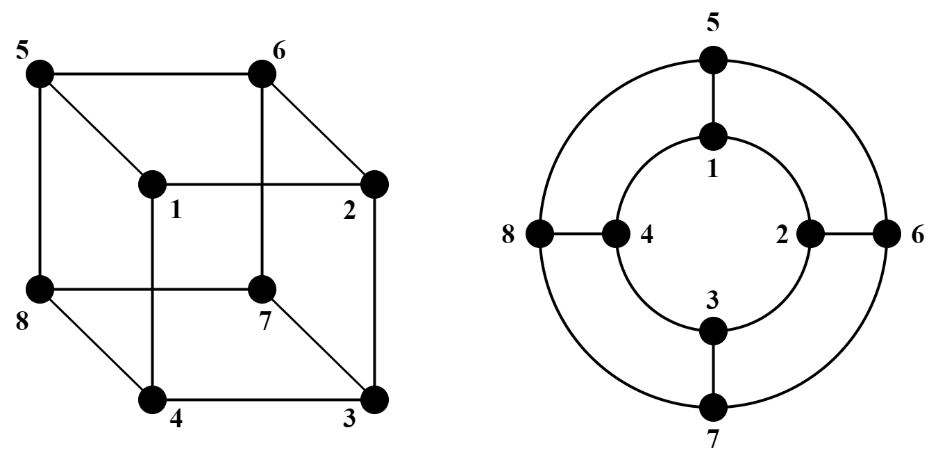


Figure 1: A Pair of Isomorphic Graphs, $G(V, E)$ and $G'(V', E')$

Despite having different labels in Figure 2, these two graphs are isomorphic. There exists a bijection map between the vertex set of the graph G and the vertex set of graph G', defined as $f : V_G \to V_{G'}$.
Example of one such f is:
$f(1) = s, f(2) = t, f(3) = u, f(4) = v, f(5) = w, f(6) = x, f(7) = y, f(8) = z$ Note that, f has the property that neighbors $N(v)$ of a vertex v in graph G is mapped to the neighbors $N(f(v))$ where, $f(v)$ is the image of v in G'.

Vertices	Neighbors
1	2 4 5
2	1 3 6
3	2 4 7
4	1 3 8
5	1 6 8
6	2 5 7
7	3 6 8
8	4 5 7

Table 1: From the table it is clear that the graphs in Figure 1 are isomorphic since they have the same set of neighbors.

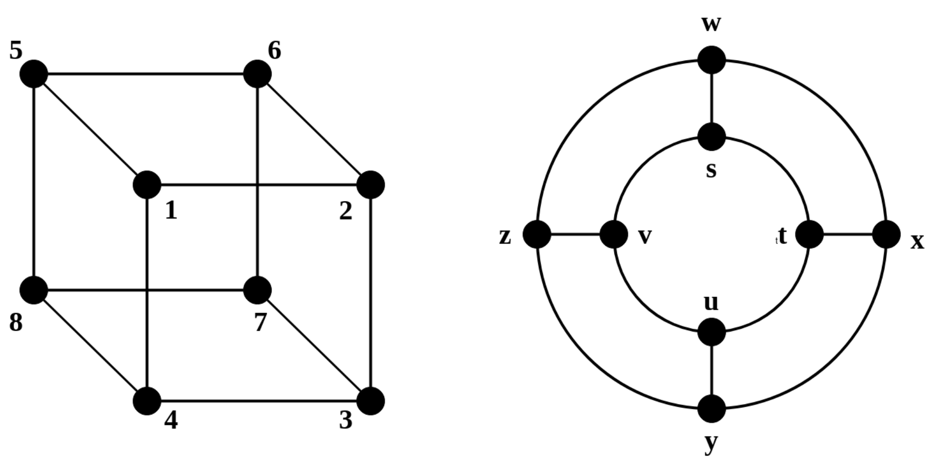
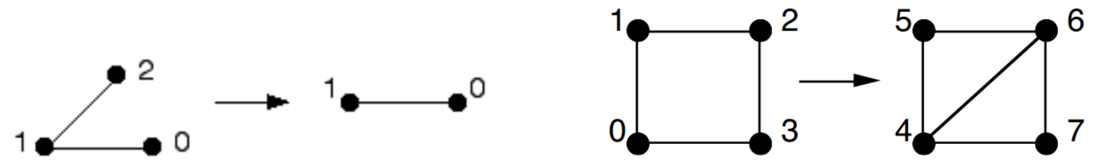


Figure 2: Same Pair of Isomorphic Graphs, $G(V, E)$ and $G'(V', E')$, with Different Labelling

Two vertices in a graph are said to be adjacent if they have an edge between them, otherwise, they are non-adjacent. Thus two graphs that differ by vertex labels or edge labels might be isomorphic to each other.
Let G and H be two simple graphs. A function $f : V_G \to V_H$ preserves **adjacency** if for every pair of adjacent vertices u and v in graph G, the vertices $f(u)$ and $f(v)$ are adjacent in graph H. Similarly,f preserves **non-adjacency** if $f(u)$ and $f(v)$ are non-adjacent whenever u and v are non-adjacent.
A bijection $f : V_G \to V_H$ between two simple graphs G and H is structure-preserving if it preserves both adjacency and non-adjacency. That is, for every pair of vertices u, v in G, u and v are adjacent in $G \iff f(u)$ and $f(v)$ are adjacent in H.

Formally we define,
Two simple graphs G and H are isomorphic, denoted $G \cong H$,if $\exists a$ structure-preserving bijection $f : V_G \to V_H$. Such a function f is called an **isomorphism** from G to H.

Let's have a look at two different pairs of graphs and the function that maps the vertex set of one another in Figure 3.



(a) Preserves adjacency and non-adjacency, but not bijective

(b) Bijective and adjacency-preserving, but not an isomorphism

Figure 3: The vertex map is defined as (a) $j \to j \bmod 2$, (b) $j \to j + 4$

Now, let's consider more general graphs having self-loops and multiple edges. Does the above definition of isomorphism still hold?

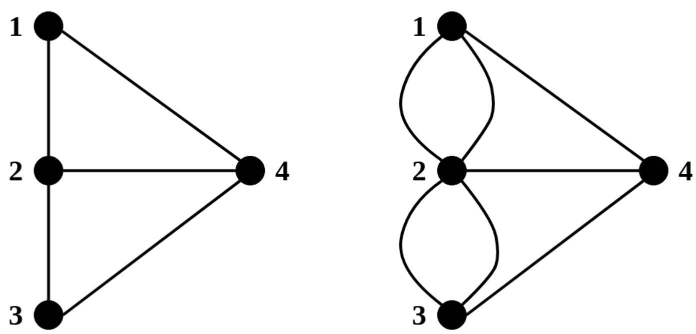


Figure 4: Graph $G(V, E)$ and $G'(V', E')$ have a bijection map on vertices

The mapping $f : V_G \to V_H$ between the vertex-sets of the two graphs shown in Figure 4 given by $f(i) = i, i = 1, 2, 3, 4$.
This shows the map preserves adjacency and non-adjacency, but the two graphs are clearly not structurally equivalent. This is due to the presence of the parallel edges.

To adapt to general graphs, we define:
A bijection $f : V_G \to V_H$ between two graphs G and H, is structure-preserving if (1) the number of edges (even if 0) between every pair of distinct vertices u and v in graph G equals the number of edges between their images $f(u)$ and $f(v)$ in graph H, and (2) the number of self-loops at each vertex x in G equals the number of self-loops at the vertex $f(x)$ in H.

Isomorphism for Graphs with Multi-Edges
Two general graphs G and H are isomorphic if there exists, a pair of bijections $f_V : V_G \to V_H$ and $f_E : E_G \to E_H$ which are consistent. We call f_V and f_E consistent if for every edge $e \in E_G$, the function f_V maps the endpoints of e to the endpoints of the edge $f_E(e)$.
Fact: If G and H are isomorphic simple graphs, then every structure-preserving vertex bijection $f_V : V_G \to V_H$ induces a unique consistent edge bijection. Indeed, we see that: $f_E(uv) \to f_V(u)f_V(v)$ is the consistent edge bijection induced by f_V.

From Figure 5, let us find the number of consistent edge bijections.

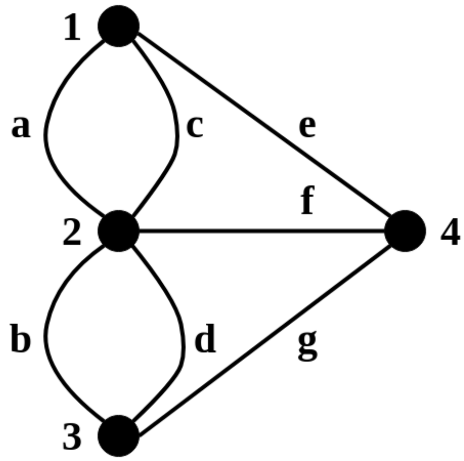


Figure 5: Graph $G(V, E)$ with labelled vertices and edges

There are 8 distinct isomorphisms from G to H.

2.1 Properties of Graph Isomorphism

Let's discuss some of the properties of graph isomorphism based on general graphs.

Proposition 2.1 *Let G and H be isomorphic graphs. Then they have the same number of vertices and edges.*

Proof. The same number of vertices and edges are given by the bijective maps f_V and f_E. □

Proposition 2.2 *Let $f : G \to H$ be a graph isomorphism and let $v \in V_G$.Then $deg(f(v)) = deg(v)$.*

Proof. Since by definition f is structure-preserving, the number of proper edges and the number of self-loops incident on vertex v equal the corresponding number for vertex $f(v)$. Thus, $deg(f(v)) = deg(v)$. □

Proposition 2.3 *Let G and H be isomorphic graphs. Then they have the same degree sequence.*
OR
If G and H are isomorphic graphs, then the degrees of the vertices of G are the same as the degrees of the vertices of H.

Proof. Let $\phi : V_G \to V_H$ be an isomorphism.
Let $v \in V_G$ and consider $u = \phi(v) \in V_H$.

Let v be adjacent to $w_1, w_2, ..., w_k \in V_G$. And v be non-adjacent to $x_1, x_2, ..., x_l \in V_G$.
Then $deg_G(v) = k$.
Then u is adjacent to $\phi(w_1), \phi(w_2), ..., \phi(w_k) \in V_H$.
Again, consider $vw_i \in E_G$ for $i = 1, 2, ..., k$ implies, $\phi(vw_i) = \phi(v)(w_i) \in E_H$ since ϕ is structure preserving.
Since isomorphism preserves adjacency and non-adjacency, $|V_H| = k + l$ and $deg_H(u) = k$.
Thus G and H have the same degree sequence. □

Proposition 2.4 *Let $f : G \to H$ be a graph isomorphism and $e \in E_G$. Then the endpoints of edge $f(e)$ have the same degrees as the endpoints of e.*

Proof. This can be shown similarly to the previous proposition. □

Extending the definition of **isomorphism to directed graphs**, two digraphs G and H are isomorphic if there is an isomorphism f between their underlying graphs that preserves the direction of each edge.

3 Equivalence Classes formed by Isomorphic Graphs

The relation isomorphism between graphs is an equivalence relation.
Prove the equivalence relation.
Given a graph X, the set of graphs isomorphic to X is called the isomorphism class of X. We saw the three conditions that two graphs must satisfy to be isomorphic, here we will see that these conditions are necessary but by no means sufficient for two graphs to be isomorphic. Consider the graph K_{33} and the Prism graph.

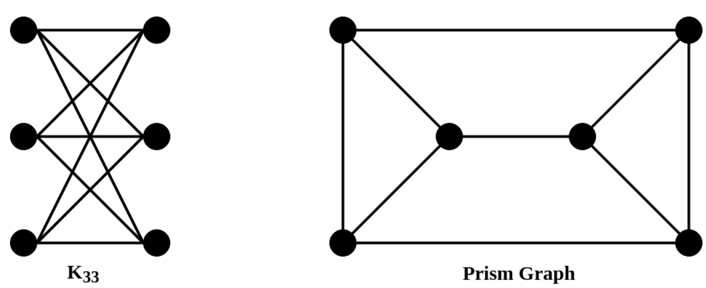


Figure 6: Graphs that satisfy the necessary condition of isomorphism but belong to two different equivalence classes

Note that K_{33} has a cycle of length 4 and is thus bipartite. Whereas, the prism graph has a cycle of length 3 and is thus non-bipartite. Interestingly, every 3 regular graphs is isomorphic to one of these graphs. Thus giving rise to the partition and forming the equivalence class.
The equivalence class of the isomorphic class of four vertices is shown in Figure 7.
Each equivalence class under $\cong$ is called an isomorphism type. As we saw in the previous example, counting isomorphism types of graphs generally involves the algebra of permutation groups.

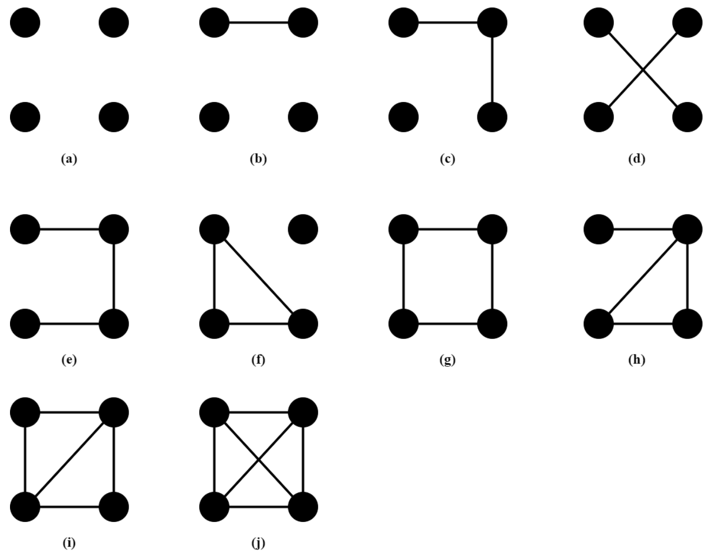


Figure 7: Equivalence class formed by the isomorphic graphs with 4 vertices

4 Graph Automorphism

An isomorphism from a graph G to itself is called an automorphism.
For example, consider the graph, G in Figure 8.
The automorphisms of the graph, G are:
1. The identity automorphism that maps every vertex of the graph to itself.
$e : u \to u, v \to v, w \to w, x \to x, y \to y$
2. Another automorphism, where the vertices are exchanged.
$\alpha : u \to y, v \to x, w \to w, x \to v, y \to u$
Note: The vertices that are flipped have the same degree. Such as, the $deg(u) = deg(y), deg(v) = deg(x)$. Again since w is the only vertex with $deg(w) = 3$, it had to be fixed.

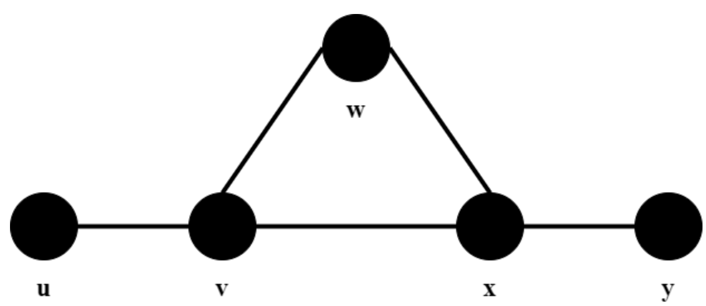


Figure 8: Graph $G(V, E)$

We can see that there are no more automorphisms of graph G that preserve the structural equivalence among the vertices, as well as among the edges.
Now we define, $Aut(G)$, where G is a graph, as the set of all the automorphisms of G.
Then from the previous example, $Aut(G) = \{e, \alpha\}$

Note that each automorphism is in itself a bijection and also structure-preserving, thus we can compose two automorphisms. The composition table of $Aut(G)$ in this example is shown below.

$\circ$	e	α
e	e	α
α	α	e

To show that $Aut(G)$ forms a Group under Composition.
To show that it is indeed an automorphism let's verify the properties:

1. **Closure**: Let $\phi, \psi \in \text{Aut}(G)$. Since ϕ and ψ are automorphisms, they preserve the structure of the graph G. Therefore, their composition $\phi \circ \psi$ also preserves the structure of G, making $\phi \circ \psi$ an automorphism. Hence, $\text{Aut}(G)$ is closed under composition.

2. **Associativity**: Composition of functions is associative, so this property holds trivially.

3. **Identity**: The identity automorphism e_G is defined as the function that maps each vertex to itself. It preserves the structure of G, making it an automorphism. For any automorphism ϕ in $\text{Aut}(G)$, $\mathrm{e}_G \circ \phi = \phi$, and $\phi \circ \mathrm{e}_G = \phi$, satisfying the identity property.

4. **Inverse**: Let $\phi \in \text{Aut}(G)$. Since ϕ is a bijection, it has an inverse ϕ^{-1}. Since ϕ and ϕ^{-1} are both automorphisms, their composition $\phi \circ \phi^{-1}$ is the identity function e_G, and $\phi^{-1} \circ \phi$ is also e_G, satisfying the inverse property.

Therefore, $\text{Aut}(G)$ forms a group under composition.
Since, $Aut(G)$ forms a group under composition, we can see that it is isomorphic to the permutation group S_2. Thus, $Aut(G) \cong S_2$.
An automorphism is therefore nothing but a permutation of the vertices and edges of G such that it maps the edges to edges and non-edges to non-edges.

A graph is said to be **asymmetric** if its automorphism group consists only of the identity map.
In other words, a graph with no isomorphic graph is called asymmetric.

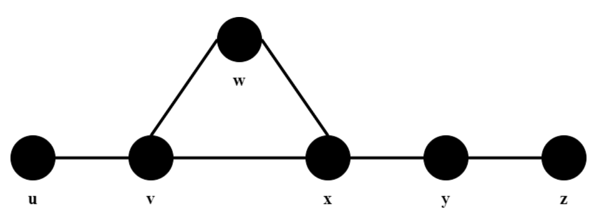


Figure 9: The smallest asymmetric graph

We can see that automorphism thereby describes the symmetry of the graph. In Figure 9, we see that there is no symmetry in the graph. This also helps us infer that it is asymmetric.

Consider the graph C_4.

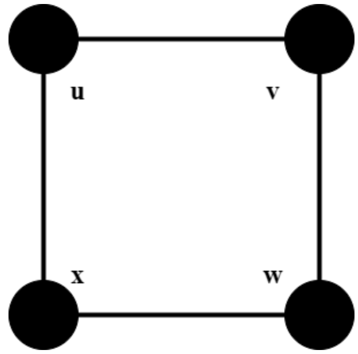


Figure 10: C_4

Let's construct the automorphism map for C_4:
Identity map: $e = \begin{pmatrix} u & v & w & x \\ u & v & w & x \end{pmatrix}$
Rotation by 90°: $\alpha_1 = \begin{pmatrix} u & v & w & x \\ v & w & x & u \end{pmatrix}$.
Rotation by 180°: $\alpha_2 = \begin{pmatrix} u & v & w & x \\ w & x & u & v \end{pmatrix}$.
Rotation by 270°: $\alpha_3 = \begin{pmatrix} u & v & w & x \\ x & u & v & w \end{pmatrix}$.
Reflection along the diagonal through vx: $\alpha_4 = \begin{pmatrix} u & v & w & x \\ w & v & x & u \end{pmatrix}$.
Reflection along the diagonal through uw: $\alpha_5 = \begin{pmatrix} u & v & w & x \\ u & x & w & v \end{pmatrix}$.
Reflection along the $x - axis$: $\alpha_6 = \begin{pmatrix} u & v & w & x \\ x & w & v & u \end{pmatrix}$.
Reflection along the $y - axis$: $\alpha_7 = \begin{pmatrix} u & v & w & x \\ v & u & x & w \end{pmatrix}$.
Therefore, $Aut(G) = \{e, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$.
It is isomorphic to the Dihedral group of order 8 (D_4).
Also, $\{e, \alpha_1, \alpha_2, \alpha_3\}$ forms a subgroup.
Thus, all automorphisms of any graph can be described by identity, reflection, or rotation preserving symmetry.

4.1 Properties of Graph Automorphism

Let's discuss some of the properties of graph automorphism based on general graphs.

Proposition 4.1 *For any graph having n vertices, the complete graph (K_n) among them has the highest number of automorphism.*

Proof. For a complete graph, the vertices are equal as all of them have the same degree. Thus we can exchange any of the vertices with one another. Thus in a complete graph, $n!$ permutations of the vertices are possible. In other words, a complete graph is the most symmetric graph for any number of vertices.
Then for any complete graph with n vertices, K_n, $Aut(K_n) \cong S_n$. □

Proposition 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 Proposition 4.1 to be S_{n-2}. It follows that the automorphism group of G is isomorphic to $S_2 \times S_{n-2}$. □

Proposition 4.3 *For the complete bipartite graph,$K_{n,m}$, where $n \geq m$: If $n > m$, then $Aut(K_{n,m}) \cong S_n \times S_m$.*

Proof. We know, $Aut(K_n) \cong S_n$. By proposition 4.1, it follows that $Aut(K_n) \cong S_n$. Thus, any automorphism of the form (x_i, x_j) or of the form (y_k, y_l) is in $Aut(K_{n,m})$. Thus, $S_n \times S_m$ is a subgroup of $Aut(K_{n,m})$. Suppose that $n > m$. Since $deg(x_i) = m$ and $deg(y_j) = n$, there is no automorphism ϕ such that $\phi(x_i) = y_j$. Thus, $Aut(K_{n,m}) \cong S_n \times S_m$. □

Proposition 4.4 ***Degree Preserving:*** *For all $u \in V_G$ and for all $\phi \in Aut(G)$, $deg(u) = deg(\phi(u))$.*

Proof. Let $u \in V_G$ with neighbors $u_1, ..., u_k$. Let $\phi \in Aut(G)$. Since ϕ preserves adjacency, it follows that $\phi(u_1), ..., \phi(u_k)$ are neighbors of $\phi(u)$. Therefore, $deg(\phi(u)) \geq k$. If $v \notin \{u_1, ..., u_k\}$ is a neighbor of $\phi(u)$, then $\phi^{-1}(v)$ is a neighbor of u. Therefore, the neighbors of $\phi(u)$ are precisely $\phi(u_1), ..., \phi(u_k)$. Thus, $deg(u) = deg(\phi(u))$. □

Proposition 4.5 ***Distance Preserving:*** *For all $u, v \in V_G$ and for all $\phi \in Aut(G)$, $d(u, v) = d(\phi(u), \phi(v))$.*

Proof. Let $u, v \in V_G$ and let $\phi \in Aut(G)$. Suppose that the distance from u to v is $d(u, v) = d$. Further, let $u = u_0, u_1, ..., u_{d-1}, u_d = v$ be a shortest path from u to v. Since ϕ

preserves adjacency, $\phi(u) = \phi(u_0), \phi(u_1), ..., \phi(u_{d-1}), \phi(u_d) = \phi(v)$ is a path from $\phi(u)$ to $\phi(v)$. Thus, $d(\phi(u), \phi(v)) \leq d = d(u, v)$. Suppose that $\phi(u), v_1, ..., v_{m-1}, \phi(v)$ is a shortest path from $\phi(u)$ to $\phi(v)$. It follows that $u, \phi^{-1}(v_1), ..., \phi^{-1}(v_{m-1}), v$ is a shortest path form u to v. It follows that $d(u, v) \leq d(\phi(u), \phi(v))$. Hence, we have equality. □

Proposition 4.6 *The automorphism group of G is equal to the automorphism group of the complement G.*

Proof. Note that automorphisms preserve not only adjacency, but non-adjacency as well. Hence, $\phi \in Aut(G)$ if and only if $\phi \in Aut(G')$. It follows that $Aut(G) \cong Aut(G')$. □

Proposition 4.7 *Given a Graph G with n vertices which is not complete, $Aut(G)$ forms a subgroup of $Aut(K_n)$ where K_n is the complete graph with n vertices.*

Proof. Any graph G on n nodes can be treated as a subgraph of K_n. We also note that the automorphism group of K_n is S_n which is all the permutations allowed on n nodes. So, a graph G on nodes would only contain a subset of these permutations in $Aut(G)$. Since , $Aut(G)$ is a group and it is also a subset of S_n, it is a subgroup of S_n. □

Proposition 4.8 *The order $|Aut(G)|$ of the automorphism group of a graph G of order n is a divisor of $n!$ and equals $n!$ iff $G = K_n$ or $G = K'_n$.*

Proof. By Lagrange's theorem, the order of a subgroup divides the order of a group. If $H \leq G$ then $\frac{|G|}{|H|} = [G : H]$.
If G is a graph of order n that is not complete, there is some pair of adjacent vertices and some pair of non-adjacent vertices.
Then there is some permutation of its vertex set that is not an automorphism.
Then, $Aut(G) \leq Aut(K_n) \cong S_n$.
We know, $|S_n| = n!$. Then, $Aut(K_n) = n!$, then $|Aut(G)||n!$.
Again, for K'_n is a graph with n isolated vertices. Thus, there can be $n!$ permutations possible of the vertices. Also from proposition 4.6., $Aut(K_n) \cong Aut(K'_n)$ which is again isomorphic to S_n.
This completes the proof. □

Frucht's Theorem Given a finite group τ, there exists a graph G such that $Aut(G)$ is isomorphic to τ.

Suppose that we are given a finite group τ. Our goal is to find a graph G such that $Aut(G) \cong \tau$. The result was proven by Frucht in 1939. The proof of Frucht's Theorem involves use of the Cayley graph(introduced in 1878).

Some application of Frucht's Theorem

Frucht's Theorem provides a connection between finite groups and graphs. It allows for the representation of finite groups as automorphism groups of certain graphs. This

connection has applications in group theory, particularly in the study of group representations and character theory.
Frucht's Theorem contributes to the development of algorithms for graph isomorphism testing. By constructing graphs with prescribed automorphism groups, it provides test cases for evaluating the performance and correctness of graph isomorphism algorithms. Understanding the automorphism structure of graphs can lead to improvements in isomorphism testing algorithms.

5 Conclusion

In abstract algebra, the concepts of graph isomorphism, equivalence classes, and automorphisms play crucial roles in analyzing structural similarities and symmetries within graphs. Graph isomorphism enables comparisons and classifications of graphs, while equivalence classes categorize graphs based on their symmetries. Automorphisms capture symmetrical transformations of graphs, providing insights into group theory and permutation groups. These concepts find applications in diverse fields, including network analysis, cryptography, and computational biology, where understanding graph structures and symmetries is essential for solving real-world problems efficiently and effectively.

References

1. Narsigh Deo, 2007. Graph Theory with applications to Engineering and Computer Science.

2. "Graph Theory: Structure and Representation" [Online].
   `http://www.cs.columbia.edu/~cs4203/files/GT-Lec2.pdf`
   Access Date: 01-04-2024

3. "A STUDY ON ISOMORPHISM OF ALGEBRAIC GRAPHS" [Online].
   `http://www.imvibl.org/buletin/bulletin_imvi_8_3_2018/bulletin_imvi_8_3_2018_523_532.pdf`
   Access Date: 01-04-2024

4. "Graph Isomorphisms" [Online]. `https://math.libretexts.org/Bookshelves/Combinatorics_and_Discrete_Mathematics/Combinatorics_(Morris)/03%3A_Graph_Theory/11%3A_Basics_of_Graph_Theory/11.04%3A_Graph_Isomorphisms`
   Access Date: 01-04-2024

5. "AUTOMORPHISM GROUPS OF SIMPLE GRAPHS" [Online]. `https://www.whitman.edu/documents/Academics/Mathematics/2014/rodriglr.pdf`
   Access Date: 10-04-2024

6. "Automorphism Group of Graphs (Supplemental Material for Intro to Graph Theory)" [Online]. `http://faculty.etsu.edu/beelerr/automorph-supp.pdf`
   Access Date: 12-04-2024