Charles University in Prague
Faculty of Mathematics and Physics
Computer Science Institute

Peter Zeman

Automorphism Groups of Geometrically Represented Graphs

Supervisor

Mgr. Pavel Klavík

SVOČ 2014, Ústí nad Labem

Contents

Abstract

Many graphs arising from various applications have nontrivial groups of automorphisms. This gives some importance to the study of the automorphism groups of graphs. It has also motivations in the complexity theory. For example, the famous graph isomorphism problem has a polynomial-time reduction to the problem of finding a generating set of the automorphism group some graph.

A famous result, known as Frucht's theorem, says that every finite group is isomorphic to the automorphism group of some finite graph. We are interested in the automorphism groups of graphs with a strong structure. Probably the first nontrivial class of graphs of which the automorphism groups were studied are finite trees. In 1869 Jordan gave a characterization of the class $\mathcal{T}$ of finite groups that are isomorphic to the automorphism group of some finite tree.

Intersection-defined classes of graphs often arise in various applications. Surprisingly, the automorphism groups of intersection graphs were studied only very briefly. We study the problem of reconstructing the automorphism group of a geometric intersection graph (a graph defined by intersections of geometric objects) from a good knowledge of the structure of its representations.

In this thesis, we deal in particular with interval graphs, intersection graphs of intervals on a real line. Our main result is that the class $\mathcal{I}$ of finite groups that are isomorphic to the automorphism group of some finite interval graph is the same as the class $\mathcal{T}$. We give a characterization of $\mathcal{I}$ in terms of group products and show that it is the same as Jordan's characterization of $\mathcal{T}$. We also show how to for a finite interval graph find a tree such that their automorphism groups are isomorphic.

1 Introduction

An automorphism of a graph X is a permutation of its vertices such that two vertices and are connected with an edge if and only if their images are connected with an edge. The group $\mathrm{Aut}(X)$ of all such permutations is called the automorphism group of X. A graph X *represents* a group G if $\mathrm{Aut}(X)$ is isomorphic to G.

Most graphs are *asymmetric*, that is, they have *no other* automorphisms that the identity automorphism (see e.g. [16]). However, many graphs arising from various algebraic, topological and combinatorial applications have nontrivial automorphism group, what gives some importance to the study of the automorphism groups of graphs.

Complexity Theory Motivation. The study of the automorphism groups of graphs is also motivated by problems in computational complexity theory. A long-standing open problem in the complexity theory is whether there exists an algorithm that can test isomorphism of finite algebraic structures in polynomial time. All such algebraic structures can be encoded by graphs in polynomial time [21, 33]. Therefore, it suffices to solve the isomorphism problem for graphs.

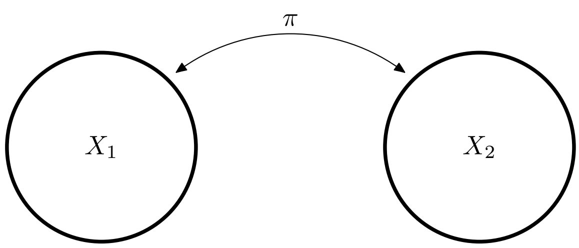
Problem:	$\textsc{GraphIso}(X_1, X_2)$
Input:	A graph X_1 and a graph X_2.
Question:	Is X_1 isomorphic to X_2?

Problem $\textsc{GraphIso}(X_1, X_2)$ is very important in the complexity theory. It is one of the few computational problems that are know to belong to NP, but are not known whether they are solvable in polynomial time and are also not know to be NP-complete. For many special classes of graphs, such as trees, planar graphs, interval graphs, $\textsc{GraphIso}(X_1, X_2)$ is know to be solvable in polynomial time. At the same time, there exists a strong theoretical evidence against NP-completeness of $\textsc{GraphIso}(X_1, X_2)$. It is known that it belongs to the low hierarchy of the class NP [36], which implies that it is not NP-complete unless the polynomial-time hierarchy collapses to its second level. For basic concepts in the complexity theory we refer to [1]. One of the most famous results concerning $\textsc{GraphIso}(X)$ is that it can be solved in polynomial time for graphs of bounded degree [28].

The graph isomorphism problem is closely related to a fundamental computational problem in algebraic graph theory, the problem of finding a generating set of the automorphism group of a graph.

Problem:	GraphAut(X)
Input:	A graph X.
Output:	Generating permutations for $\mathrm{Aut}(X)$.

Problem GraphIso(X_1, X_2) has a polynomial time reduction to GraphAut(X). Suppose that we are given two connected graphs X_1 and X_2. We set X to be the disjoint union of X_1 and X_2 and find the generating set of $\mathrm{Aut}(X)$. If the generating set contains a permutation π that swaps X_1 and X_2, then X_1 and X_2 are isomorphic. If X_1 and X_2 are disconnected, then we set X to be the disjoint union of their complements, since the automorphism group of a graph is isomorphic the automorphism group of its complement.



On the other hand, GraphAut(X) can be solved by solving GraphIso(X_1, X_2) at most $\mathcal{O}(n^4)$ times [30].

1.1 Graphs With a Strong Structure

A famous result, known as Frucht's theorem [13], says that every finite group is isomorphic to the automorphism group of some finite graph. We are interested in automorphism groups of classes of graphs with a very strong structure.

Probably the first nontrivial result in this direction is from 1869 due to Jordan [23]. He gave a characterization (see Theorem 2.6) of the automorphism groups of the class $\mathcal{T}$ of finite groups that are isomorphic to the automorphism group of some finite tree. It says that we can get the automorphism groups of trees from the trivial group by a sequance of two operations: the direct product and the wreath product with the symmetric group. The direct product constructs automorphisms that act independently on non-isomorphic components, while the wreath product constructs automorphisms that permute isomorphic components.

Another class of graphs of which the automorphism groups were studied are planar graphs. In 1973 Babai [2] gave a characterization of the automorphism groups of planar graphs.

Geometric Intersection Graphs. We can assign geometric objects to the vertices of graphs and encode its edges by intersections of these objects. More formally, an intersection representation $\mathcal{R}$ of X is a collection of sets $\{R_x \colon x \in V(X)\}$ such that $R_x \cap R_y \neq \emptyset$ if and only if $uv \in E(X)$. Every graph can be represented in this way [29]. Therefore, to obtain reasonable classes of graphs, the sets R_x are usually some very specific geometric objects. The most famous classes of geometric intersection graphs include interval graphs, circle graphs, circular-arc graphs, permutation graphs and function graphs.

The problem of characterizing the intersection graphs of families of sets having some geometrical property is an interesting problem and is often motivated by real world applications. Sometimes even application gives an intersection representation. Many hard combinatorial problems can be often solved efficiently on geometric intersection graphs. Another reason for considering an intersection representation of a graph is that it can provide much better visualisation of the graph and therefore, possibly a much better understanding of the structure of the graph. For example, the structure of the graph from Figure 1.1 is much more clear from its interval representation. For more information about intersection graph theory see for example [37, 32, 17].

Surprisingly, automorphism groups of intersection-defined classes of graphs were studied only briefly. Even for very deeply studied classes of intersection graphs the structure of their automorphism groups is not known very well. For a given intersection-defined class, the mostly studied are classical graph-theoretic properties (the chromatic number, forbidden graph characterization, and so on) or the complexity of the recognition problem.

We study the problem of reconstructing the automorphism group of geometric intersection graph from a good knowledge of the structure of its representations. In this thesis, we deal mainly with interval graphs.

Interval Graphs. Interval graphs are intersection graphs of intervals on a real line. They are one of the oldest and most studied class of graphs, first introduced by Hajós [19] in 1957.

An *interval representation* $\mathcal{R}$ of a graph X is a set of closed intervals $\{I_x : x \in V(X)\}$ such that $xy \in E(X)$ if and only if $I_x \cap I_y \neq \emptyset$. In other words, an edge of X is represented by an intersection of intervals. A graph X is an interval graph if there exists an interval representation $\mathcal{R}$ of X. Figure 1.1 shows an example of an interval graph and its interval representation.

One of the reasons why interval graphs were studied quite extensively is that they have real world applications. One application is in biology. Benzer [3] showed a direct relation between interval graphs and the arrangement of genes in the chromosome. Mutations correspond to a damaged segment on a chromosome. Each mutation can

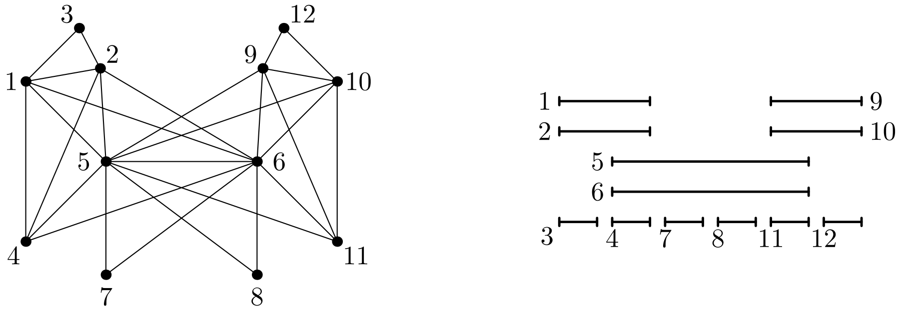


Figure 1.1: Interval graph and its interval representation.

damage a different set of genes. At that time, the only information that could be gathered was the set of deformities caused by a mutation. We can form a graph by making each mutation into a vertex and adding an edge between two vertices if the mutations share a common deformity. Benzer found that a graph formed in this way is an interval graph and this was considered a strong evidence supporting his theory that genes are arranged in a simple linear fashion. Interval graphs have also many other applications (see for example [34, 38]).

Interval graphs have also many useful theoretical properties and nice mathematical characterizations. In many cases, very hard computational problems are polynomially solvable for interval graphs. This problems include graph isomorphism, maximum clique, k-coloring, maximum independent set, and so on.

1.2 Results of This Thesis

In this thesis, we study the automorphism groups of interval graphs. The structure of their representations is already very well understood due to Booth and Lueker [4].

In 1981 Cobourn and Booth [8] designed a linear-time algorithm that for an interval graph finds the generating automorphisms of its automorphism group. Our result gives an explicit description of the automorphism group of an arbitrary interval graph in terms of group products, so also from the algorithmic point of view we get better information about the group. Moreover, our description of the automorphism groups of interval graphs is much more detailed and shows the relation between the structure of all representations of an interval graph and its automorphism group.

Let $\mathcal{I}$ be the class of finite groups that are isomorphic to the automorphism group of some interval graph and let $\mathcal{T}$ be the class of finite groups that are isomorphic to the automorphism groups of some finite tree. Our main result is the following theorem.

Thoerem. *The class $\mathcal{I}$ is the same as the class $\mathcal{T}$, that is, for each interval graph X there exists a tree T such that* $\mathrm{Aut}(X)$ *is isomorphic to* $\mathrm{Aut}(T)$ *and vice versa.*

This is surprising because the class INT of finite interval graphs and the class TREE of finite trees are two very different classes. The intersection INT $\cap$ TREE are exactly the graphs called CATERPILLARS. Those are trees in which all vertices are within distance one of central path. Another two important classes of graphs that are related to INT are the classes AT-FREE and CHOR. The first one is the class of *asteroidal triple-free* graphs. Three vertices of a graph form an *asteroidal triple* if every two of them are connected by a path avoiding the neighbourhood of the third. A graph is asteroidal triple-free if it does not contain any asteroidal triple. The class CHOR is the class of *chordal graphs.* Chordal graph is a graph that *does not*

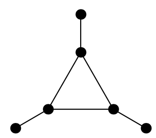


Figure 1.2: A graph that is not a tree and contains an asteroidal triple.

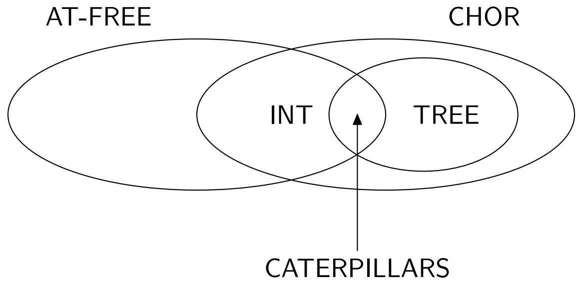


Figure 1.3: The relation of some important classes of graphs.

contain an induced cycle of length four or more. Another characterization of chordal graphs says that chordal graphs are intersection graphs of subtrees of a tree [15], which is a generalization of interval graphs (if the tree is a path, then we get interval graphs). It is well known that a graph is an interval graph if and only if it is in AT-FREE $\cap$ CHOR [26]. Chordal graphs are also interesting on their own. The problem GraphIso(X_1, X_2) is polynomially reducible to testing isomorphism of chordal graphs [27]. This means that for an arbitrary graph there exists a chordal graph with the same group of automorphisms. So, chordal graphs are universal for automorphism groups and the structural study of their groups is finished.

The equality of $\mathcal{I}$ and $\mathcal{T}$ was already mentioned by Hanlon [20] in his paper about counting interval graphs. However, his paper lacks an explanation or a proof of this result. Moreover, to find for an interval graph X a tree T such that $\mathrm{Aut}(X)$ is isomorphic to $\mathrm{Aut}(T)$ states Hanlon as an open problem. We can solve this problem easily using our description of the class $\mathcal{I}$. This is a strong evidence that our understanding of the structure of $\mathcal{I}$ is much deeper. We are also able to find for a tree an interval graph with the same group of automorphisms.

Our characterization of the class $\mathcal{I}$ is based on the Jordan's characterization (see Theorem 2.6) of the class $\mathcal{T}$. We add a third operation, the semidirect product with $\mathbb{Z}_2$, which corresponds to a reflection symmetry of the interval representation of an interval graph. Then to prove the equality of $\mathcal{I}$ and $\mathcal{T}$, we show that this third operation can be replaced by a sequence of the first two opperations.

2 Preliminaries

In Section 2.1, we recall some concepts from group theory that are essential for the main result. For a comprehensive treatment of the basics of group theory see for example [35, 10] or for a more visual treatment of group theory see [5]. In Section 2.2 we give a definition of PQ-trees and *modified* PQ-trees an explain how they capture in some sense all possible representations of an interval graph.

2.1 Group Products

Here, we explain two basic group theoretic methods for constructing larger groups from smaller ones, namely *direct product* and *semidirect product*. We show how can these group operations used to construct automorphism groups of graphs. At the end of this section, we prove Jordan's characterization 2.6 of the class $\mathcal{T}$.

Inspired by [5], we use *Cayley graphs* to visualize groups. Cayley graphs were actually invented by Cayley [6] for visualizing groups and now they and now they play an important role in combinatorial and geometric group theory. Cayley graph is a colored oriented graph that encodes the abstract structure of a group. Suppose that G is a graph and S is a generating set. The Cayley graph (G, S) is a graph constructed as follows:

- Each element of G is assigned a vertex.
- Each generator $s \in S$ is assigned a unique color $c(x)$.
- For any $g \in G$ and $s \in S$, the there is a directed edge (g, gs) of color $c(s)$.

Figure 2.1 and Figure 2.2 show examples of graphs and Cayley graphs of their automorphism groups.

2.1.1 Direct Product

The direct product $G \times H$ of groups G and H with operations $\cdot_G$ and $\cdot_H$, respectively, is the set of pairs (g, h) where $g \in G$ and $h \in H$ with operation defined componentwise:

$$(g_1, h_1) \cdot (g_2, h_2) = (g_1 \cdot_G g_2, h_1 \cdot_H h_2).$$

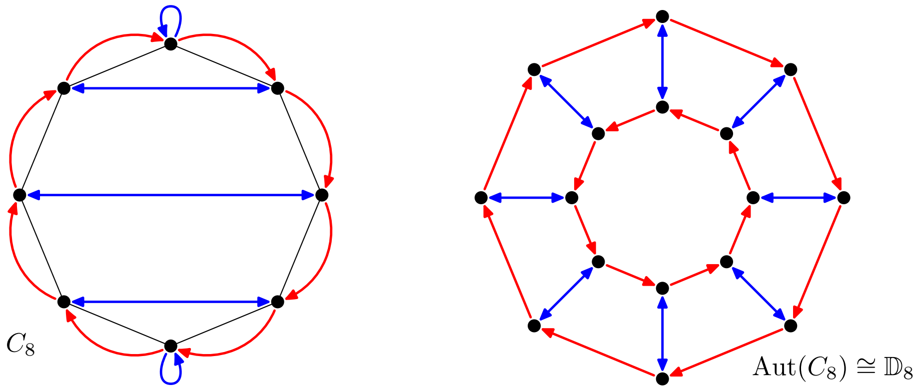


Figure 2.1: The graph C_8 with the action of $\mathrm{Aut}(C_8)$ on its vertices and Cayley graph of $\mathrm{Aut}(C_8)$. Note that $\mathrm{Aut}(C_8)$ is isomorphic to $\mathbb{D}_8$. It is generated by two automorphisms: the rotation symmetry (corresponds to the red arrow); the reflection symmetry.

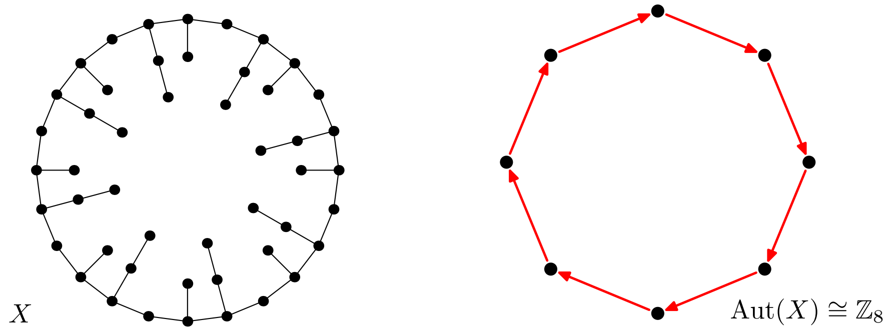


Figure 2.2: A graph with the automorphism group isomorphic to the group $\mathbb{Z}_8$ and Cayley graph of $\mathbb{Z}_8$. A graph like this one has only the rotation symmetries as automorphisms, therefore, its automorphism group is isomorphic to a subgroup of $\mathrm{Aut}(C_8)$.

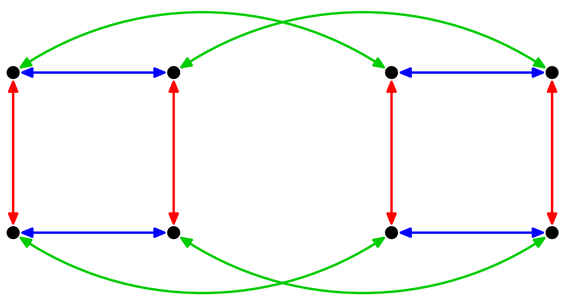
When there is no confusion we simply write $(g_1 \cdot g_2, h_1 \cdot h_2)$ or $(g_1 g_2, h_1 h_2)$. The direct product of n groups is defined analogously.

Suppose that we have the direct product $G_1 \times \cdots \times G_n$ of groups $G_1, \ldots, G_n$. We can define a homomorphism $\pi\colon G_1 \times G_2 \times \cdots \times G_n \to G_2, \times \cdots \times G_n$ by

$$\pi((g_1, g_2, \ldots, g_n)) = (g_2, \ldots, g_n).$$

The kernel $\mathrm{Ker}(\pi)$ is clearly isomorphic to G_1. Therefore, G_1 is a normal subgroup of $G_1 \times \cdots \times G_n$. Analogously, each G_i is a normal subgroup of $G_1 \times \cdots \times G_n$. On the other hand, semidirect product, discussed in Section 2.1.2, takes two groups G and H and constructs a larger group such that only G is a normal subgroup.

Example 2.1. This figure shows Cayley graph of the group $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$. Note that the group contains two copies of $\mathbb{Z}_2 \times \mathbb{Z}_2$, with the corresponding elements connected according to the pattern of $\mathbb{Z}_2$.



Direct product can be used to construct automorphism groups of graphs that are disconnected and their connected components are non-isomorphic. In this case, the automorphism group of the whole graph is the direct product of the automorphism groups of its connected components, because each automorphism of the graph acts independently on each connected component.

2.1.2 Semidirect Product

However, if we want to construct the automorphism group of a disconnected graph which has some isomorphic connected components, direct product *is not* sufficient because automorphism that permute the isomorphic components are not included in the direct product.

Example 2.2. The automorphism group of the graph X is isomorphic $\mathbb{S}_3 \times \mathbb{Z}_2$, but the automorphism group of the graph Y *is not* $\mathbb{Z}_2 \times \mathbb{Z}_2$ since the direct product does not include the automorphism which swaps the components. The automorphism group of Y is not even $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$ because, for example, swapping the components and swapping the vertices of the left component do not commute.

If we construct a larger group from some groups G and H using direct product, then both G and H are normal subgroups of the resulting group. The motivation for semidirect product is to construct a group from the groups G and H such that G does not have to be its normal subgroup.

The direct product $G \times H$ contains identical copies of G, with corresponding elements connected according to the pattern of H, as shown in Example 2.1. In the semidirect product of the groups G and H, the group H also determines a pattern according to which some copies of G are connected, however, those copies of G do not need to be all identical.

First, we explain a special case of the semidirect product, the semidirect product of the group G with its automorphism group $\mathrm{Aut}(G)$, denoted by

$$G \rtimes \mathrm{Aut}(G).$$

We define it to be the set of all pairs (g, f) such that $g \in G$ and $f \in \mathrm{Aut}(G)$, with the operation defined in the following way:

$$(g_1, f_1) \cdot (g_2, f_2) = (g_1 \cdot f_1(g_2), f_1 \cdot f_2).$$

Note that $G \rtimes \mathrm{Aut}(G)$ with the operation defined like this forms a group. It is straightforward to see that the identity element is $(1, 1)$ and that the inverse of the element (g, f) is the element $(f^{-1}(g^{-1}), f^{-1})$.

We can think of it as all possible isomorphic copies of G connected according to the pattern of $\mathrm{Aut}(G)$. The element (g_1, f_1) is in the isomorphic copy G_1 of G which we get by applying the automorphism f_1 on G. Multiplying (g_1, f_1) by $(g_2, 1)$ corresponds to a movement inside G_1. Multiplying (g_1, f_1) by $(1, f_2)$ corresponds to a movement from G_1 to another isomorphic copy of G.

In general, there exists semidirect product group for any two groups G and H, and a homomorphism $\varphi \colon H \to \mathrm{Aut}(G)$, denoted by

$$G \rtimes_\varphi H.$$

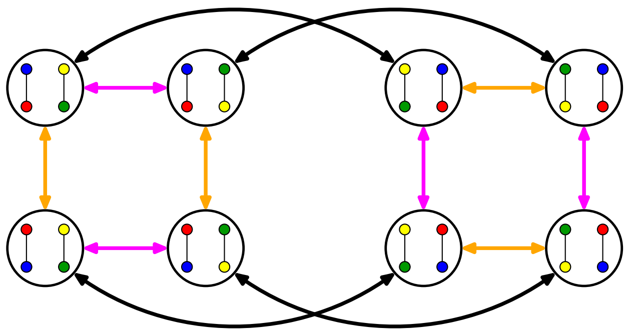
It is the set of all pairs (g, h) such that $g \in G$ and $h \in H$. The operation is defined similarly to the operation defined on $G \rtimes \mathrm{Aut}(G)$:

$$(g_1, h_1) \cdot (g_2, h_2) = (g_1 \cdot \varphi(h_1)(g_2), h_1 \cdot h_2).$$

Again, it is quite straightforward to check that $G \rtimes_\varphi H$ is a group. We can think of the homomorphism φ as if it assigns an isomorphic copy of G to each element of the group H. The isomorphic copies of G are then connected according to the pattern of the group H. We write $G \rtimes H$ when there is no danger of confusion.

Example 2.3. Dihedral group $\mathbb{D}_8$ is equal to $\mathbb{Z}_8 \rtimes \mathbb{Z}_2$. Figure 2.1 shows Cayley graph of $\mathbb{D}_8$ (on the right). The elements of the two isomorphic copies of $\mathbb{Z}_8$ are connected according to the pattern of $\mathbb{Z}_2$.

Example 2.4. Let Y be the graph from Example 2.2. The group $\mathrm{Aut}(Y)$ is isomorphic to $(\mathbb{Z}_2 \times \mathbb{Z}_2) \rtimes \mathbb{Z}_2$. The figure shows Cayley graph of $\mathrm{Aut}(Y)$. Again, the elements of the two isomorphic copies of $\mathbb{Z}_2 \times \mathbb{Z}_2$ are connected according to the pattern of $\mathbb{Z}_2$.



The generators of $\mathrm{Aut}(Y)$ are the following: the permutation that swaps the left component and fixes the right component (orange); the permutation that swaps the right component and fixes the left component (purple); the permutation that swaps the components (black). The subgroup of $\mathrm{Aut}(Y)$ which acts on the components independently and does not swap them, corresponds to the isomorphic copy of $\mathbb{Z}_2 \times \mathbb{Z}_2$ which is on the left in the Cayley graph. Swapping the components with the black automorphism changes the orange automorphism to the purple automorphism and vice versa. In other words, swapping the vertices of some component does not commute with swapping the components.

Wreath Product. Let K and L be groups, let $\rho\colon K \to \mathbb{S}_n$ be a homomorphism, let H be the direct product of n copies of L and let ψ be an injective homomorphism from $\mathbb{S}_n$ into $\mathrm{Aut}(H)$. The composition $\rho \circ \psi$ is a homomorphism from K into $\mathrm{Aut}(H)$. The *wreath product* is the semidirect product $H \rtimes K$ with respect homomorphism $\rho \circ \psi$ and is denoted by $L \wr K$. If K is $\mathbb{S}_n$, then the wreath product $L \wr \mathbb{S}_n$ is simply $L^n \rtimes_\varphi \mathbb{S}_n$ where $\varphi\colon \mathbb{S}_n \to \mathrm{Aut}(L^n)$ is s homomorphism defined by

$$\varphi(\pi) = \text{the automorphism that maps } (l_1, \dots, l_n) \text{ to } (l_{\pi(1)}, \dots, l_{\pi(n)}).$$

Theorem 2.5 shows how to construct the automorphism groups of a disconnected graph using group products. Theorem 2.6 gives the characterization of the class of the automorphism groups of trees in terms of group products.

Theorem 2.5 (Automorphism groups of disconnected graphs)**.** *If $X_1, \dots, X_n$ are pairwise non-isomorphic connected graphs and X is the disjoint union of k_i copies of X_i, $i = 1, \dots, n$, then*

$$\mathrm{Aut}(X) = \mathrm{Aut}(X_1) \wr \mathbb{S}_{k_1} \times \cdots \times \mathrm{Aut}(X_n) \wr \mathbb{S}_{k_n}.$$

Proof. First, we deal with a special case. Suppose that X consists *only* of k_1 isomorphic copies of X_1, we denote them by $X_1^1, \dots, X_1^{k_1}$. Each automorphism α of X acts independently on $X_1^1, \dots, X_1^{k_1}$ and permutes them arbitrarily. So, if $\alpha \in \mathrm{Aut}(X)$, then

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{k_1}, \pi)$$

where X_j is an automorphism of X_1^j and $\pi \in \mathbb{S}_{k_1}$. The operation on $\mathrm{Aut}(X)$ can be algebraically written as

$$\begin{aligned}\alpha \cdot \beta &= (\alpha_1, \alpha_2, \dots, \alpha_{k_1}, \pi) \cdot (\beta_1, \beta_2, \dots, \beta_{k_1}, \rho)\\ &= (\alpha_1 \cdot \beta_{\pi(1)}, \alpha_2 \cdot \beta_{\pi(2)}, \dots, \alpha_{k_1} \cdot \beta_{\pi(k_1)}, \pi \circ \rho).\end{aligned}$$

In other words, the action of $\alpha \cdot \beta$ on X can be described in the following way. First, α acts on each X_1^j using α_j and permutes $X_1^1, \ldots, X_1^{k_1}$ using π. Since after applying α on X we have components $X_1^{\pi(1)}, \ldots, X_1^{\pi(k_1)}$, we have to make each β_j act on the correct component. We do this by letting $\beta_{\pi(j)}$ act on $X_1^{\pi(j)}$. It follows that

$$\mathrm{Aut}(X) \cong \mathrm{Aut}(X_1)^{k_1} \rtimes_\varphi \mathbb{S}_{k_1} = \mathrm{Aut}(X_1) \wr \mathbb{S}_{k_1}$$

where $\varphi\colon \mathbb{S}_{k_1} \to \mathrm{Aut}(\mathrm{Aut}(X_1)^n)$ is a homomorphism defined naturally by

$$\varphi(\pi) = \text{the automorphism that maps } (\alpha_1, \alpha_2, \ldots, \alpha_{k_1}) \text{ to } (\alpha_{\pi(1)}, \alpha_{\pi(2)}, \ldots, \alpha_{\pi(k_1)}).$$

Now we consider the general case. No automorphism of X can swap some X_i and X_j because they are non-isomorphic. Therefore, each automorphism acts independently on the isomorphic copies of each X_i, so to get $\mathrm{Aut}(X)$ we only need to take the direct product. □

Theorem 2.6 (Jordan, 1869). *A finite group G is isomorphic to an automorphism group of a finite tree tree if and only if $G \in \mathcal{T}$, where the class $\mathcal{T}$ of finite groups is defined inductively as follows:*

(a) $\{1\} \in \mathcal{T}$.

(b) *If $G_1, G_2 \in \mathcal{T}$, then $G_1 \times G_2 \in \mathcal{T}$.*

(c) *If $G \in \mathcal{T}$ and $n \geq 2$, then $G \wr \mathbb{S}_n \in \mathcal{T}$.*

Proof. For each $G \in \mathcal{T}$ we construct a tree T such that $\mathrm{Aut}(T) \cong G$.

- If $G_1, G_2 \in \mathcal{T}$ and T_1, T_2 are rooted trees such that $G_1 \cong \mathrm{Aut}(T_1)$ and $G_2 \cong \mathrm{Aut}(T_2)$, the we construct the by connecting the root of T_1 with one end of an *asymmetric path* (see Figure 3.5) and T_2 with the other. Clearly, $\mathrm{Aut}(T) \cong G_1 \times G_2$.

- If $G \in \mathcal{T}$ and T' is a rooted tree such that $G \cong \mathrm{Aut}(T')$, then we construct T be taking the disjoint union of n copies of T'. By Theorem 2.6 $G \wr \mathbb{S}_n \cong \mathrm{Aut}(T)$.

Now, it remains to prove that for each rooted tree T, the group $\mathrm{Aut}(T)$ is in the class $\mathcal{T}$. Every tree has a center, a vertex or an edge, which is fixed under each automorphism. Therefore, deleting the root does no change the automorphism group of the tree. So the problem of determining automorphism groups of trees can be reduced to rooted trees.

If T is a rooted tree containing only one vertex, then clearly $\mathrm{Aut}(T) \in \mathcal{T}$. Otherwise, we delete the root and get a forest of rooted trees $T_1, \ldots, T_n$. We determine the automorphism group of each T_i recursively and use Theorem 2.5 to construct the group $\mathrm{Aut}(T)$. It follows that $\mathrm{Aut}(T) \in \mathcal{T}$. □

2.2 Tree Representations of Interval Graphs

2.2.1 PQ-trees

PQ-trees were invented for the purpose of solving the *consecutive ordering problem*. For a set S and restricting sets $R_1, \dots, R_k$, the task is to find a linear ordering of S such that every R_i appears consecutively in it as one block.

Example 2.7. Consider the set $S = \{a, b, c, d, e\}$ and the restricting sets $R_1 = \{a, b\}$, $R_2 = \{c, d, e\}$ and $R_3 = \{b, c\}$. The orderings *abcde*, *abced*, *decba* and *edcba* are the only feasible orderings of U, any other ordering violates some restriction. For instance, the ordering $a\underline{b}d\underline{c}e$ violates R_3.

A PQ-tree is a rooted tree designed for solving the consecutive ordering problem efficiently and in addition to that, for a given input, they store *all* feasible orderings of the set S.

The leaves of the tree correspond one-to-one to the elements of S. The inner nodes are of two types: The *P-nodes* and the *Q-nodes*. We assume that each P-node has at least two children and that each Q-node has at least three children. For every inner node, the order of its children is fixed.

The *frontier* of a PQ-tree T is a permutation of the set S obtained by ordering the leaves of T from left to right. The frontier of T represents one ordering of S.

To obtain all feasible orderings of S we can modify T by applying a finite sequence of the following two *equivalence transformations*:

- Arbitrarily permute the children of a P-node.
- Reverse the children of a Q-node.

We denote the PQ-tree which we get from T by applying a finite sequence of equivalence transformations ε by T_ε. A PQ-tree T' is *equivalent* with T if one can be obtained from the other using a finite sequence of equivalence transformations.

PQ-trees were invented by Booth and Lueker [4] and for the purposes of this thesis it is sufficient to know that a PQ-tree can be constructed in linear time with respect to the number of elements of S, number of restricting sets and total size of restricting sets. Figure 2.3 shows PQ-trees that represent all feasible orderings of the set S from Example 2.7, P-nodes are denoted by circles and Q-nodes by rectangles.

PQ-trees and Interval Graphs. The following characterization of interval graphs, given by Fulkerson and Gross [14], shows the relation between interval graphs and the consecutive ordering problem. We denote the set of all maximal cliques of X by $\mathcal{C}(X)$.

Lemma 2.8 (Fulkerson and Gross)**.** *A graph X is an interval graph if and only if there exists an ordering of the maximal cliques $\mathcal{C}(X)$ such that for every vertex $x \in V(X)$, the maximal cliques containing x appear consecutively in it.*

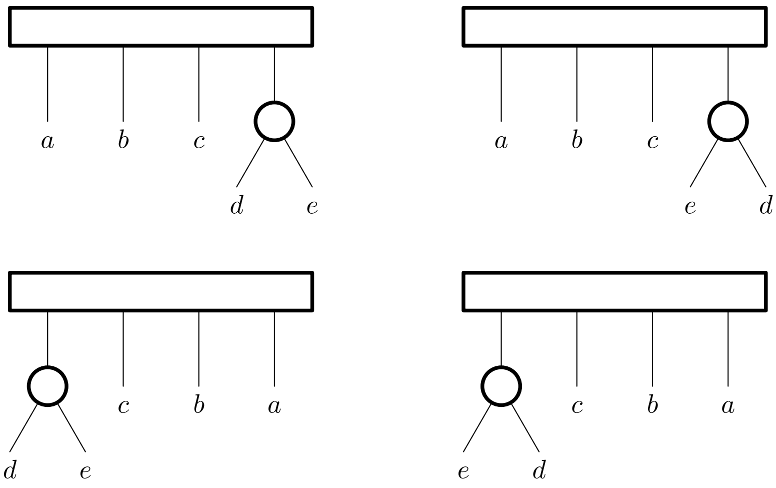


Figure 2.3: Four PQ-trees that represent all feasible orderings of the set S from Example 2.7, the circles are P-nodes and the rectangles are Q-nodes.

Proof. Let $\{I_x \colon x \in X\}$ be an interval representation of X and let $C_1, \dots, C_k$ be the maximal cliques. By Helly's theorem, the intersection $\bigcap_{x \in C_i} I_x$ is non-empty, and therefore there exist a point c_i in it. The ordering of $c_1, \dots, c_k$ from left to right gives the required ordering.

Given an ordering of the maximal cliques $C_1, \dots, C_k$, we place points $c_1, \dots, c_k$ in this ordering on a real line. To each vertex v, we assign minimal interval I_x such that $c_i \in I_x$ if and only if $x \in C_i$. We obtain a valid interval representation $\{I_x \colon x \in V(X)\}$ of X. □

We define a relation $\sim_{TW}$ on the vertices of an interval graph X where $x \sim_{TW} y$ means that x and y belong to precisely the same maximal cliques, and fix an ordering $<_{TW}$ on each equivalence class of $\sim_{TW}$. If two vertices x and y are in $\sim_{TW}$ we say that they are *twin* vertices. In other words, twin vertices are indistinguishable from the point of view of the intersection representation and not so interesting, however, they need to be considered when studying automorphism groups.

Recognition of interval graphs in linear time was an open problem, first solved by Booth and Lueker [4] using PQ-trees. By Lemma 2.8, the problem of recognizing interval graphs can be simply reduced to the consecutive ordering problem. To test whether a graph X is an interval graph we set S to be the set of all maximal cliques $\mathcal{C}(X)$. For each vertex x we define a restricting set $R_x = \{C \in \mathcal{C}(X) : x \in C\}$. Lemma 2.8 says that X is an interval graph if and only if there exist a linear ordering of S such that every R_x appears consecutively in it. The algorithm for solving the consecutive ordering problem constructs a PQ-tree T such that the frontier of T gives one possible consecutive ordering of $\mathcal{C}(X)$. We get all possible orderings of $\mathcal{C}(X)$ by applying sequences of equivalence transformations. Each *sequence of equivalence transformations encodes a permutation of* $\mathcal{C}(X)$. Figure 2.4 shows an example of an interval graphs and a PQ-tree representing it.

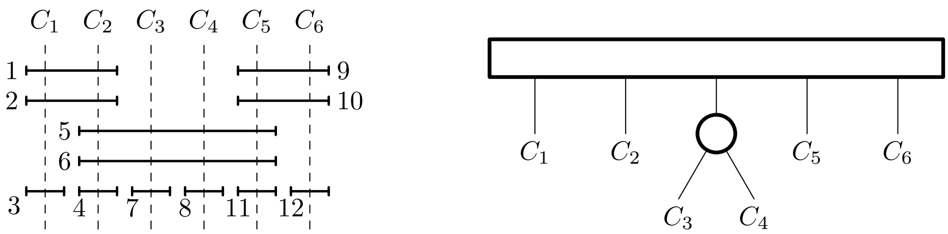


Figure 2.4: An interval graph and a PQ-tree which represents one consecutive ordering of its maximal cliques. We can get all other possible orderings by applying the equivalence transformations on the PQ-tree.

2.2.2 MPQ-trees

A modified PQ-tree (MPQ-tree) is basically a PQ-tree with some additional information about the twin vertices. MPQ-trees were first mentioned by Korte and Möhring [25], they used them to show a more simple linear time recognition algorithm for interval graphs than Booth and Lueker. MPQ-trees were also used by Coulborn and Booth [8] to design a linear time algorithm for computing s set of generator of the automorphism group of an interval graph, however, they mention them only implicitly.

Suppose that T is a PQ-tree representing an interval graph X. To obtain an MPQ-tree M from T we assign sets, called *sections*, to the nodes of T. Leafs and P-nodes are assigned only one section, while Q-nodes have a section for each of its children. We assign the sections to the nodes of T in the following way:

- For every leaf L, the section $\text{sec}(L)$ contains those vertices of X that are only in the maximal clique represented by L, and no other maximal cliques.
- For every P-node P, the section $\text{sec}(P)$ contains those vertices of X that are in all maximal cliques represented by the leaves of the subtree of P, and no other maximal cliques.
- For every Q-node Q and its children $Q_1, \dots, Q_n$, the section $\text{sec}_i(Q)$ contains those vertices of X that are in the maximal cliques represented by the leaves of the subtree of Q_i *and also* some other Q_j, but are not in any other maximal clique represented by a leaf that is not in the subtree of Q.

Figure 2.5 shows an example of an MPQ-tree.

Observation 2.9. *Vertices $x, y \in V(X)$ that are in the same sections of an MPQ-tree representing an interval graph X belong to the same maximal cliques of X, that is, $x \sim_{TW} y$.*

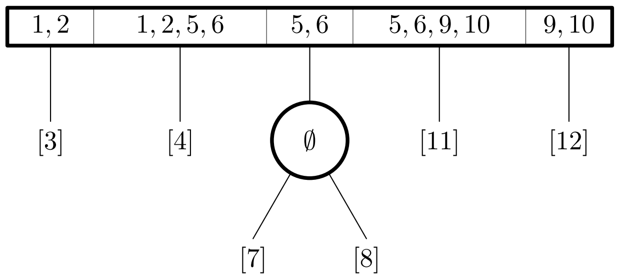


Figure 2.5: An MPQ-tree that represents the interval graph from Figure 2.4. Twin vertices belong to the same sections of the MPQ-tree, they are ordered from left to right. Note that some sections can be empty. The ordering $<_{TW}$ of the vertices that belong to the same sections is given from left to right.

3 Automorphism Groups of Interval Graphs

In this chapter, we derive a characterization of the class $\mathcal{I}$ of the automorphism groups of finite interval graphs and show that it is equal to the class $\mathcal{T}$ of the automorphism groups of finite trees.

3.1 Automorphisms Groups of PQ-trees

Here, we give a definition of an automorphism of a PQ-tree and an MPQ-tree that represent an interval graph X. We show that the automorphism group of the PQ-tree is isomorphic to a subgroup of $\mathrm{Aut}(X)$ and that the additional information in the MPQ-tree is sufficient for its automorphism group to be isomorphic to $\mathrm{Aut}(X)$.

Automorphism Groups of PQ-trees. Let T be a PQ-tree representing an interval graph X. We define each *symmetric* sequence of equivalence transformations to be an automorphism of T. More formally, a sequence of equivalence transformations $\varepsilon\colon \mathcal{C}(X) \to \mathcal{C}(X)$ is an *automorphism of* T if there exists a permutation $\alpha\colon V(X) \to V(X)$ of the vertices of X such that after replacing each leaf L in T_ε with $\alpha(L)$ we get T. We say that α *cancels* ε. Figure 3.1 shows an example of the automorphism of a PQ-tree.

Lemma 3.1. *Automorphisms of a PQ-tree T representing X form a group.*

Proof. Suppose that ε_1 and ε_2 are automorphisms of T and α_1 cancels ε_1, and α_2 cancels ε_2. The composition $\alpha_2 \circ \alpha_1$ cancels $\varepsilon_1 \cdot \varepsilon_2$, so $\varepsilon_1 \cdot \varepsilon_2$ is also an automorphism of the PQ-tree T. □

We denote the group of automorphisms of a PQ-tree T representing X by $\mathrm{Aut}(T)$. The following lemma shows that a permutation which cancels an automorphism of T is an automorphism of X.

Lemma 3.2. *If ε is an automorphism of a PQ-tree T representing X and α cancels ε, then α is an automorphism of X.*

Proof. Let $x, y \in V(X)$ be two vertices. Vertices x and y are adjacent if and only if they are contained in some maximal clique. The permutation α defines a permutation

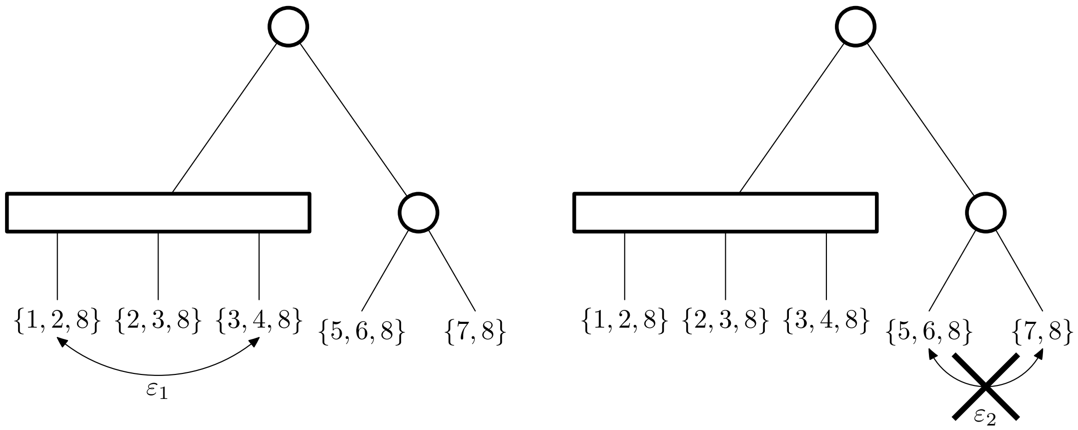


Figure 3.1: The equivalence transformation ε_1 on the left is the *only* automorphism of the PQ-tree. For example the transformation ε_2 on the right can not be an automorphism because there is no permutation α of the vertices such that $\alpha(\{7,8\}) = \{5,6,8\}$.

of the maximal cliques $\mathcal{C}(X)$, since it cancels ε. So, $\alpha(x)$ and $\alpha(y)$ are in the same maximal clique if and only if x and y are in the same maximal clique. □

An automorphism α of X reorders the maximal cliques $\mathcal{C}(X)$. Since all PQ-trees that are equivalent with a PQ-tree T representing X contain all feasible orderings of the maximal cliques, there exists an equivalence transformation ε of T which reorders $\mathcal{C}(X)$ in the same way as α. The next lemma shows that ε is an automorphism of T.

Lemma 3.3. *If $\alpha \in \mathrm{Aut}(X)$ and ε is an equivalence transformation of T such that ε reorders $\mathcal{C}(X)$ in the same way as α, then $\varepsilon \in \mathrm{Aut}(T)$.*

Proof. The automorphism α^{-1} cancels ε. □

There can be more automorphisms of X that reorder $\mathcal{C}(X)$ in the same way. If the equivalence relation $\sim_{TW}$ has an equivalence class of size greater that one, then some automorphism of X reorder $\mathcal{C}(X)$ in the same way, but permute the equivalence class differently. We define a mapping $\phi\colon \mathrm{Aut}(X) \to \mathrm{Aut}(T)$ by

$$\phi(\alpha) = \varepsilon$$

where ε is an equivalence transformation of T that gives the same reordering of $\mathcal{C}(X)$ as α. According to Lemma 3.2 and Lemma 3.3 ϕ is a well defined surjective mapping. It is straightforward to see that *ϕ is a homomorphism*. Moreover, ϕ is a quotient homomorphism, that is, it is possible that two automorphisms of X are mapped by ϕ to the same automorphism of T.

In general, the automorphism group of a PQ-tree T representing X is not isomorphic to the automorphism group of X. An automorphism $\alpha \in \mathrm{Aut}(X)$ is in $\mathrm{Ker}(\phi)$ if it only swaps vertices x, y that belong to the same maximal cliques, that is, $x \sim_{TW} y$. By the first isomorphism theorem

$$\mathrm{Aut}(T) \cong \frac{\mathrm{Aut}(G)}{\mathrm{Ker}(\phi)}.$$

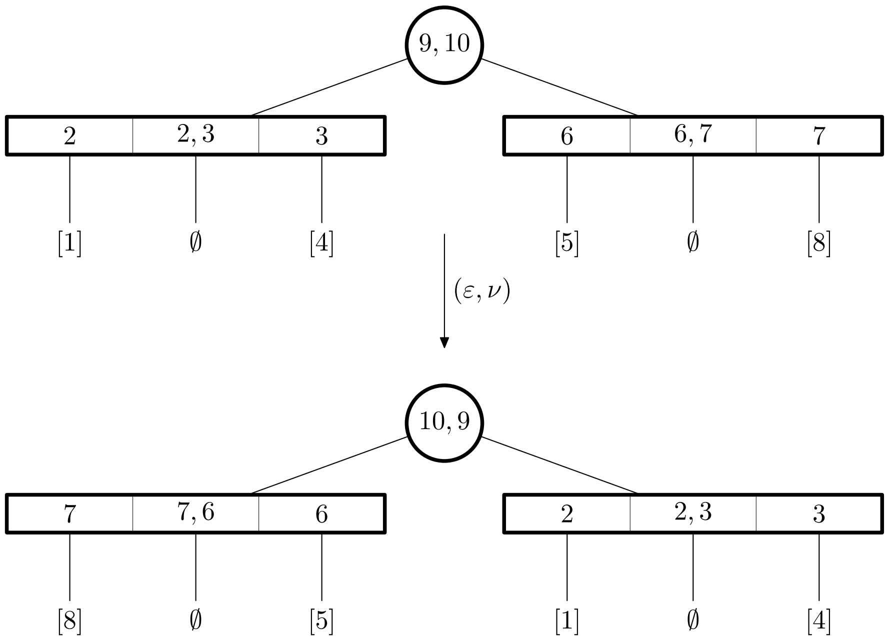


Figure 3.2: The pair (ε, ν) is an automorphism of the MPQ-tree. The automorphism ε of the underlying PQ-tree reverses the order of the children of the right Q-node, and then swaps the children of the P-node. The node preserving permutation ν swaps vertices $9, 10$ and it is the only possible node preserving permutation.

Therefore, if $\mathrm{Ker}(\phi)$ is nontrivial, then $\mathrm{Aut}(T)$ is not isomorphic to $\mathrm{Aut}(X)$. In the following text we show that an MPQ-tree representing X captures the whole automorphism group of X.

Automorphism Groups of MPQ-trees. Let M be an MPQ-tree representing an interval graph X and let T be the underlying PQ-tree. If $\varepsilon\colon \mathcal{C}(X) \to \mathcal{C}(X)$ is an automorphism of T and $\nu\colon V(X) \to V(X)$ is a permutation that only swaps vertices of X belonging to the same sections in M, then the pair (ε, ν) is an *automorphism of* M. We say that ν is a *node preserving* permutation with respect to M. Figure 3.2 shows an example of an automorphism of an MPQ-tree.

Observation 3.4. *Automorphisms of an MPQ-tree M representing an interval graph X form a group with the operation defined componentwise.*

We denote the automorphism group of an MPQ-tree by $\mathrm{Aut}(M)$. From the definition of an automorphism of M if follows that

$$\mathrm{Aut}(M) = E \times N,$$

where E is the *automorphism group of the underlying PQ-tree* T and N is the *group of all node preserving permutations.*

Proposition 3.5. *The automorphism group of M is isomorphic to the automorphism group of X.*

Proof. Let M be an MPQ-tree representing an interval graph X. We fix some consecutive ordering on the maximal cliques $\mathcal{C}(X)$ (see Lemma 2.8). Suppose that $\alpha \in$

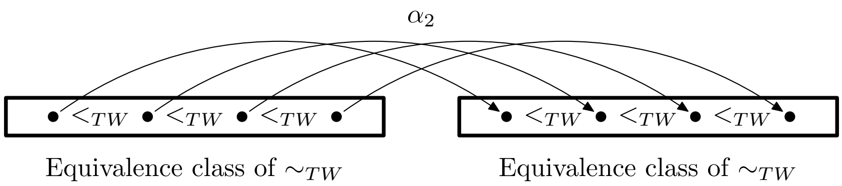


Figure 3.3: The permutation α_1 permutes the vertices in the equivalence class on the left and the permutation α_2 preserves the ordering $<_{TW}$.

$\mathrm{Aut}(X)$. Then α can be decomposed to $\alpha_1 \circ \alpha_2$ such that α_1 only permutes the twin vertices, and α_2 permutes the maximal cliques $\mathcal{C}(X)$ in the same way as α and preserves the ordering $<_{TW}$ on the equivalence classes of the twin relation $\sim_{TW}$. The decomposition is shown in Figure 3.3.

The permutation α_1 can be uniquely identified with a node preserving permutation ν and the permutation α_2 can be uniquely identified with a sequence of equivalence transformations ε. Therefore, the permutation α can be uniquely identified with the automorphism (ε, ν).

On the other hand, if (ε, ν) is an automorphism of M, then there exists a unique automorphism α of X such α' preserves the ordering $<_{TW}$ on the equivalence classes of $\sim_{TW}$ and permutes the maximal cliques $\mathcal{C}(X)$ in the same way as ε. Then the composition $\alpha = \nu \circ \alpha'$ is an automorphism of X uniquely assigned to (ε, ν).

We can define a bijective mapping $\phi\colon \mathrm{Aut}(X) \to \mathrm{Aut}(M)$ by

$$\phi(\alpha) = (\varepsilon, \nu)$$

where ε and ν are as above. It is straightforward to check that ϕ is an isomorphism. □

3.2 Characterization of the Automorphism Groups

In this section we finally derive the characterization of the class $\mathcal{I}$. We show that each group isomorphic to the automorphism group of some interval graph can be built inductively from the trivial group using group products. We use Proposition 3.5 to determine the automorphism group of an interval graph.

Suppose that X is an interval graph. Let M be an MPQ-tree representing X and let T be the underlying PQ-tree. From Proposition 3.5 we have that

$$\mathrm{Aut}(X) \cong \mathrm{Aut}(M) = E \times N$$

where E is the automorphism group of the underlying PQ-tree T and N is the group of node preserving permutations. In other words, each automorphism of X can independently perform two operations: (1) change the consecutive ordering of the maximal cliques $\mathcal{C}(X)$; (2) permute the twin vertices. Those two operations are commutative.

It is quite straightforward to see that the group N is isomorphic to a direct product of symmetric groups. Each node preserving permutation permutes vertices that belong to the same sections of M, that is, vertices that are in the same equivalence class of $\sim_{TW}$. The group which permutes one equivalence class of $\sim_{TW}$ is isomorphic

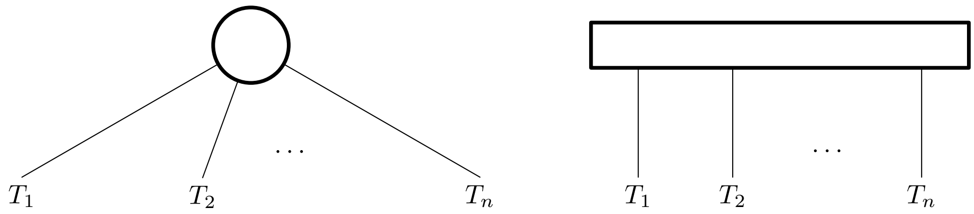


Figure 3.4: Either the root is a P-node, or a Q-node.

to $\mathbb{S}_k$. So, to get the whole group N we just take the direct product of symmetric groups. Therefore,

$$N \cong \mathbb{S}_{k_1}^{l_1} \times \mathbb{S}_{k_2}^{l_2} \times \cdots \times \mathbb{S}_{k_n}^{l_n}$$

where $k_1, \ldots, k_n, l_1, \ldots, l_n \in \mathbb{N}$ and $k_1, \ldots, k_n$ are pairwise distinct.

To determine the automorphism group of X, we need to determine E. To do this, we use a similar technique as Jordan used to determine the automorphism group of a tree (see Theorem 2.6). We distinguish two basic cases: (1) the root of T is a P-node; (2) the root of T is a Q-node. They are depicted in Figure 3.4. The following two lemmas deal with the two cases, respectively.

Lemma 3.6 (The P-node case)**.** *Suppose that the root of T is a P-node P. If $T_1, \ldots, T_n$ are pairwise non-isomorphic PQ-trees, and the subtrees of P consist of k_i isomorphic copies of T_i, $i = 1, \ldots, k$, then*

$$\mathrm{Aut}(T) \cong \mathrm{Aut}(T_1) \wr \mathbb{S}_{k_1} \times \mathrm{Aut}(T_2) \wr \mathbb{S}_{k_2} \times \cdots \times \mathrm{Aut}(T_n) \wr \mathbb{S}_{k_n}.$$

Proof. The proof is similar to the proof of Jordan's Theorem 2.6. First, we deal with a special case. Suppose that the subtrees of P consist *only* of k_1 isomorphic copies of T_1, we denote them by $T_1^1, \ldots, T_1^{k_1}$. From the definition, each automorphism ε of T acts independently on $T_1^1, \ldots, T_1^{k_1}$ and permutes them arbitrarily. So, if $\varepsilon \in \mathrm{Aut}(T)$ is an automorphism of T, then

$$\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{k_1}, \rho)$$

where ε_j is an automorphism of T_1^j and $\rho \in \mathbb{S}_{k_1}$. The operation on $\mathrm{Aut}(T)$ can be algebraically written as

$$\begin{aligned}\delta \cdot \varepsilon &= (\delta_1, \delta_2, \ldots, \delta_{k_1}, \pi) \cdot (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{k_1}, \rho)\\ &= (\delta_1 \cdot \varepsilon_{\pi(1)}, \delta_2 \cdot \varepsilon_{\pi(2)}, \ldots, \delta_{k_1} \cdot \varepsilon_{\pi(k_1)}, \pi \circ \rho).\end{aligned}$$

In other words, the action of $\delta \cdot \varepsilon$ on T can be described in the following way. First, δ acts on each T_1^j using δ_j and permutes $T_1^1, \ldots, T_1^{k_1}$ using π. Since after applying δ on T we have subtrees $T_1^{\pi(1)}, \ldots, T_1^{\pi(k_1)}$, we have to make each ε_j act on the correct subtree. We do this by letting $\varepsilon_{\pi(j)}$ act on $T_1^{\pi(j)}$. It follows that

$$\mathrm{Aut}(T) \cong \mathrm{Aut}(T_1)^{k_1} \rtimes_{\varphi} \mathbb{S}_{k_1} = \mathrm{Aut}(T_1) \wr \mathbb{S}_{k_1}$$

where $\varphi\colon \mathbb{S}_{k_1} \to \mathrm{Aut}(\mathrm{Aut}(T_1)^n)$ is a homomorphism defined naturally by

$$\varphi(\rho) = \text{the automorphism that maps } (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{k_1}) \text{ to } (\varepsilon_{\rho(1)}, \varepsilon_{\rho(2)}, \ldots, \varepsilon_{\rho(k_1)}).$$

Now we consider the general case. From the definition of an automorphism of T we know that *no automorphism* of T can swap some T_i and T_j because they are non-isomorphic. Therefore, each automorphism acts independently on the isomorphic copies of each T_i, so to get $\mathrm{Aut}(T)$ we only need to take the direct product. □

Lemma 3.7 (The Q-node case)**.** *Suppose that the root of T is a Q-node Q. If $T_1, \dots, T_n$ are the subtrees of Q and $T_1 \cong T_n$, $T_2 \cong T_{n-1}$, and so on, then*

$$\mathrm{Aut}(T) \cong (\mathrm{Aut}(T_1) \times \mathrm{Aut}(T_2) \times \cdots \times \mathrm{Aut}(T_n)) \rtimes_\varphi \mathbb{Z}_2$$

where $\varphi\colon \mathbb{Z}_2 \to \mathrm{Aut}(\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_n))$ is a homomorphism defined by

$\varphi(0) =$ *the identity automorphism,*
$\varphi(1) =$ *the automorphism that maps* $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$ *to* $(\varepsilon_n, \varepsilon_{n-1}, \dots, \varepsilon_1)$.

Proof. From the definition, each automorphism ε of T acts independently on $T_1, \dots, T_n$ and possibly reverses their order. So, if $\varepsilon \in \mathrm{Aut}(T)$ is an automorphism of T, then

$$\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n, z)$$

where ε_i is an automorphism of T_i and $z \in \mathbb{Z}_2$, and if $z = 1$, then ε reverses the order of $T_1, \dots, T_n$. The operation on $\mathrm{Aut}(T)$ can be algebraically written as

$$(\delta_1, \dots, \delta_n, z_1) \cdot (\varepsilon_1 \dots, \varepsilon_n, z_2) = \begin{cases} (\delta_1 \cdot \varepsilon_1, \delta_2 \cdot \varepsilon_2, \dots, \delta_n \cdot \varepsilon_n, z_1 + z_2) & \text{if } z_1 = 0 \\ (\delta_1 \cdot \varepsilon_n, \delta_2 \cdot \varepsilon_{n-1}, \dots, \delta_n \cdot \varepsilon_1, z_1 + z_2) & \text{if } z_1 = 1. \end{cases}$$

In other words, the action of $\delta \cdot \varepsilon$ on T can be described in the following way. First, δ acts on each T_i using δ_i. If $z_1 = 1$, then δ reverses the order of $T_1, \dots, T_n$. In this case we have to make ε_i act on the correct subtree. We achieve this by letting ε_n act on T_1, ε_2 act on T_2, and so on. It follows that

$$\mathrm{Aut}(T) \cong (\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_n)) \rtimes_\varphi \mathbb{Z}_2$$

where $\varphi\colon \mathbb{Z}_2 \to \mathrm{Aut}(\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_n))$ is a homomorphism defined as in the statement of the lemma. □

Lemma 3.6 and Lemma 3.7 suggest that the class $\mathcal{I}$ is closed under direct product, wreath product with $\mathbb{S}_n$ and semidirect product of a direct product with $\mathbb{Z}_2$. Lemma 3.8 gives the characterization of $\mathcal{I}$ in terms group products.

Lemma 3.8. *A finite group G is isomorphic to an automorphism group of a finite interval graph if and only if $G \in \mathcal{I}$, where the class $\mathcal{I}$ of finite groups is defined inductively as follows:*

(a) $\{1\} \in \mathcal{I}$.

(b) If $G_1, G_2 \in \mathcal{I}$, then $G_1 \times G_2 \in \mathcal{I}$.

(c) If $G \in \mathcal{I}$ and $n \geq 2$, then $G \wr \mathbb{S}_n \in \mathcal{I}$.

(d) If $G_1, \dots, G_m \in \mathcal{I}$, $m \geq 3$ and $G_1 \cong G_m$, $G_2 \cong G_{m-1}$, and so on, then

$$(G_1 \times \cdots \times G_m) \rtimes_\varphi \mathbb{Z}_2 \in \mathcal{I},$$

where $\varphi\colon \mathbb{Z}_2 \to \mathrm{Aut}(G_1 \times \cdots \times G_m)$ is a homomorphism defined by

$\varphi(0) =$ *the identity automorphism,*
$\varphi(1) =$ *the automorphism that maps $(g_1, g_2, \dots, g_n)$ to $(g_n, g_{n-1}, \dots, g_1)$.*

Proof. First, we prove that for each group $G \in \mathcal{I}$ there exist an interval graph X such that $G \cong \mathrm{Aut}(X)$. We proceed by induction:

- If $G = \{1\}$, then X is the graph containing one vertex.
- Suppose that $G_1, G_2 \in \mathcal{I}$. By the induction hypothesis there exist interval graphs X_1 and X_2 such that $G_1 \cong \mathrm{Aut}(X_1)$ and $G_2 \cong \mathrm{Aut}(X_2)$. We need to show that there exists a graph X such that $G_1 \times G_2 \cong \mathrm{Aut}(X)$. To do this, we just take the disjoint union of X_1 and X_2, or in case that X_1 and X_2 are isomorphic, we hang them on an asymmetric path. The latter case is showed in Figure 3.5. Since the asymmetric path is an interval graph, it follows that the whole graph X is an interval graph.
- Suppose that $G \in \mathcal{I}$ and $n \geq 2$. By the induction hypothesis there exists an interval graph X' such that $G \cong \mathrm{Aut}(X')$. We need to show that there exists an interval graph X such that $G \wr \mathbb{S}_n \cong \mathrm{Aut}(X)$. To do this, it suffices to take the disjoint union of n copies of X'. The result is an interval graph and from Theorem 2.5 we have that $\mathrm{Aut}(X)$ is isomorphic to $G \wr \mathbb{S}_n$.
- Suppose that $G_1, \dots, G_m \in \mathcal{I}$, $m \geq 3$ and $G_1 \cong G_m$, $G_2 \cong G_{m-1}$, and so on. By the induction hypothesis there exists interval graphs $X_1, \dots, X_m$ such that $G_1 \cong \mathrm{Aut}(X_1), \dots, G_m \cong \mathrm{Aut}(X_m)$. Without a loss of generality we assume that $X_1 \cong X_m$, $X_2 \cong X_{m-1}$, and so on. We need to construct a graph X such that $(G_1 \times \cdots \times G_m) \rtimes_\varphi \mathbb{Z}_2 \cong \mathrm{Aut}(X)$ where φ is a homomorphism defined as in the theorem statement. To do this, we hang $X_1, \dots, X_m$ on a path, as shown in Figure 3.6.

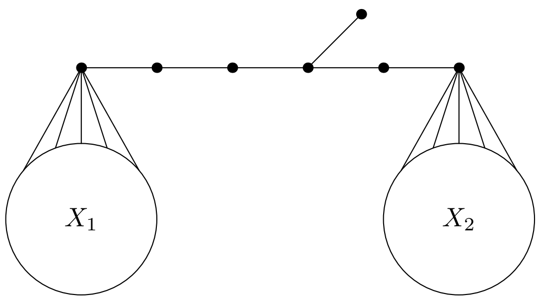


Figure 3.5: Two interval graphs are hanged on an asymmetric path. The two vertices from the asymmetric path are connected with to vertex from the corresponding interval graph.

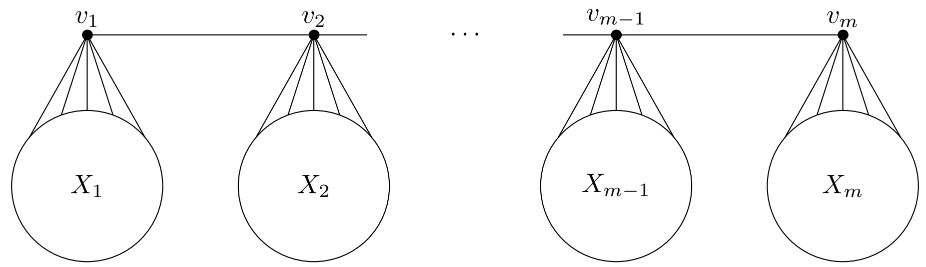


Figure 3.6: Interval graphs hanged on a path. We connect the vertex v_i to each vertex of the graph X_i.

Note that there are only two types of automorphisms of X: the automorphism that swaps the path $v_1v_2\dots v_m$ and the automorphism that fixes the path. This is because we connected v_i with each vertex of X_i. The proof that $(G_1 \times \cdots \times G_m) \rtimes_\varphi \mathbb{Z}_2 \cong \mathrm{Aut}(X)$ is very similar to the proof of Lemma 3.7.

Now, it remains to prove that for each interval graph X there exists a group $G \in \mathcal{I}$ such that $G \cong \mathrm{Aut}(X)$. Here, we make a use of Lemma 3.6, Lemma 3.7 and Proposition 3.5. To determine the group E we use Lemma 3.6 in case the root of T is a P-node, and Lemma 3.7 in case the root T is Q-node. Since $\mathrm{Aut}(M) = E \times N$ where N is a direct product of symmetric groups, we know that $\mathrm{Aut}(M) \in \mathcal{I}$. From Proposition 3.5 we know that $\mathrm{Aut}(M) \cong \mathrm{Aut}(X)$. ☐

Theorem 3.9. *The class $\mathcal{I}$ is the same as the class $\mathcal{T}$.*

Proof. We show that if we add the operation (d) from the statement of Lemma 3.8 to the Jordan's characterization 2.6 of the class $\mathcal{T}$, then we still get the class $\mathcal{T}$.

Suppose that $G_1, \dots, G_m \in \mathcal{T}$ and $T_1, \dots, T_m$ are rooted trees such that $G_1 \cong \mathrm{Aut}(T_1)$, $G_2 \cong \mathrm{Aut}(T_2)$, and so on. We assume that $T_1 \cong T_m$, $T_2 \cong T_{m-1}$, and so on. We need to construct a tree T such that $(G_1 \times \cdots \times G_m) \rtimes_\varphi \cong \mathrm{Aut}(T)$. To do this, we hang the trees $T_1, \dots, T_m$ on the vertices $v_1, \dots, v_m$ of a path by their roots, respectively. Figure 3.7 shows rooted trees hanged on a path by their roots.

It is not true that $\mathrm{Aut}(T) \cong (G_1 \times \cdots \times G_m) \rtimes_\varphi \mathbb{Z}_2$. This can be also seen from the example in Figure 3.7. However, we can fix this easily by subdividing the edges $v_1v_2, v_2v_3, \dots, v_{m-1}v_m$. We proceed from left to right. If the tree hanged on the vertex v_i is isomorphic to the tree on the left of v_i, then we subdivide the edge $v_{i-1}v_i$ and also, symmetrically, the edge $v_{m-i+1}v_{m-i+2}$. We stop in the middle. Figure 3.7 shows the subdivisions that are sufficient to make for the tree in the figure. The proof that $\mathrm{Aut}(T)$ is as desired is similar to the proof of Lemma 3.7. ☐

3.3 On Equality of The Automorphism Groups

We proved that the class $\mathcal{I}$ of the automorphism groups of finite interval graphs is the same as the class $\mathcal{T}$ of the automorphism groups of finite trees. A natural problem is to find for each interval graph X a tree T such that the automorphism group of

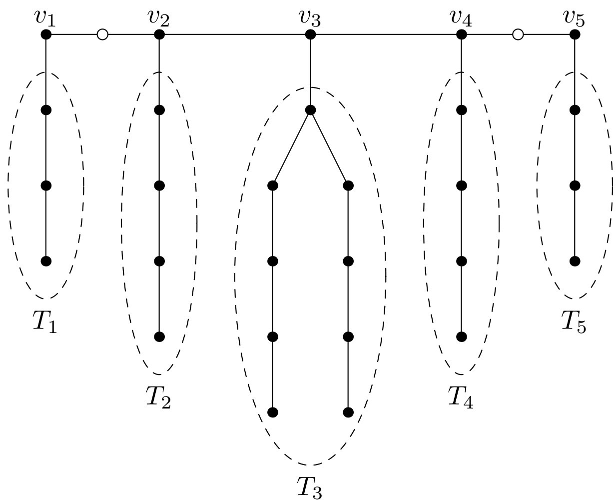


Figure 3.7: Rooted trees hanged on a path by their roots. There exists an automorphism that T_3 with the tree on the left of v_3, and an automorphism that swaps T_2 with the tree on the left of v_2. We fix this by subdividing the edge v_1v_2 and due to the symmetry of the tree also v_4v_5.

X is isomorphic to the automorphism group of T and vice versa. Here, we solve this problem what also gives an alternative proof of Theorem 3.9.

From Interval Graph to Tree. To construct a tree T for an interval graph X such that $\mathrm{Aut}(X) \cong \mathrm{Aut}(T)$ we use Proposition 3.5. Suppose that M is an MPQ-tree representing X. From Proposition 3.5 we have that $\mathrm{Aut}(X) \cong \mathrm{Aut}(M)$ and $\mathrm{Aut}(M) = E \times N$ where E the automorphism groups of the underlying PQ-tree and N is a direct product of symmetric groups. We first construct a tree T_1 that has the automorphism group isomorphic to E and a tree T_2 that has the automorphism group isomorphic to N. Then we construct the tree T by hanging T_1 and T_2 on an asymmetric path (an example of an asymmetric path is shown in Figure 3.5).

To construct T_1, we take the underlying PQ-tree of the MPQ-tree M and replace every leaf and every P-node by a single vertex, and every Q-node by a path. Figure 3.8 shows how each Q-node is replaced by a path. We need to force that every such path can be only reversed and fixed by an automorphism. To do this, we do some subdivision as in the proof of Theorem 3.9 if necessary. It is possible that some leaves in the underlying PQ-tree can not be swapped by an automorphism. This can be forced in the constructed tree T_1 by attaching asymmetric paths of various lengths to the corresponding leaves.

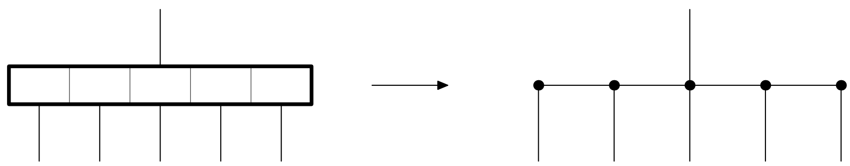


Figure 3.8: Each Q-node in the underlying PQ-tree is replaced by a path.

If N is isomorphic to $\mathbb{S}_k^l$ then T_2 is a rooted tree constructed by rooting l sub-

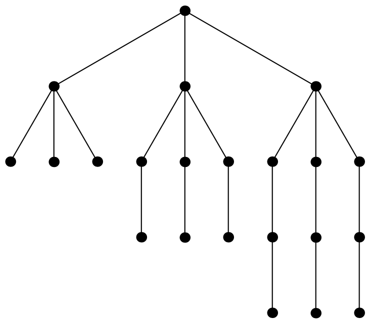


Figure 3.9: A tree with the automorphism group isomorphic to $\mathbb{S}_3^3$.

divided complete bipartite graphs $K_{1,k}$. Figure 3.9 shows an example of such tree for $N \cong \mathbb{S}_3^3$. If N is isomorphic to $\mathbb{S}_{k_1}^{l_1} \times \cdots \times \mathbb{S}_{k_n}^{l_n}$ then we construct a tree for each $\mathbb{S}_{k_i}^{l_i}$ and by rooting them together we get T_2.

From Tree to Interval Graph. The idea is to place the intervals so that they copy the pattern of the given tree T, as shown in Figure 3.10. We assume that T is a rooted tree, let r be the root and let $c_1, \dots, c_n$ be its children. We choose an interval R to represent the root r. Then we choose an interval C_i for each of its children so that $C_i \cap C_j = \emptyset$ and $C_i \subseteq R$. We recursively by construct the subtrees of each child c_i.

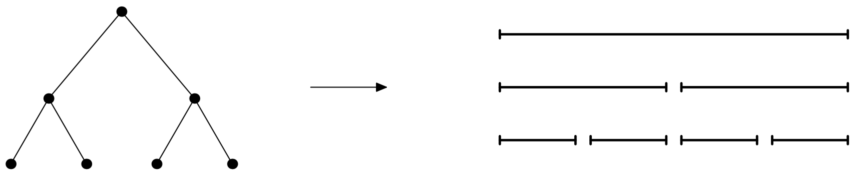


Figure 3.10: Intervals placed according to the pattern of the tree.

If T contains vertices with only one child, then it is not true that the automorphism group of the interval graph constructed in this way is isomorphic to $\mathrm{Aut}(T)$. The construction creates twin vertices that can be permuted. This can be fixed by adding asymmetric paths. Figure 3.11 shows an example.

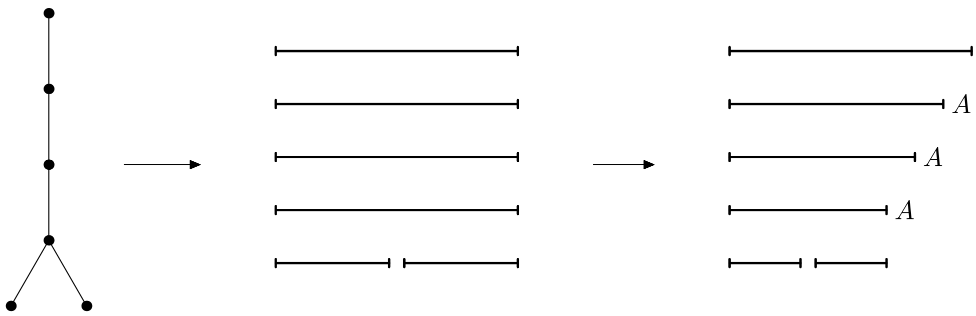


Figure 3.11: The automorphism group of an interval graph constructed by placing intervals according to the pattern of a tree can be lager than the automorphism group of the tree. We fix this by adding copies of an asymmetric path A (an example of an asymmetric path is shown in Figure 3.5), which has the automorphism group isomorphic to the trivial group.

4 Conclusions

We conclude this thesis by describing some important intersection-defined classes of graphs, namely circle graphs, function graphs and circular-arc graphs. We know that those classes of geometric intersection graphs have different automorphism groups than trees, since all of them contain the graph C_4. The automorphism group of C_4 is isomorphic to the dihedral group $\mathbb{D}_4$ which does not belong to $\mathcal{T}$.

Circle Graphs. Circle graphs are intersection graphs of chords of a circle. They were first considered by Even and Itai [11] in the study of stack sorting techniques. The structure of all representations of circle graphs is described in [7].

A *circle representation* $\mathcal{R}$ of a graph X is a set of chords $\{C_x \colon x \in V(X)\}$ such that $xy \in E(X)$ if and only if the chords C_x and C_y intersect. A graph X is a circle graph if there exists a circle representation $\mathcal{R}$ of X. Figure 4.1 shows an example of circle graph and its circle representation.

Function Graphs. A representation of a function graph assigns a continuous function $f \colon [0,1] \to \mathbb{R}$ to every vertex of the graph. Edges are represented by intersections of those functions. The class of permutation graphs, which is a subclass of function graphs, contains graphs that can be represented in the same way, but by linear functions. Figure 4.2 shows an example of a permutation graph. The structure of all representations of function is described in [24].

Function graphs are the complements of so-called *comparability graphs* [18]. A

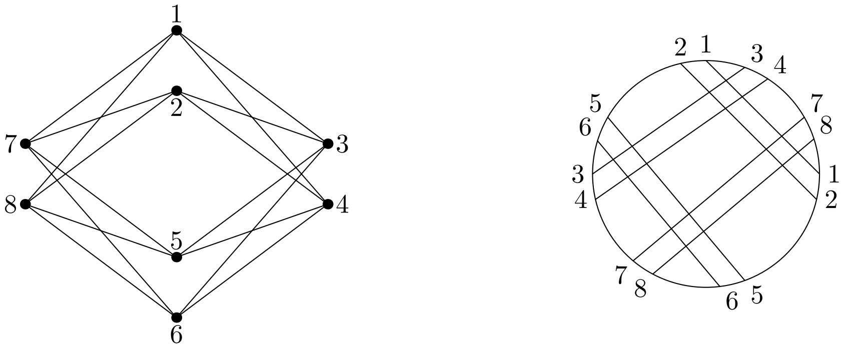


Figure 4.1: Circle graph and its circle representation. The automorphism group of the graph is isomorphic to $\mathbb{Z}_2^4 \rtimes \mathbb{D}_4$.

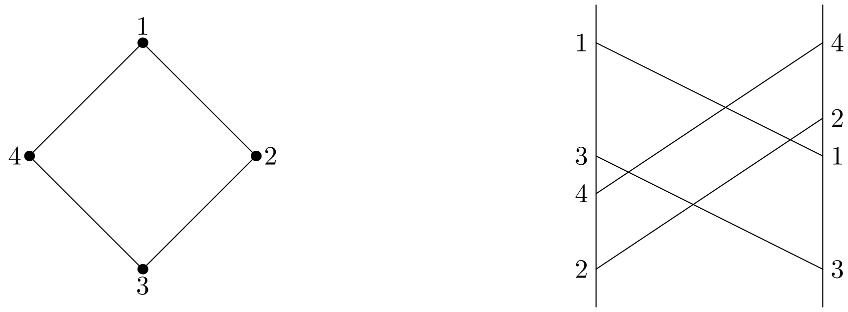


Figure 4.2: Permutation graph and its representation. The automorphism group of the graph is isomorphic to $\mathbb{D}_4$.

comparability graph the graph of some partial ordering. In other words, comparability graphs are graphs of which edges can be oriented trasitively. Permutation graphs are exactly the intersection of function and comparability graphs [12].

Circular-arc Graphs. Circular-arc graphs are intersection graphs of arcs on a circle. Figure 4.3 shows an example of a circular-arc graph. They are natural generalization of interval graphs. If there exists a point on the circle that is not covered by an arc, then the circle can be cut at that point and stretched to a line, which yields an interval representation.

The class of circular-arc graphs is very different from the class of interval graphs. The main difference is that in the case of circular-arc graphs, the maximal cliques do not behave so nicely. A circular-arc graph can have exponential number of maximal cliques.

Generalizing some of the results known for interval graphs to the class of all circular-arc graphs is a challenging problem. McConnel [31] solved the recognition problem for circular-arc graphs in linear time. However, no polynomial-time isomorphism test for circular-arc graphs is currently known. For some time it seemed that the problem is solved since Hsu [22] claimed to have a polynomial-time algorithm, but only recently it was proved he dealt incorrectly with one case [9].

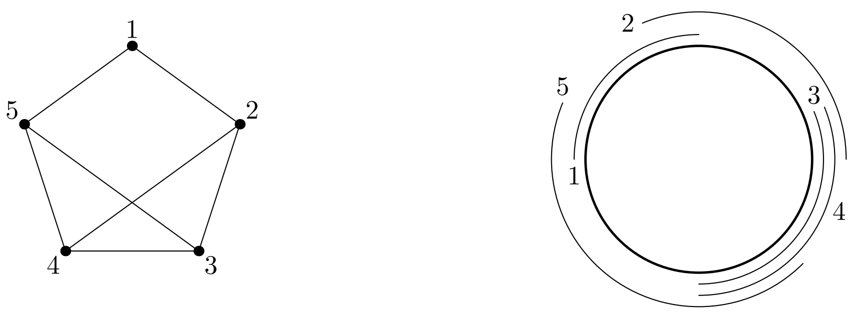


Figure 4.3: Circular-arc graph and its representation. The automorphism group of the graph is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

Bibliography

[1] Arora, S., Barak, B.: Computational complexity: a modern approach. Cambridge University Press (2009)

[2] Babai, L.: Automorphism groups of planar graphs ii. In: Infinite and finite sets (Proc. Conf. Kestzthely, Hungary) (1973)

[3] Benzer, S.: On the topology of the genetic fine structure. Proceedings of the National Academy of Sciences of the United States of America 45(11), 1607 (1959)

[4] Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. J. Comput. System Sci. 13, 335–379 (1976)

[5] Carter, N.: Visual group theory. MAA (2009)

[6] Cayley, P.: Desiderata and suggestions: No. 2. the theory of groups: graphical representation. American Journal of Mathematics 1(2), 174–176 (1878)

[7] Chaplick, S., Fulek, R., Klavík, P.: Extending partial representations of circle graphs. In: Lecture Notes in Computer Science, GD. vol. 8242, pp. 131–142 (2013)

[8] Colbourn, C.J., Booth, K.S.: Linear times automorphism algorithms for trees, interval graphs, and planar graphs. SIAM J. Comput. 10(1), 203–225 (1981)

[9] Curtis, A.R., Lin, M.C., McConnell, R.M., Nussbaum, Y., Soulignac, F.J., Spinrad, J.P., Szwarcfiter, J.L.: Isomorphism of graph classes related to the circular-ones property. arXiv preprint arXiv:1203.4822 (2012)

[10] Dummit, D.S., Foote, R.M.: Abstract algebra (2004)

[11] Even, S., Itai, A.: Queues, stacks and graphs. Theory of Machines and Computations pp. 71–86 (1971)

[12] Even, S., Pnueli, A., Lempel, A.: Permutation graphs and transitive graphs. Journal of the ACM (JACM) 19(3), 400–410 (1972)

[13] Frucht, R.: Herstellung von graphen mit vorgegebener abstrakter gruppe. Compositio Mathematica 6, 239–250 (1939)

[14] Fulkerson, D.R., Gross, O.A.: Incidence matrices and interval graphs. Pac. J. Math. 15, 835–855 (1965)

[15] Gavril, F.: The intersection graphs of subtrees in trees are exactly the chordal graphs. Journal of Combinatorial Theory, Series B 16(1), 47–56 (1974)

[16] Godsil, C.D., Royle, G., Godsil, C.: Algebraic graph theory, vol. 207. Springer New York (2001)

[17] Golumbic, M.C.: Algorithmic graph theory and perfect graphs, vol. 57. Elsevier (2004)

[18] Golumbic, M.C., Rotem, D., Urrutia, J.: Comparability graphs and intersection graphs. Discrete Mathematics 43(1), 37–46 (1983)

[19] Hajós, G.: Über eine Art von Graphen. Internationale Mathematische Nachrichten 11, 65 (1957)

[20] Hanlon, P.: Counting interval graphs. Transactions of the American Mathematical Society 272(2), 383–426 (1982)

[21] Hedrlín, Z., Pultr, A., et al.: On full embeddings of categories of algebras. Illinois Journal of Mathematics 10(3), 392–406 (1966)

[22] Hsu, W.L.: O(mn) algorithms for the recognition and isomorphism problems on circular-arc graphs. SIAM Journal on Computing 24(3), 411–439 (1995)

[23] Jordan, C.: Sur les assemblages de lignes. Journal für die reine und angewandte Mathematik 70, 185–190 (1869)

[24] Klavík, P., Kratochvíl, J., Krawczyk, T., Walczak, B.: Extending partial representations of function graphs and permutation graphs. In: Lecture Notes in Computer Science, ESA. vol. 7501, pp. 671–682 (2012)

[25] Korte, N., Möhring, R.H.: An incremental linear-time algorithm for recognizing interval graphs. SIAM J. Comput. 18(1), 68–81 (1989)

[26] Lekkeikerker, C., Boland, J.: Representation of a finite graph by a set of intervals on the real line. Fundamenta Mathematicae 51(1), 45–64 (1962)

[27] Lueker, G.S., Booth, K.S.: A linear time algorithm for deciding interval graph isomorphism. Journal of the ACM (JACM) 26(2), 183–195 (1979)

[28] Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. Journal of Computer and System Sciences 25(1), 42–65 (1982)

[29] Marczewski, E.S.: Sur deux propriétés des classes d'ensembles. Fund. Math. 33, 303–307 (1945)

[30] Mathon, R.: A note on the graph isomorphism counting problem. Information Processing Letters 8(3), 131–136 (1979)

[31] McConnell, R.M.: Linear-time recognition of circular-arc graphs. Algorithmica 37(2), 93–147 (2003)

[32] McKee, T.A., McMorris, F.R.: Topics in intersection graph theory, vol. 2. Siam (1999)

[33] Miller, G.L.: Graph isomorphism, general remarks. Journal of Computer and System Sciences 18(2), 128–142 (1979)

[34] Roberts, F.S.: Discrete mathematical models, with applications to social, biological, and environmental problems (1976)

[35] Rotman, J.J.: An introduction to the theory of groups, vol. 148. Springer (1995)

[36] Schöning, U.: Graph isomorphism is in the low hierarchy. Journal of Computer and System Sciences 37(3), 312–323 (1988)

[37] Spinrad, J.P.: Efficient Graph Representations.: The Fields Institute for Research in Mathematical Sciences., vol. 19. American Mathematical Soc. (2003)

[38] Stoffers, K.E.: Scheduling of traffic lightsa new approach. Transportation Research 2(3), 199–234 (1968)