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

Peter Zeman

Automorphism Groups of Geometrically Represented Graphs

Supervisor

RNDr. Pavel Klavík

SVOČ 2015, Bratislava

Contents

Abstract

In this thesis, we describe a technique to determine the automorphism group of a geometrically represented graph, by understanding the induced action of the automorphism group on the set of all geometric representations. Each automorphism of a graph can be then interpreted either as an automorphism of a representation, or as a morphism of a representation to another one. This is similar to known results from map theory which relate the automorphism group of a map to the automorphism group of a graph.

We apply this technique to *interval graphs* (intersection graphs of closed intervals of the real line), *permutation graphs* (intersection graphs of linear functions), and to *circle graphs* (intersection graphs of chords of a circle), which are important intersection-defined classes. For those classes of graphs, the structure of all representations can be captured using several structures from algorithmic graph theory, in particular, PQ-trees can be used for interval graphs, modular trees for permutation graph, and split trees for circle graphs. Using group theoretic techniques, e.g. group products, we use those structures to determine the automorphism groups of the corresponding graph classes.

One of our main results is a complete characterization of the automorphism groups of interval, permutation, and circle graphs. Moreover, for interval graphs, we show that their automorphism groups are the same as of trees and, for a given interval graph, we construct a tree with the same automorphism group which answers the Hanlon's question [Trans. Amer. Math. Soc 272(2), 1982].

Some classes of graphs can realize each finite group. *Comparability graphs* (transitively orientable graphs) are one of those classes, since they contain bipartite graphs. They are related to intersection graphs, since they are exactly the complements of *function graphs* (intersection graphs of continuous real-valued functions). The dimension of a poset is the least number of linear orders whose intersection gives the poset. The dimension $\dim(X)$ of a comparability graph X is the dimension of any transitive orientation of X, and by k-DIM we denote the class of comparability graphs X with $\dim(X) \leq k$.

It is known that permutation graphs are equal to 2-DIM, and our characterization of the automorphism groups of permutation graphs gives that 2-DIM is non-universal. For $k \geq 4$, we show that k-DIM is universal.

1 Introduction

The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory. Symmetries play an important role in many distinct areas. In 1846, Galois used symmetries of the roots of a polynomial in order to characterize polynomials which are solvable by radicals.

1.1 Automorphism Groups of Graphs

One possible way how to describe symmetries of a graph X is through its automorphism group $\mathrm{Aut}(X)$. Every automorphism is a permutation of the vertices which preserves adjacencies and non-adjacencies. Frucht [17] proved that every finite group is isomorphic to $\mathrm{Aut}(X)$ of some graph X. Moreover, it is well know that general mathematical structures can be encoded by graphs [30] while preserving automorphism groups.

Most graphs are asymmetric, i.e., have only the trivial automorphism [24]. However, many results in combinatorics and graph theory rely on highly symmetrical graphs. Automorphism groups are important for studying large objects, since they allow one to simplify and understand those objects. This algebraic approach is important for working with big objects.

Highly symmetrical large graphs with nice properties are often constructed algebraically from small graphs. For instance, Hoffman-Singleton graph is a 7-regular graph of diameter 2 with 50 vertices [31]. It has 252000 automorphisms and can be constructed from 25 "copies" of a small multigraph with 2 vertices and 7 edges [38, 44]. Similar constructions are used in designing large computer networks [12, 49]. For instance the well-studied degree-diameter problem asks, given integers d and k, to find a maximal graph X with diameter d and degree k. Such graphs are desirable networks having small degrees and short distances. Currently, the best constructions are highly symmetrical graphs constructed using group theory [39].

Definition 1.1. For a class $\mathcal{C}$ of graphs, we define

$$\mathrm{Aut}(\mathcal{C}) = \bigl\{G : G \text{ is an abstract group}, \exists X \in \mathcal{C} \text{ such that } G \cong \mathrm{Aut}(X)\bigr\}.$$

In other words, $\mathrm{Aut}(\mathcal{C})$ contains all abstract groups that can be realized as an automorphism group of some graph $X \in \mathcal{C}$. A class $\mathcal{C}$ of graphs is called *universal* if every finite group is contained in $\mathrm{Aut}(\mathcal{C})$, and *non-universal* otherwise.

Probably, the first restricted class of graphs whose automorphism groups were studied are *trees* (TREE). In 1869, Jordan [33] gave a characterization of the automorphism groups of trees in terms of group products. He showed that Aut(TREE) contains precisely those groups that can be obtained from the trivial group by applying the direct product and the wreath product with a symmetric group.

Theorem 1.2 (Jordan [33])**.** *The class* Aut(TREE) *is defined inductively as follows:*

(a) $\{1\} \in \mathrm{Aut}(\mathsf{TREE})$.
(b) *If* $G_1, G_2 \in \mathrm{Aut}(\mathsf{TREE})$*, then* $G_1 \times G_2 \in \mathrm{Aut}(\mathsf{TREE})$.
(c) *If* $G \in \mathrm{Aut}(\mathsf{TREE})$ *and* $n \geq 2$*, then* $G \wr \mathbb{S}_n \in \mathrm{Aut}(\mathsf{TREE})$.

A group constructed by (b) (the direct product) acts independently on some non-isomorphic subtrees. On the other hand, a group constructed by (c) (the wreath product with a symmetric group) acts on a tree by permuting some of its isomorphic subtrees.

1.2 Graph Isomorphism Problem

This famous problem asks, for two graphs X and Y, to determine whether they are the same up to some relabeling. It obviously belongs to NP, and it is not known to be polynomially-solvable or NP-complete. The graph isomorphism problem is a prime candidate for an intermediate problem with the complexity between P and NP-complete. It belongs to the low hierarchy of NP [43], which implies that it is unlikely NP-complete. (Unless the polynomial-time hierarchy collapses to its second level.)

The graph isomorphism problem is closely related to the problem of computing generators of the automorphism group of a graph. Two connected graphs X and Y are isomorphic if and only if there exists an automorphism swapping them in $X \dot\cup Y$. On the other hand, it is known that generators of $\mathrm{Aut}(X)$ can be computed by solving $\mathcal{O}(n^4)$ instances of graph isomorphism [37]. By GI, we denote the complexity class of all problems which can be reduced to the graph isomorphism problem.

For many graph classes, graph isomorphism problem was shown to be solvable in polynomial time. For instance, isomorphism of interval graphs and planar graphs [7], circle graphs [32], and permutation graphs [6], can be reduced to isomorphism of trees using structural results known for those classes.

If a class of graphs has very restricted automorphism groups, it seems that the graph isomorphism problem should be relatively easy to solve. However, the complexity of graph isomorphism testing of asymmetric graphs is unknown. There are also very complicated polynomial-time algorithms solving isomorphism for some universal classes of graphs: graphs of bounded degree [36], and graphs with excluded topological subgraphs [27].

Moreover, there exist classes of graphs for which testing graph isomorphism is GI-complete. One of them are bipartite graphs. This can be easily seen by a simple construction. For an arbitrary graph X, we define X' to be the graph obtained from X

by subdividing its edges. Note that X' is bipartite, and $X \cong Y$ if and only if $X' \cong Y'$. A similar construction is known for chordal graphs [7] and other classes.

1.3 Our Results

We study automorphism groups of *geometrically represented graphs.* A well-know class of such graphs are *planar graphs* (PLANAR), i.e., graphs that can be embedded on $\mathbb{R}^2$, or equivalently on the sphere. The main question is how the geometry influences the automorphism groups. In the case of planar graphs, we have that 3-connected planar graphs have unique embeddings on the sphere [47]. Their automorphism groups are *spherical groups.* Automorphism groups of general planar graphs are more complicated and they were described by Babai [1]; see [15, 16] for more details. We focus on intersection representations.

1.3.1 Intersection Representations

An *intersection representation* $\mathcal{R}$ of a graph X is a collection $\{R_v : v \in V(X)\}$ such that $uv \in E(X)$ if and only if $R_u \cap R_v \neq \emptyset$. To get obtain interesting classes of graphs, one typically restricts the sets R_v to particular geometrical objects; for more details on some well-known classes of intersection graphs, see the classical books [25, 45].

In this thesis, prove various results concerning automorphism groups of some well-know classes of intersection graphs and classes that are related to them. In particular, we study interval graphs, permutation graphs, comparability graphs and circle graphs. We show that a well-understood structure of all intersection representations can be used to determine the automorphism group. In the following, we give brief overview of our results and reference the particular chapters for more details.

Interval Graphs. One of the most famous classes of geometric intersection graphs is obtained by restricting the sets R_v to closed intervals of the real line. This representation is called an *interval representation* of a graph. A graph is an *interval graph* if it has an interval representation, i.e., interval graphs are intersection graphs of intervals of the real line; see Figure 1.1. We denote the class of interval graphs by INT.

Interval graphs are one of the oldest and most studied classes of intersection graphs, first introduced by Hajós [28] in 1957. They have many useful theoretical properties and interesting mathematical characterizations. Many computational problems

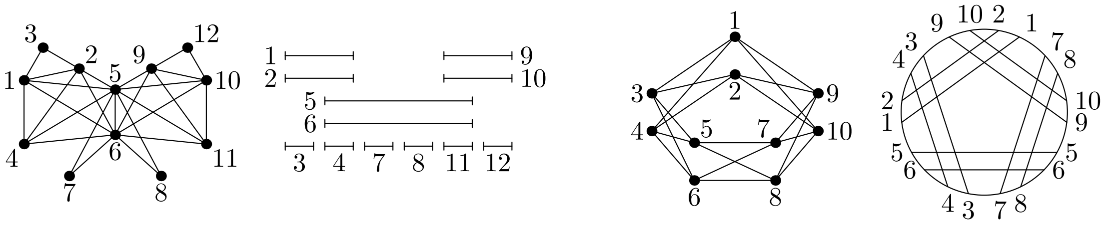

Figure 1.1: On the left, an interval graph and one of its interval representations. On the right, a circle graph and one of its circle representations.

are polynomially solvable for interval graphs. These problems include graph isomorphism, maximum clique, k-coloring, maximum independent set, etc.

One of the reasons why interval graphs were studied quite extensively is that they have real world applications, for example 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 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 from an experiment with mutations is an interval graph. This was considered a strong evidence supporting the theory that genes are arranged in a simple linear fashion. Interval graphs have also many other applications; see [40, 46].

An important subclass of interval graphs are *unit interval graphs* (UNIT INT), which are graphs that have an interval representation with each interval of the length one. *Caterpillar graphs* (CATERPILLAR) are trees with every leaf attached to a central path. They form the intersection of trees and interval graphs.

Theorem 1.3. *The following equalities hold:*

(i) Aut(INT) = Aut(TREE),
(ii) Aut(connected UNIT INT) = Aut(CATERPILLAR),

Concerning (i), this equality is not well known. It was stated by Hanlon [29] without a proof in the conclusion of his paper from 1982 on enumeration of interval graphs. Our structural analysis is based on PQ-trees [4] which combinatorially describe all interval representations of an interval graph. It explains this equality and further solves an open problem of Hanlon: for a given interval graph, to construct a tree with the same automorphism group. Without PQ-trees, this equality is surprising since these classes are very different.

Using PQ-trees, Colbourn and Booth [7] give a linear-time algorithm to compute permutation generators of the automorphism group of an interval graph. In comparison, our description allows to construct an algorithm which outputs the automorphism group in the form of group products which reveals its structure.

Caterpillar graphs which form the intersection of trees and interval graphs have very limited groups and we characterize them in Lemma 3.7. The result (ii) easily follows from the known properties of unit interval graphs and our structural understanding of Aut(INT). The automorphism group of a disconnected unit interval graph can be described using Theorem 2.3.

Comparability Graphs. A *comparability graph* is created from a poset by removing the orientation of the edges. Alternatively, every comparability graph X can be transitively oriented: if $x \to y$ and $y \to z$, then $xz \in E(X)$ and $x \to z$; see Figure 1.2a. This class was first studied by Gallai [20] and we denote it by COMP.

An important parameter of a poset P is its *Dushnik-Miller dimension* [11]. It is the least number of linear orderings $L_1, \dots, L_k$ such that $P = L_1 \cap \dots \cap L_k$. (For

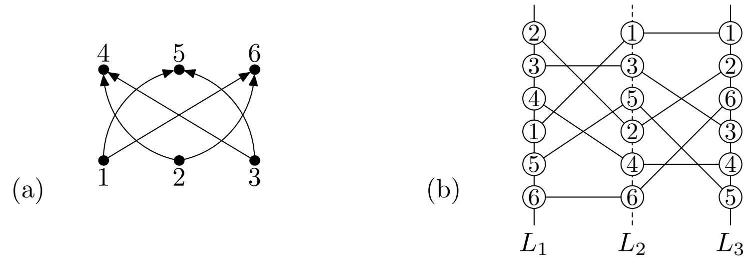

Figure 1.2: (a) A comparability graph with one of its transitive orientations. (b) A function representation of its complement constructed using three linear orders.

a finite poset P, its dimension is always finite since P is the intersection of all its linear extensions.) Similarly, we define the *dimension* of a comparability graph X, denoted by $\dim(X)$, as the dimension of any transitive orientation of X. (It can be shown that every transitive orientation has the same dimension.) By k-DIM, we denote the subclass consisting of all comparability graphs X with $\dim(X) \le k$. We get the following infinite hierarchy of graph classes:

$$\text{1-DIM} \subsetneq \text{2-DIM} \subsetneq \text{3-DIM} \subsetneq \text{4-DIM} \subsetneq \cdots \subsetneq \text{COMP}.$$

Surprisingly, comparability graphs are related to intersection graphs, namely to function and permutation graphs. *Function graphs* (FUN) are intersection graphs of continuous real-valued function on the interval $[0, 1]$. *Permutation graphs* (PERM) are function graphs which can be represented by linear functions [2]; see Figure 1.3.

Golumbic [26] proved that function graphs are the complements of comparability graphs:

$$\text{FUN} = \text{co-CO}.$$

If two functions do not intersect, we can orient the non-edge from the bottom function to the top one which gives a transitive orientation of the complement. On the other hand, suppose that a comparability graph is of dimension k, so one of its transitive orientations can be written as $L_1 \cap \cdots \cap L_k$. We place the vertices according to these orderings on k vertical lines between $[0, 1]$. Then we represent each vertex x by a piecewise linear function which intersects each of the k vertical lines at x; see Figure 1.2b for an example. We get a function representation of the complement. The second relation

$$\text{PERM} = \text{COMP} \cap \text{co-CO} = \text{2-DIM}$$

was shown by Even [14].

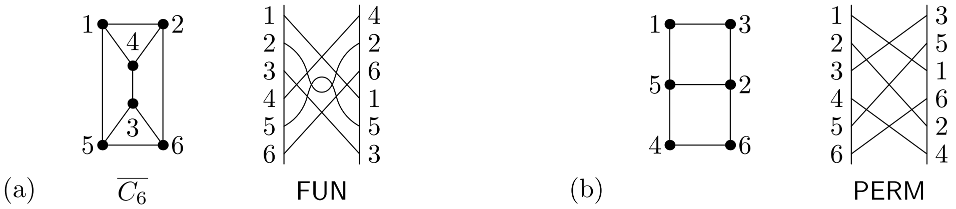

Figure 1.3: (a) A function graph which is not a permutation graph and one of its representations. (b) A permutation graph and one of its representations.

Since 1-DIM consists of all complete graphs, $\mathrm{Aut}(1\text{-DIM}) = \{\mathbb{S}_n, n = 1, \dots\}$. We describe the automorphism groups of 2-DIM = PERM precisely.

Theorem 1.4 (automorphism groups of permutation graphs)**.** *The groups in* Aut(PERM) *are described inductively as follows:*

(a) $\{1\} \in \mathrm{Aut}(\mathsf{PERM})$,
(b) *If* $G_1, G_2 \in \mathrm{Aut}(\mathsf{PERM})$, *then* $G_1 \times G_2 \in \mathrm{Aut}(\mathsf{PERM})$.
(c) *If* $G \in \mathrm{Aut}(\mathsf{PERM})$, *then* $G \wr \mathbb{S}_n \in \mathrm{Aut}(\mathsf{PERM})$.
(d) *If* $G_1, G_2, G_3 \in \mathrm{Aut}(\mathsf{PERM})$, *then* $(G_1^4 \times G_2^2 \times G_3^2) \rtimes \mathbb{Z}_2^2$.

In (d), $\mathbb{Z}_2 \times \mathbb{Z}_2$ acts on G_1^4 as on the vertices of a rectangle, on G_2^2 as on centers of two opposite edges, and on G_3^2 as on centers of the other two opposite edges. Our characterization is similar to Jordan's characterization [33] of the automorphism groups of trees which consists of (a)–(c) (see Theorem 2.4). Therefore, we have that $\mathrm{Aut}(\mathsf{TREE}) \subsetneq \mathrm{Aut}(\mathsf{PERM})$.

We study the induced action of $\mathrm{Aut}(X)$ on the set of all transitive orientations. In the case of permutation graphs, we study the action of $\mathrm{Aut}(X)$ on the pairs of orientations of the graph and its complement, and show that it is semiregular. The transitive orientations are efficiently captured by the modular decomposition which we encode into the modular tree.

We are not aware of any algorithmic result for computing automorphism groups of permutation graphs. From our description, a polynomial-time algorithm can be constructed. Further, it gives $\mathrm{Aut}(X)$ in terms of group product which gives more insight into the structure of $\mathrm{Aut}(X)$.

Comparability graphs are universal since they contain bipartite graphs; we can orient all edges from one part to the other. Since the automorphism group is preserved by complementation and FUN = co-CO, we have Aut(FUN) = Aut(COMP) and therefore function graphs are also universal. In Section 4.2, we explain the universality of FUN and COMP in more detail using the induced action on the set of all transitive orientations.

It is well-known that bipartite graphs have arbitrarily large dimensions: the *crown graph*, which is $K_{n,n}$ without a matching, has dimension n. We give a construction which encodes any graph X into a comparability graph Y with $\dim(Y) \le 4$, while preserving the automorphism group.

Theorem 1.5. *For every* $k \ge 4$*, the class* k-DIM *is universal and its graph isomorphism is* GI*-complete.*

Yannakakis [48] proved that recognizing 3-DIM is NP-complete by a reduction from 3-coloring. For each graph X, a comparability graph Y with several vertices representing each element of $V(X) \cup E(X)$ is constructed. It is shown that $\dim(Y) = 3$ if and only if X is 3-colorable. Unfortunately, the automorphisms of X are lost in Y since it depends on the labels of $V(X)$ and $E(X)$ and Y contains some additional edges according to these labels. We describe a simple and completely different construction which achieves only dimension 4, but preserves the automorphism group: for a given graph X, we create Y by replacing each edge with a path of length eight. However, it is non-trivial to show that $Y \in$ 4-DIM, and the constructed four linear orderings are inspired by [48].

Circle Graphs. *Circle Graphs* (CIRCLE) are intersection graphs of chords of a circle; see Figure 1.1 for an example. They were first considered by Even and Itai [13] in the study of stack sorting techniques, in the early 1970s. Some problem hard computational problems, such as 3-colorability [21], or maximum weighted clique and independent set [22], are solvable in polynomial time on circle graphs. However, there are also many problems that remain NP-complete when restricted to circle graphs.

Theorem 1.6 (automorphism groups of connected circle graphs)**.** *Let $\mathcal{S}$ be a class of finite groups defined inductively as follows:*

(a) $\{1\} \in \mathcal{S}$.
(b) *If* $G_1, G_2 \in \mathcal{S}$, *then* $G_1 \times G_2 \in \mathcal{S}$.
(c) *If* $G \in \mathcal{S}$, *then* $G \wr \mathbb{S}_n \in \mathcal{S}$.
(d) *If* $G_1, G_2, G_3, G_4 \in \mathcal{S}$, *then* $(G_1^4 \times G_2^2 \times G_3^2 \times G_4^2) \rtimes \mathbb{Z}_2 \in \mathcal{S}$.

Then the class Aut(connected CIRCLE) *is defined by the following:*

(e) *If* $G \in \mathcal{S}$, *then* $G^n \rtimes \mathbb{Z}_n \in$ Aut(connected CIRCLE), *for* $n \neq 2$.
(f) *If* $G_1, G_2 \in \mathcal{S}$, *then* $(G_1^n \times G_2^{2n}) \rtimes \mathbb{D}_n \in$ Aut(connected CIRCLE), *for* $n \geq 3$.

The characterization of Aut(connected CIRCLE) is based on split trees which describe all representations of circle graphs. Our approach is similar to the algorithm for circle graph isomorphism [32]. For a disconnected circle graph X, its automorphism group can be easily determined using Theorem 2.3.

Related Graph Classes. Theorems 1.3, 1.4, and 1.6 state that INT, UNIT INT, PERM, and CIRCLE are non-universal. Figure 1.4 shows that some of their well-known superclasses are already universal.

Claw-free graphs (CLAW-FREE) are graphs with no induced $K_{1,3}$. Roberts [41] proved that UNIT INT = CLAW-FREE $\cap$ INT. The *complements of bipartite graphs* (co-BIP) are claw-free and universal.

Chordal graphs (CHOR) are intersection graphs of subtrees of trees. They contain no induced cycles of length four or more and naturally generalize interval graphs. Chordal graphs are universal [35].

Interval filament graphs (IFA) are intersection graphs of the following sets. For every R_u, we choose an interval $[a, b]$ and R_u is a continuous function $[a, b] \to \mathbb{R}$ such that $R_u(a) = R_u(b) = 0$ and $R_u(x) > 0$ for $x \in (a, b)$. They are universal since they generalize circle, chordal, and function graphs.

1.3.2 Groups Acting On Geometric Intersection Representations

For the studied classes of graphs, we interpret our characterizations of their automorphism groups in terms of an action on the set of all equivalent intersection represen-

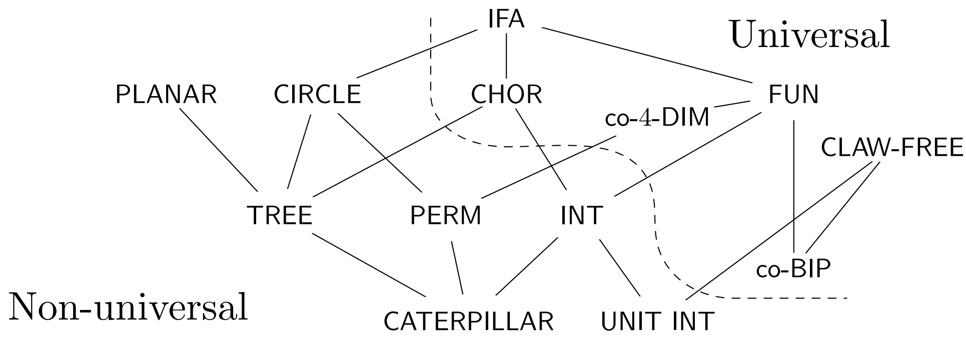

Figure 1.4: The inclusions between the considered graph classes.

tation. We describe a general technique which allows us to geometrically understand automorphism groups of some intersection-defined graph classes.

Our approach is inspired by some well-known results in map theory. A *map* $\mathcal{M}$ is a 2-cell embedding of a graph; i.e, aside vertices and edges, it prescribes a rotation scheme for the edges incident with each vertex. One defines $\mathrm{Aut}(\mathcal{M})$ as the subgroup of $\mathrm{Aut}(X)$ which preserves/reflects the rotational schemes. Unlike $\mathrm{Aut}(X)$, we know that $\mathrm{Aut}(\mathcal{M})$ is always small (since $\mathrm{Aut}(\mathcal{M})$ acts semiregularly on flags) and can be efficiently determined. If the quotient $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{M})$ exists, then it describes morphisms between different maps and can be very complicated.

Action Induced On Geometric Intersection Representations. For a graph X, we denote by $\mathfrak{Rep}$ the set of all its (interval, permutation, circle, etc.) intersection representations. An automorphism $\pi \in \mathrm{Aut}(X)$ defines a morphism of $\mathcal{R} \in \mathfrak{Rep}$ to another representation $\mathcal{R}'$ such that $R'_{\pi(u)} = R_u$, i.e., the automorphism π swaps the labels of the sets representing the vertices of X. We denote $\mathcal{R}'$ by $\pi(\mathcal{R})$. The group $\mathrm{Aut}(X)$ acts on the set $\mathfrak{Rep}$.

The set $\mathfrak{Rep}$ can be very large. Therefore, we define a suitable equivalence relation $\sim$ and we work with $\mathfrak{Rep}/\sim$. It is reasonable to assume that $\sim$ is a congruence relation with respect to the action of $\mathrm{Aut}(X)$ on $\mathfrak{Rep}$, i.e., for every $\mathcal{R} \sim \mathcal{R}'$ and $\pi \in \mathrm{Aut}(X)$, we have $\pi(\mathcal{R}) \sim \pi(\mathcal{R}')$. We consider the action of $\mathrm{Aut}(X)$ on $\mathfrak{Rep}/\sim$.

The stabilizer of $\mathcal{R} \in \mathfrak{Rep}/\sim$, denoted by $\mathrm{Aut}(\mathcal{R})$, describes automorphisms inside the representation $\mathcal{R}$. For a nice class of intersection graphs, $\mathrm{Aut}(\mathcal{R})$ is very simple. If it is a normal subgroup, then the quotient $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$ describes all morphisms which change a representation $\mathcal{R}$ into another representation belonging to the same orbit as $\mathcal{R}$. Our strategy is to understand these morphisms geometrically, which requires an understanding of the structure of all geometric intersection representation. For interval graphs, it is captured by PQ-trees, for permutation graphs by modular trees, and for circle graphs by split trees.

2 Preliminaries

Frist, in Section 2.1 we introduce some basic notation. Then in Section 2.2, we explain some basic tools for constructing larger groups from smaller ones. We conclude Section 2.2 by proving Theorem 2.3 which describes the automorphism groups of disconnected graph in terms of the automorpism groups of their connected components. The idea of the proof of Theorem 2.3 reoccurs multiple times througout the whole thsis. Finally, in Section 2.3, we describe the automorphism groups of trees.

2.1 Notation

This thesis makes links between different areas of mathematics, in particular group theory and graph theory. In this seciton, we define the notation for some basic notions from both fields, to avoid conflicts.

Graph Theory. We use X and Y to denote graphs. The set of the vertices of a graph X is denoted by $V(X)$, and the set of the edges by $E(X)$. By xy we denote and edge joining vertices x and y. The set of all maximal cliques of X is denoted by $\mathcal{C}(X)$. If two vertices x and y belong to precisely the same maximal cliques, we say that x and y are *twin* vertices. This defines an equivalence relation on $V(X)$. The equivalence classes of this relation are called *twin classes*.

Group Theory. We assume that the reader is familiar with the basic group theory. For a comprehensive treatment of the basics of group theory, see for example [42, 10], for a visual treatment of group theory see [5]. Abstract groups are usually denoted in the text by G and H.

The automorphism group of a graph X is denoted by $\mathrm{Aut}(X)$. The group $\mathrm{Aut}(X)$ *acts* on on the set of vertices $V(X)$. The stabilizer of a vertex x is defined by

$$\mathrm{Stab}_X(x) = \big\{\pi \in \mathrm{Aut}(X) : \pi(x) = x\big\},$$

and the orbit of a vertex x is defined by

$$\mathrm{Orb}_X(x) = \big\{y \in V(X) : \pi(x) = y, \pi \in \mathrm{Aut}(X)\big\}.$$

Suppose now that Y is an induced subgraph of X. We define the stabilizer $\mathrm{Stab}_X(Y)$ of Y to be the set-wise stabilizer of $V(Y)$.

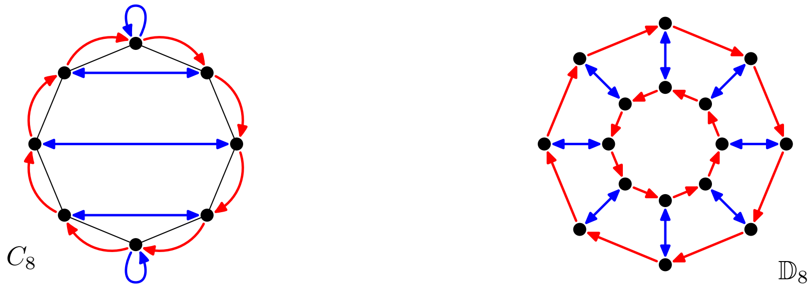

Figure 2.1: The cycle of lenght 8 with the action of $\mathbb{D}_8$ on its vertices.

We use the following notation for the standard groups:

- $\mathbb{S}_n$ is the *symmetric group* whose elements are n-element permutations,
- $\mathbb{D}_n$ is the *dihedral group* whose elements are symmetries of the regular n-gon, including both rotations and reflections,
- $\mathbb{Z}_n$ is the *cyclic group* whose elements are integers $0, \dots, n-1$ and the operation is addition modulo n.

Figures 2.1 and 2.2 show examples of graphs and their automorphism groups.

2.2 Group Products

In algebra, group products are used to decompose large groups into smaller ones. Consider for example the well know puzzle called the Rubik's Cube. The Rubik's Cube group is the set G of all *cube moves* on the Rubik's Cube. The cardinality of this group is

$$|G| = 43,252,003,274,489,856,000 \approx 4.3 \cdot 10^{19}.$$

The Rubik's Cube group is a huge object which seems to be very complicated. Using group products, one can understand the structure of this group. It is isomorphic to

$$(\mathbb{Z}_3^7 \times \mathbb{Z}_2^{11}) \rtimes \big((\mathbb{A}_8 \times \mathbb{A}_{12}) \rtimes \mathbb{Z}_2\big),$$

where $\mathbb{A}_n$ is the group of all even n-element permutations. One can combinatorially interpret the terms of these products and gain a lot of insight into the structure of

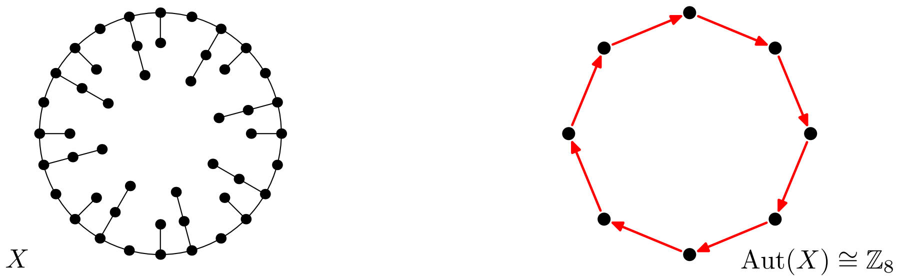

Figure 2.2: A graph X with its automorphism group $\mathrm{Aut}(X)$ isomorphic to the group $\mathbb{Z}_8$.

the Rubik's Cube. This can be used, for instance, to design algorithms solving the Rubik's Cube, or to understand the smallest number of moves necessary to solve it in any position. (Which is only 20.)

In this section, we explain two basic group theoretic methods for constructing larger groups from smaller ones, in particular *direct product* and *semidirect product*. We show how these group products can be used to construct automorphism groups of graphs. At the end of this section, we prove a formula (Theorem 2.3) for constructing the automorphism group of a disconnected graph from the automorphism groups of its connected components.

2.2.1 Direct Product

Let G and H be groups with operations $\cdot_G$ and $\cdot_H$, respectively. Their *direct product* $G \times H$ is a group having as elements all pairs (g, h) where $g \in G$ and $h \in H$. The operation is defined componentwise:

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

When there is no confusion we simply write $(g_1 \cdot g_2, h_1 \cdot h_2)$ or $(g_1 g_2, h_1 h_2)$. Figure 2.3 shows an example. The direct product of n groups is defined similarly, and we use G^n as a shorthand for the product $G \times G \times \cdots \times G$ with n terms.

Both G and H are normal subgroups of $G \times H$. On the other hand, the semidirect product, discussed in Section 2.2.2, constructs from two groups G and H a larger group for which only G is ensured to be a normal subgroup.

The direct product can be used to construct automorphism groups of graphs that are disconnected and their connected components are pairwise non-isomorphic. In this case, the automorphism group of a graph X is the direct product of the automorphism groups of its connected components $X_1, \dots, X_k$:

$$\mathrm{Aut}(X) = \mathrm{Aut}(X_1) \times \cdots \times \mathrm{Aut}(X_k).$$

The reason is that each automorphism acts independently on each component.

2.2.2 Semidirect and Wreath Products

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

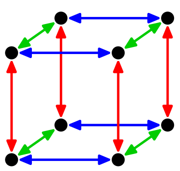

Figure 2.3: The group $\mathbb{Z}_2^3$ contains two copies of $\mathbb{Z}_2^2$, with the corresponding elements connected according to the group $\mathbb{Z}_2$.

Figure 2.4: Two graphs X and Y. We have $\mathrm{Aut}(X) \cong \mathbb{S}_3 \times \mathbb{Z}_2$, but we need the semidirect product to describe $\mathrm{Aut}(Y)$.

We start with a simple example of two graphs, shown in Figure 2.4. The automorphism group of the graph X is isomorphic to $\mathbb{S}_3 \times \mathbb{Z}_2$, but the automorphism group of the graph Y is not $\mathbb{Z}_2 \times \mathbb{Z}_2$. The direct product does not include the automorphisms which swap the components. Moreover, $\mathrm{Aut}(Y)$ is not even isomorphic to $\mathbb{Z}_2^3$ because, for example, swapping the components and swapping the vertices of the left component does not commute.

Semidirect Product (External). As already stated, both G and H are normal subgroups of $G \times H$. The *semidirect product* generalizes the direct product since it only requires G to be a normal subgroup. This is one of the motivations for studying semidirect products since they allow to decompose a bigger number of groups.

The direct product $G \times H$ contains identical copies of G, with corresponding elements connected according to H, as shown in Figure 2.3. In the semidirect product of the groups G and H, the group H also determines how some copies of G are connected. However, these copies of G do not need to be all identical.

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

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

The elements are all pairs (g, f) such that $g \in G$ and $f \in \mathrm{Aut}(G)$. The operation is 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 $G \rtimes \mathrm{Aut}(G)$ defined like this forms a group. Its identity element is $(1, 1)$ and the element (g, f) has the inverse $(f^{-1}(g^{-1}), f^{-1})$.

We can think of it as all possible isomorphic copies of G are connected according to $\mathrm{Aut}(G)$. 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 the same elements of another isomorphic copy of G.

In general, the semidirect product is defined 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 of $G \rtimes \mathrm{Aut}(G)$:

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

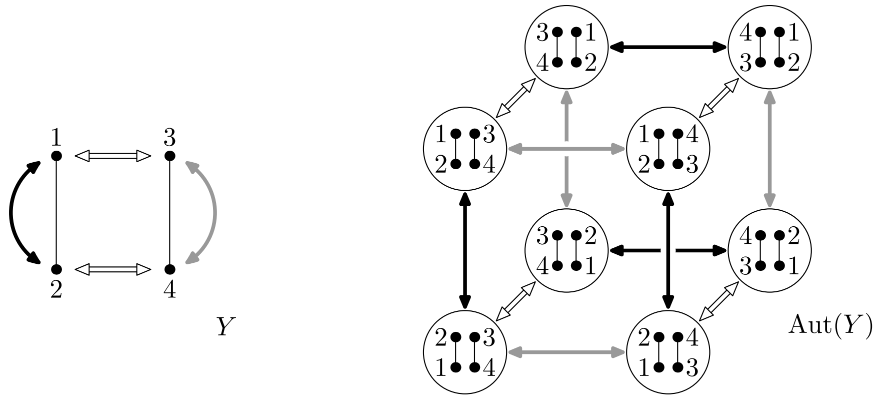

Figure 2.5: The structure of $\mathrm{Aut}(Y)$, generated by three involutions acting on Y on the left. It follows that $\mathrm{Aut}(Y)$ is isomorphic to $\mathbb{Z}_2^2 \rtimes \mathbb{Z}_2 = \mathbb{Z}_2 \wr \mathbb{Z}_2$.

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

Semidirect Product (Internal). The semidirect product be also defined internally. In the previous definition, we were given two groups and a homomorphism and we constructed a new group. Now we are given a group G, a normal subgroup N of G, and a subgroup H of G such that $N \cap H = \{1\}$. Then we say that G is the semidirect product of N and H, i.e., $G = N \rtimes H$.

The two definitions of semidirect product are equivalent, i.e., if $G = N \rtimes H$, we can find a homomorphism $\varphi\colon H \to \mathrm{Aut}(N)$ such that $G = N \rtimes_\varphi H$. The second definition can be more useful when one wants to prove that $G = N \rtimes H$, given the groups N and H.

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

Example 2.2. Let Y be the graph from Figure 2.4. The group $\mathrm{Aut}(Y)$ is isomorphic to $\mathbb{Z}_2^2 \rtimes \mathbb{Z}_2$. Figure 2.5 shows a $\mathrm{Aut}(Y)$. The elements of the two isomorphic copies of $\mathbb{Z}_2^2$ are connected according to the pattern of $\mathbb{Z}_2$.

Wreath Product. The group $G \wr \mathbb{S}_n$ is called the *wreath product* of G with $\mathbb{S}_n$.[1] It is a shorthand for the semidirect product $G^n \rtimes_\varphi \mathbb{S}_n$, where the homomorphism $\varphi\colon \mathbb{S}_n \to \mathrm{Aut}(G^n)$ is defined by

$$\varphi(\pi) = (g_1, \dots, g_n) \mapsto (g_{\pi(1)}, \dots, g_{\pi(n)}).$$

The reason for defining the wreath product is that it occurs quite often in group theory. It also plays an important role in describing the automorphism groups of graphs, as we now illustrate.

[1] For the purposes of this paper, it is sufficient to define $G \wr \mathbb{S}_n$. In general, the wreath product is defined for every pair of groups G and H [42].

2.2.3 Automorphism Groups of Disconnected Graphs

We prove Theorem 2.3. It shows how to construct the automorphism group of a disconnected graph from the automorphism groups of its connected components, using group products. The idea of the proof is essential for many further results.

Theorem 2.3. *If $X_1, \ldots, X_n$ are pairwise non-isomorphic connected graphs and X is the disjoint union of k_i copies of X_i, for $i = 1, \ldots, n$, 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. Since the action of $\mathrm{Aut}(X)$ is independent on non-isomorphic components, it is clearly the direct product of factors, each corresponding to the automorphism group of one isomorphism class of components. It remains to show that if X consists of k isomorphic components of a connected graph Y, then $\mathrm{Aut}(X) \cong \mathrm{Aut}(Y) \wr \mathbb{S}_k$.

We isomorphically label the vertices of each component. Then each automorphism $\pi \in \mathrm{Aut}(X)$ is a composition $\sigma \cdot \tau$ of two automorphisms: σ maps each component to itself, and τ permutes the components as in π while preserving the labeling. Therefore, the automorphisms σ can be bijectively identified with the elements of $\mathrm{Aut}(Y)^k$ and the automorphisms τ with the elements of $\mathbb{S}_k$.

Let $\pi, \pi' \in \mathrm{Aut}(X)$. Consider the composition $\pi \cdot \pi' = \sigma \cdot \tau \cdot \sigma' \cdot \tau'$. We want to swap τ with σ' and rewrite the automorphism $\pi \cdot \pi'$ as a composition $\sigma \cdot \hat{\sigma} \cdot \hat{\tau} \cdot \tau'$. Clearly, connected the components are permuted in $\pi \cdot \pi'$ exactly as in $\tau \cdot \tau'$, so $\hat{\tau} = \tau$. On the other hand, $\hat{\sigma}$ is not necessarily equal σ'. Let σ' be identified with the vector $(\sigma'_1, \ldots, \sigma'_k) \in \mathrm{Aut}(Y)^k$. Since σ' is applied after τ, it acts on the components permuted according to τ; see Figure 2.6. Therefore $\hat{\sigma}$ is constructed from σ by permuting the coordinates of its vector by τ:

$$\hat{\sigma} = (\sigma'_{\tau(1)}, \ldots, \sigma'_{\tau(k)}).$$

This is precisely the definition of the wreath product, so $\mathrm{Aut}(X) \cong \mathrm{Aut}(Y) \wr \mathbb{S}_k$. ☐

2.3 Automorphism Groups of Trees

In this section, we prove Jordans's characterization of Aut(TREE). It says that every group in Aut(TREE) can be build inductively from the trivial group, using the direct product and the wreath product with a symmetric group.

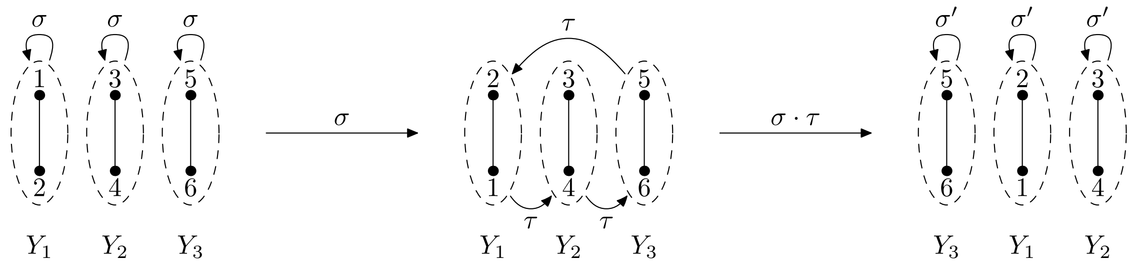

Figure 2.6: A graph X with its the components Y_1, Y_2, and Y_3. The automorphism σ' acts on the components permuted by $\pi = \sigma \cdot \tau$.

Theorem 2.4 (Jordan [33]). *The class* $\mathrm{Aut}(\mathsf{TREE})$ *is defined inductively as follows:*

(a) $\{1\} \in \mathrm{Aut}(\mathsf{TREE})$.
(b) If $G_1, G_2 \in \mathrm{Aut}(\mathsf{TREE})$*, then* $G_1 \times G_2 \in \mathrm{Aut}(\mathsf{TREE})$.
(c) If $G \in \mathcal{T}$ *and* $n \geq 2$*, then* $G \wr \mathbb{S}_n \in \mathrm{Aut}(\mathsf{TREE})$.

Proof. Every tree has a center, which is either a vertex, or an edge. If the center is an edge, then we can subdivide this edge while preserving the automorphism group. The center of a tree is fixed by every automorphism, and similarly the distance from the center is preserved. Therefore, it is sufficient to prove this theorem for rooted trees.

Clearly, $\{1\} \in \mathrm{Aut}(\mathsf{TREE})$. It remains to show that the class $\mathrm{Aut}(\mathsf{TREE})$ is closed under (b) and (c).

- Let $G_1, G_2 \in \mathrm{Aut}(\mathsf{TREE})$, and let T_1 and T_2 be rooted trees such that $\mathrm{Aut}(T_1) \cong G_1$ and $\mathrm{Aut}(T_2) \cong G_2$. We construct a rooted tree T by attaching the roots of T_1 and T_2 to a new root. If $T_1 \cong T_2$, then we further subdivide one of the newly created edges. Clearly, we get $\mathrm{Aut}(T) \cong G_1 \times G_2$.

- Let $G \in \mathrm{Aut}(\mathsf{TREE})$, and let T' be a rooted tree such that $\mathrm{Aut}(T') \cong G$. We construct T by attaching n copies of T' to the same root. It can be easily proven that $\mathrm{Aut}(T) \cong G \wr \mathbb{S}_n$. (One would proceed similarly as in the proof of Theorem 2.3).

Finally, we show that if T is a rooted tree, then $\mathrm{Aut}(T) \in \mathrm{Aut}(\mathsf{TREE})$. If T contains only one vertex, then $\mathrm{Aut}(T) \cong \{1\}$ and it belongs to $\mathrm{Aut}(\mathsf{TREE})$. Otherwise, we delete the root and get a forest of rooted trees $T_1, \dots, T_n$. By induction, we have that the automorphism group of each T_i belongs to $\mathrm{Aut}(\mathsf{TREE})$. It follows from Theorem 2.3 that we can construct $\mathrm{Aut}(T)$ using (b)–(c). Therefore $\mathrm{Aut}(T) \in \mathrm{Aut}(\mathsf{TREE})$. $\square$

3 Interval Graphs

In this chapter, we prove Theorem 1.3(i) (Section 3.2). We introduce PQ-trees which combinatorially describe all interval representations of a given interval graph. We use modified PQ-trees of Korte and Möhring [34], which capture some additional information about the vertices (Section 3.1). In Section 3.2, we derive a characterization of Aut(INT) using MPQ-trees, and prove it to be equivalent to the Jordan's characterization of Aut(TREE). In Section 3.3, we solve Hanlon's open problem [29] by constructing for a given interval graph a tree with the same automorphism group, and we also show the converse construction. Finally, in Section 3.4 we use the results of Section 3.1 and 3.2 to prove Theorem 1.3(ii).

3.1 PQ- and MPQ-trees

In 1965, Fulkerson and Gross proved the following fundamental characterization of interval graphs by orderings of maximal cliques.

Lemma 3.1 (Fulkerson and Gross [18])**.** *A graph X is an interval graph if and only if there exists an ordering $\preceq$ of $\mathcal{C}(X)$ such that for every $x \in V(X)$ the maximal cliques containing x appear consecutively in this ordering.*

Sketch of a proof. Let $\{I_x : x \in V(X)\}$ 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 it contains a point c_i. The ordering of $c_1, \dots, c_k$ from left to right gives the required ordering.

For the other implication, given an ordering of the maximal cliques $C_1, \dots, C_k$, we place points $c_1, \dots, c_k$ in this ordering on the real line. To each vertex x, we assign the minimal interval I_x such that $c_i \in I_x$ for all $x \in C_i$. We obtain a valid interval representation $\{I_x : x \in V(X)\}$ of X. $\square$

An ordering $\preceq$ of the maximal cliques satisfying the statement of Lemma 3.1 is called a *consecutive ordering*.

3.1.1 PQ-trees

Booth and Lueker [4] invented a data structure called *PQ-tree* which encodes all consecutive orderings of an interval graph. They use this structure for recognizing interval graphs in linear time which was a long standing open problem. PQ-trees give a lot of insight into the structure of all interval representations, and have applications to many problems. We use them to capture the automorphism groups of interval graphs.

PQ-tree. A rooted tree T is a PQ-tree representing an interval graph X if and only if it satisfies the following:

- It has two types of inner nodes: *P-nodes* and *Q-nodes*. For every inner node, its children are ordered from left to right. Each P-node has at least two children and each Q-node at least three.
- The leaves of T correspond one-to-one to $\mathcal{C}(X)$. The *frontier* of T is the order $\preceq$ of the leaves from left to right.
- Two PQ-trees are *equivalent* if one can be obtained from the other by a sequence of two *equivalence transformations*: (i) an arbitrary permutation of the ordering of the children of a P-node, and (ii) the reversal of the order of the children of a Q-node. The consecutive orderings of $\mathcal{C}(X)$ are exactly the frontiers of the PQ-trees equivalent with T.

Booth and Lueker [4] give a constructive proof of existence and uniqueness (up to equivalence transformations) of PQ-trees. Figure 3.1 shows an example of a PQ-tree representing an interval graph.

For a PQ-tree T, we consider all sequences of equivalent transformations. Two such sequences are *equivalent* if they transform T the same. Each sequence consists of several transformations of the inner nodes, and it is easy to see that these transformations are independent. If a sequence transforms one inner node several times, it can be replaced by a single transformation of this node. Let $\Sigma(T)$ be the quotient of all sequences of equivalent transformations of T by this equivalence. We represent each class by a sequence which transforms each node at most once.

Observe that $\Sigma(T)$ forms a group with the concatenation as the group operation. This group is isomorphic to a direct product of symmetric groups. The order of $\Sigma(T)$ is equal to the number of equivalent PQ-trees of T. Let $T' = \sigma(T)$ for some $\sigma \in \Sigma(T)$. Then $\Sigma(T') \cong \Sigma(T)$ since $\sigma' \in \Sigma(T')$ corresponds to $\sigma\sigma'\sigma^{-1} \in \Sigma(T)$.

3.1.2 MPQ-trees.

A *modified PQ-tree* is created from a PQ-tree by some adding information about the vertices. They were described by Korte and Möhring [34] to simplify linear-time recognition of interval graphs. It is not widely known but the equivalent idea was used earlier by Coulborn and Booth [7] to design a linear-time algorithm for computing generators of the automorphism group an interval graph.

MPQ-tree. We construct the MPQ-tree M form a PQ-tree T representing an interval graph X by assigning subsets of $V(X)$, called *sections*, to the nodes of T; for an

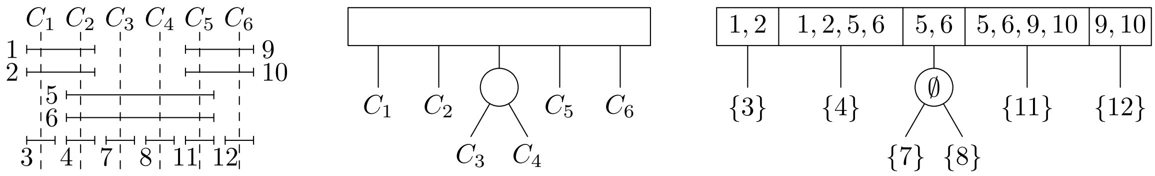

Figure 3.1: An ordering of the maximal cliques, and the corresponding PQ-tree and MPQ-tree. The P-nodes are denoted by circles, the Q-nodes by rectangles. There are four different consecutive orderings.

example see Figure 3.1. The leaves and the P-nodes have each assigned exactly one section while the Q-nodes have one section per child. We assign these sections as follows:

- For a leaf L, the section $\mathrm{sec}(L)$ contains those vertices that are only in the maximal clique represented by L, and no other maximal clique.
- For a P-node P, the section $\mathrm{sec}(P)$ contains those vertices that are in all maximal cliques of the subtree of P, and no other maximal clique.
- For a Q-node Q and its children $T_1, \dots, T_n$, the section $\mathrm{sec}_i(Q)$ contains those vertices that are in the maximal cliques represented by the leaves of the subtree of T_i and also some other T_j, but not in any other maximal clique outside the subtree of Q. We put $\mathrm{sec}(Q) = \mathrm{sec}_1(Q) \cup \cdots \cup \mathrm{sec}_n(Q)$.

Korte and Möhring [34] proved existence of MPQ-trees and many other properties, for instance each vertex appears in sections of exactly one node and in the case of a Q-node in consecutive sections. Two vertices are in the same sections if and only if they belong to precisely the same maximal cliques, i.e., if and only if they are twin vertices. Figure 3.1 shows an example.

3.1.3 Automorphisms of MPQ-trees

For every graph X, its automorphism group $\mathrm{Aut}(X)$ induces an action on $\mathcal{C}(X)$ since every automorphism permutes the maximal cliques. For an interval graph X and $\pi \in \mathrm{Aut}(X)$, a consecutive ordering $\preceq$ is transformed to another consecutive ordering, denoted by $\pi(\preceq)$.

Suppose that an MPQ-tree M representing X has the frontier $\preceq$. For every automorphism $\pi \in \mathrm{Aut}(X)$, there exists the unique MPQ-tree M' with the frontier $\pi(\preceq)$ which is equivalent to M. We define a mapping

$$\Phi : \mathrm{Aut}(X) \to \Sigma(M)$$

such that $\Phi(\pi)$ is the sequence of equivalent transformations which transforms M to M'. It is easy to observe that Φ is a group homomorphism.

By the fundamental homomorphism theorem, we know that

$$\mathrm{Im}(\Phi) \cong \mathrm{Aut}(X)/\mathrm{Ker}(\Phi).$$

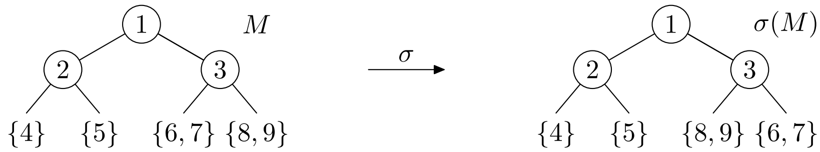

Figure 3.2: The sequence σ, which transposes the children of the P-node with the section $\{3\}$, is an automorphism since $\sigma(M) \cong M$. On the other hand, the transposition of the children of the root P-node is not an automorphism.

The kernel $\mathrm{Ker}(\Phi)$ consists of all automorphisms which fix the maximal cliques, i.e., automorphisms that permute the vertices only inside each twin class. It follows that $\mathrm{Ker}(\Phi)$ is isomorphic to a direct product of symmetric groups. The group $\mathrm{Im}(\Phi)$ almost describes $\mathrm{Aut}(X)$.

Two MPQ-trees M and M' are *isomorphic* if the underlying PQ-trees are equal and if there exists a permutation π of $V(X)$ which maps each section of M to the corresponding section of M'. In other words, M and M' are the same when ignoring the labels of the vertices in the sections. A sequence $\sigma \in \Sigma(M)$ is called an *automorphism* of M if $\sigma(M) \cong M$; see Figure 3.2. The automorphisms of M are closed under composition, so they form the automorphism group $\mathrm{Aut}(M) \le \Sigma(M)$.

Lemma 3.2. *For an MPQ-tree M, we have*

$$\mathrm{Aut}(M) = \mathrm{Im}(\Phi).$$

Proof. Suppose that $\pi \in \mathrm{Aut}(X)$. The sequence $\sigma = \Phi(\pi)$ transforms M into $\sigma(M)$. It follows that $\sigma(M) \cong M$ since $\sigma(M)$ can be obtained from M by permuting the vertices in the sections by π. So $\sigma \in \mathrm{Aut}(M)$ and $\mathrm{Im}(\Phi) \le \mathrm{Aut}(M)$.

On the other hand, suppose $\sigma \in \mathrm{Aut}(M)$. We know that $\sigma(M) \cong M$ and let π be a permutation of $V(X)$ from the definition of the isomorphism. Two vertices of $V(X)$ are adjacent if and only if they belong to the sections of M on a common path from the root. This property is preserved in $\sigma(M)$, so $\pi \in \mathrm{Aut}(X)$. Each maximal clique is the union of all sections on the path from the root to the leaf representing this clique. Therefore the maximal cliques are permuted by σ the same as by π. Thus $\Phi(\pi) = \sigma$ and $\mathrm{Aut}(M) \le \mathrm{Im}(\Phi)$. □

Lemma 3.3. *For an MPQ-tree M representing an interval graph X, we have*

$$\mathrm{Aut}(X) \cong \mathrm{Ker}(\Phi) \rtimes \mathrm{Aut}(M).$$

Proof. Let $\sigma \in \mathrm{Aut}(M)$. In the proof of Lemma 3.2, we show that every permutation π from the definition of $\sigma(M) \cong M$ is an automorphism of X mapped by Φ to σ. Now, we want to choose these permutations consistently for all elements of $\mathrm{Aut}(M)$ as follows. Suppose that $\mathrm{id} = \sigma_1, \sigma_2, \dots, \sigma_n$ are the elements of $\mathrm{Aut}(M)$. We want to find $\mathrm{id} = \pi_1, \pi_2, \dots, \pi_n$ such that $\Phi(\pi_i) = \sigma_i$ and if $\sigma_i\sigma_j = \sigma_k$, then $\pi_i\pi_j = \pi_k$. In other words, $H = \{\pi_1, \dots, \pi_n\}$ is a subgroup and $\Phi \restriction_H$ is an isomorphism between H and $\mathrm{Aut}(M) = \mathrm{Im}(\Phi)$.

Suppose that $\pi, \pi' \in \mathrm{Aut}(X)$ such that $\Phi(\pi) = \Phi(\pi')$. Then π and π' permute the maximal cliques the same and they can only act differently on twin vertices, i.e.,

$\pi\pi'^{-1} \in \mathrm{Ker}(\Phi)$. Suppose that C is a twin class, then $\pi(C) = \pi'(C)$ but they can map the vertices of C differently. To define $\pi_1, \ldots, \pi_n$, we need to define them on the vertices of the twin classes consistently. To do so, we arbitrarily order the vertices in each twin class. For each π_i, we know how it permutes the twin classes, suppose a twin class C is mapped to a twin class $\pi_i(C)$. Then we define π_i on the vertices of C in such a way that the orderings are preserved.

It is easy to see that the above definition of H is correct. Clearly, we have

$$\mathrm{Ker}(\Phi) \cap H = \{\mathrm{id}\}.$$

Therefore, we get $\mathrm{Aut}(X)$ as the internal semidirect product $\mathrm{Ker}(\Phi) \rtimes H \cong \mathrm{Ker}(\Phi) \rtimes \mathrm{Aut}(M)$. Our approach is similar to the proof of Theorem 2.3, and the external semidirect product can be constructed in the same way. □

3.2 Automorphism Groups of Interval Graphs

In this section, we give a characterization of $\mathrm{Aut}(\mathsf{INT})$ and prove Theorem 1.3(i). Let X be an interval graph, represented by an MPQ-tree M. By Lemma 3.3, we know that $\mathrm{Aut}(X)$ can be constructed from $\mathrm{Aut}(M)$ and $\mathrm{Ker}(\Phi)$. We build $\mathrm{Aut}(X)$ recursively using M, similarly as in Jordan's Theorem 2.4:

Lemma 3.4. *The groups in* $\mathrm{Aut}(\mathsf{INT})$ *are described inductively as follows:*

(a) $\{1\} \in \mathrm{Aut}(\mathsf{INT})$.
(b) If $G_1, G_2 \in \mathrm{Aut}(\mathsf{INT})$, *then* $G_1 \times G_2 \in \mathrm{Aut}(\mathsf{INT})$.
(c) If $G \in \mathrm{Aut}(\mathsf{INT})$ *and* $n \geq 2$, *then* $G \wr \mathbb{S}_n \in \mathrm{Aut}(\mathsf{INT})$.
(d) If $G_1, G_2, G_3 \in \mathrm{Aut}(\mathsf{INT})$ *and* $G_1 \cong G_3$, *then* $(G_1 \times G_2 \times G_3) \rtimes \mathbb{Z}_2 \in \mathrm{Aut}(\mathsf{INT})$

Proof. Clearly $\{1\} \in \mathrm{Aut}(\mathsf{INT})$. We show that $\mathrm{Aut}(\mathsf{INT})$ is closed under (b), (c) and (d); see Figure 3.3. For (b), we attach interval graphs X_1 and X_2 such that $\mathrm{Aut}(X_i) \cong G_i$ to an asymmetric interval graph. For (c), let $G \in \mathrm{Aut}(\mathsf{INT})$ and $n \geq 2$. There exists a connected interval graph Y such that $\mathrm{Aut}(Y) \cong G$. We construct X as the disjoint union of n copies of Y. By Theorem 2.3, we get $\mathrm{Aut}(X) \cong G \wr \mathbb{S}_n$. For (d), we construct an interval graph X by attaching X_1, X_2 and X_3 to a path, where $\mathrm{Aut}(X_i) = G_i$ and $X_1 \cong X_3$.

For the converse, let M be an MPQ-tree representing an interval graph X. Let $M_1, \ldots, M_k$ be the subtrees of the root of M and let X_i be the interval graphs induced by the vertices in the sections of M_i. We want to build $\mathrm{Aut}(X)$ from the automorphism groups of $X_1, \ldots, X_k$ using (b)–(d). By Lemma 3.3,

$$\mathrm{Aut}(X_i) \cong \mathrm{Ker}(\Phi_i) \rtimes \mathrm{Aut}(M_i)$$

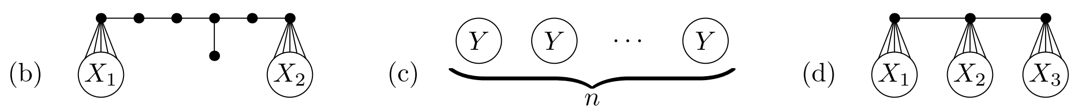

Figure 3.3: The constructions in the proof of Lemma 3.4.

which can be by the induction hypothesis constructed using operations (a)–(d).

Suppose that the root of M is a P-node P. Each sequence $\sigma \in \mathrm{Aut}(M)$ corresponds to an interior sequences in $\mathrm{Aut}(M_i)$ and some reordering σ' of the subtrees $M_1, \dots, M_k$. If $\sigma'(M_i) = M_j$, then necessarily $X_i \cong X_j$. On each isomorphism class of $X_1, \dots, X_k$, the permutations σ' behave to $\mathrm{Aut}(X_i)$ like the permutations τ to $\mathrm{Aut}(Y)$ in the proof of Theorem 2.3. Therefore the point-wise stabilizer of $\mathrm{sec}(P)$ in $\mathrm{Aut}(X)$ is constructed from $\mathrm{Aut}(X_1), \dots, \mathrm{Aut}(X_k)$ as in Theorem 2.3. Since every automorphim preserves $\mathrm{sec}(P)$, the group $\mathrm{Aut}(X)$ is obtained by the direct product of the above group with the symmetric group of order $|\mathrm{sec}(P)|$. Thus the operations (b)–(c) are sufficient.[1]

For a Q-node Q in the root, we construct the automorphism group similarly. We call Q *symmetric* if it is transformed by some sequence in $\mathrm{Aut}(M)$, and *asymmetric* otherwise. Let $M_1, \dots, M_k$ be the children of Q from left to right. If Q is asymmetric, then $\mathrm{Aut}(M)$ is the direct product $\mathrm{Aut}(X_1), \dots, \mathrm{Aut}(X_k)$ together with the symmetric groups for all twin classes of $\mathrm{sec}(Q)$. If Q is symmetric, we apply the operation (d) where G_1 corresponds to the direct product of the left part of the children and sections, G_2 to the middle part and G_3 to the right part. The semidirect product with $\mathbb{Z}_2$ adds the action of reversing Q. ☐

In the context of interval representations, the operation (b) applies to non-isomorphic independent parts of the representation, (c) to isomorphic parts which can be arbitrary permuted, and (d) to parts which can only be reflected vertically.

Proof of Theorem 1.3(i). To establish $\mathrm{Aut}(\mathsf{INT}) = \mathrm{Aut}(\mathsf{TREE})$, we use Lemma 3.4. We show that (d) can be expressed using (b) and (c). Assuming $G_1 \cong G_3$, we get

$$(G_1 \times G_2 \times G_3) \rtimes_\varphi \mathbb{Z}_2 \cong (G_1 \times G_3) \rtimes_\varphi \mathbb{Z}_2 \times G_2 \cong G_1 \wr \mathbb{Z}_2 \times G_2.$$

☐

3.3 Direct Constructions

In this section, we give an alternative proof of Theorem 1.3(i) by direct constructions. The proof of Lemma 3.5 answers the open problem of Hanlon [29]. Lemma 3.6 gives the converse construction.

Lemma 3.5. *For $X \in \mathsf{INT}$, there exists $T \in \mathsf{TREE}$ such that $\mathrm{Aut}(X) \cong \mathrm{Aut}(T)$.*

Proof. Consider an MPQ-tree M representing X. We know that $\mathrm{Aut}(X) \cong \mathrm{Ker}(\Phi) \rtimes \mathrm{Aut}(M)$. We inductively encode the structure of M into T.

Suppose a P-node P is the root of M. Then its subtrees can be encoded by trees and we just attach them to a common root. If $\mathrm{sec}(P)$ is non-empty, we attach a star with $|\mathrm{sec}(P)|$ leaves to the root (and we subdivide the edge of this star several times

[1] Alternatively, we can show that each X_i is connected and X is the disjoint union of $X_1, \dots, X_k$ together with $|\mathrm{sec}(P)|$ vertices attached to everything. So Theorem 2.3 directly applies.

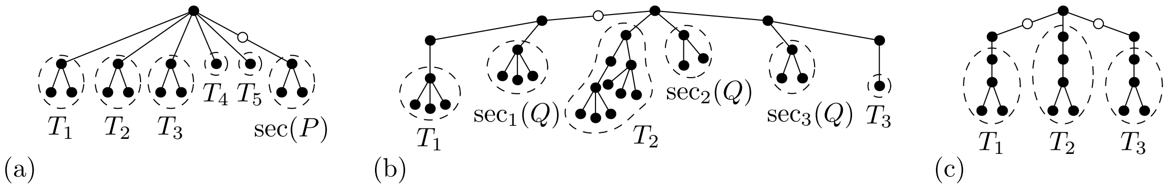

Figure 3.4: For an interval graph X, a construction of a tree T with $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$: (a) The root is a P-node. (b) The root is an asymmetric Q-node. (c) The root is a symmetric Q-node.

to make it non-isomorphic to every other subtree of the root); see Figure 3.4a. We get $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$.

Let a Q-node Q be in the root. If Q is asymmetric, we attach the trees corresponding to the subtrees of Q and stars corresponding to the vertices of the twin classes in the sections of Q to a path, and possibly modify by subdivisions to make it asymmetric; see Figure 3.4b. Finally, if Q is symmetric, then

$$\mathrm{Aut}(X) \cong (G_1 \times G_2 \times G_3) \rtimes \mathbb{Z}_2$$

and we just attach trees T_1, T_2 and T_3 such that $\mathrm{Aut}(T_i) \cong G_i$ to a path as in Figure 3.4c. In both cases, $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$. ☐

Lemma 3.6. *For* $T \in \mathsf{TREE}$, *there exists* $X \in \mathsf{INT}$ *such that* $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$.

Proof. For a rooted tree T, we construct an interval graph X such that $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$ as follows. The intervals are nested according to T as shown in Figure 3.5. Each interval is contained exactly in the intervals of its ancestors. If T contains a vertex with only one child, then $\mathrm{Aut}(T) < \mathrm{Aut}(X)$. This can be fixed by adding suitable asymmetric interval graphs, as in Figure 3.5. ☐

3.4 Unit Interval Graphs

We apply the characterization of Aut(INT) derived in Lemma 3.4 to show that the automorphism groups of connected unit interval graphs are the same as the automorphism groups of caterpillars (which form the intersection of INT and TREE).

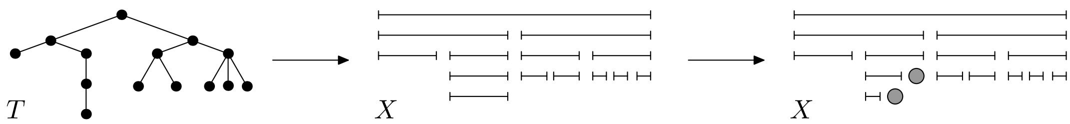

Figure 3.5: First, we place the intervals according to the structure of the tree. We get $\mathrm{Aut}(X) \cong \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_3$, but $\mathrm{Aut}(T) \cong \mathbb{S}_2 \times \mathbb{S}_3$. We fix this by adding copies of an asymmetric interval graph with the trivial automorphism group.

First, we describe Aut(CATERPILLAR). Let T be a caterpillar graph. We obtain the *central path* P be removing all leaves. We call T *symmetric* if some automorphism of T non-trivally swaps P, and it is *asymmetric* otherwise. Lemma 3.7 gives a characterization of Aut(CATERPILLAR).

Lemma 3.7. *Let T be a caterpillar graph with the central path P.*

(i) If T is asymmetric, then $\mathrm{Aut}(T)$ is a direct product of symmetric groups.
(ii) If T is symmetric, then

$$\mathrm{Aut}(T) \cong (G_1 \times G_2 \times G_3) \rtimes_\varphi \mathbb{Z}_2,$$

where G_2 is isomorphic to $\mathbb{S}_k$, $G_1 \cong G_3$ are direct products of symmetric groups, and $\varphi(1) = (g_1, g_2, g_3) \mapsto (g_3, g_2, g_1)$.

Proof. The root of an MPQ-tree M representing T is a Q-node Q (or a P-node with at most two children, which is trivial). All twin classes are trivial, since T is a tree. Each child of the root is either a P-node, or a leaf. All children of a P-node are leaves. Observe that T is symmetric if and only if Q is symmetric. We can determine $\mathrm{Aut}(X)$ as in the proof of Lemma 3.4. □

Proof of Theorem 1.3(ii). According to Corneil [8], an MPQ-tree M representing a connected unit interval graph contains only one Q-node with all children as leaves. It is possible that the sections of this Q-node are nontrivial. This equality of automorphism groups follows from Lemma 3.7 and the proof of Lemma 3.4. □

3.5 Groups Acting On Interval Representations

For an interval graph X, the set $\mathfrak{Rep}$ consists of all assignments of closed intervals which define X. It is natural to consider two interval representations equivalent if one can be transformed into the other by continuous shifting of the endpoints of the intervals while preserving the correctness of the representation. Then each representation of $\mathfrak{Rep}/\sim$ corresponds to a different consecutive ordering of the maximal cliques; see Figure 3.6 and 3.7.

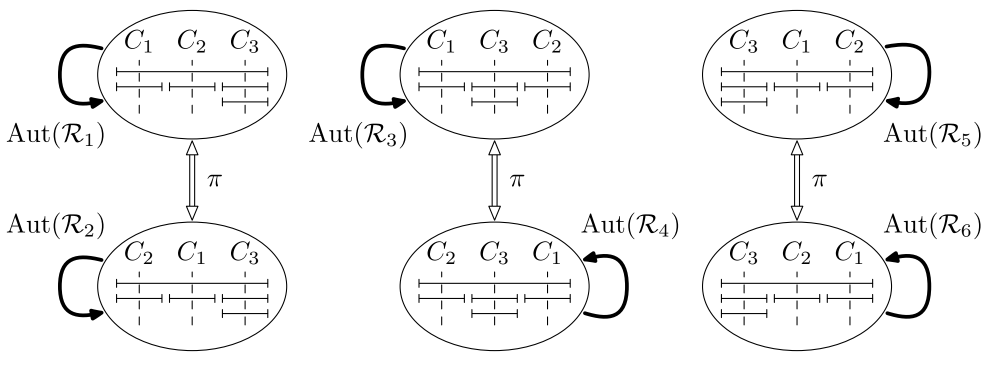

Figure 3.6: An interval graph with six non-equivalent representation. The action of $\mathrm{Aut}(X)$ has three isomorphic orbits.

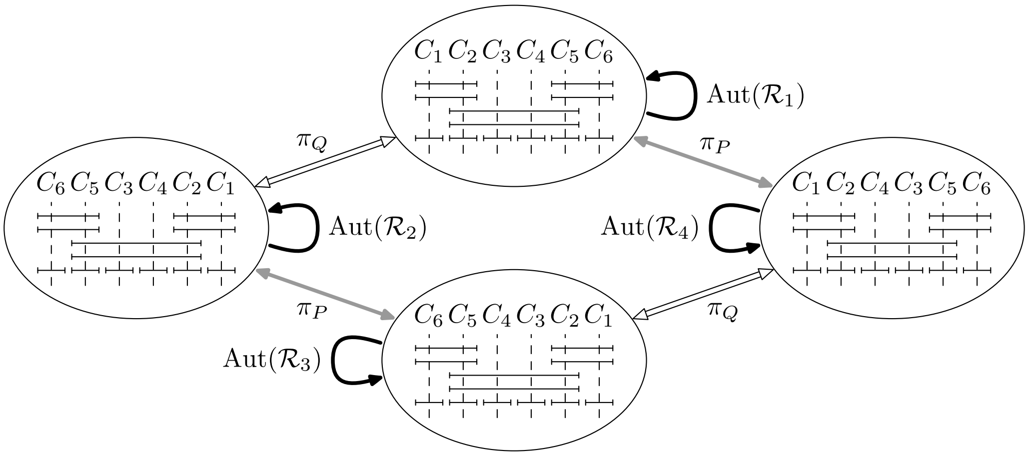

Figure 3.7: An interval graph with four non-equivalent representations. The action of $\mathrm{Aut}(X)$ is transitive. An MPQ-tree M of X is depicted in Fig. 3.1. Since there are three twin classes of size two, we have $\mathrm{Aut}(\mathcal{R}) \cong \mathbb{Z}_2^3$. The group $\mathrm{Aut}(M)$ is generated by two sequences: π_Q corresponding to flipping the Q-node, and π_P corresponding to permuting the P-node. We have $\mathrm{Aut}(M) \cong \mathbb{Z}_2^2$ and $\mathrm{Aut}(X) \cong \mathbb{Z}_2^3 \rtimes \mathbb{Z}_2^2$.

We interpret our results from the previous sections in terms of the action of $\mathrm{Aut}(X)$ on $\mathfrak{Rep}$. In Lemma 3.3, we proved that $\mathrm{Aut}(X) \cong \mathrm{Ker}(\Phi) \rtimes \mathrm{Aut}(M)$ where M is an MPQ-tree. If an automorphism belongs to $\mathrm{Aut}(\mathcal{R})$, then it fixes the ordering of the maximal cliques and it can only permute twin vertices. Therefore $\mathrm{Aut}(\mathcal{R}) = \mathrm{Ker}(\Phi)$ since each equivalence class of twin vertices consists of equal intervals, so they can be arbitrarily permuted without changing the representation. Every stabilizer $\mathrm{Aut}(\mathcal{R})$ is the same and every orbit of the action of $\mathrm{Aut}(X)$ is isomorphic, as in Figure 3.6.

Different orderings of the maximal cliques correspond to different reorderings of M. The defined $\mathrm{Aut}(M) \cong \mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$ describes morphisms of representations belonging to one orbit of the action of $\mathrm{Aut}(X)$, so these representations are the same up to the labeling of the intervals; see Figure 3.7.

4 Comparability Graphs

In this chapter, we prove Theorem 1.4 (Section 4.3) and Theorem 1.5 (Section 4.4). In Section 4.1, we introduce the modular tree representing a graph which we use in Section 4.2 and 4.3 to describe Aut(COMP) and Aut(PERM).

4.1 Modular Decomposition and Modular Tree

In this section, we introduce the modular decomposition of a graph X and show that it can be encoded by a modular tree. We further show that the automorphism group of this modular tree is isomorphic to $\mathrm{Aut}(X)$.

Modules. A *module* M of a graph X is a set of vertices such that each $x \in V(X)\setminus M$ is either adjacent to all vertices in M, or to none of them. Modules generalize connected components, but unlike connected components, one module can be a proper subset of another one. Therefore, modules lead to a recursive decomposition of a graph, instead of just a partition. See Figure 4.1a for examples. A module M is called *trivial* if $M = V(X)$ or $|M| = 1$, and *non-trivial* otherwise.

If M and M' are two disjoint modules, then either the edges between M and M' form the complete bipartite graph, or there are no edges at all; see Figure 4.1a. In the former case, M and M' are called *adjacent*, otherwise they are *non-adjacent*.

Quotient Graphs. Let $\mathcal{P} = \{M_1, \dots, M_k\}$ be a *modular partition* of $V(X)$, i.e., each M_i is a module of X, $M_i \cap M_j = \emptyset$ for every $i \neq j$, and $M_1 \cup \cdots \cup M_k = V(X)$. We define the *quotient graph* $X/\mathcal{P}$ with the vertices $m_1, \dots, m_k$ (which correspond to the modules $M_1, \dots, M_k$) where $m_i m_j \in E(X/\mathcal{P})$ if and only if M_i and M_j are adjacent.

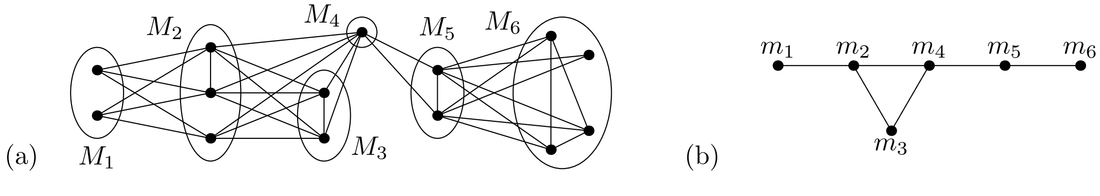

Figure 4.1: (a) A graph X with a modular partition $\mathcal{P}$ formed by its inclusion maximal non-trivial modules. (b) The quotient graph $X/\mathcal{P}$ is prime.

In other words, the quotient graph is obtained by contracting each module M_i into a single vertex m_i; see Figure 4.1b.

Modular Decomposition. We decompose a graph X by finding some modular partition $\mathcal{P} = \{M_1, \ldots, M_k\}$, computing $X/\mathcal{P}$ and recursively decomposing $X/\mathcal{P}$ and each $X[M_i]$. The recursive process stops on *prime graphs* (w.r.t. modular decomposition) which are graphs containing only trivial modules. There might be many such decompositions, depending on the choice of $\mathcal{P}$ in each step. In 1960s, Gallai [20] described the *modular decomposition* in which some special modular partitions are chosen. This modular decomposition encodes all possible decompositions.

The key is the following observation. Let M be a module of X and let $M' \subseteq M$. Then M' is a module of X if and only if it is a module of $X[M]$. We construct the modular decomposition $\mathfrak{MD}$ of a graph X in the following way:

- A graph X is called *degenerate* (w.r.t. modular decomposition) if it is K_n or $\overline{K}_n$. If X is a prime or a degenerate graph, then we add X to $\mathfrak{MD}$ and stop. We stop on degenerate graphs to make the modular decomposition unique; there are many modular partitions for them but not very interesting.

- Let X and $\overline{X}$ be connected graphs. Gallai [20] shows that the inclusion maximal non-trivial modules of X form a modular partition $\mathcal{P}$ of $V(X)$, and the quotient graph $X/\mathcal{P}$ is a prime graph; see Figure 4.1. We add $X/\mathcal{P}$ to $\mathfrak{MD}$ and recursively decompose $X[M]$ for each $M \in \mathcal{P}$.

- If X is disconnected and $\overline{X}$ is connected, then every union of several connected components is a module. All other modules are subsets of a single connected component. Therefore the connected components form a modular partition $\mathcal{P}$ of $V(X)$, and the quotient graph $X/\mathcal{P}$ is an independent set. We add $X/\mathcal{P}$ to $\mathfrak{MD}$ and recursively decompose $X[M]$ for each $M \in \mathcal{P}$.

- If $\overline{X}$ is disconnected and X is connected, then the modular decomposition is defined in the same way on the connected components of $\overline{X}$. They form a modular partition $\mathcal{P}$ and the quotient graph $X/\mathcal{P}$ is a complete graph. We add $X/\mathcal{P}$ to $\mathfrak{MD}$ and recursively decompose $X[M]$ for each $M \in \mathcal{P}$.

Gallai [20] shows that the modular decomposition of a graph is unique. It is easy to see that it captures all modules of X.

4.1.1 Modular Tree

Let $\mathfrak{MD}$ be the modular decomposition of X. We encode it by the *modular tree* T which is a graph with two types of vertices (normal and *marker* vertices) and two types of edges (normal and directed tree edges). The tree edges connect the prime and degenerate graphs obtained in $\mathfrak{MD}$ into a tree. Further every modular tree has an induced subgraph called *root node*.

If X is a prime or a degenerate graph, we define $T = X$ and its root node is equal T. Otherwise, let $\mathcal{P} = \{M_1, \ldots, M_k\}$ be the modular partition of X used in $\mathfrak{MD}$

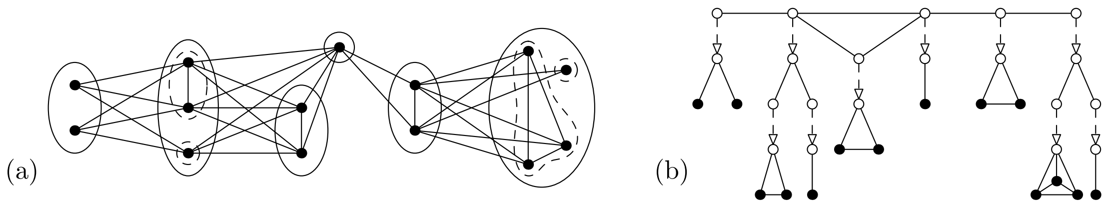

Figure 4.2: (a) The graph X from Figure 4.1 with the modular partition $\mathcal{P}$ of X is depicted, the subsequent modular partitions are depicted by dashed lines. (b) The modular tree T of X, the marker vertices are white, the tree edges are dashed.

and let $T_1, \ldots, T_k$ be the corresponding modular trees for $X[M_1], \ldots, X[M_k]$ according $\mathfrak{MD}$. The modular tree T is constructed by taking disjoint union of $T_1, \ldots, T_k$ and the quotient $X/\mathcal{P}$ with the marker vertices $m_1, \ldots, m_k$. To every graph T_i, we add a new marker vertex m'_i such that m'_i is adjacent exactly to the vertices of the root node of T_i. We further add a tree edge from m_i to m'_i. For an example, see Figure 4.2.

Since the modular decomposition of X is unique, also the modular tree of X is unique. The graphs obtained in $\mathfrak{MD}$ are called *nodes of* T, or alternatively root nodes of some modular tree in the construction of T. For a node N, its subtree is the modular tree which has N as the root node. Every node either has all vertices as marker vertices, or contains no marker vertices. In the former case, it is called an *inner node*, otherwise a *leaf node*.

The next lemma explains that T encodes adjacencies in X.

Lemma 4.1. *We have $xy \in E(X)$ if and only if there exists an alternating path $xm_1m_2 \ldots m_ky$ in the modular tree T such that each m_i is a marker vertex and precisely the edges $m_{2i-1}m_{2i}$ are tree edges.*

Proof. Suppose that $xy \in E(X)$. If $xy \in E(T)$, then we are done. We assume that $xy \notin E(T)$. The modular decomposition was constructed by a sequence of quotient operations. At some step of the construction we get the last graph X_0 such that $xy \in E(X_0)$. Let $\mathcal{P}$ be the modular partition of X_0 chosen by the modular decomposition. As in the construction of the modular tree, we denote the marker vertices obtained from the contraction of the modules by $m_1, \ldots, m_k$, and the marker vertices attached to those by tree edges by $m'_1, \ldots, m'_k$.

We consider the next step of the modular decomposition. Suppose that $x \in M_i$ and $y \in M_j$. We have that $x \in V(X_0[M_i])$ and $y \in V(X_0[M_j])$. From the construction of T, it follows that xm'_i and ym'_j are normal edges and since $xy \in E(X_0)$, we also have that $m_im_j \in E(X_0/\mathcal{P})$. The vertices $xm'_im_im_jm'_jy$ form an alternating path.

Now, we recursively construct an alternating path in T. From the construction of T, we have that the vertices x and m'_i are connected by a normal edge. Since the vertices x and m'_i are adjacent in the graph $X_0[M_i] \cup m'_i$, there exists an alternating path P_i connecting x and m'_i in the subtree of T representing $X_0[M_i] \cup m'_i$. Similarly, we have an alternating path P_j connecting y and m'_j in some subtree of T representing $X_0[M_j] \cup m'_j$. The vertices $xP_im'_im_im_jm'_jP_jy$ form a correct alternating path in T.

The converse implication can be easily derived by reversing the process described above. □

4.1.2 Automorphisms of Modular Trees

An automorphism of the modular tree T has to preserve the types of vertices and edges, i.e., map tree edges to tree edges, marker vertices to marker vertices, and fix the root. We denote the automorphism group of T by $\mathrm{Aut}(T)$.

Lemma 4.2. *If T is the modular tree representing a graph X, then*

$$\mathrm{Aut}(X) \cong \mathrm{Aut}(T).$$

Proof. First, we show that each automorphism $\sigma \in \mathrm{Aut}(T)$ induces a unique automorphism of X. We define $\alpha = \sigma \restriction_A$. By Lemma 4.1 two vertices $x, y \in V(X)$ are adjacent if and only if there exists and alternating path in T connecting them. Since σ is an automorphism, we also have an alternating path between $\sigma(x)$ and $\sigma(y)$. Therefore, $xy \in E(X) \iff \alpha(x)\alpha(y) \in E(X)$.

To obtain the converse implication, we prove that $\alpha \in \mathrm{Aut}(X)$ induces a unique automorphism $\sigma \in \mathrm{Aut}(T)$. We define $\sigma(x) = \alpha(x)$ for a non-marker vertex x. On the marker vertices, we define σ recursively as follows. Let $\mathcal{P} = \{M_1, \dots, M_k\}$ be a modular partition of X from the construction of the modular decomposition. It is easy to see that the group $\mathrm{Aut}(X)$ induces a action on the partition P. If $\alpha(M_i) = M_j$, then clearly $X[M_i]$ and $X[M_j]$ are isomorphic. We define $\sigma(m_i) = m_j$ and $\sigma(m_i') = m_j'$, and finish the rest recursively. Since σ is an automorphism at each step of the construction, it follows that $\sigma \in \mathrm{Aut}(T)$. □

Recursive Construction. We can build $\mathrm{Aut}(T)$ recursively. Suppose that we know automorphism groups $\mathrm{Aut}(T_1), \dots, \mathrm{Aut}(T_k)$ of all subtrees $T_1, \dots, T_k$ of T. Let R be the root node of T. We further color the marker vertices in R by the colors coding isomorphism classes of the subtrees $T_1, \dots, T_k$. Let $\mathrm{Aut}(R)$ be the color preserving automorphism group of R.

Lemma 4.3. *We have*

$$\mathrm{Aut}(T) \cong \big(\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k)\big) \rtimes \mathrm{Aut}(R).$$

Proof. We proceed similarly as in the proof of Theorem 2.3. We isomorphically label the vertices of the isomorphic subtrees T_i. Each automorphism $\pi \in \mathrm{Aut}(T)$ is a composition of two automorphisms $\sigma \cdot \tau$ where σ maps each subtree T_i to itself, and τ permutes the subtrees as in π while preserving the labeling. Therefore, the automorphisms σ can be bijectively identified with the elements of the direct product $\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k)$ and the automorphisms τ with some element of $\mathrm{Aut}(R)$.

Let $\pi, \pi' \in \mathrm{Aut}(T)$. Consider the composition $\sigma \cdot \tau \cdot \sigma' \cdot \tau'$, we want to swap τ with σ' and rewrite this as a composition $\sigma \cdot \hat\sigma \cdot \hat\tau \cdot \tau$. Clearly the subtrees are permuted in $\pi \cdot \pi'$ exactly as in $\tau \cdot \tau'$, so $\hat\tau = \tau$. On the other hand, $\hat\sigma$ is not necessarily equal σ'. Let σ' be identified with the vector

$$(\sigma_1', \dots, \sigma_k') \in \mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k).$$

Since σ' is applied after τ, it acts on the subtrees permuted according to τ. Thus, $\hat{\sigma}$ is constructed from σ by permuting the coordinates of its vector by τ:

$$\hat{\sigma} = (\sigma'_{\tau(1)}, \ldots, \sigma'_{\tau(k)}).$$

This is precisely the definition of the semidirect product. □

With no further assumptions on X, if R is a prime graph, then $\mathrm{Aut}(R)$ can be isomorphic to an arbitrary group. If R is a degenerate graph, then $\mathrm{Aut}(R)$ is a direct product of symmetric groups.

We note that this procedure does not lead to a polynomial-time algorithm for computing $\mathrm{Aut}(T)$. The reason is that the automorphism groups of prime graphs can be very complicated. To color the marker vertices, we have to be able to solve graph isomorphism of subtrees T_i, and then we have to find the subgroup of $\mathrm{Aut}(R)$ which preserves the colors.

4.2 Automorphism Groups of Comparability Graphs

In this section, we give a structural understanding of the automorphism groups of comparability graphs, in terms of actions on sets of transitive orientations.

Structure of Transitive Orientations. Let $\to$ be a transitive orientation of X and let T be the modular tree representing X. For modules M_1 and M_2, we write $M_1 \to M_2$ if $x_1 \to x_2$ for all $x_1 \in M_1$ and $x_2 \in M_2$. Gallai [20] shows:

- If two modules M_1 and M_2 are adjacent, then either $M_1 \to M_2$, or $M_1 \leftarrow M_2$.
- The graph X is a comparability graph if and only if each node of T is a comparability graph.
- Every prime comparability graph has exactly two transitive orientations, one being the reversal of the other.

The modular tree T encodes all transitive orientations as follows. For each prime node of T, we choose one of the two possible orientations. For each degenerate node, we choose some orientation. (If it is a complete graph K_n, it has $n!$ possible orientations, if it is an independent graph $\overline{K_n}$, it has the unique orientation). A transitive orientation of X is then constructed as follows. We orient the vertices of leaf nodes as above. For every subtree with children $M_1, \ldots, M_k$, we orient $X[M_i] \to X[M_j]$ if and only if $m_i \to m_j$ in the root node. It is easy to check that this gives a valid transitive orientation, and every transitive orientation can be constructed in this way.

Action Induced On Transitive Orientations. Let $\mathfrak{o}(X)$ be the set of all transitive orientations of X. Let $\pi \in \mathrm{Aut}(X)$ and $O \in \mathfrak{o}(X)$. We define the orientation $\pi(O)$:

$$xOy \iff \pi(x)\pi(O)\pi(y), \qquad \forall x, y \in V(X).$$

We can observe that $\pi(O)$ is a transitive orientation of X, so $\pi(O) \in \mathfrak{o}(X)$; see Figure 4.3. Therefore $\mathrm{Aut}(X)$ defines an action on $\mathfrak{o}(X)$.

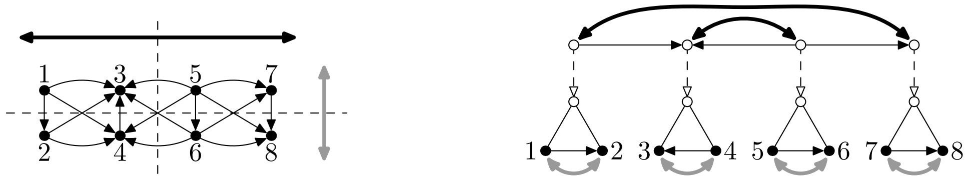

Figure 4.3: Two automorphism reflect X and change the transitive orientation. On the right, their action on the modular tree T.

Let S be the stabilizer of some orientation O. It consists of all automorphisms which preserve this orientation, so they permute only the vertices that are incomparable in O. In other words, S is the automorphism group of the poset created by the transitive orientation O of X. We want to understand it in terms of $\mathrm{Aut}(T)$ for the modular tree T representing X. Each automorphism $\mathrm{Aut}(T)$ somehow acts inside each node, and somehow permutes the nodes, as characterized in Lemma 4.3.

Consider some subtree of T with the subtrees $T_1, \dots, T_k$. Suppose that $\sigma \in S$ maps T_i to $\sigma(T_i) = T_j$. Then the marker vertices m_i and m_j have to be incomparable in the root node of the subtree T. If the root node is an independent set, the isomorphic subtrees can be arbitrarily permuted in S. If it is a complete graph, all subtrees are preserved in S. If it is a prime graph, then isomorphic subtrees of incomparable marker vertices can be permuted.

4.3 Automorphism Groups of Permutation Graphs

In this section, we derive the characterization of the automorphism groups of permutation graphs stated in Theorem 1.4.

Action Induced On Pairs of Transitive Orientations. Let X be a permutation graph. In comparison to general comparability graphs, here, the main difference is that both X and $\overline{X}$ are comparability graphs. From the results of Section 4.2 it follows that $\mathrm{Aut}(X)$ induces an action on both $\mathfrak{o}(X)$ and $\mathfrak{o}(\overline{X})$. We work with these two actions as with one action on the pair $(\mathfrak{o}(X), \mathfrak{o}(\overline{X}))$, in other words on pairs $(O, \overline{O})$ such that $O \in \mathfrak{o}(X)$ and $\overline{O} \in \mathfrak{o}(\overline{X})$. Figure 4.4 shows an example.

An action is called *semiregular* if only the identity has a fixed point. In other

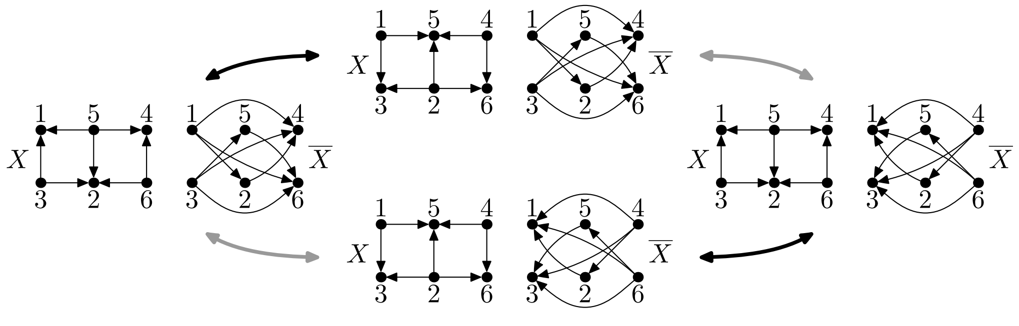

Figure 4.4: The action of $\mathrm{Aut}(X)$ on four pairs of transitive orientations X. The black generator flips the orientation of X, the gray automorphism of both X and $\overline{X}$.

words, all stabilizers of a semiregular action are trivial.

Lemma 4.4. *The action of* $\mathrm{Aut}(X)$ *on* $(\mathfrak{o}(X), \mathfrak{o}(\overline{X}))$ *is semiregular.*

Proof. We know that a permutation belonging to a stabilizer can only permute incomparable elements. Since incomparable elements in O are exactly the comparable elements in $\overline{O}$, the stabilizer is trivial. □

Lemma 4.5. *For a prime permutation graph* X*,* $\mathrm{Aut}(X)$ *is a subgroup of* $\mathbb{Z}_2 \times \mathbb{Z}_2$*.*

Proof. There are at most four pairs of orientations in $(\mathfrak{o}(X), \mathfrak{o}(\overline{X}))$, so by Lemma 4.4 the order of $\mathrm{Aut}(X)$ is at most four. If $\pi \in \mathrm{Aut}(X)$, then π^2 fixes the orientation of both X and $\overline{X}$. Therefore π^2 is an identity, π an involution and $\mathrm{Aut}(X)$ is a subgroup of $\mathbb{Z}_2 \times \mathbb{Z}_2$. □

Action Induced On Permutation Representations. We explain the result PERM = 2-DIM of Even [14]. Let $O \in \mathfrak{o}(X)$ and $\overline{O} \in \mathfrak{o}(\overline{X})$, and let $\overline{O}_R$ be the reversal of $\overline{O}$. We construct two linear orderings $L_1 = O \cup \overline{O}$ and $L_2 = O \cup \overline{O}_R$. The comparable pairs in $L_1 \cap L_2$ are precisely the edges $E(X)$.

Consider a permutation representation of a symmetric prime permutation graph. The horizontal reflection corresponds to exchanging L_1 and L_2, which is equivalent to reversing $\overline{O}$. The vertical reflection corresponds to reversing both L_1 and L_2, which is equivalent to reversing both O and $\overline{O}$. The central rotation by 180° is the combination of both, which is equivalent to reversing O; see Figure 4.5.

Now, we are ready to prove Theorem 1.4.

Proof of Theorem 1.4. Since $\{1\} \in \mathrm{Aut}(\mathsf{PERM})$, we need prove that $\mathrm{Aut}(\mathsf{PERM})$ is closed under (b)–(d).

- Let $G_1, G_2 \in \mathrm{Aut}(\mathsf{PERM})$, and let X_1 and X_2 be two permutation graphs such that $\mathrm{Aut}(X_1) \cong G_1$ and $\mathrm{Aut}(X_2) \cong G_2$. We construct a permutation graph X by attaching X_1 and X_2 to an asymmetric permutation graph; see Figure 4.6b. Clearly, we get $\mathrm{Aut}(X) \cong G_1 \times G_2$.

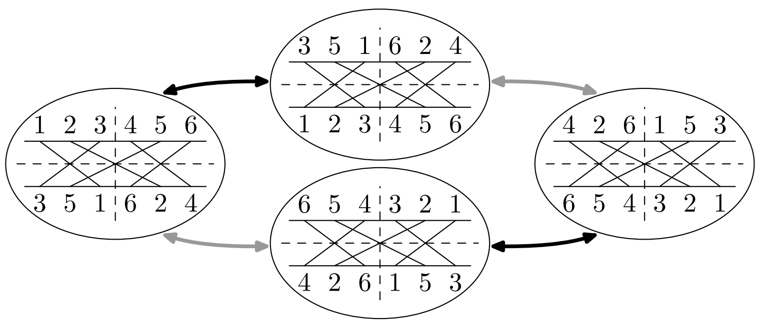

Figure 4.5: Four representations of a symmetric permutation graph. The black automorphism is the horizontal reflection with reverses O and the gray automorphism is the vertical reflection which reverses both O and $\overline{O}$.

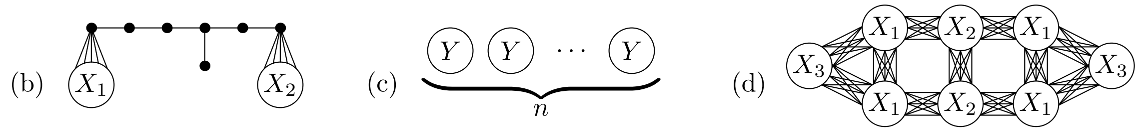

Figure 4.6: The construction of the operations (b)–(d). It is easy to check that they are permutation graphs with the correct automorphism groups.

- Let $G \in \mathrm{Aut}(\mathsf{PERM})$, and let Y be connected a permutation graph such that $\mathrm{Aut}(Y) \cong G$. We construct a graph X by taking the disjoint union of n copies of Y; see Figure 4.6. Clearly, we get $\mathrm{Aut}(X) \cong G \wr \mathbb{S}_n$.
- Let $G_1, G_2, G_3 \in \mathrm{Aut}(\mathsf{PERM})$, and let X_1, X_2, and X_3 be permutation graphs such that $\mathrm{Aut}(X_i) \cong G_i$, for $i = 1, 2, 3$. We construct a graph X as shown in Figure 4.6. Clearly, we get $\mathrm{Aut}(X) \cong \big(G_1^4 \times G_2^2 \times G_3^2\big) \rtimes \mathbb{Z}_2^2$.

To show that for a given permutation graph X the group $\mathrm{Aut}(X) \in \mathrm{Aut}(\mathsf{PERM})$ we use Lemma 4.3. Let T be the modular tree representing X, let R be its root, and let $T_1, \dots, T_k$ be the subtrees of R. By induction, we assume that $\mathrm{Aut}(T_i) \in \mathrm{Aut}(\mathsf{PERM})$, and we show that also $\mathrm{Aut}(T) \in \mathrm{Aut}(\mathsf{PERM})$. We distinguish two cases.

- If R is a degenerate node (an independent set or a complete graph), then $\mathrm{Aut}(R)$ is a direct product of symmetric groups. By Lemma 4.3, we get

  $$\mathrm{Aut}(T) \cong \big(\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k)\big) \rtimes \big(\mathbb{S}_{\ell_1} \times \cdots \times \mathbb{S}_{\ell_m}\big),$$

  where $\ell_1, \dots, \ell_m$ are the sizes of the isomorphism classes of $T_1, \dots, T_k$. Let G_i be the direct product of all $\mathrm{Aut}(T_j)$ such that T_j belong to the same isomorphism class i. We have

  $$\mathrm{Aut}(T) \cong G_1 \wr \mathbb{S}_{\ell_1} \times \cdots \times G_m \wr \mathbb{S}_{\ell_m}.$$

  Therefore $\mathrm{Aut}(X) \cong \mathrm{Aut}(T)$ can be constructed using (b) and (c) and it belongs to $\mathrm{Aut}(\mathsf{PERM})$.

- If R is a prime node, then by Lemma 4.5, $\mathrm{Aut}(R)$ is a subgroup of $\mathbb{Z}_2^2$. The only interesting case is when $\mathrm{Aut}(R) \cong \mathbb{Z}_2^2$. From the orbit-stabilizer theorem, the action of $\mathbb{Z}_2^2$ on $V(R)$ can have orbits of sizes 4, 2, and 1. Moreover, each orbit of size 2 corresponds to some stabilizer of size 2. Since there are three subgroups of $\mathbb{Z}_2^2$ of size 2, there can be possibly three types of orbits of size 2. By a geometric argument, we show that if R is a prime permutation graph, then one of the three subgroups of size 2 can not be a stabilizer of any orbit of size 2, and therefore there are at most two types of orbits of size 2.

  The non-identity elements $(1,0)$, $(0,1)$, and $(1,1)$ of $\mathbb{Z}_2^2$ correspond to the reflection f of the permutation representation along the vertical axis, reflection f' along the horizontal axis, and rotation r around the center by 180°, respectively; see Figure 4.5. The reflection f stabilizes only segments that that coincide with the vertical axis. Note that there can be at most one such segment, since otherwise R would not be prime. Therefore, the reflection f does not stabilize any orbit of size 2.

Let G_1 be the direct product of all $\mathrm{Aut}(T_j)$ such that T_j is attached to a vertex of R that belongs to an orbit of size four. The groups G_2 and G_3 are defined similarly for the two types of orbits of size two, and G_4 for the orbits of size one. We have

$$\mathrm{Aut}(T) \cong \big(G_1^4 \times G_2^2 \times G_3^2 \times G_1\big) \rtimes_\varphi \mathbb{Z}_2^2 \cong \big(G_1^4 \times G_2^2 \times G_3^2\big) \rtimes \mathbb{Z}_2^2 \times G_4,$$

where $\varphi\colon \mathbb{Z}_2^2 \to \mathrm{Aut}\big(G_1^4 \times G_2^2 \times G_3^2 \times G_1\big)$ is the homomorphism defined as follows. The automorphism $\varphi(1,0)$ swaps the first two components of G_1^4, swaps the components of of G_2^2, fixes the components of G_3^2, and fixes G_1. The automorphism $\varphi(0,1)$ swaps the second two components of G_1^4, fixes the components of G_2^2, swaps the components of G_3^2, and fixes G_1. We get that $\mathrm{Aut}(X) \cong \mathrm{Aut}(T)$ can be constructed using (b) and (d) and it belongs to Aut(PERM).

□

4.4 k-Dimensional Comparability Graphs

In this section, we prove that Aut(4-DIM) contains all finite groups, i.e., each finite group can be realised as an automorphism group of some 4-dimensional comparability graph. Our construction also shows that graph isomorphism testing of 4-DIM is GI-complete. Both results easily translate to k-DIM for $k > 4$ since 4-DIM $\subsetneq$ k-DIM.

The Construction. Let X be a graph with $V(X) = x_1, \dots, x_n$ and $E(X) = \{e_1, \dots, e_m\}$. We define

$$P = \big\{p_i : x_i \in V(X)\big\}, \qquad Q = \{q_{ik} : x_i \in e_k\}, \qquad R = \big\{r_k : e_k \in E(X)\big\},$$

where P represents the vertices, R represents the edges and Q represents the incidences between the vertices and the edges.

The constructed comparability graph C_X is defined as follows, see Figure 4.7:

$$V(C_X) = P \cup Q \cup R, \qquad E(C_X) = \{p_i q_{ik}, q_{ik} r_k : x_i \in e_k\}.$$

Lemma 4.6. *Let X be a connected graph such that $X \not\cong C_n$. Then*

$$\mathrm{Aut}(C_X) \cong \mathrm{Aut}(X).$$

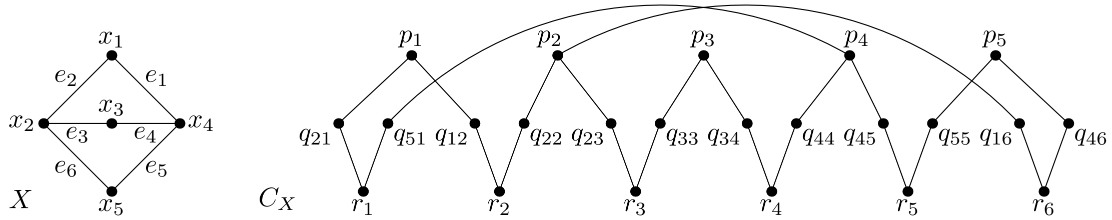

Figure 4.7: The construction C_X for the graph $X = K_{2,3}$.

Proof. All vertices of Q and R have degree two, and by our assumption at least one vertex p_i in P has a different degree. Therefore, we obtain P as the set of the vertices in C_X whose distance from p_i is divisible by four, Q as the set of their neighbors and R as the remaining vertices. Every automorphism of C_X has to preserve this partition, therefore it induces an automorphism of X. Since this construction does not depend on the labeling, every automorphism of X is an automorphism of C_X. We have that $\mathrm{Aut}(C_X) \cong \mathrm{Aut}(X)$. □

Proof of Dimension 4. The harder part is to prove that the constructed graph C_X has dimension four, which we can do when X is bipartite.

Lemma 4.7. *If X is a connected bipartite graph, then* $\dim(C_X) \leq 4$.

Proof. We construct four chains such that $L_1 \cap L_2 \cap L_3 \cap L_4$ have two vertices comparable if and only if they are adjacent in C_X. We describe linear chains as words containing each vertex of $V(C_X)$ exactly once. If $S_1, \dots, S_s$ is a sequence of strings, the symbol $\langle S_t : \uparrow t\rangle$ is the concatenation $S_1 S_2 \dots S_s$ and $\langle S_t : \downarrow t\rangle$ is the concatenation $S_s S_{s-1} \dots S_1$. When the arrows are omitted as in $\langle S_t\rangle$, we concatenate in an arbitrary order.

First, we define the *incidence string* I_i which codes p_i and its neighbors q_{ik}:

$$I_i = p_i \big\langle q_{ik} : p_i q_{ik} \in E(C_X)\big\rangle.$$

Notice that the concatenation $I_i I_j$ contains the right edges but it further contains edges going from p_i and q_{ik} to p_j and $q_{j\ell}$. We remove these edges by concatenation $I_j I_i$ in some other chain.

Since X is bipartite, let (A, B) be partition of its vertices. We define

$$\begin{aligned} P_A &= \{p_i : x_i \in A\}, \qquad Q_A = \{q_{ik} : x_i \in A\},\\ P_B &= \{p_j : x_j \in B\}, \qquad Q_B = \{q_{jk} : x_j \in B\}. \end{aligned}$$

Each vertex r_k has exactly one neighbor in Q_A and exactly one in Q_B.

We construct the four chains as follows:

$$\begin{aligned} L_1 &= \langle p_i : p_i \in P_A\rangle\langle r_k q_{ik} : q_{ik} \in Q_A, \uparrow k\rangle\langle I_i : p_i \in P_B, \uparrow i\rangle,\\ L_2 &= \langle p_i : p_i \in P_A\rangle\langle r_k q_{ik} : q_{ik} \in Q_A, \downarrow k\rangle\langle I_i : p_i \in P_B, \downarrow i\rangle,\\ L_3 &= \langle p_j : p_j \in P_B\rangle\langle r_k q_{jk} : q_{jk} \in Q_B, \uparrow k\rangle\langle I_i : p_i \in P_A, \uparrow i\rangle,\\ L_4 &= \langle p_j : p_j \in P_B\rangle\langle r_k q_{jk} : q_{jk} \in Q_B, \downarrow k\rangle\langle I_i : p_i \in P_A, \downarrow i\rangle. \end{aligned}$$

See Figure 4.8 for properties of $L_1, \dots, L_4$. It is routine to verify that the intersection $L_1 \cap L_2 \cap L_3 \cap L_4$ is correct.

The four defined chains have the following properties, see Figure 4.8:

- The intersection $L_1 \cap L_2$ forces the correct edges between Q_A and R and between P_B and Q_B. It poses no restrictions between Q_B and R and between P_A and the rest of the graph.

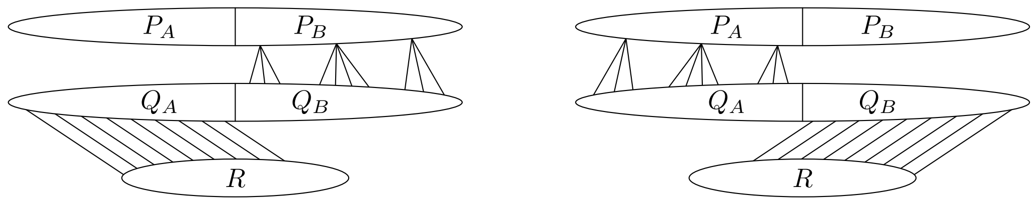

Figure 4.8: On the left, the forced edges in $L_1 \cap L_2$, on the right in $L_3 \cap L_4$.

- Similarly the intersection $L_3 \cap L_4$ forces the correct edges between Q_B and R and between P_A and Q_A. It poses no restrictions between Q_A and R and between P_B and the rest of the graph.

Claim 1: The edges in $Q \cup R$ are correct. For every k, we get r_k adjacent to both q_{ik} and q_{jk} since it appear on the left in $L_1, \dots, L_4$. On the other hand, $q_{ik}q_{jk} \notin E(C_X)$ since they are ordered differently in L_1 and L_3.

For every $k < \ell$, there are no edges between $N[r_k] = \{r_k, q_{ik}, q_{jk}\}$ and $N[r_\ell] = \{r_\ell, q_{s\ell}, q_{t\ell}\}$. This can be shown by checking the four ordering of these six elements:

$$\text{in } L_1\text{:}\quad r_k q_{ik} \boxed{r_\ell q_{s\ell}}\, q_{jk} \boxed{q_{t\ell}}, \qquad \text{in } L_2\text{:}\quad \boxed{r_\ell q_{s\ell}}\, r_k q_{ik} q_{jk} \boxed{q_{t\ell}},$$
$$\text{in } L_3\text{:}\quad r_k q_{jk} \boxed{r_\ell q_{t\ell}}\, q_{ik} \boxed{q_{s\ell}}, \qquad \text{in } L_4\text{:}\quad \boxed{r_\ell q_{t\ell}}\, r_k q_{jk} q_{ik} \boxed{q_{s\ell}},$$

where the elements of $N[r_\ell]$ are boxed. ◇

Claim 2: The edges in P are correct. We show that there are no edges between p_i and p_j for $i \neq j$ as follows. If both belong to P_A (respectively P_B), then they are ordered differently in L_3 and L_4 (respectively L_1 and L_2). If one belongs to P_A and the other one to P_B, then they are ordered differently in L_1 and L_3. ◇

Claim 3: The edges between P and $Q \cup R$ are correct. For every $p_i \in P$ and $r_k \in R$, we have $p_i r_k \notin E(C_X)$ because they are ordered differently in L_1 and L_3. On the other hand, $p_i q_{ik} \in E(C_X)$, because p_i is before q_{ik} in I_i, and for $p_i \in P_A$ in L_1 and L_2, and for $p_i \in B$ in L_3 and L_4.

It remains to show that $p_i q_{jk} \notin E(C_X)$ for $i \neq j$. If both p_i and p_j belong to P_A (respectively P_B), then p_i and q_{jk} are ordered differently in L_3 and L_4 (respectively L_1 and L_2). And if one belongs to P_A and the other one to P_B, then p_i and q_{jk} are ordered differently in L_1 and L_3. ◇

These three claims show that comparable pairs in the intersection $L_1 \cap L_2 \cap L_3 \cap L_4$ are exactly the edges of C_X, so C_X is a comparability graph with the dimension at most four. □

Finally, we prove Theorem 1.5.

Proof of Theorem 1.5. It is sufficient to prove the statement for 4-DIM. Let X be a connected graph such that $X \not\cong C_n$. First, we take the bipartite incidence graph Y between $V(X)$ and $E(X)$, and it easily follows that $\mathrm{Aut}(Y) \cong \mathrm{Aut}(X)$. Then we construct C_Y. From Lemma 4.6 it follows that $\mathrm{Aut}(C_Y) \cong \mathrm{Aut}(Y) \cong \mathrm{Aut}(X)$ and by Lemma 4.7, we have that $C_Y \in$ 4-DIM. Similarly, if two graphs X_1 and X_2 are given, we construct C_{Y_1} and C_{Y_2} such that $X_1 \cong X_2$ if and only if $C_{Y_1} \cong C_{Y_2}$; this gives the reduction which shows GI-completeness of graph isomorphism testing. □

5 Circle Graphs

Here, we study Aut(CIRCLE). Section 5.1 introduces the split tree of a graph. Split tree captures the symmetries of a graph and in case of circle graphs, it can by used to describe Aut(CIRCLE) as in Theorem 1.6. This is proved in Section 5.2. Finally, in Section 5.3 we describe the action of the automorphism group of a circle graph on the set of its circle representations.

5.1 Split Decomposition and Split Tree

First, we explain a split decomposition of a graph. Similarly to the modular decomposition, it a recursive process that decomposes a graph into indecomposable graphs. A split decomposition starts by finding a split.

A *split* of X is a partition of the vertices of X into four parts A, B, A' and B' such that:

- For every $a \in A$ and $b \in B$, we have $ab \in E(X)$.
- There are no edges between A' and $B \cup B'$, and between B' and $A \cup A'$.
- Both sides have at least two vertices: $|A \cup A'| \geq 2$ and $|B \cup B'| \geq 2$.

A split decomposition of X is a collection of graphs constructed as follows. Initially, it constains only X. Then we take a split (A, A', B, B') of X and replace X by graphs X_A and X_B defined as follows. The vertex set of X_A is the set $V(X_A) = A \cup A' \cup \{m_A\}$, where m_A is a new *marker vertex* adjacent exactly to the vertices in A. The graph X_B is defined analogously for B, B' and m_B; see Figure 5.1a. We proceed recursively for X_A and X_B.

Graphs containing no splits are called *prime graphs*. We stop the construction of a split decomposition also on *degenerate graphs* which are the complete graphs K_n and the complete bipartite graphs $K_{1,n}$. A split decomposition of a graph is therefore a collection of prime and degenerate graphs. Note that the prime and degenerate graphs with respect to a split decomposition are different as to modular decomposition (see Section 4.1).

A split decomposition does not have to be unique, however, Cunningham [9] proved that the minimal split decomposition is. A split decomposition $\mathfrak{D}$ is *minimal* if every other split decomposition is constructed by at least as many splits as $\mathfrak{D}$.

It is not difficult to see that a connected graph X is a circle graph if and only if both X_A and X_B are circle graphs. This is the key connection between the split decomposition and circle graphs. In other words, a connected graph X is a circle graph if and only if all the prime graphs obtained by the minimal split decomposition are circle graphs (the degenerate graphs are clearly circle graphs).

Notice that a representation of a circle graph X is completely determined by the circular order of the endpoints of the chords in its representation. Two chords C_x and C_y, corresponding to some vertices $x, y \in V(X)$, intersect if and only if their endpoints alternate in this circular order. According to [19], every *prime circle graph* has a unique circle representation up to rotations and reflections of the circular order of the endpoints of the chords representing it. In other words, the circular order of the endpoints of the chords representing a prime circle graph is unique, up to rotations and reflections.

5.1.1 Split Tree

The *split tree* S representing a graph X encodes the minimal split decomposition of X. A split tree, similarly as modular tree, is a graph with two types of vertices (normal and marker vertices) and two types of edges (normal and tree edges). We initially put $S = X$ and modify it according to the minimal split decomposition. If the minimal split decomposition takes a split (A, B, A', B') of Y, then we replace Y by the graphs Y_A and Y_B, and connect the marker vertices m_A and m_B by a *tree-edge* (see Figure 5.1a). We repeat this recursively on Y_A and Y_B; see Figure 5.1b.

Each prime or degenerate graph is a *node* of the split tree. Since the minimal split decomposition of a graph is unique, we also have that the split tree of a graph is unique. In the next lemma, we prove that the split tree S of a graph X captures the adjacencies in X.

Lemma 5.1. *We have $xy \in E(X)$ if and only if there exists an alternating path $xm_1m_2 \dots m_ky$ in S such that each m_i is a marker vertex and precisely the edges $m_{2i-1}m_{2i}$ are tree edges.*

Proof. Suppose that $xy \in E(X)$. We prove the existence of an alternating path between x and y by induction. If $xy \in E(S)$, then it clearly exists. Otherwise, let

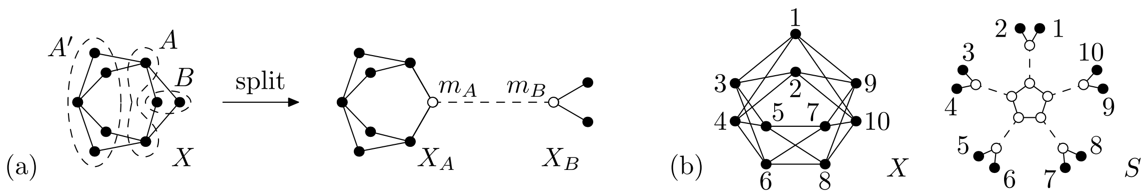

Figure 5.1: (a) An example of a split of the graph X. The marker vertices are depicted in white. The tree edge is depicted by a dashed line. (b) The split tree S of the graph X. We have that $\mathrm{Aut}(S) \cong \mathbb{Z}_2^5 \rtimes \mathbb{D}_5$.

Y be the last graph in the construction of the minimal split decomposition such that $xy \in E(Y)$. The decomposition splits Y according to the split (A, B, A', B') such that $x \in A$ and $y \in B$, otherwise Y_A or Y_B would contain xy. We have $x \in V(Y_A)$, $xm_A \in E(Y_A)$, $y \in V(Y_B)$, and $ym_B \in E(Y_B)$. By induction hypothesis, there exist alternating paths between x and m_A and between m_B and y in S. There is a tree edge m_Am_B, so we get an alternating path between x and y. On the other hand, if there exists an alternating path $xm_1 \dots m_ky$ in S, by joining all splits, we get $xy \in E(X)$. □

5.1.2 Automorphisms of Split Trees

In [23], split-trees are defined in terms of graph-labeled trees. However, our definition is more suitable for working with automorphism groups. An *automorphism of a split-tree* S is an automorphism of S which preserves the types of vertices and edges, i.e, it maps marker vertices to marker vertices, and tree-edges to tree edges. We denote the automorphism group of S by $\mathrm{Aut}(S)$.

Lemma 5.2. *If S is a split-tree representing a graph X, then* $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$.

Proof. First, we show that each $\sigma \in \mathrm{Aut}(S)$ induces a unique automorphism α of X. Since $V(X) \subseteq V(S)$, we define $\alpha = \sigma \restriction_{V(X)}$. By Lemma 5.1, $xy \in E(X)$ are adjacent if and only if there exists an alternating path between them in S. Automorphisms preserve alternating paths, so $xy \in E(X) \iff \alpha(x)\alpha(y) \in E(X)$.

It remains to show that $\alpha \in \mathrm{Aut}(X)$ induces a unique automorphism $\sigma \in \mathrm{Aut}(S)$. We define $\sigma(x) = \alpha(x)$ for every non-marker vertex x. On the marker vertices, we define σ recursively as follows. Let (A, B, A', B') be a split of X chosen by the minimal split decomposition. This split is mapped by α to another split (C, D, C', D'), i.e.,

$$\alpha(A) = C, \quad \alpha(A') = C', \quad \alpha(B) = D, \quad \text{and} \quad \alpha(B') = D'.$$

By splitting according to (A, A', B, B'), we get the graphs X_A and X_B with the marker vertices $m_A \in V(X_A)$ and $m_B \in V(X_B)$. Similarly, for (C, C', D, D') we get X_C, X_D with $m_C \in V(X_C)$ and $m_D \in V(X_D)$. Since α is an automorphism, we have that $X_A \cong X_C$ and $X_B \cong X_D$. It follows that the unique split trees of X_A and X_C are isomorphic, and similarly for X_B and X_D. Therefore, we define $\sigma(m_A) = m_C$ and $\sigma(m_B) = m_D$, and we finish the rest recursively. Since σ is an automorphism at each step of the construction of S, it follows that $\sigma \in \mathrm{Aut}(S)$. □

Recursive Construction. Let S be as split tree representing a circle graph X, and let R be an arbitrary node of S. Suppose that we root the split tree S by R. Let S_R be the resulting *rooted* split tree. Note that the group $\mathrm{Aut}(S_R)$ is exactly the stabilizer $\mathrm{Stab}_S(R)$ and in general it is not isomorphic to $\mathrm{Aut}(S)$. However, it can be constructed similarly as the automorphism group of a modular tree; see Section 4.1.

Lemma 5.3. *Let $S_1, \dots, S_k$ be the subtrees of R, and let $m_i \in V(S_i)$ be the corresponding marker vertices, for $i = 1, \dots, k$. Then*

$$\mathrm{Aut}(S_R) \cong \big(\mathrm{Stab}_{S_1}(m_1) \times \cdots \times \mathrm{Stab}_{S_k}(m_k)\big) \rtimes \mathrm{Aut}(R).$$

Proof. Analogous to the proof of Lemma 4.3. □

5.2 Automorphism Groups of Circle Graphs

We prove Theorem 1.6 which gives a characterization of Aut(connected CIRCLE). The automorphism group of a disconnected circle graph can be easily determined using Theorem 2.3. We use split trees, described in Section 5.1, similarly as modular trees were used in Section 4.3.

To prove Theorem 1.6, we proceed similarly as in the proof of Jordan's characterization 2.4 of trees. By Lemma 5.2, it suffices to determine the automorphism group of the split tree S representing a connected circle graph X.

Similarly as for trees, there exists a *center* of S. If the center is a tree edge, we can construct a new split tree S' by subdividing the tree edge with K_2. We do this by creating two new marker vertices, and connecting them by a normal edge; see Figure 5.2. It is easy to see that $\mathrm{Aut}(S) \cong \mathrm{Aut}(S')$. Therefore, we can assume that the center C of S is a prime or a degenerate node. Each automorphism of S fixes C. We have

$$\mathrm{Stab}_S(C) = \mathrm{Aut}(S).$$

Therefore, we can also assume that S is rooted by C.

The following lemma is analogous to Lemma 4.5. For a prime circle graph X (with respect to the split decomposition) and a vertex $m \in V(X)$, it determines the group $\mathrm{Stab}_X(m)$. The automorphism group of a prime circle graph can be any subgroup of a dihedral group, since according to [19], each prime circle graph has a unique representation, up to rotation and reflection. However, in a rooted split tree, each automorphism has to stabilize one vertex of a prime node that is not the root. Therefore, Lemma 5.4 is relevant.

Lemma 5.4. *Let X be a connected circle graph and let $m \in V(X)$. If X is prime, then $\mathrm{Stab}_X(m)$ is isomorphic to a subgroup of $\mathbb{Z}_2 \times \mathbb{Z}_2$.*

Proof. Each prime circle graph has a unique circle representation, up to rotations and reflections of the circular order of the endpoints of the chords in the representation [19]. Therefore, every automorphism of X is a rotation or a reflection of the circular order.

Let the circular ordering of the endpoints of the chords be

$$mA\hat{m}B,$$

where m and $\hat{m}$ are the endpoints of the chord C_m representing m, and A and B are strings of the endpoints of the chords obtained by traversing the circle counterclockwise from m to $\hat{m}$ and from $\hat{m}$ to m, respectively.

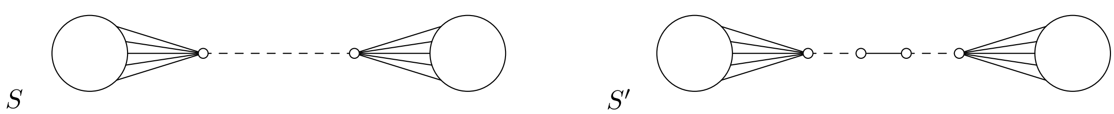

Figure 5.2: The center of the split tree S is a tree edge. We get S' by subdividing the tree edge with a graph isomorphic to K_2.

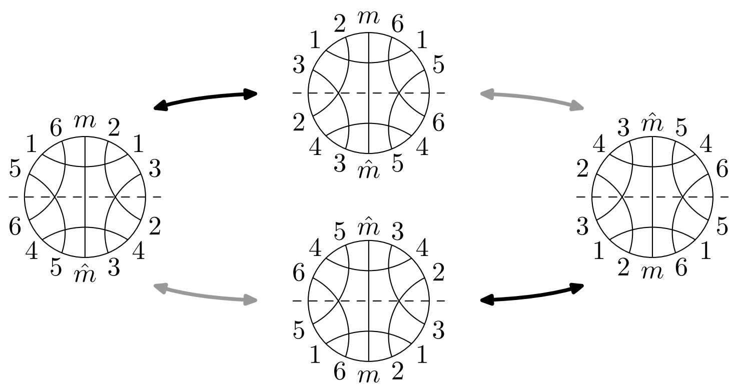

Figure 5.3: A geometrical interpretation of Lemma 5.4. The transformations f (black), and f' (gray) generate a group isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

Each automorphism of $\mathrm{Aut}(X)$ that stabilizes m either fixes both m and $\hat{m}$, or swaps them. The first can be only achieved with the reflection f of $mA\hat{m}B$ along m (and the identity) which gives $mB'\hat{m}A'$, where A' and B' are reversed A and B, respectively. The second can be only achieved with the rotation r of $mA\hat{m}B$ which gives $\hat{m}BmA$, and with the reflection f' which gives $\hat{m}A'mB'$.

It is easy to see that $f' = r \cdot f$. The transformations r and f of $mA\hat{m}B$ are involutions and therefore $\langle r, f\rangle \cong \mathbb{Z}_2 \times \mathbb{Z}_2$. The stabilizer of m is a subgroup of $\mathbb{Z}_2 \times \mathbb{Z}_2$ since r and f do not have to define an automorphism of X. ☐

Geometrically, the transformations f and f', from the proof of Lemma 5.4, correspond to the reflection of the representation along the chord C_m and to the reflection along the line perpendicular to the chord C_m, respectively. The rotation $r = f' \cdot f$ corresponds to the rotation of the representation around the center of the circle by 180°; see Figure 5.3.

Our next goal is to relate the class $\mathcal{S}$, defined in Theorem 1.6, to some suitable subclass of Aut(connected CIRCLE). We define

$$\mathrm{Stab}(\text{connected } \mathsf{CIRCLE}) = \big\{G \cong \mathrm{Stab}_X(x) : X \in \mathsf{CIRCLE}, x \in V(X)\big\}.$$

Lemma 5.5. *The class* Stab(connected CIRCLE) *contains exactly the same groups as the class $\mathcal{S}$, defined in Theorem 1.6.*

Proof. First, we construct the groups described in (a)–(d). We need to prove that Stab(connected CIRCLE) is closed under (b)–(d).

- Let $G_1, G_2 \in$ Stab(connected CIRCLE), and let X_1 and X_2 be circle graphs such that $\mathrm{Stab}_{X_1}(x_1) \cong G_1$ and $\mathrm{Stab}_{X_2}(x_2) \cong G_2$, for some $x_1 \in V(X_1)$ and $x_2 \in V(X_2)$. We construct a circle graph X by adding a new vertex x and attaching each of the vertices x_1 and x_2 to x. We possibly need to subdivide one of the edges x_1x and x_2x tho enforce $\mathrm{Stab}_X(x) \cong G_1 \times G_2$.
- Let $G \in$ Stab(connected CIRCLE), and let Y be a circle graph with $\mathrm{Stab}_Y(y) \cong G$, for some $y \in V(Y)$. We take and n copies of Y and construct a circle graph

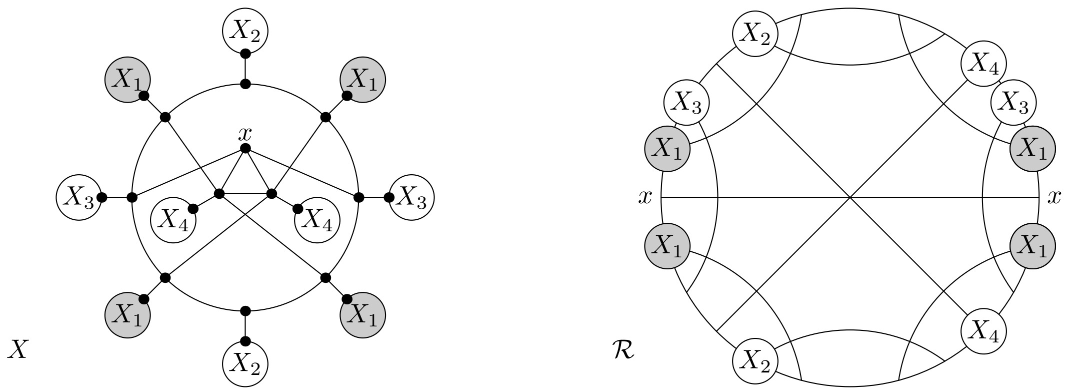

Figure 5.4: The construction of the group in (d) and a circle representation of X.

X by adding a new vertex x and an edge xy, for each copy of Y. Clearly, we get $\mathrm{Stab}_X(x) \cong G \wr \mathbb{S}_n$.

- Let $G_1, G_2, G_3, G_4 \in \mathrm{Stab}(\text{connected } \textsf{CIRCLE})$, and let X_i be a circle graph with $\mathrm{Stab}_{X_i}(x_i) \cong G_i$, for some $x_i \in V(X_i)$. We construct a graph X as shown in Figure 5.4. We get $\mathrm{Stab}_X(x) \cong \big(G_1^4 \times G_2^2 \times G_3^2 \times G_4^2\big) \rtimes \mathbb{Z}_2^2$.

It is not difficult to see that in all of the cases, the graph X is a circle graph.

It remains to show that for a circle graph X and a vertex $x \in V(X)$ the group $\mathrm{Stab}_X(x)$ is isomorphic to a group from $\mathrm{Stab}(\text{connected } \textsf{CIRCLE})$. Let S be the split tree of X. Since $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$, it suffices to prove it for $\mathrm{Stab}_S(x)$. Let R be the node of S such that $x \in V(R)$. We root S by R, and denote the resulting rooted split tree by S_R.

For each node of S_R, we need to determine how its children can be permuted. For a node N of S_R, this is given by $\mathrm{Stab}_N(y)$, where $y \in V(N)$. If $N \neq R$, then y is the marker vertex attached to the parent of N in S_R, and if $N = R$, then $y = x$, since the vertex x has to be stabilized.

We determine $\mathrm{Stab}_S(x)$ inductively, using Lemma 5.3. Let $S_1, \dots, S_k$ be the subtrees of the root N, and let $m_i \in V(S_i)$ be the corresponding marker vertices. By induction, we assume that each $\mathrm{Stab}_{S_N}(y) \in \mathrm{Stab}(\text{connected } \textsf{CIRCLE})$. It follows from Lemma 5.3 that

$$\mathrm{Stab}_{S_N}(y) \cong \big(\mathrm{Stab}_{S_1}(m_1) \times \cdots \times \mathrm{Stab}_{S_k}(m_k)\big) \rtimes \mathrm{Stab}_N(y).$$

Note that $\mathrm{Stab}_{S_R}(y) = \mathrm{Stab}_{S_R}(x) = \mathrm{Stab}_S(x)$. We distinguish two cases.

- If N is a degenerate node, then $\mathrm{Stab}_N(y)$ is a direct product of symmetric groups. We have

$$\mathrm{Stab}_{S_N}(y) \cong \big(\mathrm{Stab}_{S_1}(m_1) \times \cdots \times \mathrm{Stab}_{S_k}(m_k)\big) \rtimes \big(\mathbb{S}_{\ell_1} \times \cdots \times \mathbb{S}_{\ell_m}\big),$$

where $\ell_1, \dots, \ell_m$ are the sizes of orbits of the marker vertices m_i. Let G_i be the direct product of all $\mathrm{Stab}_{S_j}(m_j)$ such that m_j is in orbit i. We have

$$\mathrm{Stab}_{S_N}(y) \cong G_1 \wr \mathbb{S}_{\ell_1} \times \cdots \times G_m \wr \mathbb{S}_{\ell_m}.$$

Therefore $\mathrm{Stab}_{S_N}(y)$ can be constructed using (b) and (c) and it belongs to Stab(connected CIRCLE).

- If N is a prime node, then by Lemma 5.4, $\mathrm{Stab}_{S_N}(y)$ is a subgroup of $\mathbb{Z}_2^2$. The only interesting case is when $\mathrm{Stab}_{S_N}(y) \cong \mathbb{Z}_2^2$. The action of $\mathbb{Z}_2^2$ on $V(N)$ can have orbits of sizes 4, 2, and 1. Moreover, each orbit of size 2 corresponds to some stabilizer of size 2. Since there are three subgroups of $\mathbb{Z}_2^2$ of size 2, there can be possibly three types of orbits of size 2. Note that Figure 5.4 shows that unlike for permutation graphs (see proof of Theorem 1.4), the action of $\mathbb{Z}_2^2$ can have three orbits of size 2, for circle graphs.

  Let G_1 be the direct product of all $\mathrm{Stab}_{S_j}(m_j)$ such that m_j is belongs to an orbit of size four. The groups G_2, G_3, and G_4 are defined similarly for the three types of orbits of size two, and G_5 for the orbits of size one. Similarly as in the proof of Theorem 1.4, we get

$$\mathrm{Stab}_{S_N}(y) \cong \left(G_1^4 \times G_2^2 \times G_3^2 \times G_4^2\right) \rtimes_\varphi \mathbb{Z}_2^2 \times G_5,$$

  where $\varphi\colon \mathbb{Z}_2^2 \to \mathrm{Aut}\left(G_1^4 \times G_2^2 \times G_3^2 \times G_4^2\right)$ is the homomorphism defined similarly as in the proof of Theorem 1.4. Therefore $\mathrm{Stab}_{S_N}(y)$ can be constructed using (b) and (d) and it belongs to Stab(connected CIRCLE).

□

Now, we prove Theorem 1.6.

Proof of Theorem 1.6. Since $\{1\} \in$ Aut(connected CIRCLE), we need to prove that Aut(connected CIRCLE) is closed under (e)–(f).

- Let $G \in \mathcal{S}$, let $n \geq 3$, and let Y be a circle graph with $\mathrm{Stab}_Y(y) \cong G$, for some $y \in V(Y)$. We take and n copies of Y and construct a circle graph X as shown in Figure 5.5e. Clearly, we get $\mathrm{Aut}(x) \cong G^n \rtimes \mathbb{Z}_n$.
- Let $G_1, G_2 \in \mathcal{S}$, let $n \geq 3$, and let X_1 and X_2 be circle graphs such that $\mathrm{Stab}_{X_1}(x_1) \cong G_1$ and $\mathrm{Stab}_{X_2}(x_2) \cong G_2$, for some $x_1 \in V(X_1)$ and $x_2 \in V(X_2)$. We construct a circle graph X as shown in Figure 5.5f. We get $\mathrm{Aut}(X) \cong (G_1^n \times G_2^{2n}) \rtimes \mathbb{D}_n$.

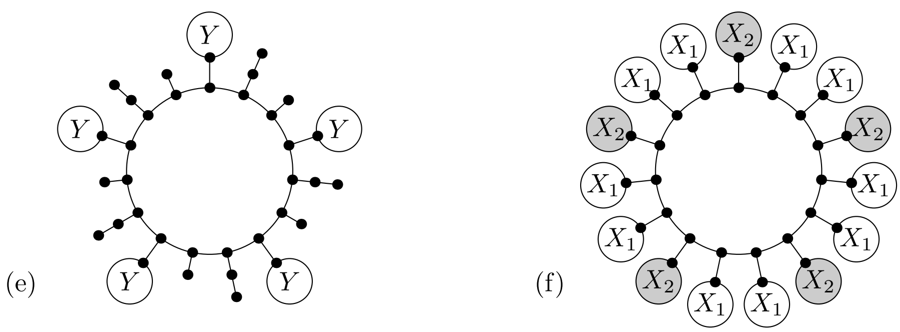

Figure 5.5: The construction of the operations (e)–(f). It is easy to check that they are circle graphs with the correct automorphism groups.

It is not difficult to see that in all of the cases, the graph X is a circle graph.

On the other hand, let X be a connected circle graph, and let S be its split tree. We show that $\mathrm{Aut}(S) \in \mathrm{Aut}(\text{connected } \mathsf{CIRCLE})$. We root S by its central node C, and let S_C be the resulting rooted split tree. Clearly, $\mathrm{Stab}_S(C) = \mathrm{Aut}(S_C) = \mathrm{Aut}(S)$. Let $S_1, \dots, S_k$ be the subtrees of C, and let $m_i \in V(S_i)$ be the corresponding marker vertices, for $i = 1, \dots, k$. Then by Lemma 5.3, we have

$$\mathrm{Aut}(S_C) \cong \big(\mathrm{Stab}_{S_1}(m_1) \times \cdots \times \mathrm{Stab}_{S_k}(m_k)\big) \rtimes \mathrm{Aut}(C).$$

According to [19], the circular ordering of the endpoints of the chords in the representation of a prime circle graph is unique, up to rotations and reflections. Therefore, each automorphism of a prime circle graph can be either a rotation, or a reflection of this circular ordering. Therefore, the group $\mathrm{Aut}(C)$ is a subgroup of a dihedral group, i.e., either $\mathrm{Aut}(C) \cong \mathbb{Z}_k$, or $\mathrm{Aut}(C) \cong \mathbb{D}_\ell$, for $k \neq 2$ and $\ell \geq 3$. From Lemma 5.5 it follows that

$$\mathrm{Stab}_{S_1}(m_1) \times \cdots \times \mathrm{Stab}_{S_k}(m_k) \in \mathrm{Stab}(\text{connected } \mathsf{CIRCLE}),$$

and therefore the group $\mathrm{Aut}(S_C)$ can be constructed using (e)–(f). Note that for $\mathrm{Aut}(C) \cong \mathbb{Z}_2$, $\mathbb{D}_1$, and $\mathbb{D}_2$ the group $\mathrm{Aut}(S_C)$ belongs to the class $\mathcal{S}$. $\square$

5.3 Groups Acting On Circle Representations

For a circle graph X, the set $\mathfrak{Rep}$ consists of all assignments of chords of a circle which define X. Two representations are equivalent if they have the same circular order of endpoints of the chords, up to reflections. Therefore each $\mathrm{Aut}(\mathcal{R})$ is a subgroup of a dihedral group. Different orbits of the action of $\mathrm{Aut}(X)$ may be non-isomorphic and $\mathrm{Aut}(\mathcal{R})$ may not be a normal subgroup of $\mathrm{Aut}(X)$.

The results of the previous sections have the following interpretation in terms of the action of $\mathrm{Aut}(X)$. Lemma 5.2 shows for the split tree S representing X that $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$. Assume that the center C is a prime circle graph, otherwise $\mathrm{Aut}(\mathcal{R})$ is much more restricted. We choose a representation $\mathcal{R}$ belonging to the smallest orbit, i.e., $\mathcal{R}$ is one of the most symmetrical representations. Then $\mathrm{Aut}(\mathcal{R})$ describes the rotations/reflections of C. Let H be the point-wise stabilizer of C in $\mathrm{Aut}(S)$. We know that H is generated by permutations of isomorphic subtrees attached to nodes $N \neq C$. If N is a prime graph, we can only apply the geometric reflection with the axis perpendicular to the chord of the marker vertex, which corresponds to reflecting a small part of a circle representation. If N is a degenerate graph, then isomorphic subtrees can be arbitrarily permuted which corresponds to permuting small identical parts of a circle representation. The proof of Theorem 1.6 shows that $\mathrm{Aut}(S) \cong H \rtimes \mathrm{Aut}(\mathcal{R})$.

6 Conclusions

We conclude this thesis with several open problems concerning intersection-defined classes of graphs.

As already mentioned in the introduction, Frucht [17] proved that each finite group G can be realized as an automorphism group of some graph X. It is known that an analogous result holds for the endomorphism monoids. An *endomorphism* of a graph X is a mapping $f\colon V(X) \to V(X)$ that preserves adjacencies, i.e., it is a graph homomorphism of X to itself. Endomorphisms of a graph X form a monoid, denoted by $\mathrm{End}(X)$. It can be shown that every finite monoid can be realized as an endomorphism monoid of X.

In Theorem 1.3, we proved that $\mathrm{Aut}(\mathsf{INT}) = \mathrm{Aut}(\mathsf{TREE})$. For a class $\mathcal{C}$ of graphs, we define

$$\mathrm{End}(\mathcal{C}) = \big\{M : M \text{ is an abstract monoid}, \exists X \in \mathcal{C} \text{ such that } G \cong \mathrm{End}(X)\big\}.$$

Similarly as $\mathrm{Aut}(\mathcal{C})$, the class $\mathrm{End}(\mathcal{C})$ contains all monoids that can be realized as an endomorphism monoid of some graph $X \in \mathcal{C}$.

Problem 6.1. *Does* $\mathrm{End}(\mathsf{TREE})$ *equal* $\mathrm{End}(\mathsf{INT})$*?*

Circular-arc graphs ($\mathsf{CIRCULAR\text{-}ARC}$) are intersection graphs of circular arcs and they naturally generalize interval graphs; see Figure 6.1. Surprisingly, this class is very complex and quite different from interval graphs. Hsu [32] gives a linear-time recognition algorithm for circular-arc, and relates them to circle graphs.

Problem 6.2. *What groups belong to* $\mathrm{Aut}(\mathsf{CIRCULAR\text{-}ARC})$*?*

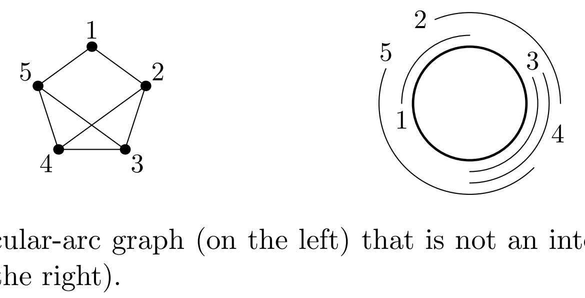

Figure 6.1: A circular-arc graph (on the left) that is not an interval graph, and its representation (on the right).

Let Y be a fixed graph. The class Y-GRAPH consists of the intersections graphs of connected subgraphs of a subdivision of Y. Observe that K_2-GRAPH = INT and

$$\bigcup_{T \in \mathsf{TREE}} T\text{-}\mathsf{GRAPH} = \mathsf{CHOR}.$$

We have an infinite hierarchy between INT and CHOR, since INT $\subseteq T$-GRAPH $\subsetneq$ CHOR. If Y contains a cycle of length at least four, then Y-GRAPH $\not\subseteq$ CHOR. The simplest of these classes are circular-arc graphs which are equal to K_3-GRAPH.

Conjecture 6.3. *For every fixed graph Y, the class Y-GRAPH is non-universal.*

The last class from the infinite hierarchy between 2-DIM and COMP whose automorphism groups remain unknown is 3-DIM.

Conjecture 6.4. *The class 3-DIM is universal.*

Bibliography

[1] L. Babai. Automorphism groups of planar graphs II. In *Infinite and finite sets (Proc. Conf. Keszthely, Hungary)*, 1973.

[2] K. A. Baker, P. C. Fishburn, and F. S. Roberts. Partial orders of dimension 2. *Networks*, 2:11–28, 1972.

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

[5] N. Carter. *Visual group theory.* MAA, 2009.

[6] C. J. Colbourn. On testing isomorphism of permutation graphs. *Networks*, 11(1):13–21, 1981.

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

[8] Derek G Corneil, Hiryoung Kim, Sridhar Natarajan, Stephan Olariu, and Alan P Sprague. Simple linear time recognition of unit interval graphs. *Information Processing Letters*, 55(2):99–104, 1995.

[9] W.H. Cunningham. Decomposition of directed graphs. *SIAM Journal on Algebraic Discrete Methods*, 3:214–228, 1982.

[10] D. S. Dummit and R. M. Foote. *Abstract algebra.* 2004.

[11] B. Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63(3):600–610, 1941.

[12] P. Erdös, S. Fajtlowicz, and A. J. Hoffman. Maximum degree in graphs of diameter 2. *Networks*, 10(1):87–90, 1980.

[13] S. Even and A. Itai. Queues, stacks and graphs. *Theory of Machines and Computations*, pages 71–86, 1971.

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

[15] J. Fiala, P. Klavík, J. Kratochvíl, and R. Nedela. Algorithmic aspects of regular graphs covers with applications to planar graphs. In *Lecture Notes in Computer Science, Automata, Languages, and Programming ICALP 2014*, volume 8572, pages 489–501, 2014.

[16] J. Fiala, P. Klavík, J. Kratochvíl, and R. Nedela. 3-connected reduction for regular graph covers. *arXiv:1503.06556*, 2015.

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

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

[19] C. P. Gabor, K. J. Supowit, and W.-L. Hsu. Recognizing circle graphs in polynomial time. *Journal of the ACM (JACM)*, 36(3):435–473, 1989.

[20] Tibor Gallai. Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18(1):25–66, 1967.

[21] Michael R Garey, David S Johnson, Gary L Miller, and Christos H Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic Discrete Methods*, 1(2):216–227, 1980.

[22] Fanica Gavril. Maximum weight independent sets and cliques in intersection graphs of filaments. *Information Processing Letters*, 73(5):181–188, 2000.

[23] E. Gioan and C. Paul. Split decomposition and graph-labelled trees: Characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Appl. Math.*, 160(6):708–733, 2012.

[24] C. D. Godsil and G. Royle. *Algebraic graph theory*, volume 207. Springer New York, 2001.

[25] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.

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

[27] M. Grohe and D. Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 173–192, 2012.

[28] G. Hajós. Über eine Art von Graphen. *Internationale Mathematische Nachrichten*, 11:65, 1957.

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

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

[31] A. J. Hoffman and R. R. Singleton. On moore graphs with diameters 2 and 3. *IBM Journal of Research and Development*, 4(5):497–504, 1960.

[32] W. L. Hsu. $\mathcal{O}(M \cdot N)$ algorithms for the recognition and isomorphism problems on circular-arc graphs. *SIAM Journal on Computing*, 24(3):411–439, 1995.

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

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

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

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

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

[38] B. D. McKay, M. Miller, and J. Širáň. A note on large graphs of diameter two and given maximum degree. *J. Combin. Theory Ser. B*, 74(1):110–118, 1998.

[39] M. Miller and J. Širáň. Moore graphs and beyond: A survey of the degree/diameter problem. *Electronic Journal of Combinatorics*, 61:1–63, 2005.

[40] F. S. Roberts. Discrete mathematical models, with applications to social, biological, and environmental problems. 1976.

[41] Fred S Roberts. Indifference graphs. *Proof techniques in graph theory*, 139:146, 1969.

[42] J. J. Rotman. *An introduction to the theory of groups*, volume 148. Springer, 1995.

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

[44] J. Šiagiová. A note on the McKay–Miller–Širáň graphs. *J. Combin. Theory Ser. B*, 81(2):205–208, 2001.

[45] J. P. Spinrad. *Efficient Graph Representations.: The Fields Institute for Research in Mathematical Sciences.*, volume 19. American Mathematical Soc., 2003.

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

[47] H. Whitney. Nonseparable and planar graphs. *Trans. Amer. Math. Soc.*, 34:339–362, 1932.

[48] M. Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic Discrete Methods*, 3(3):351–358, 1982.

[49] S. Zhou. A class of arc-transitive Cayley graphs as models for interconnection networks. *SIAM Journal on Discrete Mathematics*, 23(2):694–714, 2009.