FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Peter Zeman

Groups of automorphisms of graphs

Department of Applied Mathematics

Supervisor of the doctoral thesis: prof. RNDr. Roman Nedela, DrSc.
Consultant: prof. RNDr. Jan Kratochvíl, CSc.
Study programme: Computer Science
Study branch: Discrete Models and Algorithms

Prague 2021

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to thank my supervisor, Roman Nedela, for guiding me through my Ph.D. studies. Further, I would also like to thank my consultant, Jan Kratochvíl, for a lot of scientific and career advise. In particular, thanks to him I was able to spend a semester at the University of Oregon with Eugene Luks. Special thanks go to my coauthors, in particular, Ilia Ponomarenko, Bojan Mohar and Ken-ichi Kawarabayashi. Last but not least, I would like to thank the Department of Applied Mathematics and Computer Science Institute for providing a perfect environment.

Title: Groups of automorphisms of graphs

Author: Peter Zeman

Department: Department of Applied Mathematics

Supervisor: prof. RNDr. Roman Nedela, DrSc., Department of Applied Mathematics – external specialist

Abstract: In this thesis we investigate automorphism groups of several restricted classes of graphs from structural and computational point of view. For interval, permutation, circle, and planar graphs we give inductive characterizations of their automorphism groups in terms of group products. For chordal graphs of bounded leafage, prove that computing the automorphism group, and consequently the isomorphism problem, is fixed parameter tractable. For maps on surfaces, we give a linear time algorithm computing the automorphism group, parametrized by the genus of the underlying surface.

Keywords: automorphism groups, isomorphism problem, intersection graphs, planar graphs, maps on surfaces.

Contents

Introduction

A symmetry of an object is a transformation of the object which leaves it visually the same. The study of symmetries is an ancient topic in mathematics and its precise formulation led to the birth group theory. The symmetries of a mathematical object, viewed as mapping from the object to itself, form a structure: compositions of symmetries are symmetries and symmetries are invertible. A structure with these properties is known as a group. The group of all symmetries of an object X is known as its automorphism group $\mathrm{Aut}(X)$. Automorphisms of an object preserve all mathematical properties of the object. A closely related notion is that of an isomorphism, which is a mapping from one object to another that preserves all mathematical properties of the objects.

In computer science, we are mainly interested in the symmetries of discrete structures. Graphs (networks) are a fundamental and widely used model in computer science and many other disciplines. A graph consists of a finite set of vertices (nodes) and a set of edges, which are just unordered pairs of vertices representing relations. Existence of an efficient algorithm deciding whether or not two finite algebraic or combinatorial structures are isomorphic is a long-standing unsolved question in the theory of computing. Since all such structures can be canonically encoded by efficiently (polynomial-time) computable graphs [89, 127] in a well-defined sense, graphs play a key role.

The graph isomorphism problem is the computational problem which ask to decide whether two graphs are isomorphic. The graph automorphism problem asks to find a generating set of the automorphism group of a given graph, and, in fact, from the computational point of view it is equivalent to the graph isomorphism problem.

The graph isomorphism problem has also a practical importance. Originally, it arose in 1950s in chemistry while building chemical information systems [138]. For molecules, described by graphs, we want to test if a given molecule is already present in the database. Graphs are used in many fields to encode some sort of structural information and the graph isomorphism problem is natural when we want to match such graphs to see whether they are "the same". Another interesting application is malware detection, where static program analysis is done by matching fragments of code [165]. A different type of applications of the graph isomorphism problem could be described as symmetry breaking. Symmetries of instances of algorithmic problems can be used to simplify the instances or to prune search trees of backtracking algorithms. This has applications, for example, in SAT-solving [7] or (integer) linear programming [123].

Currently, the best upper bound for the complexity of the graph isomorphism problem comes from the quasipolynomial time algorithm due to Babai [11]. Moreover, it is believed that it is not NP-complete since otherwise the polynomial-time hierarchy would collapse to its second level [22]. The graph isomorphism problem is polynomial-time equivalent to other important problems [125]: counting the number of isomorphisms between two graphs, computing the order of the automorphism group of a graph, computing the generators of the automorphism group of a graph and to the problem of computing the orbits of the automorphism group of a graph. Moreover, it is polynomial-time equivalent to other, seemingly unrelated, problems in computational group theory [120] such as for instance to computing the generators of a stabilizer of a set, computing the intersection of two groups, computing the generators of the centralizer of an element in a given group.

By fixing natural parameters, it is possible to obtain efficient parametrized algorithms for various restricted classes of graphs. These include graphs of bounded degree [119, 83], graphs of bounded eigenvalue multiplicity [13], graphs of bounded treewidth [116, 84], graphs of bounded rank-width [84], and graphs of bounded genus [128, 134]. Moreover, for several important restricted classes of graphs, polynomial-time algorithms are available, for instance trees, interval graphs [42], permutation graphs [41], planar graphs [95, 100], to list just a few of them. On the other hand, for some graph classes the graph isomorphism problem is GI-complete, i.e., as hard as the general graph isomorphism problem. These include bipartite graphs, chordal graphs, comparability graphs, function graphs, regular graphs, and others.

Graph canonization is a related problem of finding a canonical form of a given graph. A canonical form assigns a graph $C(X)$ to any graph X such that $C(X)$ is isomorphic to X, and for graphs X and Y, the canonical forms $C(X)$ and $C(Y)$ are identical if and only if the graphs X and Y are isomorphic. It is worth mentioning that in practice, a canonization algorithm is often preferable to an isomorphism test, as it allows each graph to be treated separately, rather than having to compare graphs pairwise. Clearly, the graph canonization problem is computationally at least as hard as the graph isomorphism problem. Recently, Babai proved [12] that the graph canonization problem can be solved in quasipolynomial time as well. It is still an important open question, whether the graph isomorphism problem and the graph cannonization problem are polynomial-time equivalent.

A possible approach towards canonization is to try to find a complete set of invariants of a given graph. In many areas of mathematics, invariants and in particular functorial invariants play are important. For instance, if we wish to distinguish two vector spaces from each other, with respect to the isomorphism relation, it suffices to look at their dimensions. A more sophisticated invariant is, for example, the Euler characteristic of a topological space. This is not strong enough to distinguish all topological spaces, but for orientable surfaces it is sufficient.

The k-dimensional Weisfeiler-Leman algorithm (k-dim WL) attempts to construct a full set of invariants for graphs by iteratively coloring k-tuples of vertices and ensuring that certain regularity conditions are satisfied. For instance, in its simplest form, $k = 1$, the algorithm corresponds to the naive degree refinement: initially, vertices are coloured according to the their degrees and then the colouring is refined until each vertex has the same number of neighbours in each colour class.

The classical WL algorithm [162], or k-dim WL with $k = 2$, iteratively refines an initial coloring of ordered pairs of vertices (u, v) until the following regularity condition is satisfied: for any pair (u, v) of fixed color a, the number of paths of length 2 from u to v, such that the first step is of color b and the second step is of color c, is constant. This partitions the entries of the adjacency matrix of the graph into several color classes, each of which corresponds to a basis element of a certain accosiated matrix algebra. This can be computed in polynomial time and the isomorphism of these special matrix algebras can also be tested in polynomial time. However, this is sufficient to solve the graph isomorphism problem only in some special cases. More generally, for every positive integer k, the k-dim WL algorithm colors all k-tuples of vertices and iteratively refines the color classes based on the information from the previously obtained coloring. The k-dim WL algorithm associates to a graph a certain algebra of k-tensors.

The k-dim WL algorithm has surprising connections to seemingly unrelated areas. Two graphs are indistinguishable by k-dim WL if and only if they satisfy the same sentences with at most k variables in a certain extension of the first-order logic [29]. Further, two graphs X and Y are indistinguishable by k-dim WL if and only if for every graph Z of treewidth at most k, the number of homomorphisms from Z to X is the same as from Z to Y [58]. The most surprising connections are related to integer and semi-definite programming. By considering the Sherali-Adams hierarchy over a certain integer linear program, a characterization of distinguishability by k-dim WL is obtained [6]. In a similar way k-distinguishability can be related to the sum-of-squares hierarchy of semi-definite relaxations of the integer linear program [135].

Grohe [80] proved a remarkable result: all graph classes excluding a fixed graph as a minor have bounded WL-dimension, which is the minimum k such that all graphs in the class can be distinguished by the k-dim WL. However, this does not give any explicit bounds on the WL-dimension of the particular graph class. For example for planar graphs, bounded genus graphs, and bounded treewidth graphs, some lower and upper bounds can be determined more explicitly [103, 82, 101]. This also raises the question of what is the WL-dimension of classes of graphs that are not closed under taking minors, which is not covered by Grohe's result. For example, for interval graphs, WL-dimension can be bounded by 2 [60].

Results and organization of this thesis. In this thesis, we study automorphism groups of graphs from structural and computational point of view and also the related isomorphism problem. In Chapter 1, we introduce necessary notions from group theory, which are used in several subsequent chapters. Notation and concepts specific to particular chapters are introduced within the chapters. The subsequent chapters can be split into two parts: in Chapters 2–4 we focus on geometric intersection graphs and in Chapters 5–6 we focus on graphs and maps on surfaces.

In Chapter 2, we give a characterization of automorphism groups of automorphism groups of interval graphs, permutation graphs and circle graphs. For comparability graphs of dimension at most four, we prove that automorphism groups are universal. This chapter is based on the papers [107, 109]. In Chapter 3, we give an almost linear time algorithm for testing isomorphism of circle graphs. This chapter is based on the paper [98]. In Chapter 4, we prove that computing the automorphism group, and consequently the isomorphism problem, of a chordal graph of bounded leafage is fixed parameter tractable. We also prove that for certain related classes of graphs, the isomorphism problem is as hard as the general graph isomorphism problem. This chapter is based on the papers [5, 36]

In Chapter 5, give an inductive characterization of automorphism groups in terms of groups products, similarly as for the previously mentioned intersection classes of graphs. This chapter is based on the paper [108] Finally, in Chapter 6, we give a linear-time algorithm for computing automorphism groups of maps on surfaces, parametrized by the genus of the underlying surface. This chapter is based on the papers [99, 100].

1. Preliminaries: elements of group theory

We introduce the notation and concepts from group theory, which we use repeatedly in several subsequent chapters. We use the following notation for some standard families of groups:

- $\mathbb{S}_n$ is the *symmetric group* of all permutations of the set $\{1, \dots, n\}$,
- $\mathbb{Z}_n$ is the *cyclic group* of integers $\{0, \dots, n-1\}$ with addition modulo n,
- $\mathbb{D}_n$ is the *dihedral group* of the symmetries of a regular n-gon, and
- $\mathbb{A}_n$ is the *alternating group* of all even permutations of the set $\{1, \dots, n\}$.

We note that $\mathbb{D}_1 \cong \mathbb{Z}_2$, $\mathbb{D}_2 \cong \mathbb{Z}_2^2 = \mathbb{Z}_2 \times \mathbb{Z}_2$, $\mathbb{Z}_3 \cong \mathbb{A}_3$, and $\mathbb{D}_3 \cong \mathbb{S}_3$.

1.1 Group actions

A group G *acts* on a set Ω if there is a mapping $\cdot : G \times \Omega \to \Omega$ such that for every $x \in \Omega$ and for every $g, h \in G$, we have $1 \cdot x = x$ and $(gh) \cdot x = g \cdot (h \cdot x)$. If there is no confusion, we write gx instead of $g \cdot x$. The set Ω is called a *G-set* if G acts on Ω. The action of a group G on itself by conjugation is defined by $h \cdot g = h^{-1}gh = g^h$.

For $x \in \Omega$, the *stabilizer of x* is the group $G_x = \{g : gx = x\}$. For $S \subseteq \Omega$, we define the *set-wise stabilizer of S* is the group $G_S = \{g : gS = S\}$ and the *point-wise stabilizers of S* is the group $G_{(S)} = \{g : gx = x, x \in S\}$.

For $x \in \Omega$ the *orbit of x* is the set $[x]_G = \{gx : g \in G\}$. We write $[x]$ if G is apparent from the context. The action of G on Ω is *transitive* if it acts on Ω with a single orbit, *fixed-point-free* if G_x is trivial for every $x \in \Omega$, and *regular* if it is transitive and fixed-point-free.

Isomorphic actions. Two G-sets Ω_1 and Ω_2 are *isomorphic* if there is a bijection $f : \Omega_1 \to \Omega_2$ such that $f(g \cdot x) = g \cdot f(x)$ for every $g \in G$ and $x \in \Omega_1$.[1] If Ω is a G-set, then we say that two orbits $[x]_G$ and $[y]_G$ are *isomorphic orbits* if the induced G-sets $[x]_G$ and $[y]_G$ are isomorphic.

Example 1.1. We introduce two non-isomorphic $\mathbb{D}_n$-sets, to which we refer repeatedly. Let

$$\Omega = \{0, \dots, n-1\} \quad \text{and} \quad \Omega' = \{\{i, (i+1) \mod n\} : i = 0, \dots, n-1\}$$

be the *vertices* and *edges* of a regular n-gon, for $n \geq 2$, respectively. The *action of $\mathbb{D}_n = \langle r, t \mid r^n = (rt)^2 = 1\rangle$ on the vertices of the n-gon* is defined by setting $r \cdot i = i+1 \mod n$, for $i = 0, \dots, n-1$, and $t \cdot i = n-i \mod n$, for $i = 0, \dots, n-1$. We define the *action of $\mathbb{D}_n$ on the edges of the n-gon* to be the action on Ω' induced by the action of $\mathbb{D}_n$ on Ω. If n is even, then, by Lemma 1.4, these two actions of $\mathbb{D}_n$ are non-isomorphic. For an illustration see Figure 1.1. $\square$

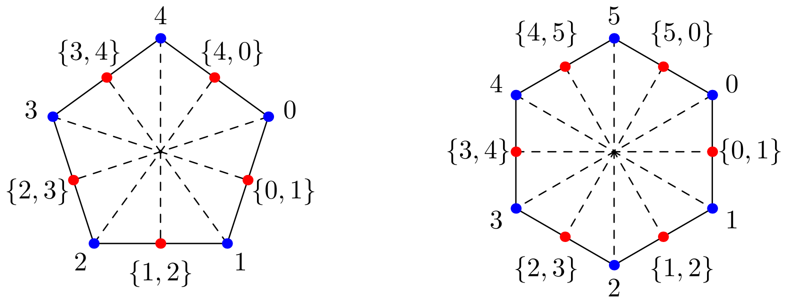


Figure 1.1: The actions of $\mathbb{D}_5$ and $\mathbb{D}_6$ on the vertices and edges of a regular 5-gon and 6-gon, respectively. For $\mathbb{D}_5$, both actions are isomorphic. For $\mathbb{D}_6$, the stabilizers of vertices have fixed points and the stabilizers of edges are fixed-point-free, hence the actions cannot be isomorphic.

For a proof the following lemma see for example [30, Theorem 1.3].

Lemma 1.2. *Let Ω be a G-set, and let G/G_x, for $x \in \Omega$, be the set of all left cosets of G_x. The sets $[x]_G$ and G/G_x are isomorphic G-sets. Moreover, two G-sets $[x]_G$ and $[y]_G$ are isomorphic if and only if G_x and G_y are conjugate in G.*

The next two lemmas are simple consequences of Lemma 1.2.

Lemma 1.3. *If G acts on Ω, then all orbits of size $|G|$ are isomorphic and all orbits of size 1 are isomorphic.*

Proof. Let $x, y \in \Omega$. If $[x]$ and $[y]$ are of size $|G|$, then both G_x and G_y are trivial. If $[x]$ and $[y]$ are of size 1, then both G_x and G_y equal to G. In both cases the statement immediately follows from Lemma 1.2. $\square$

Lemma 1.4. *Let $G \cong \mathbb{D}_n = \langle r, t \mid r^n = t^2 = (rt)^2 = 1\rangle$ act on a set Ω. If n is odd, then all orbits of size n are isomorphic. If n is even, then there are three isomorphism classes of orbits of size n determined by the subgroups $\langle r^{n/2}\rangle$, $\langle t\rangle$, $\langle rt\rangle$, respectively.*

Proof. By the orbit-stabilizer theorem, G_x is isomorphic to $\mathbb{Z}_2$. If n is even, there are exactly three conjugacy classes of subgroups of order 2. If n is odd there is just one conjugacy class of G_x in G. Thus, the statement follows from Lemma 1.2. $\square$

1.2 Group products

Given two groups K and H, and a group homomorphism $\theta\colon H \to \mathrm{Aut}(K)$, $h \mapsto \theta_h$, the *outer semidirect product* $K \rtimes_\theta H$ is the cartesian product of K and H with the operation is defined by the rule

$$(g_1, h_1) \cdot (g_2, h_2) = (g_1\theta_{h_1}(g_2), h_1h_2).$$

Note that if θ is trivial, then $K \rtimes_\theta H = K \times H$ is the *(outer) direct product.* A group G is an *inner semidirect product* if $G = KH$, where $K \lhd G$, $H \leq G$, and $K \cap H = \{1\}$.

[1] Here we follow the terminology of Peter J. Cameron [30, Page 3].

It is well-known [141] that inner and outer semidirect products are equivalent in the following sense. For every inner semidirect product $G = KH$, there is a homomorphism $\theta\colon H \to \mathrm{Aut}(K)$, defined by $\theta_h(x) = hxh^{-1}$, such that $G \cong K \rtimes_\theta H$. On the other hand, if $G = K \rtimes_\theta H$ is an outer semidirect product, then $G = K'H'$, where $K' = \{(k, 1) : k \in K\}$ and $H' = \{(1, h) : h \in H\}$.

We use only special cases of semidirect products, which we now introduce.

Wreath product. Let D and Q be groups, let Ω be a finite Q-set, and let $K = \prod_{\omega\in\Omega} D_\omega$, where $D_\omega = D$ for all $\omega \in \Omega$. Then the *wreath product* of D by Q, denoted by $D \wr Q$, is the semidirect product of K by Q, where Q acts on K by $q \cdot (d_\omega) = (d_{q\omega})$ for $q \in Q$ and $(d_\omega) \in \prod_{\omega\in\Omega} D_\omega$.

Inhomogeneous wreath product. We generalize the standard wreath product as follows. Let $D_1, \ldots, D_m$ and Q be groups, let Ω be a finite Q-set with orbits $\Omega_1, \ldots, \Omega_m$, and let $K = \prod_{i=1}^{m} \prod_{\omega\in\Omega_i} D_{i,\omega}$, where $D_{i,\omega} = D_i$ for $\omega \in \Omega_i$ and $i = 1, \ldots, m$. Then the *inhomogeneous wreath product* of the groups $D_1, \ldots, D_m$ by Q, denoted by $(D_1, \ldots, D_m) \wr\wr Q$, is the semidirect product of K by Q, where Q acts on K by $q \cdot (d_{i,\omega}) = (d_{i,q\omega})$ for $q \in Q$ and $(d_{i,\omega}) \in \prod_{i=1}^{m} \prod_{\omega\in\Omega_i} D_{i,\omega}$. Note that by choosing $D_1 = \cdots = D_m = D$ we obtain the wreath product $D \wr Q$.

It is natural to ask under which conditions an inner semidirect product $G = KQ$, where $K \lhd G$, is also an inhomogenous wreath product.

Theorem 1.5. *Let $G = KQ$ be a semidirect product with $K \lhd G$. The group $G = KQ$ is an inhomogeneous wreath product if and only if there is a set Ω and a homomorphism $\varphi\colon Q \to \mathrm{Sym}(\Omega)$ such that*

(i) $K = \prod_{\omega\in\Omega} K_\omega$,

(ii) $K_\omega^h = K_{\varphi(h)(\omega)}$, *for all $h \in H$, and,*

(iii) $g^{h_1} = g^{h_2}$ *if and only if $\varphi(h_1)(\omega) = \varphi(h_2)(\omega)$, for all $\omega \in \Omega$ and $g \in K_\omega$.*

If Λ_i is a D_i-set, for $i = 1, \ldots, m$, then $\bigcup_j(\Lambda_i \times \Omega_i)$ can be made into a $((D_1, \ldots, D_m) \wr\wr Q)$-set. Given $d \in D_i$ and $\omega \in \Omega_i$, for $i = 1, \ldots, m$, we define the permutation $d^*_{i,\omega}$ of $\bigcup_j(\Lambda_j \times \Omega_j)$ as follows: for each $(\lambda, \omega') \in \bigcup_j(\Lambda_j \times \Omega_j)$, set

$$d^*_{i,\omega}(\lambda, \omega') = \begin{cases} (d\lambda, \omega') & \text{if} \quad \omega' = \omega, \\ (\lambda, \omega') & \text{if} \quad \omega' \neq \omega. \end{cases}$$

It is easy to see that $d^*_{i,\omega} d'^*_{i,\omega} = (dd')^*_{i,\omega}$. Thus,

$$D^*_{i,\omega} = \{d^*_{i,\omega} : d \in D_i\}$$

is a subgroup of $\mathrm{Sym}(\bigcup_j \Lambda_j \times \Omega_j)$. This is certified by the map $D_i \to D^*_{i,\omega}$, given by $d \mapsto d^*_{i,\omega}$, which is an isomorphism. For each $q \in Q$, we define a permutation q^* of $\bigcup_j(\Lambda_j \times \Omega_j)$ by

$$q^*(\lambda, \omega') = (\lambda, q\omega'),$$

and define

$$Q^* = \{q^* : q \in Q\}.$$

It is easy to see that Q^* is a subgroup of $\mathrm{Sym}(\bigcup_j \Lambda_j \times \Omega_j)$. This is certified by the map $Q \to Q^*$, given by $q \mapsto q^*$, which is an isomorphism.

We note that an analogous concept for permutation groups was defined in [8, 140] under the names "G-Kranz product" and "generalized composition", respectively. In particular, if all the groups involved in the inhomogeneous wreath product are permutation groups, then these concepts coincide. The following theorem is a generalization of [141, Theorem 7.24].

Theorem 1.6. *Let $D_1, \ldots, D_m$ and Q be groups, let Ω be a Q-set with orbits $\Omega_1, \ldots, \Omega_m$, and let Λ_i be a D_i-set for $i = 1, \ldots, m$. The inhomogeneous wreath product $(D_1, \ldots, D_m) \wr Q$ is isomorphic to the group*

$$W = \langle \bigcup_{i=1}^{m} \bigcup_{\omega \in \Omega_i} D^*_{i,\omega}, Q^* \rangle \leq \mathrm{Sym}(\bigcup_{i=1}^{m} \Lambda_i \times \Omega_i).$$

In particular, $W = K^*Q^*$, where $K^* = \langle \bigcup_{i=1}^{m} \bigcup_{\omega \in \Omega_i} D^*_{i,\omega} \rangle$ is the direct product $\prod_{i=1}^{m} \prod_{\omega \in \Omega_i} D^*_{i,\omega}$.

Proof. We show that the group K^* is the direct product. If $\omega \neq \omega'$, then $d'^*_{j,\omega'} d^*_{i,\omega} d'^{*-1}_{j,\omega'} = d^*_{i,\omega}$ for all $d^*_{i,\omega}$ and $d'^*_{j,\omega'} \in K^*$. Thus, $D^*_{i,\omega} \lhd K^*$ for all $i = 1, \ldots, m$ and $\omega \in \Omega$. Further, each $d^*_{i,\omega} \in D^*_{i,\omega}$ fixes all $(\lambda, \omega') \in \bigcup_j (\Lambda_j \times \Omega_j)$ with $\omega' \neq \omega$, while each element of $\langle \bigcup_{j=1}^{m} \bigcup_{\omega' \in \Omega_i, \omega' \neq \omega} D^*_{j,\omega'} \rangle$ fixes all (λ, ω) for all $\lambda \in \Lambda_i$. It follows that if $d^*_{i,\omega} \in D^*_{i,\omega} \cap \langle \bigcup_{j=1}^{m} \bigcup_{\omega' \in \Omega_i, \omega' \neq \omega} D^*_{j,\omega'} \rangle$, then $d^*_{i,\omega} = 1$.

If $d^*_{i,\omega} \in D^*_{i,\omega}$ and $q^* \in Q^*$, then $q^* d^*_{i,\omega} q^{*-1} = d^*_{i,q\omega}$. Hence, $q^* K^* q^{*-1} \leq K^*$ for each $q^* \in Q^*$. Since $W = \langle K^*, Q^* \rangle$, it follows that $K^* \lhd W$. We get that $W = K^*Q^*$ and to see that W is a semidirect product of K^* by Q^*, it suffices to show that $K^* \cap Q^* = 1$. If $d^*_{i,\omega} \in K^*$, then either $d^*_{i,\omega}(\lambda, \omega') = (d\lambda, \omega')$ or $d^*_{i,\omega}(\lambda, \omega') = (\lambda, \omega')$, i.e., each $d^*_{i,\omega}$ always fixes the second coordinate. If $q^* \in Q^*$, then $q^*(\lambda, \omega') = (\lambda, q\omega')$, i.e., each q^* always fixes the first coordinate. Therefore every element of $K^* \cap Q^*$ fixes every (λ, ω') and hence must be equal to 1.

Finally, the map $(D_1, \ldots, D_m) \wr Q \to W$, defined by $(d_{i,\omega})q \mapsto (d^*_{i,\omega})q^*$, is an isomorphism. $\square$

Lemma 1.7. *Let $G = KQ$ be a semidirect product of K by Q. Further, let $K = K_1 \times K_2$ and $Q = Q_1 \times Q_2$ such that Q_1 fixes K_2 pointwise and Q_2 fixes K_1 pointwise. Then,*

$$G \cong (K_1 \rtimes Q_1) \times (K_2 \rtimes Q_2).$$

Proof. First, we prove that $K_1Q_1 \lhd KQ$. For $k_1q_1 \in K_1Q_1$ and $kq \in KQ$, we have

$$kqk_1q_1(kq)^{-1} = kqk_1q^{-1}qq_1q^{-1}k^{-1}.$$

There are $k'_1 \in K_1$ and $q'_1 \in Q_1$ such that the right-hand side is equal to

$$kk'_1q'_1k^{-1} = kk'_1q'_1k^{-1}q'^{-1}_1q'_1.$$

Now, if $k \in K_1$, then $q'_1k^{-1}q'^{-1}_1 \in K_1$. On the other hand if $k \in K_2$, then $q'_1k^{-1}q'^{-1}_1 = k^{-1}$ and the right-hand side is equal to $k'_1q'_1$. Similarly, one can prove that $K_2Q_2 \lhd KQ$. It is easy to see that $K_1Q_1 \cap K_2Q_2 = \{1\}$. Finally, $K_i \lhd Q_i$ and so K_iQ_i is a semidirect product, for $i = 1, 2$. $\square$

Lemma 1.8. *Let $D_1, \ldots, D_m$ and Q be groups, let Ω be a Q-set with orbits $\Omega_1, \ldots, \Omega_m$. Moreover, assume that all $\Omega_1, \ldots, \Omega_k$ are of size one for some $k \leq m$. Then,*

$$(D_1, \ldots, D_m) \wr Q \cong D_1 \times \cdots \times D_k \times (D_{k+1}, \ldots, D_m) \wr Q.$$

Proof. There are groups $D_{i,\omega} \cong D_i$, for $i = 1, \ldots, m$, such that $(D_1, \ldots, D_m) \wr Q$ is the semidirect product of K by Q, where $K = \prod_{i=1}^m \prod_{\omega \in \Omega_i} D_{i,\omega}$ and Q acts on K by $q \cdot (d_{i,\omega}) = (d_{i,q\omega})$. By the assumptions, we can write

$$K = D_{1,\omega_1} \times \cdots \times D_{k,\omega_k} \times \prod_{i=k+1}^{m} \prod_{\omega \in \Omega_i} D_{i,\omega},$$

where $\Omega_i = \{\omega_i\}$, for $i \leq k$. By the definition of inhomogeneous wreath product, all subgroups D_{i,ω_i} are centralized by Q, which completes the proof. □

Lemma 1.9. *Let $D_1, \ldots, D_m$ and Q be groups, let Ω be a Q-set with orbits $\Omega_1, \ldots, \Omega_m$, where Ω_1 and Ω_2 are isomorphic. Then,*

$$(D_1, \ldots, D_m) \wr Q \cong (D_1 \times D_2, D_3, \ldots, D_m) \wr Q.$$

Proof. There are groups $D_{i,\omega} \cong D_i$, for $i = 1, \ldots, m$, such that $(D_1, \ldots, D_m) \wr Q$ is the semidirect product of K by Q, where $K = \prod_{i=1}^m \prod_{\omega \in \Omega_i} D_{i,\omega}$ and Q acts on K by $q \cdot (d_{i,\omega}) = (d_{i,q\omega})$. Further, let $f \colon \Omega_1 \to \Omega_2$ be a bijection certifying that Ω_1 and Ω_2 are isomorphic orbits, i.e., $f(q\omega) = qf(\omega)$ for every $q \in Q$ and $\omega \in \Omega_1$. We can write

$$K = \prod_{\omega \in \Omega_1} (D_{1,\omega} \times D_{2,f(\omega)}) \times \prod_{i=3}^{m} \prod_{\omega \in \Omega_i} D_{i,\omega}.$$

Let $d_{1,\omega} d'_{2,\omega} \in D_{1,\omega} \times D_{2,f(\omega)}$ for some $\omega \in \Omega_1$. From the definition of inhomogeneous wreath product and isomorphicm of the orbits Ω_1 and Ω_2 it follows that

$$q d_{1,\omega} d'_{2,\omega} q^{-1} = q d_{1,\omega} q^{-1} q d'_{2,\omega} q^{-1} = d_{1,q\omega} d'_{2,qf(\omega)} = d_{1,\omega} d'_{2,f(q\omega)} \in D_{1,q\omega} \times D_{2,f(q\omega)}.$$

Thus, the inhomogeneous wreath product $(D_1, \ldots, D_m) \wr Q$ is determined by the induced action of Q on $\Omega \setminus \Omega_2$ and therefore it is isomorphic to $(D_1 \times D_2, D_3, \ldots, D_m) \wr Q$. □

1.3 Two simple applications

We show how can the inhomogenous wreath product be used to describe the automorphism groups of several well-known classes of graphs.

Automorphism groups of disconnected graphs. There is a well-known description of the automorphism group of a disconnected graph in terms of the automorphism groups of its connected components.

Theorem 1.10. *Let $X_1, \ldots, X_n$ be pairwise non-isomorphic simple connected graphs and let X be the disjoint union $X = \bigcup_{i=1}^n \bigcup_{j=1}^{m_i} X_{i,j}$, where $X_{i,j} \cong X_i$. Then*

$$\mathrm{Aut}(X) \cong \mathrm{Aut}(X_1) \wr \mathbb{S}_{m_1} \times \cdots \times \mathrm{Aut}(X_n) \wr \mathbb{S}_{m_n}.$$

Proof. First, we describe the automorphisms of X. We denote $V(X_{i,j}) = \{(v,j) : v \in V(X_i), j \in \{1\dots,m_i\}\}$.

For every $d \in \mathrm{Aut}(X_i)$, there is an automorphism $d^*_{i,j} \in \mathrm{Aut}(X)$ such that $d^*_{i,j}(v,j') = (dv,j')$ if $j' = j$, and $d^*_{i,j}(v,j') = (v,j')$ if $j' \neq j$. Note that the subgroup $D^*_{i,j} = \{d^*_{i,j} : d \in \mathrm{Aut}(X_i)\} \leq \mathrm{Aut}(X)$ fixes all the vertices in $V(X) \setminus V(X_{i,j})$ pointwise.

Further, for every $q \in \mathbb{S}_{m_1} \times \dots \times \mathbb{S}_{m_n}$, there is an automorphism $q^* \in \mathrm{Aut}(X)$ of X such that $q^*(v,j) = (v,qj)$. Clearly the subgroup $Q^* = \{q^* : q \in \mathbb{S}_{m_1} \times \dots \times \mathbb{S}_{m_n}\} \leq \mathrm{Aut}(X)$ acts on the set $\bigcup_i\{(i,j) : j = 1,\dots,m_i\}$ of the labels of the connected components of X.

We have $\mathrm{Aut}(X) = \langle \bigcup_i \bigcup_j D^*_{i,j}, Q^*\rangle$ and by Theorem 1.6

$$\mathrm{Aut}(X) \cong (\mathrm{Aut}(X_1),\dots,\mathrm{Aut}(X_n)) \wr (\mathbb{S}_{m_1} \times \dots \times \mathbb{S}_{m_n}).$$

Finally, applying Lemma 1.7 repeatedly on the right-hand side gives the theorem. □

Figure 1.2 illustrates the structure of the automorphism group of a simple disconnected graph.

Automorphism groups of trees. By a simple application of the previous theorem, we prove the following theorem characterizing the automorphism groups of trees.

Theorem 1.11. [10, Proposition 1.15] *The class* $\mathrm{Aut}(\mathsf{TREE})$ *is equal to the class of groups* $\mathcal{G}$ *defined inductively as follows:*

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

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

(c) If $G \in \mathcal{G}$, *then* $G \wr \mathbb{S}_n \in \mathcal{G}$ *for all* $n \in \mathbb{N}$.

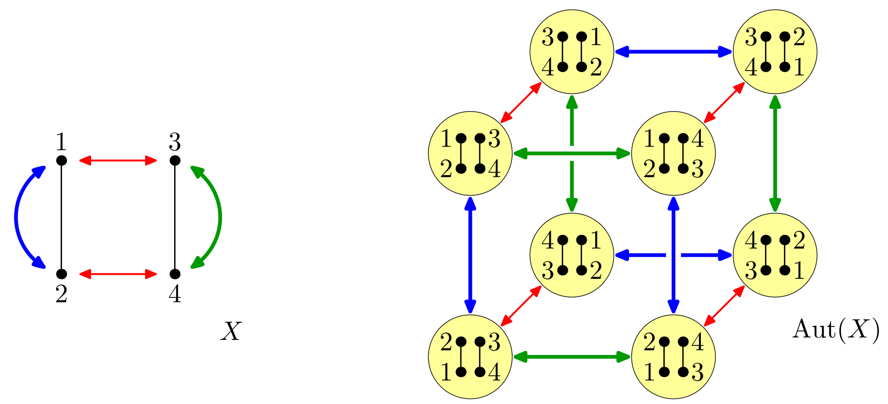


Figure 1.2: On the left, a disconnected graph X consisting of two independent edges. On the right, the Cayley graph of $\mathrm{Aut}(X)$, generated by three involutions acting on X on the left: (1 2)(3)(4), (1)(2)(3 4), and (1 3)(2 4). We have $\mathrm{Aut}(X) \cong \mathbb{Z}_2^2 \rtimes \mathbb{Z}_2 = \mathbb{Z}_2 \wr \mathbb{Z}_2$.

Proof. First, we show that $\mathrm{Aut}(\mathsf{TREE}) \subseteq \mathcal{G}$. We proceed by induction on the number of vertices. Every tree has a *center*, which is either a vertex or an edge. For a tree T, the group $\mathrm{Aut}(T)$ stabilizes its center setwise. Moreover, if the center of T is an edge, we form T' by subdividing this edge by exactly one vertex. Then $\mathrm{Aut}(T') \cong \mathrm{Aut}(T)$ and the center of T' is a vertex. Thus, we may assume that the center is a vertex v. After deleting v together with the edges incident to it, we get a disconnected graph consisting of k_i copies of the tree T_i, for $i = 1, \ldots, n$. By Theorem 1.10, we have

$$\mathrm{Aut}(T) \cong \mathrm{Aut}(T_1) \wr \mathbb{S}_{m_1} \times \cdots \times \mathrm{Aut}(T_n) \wr \mathbb{S}_{m_n},$$

and so, by the induction hypothesis, $\mathrm{Aut}(T)$ is isomorphic to a group in $\mathcal{G}$.

To prove $\mathrm{Aut}(\mathsf{TREE}) \supseteq \mathcal{G}$, we proceed by the induction on the number of operations. For each of the two operations, we easily construct a tree realizing the corresponding operation. $\square$

2. Interval, permutation and circle graphs

2.1 Introduction

In this chapter, we derive which abstract groups can be automorphism groups of interval, permutation, and circle graphs. Each of these graph classes are intersection graphs of certain geometric objects and admits a tree-like description of the structure of all representations of a given graph. For interval graphs we have PQ-trees, for permutation graphs modular trees, and for circle graphs split trees. Consequently, the automorphism groups can be determined for theses classes similarly as for trees; see Theorem 1.11. The same technique is applied in Chapter 5 to planar graphs, however, for planar graphs the technical difficulties are considerably more involved.

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 $R_u \cap R_v \neq \emptyset$; the intersections encode the edges. To get nice graph classes, one typically restricts the sets R_v to particular classes of geometrical objects; for an overview, see the classical books [78, 148]. We show that a well-understood structure of all intersection representations allows one to determine the automorphism group.

For interval graphs, circle graphs, and permutations graphs, we show that the automorphism groups can be described inductively as inhomogeneous wreath products of symmetric, dihedral, and cyclic groups. On the negative side, we show that the automorphism groups of comparability graphs of dimension 4, a superclass of permutation graphs, are already universal. We recall that a class $\mathcal{C}$ of graphs is *universal* if for every group G there is a graph $X \in \mathcal{C}$ such that $G \cong \mathrm{Aut}(X)$.

Interval graphs. In an *interval representation* of a graph, each set R_v is a closed interval of the real line. A graph is an *interval graph* if it has an interval representation; see Figure 2.1a. A graph is a *unit interval graph* if it has an interval representation with each interval of the length one. We denote these classes by INT and $\mathsf{UNIT\ INT}$, respectively. *Caterpillars* ($\mathsf{CATERPILLAR}$) are trees with all leaves attached to a central path; we have $\mathsf{CATERPILLAR} = \mathsf{INT} \cap \mathsf{TREE}$.

Theorem 2.1. *The following equalities hold:*

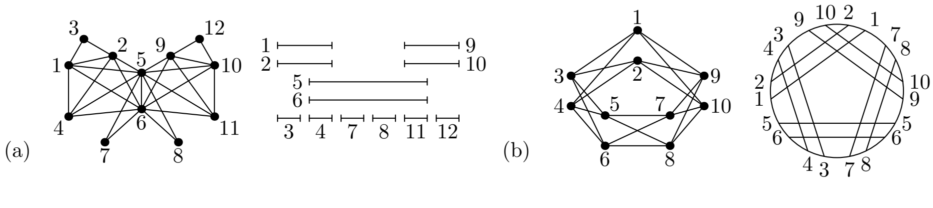


Figure 2.1: (a) An interval graph and one of its interval representations. (b) A circle graph and one of its circle representations.

(i) $\mathrm{Aut}(\mathsf{INT}) = \mathrm{Aut}(\mathsf{TREE})$,

(ii) $\mathrm{Aut}(\text{connected } \mathsf{UNIT\ INT}) = \mathrm{Aut}(\mathsf{CATERPILLAR})$,

Concerning (i), this equality is not well-known. It was stated by Hanlon [88] without a proof in the conclusion of his paper from 1982 on enumeration of interval graphs.

Our structural analysis is based on PQ-trees [20] which 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 considering PQ-trees, this equality is surprising, since these classes are very different. Caterpillars which form their intersection have very restricted automorphism groups (see Lemma 2.11) as well. The result (ii) follows from the known properties of unit interval graphs and our understanding of $\mathrm{Aut}(\mathsf{INT})$.

Booth and Lueker [20] originally invented PQ-trees to solve the more general *consecutive ordering problem* in linear time. As a consequence, they obtained the first linear-time recognition algorithm for interval graphs.

Circle graphs. In a *circle representation*, each R_v is a chord of a circle. A graph is a circle graph (CIRCLE) if it has a circle representation; see Figure 2.1b.

Theorem 2.2. *Let Σ be the class of groups defined inductively as follows:*

(a) $\{1\} \in \Sigma$.

(b) *If $G_1, G_2 \in \Sigma$, then $G_1 \times G_2 \in \Sigma$.*

(c) *If $G \in \Sigma$, then $G \wr \mathbb{S}_n \in \Sigma$.*

(d) *If $G_1, G_2, G_3 \in \Sigma$, then $(G_1^4, G_2^2, G_3^2) \rtimes \mathbb{Z}_2^2 \in \Sigma$, where the action of $\mathbb{Z}_2^2$ on the coordinates of G_1^4 is regular, and the actions on the coordinates of G_2^2 and G_3^2 are isomorphic to the actions of $\mathbb{Z}_2^2$ on the vertices and edges of a 2-gon, respectively.*

Then $\mathrm{Aut}(\text{connected } \mathsf{CIRCLE})$ *consists of the following groups:*

(e) *If $G \in \Sigma$, then $G \wr \mathbb{Z}_n \in \mathrm{Aut}(\text{connected } \mathsf{CIRCLE})$, for $n \geq 1$.*

(f) *If $G_1, G_2 \in \Sigma$, then $(G_1^{2n}, G_2^n) \rtimes \mathbb{D}_n \in \mathrm{Aut}(\text{connected } \mathsf{CIRCLE})$, for $n \geq 3$ odd, where the action of $\mathbb{D}_n$ is regular on the coordinates of G_1^{2n} and on the coordinates of G_2^n it is isomorphic to the action of $\mathbb{D}_n$ on the vertices of an n-gon.*

(g) *If $G_1, G_2, G_3 \in \Sigma$, then $(G_1^{2n}, G_2^n, G_3^n) \rtimes \mathbb{D}_n \in \mathrm{Aut}(\text{connected } \mathsf{CIRCLE})$, for $n \geq 2$ even, where the action of $\mathbb{D}_n$ is regular on the coordinates of G_1^{2n}, and the actions on the coordinates of G_2^n and G_3^n are isomorphic to the actions of $\mathbb{D}_n$ on the vertices and edges of an n-gon, respectively.*

The automorphism group of a disconnected circle graph can be easily determined using Theorem 1.10. We use split trees describing all representations of circle graphs. The class Σ consists of the stabilizers of vertices in connected circle graphs and $\mathrm{Aut}(\mathsf{TREE}) \subsetneq \Sigma$.

Comparability graphs. A *comparability graph* is derived 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 2.2a. This class was first studied by Gallai [72] and we denote it by COMP.

An important structural parameter of a poset P is its *Dushnik-Miller dimension* [57]. It is the least number of linear orderings $L_1, \dots, L_k$ such that $P = L_1 \cap \cdots \cap L_k$. (For 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. (Every transitive orientation has the same dimension; see Section 2.5.3.) 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:

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

For instance, [143] proves that the bipartite graph of the incidence between the vertices and the edges of a planar graph always belongs to 3-DIM.

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 called *segments* [14]; see Figure 2.2b and c. We have $\mathsf{FUN} = \mathsf{co\text{-}COMP}$ [79] and $\mathsf{PERM} = \mathsf{COMP} \cap \mathsf{co\text{-}COMP} = 2\text{-}\mathsf{DIM}$ [63], where $\mathsf{co\text{-}COMP}$ are the complements of comparability graphs.

Since 1-DIM consists of all complete graphs, $\mathrm{Aut}(1\text{-}\mathsf{DIM}) = \{\mathbb{S}_n : n \in \mathbb{N}\}$. The automorphism groups of $2\text{-}\mathsf{DIM} = \mathsf{PERM}$ are the following:

Theorem 2.3. *The class* $\mathrm{Aut}(\mathsf{PERM})$ *is 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* $(G_1^4, G_2^2, G_3^2) \wr\wr \mathbb{Z}_2^2 \in \mathrm{Aut}(\mathsf{PERM})$*,where the action of* $\mathbb{Z}_2^2$ *on the coordinates of* G_1^4 *is regular, and the actions on the coordinates of* G_2^2 *and* G_3^2 *are isomorphic to the actions of* $\mathbb{Z}_2^2$ *on the vertices and edges of a 2-gon, respectively.*

In comparison to Theorem 1.11, there is the additional operation (d) which shows that $\mathrm{Aut}(\mathsf{TREE}) \subsetneq \mathrm{Aut}(\mathsf{PERM})$. Geometrically, the group $\mathbb{Z}_2^2$ in (d) corresponds to the horizontal and vertical reflections of a symmetric permutation representation. Our result also easily gives the automorphism groups of *bipartite permutation graphs* ($\mathsf{BIP\ PERM}$), in particular we have $\mathrm{Aut}(\mathsf{CATERPILLAR}) \subsetneq \mathrm{Aut}(\mathsf{BIP\ PERM}) \subsetneq \mathrm{Aut}(\mathsf{PERM})$.

Corollary 2.4. *The class* $\mathrm{Aut}(\text{connected } \mathsf{BIP\ PERM})$ *consists of all abstract groups* G_1*,* $G_1 \wr \mathbb{Z}_2 \times G_2 \times G_3$*, and* $(G_1^4, G_2^2) \wr\wr \mathbb{Z}_2^2$*, where* G_1 *is a direct product of symmetric groups, and* G_2 *and* G_3 *are symmetric groups.*

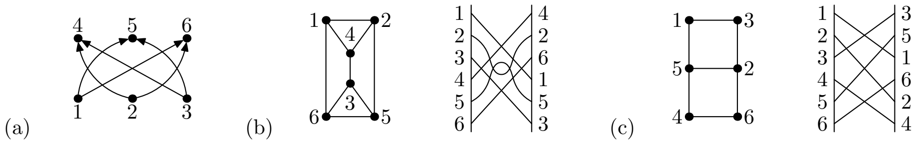


Figure 2.2: (a) A comparability graph with a transitive orientation. (b) A function graph and one of its representations. (c) A permutation graph and one of its representations.

Comparability graphs are universal since they contain bipartite graphs; we can orient all the edges from one part to the other. Since the automorphism group is preserved by complementation, $\mathsf{FUN} = \mathsf{co\text{-}COMP}$ implies that also function graphs are universal. In Section 2.5, we explain the universality of FUN and COMP in more detail. Similarly posets are known to be universal [18, 155].

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 2.5. *For every $k \geq 4$, the class k-DIM is universal and its graph isomorphism is GI-complete. The same holds for posets of the dimension k.*

Yannakakis [164] proved that recognizing 3-DIM is NP-complete by a reduction from 3-coloring. For a graph X, a comparability graph Y is constructed with several vertices representing each element of $V(X) \cup E(X)$. It is proved 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 the 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 [164]. A different construction follows from [32, 159]. In particular, in [32] it is proven that all bipartite grid intersection graphs have dimension at most 4 and in [159] it is proven that they are GI-complete.

Related graph classes. Theorems 2.1, 2.2 and 2.3 and Corollary 2.4 state that INT, UNIT INT, CIRCLE, PERM, and BIP PERM are non-universal. Figure 2.3 shows that their superclasses are already universal.

Trapezoid graphs (TRAPEZOID) are intersection graphs of trapezoids between two parallel lines and they have universal automorphism groups [153]. *Claw-free graphs* (CLAW-FREE) are graphs with no induced $K_{1,3}$. Roberts [139] 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 generalize interval graphs and are universal [117]. *Interval filament graphs* (IFA) are intersection graphs of graphs of continuous functions $f_u : [a, b] \to \mathbb{R}$ such that $f_u(a) = f_u(b) = 0$ and $f_u(x) > 0$ for $x \in (a, b)$ [73].

Outline of the chapter. In Section 2.2, we explain our general technique for determining the automorphism group from the geometric structure of all repre-

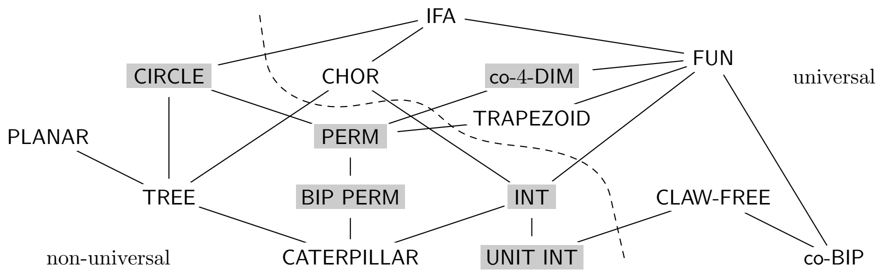


Figure 2.3: The inclusions between the considered graph classes. In this chapter, we characterize the automorphism groups of the classes in gray.

sentations, and relate it to map theory. We describe the automorphism groups of interval and unit interval graphs in Section 2.3, of circle graphs in Section 2.4, and of permutation and bipartite permutation graphs in Section 2.5. Our results are constructive and lead to polynomial-time algorithms computing automorphism groups of these graph classes; see Section 2.6. We conclude with several open problems.

2.2 Automorphism groups acting on intersection representations

In this section, we describe a general technique which allows us to geometrically understand automorphism groups of some intersection-defined graph classes. Suppose that one wants to understand an abstract group G. Sometimes, it is possible to interpret G using a natural action on some set which is easier to understand. The action is called *faithful* if no non-trivia non-trivial element of G belongs to all stabilizers. The structure of G is captured by a faithful action. We require that this action is "faithful enough", which means that the stabilizers are easily understood.

Our approach is inspired by theorey of maps. A *map* $\mathcal{M}$ is an embedding of a graph X into a surface such that every face is homeomorphic to a disk. If the underlying graph is simple, an automorphism of a map is a permutation of the vertices which preserves the vertex-edge-face incidences. If we restrict to orientable surfaces, then by choosing a global orientation of the underlying surface we get a cyclic permutation of the half-edges emanating from every vertex. One can consider the action of $\mathrm{Aut}(X)$ on the set of all maps of X: for $\pi \in \mathrm{Aut}(X)$, we get another map $\pi(\mathcal{M})$ in which the half-edges in the rotational schemes are permuted by π; see Figure 2.4. The stabilizer of a map $\mathcal{M}$ in this action, exactly corresponding to $\mathrm{Aut}(\mathcal{M})$, is the subgroup of $\mathrm{Aut}(X)$ which preserves or reflects the rotational schemes. Unlike $\mathrm{Aut}(X)$, we know that $\mathrm{Aut}(\mathcal{M})$ is always small and can be efficiently determined. The action of $\mathrm{Aut}(X)$ describes morphisms between different maps and in general can be very complicated. For more about maps, see also Chapter 6.

The induced action. For a graph X, we denote by $\mathfrak{Rep}$ the set of all its (interval, circle, etc.) intersection representations. An automorphism $\pi \in \mathrm{Aut}(X)$

creates from $\mathcal{R} \in \mathfrak{Rep}$ another representation $\mathcal{R}'$ such that $R'_{\pi(u)} = R_u$; so π swaps the labels of the sets of $\mathcal{R}$. We denote $\mathcal{R}'$ as $\pi(\mathcal{R})$, and $\mathrm{Aut}(X)$ acts on $\mathfrak{Rep}$.

The general set $\mathfrak{Rep}$ is too 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 with respect to the action of $\mathrm{Aut}(X)$: 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 induced 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 this representation. For a nice class of intersection graphs, such as interval, circle or permutation graphs, the stabilizers $\mathrm{Aut}(\mathcal{R})$ are very simple. If it is a normal subgroup, then the quotient $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$ describes all morphisms which change one representation in the orbit of $\mathcal{R}$ into another one. Our strategy is to understand these morphisms geometrically, for which we use the structure of all representations, encoded for the considered classes by PQ-, split and modular trees.

2.3 Automorphism groups of interval graphs

In this section, we prove Theorem 2.1. We introduce MPQ-tree, which is a structure that combinatorially captures all possible interval representations of an interval graph. We define the automorphism group of an MPQ-tree and derive a characterization of $\mathrm{Aut}(\mathsf{INT})$, which we prove to be equivalent to the Jordan's characterization of $\mathrm{Aut}(\mathsf{TREE})$. Finally, we answer Hanlon's question [88] by constructing for a given interval graph a tree with the same automorphism group, and we also show the converse construction.

2.3.1 PQ- and MPQ-trees

We denote the set of all maximal cliques of X by $\mathcal{C}(X)$. In 1965, Fulkerson and Gross proved the following fundamental characterization of interval graphs.

Lemma 2.6 (Fulkerson and Gross [69])**.** *A graph X is an interval graph if and only if there exists a linear ordering $\preceq$ of $\mathcal{C}(X)$ such that for every $x \in V(X)$ the maximal cliques containing x appear consecutively in this ordering.*

An ordering $\preceq$ of $\mathcal{C}(X)$ from Lemma 2.6 is called a *consecutive ordering*. Lemma 2.6 implies that an interval representation of x induces a consecutive ordering of $\mathcal{C}(X)$. Two interval representation of X are *different* if they induce different consecutive orderings.

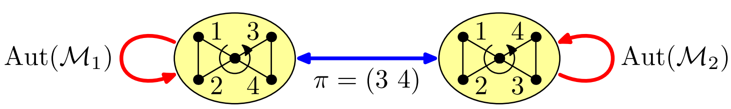


Figure 2.4: There are two different maps, depicted with the action of $\mathrm{Aut}(X)$. The stabilizers $\mathrm{Aut}(\mathcal{M}_i) \cong \mathbb{Z}_2^2$ are normal subgroups. The remaining automorphisms morph one map into the other, for instance π transposing 2 and 3. We have $\mathrm{Aut}(X) \cong \mathbb{Z}_2^2 \rtimes \mathbb{Z}_2$.

PQ-trees. For an interval graphs X, a rooted tree T is a *PQ-tree* of X if it satisfies the following:

- There are two types of inner nodes, namely *P-nodes* and *Q-nodes*.
- For every inner node, its children are ordered from left to right and a P-node has at least two children and a Q-node at least three children.
- The leaves correspond one-to-one to $\mathcal{C}(X)$.

Figure 2.5 shows an example.

The *frontier* of T is the induced ordering $\preceq$ of the leaves. Two PQ-trees are *equivalent* if one can be obtained from the other by a sequence of two *elementary transformations*: (i) an arbitrary permutation of the order of subtrees of a P-node, and (ii) the reversal of the subtrees of a Q-node. Booth and Lueker [20] proved the existence and uniqueness of PQ-trees up to equivalence transformations. Moreover, all possible consecutive orderings of $\mathcal{C}(X)$ are in one-to-one correspondence with the frontiers of all equivalent PQ-trees [20].

For a PQ-tree T, we consider all sequences of equivalent transformations. Two such sequences are *congruent* if they transform T the same. Each sequence consists of several transformations of 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 congruence. We can 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 $\sigma\sigma'\sigma^{-1} \in \Sigma(T)$.

MPQ-trees. A *modified PQ-tree* is created from a PQ-tree by adding information about the vertices. They were described by Korte and Möhring [112] to simplify linear-time recognition of interval graphs. It is not widely known but the equivalent idea was used earlier by Colbourn and Booth [39].

Let T be a PQ-tree representing an interval graph X. We construct the MPQ-tree M by assigning subsets of $V(X)$, called *sections*, to the nodes of T; see Figure 2.5. 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:

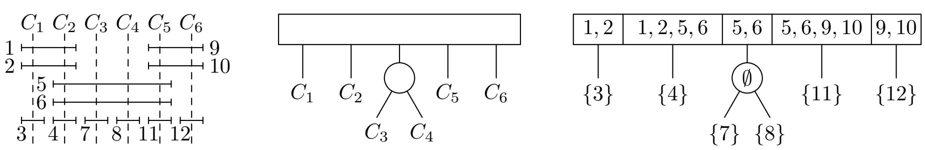


Figure 2.5: An ordering of the maximal cliques, and the corresponding PQ-tree and MPQ-tree. The P-nodes are denoted by circles, the Q-nodes by rectangles. There are four different consecutive orderings.

- 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 $\mathrm{sec}(Q) = \mathrm{sec}_1(Q) \cup \dots \cup \mathrm{sec}_n(Q)$.

Korte and Möhring [112] 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 set of sections if and only if they belong to precisely the same maximal cliques. Figure 2.5 shows an example.

We consider the equivalence relation $\sim_{TW}$ on $V(X)$ is defined as follows: $x \sim_{TW} y$ if and only if $N[x] = N[y]$. If $x \sim_{TW} y$, then we say that they are *twin vertices*. The equivalence classes of $\sim_{TW}$ are called *twin classes*. Twin vertices can usually be ignored, but they influence the automorphism group. Two vertices belong to the same set sections if and only if they are twin vertices.

2.3.2 Automorphisms of MPQ-trees

For a graph X, the automorphism group $\mathrm{Aut}(X)$ induces an action on $\mathcal{C}(X)$ since every automorphism permutes the maximal cliques. If X is an interval graph, then a consecutive ordering $\preceq$ of $\mathcal{C}(X)$ is permuted into another consecutive ordering $\pi(\preceq)$, so $\mathrm{Aut}(X)$ acts on consecutive orderings.

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 Homomorphism Theorem, we know that $\Im(\Phi) \cong \mathrm{Aut}(X)/\,\mathrm{Ker}(\Phi)$. The kernel $\mathrm{Ker}(\Phi)$ consists of all automorphisms which fix the maximal cliques, so they permute the vertices inside each twin class. Thus, $\mathrm{Ker}(\Phi)$ is isomorphic to a direct product of symmetric groups.

Two MPQ-trees M and M' are *isomorphic* if the underlying PQ-trees are equal and 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 2.6. The automorphisms of M are closed under composition, so they form the automorphism group $\mathrm{Aut}(M) \le \Sigma(M)$.

Lemma 2.7. *For an MPQ-tree M, we have* $\mathrm{Aut}(M) = \Im(\Phi)$.

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

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 $\mathrm{Aut}(M) \le \Im(\Phi)$. □

Lemma 2.8. *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 2.7, 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 of $\mathrm{Aut}(X)$ and $\Phi \restriction_H$ is an isomorphism between H and $\mathrm{Aut}(M) = \Im(\Phi)$.

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 $\pi_1, \dots, \pi_n$, 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.

The above construction of H is correct. Since H is the complementary subgroup of $\mathrm{Ker}(\Phi)$, 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 1.10, and the external semidirect product can be constructed in the same way. □

2.3.3 The inductive characterization

Let X be an interval graph, represented by an MPQ-tree M. By Lemma 2.8, $\mathrm{Aut}(X)$ can be described from $\mathrm{Aut}(M)$ and $\mathrm{Ker}(\Phi)$. We build $\mathrm{Aut}(X)$ inductively using M, similarly as in Theorem 1.11.

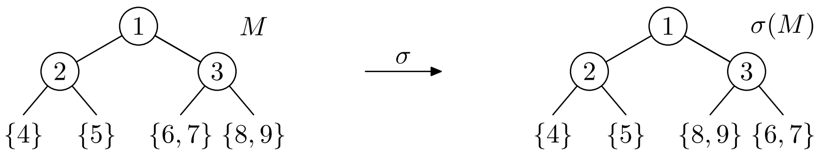


Figure 2.6: The sequence σ, which transposes the children of the P-node with the section $\{3\}$, is an automorphism since $\sigma(M) \cong M$. On the other hand, the transposition of the children the root P-node is not an automorphism.

Figure 2.7: The constructions in the proof of Theorem 2.1(i).

Proof of Theorem 2.1(i). We show that Aut(INT) is closed under (b), (c) (defined in the statement of Theorem 1.11); see Figure 2.7. For (b), we attach interval graphs X_1 and X_2 such that $\mathrm{Aut}(X_i) = G_i$ to an asymmetric interval graph. For (c), let $G \in \mathrm{Aut}(\mathsf{INT})$ and let Y be a connected interval graph with $\mathrm{Aut}(Y) \cong G$. We construct X as the disjoint union of n copies of Y.

For the converse, let M be an MPQ-tree representing an interval graph X. Without a loss of generality we assume that X is connected. Let $M_1, \dots, M_k$ be the subtrees of the root of M and let X_i be the interval subgraphs induced by the vertices of the sections of M_i. We want to build $\mathrm{Aut}(X)$ from $\mathrm{Aut}(X_1), \dots, \mathrm{Aut}(X_k)$ using (b) and (c).

Case 1: The root is a P-node P. Clearly, each X_i, for $i = 1, \dots, k$, is connected. Each sequence $\sigma \in \mathrm{Aut}(M)$ corresponds to interior sequences in $\mathrm{Aut}(M_i)$ and some reordering σ' of $M_1, \dots, M_k$. If $\sigma'(M_i) = M_j$, then necessarily $X_i \cong X_j$. Thus, by applying Theorem 1.10, we obtain the pointwise-stabilizer $\mathrm{Aut}(X)_{(\mathrm{sec}(P))}$ of $\mathrm{sec}(P)$. Finally, it is easy to see that $\mathrm{Aut}(X) \cong \mathrm{Aut}(X)_{(\mathrm{sec}(P))} \times \mathbb{S}_{|\,\mathrm{sec}(P)|}$.

Case 2: The root is a Q-node Q. Let $M_1, \dots, M_k$ be its children from left to right. We call Q symmetric if it is transformed by some sequence of $\mathrm{Aut}(M)$, and asymmetric otherwise. 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 $\mathrm{sec}(Q)$, so it can be build using (b). If Q is symmetric, each sequence $\sigma \in \mathrm{Aut}(M)$ corresponds to interior sequences in $\mathrm{Aut}(M_i)$ and σ', which reverses the order of the subtrees $M_1, \dots, M_k$. Let G_1 be the direct product of the left part of the children and twin classes, and G_2 the one of the middle part. By Lemma 1.8, we get

$$\mathrm{Aut}(X) \cong (G_1^2, G_2) \wr\wr \mathbb{Z}_2 \cong (G_1^2) \wr\wr \mathbb{Z}_2 \times G_2 \cong G_1 \wr \mathbb{Z}_2 \times G_2.$$

Therefore $\mathrm{Aut}(X)$ can be constructed using (b) and (c). $\square$

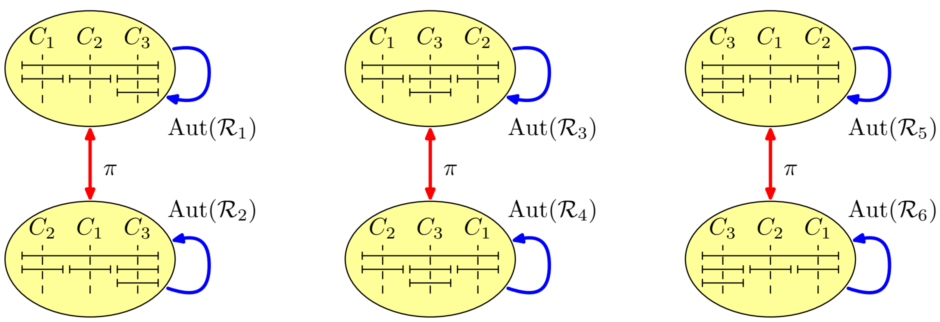


Figure 2.8: An interval graph with six non-equivalent representation. The action of $\mathrm{Aut}(X)$ has three isomorphic orbits.

2.3.4 The action 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 the representations of $\mathfrak{Rep}/\sim$ correspond to consecutive orderings of the maximal cliques; see Figure 2.8 and 2.9.

We interpret our results in terms of the action of $\mathrm{Aut}(X)$ on the set $\mathfrak{Rep}$. In Lemma 2.8, 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 twin class 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 2.8.

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 $\mathrm{Aut}(X)$, which are the same representations up to the labeling of the intervals; see Figure 2.8 and Figure 2.9.

2.3.5 Direct constructions

In this section, we explain Theorem 2.1(i) by direct constructions. The first construction answers the open problem of Hanlon [88].

Lemma 2.9. *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 $\mathrm{Aut}(X) \cong \mathrm{Ker}(\Phi) \rtimes \mathrm{Aut}(M)$ and we inductively encode the structure of M into T.

Case 1: The root is a P-node P. 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 it to make it non-isomorphic to every other subtree attached to the root); see Figure 2.10a. We get $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$.

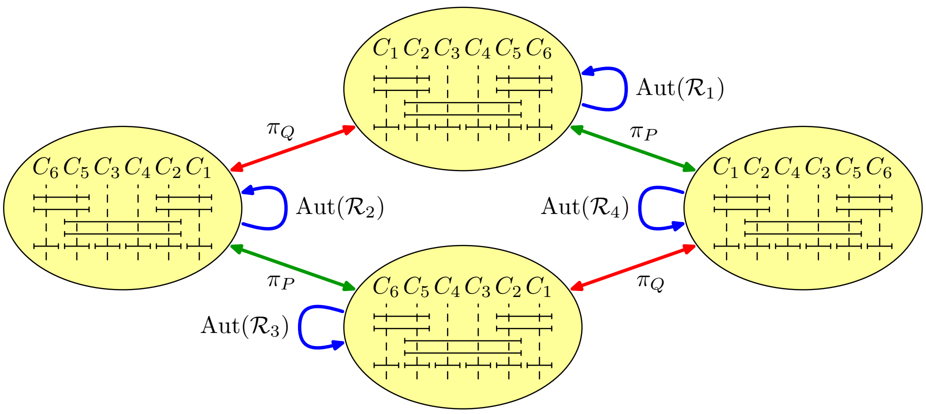


Figure 2.9: The action of $\mathrm{Aut}(X)$ is transitive. An MPQ-tree M of X is depicted in Figure 2.5. There are three twin classes of size two, so $\mathrm{Aut}(\mathcal{R}) \cong \mathbb{Z}_2^3$. The group $\mathrm{Aut}(M)$ is generated by π_Q corresponding to flipping the Q-node, and π_P permuting the P-node. We have $\mathrm{Aut}(M) \cong \mathbb{Z}_2^2$ and $\mathrm{Aut}(X) \cong \mathbb{Z}_2^3 \rtimes \mathbb{Z}_2^2$.

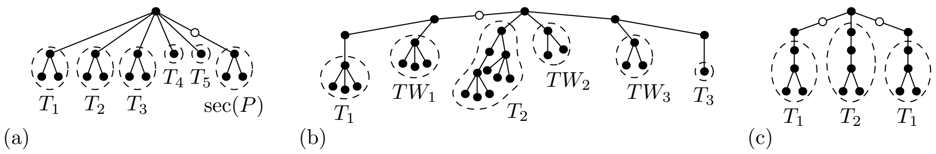


Figure 2.10: For an interval graph X, a construction of a tree T with $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$: (a) The root is a P-node. (b) The root is an asymmetric Q-node. (c) The root is a symmetric Q-node.

Case 2: The root is a Q-node Q. If Q is asymmetric, we attach the trees corresponding to the subtrees of Q and stars corresponding to the vertices of twin classes in the sections of Q to a path, and possibly modify by subdivisions to make it asymmetric; see Figure 2.10b. And if Q is symmetric, then $\mathrm{Aut}(X) \cong (G_1^2 \times G_3) \rtimes \mathbb{Z}_2$ and we just attach trees T_1 and T_2 such that $\mathrm{Aut}(T_i) \cong G_i$ to a path as in Figure 2.10c. In both cases, $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$. □

Lemma 2.10. *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 2.11. 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 Y, as in Figure 2.11. □

2.3.6 Automorphism groups of unit interval graphs

We apply the characterization of $\mathrm{Aut}(\mathsf{INT})$ derived in Theorem 2.1(i) to show that the automorphism groups of connected unit interval graphs are the same as the automorphism groups of the caterpillars (which form the intersection of INT and TREE). First, we describe $\mathrm{Aut}(\mathsf{CATERPILLAR})$:

Lemma 2.11. *The class $\mathrm{Aut}(\mathsf{CATERPILLAR})$ consists of all groups G_1 and $G_1 \wr \mathbb{Z}_2 \times G_2$ where G_1 is a direct product of symmetric groups and G_2 is a symmetric group.*

Proof. We can easily construct caterpillars with these automorphism groups. On the other hand, 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. We can determine $\mathrm{Aut}(X)$ as in the proof of Theorem 2.1(i). □

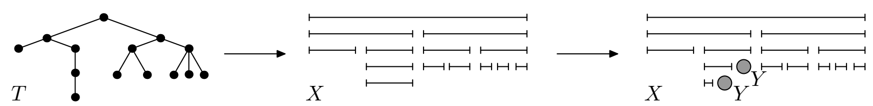


Figure 2.11: We place the intervals following the structure of the tree. We get $\mathrm{Aut}(X) \cong \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_3$, but $\mathrm{Aut}(T) \cong \mathbb{S}_2 \times \mathbb{S}_3$. We fix this by attaching asymmetric interval graphs Y.

Proof of Theorem 2.1(ii). According to Corneil [44], 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 by Lemma 2.11 and the proof of Theorem 2.1(i). □

2.4 Automorphism groups of circle graphs

In this section, we prove Theorem 2.2. We introduce the *split decomposition*. We encode the split decomposition of X by a *split tree* S which captures all circle representations of X. We define automorphisms of S and show that $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$.

2.4.1 Split decomposition

A *split* is a partition (A, B, A', B') of $V(X)$ 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 $|B \cup B'| \geq 2$.

The split decomposition D of X is a collection of graphs constructed by taking a split of X and replacing X by the graphs X_A and X_B defined as follows. The graph X_A is created from $X[A \cup A']$ together with a new *marker vertex* m_A adjacent exactly to the vertices in A. The graph X_B is defined analogously for B, B' and m_B; see Figure 2.12a. The decomposition is then applied recursively on X_A and X_B. Graphs containing no splits are called *prime graphs*. We stop the split decomposition also on *degenerate graphs* which are complete graphs K_n and stars $K_{1,n}$. A split decomposition is called *minimal* if it is constructed by the least number of splits. Cunningham [47] proved that the minimal split decomposition of a connected graph is unique.

The key connection between the split decomposition and circle graphs is the following: a graph X is a circle graph if and only if both X_A and X_B are. In other words, a connected graph X is a circle graph if and only if all prime graphs obtained by the minimal split decomposition are circle graphs.

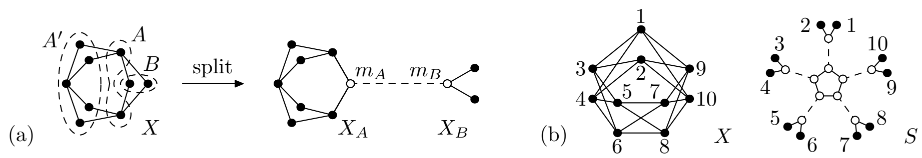


Figure 2.12: (a) An example of a split of the graph X. The marker vertices are depicted in white. The tree edge is depicted by a dashed line. (b) The split tree S of the graph X. We have that $\mathrm{Aut}(S) \cong \mathbb{Z}_2^5 \rtimes \mathbb{D}_5$.

Split tree. A split decomposition D of G is represented by the following graph-labeled tree T called the *split tree T of D* (or a split tree T of G). Initially, T consists of a single node equal to G. At each step, D applies a split on one node N of T. This node is replaced by two new nodes N_A and N_B while the tree edges incident to N are preserved in N_A and N_B and the marker vertices m_A and m_B are further adjacent by a newly formed tree edge. Figure 2.12 shows an example.

Next, we prove that the split tree S captures the adjacencies in X.

Lemma 2.12. *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 existence of an alternating path between x and y by induction according to the length of this path. If $xy \in E(S)$, then it clearly exists. Otherwise the split tree S was constructed by applying a split decomposition. Let Y be the graph in this decomposition such that $xy \in E(Y)$ and there is a split (A, B, A', B') in Y in this decomposition such that $x \in A$ and $y \in B$. 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 by joining 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 $xy \in E(X)$. □

2.4.2 Automorphisms of split trees

In [75], split trees are defined in terms of graph-labeled trees. Our definition is more suitable for automorphisms. 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 2.13. *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 π of X. Since $V(X) \subseteq V(S)$, we define $\pi = \sigma \restriction_{V(X)}$. By Lemma 2.12, $xy \in E(X)$ if and only if there exists an alternating path between them in S. Automorphisms preserve alternating paths, so $xy \in E(X) \iff \pi(x)\pi(y) \in E(X)$.

On the other hand, we show that $\pi \in \mathrm{Aut}(X)$ induces a unique automorphism $\sigma \in \mathrm{Aut}(S)$. We define $\sigma \restriction_{V(X)} = \pi$ and extend it recursively on the marker vertices. Let (A, B, A', B') be a split of the minimal split decomposition in X. This split is mapped by π to another split (C, D, C', D') in the minimal split decomposition, i.e., $\pi(A) = C$, $\pi(A') = C'$, $\pi(B) = D$, and $\pi(B') = D'$. By applying the split decomposition to the first split, 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 the second split we get X_C and X_D with $m_C \in V(X_C)$ and $m_D \in V(X_D)$. Since π 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 $\sigma \in \mathrm{Aut}(S)$. □

Similarly as for trees, there exists a *center* of S which is either a tree edge, or a prime or degenerate node. If the center is a tree edge, we can modify the split tree by adding two adjacent marker vertices in the middle of the tree edge. This clearly preserves the automorphism group $\mathrm{Aut}(S)$, so from now on we assume that S has a center C which which is a prime or degenerate node. We can assume that S is rooted by C, and for a node N, we denote by $S[N]$ the subtree induced by N and its descendants. For $N \neq C$, we call m its *root marker vertex* if it is the marker vertex of N attached to the parent of N.

Recursive construction. We can describe $\mathrm{Aut}(S)$ recursively from the leaves to the root C. Let N be an arbitrary node of S and consider all its descendants. Let $\mathrm{Stab}_{S[N]}(x)$ be the subgroup of $\mathrm{Aut}(S[N])$ which fixes $x \in V(S[N])$. We further color the non-root marker vertices in N by colors coding isomorphism classes of the subtrees attached to them.

Lemma 2.14. *Let $N \neq C$ be a node with the root marker vertex m. Let $N_1, \dots, N_k$ be the children of N with the root marker vertices $m_1, \dots, m_k$. Then*

$$\mathrm{Stab}_{S[N]}(m) \cong \Big(\mathrm{Stab}_{S[N_1]}(m_1) \times \cdots \times \mathrm{Stab}_{S[N_k]}(m_k)\Big) \rtimes \mathrm{Stab}_N(m),$$

where $\mathrm{Stab}_N(m)$ is color preserving.

Proof. We proceed similarly as in the proof of Theorem 1.10. We isomorphically label the vertices of the isomorphic subtrees $S[N_i]$. Each automorphism $\pi \in \mathrm{Stab}_{S[N]}(m)$ is a composition of two automorphisms $\sigma \cdot \tau$ where σ maps each subtree $S[N_i]$ to itself, and τ permutes the subtrees as in π while preserving the labeling. Therefore, the automorphisms σ can be identified with the elements of the direct product $\mathrm{Stab}_{S[N_1]}(m_1) \times \cdots \times \mathrm{Stab}_{S[N_k]}(m_k)$ and the automorphisms τ with the elements of $\mathrm{Stab}_N(m)$. The rest goes along the same lines as the proof of Theorem 1.10. □

The entire automorphism group $\mathrm{Aut}(S)$ is obtained by joining these subgroups at the central node C. No vertex in C has to be fixed by $\mathrm{Aut}(S)$.

Lemma 2.15. *Let C be the central node with the children $N_1, \dots, N_k$ with the root marker vertices $m_1, \dots, m_k$. Then*

$$\mathrm{Aut}(S) \cong \Big(\mathrm{Stab}_{S[N_1]}(m_1) \times \cdots \times \mathrm{Stab}_{S[N_k]}(m_k)\Big) \rtimes \mathrm{Aut}(C),$$

where $\mathrm{Aut}(C)$ is color preserving.

Proof. Similar as the proof of Lemma 2.14. □

2.4.3 The action on prime circle representations

For a circle graph X with $|V(X)| = \ell$, a representation $\mathcal{R}$ is completely determined by a circular word $r_1 r_2 \cdots r_{2\ell}$ such that each $r_i \in V(X)$ and each vertex appears exactly twice in the word. This word describes the order of the endpoints of the chords in $\mathcal{R}$ when the circle is traversed from some point counterclockwise. Two chords intersect if and only if their occurrences alternate in the circular word. Representations are equivalent if they have the same circular words up to rotations and reflections.

The automorphism group $\mathrm{Aut}(X)$ acts on the circle representations in the following way. Let $\pi \in \mathrm{Aut}(X)$, then $\pi(\mathcal{R})$ is the circle representation represented by the word $\pi(r_1)\pi(r_2)\cdots\pi(r_{2\ell})$, i.e., the chords are permuted according to π.

Lemma 2.16. *Let X be a prime circle graph. Then $\mathrm{Aut}(X)$ is isomorphic to a subgroup of a dihedral group.*

Proof. According to [71], each prime circle graph has a unique representation $\mathcal{R}$, up to rotations and reflections of the circular order of endpoints of the chords. Therefore, for every automorphism $\pi \in \mathrm{Aut}(X)$, we have $\pi(\mathcal{R}) = \mathcal{R}$, so π only rotates/reflects this circular ordering. An automorphism $\pi \in \mathrm{Aut}(X)$ is called a *rotation* if there exists k such that $\pi(r_i) = r_{i+k}$, where the indexes are used cyclically. The automorphisms, which are not rotations, are called *reflections*, since they reverse the circular ordering. For each reflection π, there exists k such that $\pi(r_i) = r_{k-i}$. Notice that composition of two reflections is a rotation. Each reflection either fixes two endpoints in the circular ordering, or none of them.

If no non-identity rotation exists, then $\mathrm{Aut}(X)$ is either $\mathbb{Z}_1$, or $\mathbb{Z}_2$. If at least one non-identity rotation exists, let $\rho \in \mathrm{Aut}(X)$ be the non-identity rotation with the smallest value k, called the *basic rotation*. Observe that $\langle\rho\rangle$ contains all rotations, and if its order is at least three, then the rotations act semiregularly on X. If there exists no reflection, then $\mathrm{Aut}(X) \cong \mathbb{Z}_n$. Otherwise, $\langle\rho\rangle$ is a subgroup of $\mathrm{Aut}(X)$ of index two. Let φ be any reflection, then $\rho\varphi\rho = \varphi$ and $\mathrm{Aut}(X) \cong \mathbb{D}_n$. □

Lemma 2.17. *Let X be a prime circle graph and let $m \in V(X)$. Then $\mathrm{Stab}_X(m)$ is isomorphic to a subgroup of $\mathbb{Z}_2^2$.*

Proof. Let $mA\hat{m}B$ be a circular ordering representing X, where m and $\hat{m}$ are the endpoints of the chord representing m, and A and B are sequences of the endpoints of the remaining chords. We distinguish m and $\hat{m}$ to make the action of $\mathrm{Stab}_X(m)$ understandable. Every $\pi \in \mathrm{Stab}_X(m)$ either fixes both m and $\hat{m}$, or swaps them.

Let A' be the reflection of A and B' be the reflection of B. If both m and $\hat{m}$ are fixed, then by the uniqueness this representation can only be reflected along the chord m. If such an automorphism exists in $\mathrm{Stab}_X(m)$, we denote it by φ_m and we have $\varphi_m(mA\hat{m}B) = mB'\hat{m}A'$. If m and $\hat{m}$ are swapped, then by the uniqueness this representation can be either reflected along the line orthogonal to the chord m, or by the 180° rotation. If these automorphisms exist in $\mathrm{Stab}_X(m)$, we denote them by $\varphi_\perp$ and ρ, respectively. We have $\varphi_\perp(mA\hat{m}B) = \hat{m}A'mB'$ and $\rho(mA\hat{m}B) = \hat{m}BmA$. Figure 2.13 shows an example.

All three automorphisms φ_m, $\varphi_\perp$ and ρ are involutions, and $\rho = \varphi_\perp \cdot \varphi_m$. Since $\mathrm{Stab}_X(m)$ is generated by those which exist, it is a subgroup of $\mathbb{Z}_2^2$. □

2.4.4 The inductive characterization

By Lemma 2.13, it is sufficient to determine the automorphism groups of split trees. We proceed from the leaves to the root, similarly as in Theorem 1.11.

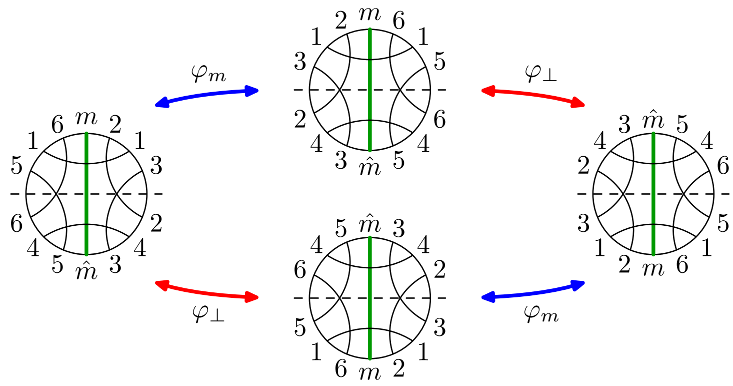


Figure 2.13: A prime circle graph X with $\mathrm{Stab}_X(m) \cong \mathbb{Z}_2^2$.

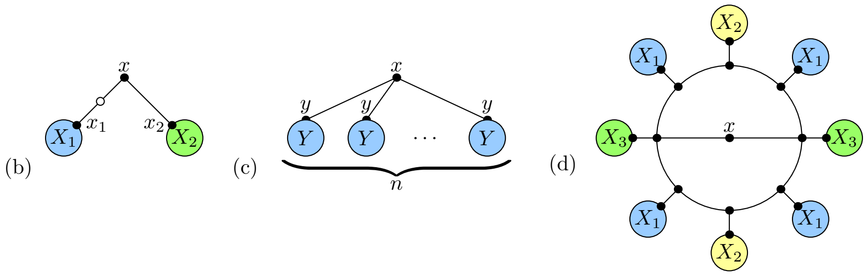


Figure 2.14: The construction of the group in (d). The eight-cycle in X can be reflected horizontally, vertically and rotated by 180°.

Lemma 2.18. *The class Σ defined in Theorem 2.2 consists of the following groups:*

$$\Sigma = \Big\{ G : X \in \text{connected } \mathsf{CIRCLE}, x \in V(X), G \cong \mathrm{Stab}_X(x) \Big\}. \tag{2.1}$$

Proof. First, we show that (2.1) is closed under (b) to (d); see Figure 2.14. For (b), let X_1 and X_2 be circle graphs such that $\mathrm{Stab}_{X_i}(x_i) \cong G_i$. We construct X as in Figure 2.14b, and we get $\mathrm{Stab}_X(x) \cong G_1 \times G_2$. For (c), let Y be a circle graph with $\mathrm{Stab}_Y(y) \cong G$. As X, we take n copies of Y and add a new vertex x adjacent to all copies of y. Clearly, we get $\mathrm{Stab}_X(x) \cong G \wr \mathbb{S}_n$. For (d), let $G_1, G_2, G_3 \in \Sigma$, and let X_i be a circle graph with $\mathrm{Stab}_{X_i}(x_i) \cong G_i$. We construct a graph X as shown in Figure 2.14. We get $\mathrm{Stab}_X(x) \cong (G_1^4, G_2^2, G_3^2) \wr\wr \mathbb{Z}_2^2$.

Next we show that every group from (2.1) belongs to Σ. Let X be a circle graph with $x \in V(X)$, and we want to show that $\mathrm{Stab}_X(x) \in \Sigma$. Since $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$ by Lemma 2.13, we have $\mathrm{Stab}_S(x) \cong \mathrm{Stab}_X(x)$ where x is a non-marker vertex. We prove this by induction according to the number of nodes of S, for the single node it is either a subgroup $\mathbb{Z}_2^2$ (by Lemma 2.17), or a symmetric group.

Let N be the node containing x, we can think of it as the root and x being a root marker vertex. Therefore, by Lemma 2.14, we have

$$\mathrm{Stab}_S(x) \cong \Big(\mathrm{Stab}_{S[N_1]}(m_1) \times \cdots \times \mathrm{Stab}_{S[N_k]}(m_k) \Big) \rtimes \mathrm{Stab}_N(x),$$

where $N_1, \dots, N_k$ are the children of N and $m_1, \dots, m_k$ their root marker vertices. By the induction hypothesis, $\mathrm{Stab}_{S[N_i]}(m_i) \in \Sigma$. There are two cases:

Case 1: N is a degenerate node. Then $\mathrm{Stab}_N(x)$ is a direct product of symmetric groups. The subtrees attached to marker vertices of each color class can

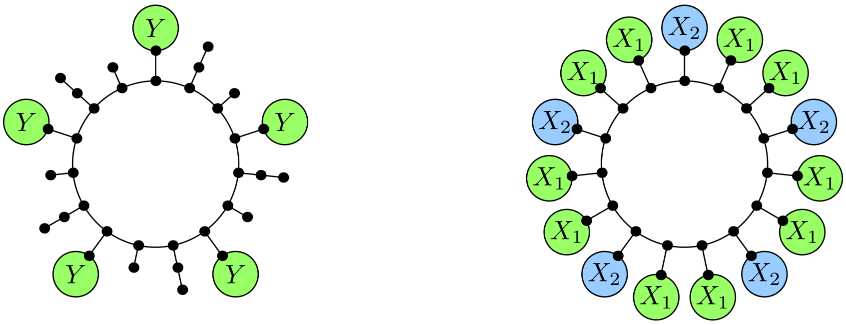


Figure 2.15: The construction of the described groups.

be arbitrarily permuted, independently of each other. Therefore $\mathrm{Stab}_S(x)$ can be constructed using (b) and (c), exactly as in Theorem 1.10.

Case 2: N is a prime node. By Lemma 2.17, $\mathrm{Stab}_N(x)$ is a subgroup of $\mathbb{Z}_2^2$. When it is trivial or $\mathbb{Z}_2$, observe that $\mathrm{Stab}_S(x)$ can be constructed using (b) and (c). The only remaining case is when it is $\mathbb{Z}_2^2$. The action of $\mathbb{Z}_2^2$ on $\{m_1, \dots, m_k\}$ has orbits $\{\Omega_1, \dots, \Omega_m\}$ of possible sizes 1, 2, and 4. By Lemma 1.4, actions of $\mathbb{Z}_2^2$ on the orbits of size 2 are either isomorphic to its action on the vertices or on the edges of an 2-gon. Thus, without a loss of generality, there are integers $0 \le t_1 \le t_2 \le t_3 \le m$ such that the orbits $\Omega_1, \dots, \Omega_{t_1}$ are of size 1, $\Omega_{t_1+1}, \dots, \Omega_{t_2}$ are of size 2, $\Omega_{t_2+1}, \dots, \Omega_{t_3}$ are of size 2, $\Omega_{t_3+1}, \dots, \Omega_m$ are of size 4, and the $\mathbb{Z}_2^2$-spaces $(\mathbb{Z}_2^2, \Omega_i)$ and $(\mathbb{Z}_2^2, \Omega_j)$, for $t_1 + 1 \le i \le t_2$ and $t_2 + 1 \le j \le t_3$, are non-isomorphic. Then, we have

$$\mathrm{Stab}_S(x) \cong \left(\prod_{i=1}^{t_1} K_i \times \prod_{i=t_1+1}^{t_2} K_i^2 \times \prod_{i=t_2+1}^{t_3} K_i^2 \times \prod_{i=t_3+1}^{m} K_i^4 \right) \rtimes \mathbb{Z}_2^2,$$

where $K_i \in \Sigma$, for $i = 1, \dots, m$, by induction. Denote $G_1 = K_1, \dots, K_{t_1}$, $G_2 = K_{t_1+1}, \dots, K_{t_2}$, $G_3 = K_{t_1+2}, \dots, K_{t_3}$, $G_4 = K_{t_1+3}, \dots, K_m$. By Lemma 1.8,

$$\mathrm{Stab}_S(x) \cong G_1 \times (K_{t_1+1}^2, \dots, K_{t_3}^2, K_{t_3+1}^4, \dots, K_m^4) \wr \mathbb{Z}_2^2.$$

By Lemma 1.9,

$$\mathrm{Stab}_S(x) \cong G_1 \times (G_2^2, G_3^2, G_4^4) \wr \mathbb{Z}_2^2,$$

which belongs to Σ. □

Now, we prove Theorem 2.2.

Proof of Theorem 2.2. We first prove that Aut(connected CIRCLE) contains all described groups. Let $G \in \Sigma$ and let Y be a connected circle graph with $\mathrm{Stab}_Y(y) \cong G$. We take n copies of Y and attach them by y to the graph depicted in Figure 2.15 on the left. Clearly, we get $\mathrm{Aut}(x) \cong G^n \rtimes \mathbb{Z}_n$. Let $G_1, G_2 \in \Sigma$ and let X_1 and X_2 be connected circle graphs such that $\mathrm{Stab}_{X_i}(x_i) \cong G_i$ and $X_1 \not\cong X_2$. We construct a graph X by attaching n copies of X_1 by x_1 and $2n$ copies of X_2 by x_2 as in Figure 2.15 on the right. We get $\mathrm{Aut}(X) \cong (G_1^n \times G_2^{2n}) \rtimes \mathbb{D}_n$.

Let X be a connected circle graph, we want to show that $\mathrm{Aut}(X)$ can be constructed in the above way. Let S be its split tree, by Lemma 2.13 we have $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$. For the central node C, we get by Lemma 2.15 that

$$\mathrm{Aut}(S) \cong \Big(\mathrm{Stab}_{S[N_1]}(m_1) \times \cdots \times \mathrm{Stab}_{S[N_k]}(m_k) \Big) \rtimes \mathrm{Aut}(C),$$

where $N_1, \dots, N_k$ are children of C and $m_1, \dots, m_k$ are their root marker vertices. By Lemma 2.18, we know that each $\mathrm{Stab}_{S[N_i]} \in \Sigma$ and also $\prod \mathrm{Stab}_{S[N_i]}(m_i) \in \Sigma$. The rest follows by analysing the automorphism group $\mathrm{Aut}(C)$ and its orbits.

Case 1: C is a degenerate node. This is exactly the same as Case 1 in the proof of Lemma 2.18. We get that $\mathrm{Aut}(S) \in \Sigma$, so it is the semidirect product with $\mathbb{Z}_1$.

Case 2: C is a prime node. By Lemma 2.16, we know that $\mathrm{Aut}(C)$ is isomorphic to either $\mathbb{Z}_n$, or $\mathbb{D}_n$. If $n \le 2$, we can show by a similar argument that $\mathrm{Aut}(S) \in \Sigma$.

If $\mathrm{Aut}(C) \cong \mathbb{Z}_n$, where $n \ge 3$, then by Lemma 2.16 we know that $\mathrm{Aut}(C)$ consists of rotations which act semiregularly. Therefore each orbit of $\mathrm{Aut}(C)$ is of size n and $\mathrm{Aut}(C)$ acts isomorphically on them. Let $G \in \Sigma$ be the direct product of $\mathrm{Stab}_{S[N_i]}(m_i)$, one for each orbit of $\mathrm{Aut}(C)$. It follows that

$$\mathrm{Aut}(S) \cong G^n \rtimes \mathrm{Aut}(C) = G \wr \mathbb{Z}_n.$$

Let $\mathrm{Aut}(C) \cong \mathbb{D}_n$, where $n \ge 2$ even. The action of $\mathbb{D}_n$ on $\{m_1, \dots, m_k\}$ has orbits $\{\Omega_1, \dots, \Omega_m\}$ of possible sizes 1, 2, and 4. By Lemma 1.4, actions of $\mathbb{D}_n$ on the orbits of size n are either isomorphic to its action on the vertices or on the edges of an n-gon. Thus, without a loss of generality, there are integers $0 \le t_1 \le t_2 \le t_3 \le m$ such that the orbits $\Omega_1, \dots, \Omega_{t_1}$ are of size 1, $\Omega_{t_1+1}, \dots, \Omega_{t_2}$ are of size n, $\Omega_{t_2+1}, \dots, \Omega_{t_3}$ are of size n, $\Omega_{t_3+1}, \dots, \Omega_m$ are of size $2n$, and the $\mathbb{D}_n$-spaces $(\mathbb{D}_n, \Omega_i)$ and $(\mathbb{D}_n, \Omega_j)$, for $t_1+1 \le i \le t_2$ and $t_2+1 \le j \le t_3$, are non-isomorphic. Then, we have

$$\mathrm{Aut}(S) \cong \left(\prod_{i=1}^{t_1} K_i \times \prod_{i=t_1+1}^{t_2} K_i^n \times \prod_{i=t_2+1}^{t_3} K_i^n \times \prod_{i=t_3+1}^{m} K_i^{2n} \right) \rtimes \mathbb{Z}_2^2,$$

where $K_i \in \Sigma$, for $i = 1, \dots, m$. Denote $G_1 = K_1, \dots, K_{t_1}$, $G_2 = K_{t_1+1}, \dots, K_{t_2}$, $G_3 = K_{t_1+2}, \dots, K_{t_3}$, $G_4 = K_{t_1+3}, \dots, K_m$. By Lemma 1.8,

$$\mathrm{Aut}(C) \cong G_1 \times (K_{t_1+1}^2, \dots, K_{t_3}^2, K_{t_3+1}^4, \dots, K_m^4) \wr\wr \mathbb{D}_n.$$

By Lemma 1.9,

$$\mathrm{Aut}(C) \cong G_1 \times (G_2^n, G_3^n, G_4^{2n}) \wr\wr \mathbb{D}_n,$$

which belongs to Aut(connected CIRCLE). The proof for the case when $n \ge 3$ and odd is the same. The only difference is that by Lemma 1.4 all orbits of size n are isomorphic. Thus in this case we get $\mathrm{Aut}(C) \cong G_1 \times (G_2^n, G_3^{2n}) \wr\wr \mathbb{D}_n$, for some $G_1, G_2, G_3 \in \Sigma$. □

2.4.5 The action on circle representations

For a connected circle graph X, the set $\mathfrak{Rep}/\sim$ consists of all circular orderings of the endpoints of the chords which give a correct representation of X. Then $\pi(\mathcal{R})$ is the representation in which the endpoints are mapped by π. The stabilizer $\mathrm{Aut}(\mathcal{R})$ can only rotate/reflect this circular ordering, so it is a subgroup of a dihedral group. For prime circle graphs, we know that $\mathrm{Aut}(\mathcal{R}) = \mathrm{Aut}(X)$. A general circle graph may have many different representations, and the action of $\mathrm{Aut}(X)$ on them may consist of several non-isomorphic orbits and $\mathrm{Aut}(\mathcal{R})$ may not be a normal subgroup of $\mathrm{Aut}(X)$.

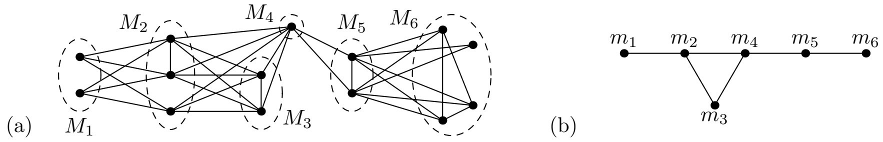


Figure 2.16: (a) A graph X and a modular partition $\mathcal{P} = \{M_1, M_2, M_3, M_4, M_5, M_6\}$. (b) The quotient graph $X/\mathcal{P}$ is prime.

The above results have the following interpretation in terms of the action of $\mathrm{Aut}(X)$. By Lemma 2.13, we know that $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$. We assume that the center C is a prime circle graph, otherwise $\mathrm{Aut}(\mathcal{R})$ is very restricted ($\mathbb{Z}_1$ or $\mathbb{Z}_2$) and not very interesting. We choose a representation $\mathcal{R}$ belonging to the smallest orbit, i.e., $\mathcal{R}$ is one of the most symmetrical representations. Then $\mathrm{Aut}(\mathcal{R})$ consists of the rotations/reflections of C described in the proof of Theorem 2.2.

The action of $\mathrm{Aut}(X)$ on this orbit is described by the point-wise stabilizer H of C in $\mathrm{Aut}(S)$. We know that $H = \prod \mathrm{Stab}_{S[N_i]}(m_i)$ as described in Lemma 2.18. When N_i is a prime graph, we can apply reflections and rotations described in Lemma 2.17, so we get a subgroup of $\mathbb{Z}_2^2$. If N_i is a degenerate graph, then isomorphic subtrees can be arbitrarily permuted which corresponds to permuting small identical parts of a circle representation. It follows that $\mathrm{Aut}(X) \cong H \rtimes \mathrm{Aut}(\mathcal{R})$.

2.5 Comparability and permutation graphs

All transitive orientations of a graph are efficiently captured by the modular decomposition which is encoded by the modular tree.

We study the induced action of $\mathrm{Aut}(X)$ on the set of all transitive orientations. We show that this action can be reconstructed from the modular tree, but for general comparability graphs the stabilizers can be arbitrary groups. 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 fixed-point-free. From this, we deduce Theorem 2.3. Finally, we show that an arbitrary graph can be encoded into a comparability graph of the dimension at most four, while preserving the automorphism group and isomorphism relation, which establishes Theorem 2.5.

2.5.1 Modular decomposition

A set $M \subseteq V(X)$ is a *module* if 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 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 2.16a 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' induce a complete bipartite subgraph, or there are no edges at all; see Figure 2.16a. 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 $m_1, \dots, m_k$ corresponding to $M_1, \dots, M_k$ where $m_i m_j \in E(X/\mathcal{P})$ if and only if M_i and M_j are adjacent. In other words, the quotient graph is obtained by contracting each module M_i into the single vertex m_i; see Figure 2.16.

Modular decomposition. To decompose X, we find some modular partition $\mathcal{P} = \{M_1, \dots, M_k\}$ of X, compute $X/\mathcal{P}$ and recursively decompose $X/\mathcal{P}$ and each $X[M_i]$, for $i = 1, \dots, k$. The recursive process terminates on *prime graphs* which are graphs containing only trivial modules. There might be many such decompositions, for different choices of $\mathcal{P}$ in each step. In 1967, Gallai [72] described the *modular decomposition* which implicitly represents all decompositions by choosing special modular partitions.

Modular decomposition is based on the observation that if M is a module of X, then $M' \subseteq M$ is a module of X if and only if it is a module of $X[M]$. A graph is called *degenerate* if it is a complete graph or an edge-less graph. We construct the modular decomposition of a graph X in the following way; see Figure 2.17:

- If X is a prime or a degenerate graph, then we terminate.
- Let X and $\overline{X}$ be connected graphs. Let $\mathcal{P}$ be the inclusion maximal proper subsets of $V(X)$ which are modules. Gallai [72] proved that $\mathcal{P}$ is a modular partition of X and $X/\mathcal{P}$ is a prime graph; see Figure 2.16. We recursively decompose $X[M]$ for each $M \in \mathcal{P}$.
- Let X be disconnected and $\overline{X}$ is connected. The connected components form a modular partition $\mathcal{P}$ of X, and the quotient graph $X/\mathcal{P}$ is an independent set. We recursively decompose $X[M]$ for each $M \in \mathcal{P}$.
- Let $\overline{X}$ be disconnected and X be connected. The connected components of $\overline{X}$ form a modular partition $\mathcal{P}$ of X and the quotient graph $X/\mathcal{P}$ is a complete graph. We recursively decompose $X[M]$ for each $M \in \mathcal{P}$.

Modular decomposition is unique [72]. This is guaranteed by the special choice of a modular partition in every step and by terminating when the graph is degenerate.

Modular tree. We encode the modular decomposition by the *modular tree* T. The modular tree T is a graph with two types of vertices: (normal and *marker vertices*) and two types of edges (normal and *directed tree edges*). The directed

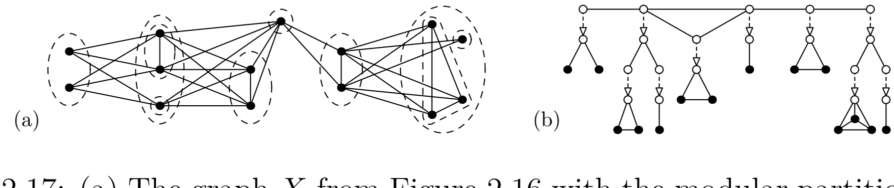


Figure 2.17: (a) The graph X from Figure 2.16 with the modular partitions used in the modular decomposition. (b) The modular tree T of X, the marker vertices are white, the tree edges are dashed.

tree edges connect the prime and degenerate graphs encountered in the modular decomposition (as quotients and terminal graphs) into a rooted tree.

We give a recursive definition. 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 equals T. Otherwise, let $\mathcal{P} = \{M_1, \dots, M_k\}$ be the used modular partition of X and let $T_1, \dots, T_k$ be the modular trees corresponding to $X[M_1], \dots, X[M_k]$. The modular tree T is the disjoint union of $T_1, \dots, T_k$ and of $X/\mathcal{P}$ with the marker vertices $m_1, \dots, m_k$. 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 oriented from m_i to m'_i. For an example, see Figure 2.17b.

The modular tree of X is unique. The graphs encountered in the modular decomposition are called *nodes of* T, or alternatively root nodes of some modular trees in the construction of T. For a node N, its subtree is the modular tree which has N as the root node. *Leaf nodes* correspond to the terminal graphs in the modular decomposition, and *inner nodes* are the quotients in the modular decomposition. All vertices of X are in leaf nodes and all marker vertices correspond to modules of X. All inner nodes consist of marker vertices.

Similarly as in Lemma 2.12, the modular tree T captures the adjacencies in X.

Lemma 2.19. *We have $xy \in E(X)$ if and only if there exists an alternating path $xm_1m_2 \dots m_ky$ 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. Both x and y belong to leaf nodes. If there exists an alternating path, let N be the node which is the common ancestor of x and y. This path has an edge $m_{2i}m_{2i+1}$ in N. These vertices correspond to adjacent modules M_{2i} and M_{2i+1} such that $x \in M_{2i}$ and $y \in M_{2i+1}$. Therefore $xy \in E(X)$.

On the other hand, let N be the common ancestor of x and y, such that m_x is the marker vertex on a path from x to N and similarly m_y is the marker vertex for y and N. If $xy \in E(X)$, then the corresponding modules M_x and M_y has to be adjacent, so we can construct an alternating path from x to y. $\square$

2.5.2 Automorphisms of modular trees

An automorphism of the modular tree T has to preserve the types of vertices and edges and the orientation of tree edges. We denote the automorphism group of T by $\mathrm{Aut}(T)$.

Lemma 2.20. *If T is the modular tree of 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. Since $V(X) \subseteq V(T)$, we define $\pi = \sigma \restriction_{V(X)}$. By Lemma 2.19, $xy \in E(X)$ if and only if there exists an alternating path in T connecting them. Automorphisms preserve alternating paths, so $xy \in E(X) \iff \pi(x)\pi(y) \in E(X)$.

For the converse, we prove that $\pi \in \mathrm{Aut}(X)$ induces a unique automorphism $\sigma \in \mathrm{Aut}(T)$. We define $\sigma \restriction_{V(X)} = \pi$ and extend it recursively on the marker vertices. Let $\mathcal{P} = \{M_1, \dots, M_k\}$ be the modular partition of X used in the modular decomposition. It is easy to see that $\mathrm{Aut}(X)$ induces an action on $\mathcal{P}$. If $\pi(M_i) = M_j$, 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. Let N be the root node of T. Suppose that we know the automorphism groups $\mathrm{Aut}(T_1), \ldots, \mathrm{Aut}(T_k)$ of the subtrees $T_1, \ldots, T_k$ of all children of N. We further color the marker vertices in N by colors coding isomorphism classes of the subtrees $T_1, \ldots, T_k$.

Lemma 2.21. *Let N be the root node of T with subtrees $T_1, \ldots, T_k$. Then*

$$\mathrm{Aut}(T) \cong \Big(\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k)\Big) \rtimes \mathrm{Aut}(N),$$

where $\mathrm{Aut}(N)$ *is color preserving.*

Proof. Recall the proof of Theorem 1.10. 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 identified with the elements of $\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k)$ and the automorphisms τ with the elements of $\mathrm{Aut}(N)$. The rest is exactly as in the proof of Theorem 1.10. □

With no further assumptions on X, if N is a prime graph, then $\mathrm{Aut}(N)$ can be isomorphic to an arbitrary group, as shown in Section 2.5.6. If N is a degenerate graph, then $\mathrm{Aut}(N)$ is a direct product of symmetric groups.

Automorphism groups of interval graphs. In Section 2.3, we proved using MPQ-trees that $\mathrm{Aut}(\mathsf{INT}) = \mathrm{Aut}(\mathsf{TREE})$. The modular decomposition gives an alternative derivation that $\mathrm{Aut}(\mathsf{INT}) \subseteq \mathrm{Aut}(\mathsf{TREE})$ by Lemma 2.21 and the following:

Lemma 2.22. *For a prime interval graph X,* $\mathrm{Aut}(X)$ *is a subgroup of* $\mathbb{Z}_2$.

Proof. Hsu [96] proved that prime interval graphs have exactly two consecutive orderings of the maximal cliques. Since X has no twin vertices, $\mathrm{Aut}(X)$ acts semiregularly on the consecutive orderings and there is at most one non-trivial automorphism in $\mathrm{Aut}(X)$. □

2.5.3 Automorphism groups of comparability graphs

In this section, we explain the structure 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. 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 [72] shows the following properties. If M_1 and M_2 are adjacent modules of a partition used in the modular decomposition, then either $M_1 \to M_2$, or $M_1 \leftarrow M_2$. The graph X is a comparability graph

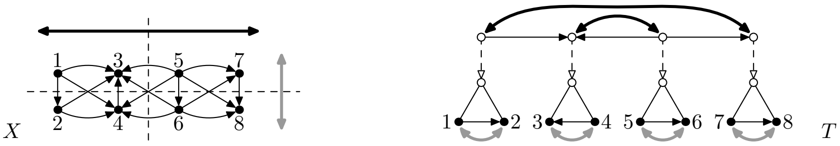


Figure 2.18: Two automorphisms reflect X and change the transitive orientation, and their action on the modular tree T.

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 arbitrarily choose one of the two possible orientations. For each degenerate node, we choose some orientation. (Where K_n has $n!$ possible orientations and $\overline{K_n}$ has the unique orientation.) A transitive orientation of X is then constructed as follows. We orient the edges of leaf nodes as above. For a node N partitioned in the modular decomposition by $\mathcal{P} = \{M_1, \dots, M_k\}$, we orient $X[M_i] \to X[M_j]$ if and only if $m_i \to m_j$ in N. It is easy to check that this gives a valid transitive orientation, and every transitive orientation can be constructed by some orientations of the nodes of T. We note that this implies that the dimension of the transitive orientation is the maximum over the dimensions of all nodes of T, and that this dimension is the same for every transitive orientation.

Action induced on transitive orientations. Let $\mathfrak{to}(X)$ be the set of all transitive orientations of X. Let $\pi \in \mathrm{Aut}(X)$ and $\to \in \mathfrak{to}(X)$. We define the orientation $\pi(\to)$ as follows:

$$x \to y \implies \pi(x)\,\pi(\to)\,\pi(y), \qquad \forall x, y \in V(X).$$

We can observe that $\pi(\to)$ is a transitive orientation of X, so $\pi(\to) \in \mathfrak{to}(X)$; see Figure 2.18. It easily follows that $\mathrm{Aut}(X)$ defines an action on $\mathfrak{to}(X)$.

Let $\mathrm{Stab}(\to)$ be the stabilizer of some orientation $\to \in \mathfrak{to}(X)$. It consists of all automorphisms which preserve this orientation, so only the vertices that are incomparable in $\to$ can be permuted. In other words, $\mathrm{Stab}(\to)$ is the automorphism group of the poset created from the transitive orientation $\to$ of X. Since posets are universal [18, 155], $\mathrm{Stab}(\to)$ can be arbitrary group and in general the structure of $\mathrm{Aut}(X)$ cannot be derived from its action on $\mathfrak{to}(X)$.

Lemma 2.21 allows to understand it in terms of $\mathrm{Aut}(T)$ for the modular tree T representing X. Each automorphism of $\mathrm{Aut}(T)$ somehow acts inside each node, and somehow permutes the attached subtrees. Consider a node N with attached subtrees $T_1, \dots, T_k$. If $\sigma \in \mathrm{Stab}(\to)$, then it preserves the orientation in N. Therefore if it maps T_i to $\sigma(T_i)$, the corresponding marker vertices are necessarily incomparable in N. If N is an independent set, the isomorphic subtrees can be arbitrarily permuted in $\mathrm{Stab}(\to)$. If N is a complete graph, all subtrees are preserved in $\mathrm{Stab}(\to)$. If N is a prime graph, then isomorphic subtrees of incomparable marker vertices can be permuted according to the structure of N which can be complex.

It is easy to observe that stabilizers of all orientations are the same and that $\mathrm{Stab}(\to)$ is a normal subgroup. Let $H = \mathrm{Aut}(X)/\,\mathrm{Stab}(\to)$, so H captures the action of $\mathrm{Aut}(X)$ on $\mathfrak{to}(X)$. This quotient group can be constructed recursively from

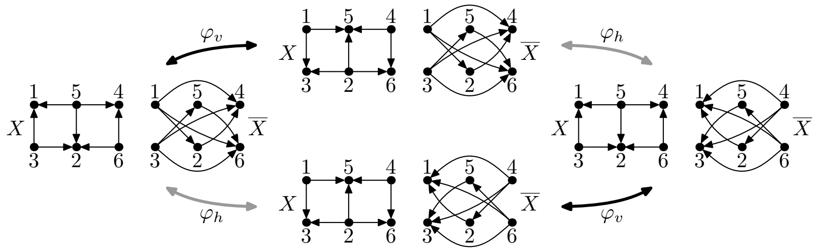


Figure 2.19: The action of $\mathrm{Aut}(X)$ on four pairs of transitive orientations X. The automorphism φ_v flips the orientation of X, the automorphism φ_h flips the orientation of both X and $\overline{X}$.

the structure of T, similarly to Lemma 2.21. Suppose that we know $H_1, \dots, H_k$ of the subtrees $T_1, \dots, T_k$. If N is an independent set, there is exactly one transitive orientation, so $H \cong H_1 \times \cdots \times H_k$. If N is a complete graph, isomorphic subtrees can be arbitrarily permuted, so H can be constructed exactly as in Theorem 1.10. If N is a prime node, there are exactly two transitive orientations. If there exists an automorphism changing the orientation of N, we can describe H by a semidirect product with $\mathbb{Z}_2$ as in Theorem 1.10. And if N is asymmetric, then $H \cong H_1 \times \cdots \times H_k$. In particular, this description implies that $H \in \mathrm{Aut}(\mathsf{TREE})$.

2.5.4 Automorphism groups of permutation graphs

In this section, we derive the characterization of $\mathrm{Aut}(\mathsf{PERM})$ stated in Theorem 2.3.

Action induced on pairs of transitive orientations. Let X be a permutation graph. In comparison to general comparability graphs, the main difference is that both X and $\overline{X}$ are comparability graphs. From the results of Section 2.5.3 it follows that $\mathrm{Aut}(X)$ induces an action on both $\mathfrak{to}(X)$ and $\mathfrak{to}(\overline{X})$. Let $\mathfrak{to}(X,\overline{X}) = \mathfrak{to}(X) \times \mathfrak{to}(\overline{X})$, and we work with one action on the pairs $(\rightarrow, \rightrightarrows) \in \mathfrak{to}(X,\overline{X})$. Figure 2.19 shows an example.

Lemma 2.23. *For a permutation graph X, the action of* $\mathrm{Aut}(X)$ *on* $\mathfrak{to}(X,\overline{X})$ *is semiregular.*

Proof. Since a permutation belonging to the stabilizer of $(\rightarrow, \rightrightarrows)$ fixes both orientations, it can only permute incomparable elements. But incomparable elements in $\rightarrow$ are exactly the comparable elements in $\rightrightarrows$, so the stabilizer is trivial. $\square$

Lemma 2.24. *For a prime permutation graph X,* $\mathrm{Aut}(X)$ *is a subgroup of $\mathbb{Z}_2^2$.*

Proof. There are at most four pairs of orientations in $\mathfrak{to}(X,\overline{X})$, so by Lemma 2.23 the order of $\mathrm{Aut}(X)$ is at most four. If $\pi \in \mathrm{Aut}(X)$, then π^2 fixes the orientations of both X and $\overline{X}$. Therefore π^2 belongs to the stabilizers and it is an identity. Thus π is an involution and $\mathrm{Aut}(X)$ is a subgroup of $\mathbb{Z}_2^2$. $\square$

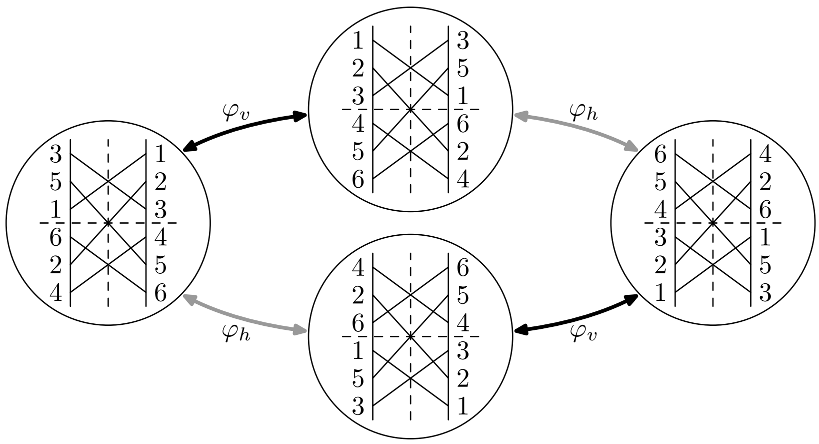


Figure 2.20: Four representations of a symmetric permutation graph. The automorphism φ_v is the vertical reflection, the automorphism φ_h is the horizontal reflection.

Geometric interpretation. First, we explain the result $\mathsf{PERM} = 2\text{-}\mathsf{DIM}$ of Even et al. [63]. Let $\rightarrow\, \in \mathfrak{to}(X)$ and $\rightrightarrows\, \in \mathfrak{to}(\overline{X})$, and let $\rightrightarrows_R$ be the reversal of $\rightrightarrows$. We construct two linear orderings $L_1 = \rightarrow \cup \rightrightarrows$ and $L_2 = \rightarrow \cup \rightrightarrows_R$. 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 vertical reflection φ_v corresponds to exchanging L_1 and L_2, which is equivalent to reversing $\rightrightarrows$. The horizontal reflection φ_h corresponds to reversing both L_1 and L_2, which is equivalent to reversing both $\rightarrow$ and $\rightrightarrows$. We denote the central 180° rotation by $\rho = \varphi_h \cdot \varphi_v$ which corresponds to reversing $\rightarrow$; see Figure 2.20.

The inductive characterization. Now, we are ready to prove Theorem 2.3.

Proof of Theorem 2.3. First, we show that $\mathrm{Aut}(\mathsf{PERM})$ is closed under (b) to (d). For (b), let $G_1, G_2 \in \mathrm{Aut}(\mathsf{PERM})$, and let X_1 and X_2 be two permutation graphs such that $\mathrm{Aut}(X_i) \cong G_i$. We construct X by attaching X_1 and X_2 as in Figure 2.21b. Clearly, $\mathrm{Aut}(X) \cong G_1 \times G_2$. For (c), let $G \in \mathrm{Aut}(\mathsf{PERM})$ and let Y be a connected permutation graph such that $\mathrm{Aut}(Y) \cong G$. We construct X as the disjoint union of n copies of Y; see Figure 2.21c. 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 $\mathrm{Aut}(X_i) \cong G_i$. We construct X as in Figure 2.21d. We get $\mathrm{Aut}(X) \cong (G_1^4, G_2^2, G_3^2) \wr\wr \mathbb{Z}_2^2$.

We show the other implication by induction. Let X be a permutation graph and let T be the modular tree representing X. By Lemma 2.20, we know that $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$. Let N be the root node of T, and let $T_1, \dots, T_k$ be the subtrees attached to N. By the induction hypothesis, we assume that $\mathrm{Aut}(T_i) \in$

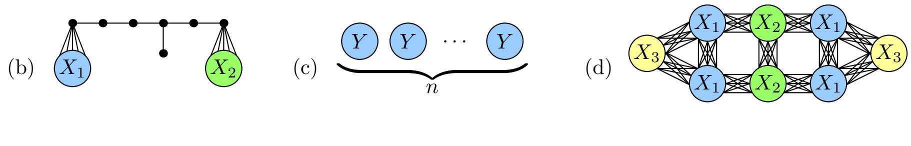


Figure 2.21: The constructions in the proof of Theorem 2.3.

Aut(PERM). By Lemma 2.21,

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

Case 1: N is a degenerate node. Then $\mathrm{Aut}(N)$ is a direct product of symmetric groups. The subtrees attached to marker vertices of each color class can be arbitrarily permuted, independently of each other. Therefore $\mathrm{Aut}(T)$ can be constructed using (b) and (c), exactly as in Theorem 1.10.

Case 2: N is a prime node. By Lemma 2.24, $\mathrm{Aut}(N)$ is a subgroup of $\mathbb{Z}_2^2$. If it is trivial or $\mathbb{Z}_2$, observe that it can be constructed using (b) and (c). The only remaining case is when $\mathrm{Aut}(N) \cong \mathbb{Z}_2^2$. The action of $\mathbb{Z}_2^2$ on $V(N)$ has orbits $\{\Omega_1, \dots, \Omega_m\}$ of possible sizes 1, 2, and 4. By Lemma 1.4, actions of $\mathbb{Z}_2^2$ on the orbits of size 2 are either isomorphic to its action on the vertices or on the edges of an 2-gon. Thus, without a loss of generality, there are integers $0 \le t_1 \le t_2 \le t_3 \le m$ such that the orbits $\Omega_1, \dots, \Omega_{t_1}$ are of size 1, $\Omega_{t_1+1}, \dots, \Omega_{t_2}$ are of size 2, $\Omega_{t_2+1}, \dots, \Omega_{t_3}$ are of size 2, $\Omega_{t_3+1}, \dots, \Omega_m$ are of size 4, and the $\mathbb{Z}_2^2$-spaces $(\mathbb{Z}_2^2, \Omega_i)$ and $(\mathbb{Z}_2^2, \Omega_j)$, for $t_1 + 1 \le i \le t_2$ and $t_2 + 1 \le j \le t_3$, are non-isomorphic. Then, we have

$$\mathrm{Aut}(T) \cong \left(\prod_{i=1}^{t_1} K_i \times \prod_{i=t_1+1}^{t_2} K_i^2 \times \prod_{i=t_2+1}^{t_3} K_i^2 \times \prod_{i=t_3+1}^{m} K_i^4\right) \rtimes \mathbb{Z}_2^2,$$

where $K_i \in \Sigma$, for $i = 1, \dots, m$, by induction. Denote $G_1 = K_1, \dots, K_{t_1}$, $G_2 = K_{t_1+1}, \dots, K_{t_2}$, $G_3 = K_{t_1+2}, \dots, K_{t_3}$, $G_4 = K_{t_1+3}, \dots, K_m$. By Lemma 1.8,

$$\mathrm{Aut}(T) \cong G_1 \times (K_{t_1+1}^2, \dots, K_{t_3}^2, K_{t_3+1}^4, \dots, K_m^4) \wr\wr \mathbb{Z}_2^2.$$

By Lemma 1.9,

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

which belongs to Aut(PERM). $\square$

2.5.5 Bipartite permutation graphs

We use the modular trees to characterize Aut(connected BIP PERM). For a connected bipartite graph, every non-trivial module is an independent set, and the quotient is a prime bipartite graph. Therefore, the modular tree T has a prime root node N, to which there are attached leaf nodes which are independent sets.

Proof of Corollary 2.4. Every abstract group from Corollary 2.4 can be constructed as shown in Figure 2.22. Let T be the modular tree representing X. By Lemmas 2.20 and 2.21,

$$\mathrm{Aut}(X) \cong \Big(\mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_k)\Big) \rtimes \mathrm{Aut}(N),$$

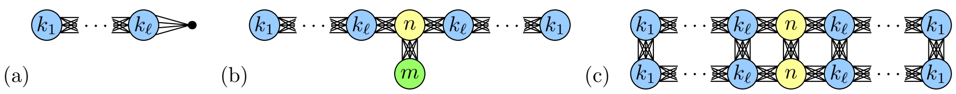


Figure 2.22: Let $G_1 = \mathbb{S}_{k_1} \times \cdots \times \mathbb{S}_{k_\ell}$, $G_2 = \mathbb{S}_n$ and $G_3 = \mathbb{S}_m$. The constructed graphs consist of independent sets of given sizes joined by complete bipartite subgraphs. They have the following automorphism groups: (a) G_1, (b) $G_1 \wr \mathbb{Z}_2 \times G_2 \times G_3$, (c) $(G_1^4, G_2^2) \wr\wr \mathbb{Z}_2^2$.

where $\mathrm{Aut}(N)$ is isomorphic to a subgroup of $\mathbb{Z}_2^2$ (by Lemma 2.24), and each $\mathrm{Aut}(T_i)$ is a symmetric group since T_i is an independent set.

Consider a permutation representation of N in which the endpoints of the segments, representing $V(N)$, are placed equidistantly as in Figure 2.20. By [142], there are no segments parallel with the horizontal axis, so the reflections φ_v and φ_h fix no segment. Further, since N is bipartite, there are at most two segments crossing the central point, so the rotation ρ can fix at most two segments.

Case 1: $\mathrm{Aut}(N)$ *is trivial.* Then $\mathrm{Aut}(X)$ is a direct product of symmetric groups.

Case 2: $\mathrm{Aut}(N) \cong \mathbb{Z}_2$. Let G_1 be the direct product of all $\mathrm{Aut}(T_i)$, one for each orbit of size two. Notice that $\mathrm{Aut}(N)$ is generated by exactly one of φ_v, φ_h, and ρ. For φ_v or φ_h, all orbits are of size two, so $\mathrm{Aut}(X) \cong G_1 \wr \mathbb{Z}_2$. For ρ, there are at most two fixed segments, so $\mathrm{Aut}(X) \cong G_1 \wr \mathbb{Z}_2 \times G_2 \times G_3$, where G_2 and G_3 are isomorphic to $\mathrm{Aut}(T_i)$, for each of two orbits of size one.

Case 3: $\mathrm{Aut}(N) \cong \mathbb{Z}_2^2$. Then $\mathrm{Aut}(N)$ has no orbits of size 1, at most one of size 2, and all other of size 4. Let G_1 be the direct product of all $\mathrm{Aut}(T_i)$, one for each orbit of size 4, and let G_2 be $\mathrm{Aut}(T_i)$ for the orbit of size 2. We have $\mathrm{Aut}(X) \cong (G_1^4, G_2^2) \wr\wr \mathbb{Z}_2^2$. $\square$

2.5.6 Comparability graphs of dimension k

In this section, we prove that $\mathrm{Aut}(4\text{-}\mathsf{DIM})$ contains all abstract 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 $4\text{-}\mathsf{DIM}$ is GI-complete. Both results easily translate to $k\text{-}\mathsf{DIM}$ for $k > 4$ since $4\text{-}\mathsf{DIM} \subsetneq k\text{-}\mathsf{DIM}$.

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 2.23:

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

So C_X is created from X by replacing each edge with a path of length four.

Lemma 2.25. *For a connected graph $X \not\cong C_n$,* $\mathrm{Aut}(C_X) \cong \mathrm{Aut}(X)$. $\square$

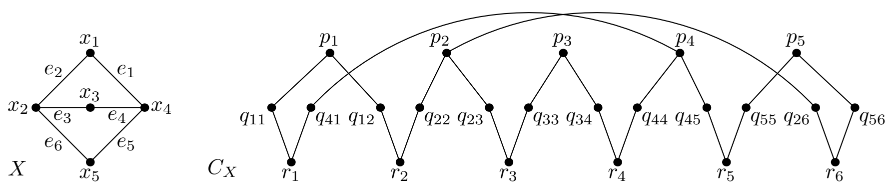


Figure 2.23: The construction C_X for the graph $X = K_{2,3}$.

Lemma 2.26. *If X is a connected bipartite graph, then* $\dim(C_X) \le 4$.

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 words, 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 an arrow is 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 the concatenation $I_j I_i$ in some other chain.

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

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

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

We construct the four chains as follows:

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

The four defined chains have the following properties, see Figure 2.24:

- 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.

- 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.

It is routine to verify that the intersection $L_1 \cap L_2 \cap L_3 \cap L_4$ is correct.

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.

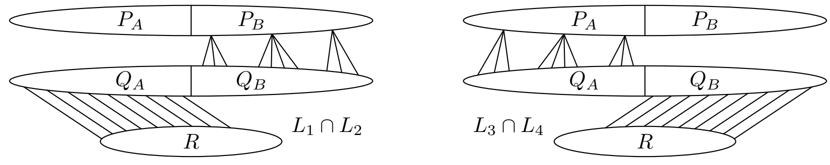


Figure 2.24: The forced edges in $L_1 \cap L_2$ and $L_3 \cap L_4$.

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 orderings 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. $\diamond$

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. $\diamond$

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 $p_i \in P_B$ 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. $\diamond$

These three established 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 $C_X \in 4\text{-}\mathsf{DIM}$. $\square$

Universality of k-DIM. We are ready to prove Theorem 2.5.

Proof of Theorem 2.5. We prove the statement for $4\text{-}\mathsf{DIM}$. Let X be a connected graph such that $X \ncong C_n$. First, we construct the bipartite incidence graph Y between $V(X)$ and $E(X)$. Next, we construct C_Y from Y. From Lemma 2.25 it follows that $\mathrm{Aut}(C_Y) \cong \mathrm{Aut}(Y) \cong \mathrm{Aut}(X)$ and by Lemma 2.26, we have that $C_Y \in 4\text{-}\mathsf{DIM}$. 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 polynomial-time reduction shows GI-completeness of graph isomorphism testing.

The constructed graph C_Y is a prime graph. We fix the transitive orientation in which P and R are the minimal elements and get the poset P_Y with $\mathrm{Aut}(P_Y) \cong \mathrm{Aut}(C_Y)$. Hence, our results translate to posets of the dimension at most four. $\square$

2.6 Algorithms

In this section, we briefly explain algorithmic implications of our results. We first describe the classical algorithm for graph isomorphism of rooted trees. We process both trees from the leaves to the root and compute partial isomorphisms between processed subtrees, using dynamic programming. Every leaf of X can be mapped to every leaf of Y. Consider a subtree T of X with subtrees $T_1, \dots, T_k$ attached to the root, and another subtree T' of Y with subtrees $T'_1, \dots, T'_k$ attached to the root. We want to decide whether $T \cong T'$, i.e, whether it is possible to isomorphically pair $T_1, \dots, T_k$ with $T'_1, \dots, T'_k$. For an efficient linear-time implementation, we assign colors to processes subtrees, coding isomorphism classes.

We just compare sizes of color classes of $T_1, \dots, T_k$ with the sizes of color classes of $T'_1, \dots, T'_k$.

Colbourn and Booth [39] describe how this algorithm can be modified to compute permutation generators of automorphism groups of trees in linear time, in terms of permutation generators. It follows from our work that automorphism groups of interval graphs, circle graphs and permutation graphs are captured by MPQ-trees, split trees and modular trees, respectively. Therefore, we can use these data structures to compute automorphism groups efficiently.

Our structural results allow to represent these automorphism groups by rooted trees called *group product trees*. Their leaves correspond to trivial groups $\{1\}$, acting on pairwise disjoint subsets of vertices. Their inner nodes correspond one of the following operations applied on the attached subtrees: the direct product, the wreath product with $\mathbb{Z}_n$ or $\mathbb{D}_n$, the semidirect product with $\mathbb{D}_n$ or $\mathbb{Z}_2^2$. Group product trees describe $\mathrm{Aut}(X)$ much better than lists of permutations generators which can be easily computed from them. Many tools of the computational group theory are devoted to getting better understanding of an unknown group, and group product trees give this structural understanding for free.

Interval graphs. Using MPQ-trees, Colbourn and Booth [39] give a linear-time algorithm to compute permutation generators of automorphism groups of interval graphs. Using our structural results described in Section 2.3, we can modify this algorithm to compute group product trees of automorphism groups of interval graphs in linear time, and we sketch this below (missing details can be found in [39]). For an input interval graph X, its MPQ-tree M can be computed in linear time [112]. The algorithm processes M from the bottom to the root. For a processed node, we construct $\mathrm{Aut}(X)$ from the automorphism groups of its children $\mathrm{Aut}(X_1), \dots, \mathrm{Aut}(X_k)$, as described in the proof of Theorem 2.1(i).

Corollary 2.27. *For an interval graph X, we can compute a group product tree of $\mathrm{Aut}(X)$ in time $\emptyset(n+m)$, where $n = |V(X)|$ and $m = |E(X)|$.* □

Circle graphs. Hsu [96] describes an algorithm solving the graph isomorphism problem of circle graphs based on the minimal split decomposition in time $\mathcal{O}(n^3)$. The running time was recently improved to almost linear-time [98]. It is straightforward to modify the latter algorithm to compute generators of the automorphism groups in the same running time.

Permutation graphs. Colbourn [38] described an $\emptyset(n^3)$ algorithm for graph isomorphism of permutation graphs. This was improved by Spinrad [146] to $\emptyset(n^2)$ by computing modular decompositions and testing tree isomorphism on them. The bottleneck of this algorithm is computing the modular decomposition, so by combining with [126], the running time is improved to $\emptyset(n+m)$. We are not aware of any previous polynomial-time algorithm for computing automorphism groups of permutation graphs, aside applying the above algorithms together with [124].

Corollary 2.28. *For a permutation graph X, we can compute a group product tree of $\mathrm{Aut}(X)$ in time $\emptyset(n+m)$, where $n = |V(X)|$ and $m = |E(X)|$.*

Proof. We find the modular tree T of X. For each prime permutation node N, we finding at most four pairs of transitive orientations $\mathfrak{to}(N, \overline{N})$. For every pair $(\rightarrow, \rightrightarrows)$, we construct the linear ordering $\rightarrow \cap \rightrightarrows$ of endpoints of segments on a line in a representation. This gives four linear orderings L_1, L_2, L_3, and L_4 which correspond to different labelings of vertices. All above can be computed in linear time using [126]. For each pair L_i and L_j, we check whether $\pi_{i,j}$ mapping the k-th vertex of L_i to the k-th vertex of L_j is an automorphism. Since $\mathrm{Aut}(N)$ acts semiregularly on $\mathfrak{to}(N, \overline{N})$, by Lemma 2.23, these automorphism generate $\mathrm{Aut}(N)$.

We apply the following bottom-up procedure which computes colors of marker vertices, coding isomorphism classes of their attached subtrees. Suppose that we process nodes on one level in the modular tree and let $N_1, \dots, N_k$ be all their children, already processed before with their marker vertices colored by colors $1, \dots, c$.

Case 1: Degenerate nodes. For each degenerate node with ℓ vertices, we consider a vector $\boldsymbol{v} = (t, c_1, c_2, \dots, c_\ell)$, where t is the type of the node (complete/independent) and $c_1, \dots, c_\ell$ are the sorted colors of its vertices. We sort these vectors $\boldsymbol{v}$ of all degenerate nodes lexicographically and we assign new colors $1, \dots, c'$ to the computed isomorphism classes.

Case 2: Prime nodes. For each prime node with ℓ edges, we have four possible labelings of vertices $L_1, \dots, L_4$. Each of them defines a vector $\boldsymbol{v} = \Big\{(x_i, y_i, c_i)\Big\}_{i=1}^{\ell}$, where $x_i y_i$ is the i-th edge. We use the vector $\boldsymbol{v}$ of an ordering L_j which is lexicographically minimal. We compute these vectors $\boldsymbol{v}$ for each prime node, sort them lexicographically and assign colors $c'+1, \dots, c''$ to the obtain isomorphism classes. There exists a color preserving isomorphism between two prime graphs if and only if their vectors $\boldsymbol{v}$ are the same.

Using Bucket sort, the above procedure can be implemented in linear time. For each node N we compute the color-preserving automorphism group $\mathrm{Aut}(N)$. We apply Lemma 2.21 and we output $\mathrm{Aut}(X)$ by a group product tree. The size of this tree is linear in the size of X since every vertex and module of the modular decomposition is represented by constantly many vertices of the tree. This algorithm can be implemented in linear time. □

2.7 Open problems

We conclude this chapter with several open problems concerning automorphism groups of other intersection-defined classes of graphs; for an overview see [78, 148].

Circular-arc graphs (CIRCULAR-ARC) are intersection graphs of circular arcs and they naturally generalize interval graphs. Surprisingly, this class is very complex and quite different from interval graphs. Hsu [96] relates them to circle graphs.

Problem 1. *What is* Aut(CIRCULAR-ARC)*?*

The last conjecture involves the open case of 3-DIM.

Conjecture 2.29. *The class* 3-DIM *is universal and its graph isomorphism problem is* GI*-complete.*

3. Isomorphism of circle graphs in almost linear time

3.1 Introduction

Recall that, for a graph X, a *circle representation* $\mathcal{R}$ of a graph X is a collection of sets $\{\langle v\rangle : v \in V(X)\}$ such that each $\langle v\rangle$ is a chord of some fixed circle, and $\langle u\rangle \cap \langle v\rangle \neq \emptyset$ if and only if $uv \in E(X)$. Observe that $\mathcal{R}$ is determined by the circular word giving the clockwise order of endpoints of the chords in which $uv \in E(X)$ if and only if their endpoints alternate as $uvuv$ in this word. A graph is called a *circle graph* if and only if it has a circle representation; see Fig. 3.1.

Circle graphs were introduced by Even and Itai [62] in the early 1970s. They are related to Gauss words [53], matroid representations [24, 52], and rank-width [136]. The complexity of recognition of circle graphs was a long-standing open problem, resolved in mid-1980s [23, 70, 130]. Currently, the fastest recognition algorithm [76] runs in almost linear time. We use this recognition algorithm as a subroutine and solve the graph isomorphism problem of circle graphs in the same running time.

Theorem 3.1. *The graph isomorphism problem of circle graphs and the canonization problem of circle graphs can be solved in time $\mathcal{O}((n+m)\cdot\alpha(n+m))$, where n is the number of vertices, m is the number of edges, and α is the inverse Ackermann function. Further, if circle representations are given as a part of the input, the running time improves to $\mathcal{O}(n+m)$.*

Two circle representation are isomorphic if by relabeling the endpoints we get identical circular orderings. In Section 3.4, we show that isomorphism of circle representations can be tested in time $\mathcal{O}(n)$. When circle graphs X and Y have isomorphic circle representations $\mathcal{R}_X$ and $\mathcal{R}_Y$, clearly $X \cong Y$. But in general, the converse does not hold since a circle graph may have many non-isomorphic circle representations.

The main tool is the split decomposition which is a recursive process decomposing a graph into several indecomposable graphs called *prime graphs*. Each split decomposition can be described by a split tree whose nodes are the prime graphs on which the decomposition terminates. The key property is that the initial graph is a circle graph if and only if all prime graphs are circle graphs. Further, each prime circle graph has a unique representation up to reversal, so

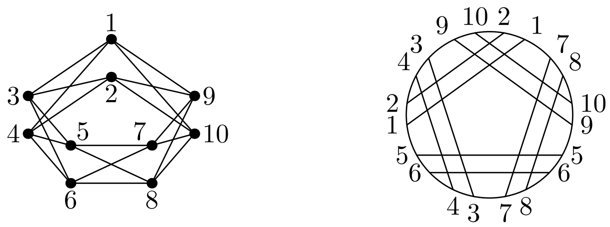


Figure 3.1: A circle graph and one of its circle representations corresponding to the circular word $10, 2, 1, 7, 8, 10, 9, 5, 6, 8, 7, 3, 4, 6, 5, 1, 2, 4, 3, 9$.

isomorphism for them can be tested in $\mathcal{O}(n)$, using the approach described in Section 3.4.

One might want to reduce the isomorphism problem of circle graphs to the isomorphism problem of split trees. Unfortunately, a graph may posses many different split decompositions corresponding to non-isomorphic split trees. The seminal paper of Cunnigham [48, Theorem 3] shows that for every connected graph, there exists a minimal split decomposition; this result was also proven in [49, Theorem 11]. The split tree associated to the minimal split decomposition is then also unique and it follows that the isomorphism problem of circle graphs reduces to the isomorphism problem of minimal split trees.

This approach was used by Hsu [96] to solve the graph isomorphism problem of circle graphs in time $\mathcal{O}(nm)$. He actually concentrates on circular-arc graphs, which are intersection graphs of circular arcs, and builds a decomposition technique which generalizes minimal split decomposition. The main results are recognition and graph isomorphism algorithms for circular-arc graphs running in $\mathcal{O}(nm)$. Unfortunately, a mistake in this general decomposition technique was pointed out by [50]. This mistake does not affect the graph isomorphism algorithm for circle graphs. Moreover, as pointed out in [50, page 180], the complexity of Hsu's algorithm improves to $\mathcal{O}((n+m)\cdot\alpha(n+m))$ if the fastest recognition algorithm of circle graphs [76] is employed. In [50, page 180], it is stated: "*If chord models are given as an input, then the running time of the isomorphism test can be reduced to* $\mathcal{O}(n+m)$ *using techniques similar to those used in [115] and in our paper*". In this chapter, we fill in those details.

Our method, outlined above, is standart and similar to all isomorphism algorithms for graph classes admiting tree-like decompositions. These include, for example, planar graphs (SPQR-trees), interval graphs (PQ-trees) and others. Then roughly speaking, the whole algorithm consists of testing isomorphism of the indecomposable parts and applying an algorithm similar to the one for trees. However, each graph class is its own nuances which need to be addressed. For instance, in [50], the authors solve the isomorphism problem for several subclasses of circular-arc graphs by reducing the problem to testing isomorphism of PC-trees.

A well-known subclass of circle graphs are proper circular-arc graphs, which are intersection graphs of circular arcs such that no arc is properly contained in another one. The methods used in the linear-time algorithm testing isomorphism of proper circular-arc graphs, given in [115], are similar to the case of prime circle graphs (see Section 3.4). In particular, co-bipartite circular-arc graphs have unique representations and, in this case, the problem can be reduced to finding minimal circular string.

Outline of the chapter. Section 3.2 gives an overview of minimal split decomposition and minimal split trees. In Section 3.3, we describe a meta-algorithm computing cannonical form of a general tree whose nodes are labeled by graphs for which a linear-time cannonization is known. For a circle graph, its unique minimal split tree is labeled by prime circle graphs and degenerate graphs (complete graphs and stars). In Section 3.4, we give linear-time cannonization algorithms for prime and degenerate circle graphs. By putting this together in Section 3.5, we prove Theorem 3.1. In Finally, we discuss related results and open problems.

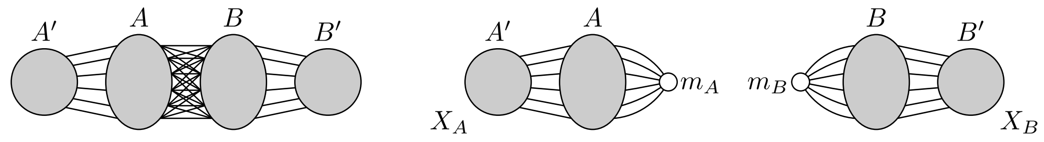


Figure 3.2: On the left, a split in X between A and B. On the right, application of this split produces graphs X_A and X_B with newly created marker vertices denoted by big white circles.

3.2 Minimal split decomposition and split trees

In this section, we describe several known properties of split decompositions and split trees. We assume that all graphs are connected, otherwise split decomposition is applied independently on each component.

Splits. For a graph X, a *split* is a partition (A, B, A', B') of $V(X)$ 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 $|B \cup B'| \geq 2$.

See Fig. 3.2 on the left. In other words, the cut between A and B is the complete bipartite graph. A split in a graph can be found in polynomial time [147]. Graphs containing no splits are called *prime graphs*. Since the sets A and B already uniquely determine the split, we call it the *split between A and B*.

We can *apply* a split between A and B to divide the graph X into two graphs X_A and X_B defined as follows. The graph X_A is created from $X[A \cup A']$ together with a new *marker vertex* m_A adjacent exactly to the vertices in A. The graph X_B is defined analogously for B, B' and m_B. See Fig. 3.2 on the right.

Split decomposition and split trees. A split decomposition D of X is a sequence of splits defined as follows. At the beginning, we start with the graph X. In the k-th step, we have graphs $X_1, \dots, X_k$ and we apply a split on some X_i, dividing it into two graphs X_i' and X_i''. The next step then applies to one of the graphs $X_1, \dots, X_{i-1}, X_i', X_i'', X_{i+1}, \dots, X_k$, and so on.

A split decomposition can be captured by a graph-labeled tree T. The vertices of T are called nodes to distinguish them from the vertices of X and from the added marker vertices, and nodes correspond to subsets of these vertices. To simplify the definition of graph isomorphism of graph-labeled trees, we give a slightly different formal definition in which T is not necessarily a tree.

Definition 3.2. *A graph-labeled tree T is a graph (V, E) with $E = E_N \dot{\cup} E_T$ where E_N are called the* normal edges *and E_T are called the* tree edges*. A* node *is a connected component of (V, E_N). There are no tree edges between the vertices of one node and no vertex is incident to two tree edges. The incidence graph of nodes must form a tree. The* size *of T is $|V| + |E|$.*

A graph-labeled tree T might not be a tree, but the underlying structure of tree edges forms a tree of nodes. The vertices of T incident to tree edges are called *marker vertices*.

A split decomposition D of X is represented by the following graph-labeled tree T called the *split tree T of D* (or a split tree T of X). Initially, T consists of a single node equal to X. At each step, D applies a split on one node N of T. This node is replaced by two new nodes N_A and N_B while the tree edges incident to N are preserved in N_A and N_B and the marker vertices m_A and m_B are further adjacent by a newly formed tree edge. Figure 3.3 shows an example. It can be observed that the total size of every split tree is $\mathcal{O}(n+m)$ where n is the number of vertices and m is the number of edges of the original graph.

From a split tree T, the original graph X can be reconstructed by *joining neighboring nodes*. For a tree edge $m_A m_B$, we remove m_A and m_B while adding all edges uv for $u \in N(m_A)$ and $v \in N(m_B)$.

Recognition of circle graphs. A split decomposition can be applied to recognize circle graphs. The key is the following observation.

Lemma 3.3. *A graph is a circle graph if and only if both X_A and X_B are circle graphs.*

The proof is illustrated in Figure 3.4, which can be easily formalized; see for example [148]. A prime circle graph has a unique circle representation up to reversal [51] which can be constructed in polynomial time [76].

Minimal split decomposition. A graph is called *degenerate* if it is the complete graph K_n or the star S_n. Suppose that we have a split decomposition D ending on prime graphs. Its split tree is not uniquely determined, for instance degenerate graphs have many different split trees. Cunnignham [48] resolved this issue by terminating the split decomposition not only on prime graphs, but also on the degenerate graphs.

Cunnignham [48] introduced the notion of a *minimal split decomposition.* A split decomposition is minimal if the corresponding split tree has all nodes as prime and degenerate graphs, and joining any two neighboring nodes creates a non-degenerate graph.

Theorem 3.4 (Cunningham [48], Theorem 3)**.** *For a connected graph X, the split tree of a minimal split decomposition terminating on prime and degenerate graphs is uniquely determined.*

The split tree of a minimal split decomposition of X is called the *minimal split tree* of X and it is denoted T_X.

It was stated in [77, Theorem 2.17] that a split decomposition is minimal if the corresponding split tree has no two neighboring nodes such that

- either both are complete,
- or both are stars and the tree edge joining these stars is incident to exactly one of their central vertices.

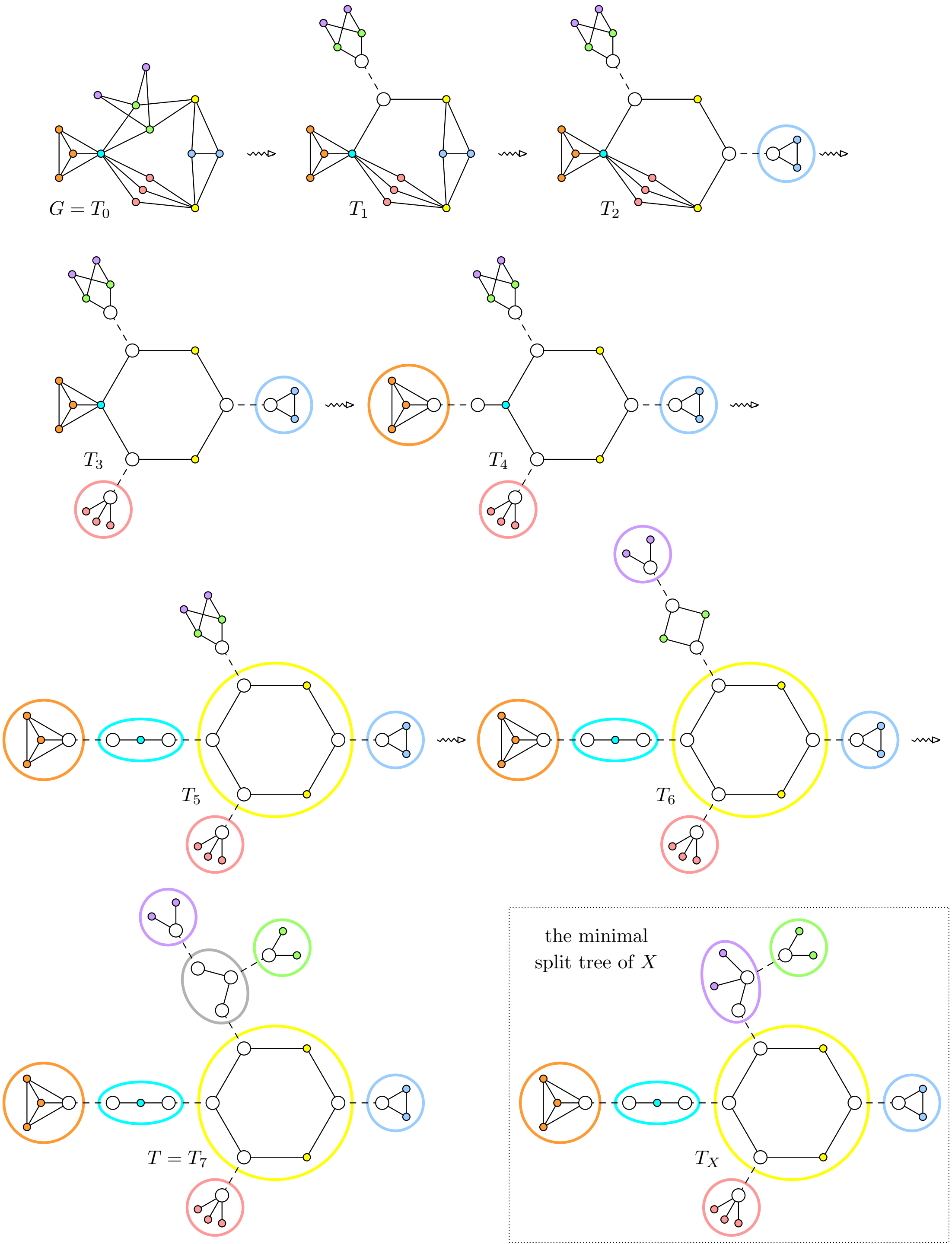


Figure 3.3: An example of a split tree T of a split decomposition D terminating with highlighted prime and degenerate graphs (see the definition below). The split decomposition D is not minimal: the gray and purple stars can be joined in T to form the minimal split tree T_X in the box.

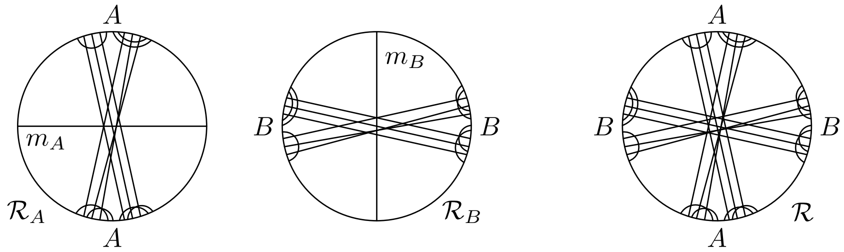


Figure 3.4: On the left, circle representations $\mathcal{R}_A$ and $\mathcal{R}_B$ of graphs X_A and X_B. They are combined into a circle representation $\mathcal{R}$ of X.

The reason is that such neighboring nodes can be joined into a complete node or a star node, respectively. Therefore, the minimal split tree can be constructed from an arbitrary split tree by joining neighboring complete graphs and stars. For instance, the split tree T in Fig. 3.3 is not minimal since the purple and gray stars can be joined, creating the minimal split tree T_X.

Cunnignham's definition of a minimal split decomposition is with respect to inclusion. Since the minimal split decomposition is uniquely determined, it is equivalently the split decomposition terminating on prime and degenerate graphs using the least number of splits.

Computation of minimal split trees. The minimal split tree can be computed in time $\mathcal{O}(n + m)$ using the algorithm of [51]. For the purpose of this chapter, we use the following slower algorithm since it also computes the unique circle representations of encountered prime circle graphs:

Theorem 3.5 (Gioan et al. [77, 76])**.** *The minimal split tree T_X of a circle graph X can be computed in time $\mathcal{O}((n+m)\cdot\alpha(n+m))$ where n is the number of vertices, m is the number of edges, and α is the inverse Ackermann function. Further, the algorithm also computes the unique circle representation of each prime circle node of T_X.*

Graph isomorphism via minimal split decompositions. Let T and T' be two graph-labeled trees. An *isomorphism* $\pi : T \to T'$ is an isomorphism which maps normal edges to normal edges and tree edges to tree edges. Notice that π maps nodes of T to isomorphic nodes of T' while preserving tree edges.

We use minimal split trees to test graph isomorphism of circle graphs:

Lemma 3.6. *Two connected graphs X and Y are isomorphic if and only if the minimal split trees T_X and T_Y are isomorphic.*

Proof. Let T'_X and T'_Y be any split trees of X and Y, respectively, and $\pi : T'_X \to T'_Y$ be an isomorphism. We want to show that $X \cong Y$. Choose an arbitrary tree edge $e = m_A m_B$ in T'_X, we know that $\pi(e) = \pi(m_A)\pi(m_B)$ is a tree edge in T'_Y. We join T'_X over e and T'_Y over $\pi(e)$. We get that the restriction $\pi|_{T'_X \setminus \{m_A, m_B\}}$ is an isomorphism of the constructed graph-labeled trees. By repeating this process, we get single nodes isomorphic graph-labeled trees which are X and Y respectively. So $X \cong Y$.

For the other implication, suppose that $\pi : X \to Y$ is an isomorphism. Let D_X be a minimal split decomposition, constructing the minimal split tree T_X.

We use π to construct a split decomposition D_Y and a split tree T_Y of Y such that $T_X \cong T_Y$. Before any splits, the trees $T_X^0 \cong X$ and $T_Y^0 \cong Y$ are isomorphic. Suppose that $T_X^k \cong T_Y^k$ and D_X then uses a split between A and B in some node N. Then D_Y will use the split between $\pi(A)$ and $\pi(B)$, and since π is an isomorphism, it is a valid split in $\pi(N)$. We construct T_X^{k+1} by splitting N into two nodes N_A and N_B and adding marker vertices m_A and m_B, and similarly for T_Y^{k+1} with marker vertices $m_{\pi(A)}$ and $m_{\pi(B)}$. We extend π to an isomorphism from T_X^{k+1} to T_Y^{k+1} by setting $\pi(m_A) = m_{\pi(A)}$ and $\pi(m_B) = m_{\pi(B)}$. Therefore, the resulting split trees T_X and T_Y are isomorphic. By Theorem 3.4, the minimal split tree of Y is uniquely determined, so it has to be isomorphic to the constructed T_Y. $\square$

3.3 Canonization of graph-labeled trees

In the rest of the chapter, we work with colored graphs and isomorphisms are required to be color-preserving. Colors are represented as non-negative integers.

Definition of canonization. Let X be a colored graph with n vertices with colors in the range $0, \dots, n-1$ and m edges. An *encoding* $\varepsilon(X)$ of X is a sequence of non-negative integers. The encoding $\varepsilon(X)$ is *linear* if it contains at most $\mathcal{O}(n+m)$ integers, each in range $0, \dots, n-1$. We denote the class of all encodings by $\mathcal{E}$. For a class of graphs $\mathcal{C}$, a *linear canonization* is some function $\gamma : \mathcal{C} \to \mathcal{E}$ such that $\gamma(X)$ is a linear encoding of X and for $X, Y \in \mathcal{C}$, we have $X \cong Y$ if and only if $\gamma(X) = \gamma(Y)$.

Fast lexicographic sorting. Since we want to sort these encodings lexicographically, we frequently use the following well-known algorithm:

Lemma 3.7 (Aho, Hopcroft, and Ullman [2], Algorithm 3.2, p. 80)**.** *It is possible to lexicographically sort sequences of numbers $0, \dots, t-1$ of arbitrary lengths in time $\mathcal{O}(\ell + t)$ where ℓ is the total length of these sequences.*

It is easy to modify the above algorithm to get the same running time when all numbers belong to $\{0, 1, 2, \dots, t+1, s, s+1, \dots, s+t-1\}$.

Canonization algorithm. In the rest of this section, we are going to describe the following meta-algorithm.

Lemma 3.8. *Let $\mathcal{C}$ be a class of graphs and $\mathcal{T}$ be a class of graph-labeled trees whose nodes belong to $\mathcal{C}$. Suppose that we can compute a linear-space canonization γ of colored graphs in $\mathcal{C}$ in time $f(n+m)$ where n is the number of vertices, m is the number of edges, and f is convex. Then we can compute a linear-space canonization $\widetilde{\gamma}$ of graph-labeled trees from $\mathcal{T}$ in time $\mathcal{O}(n+m+f(n+m))$.*

Let $T \in \mathcal{T}$ be a graph-labeled tree. Recall that every tree has either a central vertex or a central edge. We may assume that T is rooted at the central node: If a tree edge is central, we insert another node having a single vertex. Also, we orient all tree edges towards the root. For a node N, we denote by $T[N]$ the graph-labeled subtree induced by N and all descendants of N. Initially, we color all marker vertices by the color 0 and all other vertices by the color 1. Throughout the algorithm, only marker vertices change colors.

The k-th *layer* in T is formed by all nodes of the distance k from the root. Notice that every isomorphism from T to T' maps, for every k, the k-th layer of T to the k-th layer of T'. Also, every node N aside the root is incident to exactly one out-going tree edge whose incident marker vertex outside N is called the *parent marker vertex of* N.

The algorithm starts from the bottom layer of T and process the layers towards the root. When a node N is processed, we assign a color c to N. This color c corresponds to a certain linear encoding $\gamma'(N)$ which is created by modifying $\gamma(N)$. Further, we store the mapping ε from colors to these linear encodings, so $\varepsilon(c) = \gamma'(N)$. The assigned colors have the property that two nodes N and N' have the same assigned color if and only if the rooted graph-labeled subtrees induced by N and all the nodes below and by N' and all the nodes below, respectively, are isomorphic. We remove the node N from T and assign its color to the parent marker vertex of N. Also notice that when a non-root node is processed, all marker vertices except for one have colors different from 0, and for the root node, all marker vertices have colors different from 0.

Let $N_1, \dots, N_k$ be the nodes of the currently processed layer such that each vertex in these nodes has one color from $\{0, 1, \dots, t+1, s, s+1, \dots, s+t-1\}$. For each node N_i, we want to use the canonization subroutine to compute the linear encoding $\gamma(N_i)$. But the assumptions require that for ℓ vertices in N_i, all colors are in range $0, \dots, \ell - 1$, but we might have $s \gg \ell$. We can avoid this by renumbering the colors since at most ℓ different colors are used on N_i. Suppose that exactly c_i different colors are used in N_i, and we define the injective mapping $\varphi_{N_i} : \{0, 1, \dots, c_i - 1\} \to \{0, 1, \dots, t+1, s, s+1, \dots, s+t-1\}$ such that the smallest used color is $\varphi_{N_i}(0)$, the second smallest is $\varphi_{N_i}(1)$, and so on till $\varphi_{N_i}(c_i - 1)$. The renumbering of colors on N_i is given by the inverse $\varphi_{N_i}^{-1}$.

After renumbering, the algorithm runs the canonization subroutine to compute the linear encodings $\gamma(N_1), \dots, \gamma(N_k)$. We create the modified linear encoding $\gamma'(N_i)$ by pre-pending $\gamma(N_i)$ with the sequence c_i, $\varphi_{N_i}(0)$, $\varphi_{N_i}(1)$, $\dots$, $\varphi_{N_i}(c_i - 1)$ where c_i is the number of different colors used in N_i. The algorithm lexicographically sorts these modified encodings $\gamma'(N_1), \dots, \gamma'(N_k)$ using Lemma 3.7. Next, we assign the color $s + t$ to the nodes having the smallest encodings, the color $s + t + 1$ to the nodes having the second smallest encodings, and so on. For every node N_i, we remove it and set the color of the parent marker vertex of N_i to the color assigned to N_i.

Suppose that the root node N has the color c assigned, so throughout the algorithm, we have used the colors $0, \dots, c$. The computed linear encoding of T is an encoded concatenation of $\varepsilon(2), \varepsilon(3), \dots, \varepsilon(c)$.

Lemma 3.9. *The described algorithm produces a correct linear canonization $\widetilde{\gamma}$ of graph-labeled trees in $\mathcal{T}$, i.e., for $T, T' \in \mathcal{T}$, we have $T \cong T'$ if and only if $\widetilde{\gamma}(T) = \widetilde{\gamma}(T')$.*

Proof. Let $\pi : T \to T'$ be an isomorphism. We prove by induction from the bottom layer to the root that $\gamma'(N) = \gamma'(\pi(N))$. Suppose that we are processing the ℓ-th layer of T and T' and all previously used colors have the same assigned encodings in T and T'. For every node N of the ℓ-th layer of T, then $\pi(N)$ belongs to the ℓ-th layer of T'. We argue that π also preserves the colors of N. Since π is an isomorphism of graph-labeled trees, it maps marker vertices to marker vertices and non-marker vertices to non-marker vertices. Let $N_1, \dots, N_k$ be the children of N. By the induction hypothesis, we have $\gamma'(N_i) = \gamma'(\pi(N_i))$, so the same colors are assigned to N_i and $\pi(N_i)$. Therefore, π preserves the colors of N, and this property still holds after renumbering the colors by $\varphi_N^{-1} = \varphi_{\pi(N)}^{-1}$. Therefore, we have $\gamma(N) = \gamma(\pi(N))$ and thus $\gamma'(N) = \gamma'(\pi(N))$. Thus, the lexicographic sorting of the nodes in the ℓ-th layer of T and of T' is the same, so N and $\pi(N)$ have the same color assigned. Finally, the encodings $\varepsilon(2), \dots, \varepsilon(c)$ are the same in T and T', so $\widetilde{\gamma}(T) = \widetilde{\gamma}(T')$.

For the other implication, we show that a graph-labeled tree T' isomorphic to T can be reconstructed from $\widetilde{\gamma}(T)$. We construct the root node N from $\varepsilon(c)$. Since γ is a canonization of $\mathcal{C}$, we obtain N by applying γ^{-1}. Next, we invert the recoloring by applying $\varphi(N)$ on the colors of N. Next, we consider each marker vertex in N. If it has some color c_i, we use $\varepsilon(c_i)$ to construct a child node N' of N exactly as before. We proceed in this way till all nodes are expanded and only the colors 0 and 1 remain. It is easy to prove by induction that $T \cong T'$ since each $\varepsilon(c_i)$ uniquely determines the corresponding subtree in T. □

Lemma 3.10. *The described algorithm runs in time $\mathcal{O}(n + m + f(n + m))$.*

Proof. When we run the canonization subroutine on a node having n' vertices and m' edges, it has all colors in range $0, \dots, n' - 1$, so we can compute its linear encoding in time $f(n' + m')$. Since f is convex and the canonization subroutine runs on each node exactly once, the total time spend by this subroutine is bounded by $f(n + m)$.

The total count of used colors is clearly bounded by n and each layer uses different colors except for 0 and 1. Consider a layer with nodes $N_1, \dots, N_k$ having ℓ vertices and ℓ' edges in total. All these nodes use at most ℓ different colors. Therefore, the modified encodings $\gamma'(N_1), \dots, \gamma'(N_k)$ consisting of integers from $\{0, 1, 2, \dots, \ell{+}1, s, s{+}1, \dots, s{+}\ell{-}1\}$, for some value s, and are of the total length $\mathcal{O}(\ell + \ell')$. Therefore, lexicografic sorting of these modified encodings can be done in time $\mathcal{O}(\ell + \ell')$, and this sorting takes total time $\mathcal{O}(n + m)$ for all layers of T. Therefore, the total running time of the algorithm is $\mathcal{O}(n + m + f(n + m))$. □

Proof of Lemma 3.8. It follows by Lemmas 3.9 and 3.10. □

3.4 Canonization of prime and degenerate circle graphs

Let $\mathcal{C}$ be the class of colored prime and degenerate circle graphs. Recall that all nodes of minimal split trees of connected circle graphs belong to $\mathcal{C}$. To apply the meta-algorithm of Lemma 3.8, we need to show that the linear canonization γ of $\mathcal{C}$ can be computed in time $\mathcal{O}(n + m)$ where n is the number of vertices and m is the number of edges.

Linear canonizations of degenerate graphs. For a colored complete graph $X = K_n$, we sort its colors using bucket sort in time $\mathcal{O}(n)$, so the vertices have the colors $c_1 \le c_2 \le \cdots \le c_n$. The computed linear canonization $\gamma(X)$ is $0, c_1, c_2, \ldots, c_n$.

For a star $X = S_n$, we sort the colors of leaves using bucket sort in time $\mathcal{O}(n)$, so they have the colors $c_1 \le c_2 \le \cdots \le c_n$, while the center has the color c_0. The computed linear canonization $\gamma(X)$ is $1, c_0, c_1, c_2, \ldots, c_n$.

Linear encodings of colored cycles. As a subroutine, we need to find a canonical form of a colored cycle. To do this, it suffices to find the the lexicographically minimal rotation of a circular string. This can be done using $\mathcal{O}(n)$ comparisons over some alphabet Σ [21, 145].

Linear canonizations of circle representations. Let X be an arbitrary colored circle graph on n vertices together with an arbitrary circle representation $\mathcal{R}$. The standard way to describe $\mathcal{R}$ is to arbitrarily order the vertices $1, \ldots, n$ and to give a circular word ω consisting of $2n$ integers from $1, \ldots, n$, each appearing exactly twice, in such a way that the occurrences of i and j alternate (i.e., appear as $ijij$) if and only if $ij \in E(X)$. This circular word describes the ordering of the endpoints of the chords in, say, the clockwise direction.

Let X and Y be two colored circle graphs on n vertices labeled $1, \ldots, n$ with circle representations $\mathcal{R}_X$ and $\mathcal{R}_Y$ represented by $\omega_X = \omega_X^1, \ldots, \omega_X^{2n}$ and $\omega_Y = \omega_Y^1, \ldots, \omega_Y^{2n}$. We say that $\mathcal{R}_X \cong \mathcal{R}_Y$ if and only if there exists a bijection $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that

- the vertices i in X and $\pi(i)$ in Y have identical colors, and
- the circular words ω_Y and $\pi(\omega_X^1), \ldots, \pi(\omega_X^{2n})$ are identical.

Notice that when $\mathcal{R}_X \cong \mathcal{R}_Y$, necessarily $X \cong Y$, but in general the converse is not true. We want to construct a linear canonization γ such that $\mathcal{R}_X \cong \mathcal{R}_Y$ if and only if $\gamma(\mathcal{R}_X) = \gamma(\mathcal{R}_Y)$.

To this end, we consider a different encoding of the representation which is invariant on rotation. For each of $2n$ endpoints $e_1, \ldots, e_{2n}$, we store two numbers:

- The color $c_i \in \{0, \ldots, n-1\}$ of the vertex of the chord corresponding to e_i.
- The number of endpoints g_i in the clockwise direction between e_i and the other endpoint corresponding to the same chord. We have $g_i \in \{0, \cdots, 2n-2\}$ and when the e_i and e_j correspond one chord, then $g_i + g_j = 2n - 2$.

To distinguish c_i from g_i, we increase all c_i by $2n-1$, so $c_i \in \{2n-1, \cdots, 3n-2\}$. Then we may consider the circular word $\lambda_X = g_1, c_1, g_2, c_2, \ldots, g_{2n}, c_{2n}$ of length $4n$.

Lemma 3.11. *Let X and Y be two colored circle graphs with representations $\mathcal{R}_X$ and $\mathcal{R}_Y$. We have $\mathcal{R}_X \cong \mathcal{R}_Y$ if and only if the circular words λ_X and λ_Y are identical.*

Proof. If $\mathcal{R}_X \cong \mathcal{R}_Y$, there exists an index $k \in \{0, \ldots, 2n-1\}$ such that rotating the representation $\mathcal{R}_X$ by k endpoints produces $\mathcal{R}_Y$. When

$$\lambda_X = g_1, c_1, \ldots, g_{2n}, c_{2n} \quad \text{and} \quad \lambda_Y = g'_1, c'_1, \ldots, g'_{2n}, c'_{2n},$$

it cyclically holds that $g_i = g'_{i+k}$ and $c_i = c'_{i+k}$. So the circular words λ_X and λ_Y are identical.

For the other implication, observe that the circle representation $\mathcal{R}_X$ and the circle graph X can be reconstructed from λ_X. If λ_X and λ_Y are identical, we reconstruct isomorphic representations $\mathcal{R}_X$ and $\mathcal{R}_Y$. $\square$

Lemma 3.12. *We can compute the linear encoding γ of colored circle representations in time $\mathcal{O}(n)$.*

Proof. For a representation $\mathcal{R}_X$, we can clearly compute λ_X in time $\mathcal{O}(n)$. Next, we apply to λ_X a cycle canonization algorithm [21, 145] which computes $\gamma(\mathcal{R}_X)$ in time $\mathcal{O}(n)$. $\square$

Linear canonization of prime circle graphs. Let X be a prime circle graph. It has at most two different representations $\mathcal{R}_X$ and $\mathcal{R}'_X$ where one is the reversal of the other. Using Lemma 3.12, we compute their linear encodings λ_X and λ'_X. As the linear encoding $\gamma(X)$, we chose the lexicographically smallest of λ_X and λ'_X, prepended with the value 2. Clearly, colored prime circle graphs X and Y are isomorphic if and only if $\gamma(X) = \gamma(Y)$.

By putting the results of this section together, we get the following:

Lemma 3.13. *We can compute linear canonization of colored prime circle graphs and degenerate graphs in time $\mathcal{O}(n)$.*

3.5 Canonization and graph isomorphism of circle graphs

In this section, we combine the presented results to show that a linear canonization γ of circle graphs can be computed in time $\mathcal{O}((n+m)\cdot\alpha(n+m))$. This algorithm clearly implies Theorem 3.1 since circle graphs X and Y are isomorphic if and only if $\gamma(X) = \gamma(Y)$.

Suppose that X is a connected circle graph. We apply the algorithm of Theorem 3.5 to compute the minimal split decomposition T_X of X and the unique circle representation for each prime circle graph (up to reversal). We halt if some circle representations does not exist since X is not a circle graph. The total running time of preprocessing is $\mathcal{O}((n+m)\cdot\alpha(n+m))$, and the remainder of the algorithm runs in time $\mathcal{O}(n+m)$, so this step is the bottleneck. Next, we use Lemmas 3.13 and 3.8 to compute a linear canonization $\gamma(T_X)$ and we put $\gamma(X) = \gamma(T_X)$.

Suppose that the circle graph X is disconnected, and let $X_1, \dots, X_k$ be its connected components. We compute their linear encodings $\gamma(X_1), \dots, \gamma(X_k)$, lexicografically sort them in time $\mathcal{O}(n+m)$ using Lemma 3.7, and output them in $\gamma(X)$ sorted as a sequence. The total running time is $\mathcal{O}((n+m)\cdot\alpha(n+m))$.

When the input also gives a circle representation $\mathcal{R}$, we can avoid using Theorem 3.5. Instead, we compute a split decomposition and the corresponding split tree in time $\mathcal{O}(n+m)$ using [51]. We can easily modify this split tree into the minimal split tree by joining neighboring complete vertices and stars as discussed in Section 3.2. For each prime node N, we obtain its unique circle representation by restricting $\mathcal{R}$ to the vertices of N. Since the avoided algorithm of Theorem 3.5 was the bottleneck, we get the total running time $\mathcal{O}(n+m)$.

3.6 Open problems

We conclude this chapter by discussing several possible research directions and open problems. The main used tool is minimal split decomposition. The algorithm finds the cannonical form of the unique minimal split tree T, using the canonical forms of every prime and degenerate circle graph appearing as a node of T. We obtain an algorithm computing the cannonical form for every circle graph.

Problem 2. *Does the minimal split tree capture all possible representations of circle graph?*

The *k-dimensional Weisfeiler-Leman* [162] algorithm (k-dim WL) is a fundamental algorithm used as a subroutine in graph isomorphism testing. The algorithm colors k-tuples of vertices of two input graphs and iteratively refines the color classes until the coloring becomes stable. We say that k-dim WL *distinguishes* two graph X and Y if and only if its application to each of them gives different colorings. Two distinguished graphs are clearly non-isomorphic, however, for every k, there exist non-isomorphic graphs not distinguished by k-dim WL. For a class of graphs $\mathcal{C}$, the *Weisfeiler-Leman dimension* of $\mathcal{C}$ is the minimum integer k such that k-dim WL distinguishes every $X, Y \in \mathcal{C}$ such that $X \ncong Y$.

Problem 3. *What is the Weisfeiler-Leman dimension of circle graphs?*

For many classes of graphs such as interval graphs [59], planar graphs [102], or more generally graphs with excluded minors [81], and recently also circular-arc graphs [132], the Weisfeiler-Leman dimension is finite. A major open problem is for which classes of graphs can isomorphism be tested in polynomial time without using group theory, i.e., by a combinatorial algorithm, often meant some k-dim WL.

Further, a natural question to ask, especially for problems solvable in linear time, is whether they can be solved using logarithmic space. This is, for example, known for interval graphs [110] and Helly circular-arc graphs [111]. Recently, a parametrized logspace algorithm was given for testing isomorphism of circular-arc graphs in [31].

Problem 4. *Can isomorphism of circle graphs be tested using logarithmic space?*

Finally, we mention the *partial representation extension* problem which is a generalization of the recognition problem. The input consists of, in our case, a circle graph and a circle representation of its induced subgraph and the task is to complete the representation or output that it is not possible. Obviously, this problem can be asked for various classes of graphs and it was extensively studied [106, 104, 105, 33, 15, 113].

In [33], the authors give an $\mathcal{O}(n^3)$ algorithm to solve this problem. They give an elementary recursive description of the structure of all representations and try to match the partial representation on it. It is a natural question whether the minimal split trees can be used to solve the partial representation extension problem faster.

Problem 5. *Can the partial representation extension problem for circle graphs be solved using minimal split decomposition faster that $\mathcal{O}(n^3)$?*

4. Isomorphism problem for chordal graphs

4.1 Introduction

In this chapter, we deal with parameterized complexity of the graph isomorphism problem for the class of chordal graphs. An undirected graph is said to be *chordal* if it has no chordless cycle of length at least four. Every chordal graph admits a representation as the intersection graph of subtrees of some tree [74]. The *leafage* $\ell(X)$ of a chordal graph X is the minimum integer $\ell \geq 1$ such that X has a representation on a tree with ℓ leaves. The leafage was introduced in [114] and provides a natural parametrization of the class of all chordal graphs.

One can see that, $\ell(X) = 1$ if and only if X is complete, and $\ell(X) \leq 2$ if and only if X is an interval graph, i.e., the intersection graph of finitely many intervals on the real line. Thus the leafage measures how close is a chordal graph to interval graph, which has some important algorithmic consequences. For instance, efficient solutions to some NP-hard problems on interval graphs naturally extend to chordal graphs of bounded leafage; e.g, [149].

The graph isomorphism problem restricted to the class of chordal graphs remains as hard as the general graph isomorphism problem [118, Theorem 5]. On the other hand, the problem can be solved in linear time for interval graphs [118]. The main result of this chapter can be considered as a substantial generalization of the latter.

Theorem 4.1. *Testing isomorphism of chordal graphs of bounded leafage is fixed-parameter tractable.*

Denote by $\mathfrak{K}_\ell$ the class of graphs of all chordal graphs of leafage at most ℓ. To test if two connected graphs $X, Y \in \mathfrak{K}_\ell$ are isomorphic, it suffices to find a generator of the automorphism group of the disjoint union $X \cup Y$, which swaps X and Y. Since the graph $X \cup Y$ belongs to the class $\mathfrak{K}_{2\ell}$, the graph isomorphism problem for the graphs in $\mathfrak{K}_\ell$ is reduced to the problem of finding the automorphism group of a given graph in $\mathfrak{K}_{2\ell}$. Thus Theorem 4.1 is an immediate consequence of the following theorem which is proved in this chapter.

Theorem 4.2. *Given an n-vertex graph $X \in \mathfrak{K}_\ell$, a generating set of the group* $\mathrm{Aut}(X)$ *can be found in time* $t(\ell)\,\mathrm{poly}(n)$*, where $t(\cdot)$ is a function independent of n.*

The function t from Theorem 4.2 is bounded from above by a polynomial in $(\ell 2^\ell)!$. This estimate does not appear to be final and, most likely, can be significantly improved.

The proof of Theorem 4.2 is given in Section 4.7. The first main idea is to represent the group $\mathrm{Aut}(X)$ as the automorphism group $\mathrm{Aut}(H)$ of an order-3 hypergraph $H = H(X)$. In contrast to ordinary hypergraph with vertex set V and hyperedge set contained in the power set $\mathcal{E}_1 = 2^V$, the structure of H includes order-2 and order-3 hyperedges which are elements of $\mathcal{E}_2 = 2^{\mathcal{E}_1}$ and $\mathcal{E}_3 = 2^{\mathcal{E}_2}$, respectively. Of course, the size of H is still bounded by a polynomial in n. The reduction of finding $\mathrm{Aut}(X)$ to finding $\mathrm{Aut}(H)$ is presented in Sections 4.4 and 4.5. The key point of the reduction is a graph-theoretical analysis of the vertex coloring of the graph X, obtained by the Weisfeiler-Leman algorithm [162]. At the end of the reduction, we arrive to the colored order-3 hypergraph H such that the size of each color class of vertices contains at most $b = \ell 2^\ell$ vertices, where $\ell = \ell(X)$.

At this point, we come to a general problem of finding the automorphism group of the order-k hypergraph H ($k \geq 1$) by an FPT algorithm with respect to the parameter b. This problem seems to be important in itself. For ordinary hypergraphs, it was solved in [4]. A relevant generalization of the algorithm proposed there is given in Section 4.6.

Further motivation for our work stems from the concept of H-graphs, originally introduced in [19]. A graph is an H-graph if it is an intersection graphs of connected subgraphs of a subdivision of H. This generalizes many important classes: interval graphs are K_2-graphs, circular-arc graphs are K_3-graphs, and chordal graphs are the union of all T-graphs, where T is a tree. Every graph is an H-graph for a suitable H, which gives a parametrization of all graphs.

Surprisingly, some basic algorithmic results, including the recognition and isomorphism problem, were obtained for H-graphs only recently; see e.g., [34, 35, 67]. It was proved in [1] that isomorphism of S_d-graphs, where S_d is a star of degree d, is in FPT. Since every S_d-graph is a chordal graph of leafage bounded by d, our result improves that in [1]. On the other hand, the isomorphism problem for H-graphs is as hard as the general graph isomorphism problem if H is not unicyclic [36]. It remains open whether isomorphism can be solved in polynomial time for the unicyclic case.

4.2 Notation

In this chapter, we use concepts from permutation group theory, graph theory, and the theory of coherent configurations. Thus, we introduce some notation specific for this chapter.

General notation. Throughout the whole chapter, Ω is a finite set. Given a bijection f from Ω to another set and a subset $\Delta \subseteq \Omega$, we denote by f^Δ the bijection from Δ to $\Delta^f = \{\delta^f : \delta \in \Delta\}$. For a set S of bijections from Δ to another set, we put $S^\Delta = \{f^\Delta : f \in S\}$.

The group of all permutations of a set Ω is denoted by $\mathrm{Sym}(\Omega)$. When a group G acts on Ω, we set $G^\Omega = \{g^\Omega : g \in G\}$ to be the permutation group induced by this action. Concerning standard permutation group algorithms we refer the reader to [144].

Let π be a partition of Ω. The set of all unions of the classes of π is denoted by $\pi^\cup$. The partition π is a *refinement* of a partition π' of Ω if each class of π' belongs to $\pi^\cup$; in this case, we write $\pi \geq \pi'$, and $\pi > \pi'$ if $\pi \geq \pi'$ and $\pi \neq \pi'$. The partition of $\Delta \subseteq \Omega$ induced by π is denoted by π_Δ.

Graphs. Let X be an undirected graph. The vertex and edge sets of X are denoted by $\Omega(X)$ and $E(X)$, respectively. The automorphism group of X is denoted by $\mathrm{Aut}(X)$. The set of all isomorphisms from X to a graph X' is denoted by $\mathrm{Iso}(X, X')$.

The set of all leaves and of all connected components of X are denoted by $L(X)$ and $\mathrm{Conn}(X)$, respectively. For a vertex α, we denote by αX the set of neighbors of α in X. The vertices α and β are called *twins* in X if every vertex other than α and β is adjacent either to both α and β or neither of them. The graph X is said to be *twinless* if no two distinct vertices of X are twins.

Let $\Delta, \Gamma \subseteq \Omega(X)$. We denote by $X_{\Delta,\Gamma}$ the graph with vertex set $\Delta \cup \Gamma$ in which two vertices are adjacent if and only if one of them is in Δ, the other one is in Γ, and they are adjacent in X. Thus, $X_\Delta = X_{\Delta,\Delta}$ is the subgraph of X induced by Δ, and $X_{\Delta,\Gamma}$ is bipartite if $\Delta \cap \Gamma = \varnothing$.

Let $\Delta \subseteq \Omega(X)$ and $Y = X_\Delta$. The set of all vertices adjacent to at least one vertex of Δ and not belonging to Δ is denoted by ∂Y. The subgraph of X, induced by $\Delta \cup \partial Y$ is denoted by $\overline{Y}$.

For a tree T, let $S(T) = \{\Omega(T') : T' \text{ is a subtree of } T\}$ be the set of all vertex sets of the subtrees of T. A *tree-representation* of a graph $X = (\Omega, E)$ on T is a function $R\colon \Omega \to S(T)$ such that for all $u, v \in \Omega$,

$$R(u) \cap R(v) \neq \varnothing \Leftrightarrow \{u, v\} \in E.$$

It is known that a graph X is chordal if and only if X has a such a representation [74]. The leafage $\ell(X)$ of X is defined to be the minimum of $|L(T)|$ taken over all tree-representations of X on T and all trees T for which such a representation exists. In particular, the graph X is interval if and only if $\ell(X) \leq 2$.

Colorings. A partition π of Ω is said to be a *coloring* (of Ω) if the classes of π are indexed by elements of some set, called the *colors*. In this case, the classes of π are called *color classes* and the color class containing $\alpha \in \Omega$ is denoted by $\pi(\alpha)$. Usually the colors are assumed to be linearly ordered. A bijection f from Ω to another set equipped with coloring π', is said to be *color preserving* if $\pi(\alpha) = \pi'(f(\alpha))$.

A graph equipped with a coloring of the vertex set (respectively, edge set) is said to be *vertex colored* (respectively, *edge colored*); a graph which is both vertex and edge colored is said to be *colored*. The isomorphisms of vertex/edge colored graphs are ordinary isomorphisms which are color preserving. To emphasize this fact, we sometimes use notation $\mathrm{Aut}(X, \pi)$ for the automorphism group of a graph X with coloring π.

Let π be a coloring of a graph X. Consider the application of the Weisfeiler-Leman algorithm (2-dim WL) to the vertex-colored graph X [162]. The output of this algorithm defines a new coloring $\mathrm{WL}(X, \pi) \geq \pi$ of X. We say that π is *stable* if $\mathrm{WL}(X, \pi) = \pi$. Using the language of coherent configurations, π is stable if and only if the classes of π are the fibers of a coherent configuration (details can be found in the monograph [37]). In the sequel, we will use some elementary facts from theory of coherent configurations. The following statement summarizes relevant properties of stable colorings.

Lemma 4.3. *Let X be a graph and π be a stable coloring of X. Then*

1. *given $\Delta, \Gamma \in \pi$, the number $|\delta X \cap \Gamma|$ does not depend on $\delta \in \Delta$,*

2. *if $\Delta \in \pi^{\cup}$ or $X_\Delta \in \mathrm{Conn}(X)$, then the coloring π_Δ is stable.*

A coloring π of the vertices of a graph X is said to be *invariant* if every class of π is $\mathrm{Aut}(X)$-invariant. In this case, the coloring $\mathrm{WL}(X, \pi)$ is also invariant and stable. Since the coloring of the vertices in one color is invariant and the Weisfeiler-Leman algorithm is polynomial-time, we may always deal with invariant stable colorings.

Hypergraphs. Let V be a finite set. The set $\mathcal{E}_k = \mathcal{E}_k(V)$ of the *order-k hyperedges* on V is defined recursively as follows:

$$\mathcal{E}_0 = V, \qquad \mathcal{E}_k = \mathcal{E}_{k-1} \cup 2^{\mathcal{E}_{k-1}} \text{ for } k > 1.$$

Thus we consider elements of V as order-0 hyperedges, and order-k hyperedges include all order-$(k-1)$ hyperedges and their subsets.

Let $U \subseteq V$ and $e \in \mathcal{E}_k$ ($k \geq 1$). We recursively define the *projection* of e on U as the multiset

$$e^U = \begin{cases} e \cap U & \text{if } k = 1, \\ \{\{\widetilde{e}^{\,U} : \ \widetilde{e} \in e\}\} & \text{if } k > 1. \end{cases}$$

We extend this definition to all sets $E \in \mathcal{E}_k$ by putting $E^U = \{e^U : \ e \in E\}$.

Definition 4.4 (order-k hypergraph)**.** *An order-k hypergraph ($k \geq 1$) on V is a pair $H = (V, E)$, where $E \subseteq 2^{\mathsf{E}_k}$; the elements of V and E are called vertices and hyperedges of H, respectively.*

It is easily seen that order-1 hypergraphs are usual hypergraphs and the higher-order hypergraphs (i.e., order-k hypergraph for some k) are combinatorial objects in the sense of [25]. The concepts of isomorphism and coloring extend to higher-order hypergraphs in a natural way.

Let $k \geq 2$. The $(k-1)$-*skeleton* of an order-k hypergraph $H = (V, E)$ is an order-$(k-1)$ hypergraph $H^{(k-1)}$ on V with the hyperedge set

$$E^{(k-1)} = \{\widetilde{e} \in \mathcal{E}_{k-1} : \ \widetilde{e} \text{ belongs to some } e \in \mathcal{E}_k\}.$$

It is easily seen that for every order-k hypergraph $H' = (V', E')$

$$\mathrm{Iso}(H, H') = \{f \in \mathrm{Iso}(H^{(k-1)}, H'^{(k-1)}) : \ e \in E \ \Leftrightarrow \ e^f \in E'\}. \tag{4.1}$$

where for each order-k hyperedge $e = \{e_1, \ldots, e_a\}$ we set $e^f = \{e_1^f, \ldots, e_a^f\}$.

Let $H_1 = (V_1, E_1)$ be an order-k hypergraph for some k and $H_2 = (V_2, E_2)$ be a usual hypergraph such that $V_2 = E_1$. Then each hyperedge $e \in E_2$ is a subset of hyperedges of E_1. We define the *hypergraph composition* of H_1 and H_2 to be the order-$(k+1)$ hypergraph

$$H := H_1 \uparrow H_2 = (V, E_1 \cup E_2).$$

Note that if H_1 is a k_1-order hypergraph and H_2 is a k_2-order hypergraph, then $H := H_1 \uparrow H_2$ is a $(k_1 + k_2)$-order hypergraph.

When the hypergraphs H_1 and H_2 are colored, the coloring of every hyperedge set $\mathcal{E}_i \cap E(H)$ is inherited from the coloring of E_1 for $i \le k$; the color of every hyperedge $e \in \mathcal{E}_{k+1} \cap E(H)$ is defined to be the pair consisting of the color of $e \in E_2$ and the multiset of the vertex colors of the elements of e (which are the elements of V_1).

4.3 Stable colorings in chordal graphs

We prove several auxiliary statements about the structure of subgraphs of a chordal graph, induced by one or two color classes of a stable coloring.

Lemma 4.5. *Let X be a chordal graph and π a stable coloring of X. Then for every $\Delta, \Gamma \in \pi$, the following statements hold:*

1. $\mathrm{Conn}(X_\Delta)$ *consists of cliques of the same size,*
2. *if* $|\mathrm{Conn}(X_\Delta)| \le |\mathrm{Conn}(X_\Gamma)|$, *then* $\mathrm{Conn}(X_\Delta) = \{Y_\Delta : Y \in \mathrm{Conn}(X_{\Delta\cup\Gamma})\}$,
3. *if the graphs X_Δ and X_Γ are complete, then $X_{\Delta,\Gamma}$ is either complete bipartite or empty.*

Proof. (1) By Lemma 4.3(1) for $\Delta = \Gamma$, the graph X_Δ is regular. It suffices to verify that every graph $Y \in \mathrm{Conn}(X_\Delta)$ is complete. Since Y is chordal it contains a simplicial vertex, i.e., a vertex whose neighborhood induces a complete graph. As Y is regular, all its vertices are simplicial. Thus, Y is complete.

(2) Let $\overline{X}$ be a bipartite graph with parts $\overline{\Delta} = \mathrm{Conn}(X_\Delta)$ and $\overline{\Gamma} = \mathrm{Conn}(X_\Gamma)$ in which two vertices $\overline{\alpha} \in \overline{\Delta}$ and $\overline{\beta} \in \overline{\Gamma}$ are adjacent if and only if there are vertices $\alpha \in \overline{\alpha}$ and $\beta \in \overline{\beta}$ adjacent in X. By statement (1), the components of $X_{\Delta\cup\Gamma}$ are in one-to-one correspondence with the components of $\overline{X}$. Denote by $\overline{Y}$ the component of $\overline{X}$, corresponding to the component $Y \in \mathrm{Conn}(X_{\Delta\cup\Gamma})$.

The partition $\overline{\pi} = \{\overline{\Delta}, \overline{\Gamma}\}$ is a stable coloring of $\overline{X}$. Indeed, it suffices to find a coherent configuration on $\Omega(\overline{X})$, for which $\overline{\Delta}$ and $\overline{\Gamma}$ are fibers. As this configuration, one can take the quotient of the coherent configuration corresponding to π, modulo the equivalence relation on $\Delta \cup \Gamma$ the classes of which are vertex sets of the graphs belonging to $\overline{\Delta}$ and $\overline{\Gamma}$, see [37, Section 3.1.2].

By Lemma 4.3(1), any two vertices of $\overline{X}$, belonging to the same part, have the same degree. Moreover, the graph $\overline{X}$ is obviously chordal. Consequently, it is acyclic: otherwise, $\overline{X}$ being bipartite contains an induced cycle of length at least 4, which is impossible for a chordal graph. It follows that $\overline{X}$ contains a vertex $\overline{\alpha}$ of degree 1. Since all vertices of the part containing $\overline{\alpha}$ have the same degree, each $\overline{Y} \in \mathrm{Conn}(\overline{X})$ is a star. The center of this star lies in $\overline{\Delta}$, because $|\overline{\Delta}| \le |\overline{\Gamma}|$. Thus, $Y_\Delta \in \mathrm{Conn}(X_\Delta)$, which implies the required statement.

(3) Without loss of generality we may assume that $\Delta \ne \Gamma$. Suppose to the contrary that there are $\delta_1, \delta_2 \in \Delta$ such that $\delta_1 X \cap \Gamma \ne \delta_2 X \cap \Gamma$. By Lemma 4.3(1), we have $|\delta_1 X \cap \Gamma| = |\delta_2 X \cap \Gamma|$. Thus there exist $\gamma_1 \in \delta_1 X \cap \Gamma$ and $\gamma_2 \in \delta_2 X \cap \Gamma$ such that

$$\gamma_1 \notin \delta_2 X \quad \text{and} \quad \gamma_2 \notin \delta_1 X.$$

By the assumption, the graphs X_Δ and X_Γ are complete. It follows that δ_1 and δ_2 are adjacent, and that γ_1 and γ_2 are adjacent. Therefore, the vertices $\delta_1, \gamma_1, \gamma_2, \delta_2$ form an induced 4-cycle of X, a contradiction. $\square$

Remark 4.6. Recall that stable coloring was defined via 2-dimensional Weisfeiler-Leman algorithm. While the 1-dimensional Weisfeiler-Leman algorithm suffices for the first part of Lemma 4.5, it is worth noting that the second part requires the 2-dimensional algorithm.

Lemma 4.7. *Let X be a connected chordal graph and let π be a stable partition of Ω. There exists $\Delta \in \pi$ such that the graph X_Δ is complete.*

Proof. The statement immediately follows from Lemma 4.5(1) if $|\pi| = 1$. Assume that $|\pi| > 1$. Since π is stable, the classes of π are the fibers of some coherent configuration on Ω, see Section 4.2.

Suppose to the contrary that the graph X_Δ is not complete for any $\Delta \in \pi$. Let Δ be a class of π, containing a simplicial vertex of X. Then all vertices in Δ are simplicial, see [61, Lemma 8.1]. It follows that the graph $X' := X_{\Omega\setminus\Delta}$ is connected and chordal. Moreover, the partition $\pi' := \pi_{\Omega\setminus\Delta}$ is stable by Lemma 4.3(2). Since $|\pi'| < |\pi|$, we conclude by induction that there is $\Delta' \in \pi'$ such that $X'_{\Delta'}$ is complete, which is not possible, because $\Delta' \in \pi$. $\square$

Estimates depending on the leafage. The following two lemmas are crucial for estimating the complexity of the main algorithm.

Lemma 4.8. *Let X be a chordal graph, Δ a subset of its vertices, and*

$$S = S(X, \Delta) = \{Y \in \mathrm{Conn}(X - \Delta) : \ \overline{Y} \text{ is not interval}\}, \tag{4.2}$$

Then $|S| \le \ell(X) - 2$.

Proof. Let R be a tree-representation of X on a tree T such that $|L(T)| = \ell(X)$, and let n_3 be the number of all vertices of T of degree at least 3. It is not hard to verify that

$$n_3 \le \ell - 2, \tag{4.3}$$

where $\ell = \ell(X)$.

Let $Y \in S(X, \Delta)$, and let $R(Y)$ be the union of all subtrees $R(\alpha)$, $\alpha \in \Omega(Y)$. Then $R(Y)$ is a subtree of T. We claim that $R(Y)$ contains a vertex t_Y of degree at least 3. Indeed, otherwise, $R(Y)$ is a path in T. Moreover, if $\alpha \in \partial Y$, then either $R(\alpha)$ is a subpath of P, or $R(\alpha)$ contains at least one end of P. This implies that the restriction of R to the set $\Omega(Y) \cup \partial Y$ is a tree-representation of $\overline{Y}$ on P. But then $\overline{Y}$ is interval, a contradiction.

To complete the proof, we note that the sets $R(Y)$, $Y \in S$, are pairwise disjoint. Therefore the vertices t_Y are pairwise distinct. By inequality (4.3), this yields

$$|S| = |\{t_Y : \ Y \in S\}| \le n_3 \le \ell - 2,$$

as required. $\square$

Let π be a vertex coloring of X. Given a pair $(\Delta, \Gamma) \in \pi \times \pi$, we define an equivalence relation $e_{\Delta,\Gamma}$ on Δ by setting

$$(\delta, \delta') \in e_{\Delta,\Gamma} \;\Leftrightarrow\; \delta \text{ and } \delta' \text{ are twins in } X_{\Delta,\Gamma}. \tag{4.4}$$

Note that the equivalence relation $e_{\Gamma,\Delta}$ is defined on Γ, and coincides with $e_{\Delta,\Gamma}$ only if $\Gamma = \Delta$. The sets of classes of $e_{\Delta,\Gamma}$ and $e_{\Gamma,\Delta}$ are denoted by $\Delta/e_{\Delta,\Gamma}$ and $\Gamma/e_{\Gamma,\Delta}$, respectively.

Lemma 4.9. *Let X be a chordal graph, π a stable coloring, and $\Delta, \Gamma \in \pi$. Assume that the graph X_Δ is complete. Then*

$$|\Delta/e_{\Delta,\Gamma}| \le 2^\ell \quad \textit{and} \quad |\Gamma/e_{\Gamma,\Delta}| \le \ell, \tag{4.5}$$

where $\ell = \ell(X)$.

Proof. Without loss of generality we may assume that $X = X_{\Delta\cup\Gamma}$ (because $\ell(X_{\Delta\cup\Gamma}) \le \ell(X)$), and the graph $X_{\Delta,\Gamma}$ is neither complete bipartite, nor empty (otherwise, $|\Delta/e_{\Delta,\Gamma}| = 1$ and $|\Gamma/e_{\Gamma,\Delta}| = 1$, and both statements are trivial). Thus, X_Γ is not complete by Lemma 4.5(3) and Δ is a maximal clique of X_Δ; in particular, $\Delta \ne \Gamma$.

Let $R : \Omega \to S(T)$ be a tree-representation of the graph X on a tree T with ℓ leaves. Without loss of generality, we may assume that the set $\Omega(T)$ is the minimum possible. Since Δ is a clique of X, the intersection of the subtrees $R(\delta)$, $\delta \in \Delta$, contains at least one point t.

Let $\gamma \in \Gamma$. Then $t \notin R(\gamma)$ by the maximality of Δ. Denote by t_γ the point of $R(\gamma)$, lying at the minimum distance from t in T. Let P_γ be the path connecting t and t_γ; note that P_γ has at least two vertices, because $t \ne t_\gamma$.

Let us define a partial order on $\mathcal{T} = \{t_\gamma : \gamma \in \Gamma\}$ by setting $t_\gamma \preceq t_{\gamma'}$ if and only if t_γ lies in $P_{\gamma'}$ (in particular, t_γ is closer to t than $t_{\gamma'}$), or equivalently, $P_\gamma \subseteq P_{\gamma'}$.

Claim. If $t_\gamma \preceq t_{\gamma'}$, then $(\gamma, \gamma') \in e_{\Gamma,\Delta}$.

Proof. Let $\delta \in \gamma' X \cap \Delta$. Then the intersection $R(\delta) \cap R(\gamma')$ is not empty. Moreover, it contains $t_{\gamma'}$: for otherwise, because $t \in R(\delta)$, the set $R(\gamma')$ contains a vertex which is closer to t than $t_{\gamma'}$. Consequently, $P_{\gamma'} \subseteq R(\delta)$. Since $P_\gamma \subseteq P_{\gamma'}$, this implies that

$$t_\gamma \in P_\gamma \subseteq R(\gamma') \subseteq R(\delta),$$

i.e., the intersection $R(\delta) \cap R(\gamma) \ni t_\gamma$ is not empty; in particular, $\delta \in \gamma X \cap \Delta$. It follows that $\gamma' X \cap \Delta \subseteq \gamma X \cap \Delta$. Since also $|\gamma' X \cap \Delta| = |\gamma X \cap \Delta|$ by Lemma 4.3(1), we are done. $\square$

Let $\mathcal{T}_{\min} \subseteq \mathcal{T}$ be the set of all minimal points with respect to the partial order on $\mathcal{T}$. By the claim, for every $\gamma' \in \mathcal{T} \setminus \mathcal{T}_{\min}$ there is $\gamma \in \mathcal{T}_{\min}$ such that $(\gamma, \gamma') \in e_{\Gamma,\Delta}$. Thus,

$$|\Gamma/e_{\Gamma,\Delta}| \le |\mathcal{T}_{\min}|.$$

On the other hand, by the minimality of T, every leaf of T belongs to $R(\gamma)$ for some $\gamma \in \Gamma$. Consequently, the path from any leaf of T to t contains at most one point of $\mathcal{T}_{\min}$. Thus, $|\mathcal{T}_{\min}| \le \ell$ and so

$$|\Gamma/e_{\Gamma,\Delta}| \le |\mathcal{T}_{\min}| \le \ell,$$

which proves the second inequality in (4.5).

To complete the proof, we observe that if $\delta \in \Delta$, then the set $\delta X \cap \Gamma$ is a union of some classes of $\Gamma/e_{\Gamma,\Delta}$. Denote this union by Γ_δ. Note that if $\delta, \delta' \in \Delta$, then $\Gamma_\delta = \Gamma_{\delta'}$ if and only if $(\delta, \delta') \in e_{\Delta,\Gamma}$. Therefore, the number $|\Delta/e_{\Delta,\Gamma}|$ is at most

$$|\Delta/e_{\Delta,\Gamma}| \le |2^{\Gamma/e_{\Gamma,\Delta}}| = 2^{|\Gamma/e_{\Gamma,\Delta}|} \le 2^\ell,$$

which proves the first inequality in (4.5). □

4.4 Critical set of a chordal graph

Let X be a chordal graph and π a stable coloring. Denote by $\Omega^* = \Omega^*(X,\pi)$ the union of all $\Delta \in \pi$ such that

$$|\operatorname{Conn}(X_\Delta)| \le \ell(X). \tag{4.6}$$

By Lemma 4.5(1), the graph X_Δ is a disjoint union of cliques; thus the above condition means that the number of them is at most $\ell(X)$. By Lemma 4.7, the set Ω^* is not empty if the graph X is connected.

Theorem 4.10. *Let X be a chordal graph and $\Omega^* = \Omega^*(X,\pi)$. Then one of the following statements holds:*

i for every $Y \in \operatorname{Conn}(X - \Omega^*)$, the graph $\overline{Y}$ is interval,

ii there is a invariant stable coloring $\pi' > \pi$.

Moreover, in case (ii), the coloring π' can be found in polynomial time in $|\Omega|$.

Proof. Assume that (i) does not hold. Then the set the set $S = S(X, \Delta^*)$ defined by formula (4.2) for $\Delta = \Omega^*$ is not empty. By Lemma 4.8, we have

$$|S| \le \ell - 2, \tag{4.7}$$

where $\ell = \ell(X)$. Take an arbitrary $Y \in S$. By Lemma 4.3(2), the coloring $\pi_Y := \pi_{\Omega(Y)}$ is stable. By Lemma 4.7, there is $\Gamma' \in \pi_Y$ such that the graph $Y_{\Gamma'}$ is complete. Let Γ be the class of π, containing Γ'. Then

$$\Gamma \cap \Omega^* = \varnothing, \tag{4.8}$$

because Γ intersects $\Omega \setminus \Omega^* \in \pi^\cup$. Moreover, every automorphism of X preserves the sets S and Γ and hence preserves the set

$$S' = \{Z \in S :\ Z_{\Gamma\cap\Omega(Z)} \text{ is complete}\}.$$

Thus the union Γ_0 of all sets $\Gamma \cap \Omega(Z)$, $Z \in S'$, is a nonempty $\operatorname{Aut}(X)$-invariant set contained in Γ. Now if $\Gamma_0 \ne \Gamma$, then we come to case (ii) with

$$\pi' = (\pi \setminus \{\Gamma\}) \cup \{\Gamma \setminus \Gamma_0, \Gamma \cap \Gamma_0\}.$$

To complete the proof, assume that $\Gamma_0 = \Gamma$. Then by inequality (4.7), the graph X_Γ is the union of at most $|S'| \le |S| \le \ell - 2$ cliques. By the definition of Ω^*, this yields $\Gamma \subseteq \Omega^*$, which contradicts relation (4.8). □

We say that Ω^* is a *critical set* of X (with respect to π) if statement (i) of Theorem 4.10 holds. In the rest of the section we define a hypergraph $\mathcal{H}^*$ associated with the critical set Ω^* and show that the groups $\mathrm{Aut}(\mathcal{H}^*)^{\Omega^*}$ and $\mathrm{Aut}(X)^{\Omega^*}$ are closely related.

The vertices of $\mathcal{H}^*$ are set to be the elements of the disjoint union

$$V = \bigcup_{\Delta\in\pi_{\Omega^*}} \bigcup_{\Gamma\in\pi} \Delta/e_{\Delta,\Gamma},$$

where $e_{\Delta,\Gamma}$ is the equivalence relation on Δ, defined by formula (4.4). Thus any vertex of $\mathcal{H}^*$ is a class of some $e_{\Delta,\Gamma}$. Taking the disjoint union means, in particular, that if Λ is a class of $e_{\Delta,\Gamma}$ and $e_{\Delta,\Gamma'}$, then V contains two vertices corresponding to Λ. The partition

$$\overline{\pi} = \{\Delta/e_{\Delta,\Gamma}:\ \Delta\in\pi_{\Omega^*},\ \Gamma\in\pi\}$$

of the set V is treated as a coloring of V.

Let us define the hyperedges of $\mathcal{H}^*$. First, let $\alpha\in\Omega^*$. Denote by Δ the class of π, containing α. Then $\Delta\in\pi_{\Omega^*}$. Moreover, for every $\Gamma\in\pi$, there is a unique class $\Lambda_\alpha(\Delta,\Gamma)$ of the equivalence relation $e_{\Delta,\Gamma}$, containing α. Put

$$\overline{\alpha} = \{\Lambda_\alpha(\Delta,\Gamma):\ \Gamma\in\pi\},$$

in particular, $\overline{\alpha}\subseteq V$. It is easily seen that $\overline{\alpha}=\overline{\beta}$ if and only if the vertices α and β are twins in X, lying in the same class of π. Next, let $\beta\in\Omega^*$ is adjacent to α in X, and Γ the class of π, containing β. Then every vertex in $\Lambda_\alpha(\Delta,\Gamma)$ is adjacent to every vertex of $\Lambda_\beta(\Gamma,\Delta)$. Put

$$\overline{\{\alpha,\beta\}} = \{\Lambda_\alpha(\Delta,\Gamma),\Lambda_\beta(\Gamma,\Delta)\},$$

again $\overline{\{\alpha,\beta\}}\subseteq V$. With these notation, the hyperedge set of $\mathcal{H}^*$ is defined as follows:

$$E^* = \{\overline{\alpha}:\ \alpha\in\Omega^*\}\cup\{\overline{\{\alpha,\beta\}}:\ \alpha,\beta\in\Omega^*,\ \beta\in\alpha X\}.$$

One can see that the hypergraph $\mathcal{H}^*=(V,E^*)$ and the coloring $\overline{\pi}$ can be constructed in polynomial time in $|\Omega|$.

Theorem 4.11. *Let X be a chordal graph, π an invariant stable vertex coloring of X, $\Omega^*=\Omega^*(X,\pi)$ the critical set, and $\mathcal{H}^*=(V,E^*)$ is the above hypergraph with vertex coloring $\overline{\pi}$. Then*

i $\max\{|\Delta|:\ \Delta\in\overline{\pi}\}\le \ell 2^\ell$, *where* $\ell=\ell(X)$,

ii if X is twinless, then the mapping $f:\Omega^*\to E^*$, $\alpha\mapsto\overline{\alpha}$, is a bijection,

iii if X is twinless, $G\le\mathrm{Sym}(E^*)$ is the group induced by the natural action of $\mathrm{Aut}(\mathcal{H}^*)$ *on $E^*=E(\mathcal{H}^*)$, then*

$$\mathrm{Aut}(X)^{\Omega^*}\le G^{f^{-1}}\le \mathrm{Aut}(X_{\Omega^*}),\tag{4.9}$$

where $G^{f^{-1}}=fGf^{-1}$.[1]

[1]Note that the composition fGf^{-1} is defined from left to right.

Proof. (i) The color classes of $\overline{\pi}$ are the sets $\Delta/e_{\Delta,\Gamma}$, where $\Delta \in \pi_{\Omega^*}$ and $\Gamma \in \pi$. By the definition of Ω^*, we have $|\operatorname{Conn}(X_\Delta)| \le \ell$, and Lemma 4.5(2) yields

$$|\operatorname{Conn}(X_{\Delta\cup\Gamma})| \le \min\{|\operatorname{Conn}(X_\Delta)|, |\operatorname{Conn}(X_\Gamma)|\} \le \ell. \tag{4.10}$$

Further, let $Y \in \operatorname{Conn}(X_{\Delta\cup\Gamma})$. Then by Lemma 4.3(2), the coloring π_Y is stable. It has two classes, one inside Δ and the other one inside Γ; denote them by Δ_Y and Γ_Y, respectively. Note that by Lemma 4.3(2), at least one of the graphs X_{Δ_Y}, X_{Γ_Y} is complete. From Lemma 4.9, we obtain

$$|\Delta/e_{\Delta,\Gamma}| \le \max\{\ell, 2^\ell\} \le 2^\ell. \tag{4.11}$$

Since the equivalence relation $e_{\Delta,\Gamma}$ is the union of the equivalence relations e_{Δ_Y,Γ_Y}, $Y \in \operatorname{Conn}(X_{\Delta\cup\Gamma})$, inequalities (4.10) and (4.11) imply

$$|\Delta/e_{\Delta,\Gamma}| = \sum_{Y\in\operatorname{Conn}(X_{\Delta\cup\Gamma})} |e_{\Delta_Y,\Gamma_Y}| \le \ell\, 2^\ell,$$

as required.

(ii) Assume that X is twinless. Let $\alpha \in \Omega^*$ and let $\Delta \in \pi$ contain α. Denote by Λ_α the intersection of all $\Lambda_\alpha(\Delta,\Gamma)$, $\Gamma \in \pi$. Note that every $\beta \in \Lambda_\alpha$ belongs to Δ. Moreover,

$$\alpha X \cap \Gamma = \beta X \cap \Gamma$$

for all $\Gamma \ne \Delta$, and

$$(\alpha X \cap \Delta) \setminus \{\beta\} = (\beta X \cap \Delta) \setminus \{\alpha\}.$$

It follows that α and β are twins in X. Since X is twinless, we conclude that $\alpha = \beta$. Thus,

$$\Lambda_\alpha = \{\alpha\} \quad \text{for all } \alpha \in \Omega^*.$$

Now assume that $f(\alpha) = f(\beta)$ for some $\alpha, \beta \in \Omega^*$. Then $\Lambda_\alpha = \Lambda_\beta$ and the above formula implies $\{\alpha\} = \Lambda_\alpha = \Lambda_\beta = \{\beta\}$. Thus, $\alpha = \beta$ and f is injective.

(iii) Assume that X is twinless. By (ii), the mapping f is a bijection. Let $g \in \operatorname{Aut}(X)$ and $\alpha \in \Omega^*$. Then α lies in some $\Delta \in \pi_{\Omega^*}$. Since π is invariant and stable, we have $e_{\Delta^g,\Gamma^g} = e_{\Delta,\Gamma}$ and so

$$\Lambda_\alpha(\Delta,\Gamma)^g = \Lambda_{\alpha^g}(\Delta,\Gamma)$$

for every $\Gamma \in \pi$. Therefore,

$$(\overline{\alpha})^{f^{-1}gf} = \{\Lambda_\alpha(\Delta,\Gamma)^g : \ \Gamma \in \pi\}^f = \{\Lambda_{\alpha^g}(\Delta,\Gamma) : \ \Gamma \in \pi\}^f = \overline{\alpha^g}$$

and

$$\overline{\{\alpha,\beta\}}^{f^{-1}gf} = \{\Lambda_\alpha(\Delta,\Gamma)^g, \Lambda_\beta(\Gamma,\Delta)^g\}^f = \{\Lambda_{\alpha^g}(\Delta,\Gamma), \Lambda_{\beta^g}(\Gamma,\Delta)\}^f = \overline{\{\alpha^g,\beta^g\}}. \tag{4.12}$$

Thus, $f^{-1}gf \in G$, which proves the left-hand side inclusion in (4.9).

Let $\alpha, \beta \in \Omega^*$. Denote by Δ and Γ the classes of π, containing α and β, respectively. Then α and β are adjacent in X if and only if every vertex in $\Lambda_\alpha(\Delta,\Gamma)$ is adjacent to every vertex of $\Lambda_\beta(\Gamma,\Delta)$, or equivalently, $\overline{\{\alpha,\beta\}} \in E^*$. Thus, the right-hand side inclusion in (4.9) follows from (4.12). □

4.5 A hypergraph associated with complement of the critical set

The goal of this section is to provide some tools related with critical set to find the automorphism group of a chordal graph.

Theorem 4.12. *Let X be a chordal graph on Ω, π an invariant stable coloring of X, and Ω^* the critical set of X with respect to π. Denote by $G^\diamond = G^\diamond(X)$ the kernel of the restriction homomorphism* $\mathrm{Aut}(X) \to \mathrm{Aut}(X)^{\Omega^*}$. *Then a generating set of $G^\diamond$ can be found in in polynomial time in $|\Omega|$.*

Proof. Without loss of generality, we may assume that the set $\Omega^\diamond = \Omega \setminus \Omega^*$ is not empty. Let us define a vertex coloring $\pi^\diamond$ of the graph $X^\diamond = X_{\Omega^\diamond}$, such that $\pi^\diamond(\alpha) = \pi^\diamond(\beta)$ if and only if $\pi(\alpha) = \pi(\beta)$ and $\alpha X \cap \Omega^* = \beta X \cap \Omega^*$. It is not hard to see that

$$(G^\diamond)^{\Omega^\diamond} = \mathrm{Aut}(X^\diamond, \pi^\diamond).$$

Since also the graph $X^\diamond$ is interval (see the definition of the critical set), a generating set of $(G^\diamond)^{\Omega^\diamond}$ can be found by the algorithm in [40, Theorem 3.4], which constructs a generating set of the automorphism group of a vertex colored interval graph efficiently. Since $(G^\diamond)^{\Omega^*} = \{\mathrm{id}_{\Omega^*}\}$, we are done. □

In what follows, X is a chordal graph, π a stable coloring of X, and Ω^* the critical set of X with respect to π. Recall that by the definition of critical set, every graph $\overline{Y}$, $Y \in \mathrm{Conn}(X_{\Omega^\diamond})$, is interval, and

$$\partial Y = \Omega(\overline{Y}) \cap \Omega^*.$$

Lemma 4.13. *For every $Y \in \mathrm{Conn}(X_{\Omega^\diamond})$, there is a colored hypergraph $H = H_Y$ whose vertex set is ∂Y colored by $\pi_{\partial Y}$, and such that*

$$\mathrm{Iso}(H_Y, H_{Y'}) = \mathrm{Iso}(\overline{Y}, \overline{Y'})^{\partial Y}, \quad Y' \in \mathrm{Conn}(X_{\Omega^\diamond}). \tag{4.13}$$

Moreover, in polynomial time in the size of $\overline{Y}$ one can

a construct the hypergraph H_Y,

b given $\overline{g} \in \mathrm{Iso}(H_Y, H_{Y'})$, find $g \in \mathrm{Iso}(\overline{Y}, \overline{Y'})^{\partial Y}$ such that $g^{\partial Y} = \overline{g}$.

Proof. We make use of the results of [61]. Namely, let Z be an interval graph and π_Z a stable coloring of Z. From [61, Theorem 6.10 and Proposition 6.4], it follows that there exists a canonical rooted tree $T = T(Z)$ and a stable coloring π_T of T such that

$$L(T) = \Omega(Z) \quad \text{and} \quad \pi_Z = (\pi_T)_{L(T)}. \tag{4.14}$$

The term "canonical" means that for every interval vertex colored graph Z', the isomorphisms between Z and Z' are related with the isomorphisms between T and $T' = T(Z')$ as follows:

$$\mathrm{Iso}(T, T')^{L(T)} = \mathrm{Iso}(Z, Z'). \tag{4.15}$$

Moreover, the proof of [61, Proposition 6.4] shows that the sizes of T and π_T are polynomials in $|\Omega(Z)|$, and T and π_T can be constructed in polynomial time.

Now let $Y \in \mathrm{Conn}(X_{\Omega^\diamond})$. Since the graph $\overline{Y}$ is interval, one can define the rooted tree $T = T(\overline{Y})$ as above. Next, for each vertex x of T, we introduce the following notation:

- $L(x)$ is the set of all descendants of x in T, lying in $L(T)$,
- T_x is the subtree of T rooted at x and such that $L(T_x) = L(x) \setminus \partial Y$,
- $F(T_x)$ is a string encoding the isomorphism type of the rooted tree T_x.

Now for each x with $L(x) \cap \partial Y \neq \varnothing$ and $L(x) \setminus \partial Y \neq \varnothing$, we delete from T all the vertices of T_x, except for x, and define the new color of x to be equal $(\pi_T(x), F(T_x))$. Denote the resulting tree and its vertex coloring by $T_1 = T_1(Y)$ and $\pi_1 = \pi_1(Y)$, respectively. Then

$$L(T_1) = \partial Y. \tag{4.16}$$

It is not hard to see that T_1 and π_1 can efficiently be constructed, and T and π_T are uniquely recovered from T_1 and π_1. In particular,

$$\mathrm{Iso}(T_1, T_1')^{\partial Y} = \mathrm{Iso}(\overline{Y}, \overline{Y}'), \tag{4.17}$$

where $Y' \in \mathrm{Conn}(X - \Omega^*)$ and $T_1' = T_1(Y')$, respectively, cfg., (4.14) and (4.15).

At this point we can define the required hypergraph $H_Y = (\partial Y, E_Y)$, where

$$E_Y = \{L(x) :\ x \in \Omega(T_1(Y))\}.$$

Note that $L(x) = L(y)$ if and only if $x = y$ or one of x, y is the descendants of the other in T_1, and if, say y is the descendant of x, then for each vertex $z \neq y$ of the path P_{yx} connecting y with x, we have $L(x) = L(z)$; moreover, in the latter case, z has a unique son in T_1. Thus for each $e \in E_Y$ there exist uniquely determined vertex x_e and its descendant y_e such that

$$L(z) = e \ \Leftrightarrow\ z \in \Omega(P_{y_e x_e}) \text{ and } z \neq y_e \text{ if } x_e \neq y_e.$$

In particular, if $x_e = y_e := x$, then $L(x) = L(z)$ if and only if $x = z$.

To define the color of the hyperedge $e \in E_Y$, let $\Omega(P_{y_e x_e}) = \{y_0, y_1, \ldots y_k\}$, where $k \geq 0$ is the length of $P_{y_e x_e}$, $y_0 = y_e$, $y_k = x_e$, and y_i is the son of y_{i+1}, $i = 0, \ldots, k-1$. Then the color of e is set to be the tuple

$$(\pi_1(y_1), \ldots, \pi_1(y_k)).$$

Again, it is clear that the hypergraph H_Y and its coloring can be constructed efficiently and that they determine the colored tree T_1 in a unique way. Thus the statement of the lemma is a consequence of formulas (4.16) and (4.17). $\square$

Let us define an order-2 colored hypergraph $\mathcal{H}^\diamond$ with vertex set Ω^* and hyperedge set $\mathcal{E}_1 \cup \mathcal{E}_2$, where

$$\mathcal{E}_1 = \bigcup_{Y \in \mathrm{Conn}(X_{\Omega^\diamond})} E(H_Y) \quad \text{and} \quad \mathcal{E}_2 = \{E(H_Y) :\ Y \in \mathrm{Conn}(X_{\Omega^\diamond}\}.$$

The vertex coloring of $\mathcal{H}^\diamond$ is set to be π_{Ω^*}. Note that the union in the definition of $\mathcal{E}_1$ is not disjoint; the color $\pi^\diamond(e)$ of a hyperedge $e \in \mathcal{E}_1$ is defined to be the multiset of the colors of e in $\mathcal{H}_Y$, where Y runs over all graphs $Y \in \mathrm{Conn}(X_{\Omega^\diamond})$ such that $e \in E(H_Y)$.

To define a coloring of $\mathcal{E}_2$, denote by $\sim$ the equivalence relation on $\mathrm{Conn}(X_{\Omega^\diamond})$ by setting

$$Y \sim Y' \quad \Leftrightarrow \quad H_Y = H_{Y'}.$$

Condition (4.13) implies that $Y \sim Y'$ if and only if there exists an isomorphism $g \in \mathrm{Iso}(\overline{Y}, \overline{Y}')$ such that the bijection $g^{\partial Y}$ is identical. The color $\pi^\diamond(e)$ of the hyperedge $e \in \mathcal{E}_2$ is defined to be so that if $e = \{E(H_Y)\}$ and $e' = \{E(H_{Y'})\}$, then

$$\pi^\diamond(e) = \pi^\diamond(e') \quad \Leftrightarrow \quad \mathrm{Iso}(\overline{Y}, \overline{Y}') \neq \varnothing \quad \text{and} \quad n_Y = n_{Y'}, \tag{4.18}$$

where n_Y and $n_{Y'}$ are the cardinalities of the classes of the equivalence relation $\sim$, containing Y and Y', respectively.

Remark 4.14. Let $e \in \mathcal{E}_2$ and $Y \in \mathrm{Conn}(X - \Omega^*)$ be such that $e = E(H_Y)$. In general, the coloring π_e of the hyperedges of $\mathcal{E}_1$, contained in e, is different from the coloring π_Y of the corresponding hyperedges of H_Y. However, $\pi_e \geq \pi_Y$ and π_Y is uniquely determined by π_e.

Lemma 4.15. *Let X' be a colored graph obtained from X by deleting all edges of the induced subgraph X_{Ω^*}. Then*

$$\mathrm{Aut}(\mathcal{H}^\diamond) = \mathrm{Aut}(X')^{\Omega^*}.$$

Moreover, given $\overline{g} \in \mathrm{Aut}(\mathcal{H}^\diamond)$ one can construct $g \in \mathrm{Aut}(X')$ such that $g^{\Omega^*} = \overline{g}$ in polynomial time in $|\Omega|$.

Proof. Let $g \in \mathrm{Aut}(X')$. Since the set Ω^* is $\mathrm{Aut}(X')$-invariant, the permutation $\overline{g} = g^{\Omega^*}$ preserves the coloring π_{Ω^*}. Moreover, g induces a permutation

$$Y \mapsto Y', \quad Y \in \mathrm{Conn}(X - \Omega^*), \tag{4.19}$$

such that $(\partial Y)^g = \partial Y'$ for all Y, and the isomorphisms

$$g_Y \in \mathrm{Iso}(\overline{Y}, \overline{Y'}), \quad Y \in \mathrm{Conn}(X - \Omega^*).$$

By formula (4.13), we have $(g_Y)^{\partial Y} = g^{\partial Y} \in \mathrm{Iso}(H_Y, H_{Y'})$. Now, let $e \in \mathcal{E}_1$. Then $e \in E(H_Y)$ for some $Y \in \mathrm{Conn}(X - \Omega^*)$. It follows that

$$e^g = e^{g_Y} \in E(H_{Y'}) \quad \text{for all } \; e \in E(H_Y).$$

Consequently, the permutation $\overline{g}$ preserves the hyperedges of $\mathcal{E}_1$. Because the isomorphism g_Y is color preserving, $\overline{g}$ preserves also the colors of them. Finally, the automorphism $g \in \mathrm{Aut}(X')$ preserves the relations on the right-hand side of formula (4.18) and hence the permutation (4.19) leaves the equivalence relation $\sim$ fixed. Since g induces the same permutation, we conclude that $\overline{g}$ preserves the colors of the hyperedges of $\mathcal{E}_2$. Thus, $\overline{g} \in \mathrm{Aut}(\mathcal{H}^\diamond)$.

Conversely, let $\overline{g} \in \mathrm{Aut}\,\mathcal{H}^\diamond$. Formula (4.18) implies that $\overline{g}$ induces a cardinality preserving permutation of the classes of the equivalence relation $\sim$. Consequently, there is a permutation (4.19) such that $\mathrm{Iso}(H_Y, H_{Y'}) \neq \varnothing$; although such a permutation is not necessarily unique, one can efficiently find at least one such permutation.

Recall that $\mathcal{E}_1^{\overline{g}} = \mathcal{E}_1$. Moreover, the hyperedges from $E(H_Y) \in \mathcal{E}_2$ go to the edges from $E(H_{Y'}) \in \mathcal{E}_2$. Therefore (see Remark 4.14),

$$\overline{g}_Y := \overline{g}^{\partial Y} \in \mathrm{Iso}(H_Y, H_{Y'}). \tag{4.20}$$

By formula (4.13), there exists a bijection $g_Y \in \mathrm{Iso}(\overline{Y}, \overline{Y'})$ such that

$$g_Y^{\partial Y} = \overline{g}_Y, \tag{4.21}$$

and this bijection can efficiently be found (Lemma 4.13(b)). Now we define a permutation $g \in \mathrm{Sym}(\Omega)$ by setting $\alpha^g = \alpha^{g_Y}$, where Y is an arbitrary element of $\mathrm{Conn}((X - \Omega^*)$, for which $\alpha \in \Omega(\overline{Y})$. The permutation g is well defined, because by (4.20) and (4.21),

$$\alpha^{g_Y} = \alpha^{\overline{g}_Y} = \alpha^{\overline{g}} = \alpha^{\overline{g}_Z} = \alpha^{g_Z}$$

for all $Z \in \mathrm{Conn}((X - \Omega^*)$ and all $\alpha \in \partial Y \cap \partial Z$. It remains to note that $g \in \mathrm{Aut}(X')$, because g moves edges of each $\overline{Y}$ to $\overline{Y'}$, and $E(X')$ is the union of the sets $E(\overline{Y})$. □

The following theorem is the main result of the section, which together with Theorem 4.12 essentially provides a polynomial-time reduction of finding the group $\mathrm{Aut}(X)$ to finding the groups $\mathrm{Aut}(\mathcal{H}^*)$ and $\mathrm{Aut}(\mathcal{H}^\diamond)$.

Theorem 4.16. *In the conditions of Theorem 4.11,*

$$\mathrm{Aut}(X)^{\Omega^*} = \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}} \cap \mathrm{Aut}(\mathcal{H}^\diamond).$$

Moreover, every permutation $\overline{g} \in \mathrm{Aut}(\mathcal{H}^\diamond) \cap \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}}$ can be lifted to an automorphism $g \in \mathrm{Aut}(X)$ such that $g^{\Omega^*} = g$ in polynomial time in $|\Omega|$.

Proof. By Theorem 4.11, we have $\mathrm{Aut}(X)^{\Omega^*} \leq \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}}$. Furthermore, the group $\mathrm{Aut}(X)$ is a subgroup of $\mathrm{Aut}(X')$, where X' is the graph from Lemma 4.15. By that lemma, this implies that $\mathrm{Aut}(X)^{\Omega^*} \leq \mathrm{Aut}(X')^{\Omega^*} = \mathrm{Aut}(\mathcal{H}^\diamond)$. Thus,

$$\mathrm{Aut}(X)^{\Omega^*} \leq \mathrm{Aut}(\mathcal{H}^\diamond) \cap \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}}.$$

Conversely, let $\overline{g} \in \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}} \cap \mathrm{Aut}(\mathcal{H}^\diamond)$. By Lemma 4.15, one can efficiently find $g \in \mathrm{Aut}(X')$ such that $g^{\Omega^*} = \overline{g}$. Now, by Theorem 4.11 the permutation g preserves the edges of X contained in $E(X_{\Omega^*})$. The other edges of X are exactly those in $E(X')$ and g preserves them by Lemma 4.15. Thus,

$$E(X)^g = (E(X_{\Omega^*}) \cup E(X'))^g = E(X_{\Omega^*})^g \cup E(X')^g = E(X_{\Omega^*}) \cup E(X') = E(X),$$

i.e., $g \in \mathrm{Aut}(X)$, as required. □

4.6 Order-k hypergraph isomorphism: bounded color classes

The goal of this section is to design an FPT algorithm for testing isomorphism of colored k-hypergraphs in which the sizes of vertex color classes are bounded by a fixed parameter; no assumption is made on the hyperedge color class sizes. The algorithm we present is a generalization of the one for usual hypergraphs [4].

Theorem 4.17. *Let $k \geq 1$. Given two colored order-k hypergraphs H and H', the isomorphism coset* $\mathrm{Iso}(H, H')$ *can be computed in time $(b!\, s)^{O(k)}$, where b is the maximal size of a vertex color class of H and s is the size of H. In particular, the group* $\mathrm{Aut}(H)$ *can be found within the same time.*

The proof of Theorem 4.17 is given in the end of the section. We start with some notation and definitions; most of them go back to those in [4]. In what follows, we fix a finite set V and the decomposition of V into the disjoint union of its color classes,

$$V = C_1 \sqcup C_2 \sqcup \cdots \sqcup C_m, \tag{4.22}$$

where $m \geq 1$ and $|C_i| \leq b$ for each i. For every higher order hyperedge e, we consider its projections to unions of the color classes,

$$e^{(i)} = e^{C_1 \cup C_2 \cup \cdots \cup C_i}, \quad 0 \leq i \leq m,$$

see Section 4.2. Obviously, $e^{(0)} = \varnothing$ and $e^{(m)} = e$.

i-equivalence. Let $i \in \{0, \ldots, m\}$. Two order-k hyperedges e and e' are said to be *i-equivalent* if the multisets $e^{(i)}$ and ${e'}^{(i)}$ are equal. The following statement is straightforward.

Proposition 4.18.

1. *any two high order hyperedges are 0-equivalent,*
2. *for $i \geq 1$, any two i-equivalent high order hyperedges are $(i-1)$-equivalent,*
3. *two high order hyperedges are m-equivalent if and only if they are equal.*

i-blocks. Let $H = (V, E)$ be an order-k hypergraph. For every $i = 0, \ldots, m$, the i-equivalence partitions the set E into equivalence classes called *i-blocks*; the set of all of them is denoted by $\widehat{E}_i$. From Proposition 4.18, it follows that

$$\widehat{E}_0 = \{E\} \quad \text{and} \quad \widehat{E}_m = E. \tag{4.23}$$

Hypergraphs $A[i]$ associated with i-blocks. Each i-block $A \in \widehat{E}_i$ defines an order-k hypergraph (V, A), which is just H if $i = 0$, and is essentially the order-$(k-1)$ hypergraph (V, e) if $i = m$ and $A = \{e\}$ for some $e \in E$. Denote by $A[i]$ the order-k hypergraph on the set

$$V_i = C_i \sqcup C_{i+1} \sqcup \cdots \sqcup C_m,$$

obtained from the projection A^{V_i} of A to V_i by replacing each multiset e^{V_i}, $e \in A$ with the corresponding set (without repetitions). Then $A[0] = H$.

Coloring of $A[i]$. Assume that the hypergraph H is colored. The vertex coloring of the hypergraph $A[i]$ is defined in a natural way, whereas the color of the hyperedge corresponding to e^{V_i} is defined as a multiset

$$\{\{\pi(\widetilde{e}):\ \widetilde{e}^{\,V_i}=e^{V_i},\ \widetilde{e}\in A\}\},$$

where π is the coloring of $E(H)$. One can see that if $H'=(V,E')$ is an order-k hypergraph, $A'\in\widehat{E}'_i$, and $f\in\mathrm{Iso}(H,H')$ is such that f^{V_i} is a color preserving isomorphism from $A[i]$ to $A'[i]$, then f is also color preserving.

Proof of Theorem 4.17. Let $H=(V,E)$ and $H'=(V',E')$ be colored order-k hypergraphs. Without loss of generality we may assume that there is a decomposition of V' similar to (4.22) with the same m and b. Our aim is to design an algorithm of running time $x(k,s,b)=(b!\,s)^{O(k)}$ for computing the coset $\mathrm{Iso}(H,H')$.

Inductively, assume that $k\geq 2$ and we have such an algorithm for order-$(k-1)$ hypergraphs of running time $x(k-1,s,b)$. As the base case for the induction, by [4, Corollary 9], we already have

$$x(1,s,b)=2^{O(b)}\,\mathrm{poly}(s). \tag{4.24}$$

The algorithm for order-k hypergraphs will invoke as subroutine the algorithm for order-$(k-1)$ hypergraphs. Put

$$C(k,i;H,H')=\{\mathrm{Iso}(A[i],A'[i]):\ A\in\widehat{E}_i,\ A'\in\widehat{E}'_i\},\quad 0\leq i\leq m.$$

The algorithm below computes the collections $C(k,i;H,H')$ for decreasing values of i from m down to 0. Specifically, for each i, it first computes the set $C(k,i+1;H,H')$ and uses it for computing the set $C(k,i;H,H')$. Since $A[0]=H$ and $A'[0]=H'$, notice that we will finally have computed $\mathrm{Iso}(H,H')=C(k,0;H,H')$ as required.

Algorithm for computing $C(k,0;H,H')$

Input: colored order-k hypergraphs $H=(V,E)$ and $H'=(V',E')$, $k>1$.

Output: the table of all $C(k,i;H,H'), 0\leq i\leq m$.

For $i:=m$ **down to** 1 **do**

for all $A\in\widehat{E}_i$ and $A'\in\widehat{E}'_i$ add to $C(k,i;H,H')$ the coset $\mathrm{Iso}(A[i],A'[i])$ computed below.

Step 0. If $i=m$

then $A[i]$ and $A'[i]$ are order-k hypergraphs on the sets C_m and C'_m of cardinality at most b. In this case $\mathrm{Iso}(A[i],A'[i])$ can be computed in time $O(b!)$.

else

Step 1. Construct the $(k-1)$-skeleton hypergraphs $Y=A[i]^{(k-1)}$ and $Y'=A'[i]^{(k-1)}$ (see Section 4.2).

Step 2. Compute $K\tau:=\mathrm{Iso}(Y,Y')=C(k-1,0;Y,Y')$ by using the algorithm for order-$(k-1)$ hypergraphs as subroutine.

Step 3. Computation of $\mathrm{Iso}(A[i],A'[i])$:

Step 3.1. Let $A_1,A_2,\ldots,A_\ell$ and $A'_1,A'_2,\ldots,A'_{\ell'}$ be the $(i+1)$-blocks contained in A and A', respectively; if $\ell\neq\ell'$, then set $\mathrm{Iso}(A[i],A'[i])=\varnothing$.

Step 3.2. Find the set $P \leq \mathrm{Sym}(\ell)$ of all permutations induced by $K\tau$ as the bijections from C_{i+1} to C'_{i+1} which maps the set $\{A_1, A_2, \ldots, A_\ell\}$ to $\{A'_1, A'_2, \ldots, A'_\ell\}$; note that $|P| \leq b!$.

Step 3.3. Using the algorithm in [4, Theorem 5], compute the coset

$$\mathrm{Iso}(A[i], A'[i]) = \bigcup_{\pi \in P} \bigcap_{j=1}^{\ell}, \mathrm{Iso}(A_j[i+1], A'_{\pi(j)}[i+1]) \tag{4.25}$$

where the cosets on the right-hand side are available from the set $C(k, i+1; H, H')$ found earlier.

end-for

Correctness and Analysis. By induction, it suffices to see how Step 3 computes $\mathrm{Iso}(A[i], A'[i])$. Notice that the union on the right-hand side of (4.25) with P replaced by the set of *all* bijections from C_{i+1} to C'_{i+1} gives the coset $H\nu$ of all isomorphisms from $A[i]$ to $A'[i]$ *projected* to V_{i+1} and V'_{i+1}. Since A and A' are i-blocks, they are single order-k hyperedges on color class C_i and C'_i, respectively. In view of formula (4.1), the coset $K\tau = \mathrm{Iso}(Y, Y')$ restricted to C_i and C'_i precisely includes all the isomorphisms from $A[i]$ to $A'[i]$ restricted to C_i and C'_i. Hence, $K\tau \cap H\nu$ is precisely $\mathrm{Iso}(A[i], A'[i])$ which is computed at Steps 3.3.

We now analyze the running time $x(k, s, b)$ for the computation of the set $C(k, 0; H, H')$. The outer for-loop executes m times and the inner for-loop executes at most $|E|^2$ times (for each pair A, A' of i-blocks).

We bound the time required for computing each $C(k, i; H, H')$. By induction, each iteration of Steps 0-2 require time

$$O(|E|^2 \cdot b!) + x(k-1, s, b) + \mathrm{poly}(s).$$

The number ℓ in Step 3.1 is at most $|E|$. Therefore the cost of Steps 3.1-3.2 is at most $|E|\,|P|\,\mathrm{poly}(s) \leq b!\,\mathrm{poly}(s)$. Finally, in Step 3.3, we compute at most $b!$ intersections of ℓ cosets available in the already computed set $C(k, i+1, H, H')$. Since the intersection of two such cosets by the algorithm from [4, Theorem 5] requires $2^{O(b)} \cdot \mathrm{poly}(s)$ time, the overall cost of Step 3 is at most $O(b!)\,\mathrm{poly}(s)$. Putting it together, the time spent in computing $C(k, i; H, H')$, given the pre-computed table entries for $C(k, i+1, H, H')$, is bounded by $x(k-1, s, b) \cdot O(b!)\,\mathrm{poly}(s)$. It follows that the overall time for computing $C(k; H, H')$ is bounded by $m \cdot |E|^2 \cdot x(k-1, s, b) \cdot O(b!)\,\mathrm{poly}(s)$. Thus, we have

$$x(k, s, b) \leq m \cdot |E|^2 \cdot x(k-1, s, b) \cdot O(b!)\,\mathrm{poly}(s) \leq x(k-1, s, b) \cdot (b! \cdot s)^c,$$

for a suitable constant $c > 0$. By induction hypothesis $x(k-1, s, b) \leq (b! \cdot s)^{c \cdot (k-1)}$. Hence, we obtain an overall upper bound of $(b! \cdot s)^{c \cdot k}$ for the running time of the algorithm for order-k hypergraphs.

□

4.7 Proof of Theorem 4.2

Based on the results obtained in the previous sections, we present an algorithm that constructs the automorphism group of a chordal twinless graph.

Main Algorithm

Input: a chordal twinless graph X and vertex coloring π of X.
Output: the group $\mathrm{Aut}(X,\pi)$.

Step 1. Construct $\pi = \mathrm{WL}(X,\pi)$ and $\Omega^* = \Omega^*(X,\pi)$.

Step 2. While the set Ω^* is not critical with respect to π, find $\pi := \mathrm{WL}(X,\pi')$ and set $\Omega^* := \Omega^*(X,\pi)$, where π' is the coloring from Theorem 4.10(ii).

Step 3. If $\Omega^* = \varnothing$, then X is interval and we output the group $\mathrm{Aut}(X,\pi)$ found by the algorithm from [118, Theorem 5].

Step 4. Construct the mapping f and colored hypergraph $\mathcal{H}^*$ on $(\Omega^*)^f$, defined in Section 4.4, and the colored hypergraph $H^\diamond$ on Ω^*, defined in Section 4.5.

Step 5. Using the algorithm from Theorem 4.17, find a generating set $\overline{S}$ of the automorphism group of the colored order-3 hypergraph $\mathcal{H}^* \uparrow (\mathcal{H}^\diamond)^f$.

Step 6. For each $\overline{g} \in \overline{S}$ find a lifting $g \in \mathrm{Aut}(X,\pi)$ of $f\overline{g}f^{-1} \in \mathrm{Sym}(\Omega^*)$ by the algorithm from Theorem 4.16; let S be the set of all these automorphisms g's.

Step 7. Output the group $\mathrm{Aut}(X,\pi) = \langle G^\diamond, S\rangle$, where $G^\diamond$ is the group defined in Theorem 4.12.

Theorem 4.19. *The Main Algorithm correctly finds the group* $\mathrm{Aut}(X,\pi)$ *in time* $t(\ell)\cdot n^{O(1)}$, *where* $n = |\Omega(X)|$, g *is a function independent of* n, *and* $\ell = \ell(X)$.

Proof. Note that the number of iterations of the loop at Step 2 is at most n, because $|\pi| \le n$ and $|\pi'| > |\pi|$. Next, the running time at each other step, except for Step 5, is bounded by a polynomial in n, see the time bounds in the used statements. On the other hand, at Step 5, the cardinality of each vertex color class of the order-3 hypergraph $\mathcal{H}^* \uparrow (\mathcal{H}^\diamond)^f$ is at most $\ell 2^\ell$ (Theorem 4.11(i)). By Theorem 4.17 for $b = \ell 2^\ell$, the running time of the Main Algorithm is at most $t(\ell)\cdot n^{O(1)}$ with $t(\ell) = (\ell 2^\ell)!$.

To prove the correctness of the algorithm, we exploit the natural restriction homomorphism

$$\varphi : \mathrm{Aut}(X) \to \mathrm{Sym}(\Omega^*),\ g \mapsto g^{\Omega^*}.$$

Given a generating set S' of the group $\Im(\varphi)$, we have $\mathrm{Aut}(X) = \langle \mathrm{Ker}(\varphi), S\rangle$, where $S \subseteq \mathrm{Aut}(X)$ is a set of cardinality $|\overline{S}|$ such that $S' = \{\varphi(g) :\ g \in S\}$.

According to Step 7, $\mathrm{Ker}(\varphi) = G^\diamond$. Thus, it suffices to verify that as the set S' one can take the set $\{f\overline{g}f^{-1} :\ \overline{g} \in \overline{S}\}$, where f is the bijection found at Step 4 and $\overline{S}$ is the generating set of the group $\mathrm{Aut}(\mathcal{H}^* \uparrow (\mathcal{H}^\diamond)^f)$, found at Step 5. By Theorem 4.16, we need to check that

$$\mathrm{Aut}(\mathcal{H}^* \uparrow (\mathcal{H}^\diamond)^f)^{f^{-1}} = \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}} \cap \mathrm{Aut}(\mathcal{H}^\diamond). \tag{4.26}$$

Notice that

$$\begin{aligned} h \in \mathrm{Aut}(\mathcal{H}^* \uparrow (\mathcal{H}^\diamond)^f) &\Leftrightarrow h \in \mathrm{Aut}(\mathcal{H}^*) \quad \text{and} \quad (E(\mathcal{H}^\diamond)^f)^h = E(\mathcal{H}^\diamond)^f \\ &\Leftrightarrow fhf^{-1} \in \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}} \quad \text{and} \quad fhf^{-1} \in \mathrm{Aut}(\mathcal{H}^\diamond) \\ &\Leftrightarrow fhf^{-1} \in \mathrm{Aut}(\mathcal{H}^*)^{f^{-1}} \cap \mathrm{Aut}(\mathcal{H}^\diamond), \end{aligned}$$

which proves equality (4.26). $\square$

Proof of Theorem 4.2. Denote by e_X the equivalence relation on $\Omega = \Omega(X)$ such that $(\alpha, \beta) \in e_X$ if and only if the vertices α and β are twins in X. Since e_X is $\mathrm{Aut}(X)$-invariant, there is a natural homomorphism

$$\varphi : \mathrm{Aut}(X) \to \mathrm{Sym}(\Omega/e_X).$$

To find the group $\mathrm{Aut}(X)$, it suffices to construct generating sets of the groups $\mathrm{Ker}(\varphi)$ and $\Im(\varphi)$, and then to lift every generator of the latter to an automorphism of X.

First, we note that every class of the equivalence relation e_X consists of twins of X. Consequently,

$$\mathrm{Ker}(\varphi) = \prod_{\Delta \in \Omega/e_X} \mathrm{Sym}(\Delta),$$

and this group can efficiently be found.

Now let X' be the graph with vertex set Ω/e, in which the classes Δ and Γ are adjacent if and only if some (and hence each) vertex in Δ is adjacent to some (and hence each) vertex of Γ. Note that X' is isomorphic to an induced subgraph of X, and hence belongs to the class $\mathfrak{K}_\ell$. Let π' be the vertex coloring of X' such that $\pi'(\Delta) = \pi'(\Gamma)$ if and only if $|\Delta| = |\Gamma|$. Then

$$\Im(\varphi) = \mathrm{Aut}(X', \pi'),$$

and this group can efficiently be found in time $t(\ell) \cdot n^{O(1)}$ by Theorem 4.19.

To complete the proof, we need to show that given $g' \in \mathrm{Aut}(X', \pi')$, one can efficiently find $g \in \mathrm{Aut}(X)$ such that $\varphi(g) = g'$. To this end, choose an arbitrary bijection $g_\Delta : \Delta \to \Delta^{\overline{g}}$; recall that $\pi'(\Delta) = \pi'(\Delta^{\overline{g}})$ and so $|\Delta| = |\Delta^{\overline{g}}|$. Then the mapping g taking a vertex $\alpha \in \Omega$ to the vertex α^{g_Δ}, where Δ is the class of e_X, containing α is a permutation of Ω. Moreover, from the definition of e_X, it follows that $g \in \mathrm{Aut}(X)$. It remains to note that g can efficiently be constructed. $\square$

4.8 GI-completeness for H-graphs

In the previous sections, we derived an FPT algorithm for testing isomorphism of bounded leafage chordal graphs. In this last section, we prove a hardness result for related classes of graphs, called H-graphs, which were already mentioned briefly in the beginning of the chapter.

For a graph H let $\mathcal{S}(H)$ be the set of all of its connected subgraphs. A *subdivision* of a graph H is obtained by replacing the edges of H by internally disjoint paths of arbitrary length. An *H-representation* of a graph G is a mapping $R\colon V(G) \to \mathcal{S}(H)$ such that $R(v) \cap R(u) \neq \varnothing$ if and only if the vertices u and v are adjacent. We say that a graph G is an *H-graph* if there is a subdivision H' of H such that G has an H'-representation.

This concept generalizes many importnat classes of graphs: interval graphs are K_2-graphs, circular-arc graphs are K_3-graphs, chordal graphs are the union of all T-graphs, where T is a tree. Moreover, it is easy to see that every graph is an H-graph for a suitable graph H. Thus, H-graphs provide an interesting parametrization of all graphs.

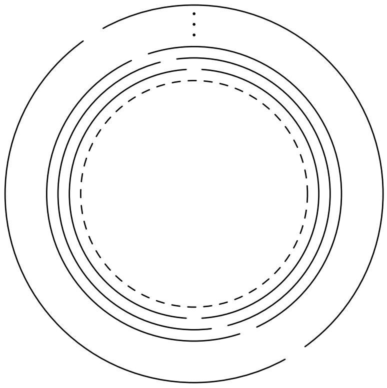


Figure 4.1: A circular-arc representation of the complement of a matching $K_{2n} - 2K_n$.

We prove that if H is not a unicyclic graph, then the isomorphism problem on H-graphs is *GI-complete*, i.e., as hard as the general graph isomorphism problem. The key is the following theorem, where B denotes the 5-vertex graph consisting of two triangles with one vertex identified.

Theorem 4.20. *The class of all B-graphs is GI-complete.*

Note that if a graph H_1 is a minor of a graph H_2, then every H_1-graph is an H_2-graph. This immediatelly gives the following.

Corollary 4.21. *If H is not unicyclic, then the class of all H-graphs is GI-complete.*

To prove Theorem 4.20, we first prove several lemmas.

Lemma 4.22. *If G be a disjoint union of stars, then the complement of G is a circular-arc graph.*

Proof. Let $X \subseteq V(G)$ be the vertices of degree 1 in G and let $Y = V(G) \setminus X$. For every vertex $v \in Y$, the neighborhood $N_G(v)$ of v forms an equivalence class of true twins in $\overline{G}$. The quotient graph of $\overline{G}$ with respect to this equivalence relation is exactly the complement of a matching, which is a well-known circular-arc graph; see Figure 4.1. This implies that $\overline{G}$ is also a circular-arc graph. □

For any connected graph $G = (V, E)$, we construct a new graph G' in several steps as follows. First, let $G_1 = (V \cup V_1, E')$ be the graph resulting from sudividing each edge of G and let $G_2 = (V \cup V_1 \cup V_2, E'')$ be the graph resulting from subdividing each edge of G_1. Further, let G_3 be the graph obtained from G_2 by adding all the edges between V and V_1. Finally, we set the graph G' to be the complement of G_3.

Lemma 4.23. *Let G be a connected graph. With the above notation, the graph G' is an B-graph.*

Proof. First note that the induced subraphs $G_2[V \cup V_2]$ and $G_2[V_1 \cup V_2]$ are disjoint unions of stars. Thus, by Lemma 4.22, the induced subgraphs $G'[V \cup V_2]$ and $G'[V_1 \cup V_2]$ are circular-arc graphs with representations $R_1 \colon V \cup V_2 \to \mathcal{S}(C_1)$ and

$R_2\colon V_1 \cup V_2 \to \mathcal{S}(C_2)$, respectively, for some cycles C_1 and C_2. Moreover, both representations can be chosen so that there are points $p_1 \in V(C_1)$ and $p_2 \in V(C_2)$ such that

$$p_1 \in \bigcap_{v \in V_2} R_1(v), \quad p_1 \notin \bigcup_{v \in V} R_1(v), \quad \text{and} \quad p_2 \in \bigcap_{v \in V_2} R_2(v), \quad p_2 \notin \bigcup_{v \in V_1} R_2(v). \tag{4.27}$$

Let H be the graph obtained from C_1 and C_2 by indentifying the points p_1 and p_2. Clearly H is a subdivision of B. Let $R\colon V \cup V_1 \cup V_2 \to \mathcal{S}(H)$ be the mapping defined by

$$R(v) = \begin{cases} R_1(v) & \text{if } v \in V, \\ R_1(v) \cup R_2(v) & \text{if } v \in V_2, \\ R_2(v) & \text{if } v \in V_1. \end{cases}$$

Since there are no edges between V and V_1 in G', it follows from (4.27) that R is an H-representation of G' and, therefore, G' is an B-graph. □

Lemma 4.24. *Let G and H be connected graphs with minimum degree at least three. With the above notation, the graphs G and H are isomorphic if and only if the graphs G' and H' are isomorphic.*

Proof. Since subdividing each edge and taking complements preserves the isomorphism relation, it follows that $G \cong H$ if and only if $G_2 \cong H_2$, and, $G_3 \cong H_3$ if and only if $G' \cong H'$. Thus, it suffices to prove that $G_2 \cong H_2$ if and only if $G_3 \cong H_3$.

Let $f\colon G \to H$ be an isomorphism and let $f_2\colon G_2 \to H_2$ be the isomorphism such that $f_2|_{V(G)} = f$. We have

$$f_2(V(G)) = V(H) \quad \text{and} \quad f_2(V_1(G)) = V_1(H). \tag{4.28}$$

The graph G_3 is constructed by adding all edges between $V(G)$ and $V_1(G)$. Likewise, the graph H_3 is constructed by adding all edges between $V(H)$ and $V_1(H)$. From (4.28) it follows that f_2 is also an isomorphism from G_3 to H_3.

For the reverse implication, first note that for any vertex v of G_3, we have

$$\deg_{G_3}(v) = \begin{cases} \deg_G(v) + |V_1(G)| & \text{if } v \in V(G), \\ 2 + |V(G)| & \text{if } v \in V_1(G), \\ 2 & \text{if } v \in V_2(G). \end{cases}$$

Note that the degree is constant on $V_1(G)$ and on $V_2(G)$. An analogous formula is true for every vertex in H_3. Since the minimum degree in G is three, we have

$$2 + |V(G)| \geq 6 \quad \text{and} \quad \deg_G(v) + |V_1(G)| \geq 3 + 3/2|V(G)| > 2 + |V(G)|.$$

Again, analogous inequalities hold in H_3. We have that an isomorphism $f_3\colon G_3 \to H_3$ satisfies

$$f_3(V(G)) = V(H), \quad f_3(V_1(G)) = V_1(H) \quad \text{and} \quad f_3(V_2(G)) = V_2(H).$$

Thus, f_3 is also an isomorphism $G_2 \to H_2$. □

Proof of Theorem 4.20. Clearly, for a graph G, the graph G' can be constructed in polynomial time. By Lemma 4.23, G' is an 8-graph. Clearly the class of all graphs with minimum degree three is GI-complete. By Lemma 4.24, two graphs G and H with minimum degree three are isomorphic if and only if the graphs G' and H' are isomorphic. It follows that the class of all B-graphs is GI-comlete. □

5. Automorphism groups of planar graphs

5.1 Introduction

Automorphism groups of planar graphs were first studied by Babai in 1973. The automorphism groups of 3-connected planar graphs are the well-known spherical groups. Thus, it is natural to try to reduce the problem to 3-connected planar graphs. Indeed, Babai first decomposes a planar graph into blocks and after that each block into 3-connected planar graphs. The key problem is to understand how the automorphism group of a planar graph can be reconstructed from the automorphism groups of these 3-connected components. At some level, Babai solved this problem in [8, Corollary 8.12], however, he points out that his characterization is not inductive:

> "For the case of planar graphs, we determine the groups occurring in the Main Theorem, as abstract groups (up to isomorphism). [...] It cannot be, however, considered as a characterization by recursion of the automorphism groups of the planar graphs, since the group construction refers to the action of the constituents of the wreath products."

Our characterization. The characterization of the automorphism groups of planar graphs, described in this chapter, is inductive. The proof uses three main ideas/concepts representing three fields of mathematics: graph theory, group theory and geometry.

The first key idea, is the idea of *3-connected reduction*. The reduction can be viewed as a function associating to a given planar graph X an irreducible planar graph X^R which is either 3-connected, cycle, K_2 or K_1. The crucial feature of the reduction is that information on the automorphism group is preserved, i.e., the automorphism group of the original graph can be reconstructed from X^R applying a reverse procedure. The idea of the 3-connected reduction was first introduced in the seminal papers by Mac Lane [121] and Trakhtenbrot [156]. It was further extended in [157, 91, 92, 46, 160, 16]. The related decomposition (onto 3-connected components) can be represented by a tree whose nodes are 3-connected graphs, and this tree is known in the literature mostly under the name *SPQR-tree* [54, 55, 56, 87]. The main difference, compared to our approach, is that the reduction used in the former literature applies exclusively to 2-connected graphs. Note that a similar two-stage reduction was used by Babai as well. We introduce a reduction which reduces simultaneously inclusion minimal parts separated by 1-cuts and 2-cuts. This allows one to control the changes of the symmetries in each elementary reduction, see Theorem 5.10 for details. A similar reduction was used in [64, 65] to study the behaviour of semiregular subgroups of $\mathrm{Aut}(X)$ with respect to 1-cuts and 2-cuts. The 3-connected reduction and its behaviour with respect to automorphisms of a graph is investigated in Section 5.3.

The *inhomogeneous wreath product* of groups is the next crucial concept. Similarly to the standard wreath product, it is a particular form of a semidirect product $K \rtimes H$, where the normal part K factorises into a direct product of groups. Unlike the standard wreath product, these groups are not necessarily isomorphic. The outer group H acts on the factors of K, permuting the factors. If all the factors of K represent the same group G, then the inhomogeneous wreath product is just the standard wreath product $G \wr H$. Lemmas 1.7, 1.8, 1.9 are of particular importance, since they allow us to reduce a potentially unbounded number of group operations to a few operations used in the main Theorems 5.28 and 5.31. It transpires that by employing the concept of the inhomogeneous wreath product we are able to express the main result in a comprehensive way (compare to [8, Pages 69-70]).

The third fundamental tool is of geometric nature. The description of the wreath products that appear in Theorem 5.31 relies on determining the *isomorphism classes of point orbits* in the actions of spherical groups. This part, omitted in the Babai's work, is done in Section 5.4.

All the above ideas are combined in Section 5.6, where the main results of the chapter are proved. First, in Theorem 5.28 , we present an inductive characterization of the vertex-stabilizers of automorphism groups of planar graphs. The basic groups are cyclic, dihedral and symmetric groups. The set of abstract groups that appear as the stabilizers of vertices of planar graphs is characterized as the smallest set of groups containing the basic groups and closed with respect to five types of inhomogenous wreath products. Theorem 5.31 completes the characterization. The resulting set of groups is obtained from the set of vertex-stabilisers using a small number of well-defined inhomogeneous wreath products with the spherical groups. More precisely, each spherical group determines one inhomogeneous wreath product. It follows that our characterization describes the automorphism groups of planar graphs without referring to planarity explicitly, as a simple recursive process which builds them from the basic groups. This provides an efficient step-by-step algorithm for generating the automorphism groups of planar graphs.

5.2 Extended graphs

For the purposes of this chapter, we define graphs following Tutte [158]. A *graph* X is a 3-tuple (E, V, ι), where E is a finite set of edges, V is a finite set of vertices with $V \cap E = \emptyset$, and $\iota : E \to \binom{V}{2} \cup \binom{V}{1}$ is an incidence function. An edge $e \in E$ and a vertex $v \in V$ are *incident* if $v \in \iota(e)$. An edge $e \in E$ is *ordinary* if $|\iota(e)| = 2$, otherwise it is a *semiedge*[1] if $|\iota(e)| = 1$. Two edges $e, f \in E$ are called *parallel* if $\iota(e) = \iota(f)$. A graph is *simple* if it has no semiedges and no parallel edges. A semiedge e is called *single* if there is no edge parallel with e.

To avoid confusion when working with more graphs, we use $V(X)$ and $E(X)$ to denote the vertices and edges of X, respectively. For any $v \in V$, we define the *degree* of v, denoted by $\deg(v)$, to be the number of edges incident to v.

We denote the n-cycle, for $n \geq 3$, by C_n. A *dipole* D_n is a graph consisting of two vertices joined by $n \geq 2$ parallel edges. A *semistar* S_n is a one-vertex graph with $n \geq 1$ semiedges. The semistar S_1 is called *trivial.*

[1] Tutte calls these edges *loops*. In the context of this chapter the only difference between loop and semiedge is that a semiedge contributes to the degree of a vertex by 1.

A graph $X' = (E', V', \iota')$ is a *subgraph* of $X = (E, V, \iota)$ if $E' \subseteq E$, $V' \subseteq V$, $\iota(e) \subseteq V'$ for every $e \in E'$, and $\iota' = \iota_{\restriction E'}$. The subgraph with $E' = \emptyset$ and $V' = \emptyset$ will be called *null graph*. The null graph and the graph X are trivial subgraphs. If a subgraph is not trivial, it will be called *proper*. For every $v \in V$, we define the *degree of v in X'*, denoted by $\deg_{X'}(v)$, to be the number of edges of X' incident to v. It follows that the set of all subgraphs of X is partially ordered by the relation "to be subgraph". For a subset $U \subseteq V$ a *subgraph of X induced by U*, denoted $X[U]$, is the maximal subgraph of X with vertex set U.

Let $Y = (E_1, V_1, \iota_1)$ and $Z = (E_2, V_2, \iota_2)$ be two subgraphs of a graph $X = (E, V, \iota)$. The *union* $Y \cup Z$ is the subgraph $(E_1 \cup E_2, V_1 \cup V_2, \iota_{\restriction E_1 \cup E_2})$. The *intersection* is the subgraph $Y \cap Z = (E_1 \cap E_2, V_1 \cap V_2, \iota_{\restriction E_1 \cap E_2})$.

Let n be a non-negative integer. A pair (Y, Z) of proper subgraphs of a graph X is called an *n-separation* if $E(Y) \cap E(Z) = \emptyset$, $X = Y \cup Z$, $V(Y) \cap V(Z) = U$, $|U| = n$, and both Y and Z have at least n edges. A set U of vertices of X is an *n-cut*, $n \geq 0$, if there exists an n-separation (Y, Z) such that $V(Y) \cap V(Z) = U$. Alternatively, we say that Y is *separated* by U in the graph X, and we set $\partial Y = U$ to be the *boundary* of Y. The *interior* $\mathring{Y}$ of Y is the induced subgraph $\mathring{Y} = Y[V(Y) \setminus \partial Y] = X[V(Y) \setminus \partial Y] \subset Y$. Note that if Y is a dipole or a semistar, then $\mathring{Y}$ is the null graph. A graph X is *m-connected* if it has no n-cut, for every $n < k$.

The unique vertex of a 1-cut is called an *articulation*. A maximal 2-connected subgraph of a graph X is called a *block*. Given n-separation (Y, Z) of X *determines* an n-cut U uniquelly. On the other hand side, an n-cut may be determined by a lot of n-separations. A 1-separation (Y, Z) of X is *essential* if neither Y, nor Z is isomorphic to S_1. An articulation w is *essential*, if there exists an essential 1-separation determining $\{w\}$. A 2-separation (Y, Z) of X dermining a 2-cut $\{u, v\}$ is *essential* if there there exists a block $B \subseteq X$ with at least three vertices such that $(Y \cap B, Z \cap B)$ is a 2-separation of B and the degrees $\deg_B(u) \geq 3$, $\deg_B(v) \geq 3$. A 2-cut $U = \{u, v\}$ of X is *essential* if U is 2-cut determined by an essential 2-separation.

Observe that a 2-connected graph with at least 2 edges has no semiedges. Similarly, a 3-connected graph with at least four edges is simple, see [158, page 71]. There are exactly 6 graphs that are m-connected, for any $m \geq 1$. These are the dipoles D_2, D_3, the complete graphs K_1, K_2, K_3 and the semistar S_1, see [158, page 72].

Colored graphs with oriented edges. We consider graphs with colored edges. Moreover, some of the edges of the considered graphs may be endowed with an orientation. By choosing an orientation for an ordinary edge e incident to vertices u, v in a graph X we mean that we associate to e one of the pairs (u, v) or (v, u). Even if the orientation is choosen for some of the edges, the graph itself remains unoriented. The orientations and colors of edges will play role exclusively only when we consider automorphisms of X. The other properties of X remain untouched by an orientation and coloring of some edges of X.

Homomorphisms. Let $X = (E, V, \iota)$ and $X' = (E', V', \iota')$ be two graphs. A *homomorphism* $X \to X'$ is a mapping $\alpha : E \cup V \to E' \cup V'$ such that $\alpha(E) \subseteq E'$,

$\alpha(V) \subseteq V'$, and

$$\alpha(\iota(e)) = \iota'(\alpha(e)) \quad \text{for every } e \in E.$$

It is assumed that for colored graphs with oriented edges, a homomorphism preserves both the colors and orientations of edges. The latter means: if an edge e has orientation (u, v), then $\alpha(e)$ has orientation $(\alpha(u), \alpha(v))$. An *isomorphism* X and X' is a bijective homomorphism $X \to X'$. If X and X' are isomorphic, we write $X \cong X'$. An *automorphism* of X is an isomorphism of X to itself. We denote the *automorphism group* of X by $\mathrm{Aut}(X)$.

For subgraphs of X, we usually consider only isomorphisms preserving their boundaries. Let A, A' be subgraphs of X. An isomorphism $\alpha\colon A \to A'$ is called a *∂-isomorphism* if $\alpha(\partial A) = \partial\alpha(A)$. If such a ∂-isomorphism exists, we say that A is ∂-isomorphic to A', denoted $A \cong_\partial A'$. Observe that for every subgraph A and every automorphism α of X, the mapping $\alpha|_A$ is a ∂-isomorphism from A to $\alpha(A)$.

5.3 Reduction to 3-connected graphs

We develop a reduction procedure which applies to any input graph X and terminates in an irreducible graph that does not contain non-trivial semistars or dipoles, and is irreducible with respect to essential n-cuts for $n \leq 2$. Under certain conditions, we show that the automorphism group $\mathrm{Aut}(X)$ can be inductively reconstructed from the automorphism group of the terminal primitive graph. We stress that planarity of X is not assumed in this section.

5.3.1 Parts

In this subsection we introduce a fundamental concept of atom.

First we define several types of subgraphs of a connected graph X, which we will call *parts*. A *star part* is any maximal non-trivial semistar, and a *dipole part* is any maximal dipole. By $\mathcal{S}(X)$, $\mathcal{D}(X)$ we denote respectively the set of all star parts, dipole parts, of X.

Let w be an articulation. A subgraph $Y \subseteq X$ with at least two vertices is a *block part of X separated by* $\{w\}$ if it is an inclusion minimal subgraph satisfying the following property:

(B) there exists a subgraph $Z \subseteq X$ such that (Y, Z) is an essential 1-separation determining $\{w\}$.

Let U be a 2-cut. A subgraph $Y \subseteq X$ with at least three vertices is a *proper part separated by U* if it is an inclusion minimal subgraph satisfying the following property:

(P) there exists a subgraph $Z \subseteq X$ such that (Y, Z) is an essential 2-separation determining U.

Lemma 5.1. *An essential articulation in a connected graph with at least* 2 *vertices is a boundary of at least one block part.*

An essential 2*-cut in a connected graph with at least* 3 *vertices is a boundary of at least one proper part.*

Proof. Assume that the statement does not hold, and let (Y, Z) be an essential n-separation for $n \in \{1, 2\}$. If $n = 1$, then both Y, Z are semistars, and if $n = 2$ both Y, Z are dipoles. In both cases it means $|V(X)| = n$, a contradiction. $\square$

The next lemma summarise basic properties of block and of proper parts.

Lemma 5.2. *Let Y be a block part, or a proper part in X. Then both $\mathring{Y}$ and Y is connected, $Y[\partial Y]$ is edgeless and $X[V(\mathring{Y})] = \mathring{Y}$.*

Proof. Let $U = \partial Y$ be the n-cut determined by an essential n-separation (Y, Z) of X, $n \in \{1, 2\}$. By minimality the edges of $X[U]$ belong to Z, hence $Y[\partial Y]$ is edgeless.

Since Y is neither a semistar nor a dipole, there exists a vertex $w \in V(Y) \backslash U = V(\mathring{Y})$. Denote by $C_w \subseteq X[V(\mathring{Y})]$ the connectivity component of containig w. We claim $C_w \subseteq \mathring{Y}$. Suppose, to the contrary, that this is not the case. Then C_w has an edge $e \in Z$ such that $\iota(e) \subset V(C_w) \subset V(\mathring{Y})$. On the other hand side, a vertex incident to e belongs to ∂Y, a contradiction. Hence every connectivity componenent of $X[V(\mathring{Y})]$ is a subgraph of $\mathring{Y}$. Suppose $\mathring{Y}$ has another component $C \neq C_w$. We form a new essential n-separation (Y_1, Z_1) by setting $V(Y_1) = V(Y) \setminus V(C)$, $V(Z_1) = V(Z) \cup V(C)$, and $E(Y_1) = E(Y) \setminus E(Y[V(C) \cup U])$, $E(Z_1) = E(Y) \cup E(Y[V(C) \cup U])$. Since $Y_1 \subset Y$ is a proper subgraph we get a contradiction to the minimality of Y. Thus $X[V(\mathring{Y})] = \mathring{Y} = C_w$. In particular, $\mathring{Y}$ is connected. Since X is connected, there exists a path P joining v to U. Since U is a cut, $P \subseteq Y$. If $|U| = 1$, this proves that Y is connected. If $U = \{u, v\}$ is of size two, we may choose $w \in \mathring{Y}$ such that w belongs to a block B containing both u and v. Then there are two (internally disjoint) paths P and Q joining w to u and to v, respectively. Since $w \in V(\mathring{Y})$ and $\{u, v\}$ is a 2-separation both $P \subseteq Y$ and $Q \subseteq Y$. Thus Y is connected.

$\square$

We denote by $\mathcal{P}_B(X, w)$ the set of all block parts of X separated by articulation w. Similarly, we denote by $\mathcal{P}_P(X, U)$ the set of all proper parts of X separated by 2-cut U. Let $\mathcal{P}_B(X) = \cup_w \mathcal{P}_B(X, w)$, where w ranges through all articulations of X, and let $\mathcal{P}_P(X) = \cup_U \mathcal{P}_B(X, U)$ where U ranges through all 2-cuts of X. Finally, set $\mathcal{P}(X) = \mathcal{P}_B(X) \cup \mathcal{P}_P(X) \cup \mathcal{S}(X) \cup \mathcal{D}(X)$ to be the set of all *parts* of the graph X.

5.3.2 Atoms and primitive graphs

The inclusion-wise minimal elements of $\mathcal{P}(X)$ are called *atoms* and the set of all atoms of X will be denoted by $\mathcal{A}(X)$. We distinguish *star atoms*, *dipole atoms*, *block atoms*, and *proper atoms*, according to the type of the defining part. Observe that all star parts and dipole parts are already atoms. Figure 5.1 displays distinguished types of atoms. A connected graph containing no atoms is called *primitive*.

The graph $\overline{X}$ obtained from X by removing all the semiedges is called the *essence* of X. We say that X is *essentially* 3-*connected* if X contains no non-trivial semistars and $\overline{X}$ is 3-connected. Note that a 3-connected graph is either simple, or one of S_1, D_2, D_3. Similarly, X is *essectially a cycle* if X has no non-trivial semistars and $\overline{X}$ is a cycle. Let A be a proper atom with $\partial A = \{u, v\}$.

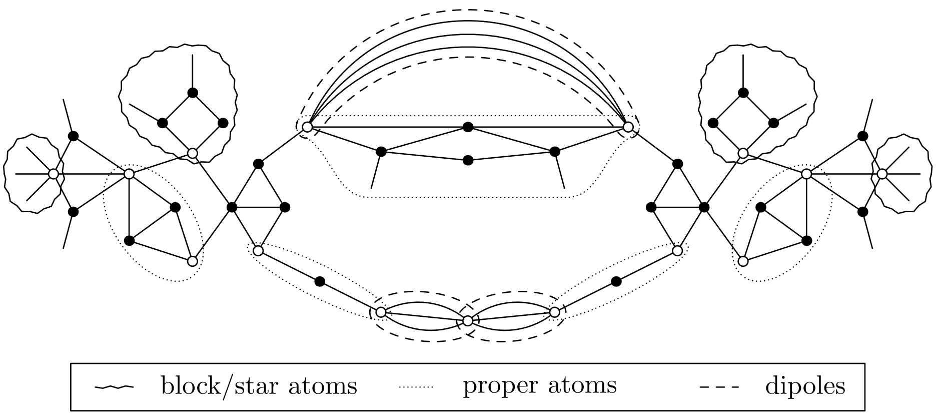


Figure 5.1: An example of a graph with denoted atoms. The white vertices belong to the boundary of some atom, possibly several of them.

By Lemma 5.2 the vertices u, v are non-adjacent. We define the *extended proper atom* by setting $A^+ = A + e$, where e is an ordinary edge joining u to v. Now, we investigate some properties of atoms.

Lemma 5.3. *If X is a primitive graph, or a block atom, or an extended proper atom, then the essence $\overline{X}$ of X is either 3-connected, or a cycle.*

Proof. By Lemma 5.1, $\bar{X}$ is connected. If $\overline{X}$ has an articulation, then either X contains a block part, or X is a non-trivial semistar. By Lemma 5.1 X has an atom, a contradiction. Hence, $\overline{X}$ is 2-connected. Let u be the vertex of minimum degree in $\overline{X}$. If $\deg(u) \leq 1$, then $\overline{X}$ is K_1, or K_2, which are 3-connected. If $\deg(u) = 2$, then u is an inner vertex of a maximal induced path P. Either the end-vertices of P are of degree ≥ 3, or they are both of degree two. In the first case, P is a proper atom, a contradiction. In the second case $\overline{X}$ is a cycle. Finally, if $\deg(u) \geq 3$, then every 2-cut is essential. If $\overline{X}$ contains a non-trivial 2-cut, then by Lemma 5.1 $\overline{X}$ contains a proper part and therefore also an atom, a contradiction. Thus, $\overline{X}$ has no 2-cut, and consequently, it is 3-connected. □

Corollary 5.4. *For any connected graph X the following statements hold true:*

(i) if X is primitive, then $\overline{X}$ is a cycle, or $\overline{X}$ is simple and 3-connected,

(ii) if X is a block atom in a connected graph, then $\overline{X}$ is a cycle, or $\overline{X}$ is simple and 3-connected with at least two vertices,

(iii) if X is an extended proper atom of a connected graph, then $\overline{X}$ is a cycle, or $\overline{X}$ is simple and 3-connected with at least four vertices.

Moreover, in each of the three cases either $X = \overline{X}$, or X arises from $\overline{X}$ by attaching single semiedges to some vertices of $\overline{X}$.

Proof. Assume the essence $\overline{X}$ is not a cycle. By Lemma 5.3 $\overline{X}$ is 3-connected. The only non-simple 3-connected graphs are D_2, D_3 and S_1. By definitions these cannot be subgraphs of the essence of considered graphs. Further, K_1 cannot be the essence of a block part, and neither K_1 nor K_2 can be the essence of a proper part. □

One may ask, whether a graph satisfying the necessary conditions from Collorary 5.4 is isomorphic to a primitive graph, block atom or to a proper atom. The answer is positive.

If X is a cycle, or X is simple 3-connected, then it contains no parts, and therefore is primitive.

Assume now that X is a cycle, or simple 3-connected with at least two vertices. We construct a new graph first by taking two copies X_1 and X_2 of X. Choose $v_i \in V(X_i)$, $i = 1, 2$, and form Y by identifying v_1 and v_2 in the disjoint union $X_1 \cup X_2$. Clearly $Y = A \cup B$, where $A \cong X_1$ and $B \cong X_2$ are block atoms of Y.

Cyclic extended proper atoms isomorphic C_n are obtained from path-like proper atoms with n vertices. They appear in a graph obtained from a simple 3-connected graph Y with at least 4 vertices by subdivision of an edge of Y with $n - 2$ vertices.

Let X be simple 3-connected with at least 4 vertices. Take to copies X_i, $i = 1, 2$ and form $X_i' = X_i - e_i$, where $e_i \in E(X_i)$. Let u_i, v_i be vertices incident e_i. Form Y by identifying u_1 with u_2 and v_1 with v_2, in the disjoint union $X_1' \cup X_2'$. Then $Y = A \cup B$, $A \cong X_i' \cong B$, $i = 1, 2$, are proper atoms whose extensions are isomorphic to X.

Corollary 5.5. *The essence of a proper atom in a graph X is either 2-connected, or a path where the internal vertices are of degree two in $\overline{X}$ and the end-vertices are of degree at least three in $\overline{X}$.*

Proof. By Corollary 5.4(iii) an extended proper atom is either a cycle of length ≥ 3, or a 3-connected graph with at least 4 vertices. Clearly, an edge deletion cannot decrease the connectivity by more than one. Hence A is 2-connected in the latter case. If A^+ is a cycle of length $k \geq 3$, then the essence $\bar{A}$ is a path of length $k - 1$. By definition of a proper atom each of the end-vertices is incident to at least 3 ordinary edges. □

The following important lemma states that atoms can overlap only in their boundaries. Among other it guaranties that the following reduction process is correctly determined.

Lemma 5.6. *Let A and B be two distinct atoms in a connected graph. Then $V(A \cap B) = \partial A \cap \partial B$, in particular $\mathring{A} \cap \mathring{B} = \emptyset$.*

Proof. Assume, to the contrary, that there exists a vertex $w \in V(\mathring{A}) \cap V(B)$. If $V(B) \subseteq V(A)$, then by minimality $B = A$, a contradiction. Hence, there exists a vertex z of B, $z \notin V(A)$. By Lemma 5.3, B is essentially 2-connected, or essentially a path. Since B is connected, there exists a path joining z to w intersecting ∂A. Hence $|B| \geq 3$, and consequently, either B is a block atom, or a proper atom.

Assume $\bar{B}$ is 2-connected. Then there is a cycle $C \subseteq B$ containing both w and z. It follows that $|\partial A| = 2$, and both vertices of ∂A are contained in C. It follows that ∂A is an essential 2-cut separating $A \cap B$. Hence $A \cap B \subseteq A$ contains a proper part, implying $A = B$, a contradiction.

Let $\bar{B}$ be a path. All the internal vertices of $\bar{B}$ are of degree 2 and the vertices in ∂A are of degree > 2. It follows $\mathring{B} \subseteq \mathring{A}$, and consequently, $B \subseteq A$, a contradiction to existence of $z \in V(B) \setminus V(A)$.

We proved $V(\mathring{A}) \cap V(B) = \emptyset$. By symmetry $V(\mathring{B}) \cap V(A) = \emptyset$, as well. Hence $V(A) \cap V(B) = \partial A \cap \partial B$, and we are done. □

Action of the automorphism group on atoms. We observe that atoms behave well with respect to the action of the automorphism group, making it possible to replace them by edges, while preserving some information about the automorphism group of the graph.

Lemma 5.7. *Let A be an atom and let $g \in \mathrm{Aut}(X)$. Then the following hold:*

- *We have $g(A) \cong A$, $g(\partial A) = \partial g(A)$, and $g(\mathring{A}) = g(\mathring{A})$.*
- *If $g(A) \neq A$, then $g(\mathring{A}) \cap \mathring{A} = \emptyset$ and $g(A) \cap A = \partial g(A) \cap \partial A$.*

Proof. The statement follows from Lemma 5.6 and from the fact that an automorphism takes an atom onto an atom. □

An atom A of X is *symmetric* if $Aut(A)$ is transitive on ∂A, otherwise A is *asymmetric*.

5.3.3 Reduction

For a graph X, the reduction produces a series of graphs $X = X_0, \ldots, X_r$. To construct the graph X_{i+1} from X_i, we first find the collection of all atoms $\mathcal{A} = \mathcal{A}(X_i)$ of X_i. To the isomorphism class of each $\mathcal{A}$, we assign one new color not yet used in the graphs $X_0, \ldots, X_i$, so that the coloring is injective.

Elementary reduction. An elementary reduction is performed as follows. Every atom A with $\partial A = \{u\}$ is replaced by a semiedge e_A of the assigned color incident to the unique vertex $u \in \partial A$ and every atom A with $\partial A = \{u, v\}$ is replaced by an ordinary edge e_A of the assigned color incident to both $u, v \in \partial A$. If A is symmetric, the edge e_A is unoriented. If A is asymmetric with $\partial A = \{u, v\}$, we choose an arbitrary orientation of e_A, say (u, v). By the assumption u and v belong to different orbits in the action of $\mathrm{Aut}(X_i)$. If B is an atom in the same orbit as A, the edge e_B gets orientation $(\psi(u), \psi(v))$, for $\psi \in \mathrm{Aut}(X_i)$.

According to Lemma 5.6, the replaced interiors of the atoms are pairwise disjoint, so the reduction is well-defined, and the number of edges decreases in

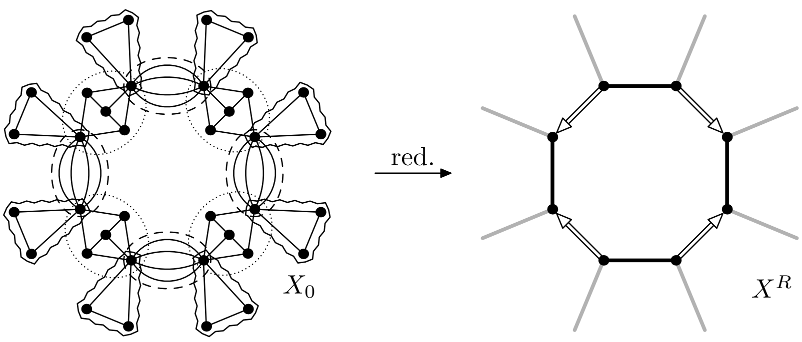


Figure 5.2: On the left, the graph X_0 has three isomorphism classes of atoms, one of each type. We reduce X_0 to X^R which is an eight cycle with single semiedges, with four black symmetric edges replacing the dipoles, four gray undirected edges replacing the block atoms, and four white directed edges replacing the proper atoms.

each step. By applying the elementary reductions repeatedly we obtain, after r steps, a primitive graph X_r. For a given graph X, we denote by $X^R = X_r$ the result of applying the reduction process to X. A final reduction step is demonstrated on Figure 5.2.

Expansions of edges. Consider the reductions series $X = X_0, \ldots, X_r$. For an edge $e \in E(X_i)$, for $i > 0$, let A_e be either the atom of X_{i-1} which reduces to e, or A_e is the subgraph induced by e if $e \in E(X_{i-1})$. The *expansion* A_e^* of $e \in E(X_i)$ is the unique subgraph of $X = X_0$ which reduces to e after i steps. Observe that for all $i = 1, \ldots, r$, we have $V(X_i) \subseteq V(X_{i-1})$ and $\iota_i(e) = \partial A_e = \partial A_e^*$, where ι_i is the incidence function of X_i.

Reduction epimorphism. We investigate how $\mathrm{Aut}(X_i)$ and $\mathrm{Aut}(X_{i+1})$ are related. We define a mapping $\varphi_i\colon \mathrm{Aut}(X_i) \to \mathrm{Aut}(X_{i+1})$ as follows. For $g \in \mathrm{Aut}(X_i)$, if $g(A_e) = A_f$, then we set $\varphi_i(g)(e) = f$, and for $u \in V(X_{i+1})$, we set $\varphi_i(g)(u) = g(u)$. Recall that $V(X_{i+1}) \subseteq V(X_i)$ is a union of orbits of $\mathrm{Aut}(X_i)$.

Lemma 5.8. *The mapping* $\varphi_i\colon \mathrm{Aut}(X_i) \to \mathrm{Aut}(X_{i+1})$ *is a group epimophism.*

Proof. For $g, h \in \mathrm{Aut}(X_i)$, we have

$$\varphi_i(gh)(e_A) = e_{gh(A)} = e_{g(h(A))} = \varphi_i(g)(e_{h(A)}) = \varphi_i(g)\varphi_i(h)(e_A).$$

Similarly,

$$\varphi_i(gh)(u) = gh(u) = g(h(u)) = \varphi_i(g)\varphi_i(h)(u).$$

Hence, φ_i is a group homomorphism.

We show that φ_i is surjective. For $g' \in \mathrm{Aut}(X_{i+1})$, we want to extend g' to $g \in \mathrm{Aut}(X_i)$ such that $\varphi_i(g) = g'$. If $u \in V(X_{i+1}) \subseteq V(X_i)$ we set $g(u) = g'(u)$. Let $e \in E(X_{i+1})$ be an edge. If $e \in E(X_i) \cap E(X_{i+1})$, we set $g(e) = g'(e)$. Suppose that $e = e_A$, for some atom A of X_i. If $f = g'(e)$, then f is an edge of the same color and the same type as e. Hence, f expands to an atom $A_f \cong A_e = A$. The automorphism g' prescribes the action of g on the boundary ∂A. We need to show that it is possible to extend the action of g' to $\mathring{A}$ consistently. The proof splits depending on symmetry type of A.

- *A is a star atom or a block atom.* Then edges e and $f = g'(e)$ are semiedges, incident to articulations u and $v = g'(u)$. We define $g_{\restriction \mathring{A}_e}$ to be a ∂-isomorphism $A_e \to A_f$.
- *A is an asymmetric proper atom or an asymmetric dipole.* By the definition, the orientations of e and $f = g'(e)$ is consistent with respect to g'. Since $\mathring{A}_e \cong \mathring{A}_f$, we define $g_{\restriction \mathring{A}_e}$ to be an orientation-preserving ∂-isomorphism $A_e \to A_f$.
- *A is a symmetric proper atom or a symmetric dipole.* Let $h\colon A_e \to A_f$ be a ∂-isomorphism. If h maps ∂A_e consistently with g', then we set $g_{\restriction \mathring{A}_e} = h_{\restriction \mathring{A}_e}$. Otherwise, we set $g_{\restriction \mathring{A}_e} = h_{\restriction \mathring{A}_e} \circ k$, where k is an automorphism of A_e swapping the two vertices in ∂A_e.

It follows that φ_i is an epimorphism. $\square$

By the “First isomorphism theorem”, $\mathrm{Aut}(X_i)$ is an extension of $\mathrm{Ker}(\varphi_i)$ by $\mathrm{Aut}(X_{i+1})$ and

$$\mathrm{Aut}(X_{i+1}) \cong \mathrm{Aut}(X_i)/\,\mathrm{Ker}(\varphi_i).$$

The following lemma gives a structural information on the kernel of φ_i.

Lemma 5.9. *The kernel* $\mathrm{Ker}(\varphi_i)$ *is isomorphic to the direct product*

$$\prod_{e\in E(X_{i+1})} \mathrm{Aut}(A_e)_{(\partial A_e)}.$$

Proof. For an atom $A \in \mathcal{A}(X_i)$, denote by K_A the point-wise stabilizer of $X \setminus \mathring{A}$ in $\mathrm{Ker}(\varphi_i) \leq \mathrm{Aut}(X_i)$. Clearly, $K_A \cong \mathrm{Aut}(A)_{(\partial A)}$. According to Lemma 5.6, the interiors of any two distinct atoms are pairwise disjoint. It follows that $K_A \cap \langle K_B : B \neq A, B \in \mathcal{A}(X_i)\rangle = \mathrm{id}$, and all K_A are normal subgroups. Hence,

$$\mathrm{Ker}(\varphi_i) = \prod_{A\in\mathcal{A}(X_i)} K_A \cong \prod_{A\in\mathcal{A}(X_i)} \mathrm{Aut}(A)_{(\partial A)} \cong \prod_{e\in E(X_{i+1})} \mathrm{Aut}(A_e)_{(\partial A_e)}.$$

□

We say that an atom A with $\partial A = \{u, v\}$ is *centrally symmetric* if there exists an automorphism $h \in \mathrm{Aut}(A)_{\partial A}$ such that $h(u) = v$, $h(v) = u$, $h^2 = \mathrm{id}$, and h centralizes $\mathrm{Aut}(A)_{(\partial A)}$. Note that all star and block atoms are centrally symmetric. Also, every symmetric dipole is centrally symmetric. Moreover, if each symmetric proper atom is centrally symmetric, then the following theorem hold.

Theorem 5.10. *Let $X_0, \dots, X_r$ be reduction series of a graph X. If every symmetric proper atom of X_i, is centrally symmetric, for $i < r$, then*

$$\mathrm{Aut}(X_i) \cong \left(\prod_{e\in E(X_{i+1})} \mathrm{Aut}(A_e)_{(\partial A_e)}\right) \rtimes \mathrm{Aut}(X_{i+1})$$

is the inhomogeneous wreath product defined by the action of the group $\mathrm{Aut}(X_{i+1})$ *on* $E(X_{i+1})$.

Proof. For simplicity, we denote $G_i = \mathrm{Aut}(X_i)$ and $K_i = \mathrm{Ker}(\varphi_i)$. To prove the theorem, we first find a subgroup H of G_i such that $G_i = K_i H$, $K_i \cap H = \{\mathrm{id}\}$, and $H \cong \mathrm{Aut}(X_{i+1})$.

The idea of the proof is as follows. We form a sequence of graphs $X_{i+1} = Y_0, \dots, Y_s = X_i$, where s is the number of all edge-orbits of G_{i+1} and Y_j is obtained from Y_{j-1} by expanding all the edges of one edge-orbit into atoms. First, we set $H_0 = G_{i+1}$. For $j > 0$, let $\mathcal{O} = [o]_{G_{i+1}}$ be the jth edge-orbit of G_{i+1}, where $o \in E(X_{i+1})$. For $g' \in G_{i+1}$, we find an extension $g \in \mathrm{Aut}(Y_j)$ such that $g(A_e) = A_f$ if and only if $g'(e) = f$ for all $e, f \in \mathcal{O}$, for $j = 0, 1, \dots, s$. The set $H_j = \{g : g' \in G_{i+1}\}$ is then a subgroup of $\mathrm{Aut}(Y_j)$ isomorphic to G_{i+1}. It is easy to see that $H_j \cap K_i = \{\mathrm{id}\}$, for $j = 0, 1, \dots, s$. Finally, we put $H = H_s$.

We describe the outlined construction of H_j, for every fixed $j > 0$. Assume that we already constructed the groups $H_0, \dots, H_{j-1}$. Let $\mathcal{O} = [o]_{G_{i+1}}$ be the jth edge-orbit of G_{i+1}, as above. To construct H_j, we distinguish several cases according to the type of A_o.

- *Case 1: The atom A_o is a star or a block atom.* n this case, all the edges in $\mathcal{O}$ are semiedges. For $e \in \mathcal{O}$, let u_e be the articulation such that $\partial A_e = \{u_e\}$. Choose arbitrarily ∂-isomorphisms $s_{o,e}$ from A_o to A_e, and put $s_{o,o} = \mathrm{id}$ and $s_{e,f} = s_{o,f} s_{o,e}^{-1}$. If $g'(e) = f$, we set $g{\restriction_{\hat{A_e}}} = s_{e,f}{\restriction_{\hat{A_e}}}$. Since
$$s_{e,c} = s_{f,c} s_{e,f}, \qquad \forall e, f, c \in \mathcal{O}, \tag{5.1}$$
the composition of the extensions g_1 and g_2 of two automorphisms g_1' and g_2' is defined on the interiors of all $\{A_e : e \in \mathcal{O}\}$ exactly as the extension of $g_2' g_1'$. Therefore the extensions form a group.

- *Case 2: The atom A_o is an asymmetric proper atom or dipole.* All the edges in the orbit $\mathcal{O}$ are oriented consistently with the action of G_{i+1}, and the end-vertices form two orbits. For $e \in \mathcal{O}$, let $\iota(e) = \{u_e, v_e\}$, where $\{u_e : e \in \mathcal{O}\}$ and $\{v_e : e \in \mathcal{O}\}$ are the two vertex-orbits. Further, the construction proceeds as in Case 1, in addition, since A_o is asymmetric, we have that $s_{o,e}(u_o) = u_e$ and $s_{o,e}(v_o) = v_e$.

- *Case 3: The atom A_o is a symmetric proper atom or a dipole.* All the edges in the orbit $\mathcal{O}$ are standard edges, and their end-vertices form one orbit. For $e \in \mathcal{O}$, let $\iota(e_i) = \{u_e, v_e\}$, where $\{u_e, v_e : e \in \mathcal{O}\}$ is the vertex-orbit. Further, for $e \in \mathcal{O}$ we arbitrarily choose one ∂-isomorphism $s_{o,e}$ from A_o to A_e such that $s_{o,e}(u_o) = u_e$ and $s_{o,e}(v_o) = v_e$, and set $s_{e,f} = s_{o,f} s_{o,e}^{-1}$, for $e, f \in \mathcal{O}$.

  By the assumptions, there is a central involution $t_o \in \mathrm{Aut}(A_o)_{\partial A_o}$ which exchanges u_o and v_o. Then t_o determines a central involution $t_e \in \mathrm{Aut}(A_e)_{\partial A_e}$ by conjugation $t_e = s_{o,e} t_o s_{o,e}^{-1}$. It follows that
$$t_f = s_{e,f} t_e s_{e,f}^{-1}, \qquad \text{and consequently} \qquad s_{e,f} t_e = t_f s_{e,f}, \qquad \forall e, f \in \mathcal{O}. \tag{5.2}$$
  We put $\hat{s}_{e,f} = s_{e,f} t_e = t_f s_{e,f}$ which is an isomorphism mapping A_e to A_f such that $\hat{s}_{e,f}(u_e) = v_f$ and $\hat{s}_{e,f}(v_e) = u_f$. In the extension, we put $g{\restriction_{\hat{A_e}}} = s_{e,f}{\restriction_{\hat{A_e}}}$ if $g'(u_e) = u_f$, and $g{\restriction_{\hat{A_e}}} = \hat{s}_{e,f}{\restriction_{\hat{A_e}}}$ if $g'(u_e) = v_f$.

  Besides (5.1), we get the following additional identities:
$$\hat{s}_{e,c} = s_{f,c} \hat{s}_{e,f}, \qquad \hat{s}_{e,c} = \hat{s}_{f,c} s_{e,f}, \quad \text{and} \quad s_{e,c} = \hat{s}_{f,c} \hat{s}_{e,f}, \qquad \forall e, f, c \in \mathcal{O}. \tag{5.3}$$

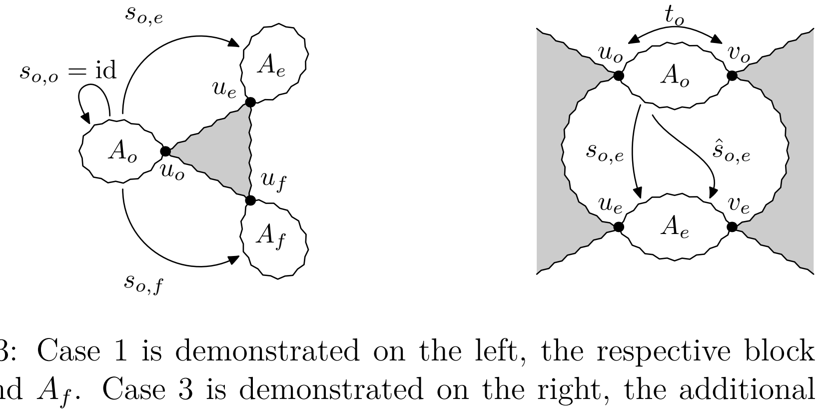


Figure 5.3: Case 1 is demonstrated on the left, the respective block atoms are A_o, A_e and A_f. Case 3 is demonstrated on the right, the additional involution $t_e \in \mathrm{Aut}(A_e)_{\partial A_e}$ transposes u_e and v_e.

Indeed, we have:

$$s_{f,c}\hat{s}_{e,f} = s_{f,c}s_{e,f}t_e = s_{e,c}t_e = \hat{s}_{e,c},$$

$$\hat{s}_{f,c}s_{e,f} = s_{f,c}t_f s_{e,f} = s_{f,c}s_{e,f}t_e = s_{e,c}t_e = \hat{s}_{e,c},$$

$$\hat{s}_{f,c}\hat{s}_{e,f} = t_c(s_{f,c}s_{e,f})t_e = t_c s_{e,c}t_e = t_c t_c s_{e,c} = s_{e,c}.$$

It follows that the composition of the extensions g_2g_1 is correctly defined, and it coincides with the extension of $g_2'g_1'$.

From the construction of H it follows that $H \cong \mathrm{Aut}(X_{i+1})$ and $H \cap K_i = \{\mathrm{id}\}$. Therefore, $G_i = K_iH$ is a semidirect product. Note that the complement H of $\mathrm{Ker}(\varphi_i)$, in the statement, is not uniquely determined. The group H depends on the choice of the isomorphism $s_{o,e}$ and of the involution t_o.

By Lemma 5.9 we have that K_i is the internal direct product

$$K_i = \prod_{e \in E(X_{i+1})} K_{A_e}.$$

Let $\theta\colon H \to G_{i+1}$ be the natural isomorphism $\varphi_i \restriction_H$ induced by the epimorphism $\varphi_i\colon G_i \to G_{i+1}$. By Theorem 1.5, in order to finish the proof we need to verify that the action of H by conjugation on K_i in the semidirect product K_iH satisfies the following: For every $e \in E(X_{i+1})$, $h, h_1, h_2 \in H$,

(i) $(K_{A_e})^h = K_{A_{\theta(h)(e)}}$,

(ii) for every $g \in K_{A_e}$ we have $g^{h_1} = g^{h_2}$ if and only if $\theta(h_1)(e) = \theta(h_2)(e)$.

Item (i) is trivially satisfied, since each h is an extension of an automorphism of X_{i+1}. For $g \in K_{A_e}$ denote by g_e the restriction of g onto A_e. For $h \in H$ taking $A_e \to A_f$, denote h_e the restriction of h to A_e. Let $s_{e,f}$ be the ∂-isomorphism mapping $A_e \to A_f$, chosen in the construction of H. Then $\theta(h)(e) = f$. Let $u_e \in \iota(e)$. If $s_{e,f}(u_e) = h(u_e)$, then

$$hgh^{-1}{\restriction_{A_f}} = h_e g_e h_f^{-1} = s_{e,f} g_e s_{e,f}^{-1}.$$

If $s_{e,f}(u_e) \neq h(u_e)$, then

$$hgh^{-1}{\restriction_{A_f}} = t_f s_{e,f} g_e s_{e,f}^{-1} t_f = s_{e,f} t_e g_e t_e s_{e,f}^{-1} = s_{e,f} g_e s_{e,f}^{-1}.$$

The second equality follows from (5.2) and the last equality holds since $t_e \in \mathrm{Aut}(A_e)_{\partial A_e}$ is a central involution. In each case the result does not depend on the choice of $h \in \{k \in H : \theta(k)(e) = f\}$.

□

5.3.4 Recursive construction of automorphism groups

The reduction can be used to describe inductively the automorphism groups of graphs in terms of the automorphism groups of their 3-connected components. Let $\theta_i = \varphi_{i-1} \circ \cdots \circ \varphi_0$ denote the epimorphism $\mathrm{Aut}(X_0) \to \mathrm{Aut}(X_i)$, for $i = 1, \dots, r$.

Lemma 5.11. *The kernel* $\mathrm{Ker}(\theta_i)$ *is isomorphic to the direct product*

$$\prod_{e\in E(X_i)} \mathrm{Aut}(A_e^*)_{(\partial A_e^*)}.$$

Proof. The statement follows by repeatedly applying Lemma 5.9. □

Theorem 5.12. *Let* $X_0,\ldots,X_r$ *be the reduction series of a graph* X*. If every symmetric proper atom of* $X_0,\ldots,X_i$ *is centrally symmetric, for* $i<r$*, then*

$$\mathrm{Aut}(X) \cong \left(\prod_{e\in E(X_{i+1})} \mathrm{Aut}(A_e^*)_{(\partial A_e^*)}\right) \rtimes \mathrm{Aut}(X_{i+1})$$

is the inhomogeneous wreath product defined by the action of the group $\mathrm{Aut}(X_{i+1})$ *on* $E(X_{i+1})$*.*

Proof. By Theorem 5.10 the statement holds for $i=0$. If $i>0$, we shall use Theorem 5.10 repeatedly, to conclude that $\mathrm{Aut}(X)$ contains an isomorphic copy of $\mathrm{Aut}(X_{i+1})$ forming a complement of $\mathrm{Ker}(\theta_{i+1})$. □

Theorem 5.13. *Let* $X_0,\ldots,X_r$ *be the reduction series of a graph* X*. If every symmetric proper atom of* $X_0,\ldots,X_i$ *is centrally symmetric, for* $i<r$*, and* $f\in E(X_{i+1})$*, then*

$$\mathrm{Aut}(A_f^*)_{(\partial A_f^*)} \cong \left(\prod_{e\in E(A_f)} \mathrm{Aut}(A_e^*)_{(\partial A_e^*)}\right) \rtimes \mathrm{Aut}(A_f)_{(\partial A_f)}$$

is the inhomogeneous wreath product defined by the action of $\mathrm{Aut}(A_f)_{(\partial A_f)}$ *on* $E(A_f)$*.*

Proof. Observe that $\mathrm{Aut}(A_f^*)_{(\partial A_f^*)}$ is embedded in $\mathrm{Aut}(X)$, and $\mathrm{Aut}(A_f)_{(\partial A_f)}$ is embedded in $\mathrm{Aut}(X_i)$. Let $\{u,v\}=\partial A=\partial A^*$, where u is not necessarily distinct from v. Now the statement follows from Theorem 5.12 by setting $X=A_f^*$ and $X_{i+1}=A_f$, with the two vertices u and v colored by different colors. The coloring implies $\mathrm{Aut}(X_{i+1})=\mathrm{Aut}(A_f)_{(\partial A_f)}$ and $\mathrm{Aut}(X)=\mathrm{Aut}(A_f^*)_{(\partial A_f^*)}$. With this identification in mind, the statement follows.

□

5.4 Point-orbits of spherical groups

In order to apply the results proved in Section 5.3 to a particular family $\mathcal{F}$ of graphs one needs to analyze the structure of automorphism groups of 3-connected graphs in $\mathcal{F}$. Since we aim to derive a recursive characterisation of automorphism groups of planar graphs, where the operations will be described in terms of particular inhomogoneous wreath products, one needs to understand the automorphism groups of 3-connected planar graphs not just as abstract groups, but we need to describe the actions as well. The analysis cannot be done in a purely group theoretical settings, although the final result is geometry free. A bridge between 3-connected planar graphs and geometry is established by the well known theorems by Whitney [163], Steinitz [150] and Mani [122] stated in the next section.

In this section the main aim is to determine isomorphism classes of G-orbits for each spherical group G.

A finite subgroup of the orthogonal group $O(3,\mathbb{R})$ of 3×3 orthogonal matrices is called a *spherical group*. Classification of spherical groups is well known. Abstractly, these groups include infinite families $\mathbb{Z}_n$, $\mathbb{D}_n$, $\mathbb{Z}_n \times \mathbb{Z}_2$, and $\mathbb{D}_n \times \mathbb{Z}_2$, the symmetry groups of the platonic solids, namely, $\mathbb{A}_5 \times \mathbb{Z}_2$, $\mathbb{S}_4 \times \mathbb{Z}_2$, and $\mathbb{S}_4$, and all the subgroups of these groups. Recall that every 3×3 orthogonal matrix is uniquely determined by its action on the sphere $S^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$. Hence, there is a one-to-one correspondence between the elements of $O(3)$ and the *isometries* (motions) of S^2. Up to isomorphism, there are 14 possible types of spherical groups identified by the Conway's notation [43]; see the left collumn of Table 5.1.

There are three basic types of isometries of S^2, namely rotations, reflections, and the antipodal involution, and every isometry is a composition of these. A *rotation* is an orientation-preserving isometry which fixes a pair of antipodal points, a *reflection* is an orienation-reversing isometry fixing a great circle, and the *antipodal involution* is an orientation-reversing isomotery swapping the pairs of antipodal points.

Point-orbits. Let g be an isometry of S with a fixed point. If g is orientation-preserving, then g is a *rotation*, if g is orientation-reversing, then g is a *reflection*. It follows that a point-stabilizer in a spherical group is either cyclic or a dihedral group.

For a spherical group G and a point x of the sphere, there are four types orbits $O = [x]_G$ according to the structure of the point-stabilizers:

- *regular*: $|O| = |G|$, the stabilizer of each point O is trivial,
- *regular-reflexive*: $|O| = |G|/2$ and the stabilizer of each point in O is generated by a unique reflection,
- *singular*: $|O| = |G|/m$, $m \geq 2$, and the stabilizer of each point in O is generated by a rotation of order m,
- *singular-reflexive*: $|O| = |G|/2m$, $m \geq 2$, and the stabilizer of each point in O is generated by a rotation and reflection generating the dihedral group of order $2m$.

With each spherical group G there is an associated *quotient orbifold* $\mathcal{O} = S/G$ whose points represent orbits of the action. In what follows, we denote by $\omega\colon S \to \mathcal{O}$ the natural projection $x \mapsto [x]_G$. Topologically, $\mathcal{O}$ is a either a sphere, or a disk, or the projective plane. The interior points of $\mathcal{O}$ correspond to regular orbits and to singular orbits. The boundary point of $\mathcal{O}$ (if the quotient orbifold is topologically a disk) represents a regular-reflexive, or a singular-reflexive orbit. By Riemann-Hurwitz equation, there can be only a finite number of singular and singular-reflexive orbits giving rise to finitely many *branch points* on $\mathcal{O}$. Moreover, the number of branch points is bounded by three, and all possible sequences of branch indices can be read from Table 5.1. A branch-point x is of index $m > 1$, if $\omega^{-1}(x)$ is a singular orbit of size $|G|/m$, or if $\omega^{-1}(x)$ is a reflexive singular orbit of size $|G|/2m$. The set of all branch points will be denoted by $\mathcal{B}$.

Regular orbits. For regular orbits, Lemma 1.3 implies the following.

Lemma 5.14. *All regular point-orbits in an action of G on S^2 are isomorphic.*

Regular-reflexive orbits. If the set of regular-reflexive orbits is non-empty, then $\mathcal{O}$ is topologically a disk and all the regular-reflexive orbits are projected on the boundary. From the classification of spherical groups, see Table 5.1 Collumn "Action", it follows that $|\mathcal{B} \cap \partial\mathcal{O}| \leq 3$ (for each action the number of singular reflexive points is equal to the number of integers following the star).

Let O_1 and O_2 be two regular-reflexive orbits which are projected by ω to $\bar{x}$ and $\bar{y}$ in $\partial\mathcal{O}$, respectively. There are several cases.

- *Case 1*: The points $\bar{x}$ and $\bar{y}$ are in the same connected component of $\partial\mathcal{O}\backslash\mathcal{B}$.
- *Case 2*: The points $\bar{x}$ and $\bar{y}$ are in different connected components of $\partial\mathcal{O}\backslash\mathcal{B}$ incident to a branch point of an odd index.
- *Case 3*: The points $\bar{x}$ and $\bar{y}$ are in different connected components of $\partial\mathcal{O}\backslash\mathcal{B}$ and one of $\bar{x}$, $\bar{y}$, is in a component separated by branch points of an even index.

Lemma 5.15. *In Cases 1 and 2, the regular-reflexive orbits $O_1 = \omega^{-1}(\bar{x})$ and $O_2 = \omega^{-1}(\bar{y})$ are isomorphic G-orbits.*

Proof. In Case 1, there exists a path $Q \subseteq \partial\mathcal{O}$ joining $\bar{x}$ and $\bar{y}$ and avoiding the branch points (if there are any). Then the $|G|/2$ lifts of Q give a matching of O_1 and O_2, determining the isomorphism.

Consider the induced action of G on the connected components of $\omega^{-1}(\partial\mathcal{O}\backslash\mathcal{B})$, which we will call segments. For $x \in S^2$, let s_x be the segment containing the point x. Then two orbits $[x]_G$ and $[y]_G$ are G-isomorphic if and only if $[s_x]_G$ and $[s_y]_G$ are G-isomorphic. This observation reduces the identification of the isomorphism classes to the problem of determining isomorphism classes of orbits of the action of G on the segments. By Lemma 1.2, $[s_x]_G$ and $[s_y]_G$ are isomorphic if and only if G_{s_x} and G_{s_y} are conjugate in G. We may assume that s_x and s_y are incident to $b \in \omega^{-1}(\bar{b})$, for some $\bar{b} \in \mathcal{B}$. Since the action of G is transitive on $\omega^{-1}(\bar{b})$, the groups G_{s_x} and G_{s_y} are conjugate in G if they are conjugate in $G_b \cong \mathbb{D}_m$, where m is the branch-index of $\bar{b}$. Using Lemma 1.4, this concludes the proof for the Case 2. □

Lemma 5.16. *The regular-reflexive orbits $O_1 = \omega^{-1}(\bar{x})$ and $O_2 = \omega^{-1}(\bar{y})$ are non-isomorphic in the Case 3.*

Proof. Let b_1 and b_2 be the branch points of even index separating the segment containing $\bar{x}$. By the classification of spherical groups, G is one of the groups *432, *22n, and *nn (n is even).

First, we deal with the case *nn. The boundary of $\mathcal{O}$ lifts into an embedding D_{2n} of a $2n$-dipole on the sphere, where the two branch points of index n lift to the two $2n$-valent vertices of the dipole, respectively. Without a loss of generality, we can assume that the vertices of the dipole correspond to the north and south pole of the sphere. The embedding determines a cyclic ordering $e_1, e_2, \ldots, e_{2n}$

of the edges of D_{2n}. We may assume that in this ordering, every odd edge e_i contains the unique preimage x_i of $\bar{x}$ and every even edge e_j contains the unique preimage y_j of $\bar{y}$. Since $\mathbb{D}_n$ acts as a group isometries fixing the north- and south-pole of the sphere, the great circle forming the equator C is preserved. The action gives two non-isomorphic point-orbits $C \cap \{e_{2i-1}; i = 1, 2, \ldots, n\}$ and $C \cap \{e_{2i}, i = 1, 2, \ldots, n\}$, see Lemma 1.4.

Now we deal with the case *22n (n is even). Similarly as above, $\mathcal{O}$ lifts to a triangulation of the sphere determined by a $2n$-sided bi-pyramid with the two $2n$-valent vertices located at the north and south pole. The embedding determines a natural cyclic ordering $e_1, e_2, \ldots, e_{2n}$ of the edges emanating from the vertex in the north pole and the cyclic ordering $e'_1, e'_2, \ldots, e'_{2n}$ at the south pole, where e_i meets with e'_i at the equator. Similarly, as in the previous paragraph, every odd edge consists of a unique preimage of $\bar{x}$ and every even edge consists of unique a preimage of $\bar{y}$. We may assume that in this ordering, the odd edges e_i and e'_i contain the unique preimages x_i and x'_i of $\bar{x}$, respectively, and the even edges e_j and e'_j contain the preimages y_j and y'_j of $\bar{y}$. The group $G \cong \mathbb{D}_n \times \mathbb{C}_2$ contains a normal subgroup $\mathbb{D}_n$ acting on the $2n$ vertices on the equator splitting this set on two non-isomorphic orbits as in in the previous case. The complementary central involution swaps the north- and south-pole, as well as x_i and x'_i, and y_j and y'_j, but fixes the equator point-wise.

Case *432. The lift of the respective orbifold gives a barycentric subdivision of the cube. Choose a triangle in the barycentric subdivision. The stabilizer of one side in the automorphism group of cube is generated by an involution. Thus the three sides give three involutions a, b, c, where $G = \langle a, b, c \rangle \leq \mathbb{S}_8$ is the automorphism group of the cube. Let a be the stabilzer of the side joining the center of edge to the center of a face of the cube. Then a is a product of four disjoint transpositions, while both b and c are products of two disjoint transpositions. Recall that the stabilizer of a vertex in a group of automorphism of a polyhedron is faithfull on the adjacent vertices. The involutions b and c generate the stabilizer of a vertex isomorphic to $\mathbb{D}_3$, hence they are conjugate. The involution a is not a conjugate of b, since a and b have different cycle structure. By Lemma 1.2 we are done. □

Singular and singular-reflexive orbits. The isomorphism of singular and singular-reflexive orbits will be discussed for each spherical group separately. Since the stabilizers in isomorphic orbits have the same size, it is sufficient to investigate the following spherical groups: *332, *22n, 332, 22n, *nn, and nn.

Lemma 5.17. *Let G be a spherical group of type *332 or 332. Then the two orbits of size* 4 *are isomorphic.*

Proof. The two orbits of size 4 are represented by the vertices and centers of faces of the spherical tetrahedron. Observe that antipodal point to a point representing a vertex of the tetrahedron is the center of the opposite face. This defines a matching between the vertices and the centers of faces of the tetrahedron, giving the isomorphism between the two singular orbits of size 4. □

Lemma 5.18. *Let G be a spherical group of type 22n or *22n. Then the following statements hold:*

- *If $n = 2$, then the three orbits of order 2 are pairwise non-isomorphic.*
- *If $n \geq 3$ and n is odd, then the two orbits of order n are isomorphic.*
- *If $n \geq 4$ and n is even, then the two orbits of order n are non-isomorphic.*

Proof. Let G be of type $22n$. Construct a geodesic triangle with vertices being the three branch points. Then the triangle lifts to the $2n$-sided by-pyramid, where the orbits O_1 and O_2 form the vertices of the cycle C of length $2n$ lying on the equator, thus the opposite points of C are antipodal. Now the statement follows from Lemma 1.4 if $n > 2$. If $n = 2$, then the by-piramid is the octahedron, where the three orbits of size two are formed by the three pairs of antipodal vertices. The stabilizers of these orbits correspond to the three non-trivial subgroups of $G \cong \mathbb{Z}_2^2$. Since $\mathbb{Z}_2^2$ is abelian, all conjugacy classes of subgroups are trivial, and the orbits are non-isomorphic by Lemma 1.2.

Let G be of type $*22n$. Then the orbifold $\mathcal{O}$ lifts to the $2n$-sided by-pyramid, where the orbits O_1 and O_2 form the vertices of the cycle C of length $2n$ lying on the equator, thus the opposite points of C are antipodal. Observe that G acts on C as the dihedral group $\mathbb{D}_n$. The statement follows from Lemma 1.4 if $n > 2$. If $n = 2$, then the by-piramid is the octahedron, where the three orbits of size two are formed by the three pairs of antipodal vertices. The stabilizers of these orbits correspond to three distinct subgroups of $G \cong \mathbb{Z}_2^3$. Since $\mathbb{Z}_2^3$ is abelian, all conjugacy classes of subgroups are trivial, and the orbits are non-isomorphic by Lemma 1.2. □

Lemma 5.19. *Let G be a spherical group of type *nn or of type nn. Then the two singular orbits of order 1 are isomorphic.*

Proof. Follows trivially from Lemma 1.3 □

We use the following presentations of the abstract groups isomorphic to the spherical groups:

- $*2km$: $\langle t_1, t_2, t_3 \mid t_1^2 = t_2^2 = t_3^2 = (t_1t_3)^k = (t_1t_2)^m = (t_2t_3)^2 = 1\rangle$, $k \leq m$,
- $2km$: $\langle r_1, r_2 \mid r_1^k = r_2^m = (r_1r_2)^2 = 1\rangle$, $k \leq m$,
- $2{*}n$: $\langle r, t \mid r^{2n} = t^2 = (rt)^2 = 1\rangle$,
- $3{*}2$: $\langle r, t, z \mid r^3 = t^2 = (rt)^3 = z^2 = [z, r] = [z, t] = 1\rangle$,
- $*nn$: $\langle t_1, t_2 \mid t_1^2 = t_2^2 = (t_1t_2)^n = 1\rangle$,
- nn: $\langle r \mid r^n = 1\rangle$,
- $n\times$, n even: $\langle r \mid r^n = 1\rangle$,
- $n*$: $\langle r, t \mid r^n = t^2 = [r, t] = 1\rangle$.

Using Lemmas 5.14–5.19 and an observation that in the action of an abelian group the conjugacy classes of subgroups are trivial we get the following theorem.

Action	G	$\|G\|$	Point-orbits	Stabilizers
*532	$\mathbb{A}_5 \times \mathbb{C}_2$	120	$120^\infty, 60^\infty, 30^1, 20^1, 12^1$	$1, \langle t_1\rangle, \langle t_2, t_3\rangle, \langle t_1, t_3\rangle, \langle t_1, t_2\rangle$
*432	$\mathbb{S}_4 \times \mathbb{C}_2$	48	$48^\infty, 2 \cdot 24^\infty, 12^1, 8^1, 6^1$	$1, \langle t_1\rangle, \langle t_2\rangle, \langle t_2, t_3\rangle, \langle t_1, t_3\rangle, \langle t_1, t_2\rangle$
*332	$\mathbb{S}_4$	24	$24^\infty, 12^\infty, 6^1, 4^2$	$1, \langle t_1\rangle, \langle t_2, t_3\rangle, \langle t_1, t_2\rangle$
*22n	$\mathbb{D}_n \times \mathbb{C}_2$, $n \geq 3$, odd	$4n$	$(4n)^\infty, 2 \cdot (2n)^\infty, n^2, 2^1$	$1, \langle t_1\rangle, \langle t_2\rangle, \langle t_2, t_3\rangle, \langle t_1, t_2\rangle$
*22n	$\mathbb{D}_n \times \mathbb{C}_2$, $n \geq 2$, even	$4n$	$(4n)^\infty, 3 \cdot (2n)^\infty, 2 \cdot n^1, 2^1$	$1, \langle t_1\rangle, \langle t_2\rangle, \langle t_3\rangle, \langle t_2, t_3\rangle, \langle t_1, t_3\rangle, \langle t_1, t_2\rangle$
532	$\mathbb{A}_5$	60	$60^\infty, 30^1, 20^1, 12^1$	$1, \langle r_1\rangle, \langle r_2\rangle, \langle r_1, r_2\rangle$
432	$\mathbb{S}_4$	24	$24^\infty, 12^1, 8^1, 6^1$	$1, \langle r_1\rangle, \langle r_2\rangle, \langle r_1, r_2\rangle$
332	$\mathbb{A}_4$	12	$12^\infty, 6^1, 4^2$	$1, \langle r_1\rangle, \langle r_2\rangle$
22n	$\mathbb{D}_n$, $n \geq 3$, odd	$2n$	$(2n)^\infty, n^2, 2^1$	$1, \langle r_2\rangle, \langle r_1, r_2\rangle$
22n	$\mathbb{D}_n$, $n \geq 2$, even	$2n$	$(2n)^\infty, 2 \cdot n^1, 2^1$	$1, \langle r_1\rangle, \langle r_2\rangle, \langle r_1, r_2\rangle$
3*2	$\mathbb{A}_4 \times \mathbb{C}_2$	24	$24^\infty, 12^\infty, 8^1, 6^1$	$1, \langle z\rangle, \langle r\rangle, \langle z, t\rangle$
2*n	$\mathbb{D}_{2n}$, $n \geq 2$	$4n$	$(4n)^\infty, (2n)^\infty, (2n)^1, 2^1$	$1, \langle r^{n/2}\rangle, \langle t\rangle, \langle r^2, t\rangle$
*nn	$\mathbb{D}_n$, $n \geq 3$, odd	$2n$	$(2n)^\infty, n^\infty, 1^2$	$1, \langle t_1\rangle, \langle t_1, t_2\rangle$
*nn	$\mathbb{D}_n$, $n \geq 2$, even	$2n$	$(2n)^\infty, 2 \cdot n^\infty, 1^2$	$1, \langle t_1\rangle, \langle t_2\rangle, \langle t_1, t_2\rangle$
nn	$\mathbb{C}_n$, $n \geq 2$	n	$n^\infty, 1^2$	$1, \langle r\rangle$
$n\times$	$\mathbb{C}_{2n}$, $n \geq 1$	$2n$	$(2n)^\infty, 2^1$	$1, \langle r^2\rangle$
n*	$\mathbb{C}_n \times \mathbb{C}_2$, $n \geq 2$	$2n$	$(2n)^\infty, n^\infty, 2^1$	$1, \langle t\rangle, \langle r, t\rangle$

Table 5.1: For each (parametrized) spherical group G, the first entry defines the action on the sphere following the Conway's [43] notation, the second entry determines G as an abstract group, the third entry gives the order of G. The fourth entry gives the numbers of isomorphism classes of point-orbits of G. By $c \cdot a^b$, we denote that there are c isomorphism classes of point-orbits of size a and each isomorphism class consists of b orbits. The fifth entry gives for each isomorphism class of orbits its representation in terms of a coset space of G; see presentations of the spherical groups.

Theorem 5.20. *The isomorphism classes of point-orbits of spherical groups are enumerated in the fourth column of Table 5.1. Moreover, given spherical group G, representatives of isomorphism classes of G-orbits are determined in terms of stabilizers in Column 5 of Table 5.1, where the stabilizers are described by means of the above presentations. Each orbit is isomorphic to the G-set formed by the set of left cosets of the stabilizer with G acting by left multiplication.*

Observe that the above description of the vertex-orbits does not require direct reference to spherical groups.

Constructions of polyhedra. In Figure 5.4, for each spherical group G the quotient orbifold $O = S^2/G$ is depicted together with a graph X embedded in O. Each embedding lifts along the projection $S^2 \to O$ onto the 2-skeleton of a (spherical) polyhedron P. The polyhedron P satisfies the following properties:

- G is a subgroup of $\mathrm{Aut}(P)$, and
- for every isomorphism class of point-orbits of G there exists a vertex-orbit in P representing the class.

In each quotient graph X embedded in O, see Figure 5.4, the vertices are labeled by subgroups of G. For each vertex v the associated subgroup determines (up to conjugacy in G) the stabilizers of its lifts. The polyhedrality of the lifted maps can be checked in each case directly. For instance, it is not difficult to

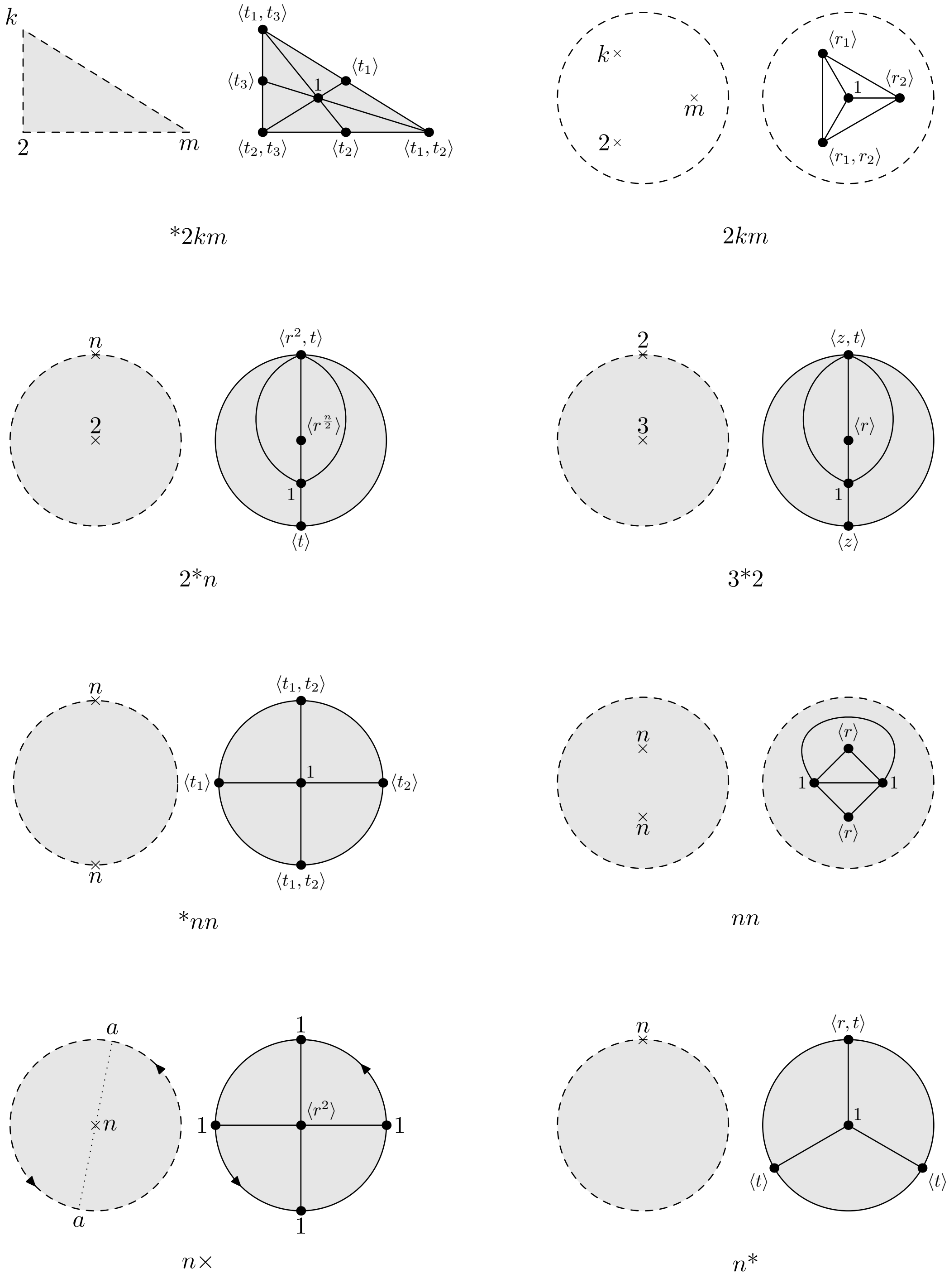


Figure 5.4: Quotient orbifolds and construction of polyhedra containing all types of orbits. For each spherical group, the left figure displays the respective quotient orbifold S^2/G. The associated right figure shows a map on S^2/G, whose lift to S^2 is a 2-skeleton of a polyhedron satisfying the required properties.

identify the lifted maps in case nn, see Figure 5.4. These are the 2-skeletons of $2n$-sided bipyramids. In case $*2km$ the lifted maps arise by a local operation applied to the barycentric subdivision of the 2-skeletons of the five Platonic solids and to cycles (if one of the k and m is equal 2).

5.5 Automorphism groups of 3-connected planar graphs

A graph is *planar* if it admits an embedding into the Euclidean plane. Similarly, a graph is *spherical* if and only if it admits an embedding into the sphere. A correspondence between a spherical and plane embeddings of a graph is via the stereographic projection. Thus a graph is planar if and only if it is spherical. A combinatorial characterization of planarity gives the well-known Kuratowski theorem. Both the definition of planarity and Kuratowski theorem easily apply to the graphs considered in this chapter, alternatively see [158, Chapter 11]. Recall that 3-connected graphs with at least 4 vertices are simple.

In this section, we recall the classification of automorphism groups of 3-connected planar graphs. The classification is based on Whitney's theorem [163] stating that a 3-connected planar graph has a unique embedding into the sphere, and on Mani's theorem [122] establishing that the unique embedding can be realized on the sphere such that all automorphisms of the graph extend to isometries of the underlying sphere.

Maps on surfaces. A *map* $\mathcal{M}$ is a 2-cell decomposition of a compact connected surface Σ. A map is usually defined by a 2-cell embedding of a connected graph $\varepsilon\colon X \hookrightarrow \Sigma$. The connectivity components of $\Sigma \setminus \varepsilon(X)$ are called faces of $\mathcal{M}$. An automorphism of a map is an automorphism of the graph preserving the incidences between the vertices, edges, and faces. If $\mathcal{M}$ is a spherical map, then $\mathrm{Aut}(\mathcal{M})$ is one of the spherical groups and with the exception of paths and cycles, it is a subgroup of $\mathrm{Aut}(X)$. As a consequence of Whitney's theorem [163] we have the following.

Theorem 5.21. *Let $\mathcal{M}$ be the map given by the unique 2-cell embedding of a 3-connected graph X into the sphere. Then $\mathrm{Aut}(X) = \mathrm{Aut}(\mathcal{M})$.*

By Mani's theorem [122], there exists a polyhedron P, whose 1-skeleton is isomorphic to X, such that the group of isometries of P coincides with $\mathrm{Aut}(X)$. Also, the polyhedron P can be placed in the interior of a sphere and projected onto it, so that each isometry of P corresponds to an isometry of the sphere. Therefore, every automorphism in $\mathrm{Aut}(X)$ can be viewed as an isometry of the sphere. In particular, we have the following corollary.

Theorem 5.22. *Let X be a 3-connected planar graph. Then $\mathrm{Aut}(X)$ is isomorphic to one of the spherical groups.*

Note that Mani proved his theorem 3-connected planar graphs that are polyhedral. These graphs are exactly 3-connected planar graphs with at least four vertices. Following the definitions, the 3-connected graphs with at most 3 vertices are the dipoles D_2, D_3, the complete graphs K_1, K_2, K_3 and the semistar S_1. All the six graphs are planar and their automorphism groups are spherical.

We recall some basic definitions from geometry [151, 133]. An automorphism of a 3-connected planar graph G is called *orientation preserving*, if the respective isometry preserves the global orientation of the sphere. It is called *orientation reversing* if it changes the global orientation of the sphere. A subgroup of $\mathrm{Aut}(X)$ is called *orientation preserving* if all its automorphisms are orientation preserving, and *orientation reversing* otherwise. Since the composition of orientations reversing automorphisms is orientations preserving, every orientation reversing subgroup contains an orientation preserving subgroup of index 2.

Vertex- and edge-stabilizers. Let X be a 3-connected planar graph and let $u \in V(X)$. Consider a spherical embedding of X, where the automorphisms of X are realized as isometries. The stabilizer of u in $\mathrm{Aut}(X)$ is a subgroup of a dihedral group and it has the following description in the language of isometries. If $\mathrm{Aut}(X)_u \cong \mathbb{Z}_n$, for $n \geq 3$, it is generated by a rotation of order n that fixes u and the antipodal point of the sphere, and fixing no other point of the sphere. The antipodal point of the sphere may be another vertex or a center of a face. If $\mathrm{Aut}(X)_u \cong \mathbb{D}_n$, for $n \geq 2$, it consists of rotations fixing u and the antipodal point of the sphere and reflections fixing a great circle passing through u and through the antipodal point. Each reflection always fixes either a center of some edge, or another vertex. When $\mathrm{Aut}(X)_u \cong \mathbb{D}_1 \cong \mathbb{Z}_2$, it is generated either by a 180° rotation, or by a reflection.

Let $e \in E(X)$. The stabilizer of e in $\mathrm{Aut}(X)$ is a subgroup of $\mathbb{Z}_2^2$. When $\mathrm{Aut}(X)_e \cong \mathbb{Z}_2^2$, it contains the following three non-trivial isometries. First, the 180° rotation around the center of e and the antipodal point of the sphere that is a vertex, center of an edge, or center of an even face. Next, two reflections perpendicular to each other which fix the center of e, the antipodal point of the sphere. When $\mathrm{Aut}(X)_e \cong \mathbb{Z}_2$, it is generated by only one of these three isometries.

We summarize the above discussion in the following lemma.

Lemma 5.23. *Let X be a 3-connected planar graph, let u be a vertex of degree n, and let e be an edge. Then $\mathrm{Aut}(X)_u$ is isomorphic to a subgroup of $\mathbb{D}_n$, and $\mathrm{Aut}(X)_e$ is isomorphic to a subgroup of $\mathbb{Z}_2^2$. Both $\mathrm{Aut}(X)_u$ and $\mathrm{Aut}(X)_e$ embedd into a spherical group.*

Stabilizers of atoms in planar graphs.

Lemma 5.24. *Let A be an atom in a planar graph X.*

(a) If A is a star atom, then $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)}$ and it is isomorphic to a direct product of symmetric groups.

(b) If A is a block atom, then $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)}$ and it is isomorphic to a subgroup of a dihedral group $\mathbb{D}_n$, where n is the degree of the articulation separating A.

(c) If A is a proper atom, then $\mathrm{Aut}(A)_{\partial A}$ and it is isomorphic to a subgroup of $\mathbb{Z}_2^2$ and $\mathrm{Aut}(A)_{(\partial A)}$ is a subgroup of $\mathbb{Z}_2$.

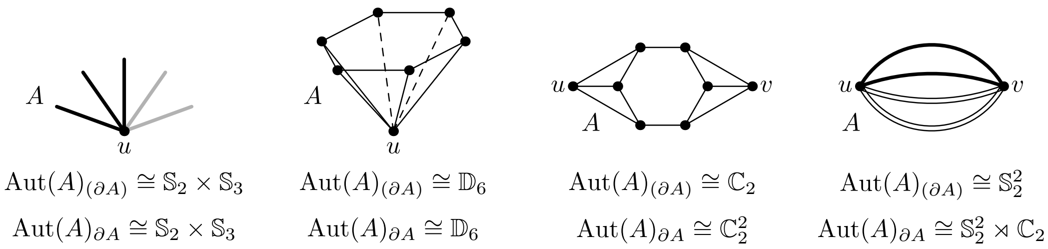


Figure 5.5: An atom A together with its groups $\mathrm{Aut}(A)_{(\partial A)}$ and $\mathrm{Aut}(A)_{\partial A}$. From left to right, a star atom, a block atom, a proper atom, and a dipole.

(d) *If A is a dipole, then* $\mathrm{Aut}(A)_{(\partial A)}$ *and it is isomorphic to a direct product of symmetric groups. If A is symmetric, then* $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)} \times \mathbb{Z}_2$. *If A is asymmetric, then* $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)}$.

Proof. (a) Since $|\partial A| = 1$, we have $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)}$. Clearly, the group $\mathrm{Aut}(A)_{(\partial A)}$ is a direct product of symmetric groups.

(b) Similarly as in Case (a), $|\partial A| = 1$, and we have $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)}$. Set $B = \overline{A}$. It follows that $\mathrm{Aut}(A)_{\partial A} \le \mathrm{Aut}(B)_{\partial B}$. By Corollary 5.4 (ii), either B is a cycle, or a 3-connected planar graph. In the first case, $\mathrm{Aut}(B)_{\partial B}$ is a subgroup of $\mathbb{Z}_2$, while in the second case, it is the stabilizer of a vertex in a 3-connected planar graph which is by Lemma 5.23 a subgroup of $\mathbb{D}_n$, where n is the degree of the articulation separating A.

(c) Let A be a proper atom with $\partial A = \{u, v\}$. Let $B = \overline{A}$, so $\mathrm{Aut}(A)_{\partial A} \le \mathrm{Aut}(B)_{\partial B}$ and $\mathrm{Aut}(A)_{(\partial A)} \le \mathrm{Aut}(B)_{(\partial B)}$. Clearly, $\mathrm{Aut}(B)_{\partial B} = \mathrm{Aut}(B^+)_{\partial B^+}$, where $B^+ = B + uv$. By Corollary 5.4, B^+ is either a cycle, or a 3-connected graph with at least four vertices. In the former case, $\mathrm{Aut}(B^+)$ is a subgroup of $\mathbb{Z}_2$ and $\mathrm{Aut}(B^+)_{(\partial B^+)}$ is trivial. In the latter case, we claim that B^+ is planar. First, by the definition of a proper atom, B is a subgraph of a block C in a planar graph X such that $\{u, v\}$ is a 2-separation of C. It follows that $X \setminus \mathring{B}$ is a connected plane graph, in particular, the boundary vertices u and v are connected by a path in $X \setminus \mathring{B}$. Thus, in the induced embedding of B into the sphere the vertices u and v appear in the boundary of the same face, i.e., B^+ is planar. By Lemma 5.23, $\mathrm{Aut}(B^+)_e$ of $e = uv$ is isomorphic to a subgroup of $\mathbb{Z}_2^2$. It follows that $\mathrm{Aut}(A)_{\partial A}$ is a subgroup of $\mathbb{Z}_2^2$ and $\mathrm{Aut}(A)_{(\partial A)}$ is a subgroup of $\mathbb{Z}_2$.

(d) Let $\{u, v\}$ be the vertex set of the dipole A. If A is asymmetric, we have $\mathrm{Aut}(A)_{\partial A} = \mathrm{Aut}(A)_{(\partial A)}$ which is a direct product of symmetric groups. If A is symmetric, there is an involution $t \in \mathrm{Aut}(A)_{\partial A}$ which swaps u with v and fixes all the edges. Clearly, we have $\langle t \rangle \cap \mathrm{Aut}(A)_{(\partial A)} = \{id\}$ and t centralizes $\mathrm{Aut}(A)_{(\partial A)}$. □

Lemma 5.25. *Every planar symmetric proper atom and symmetric dipole atom is centrally symmetric.*

Proof. Follows from the parts (c) and (d) of Lemma 5.24. □

Automorphism groups of planar primitive graphs.

Lemma 5.26. *The automorphism group* $\mathrm{Aut}(X)$ *of a planar primitive graph X is isomorphic to a spherical group.*

Proof. By Corollary 5.4 (i), the essence $\overline{X}$ of X is 3-connected, or an n-cycle. If $\overline{X}$ is 3-connected, then $\mathrm{Aut}(\overline{X})$ is a spherical group by Theorem 5.22. If $\overline{X}$ is an n-cycle, then $\mathrm{Aut}(\overline{X})$ is a subgroup of $\mathbb{D}_n$. By Corollary 5.4, $\mathrm{Aut}(X)$ is a subgroup of $\mathrm{Aut}(\overline{X})$. Since the family of spherical groups is closed under taking subgroups, we are done. □

5.6 Jordan-like characterization

In this section, we present and prove the main result of this chapter: a complete recursive description of the automorphism groups of connected planar graphs. Note that in combination with Theorem 1.10, this gives the description of the automorphism groups of all planar graphs. In the first step, we determine in Subsection 5.6.1 the abstract groups that can be realized as vertex-stabilizers of planar graphs. In the second step, we determine in Subsection 5.6.2 how these groups are composed with spherical groups.

5.6.1 Vertex-stabilizers of planar graphs

The aim of this section is to analyze the following class of abstract groups

$$\mathrm{Stab}(\mathsf{PLANAR}) = \{G \cong \mathrm{Aut}(X)_u : X \in \mathsf{PLANAR}, u \in V(X)\}$$

The following lemma relates $\mathrm{Stab}(\mathsf{PLANAR})$ to pointwise stabilizers of boundaries of expanded atoms, allowing us to apply the theory developed in Section 5.3.

Lemma 5.27. *The class* $\mathrm{Stab}(\mathsf{PLANAR}) = \mathrm{Fix}(\mathsf{PLANAR})$*, where*

$$\mathrm{Fix}(\mathsf{PLANAR}) = \{G \cong \mathrm{Aut}(X)_{(\partial X)} : X \in \mathsf{PLANAR}, |E(X^R)| = 1, \partial X = V(X^R)\}.$$

Proof. Let, $G \in \mathrm{Fix}(\mathsf{PLANAR})$ and let X be a graph certifying that. Let $v \in \partial X$. If $\partial X = \{v\}$, then $G \cong \mathrm{Aut}(X)_{(\partial X)} = \mathrm{Aut}(X)_v \in \mathrm{Stab}(\mathsf{PLANAR})$. If $\partial X = \{u, v\}$, then we form the graph Y by indentifying the vertex u with an end-vertex of a path P, where $|E(P)| > |E(X)|$. Then $G \cong \mathrm{Aut}(X)_{(\partial X)} \cong \mathrm{Aut}(Y)_v \in \mathrm{Stab}(\mathsf{PLANAR})$. Thus $\mathrm{Fix}(\mathsf{PLANAR}) \subseteq \mathrm{Stab}(\mathsf{PLANAR})$.

Let $G \in \mathrm{Stab}(\mathsf{PLANAR})$ and let X and $v \in V(X)$ be a graph and a vertex certifying that. Let Y be the graph formed by identifying v with an end-vertex of a path P, where $|E(P)| > |E(X)|$. Now, the graph Y^R is either a two-vertex graph, or K_1 with a single semiedge attached, depending on the parity of $|E(P)|$. The latter case can be alsways ensured by suitably adjusting the parity of $|E(P)|$. We have $|E(Y^R)| = 1$ and $G \cong \mathrm{Aut}(X)_u \cong \mathrm{Aut}(Y)_{(\partial Y)} \in \mathrm{Fix}(\mathsf{PLANAR})$. Thus $\mathrm{Stab}(\mathsf{PLANAR}) \subseteq \mathrm{Fix}(\mathsf{PLANAR})$. □

The next theorem gives a recursive characterization of $\mathrm{Stab}(\mathsf{PLANAR})$.

Theorem 5.28. *The class* $\mathrm{Stab}(\mathsf{PLANAR}) = \mathcal{F}$*, where* $\mathcal{F}$ *is defined inductively as follows:*

- *(a)* $\{1\} \in \mathcal{F}$.
- *(b) If* $G_1, G_2 \in \mathcal{F}$*, then* $G_1 \times G_2 \in \mathcal{F}$.

(c) If $G \in \mathcal{F}$, then $G \wr \mathbb{S}_n \in \mathcal{F}$.

(d) If $G \in \mathcal{F}$, then $G \wr \mathbb{Z}_n \in \mathcal{F}$.

(e) If $G_1, G_2 \in \mathcal{F}$, then $(G_1, G_2) \wr\wr \mathbb{D}_n \in \mathcal{F}$, for $n \geq 3$ odd, where $|\Omega_1| = 2n$ and $|\Omega_2| = n$.

(f) If $G_1, G_2, G_3 \in \mathcal{F}$, then $(G_1, G_2, G_3) \wr\wr \mathbb{D}_n \in \mathcal{F}$, for $n \geq 2$ even, where $|\Omega_1| = 2n$ and $|\Omega_2| = |\Omega_3| = n$, and $\mathbb{D}_n$ acts on Ω_1 regularly, and acts on Ω_2 and Ω_3 as on the vertices and edges of the regular n-gon, respectively.

We split the proof into the following two lemmas.

Lemma 5.29. *Each group in* Fix(PLANAR) *is isomorphic to a group in $\mathcal{F}$.*

Proof. Let $X = X_0, X_1, \ldots, X_r$ be the reduction series of a planar graph X. We proceed by an induction on $i = 1, \ldots, r$. If $i = 1$, then $A_e^* = A_e$, for any $e \in E(X_1)$, is an atom of $X = X_0$, or an edge of X_0. By Lemma 5.24, we have $\mathrm{Aut}(A_e)_{(\partial A_e)} \in \mathcal{F}$.

If $i > 1$, we fix an edge $f \in E(X_i)$ and set $A = A_f$. By Theorem 5.10,

$$\mathrm{Aut}(A^*)_{(\partial A^*)} \cong \left(\prod_{e \in E(A)} \mathrm{Aut}(A_e^*)_{(\partial A_e^*)} \right) \rtimes \mathrm{Aut}(A)_{(\partial A)},$$

where the right hand side is the inhomogeneous wreath product defined by the action of $\mathrm{Aut}(A)_{(\partial A)}$ on $E(A)$. Recall that if an edge $e \in E(A)$ does not expand into an atom, then $\mathrm{Aut}(A_e^*)_{(\partial A_e^*)}$ is trivial.

Let $\{\Omega_1, \ldots, \Omega_m\}$ be the partition of $E(A)$ into the orbits of the action of $\mathrm{Aut}(A)_{(\partial A)}$. By induction hypothesis, for every $j \in \{1, \ldots, m\}$ and for every $e \in \Omega_j$ and there is $K_j \in \mathcal{F}$ such that $K_j \cong \mathrm{Aut}(A_e^*)_{(\partial A_e^*)}$. Then

$$\mathrm{Aut}(A^*)_{(\partial A^*)} \cong G = \left(K_1^{\ell_1} \times \cdots \times K_m^{\ell_m} \right) \rtimes S, \tag{5.4}$$

where $\ell_i = |\Omega_i|$, and $S \cong \mathrm{Aut}(A)_{(\partial A)}$ acts on $E(A)$. We split the proof into several cases, depending on the type of A.

Case 1: A is a star atom or a dipole atom. By Lemma 5.24, $S = \mathbb{S}_{\ell_1} \times \cdots \times \mathbb{S}_{\ell_m}$, where $\mathbb{S}_{\ell_j}$ is isomorphic to the subgroup of $\mathrm{Aut}(A)_{(\partial A)}$ fixing every edge not in the orbit Ω_j. For every j, G contains a subgroup $G_j \cong K_j \wr \mathbb{S}_{\ell_j} \in \mathcal{F}$. By definition, the subgroups G_j and G_k, for $j \neq k$, are disjoint. Moreover, if $g \in G_j$ and $g' \in G_k$ then $gg' = g'g$. Therefore,

$$\mathrm{Aut}(A^*)_{(\partial A^*)} \cong G \cong K_1 \wr \mathbb{S}_{\ell_1} \times \cdots \times K_m \wr \mathbb{S}_{\ell_m}.$$

The group on the right side belongs to $\mathcal{F}$.

Case 2: A is a proper atom. By Lemma 5.24, either $\mathrm{Aut}(A)_{(\partial A)}$ is trivial, or $\mathrm{Aut}(A)_{(\partial A)} \cong \mathbb{Z}_2$. In the first case, there is nothing to prove since each $\mathrm{Aut}(A_e^*)_{(\partial A_e^*)}$, for $e \in E(A)$, is isomorphic to a group from $\mathcal{F}$ and so is their direct product.

In the second case, the induced action of $\mathbb{Z}_2$ on $E(A)$ has orbits $\{\Omega_1, \ldots, \Omega_m\}$ of size at most 2. Let $\Omega_1, \ldots, \Omega_t$, for some $0 \leq t \leq m$, be the orbits of size 1. By substituting to (5.4), we get

$$G = \left(K_1 \times \cdots \times K_t \times K_{t+1}^2 \times \cdots \times K_m^2 \right) \rtimes \mathbb{Z}_2,$$

where $K_i \in \mathcal{F}$, for $i = 1, \dots, m$, by induction. Denote $G_1 = K_1 \times \cdots \times K_t$ and $G_2 = K_{t+1} \times \cdots \times K_m$. By Lemma 1.8

$$G \cong G_1 \times (K_{t+1}, \dots, K_m) \wr\wr \mathbb{Z}_2.$$

By Lemma 1.9

$$G \cong G_1 \times G_2 \wr \mathbb{Z}_2,$$

which belongs to $\mathcal{F}$.

Case 3: A is a block atom. Since A is essentially 3-connected, $\mathrm{Aut}(A)_{(\partial A)}$ is isomorphic to a point-stabilizer of a spherical group. By Lemma 5.24, we have $\mathrm{Aut}(A)_{(\partial A)} \cong S$ is a subgroup of a dihedral group, in particular, either S cyclic of type nn or S is dihedral of type $*nn$. We distinguish three cases: $S \cong \mathbb{Z}_n$ with $n \geq 2$, $S \cong \mathbb{D}_n$ with $n \geq 3$ and n odd, $S \cong \mathbb{D}_n$ with $n \geq 2$ and n even.

Let $\mathrm{Aut}(A)_{(\partial A)} \cong \mathbb{Z}_n$. The induced action of the group $\mathbb{Z}_n$ on $E(A)$ has orbits $\{\Omega_1, \dots, \Omega_m\}$ of size 1 or n; see Table 5.1. Let $\Omega_1, \dots, \Omega_t$ be the orbits of size 1, for some $0 \leq t \leq 2$. By substituting to (5.4), we get

$$G = \left(K_1 \times \cdots \times K_t \times K_{t+1}^n \times \cdots \times K_m^n\right) \rtimes \mathbb{Z}_n,$$

where $K_i \in \mathcal{F}$, for $i = 1, \dots, m$, by induction. Denote $G_1 = K_1 \times \cdots \times K_t$ and $G_2 = K_{t+1} \times \cdots \times K_m$. By Lemma 1.8

$$G \cong G_1 \times (K_{t+1}, \dots, K_m) \wr\wr \mathbb{Z}_n.$$

Since the action of $\mathbb{Z}_n$ is regular on each Ω_j, for $j > t$, the actions of $\mathbb{Z}_n$ on Ω_j are all isomorphic. By Lemma 1.9

$$G \cong G_1 \times G_2 \wr \mathbb{Z}_n,$$

which belongs to $\mathcal{F}$.

Let $\mathrm{Aut}(A)_{(\partial A)} \cong \mathbb{D}_n$ with $n \geq 3$ and n odd. The induced action of $\mathbb{D}_n$ on $E(A)$ has orbits $\{\Omega_1, \dots, \Omega_m\}$ of possible sizes 1, n, and $2n$; see Table 5.1. There are integers $0 \leq t_1 \leq t_2 \leq m$, $t_1 \leq 2$, such that the orbits $\Omega_1, \dots, \Omega_{t_1}$ are of size 1, the orbits $\Omega_{t_1+1}, \dots, \Omega_{t_2}$ are of size n, and the orbits $\Omega_{t_2+1}, \dots, \Omega_m$ are of size $2n$. By substituting to (5.4), we get

$$G \cong \left(K_1 \times \cdots \times K_{t_1} \times K_{t_1+1}^n \times \cdots \times K_{t_2}^n \times K_{t_2+1}^{2n} \times \cdots \times K_m^{2n}\right) \rtimes \mathbb{D}_n,$$

where $K_i \in \mathcal{F}$, for $i = 1, \dots, m$, by induction. Denote $G_1 = K_1 \times \cdots \times K_{t_1}$, $G_2 = K_{t_1+1} \times \cdots \times K_{t_2}$, and $G_3 = K_{t_2+1} \times \cdots \times K_m$. By Lemma 1.8

$$G \cong G_1 \times (K_{t_1+1}, \dots, K_m) \wr\wr \mathbb{D}_n.$$

By Lemma 1.4, the actions of $\mathbb{D}_n$ on all Ω_j, for $t_1 < j \leq t_2$, are isomorphic. The actions of $\mathbb{D}_n$ on all Ω_j, for $j > t_2$, are all regular and therefore isomorphic. By Lemma 1.9

$$G \cong G_1 \times (G_2, G_3) \wr\wr \mathbb{D}_n,$$

which belongs to $\mathcal{F}$.

Let $\mathrm{Aut}(A)_{(\partial A)} \cong \mathbb{D}_n$ with $n \geq 2$ and n even. The induced action of $\mathbb{D}_n$ on $E(A)$ has orbits $\{\Omega_1, \dots, \Omega_m\}$ of possible sizes 1, n, and $2n$; see Table 5.1. By

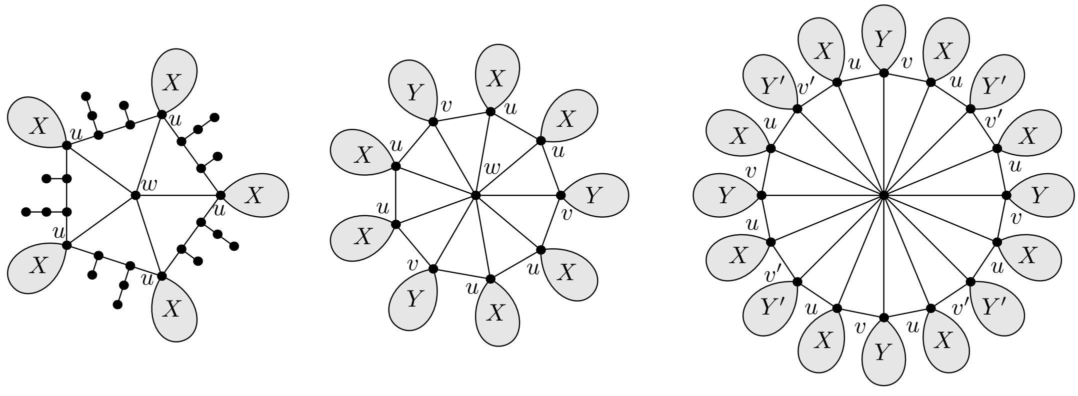


Figure 5.6: (a) $\mathrm{Aut}(Z) \cong \mathrm{Aut}(X) \wr \mathbb{Z}_5$, (b) $\mathrm{Aut}(Z) \cong (\mathrm{Aut}(X), \mathrm{Aut}(Y)) \wr\wr \mathbb{D}_3$, (c) $\mathrm{Aut}(Z) \cong (\mathrm{Aut}(X), \mathrm{Aut}(Y), \mathrm{Aut}(Y')) \wr\wr \mathbb{D}_4$

Lemma 1.4, there are two possible non-isomorphic actions of $\mathbb{D}_n$ on n points. Thus, there are integers $0 \le t_1 \le t_2 \le t_3 \le m$ such that the orbits $\Omega_1, \dots, \Omega_{t_1}$ are of size 1, the orbits $\Omega_{t_1+1}, \dots, \Omega_{t_2}$ are of size n, the orbits $\Omega_{t_2+1}, \dots, \Omega_{t_3}$ are of size n, and the orbits $\Omega_{t_3+1}, \dots, \Omega_m$ are of size $2n$, and the $\mathbb{D}_n$-sets Ω_i and Ω_j, for $t_1 < i \le t_2$ and $t_2 < j \le t_3$, are non-isomorphic. By substituting to (5.4), we get

$$G = \left(\prod_{i=1}^{t_1} K_i \times \prod_{i=t_1+1}^{t_2} K_i^n \times \prod_{i=t_2+1}^{t_3} K_i^n \times \prod_{i=t_3+1}^{m} K_i^{2n} \right) \rtimes \mathbb{D}_n,$$

where $K_i \in \mathcal{F}$, for $i = 1, \dots, m$, by induction. Denote $G_1 = K_1 \times \cdots \times K_{t_1}$, $G_2 = K_{t_1+1} \times \cdots \times K_{t_2}$, $G_3 = K_{t_2+1} \times \cdots \times K_{t_3}$, and $G_4 = K_{t_3+1} \times \cdots \times K_m$. By Lemma 1.8

$$G \cong G_1 \times (K_{t_1+1}, \dots, K_m) \wr\wr \mathbb{D}_n.$$

By Lemma 1.9

$$G \cong G_1 \times (G_2, G_3, G_4) \wr\wr \mathbb{D}_n,$$

which belongs to $\mathcal{F}$. $\square$

Lemma 5.30. *Each group in $\mathcal{F}$ is isomorphic to a group in* Stab(PLANAR).

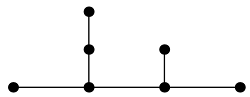
Proof. We prove the statement by induction on the number of operations needed to construct a group in $\mathcal{F}$. Clearly, the trivial group is in Stab(PLANAR). Let $H \in \mathcal{F}$ be nontrivial, then it is constructed by one of the operations (b)–(f), for which we prove the statement separately.

Let $H = G_1 \times G_2 \in \mathcal{F}$, for $G_1, G_2 \in$ Stab(PLANAR). By the induction hypothesis, there are X and Y such that $G_1 \cong \mathrm{Aut}(X)_u$ and $G_2 \cong \mathrm{Aut}(Y)_v$. If $X \cong Y$, we modify Y by attaching an end-vertex of a copy of P_2 to all vertices in $V(Y) \setminus \{v\}$. We form a Z with $w \in V(Z)$, where w arises by identifying u and v. Now, $H = G_1 \times G_2 \cong \mathrm{Aut}(Z)_w \in$ Stab(PLANAR).

Let $H = G \wr \mathbb{S}_n \in \mathcal{F}$, for $G \in$ Stab(PLANAR). By the induction hypothesis, there is a graph X and u such that $G \cong \mathrm{Aut}(X)_u$. We form a Z with $w \in V(Z)$, where w is arises by identifying n copies of X at the vertex u. Now, $H = G \wr \mathbb{S}_n \cong \mathrm{Aut}(Z)_w \in$ Stab(PLANAR).

Let $H = G \wr \mathbb{Z}_n \in \mathcal{F}$, for $G \in$ Stab(PLANAR). By the induction hypothesis, there is a graph X and u such that $G \cong \mathrm{Aut}(X)_u$. We form a graph Z as follows.

We take an n-wheel W_n with the center w and outer vertices labeled $\{0,\ldots,n-1\}$. We indentify the vertex i with the vertex u in a copy of X. Finally, we replace every edge joining the vertices i and $i+1 \mod n$ by the graph below.



Now, $H = G \wr \mathbb{Z}_n \cong \mathrm{Aut}(Z)_w \in \mathrm{Stab}(\mathsf{PLANAR})$.

Let $H = (G_1, G_2) \wr\wr \mathbb{D}_n \in \mathcal{F}$, for $G_1, G_2 \in \mathrm{Stab}(\mathsf{PLANAR})$ and $n \geq 3$ odd. By the induction hypothesis, there are graphs X and Y such that $G_1 \cong \mathrm{Aut}(X)_u$ and $G_2 \cong \mathrm{Aut}(Y)_v$. We form a graph Z as follows. We take an $3n$-wheel W_{3n} with the center w and outer vertices labeled $\{0,\ldots,3n-1\}$. For every $i = 0 \pmod 3$, we indentify the vertex i with the vertex v in a copy of Y. For every $i \neq 0 \pmod 3$, we identify the vertex i with the vertex u in a copy of X. Now, $H = (G_1, G_2) \wr\wr \mathbb{D}_n \cong \mathrm{Aut}(Z)_w \in \mathrm{Stab}(\mathsf{PLANAR})$.

Let $H = (G_1, G_2, G_3) \wr\wr \mathbb{D}_n \in \mathcal{F}$, for $G_1, G_2, G_3 \in \mathrm{Stab}(\mathsf{PLANAR})$ and $n \geq 2$ even. By the induction hypothesis, there are graphs X, Y, and Y' such that $G_1 \cong \mathrm{Aut}(X)_u$, $G_2 \cong \mathrm{Aut}(Y)_v$, and $G_3 \cong \mathrm{Aut}(Y')_{v'}$. We form a graph Z as follows. We take an $4n$-wheel W_{4n} with the center w and outer vertices labeled $\{0,\ldots,4n-1\}$. For every $i = 1 \pmod 2$, we indentify the vertex i with the vertex u in a copy of X. For every $i = 0 \pmod 4$, we identify the vertex i with the vertex v in a copy of Y. For every remaining i, we identify the vertex i with the vertex v' in a copy of Y'. Now, $H = (G_1, G_2, G_3) \wr\wr \mathbb{D}_n \cong \mathrm{Aut}(Z)_w \in \mathrm{Stab}(\mathsf{PLANAR})$.

Figure 5.6 demonstrates the last three constructions. $\square$

5.6.2 Composing spherical groups with vertex-stabilizers

Finally, we can state our main theorem. Recall that for every spherical group Q, the complete sequence of pairwise distinct isomorphism classes of orbits of the action of Q is in terms of the respective point stabilizers determined in the last column of Table 5.1. Each representative of an orbit of Q can be constructed as the set of left cosets of the stabiliser with the action of Q by left multiplication.

Theorem 5.31. *We have* $\mathrm{Aut}(\mathsf{PLANAR}) = \mathcal{P}$*, where* $\mathcal{P}$ *is defined as follows: if* $G_1, \ldots, G_m \in \mathrm{Stab}(\mathsf{PLANAR})$ *and* Q *is a spherical group acting on* Ω *with* m *pairwise non-isomorphic orbits, then* $(G_1, \ldots, G_m) \wr\wr Q \in \mathcal{P}$*, where the index sets* $\Omega_1, \ldots, \Omega_m$ *and the action of* Q *on* $\Omega = \bigcup_{i=1}^m \Omega_i$ *determining the inhomogeneous wreath product is defined by left multiplication on the left coset spaces* Ω_i *of subgroups of* Q*, listed in the last entry of the respective row in Table 5.1.*

Proof of Theorem 5.31. By Lemma 5.27, we have that

$$\mathrm{Stab}(\mathsf{PLANAR}) = \mathrm{Fix}(\mathsf{PLANAR}),$$

thus we shall use them interchangebly.

$\mathrm{Aut}(\mathsf{PLANAR}) \subseteq \mathcal{P}$: Let X be a planar graph, X^R be its reduction and let $Q \cong \mathrm{Aut}(X^R)$. By Lemma 5.26, Q is a spherical group. By Theorem 5.12, $\mathrm{Aut}(X) \cong (K_1, \ldots, K_m) \wr\wr_{\Omega'} Q$, where $\Omega' = E(X^R)$, and m is the number of edge-orbits in the action of Q on Ω', and $G_i \in \mathrm{Fix}(\mathsf{PLANAR})$, for $i = 1, \ldots, m$. By Lemma 1.9, there are groups $G_1, \ldots, G_m \in \mathrm{Fix}(\mathsf{PLANAR})$ such that $\mathrm{Aut}(X) \cong$

$(G_1, \dots, G_m) \wr_\Omega Q$, where the Q-sets Ω_i, for $i = 1, \dots, m$, are pairwise non-isomorphic. In Table 5.1 (last column), each Q-set Ω_i is determined as the left coset space of a subgroup of Q.

$\mathrm{Aut}(\mathsf{PLANAR}) \supseteq \mathcal{P}$: If $G_1, \dots, G_m \in \mathrm{Stab}(\mathsf{PLANAR})$ and Q is a spherical group acting on $\Omega = \bigcup_{i=1}^m \Omega_i$ with pairwise non-isomorhic orbits $\Omega_1, \dots, \Omega_m$, then let $G = (G_1, \dots, G_m) \wr Q \in \mathcal{P}$. First, we construct a 3-connected primitive planar graph Y, with $\mathrm{Aut}(Y) \cong Q$, and with orbits $V_1, \dots, V_m \subseteq V(Y)$ such that the Q-sets V_i and Ω_i are isomorphic. Such a 3-connected planar graph Y is uniquely determined by its quotient Y/Q embedded in the corresponding orbifold S^2/Q. For each spherical group Q, such a quotient is depicted in Figure 5.4. It may happen that Q is isomorphic to a proper subgroup of $\mathrm{Aut}(Y)$, in this case we use a suitable coloring of edges to ensure $\mathrm{Aut}(Y) \cong Q$.

Now, to construct X, we proceed as follows. For $i = 1, \dots, m$, we choose a planar graph Y_i with $G_i \cong \mathrm{Aut}(Y_i)_{(\partial Y_i)}$, and for every $v \in V_i$, we take a copy of Y_i and identify the unique vertex of ∂Y_i with v. $\square$

6. Automorphism groups of maps in linear time

6.1 Introduction

In this chapter, we are interested in planar graphs and, more generally, graphs of bounded genus. In 1966, Weinberg [161] gave a very simple quadratic algorithm for the graph isomorphism of planar graphs. This was improved by Hopcroft and Tarjan [93] to $\mathcal{O}(n \log n)$. Building, on this earlier work, Hopcroft and Wong [94] published in 1974 a paper, where they described a linear-time algorithm for isomorphism testing of planar graphs.

For graphs on surfaces of higher genus, the graph isomorphism problem seems much harder. This can be perhaps explained in the following way. We can rather easily reduce the problem to 3-connected graphs. For planar graphs, the famous result of Whitney [163] says that embeddings of 3-connected planar graphs in the plane are (combinatorially) unique. However, for every connected surface S of non-positive Euler characteristic, there exist 3-connected graphs with exponentially many embeddings into S. This makes an essential difference between planar graphs and graphs of higher genus.

For quite a long time it has been known that the isomorphism of bounded genus graphs can be solved in time $n^{\mathcal{O}(g)}$, where g is the genus of the underlying surface; see for example [137]. However, an interesting question is whether the result of Hopcroft and Wong [94] can be generalized also for the bounded genus graphs, i.e., whether the isomorphism problem for graphs of bounded genus can be solved in time $f(g) \cdot n$, for some computable function f. This motivates the study of the isomorphism problem for embedded graphs first.

By a *topological map* we mean a 2-cell decomposition of a closed compact surface, i.e., an embedding of a graph into a surface such that every face is homeomorphic to an open disc. An *isomorphism* of two maps is an isomorphism of the underlying graphs, which preserves the vertex-edge-face incidences. In particular, a map isomorphism induces a homeomorphism of the underlying surfaces. Our main result reads as follows.

Theorem 6.1. *Let M_1 and M_2 be maps on a surface of genus g. The set of all isomorphisms* $\mathrm{Iso}(M_1, M_2)$ *from M_1 to M_2 can be determined in time $f(g) \cdot (\|M_1\| + \|M_2\|)$, where f is some computable function and $\|M\|$ denotes the size of the map M.*

Determining the set of all isomorphisms between two maps is closely related to finding the generators of the automorphism group $\mathrm{Aut}(M)$ of a map M, where an automorphism of M is just an isomorphism $M \to M$. More precisely, the set of all isomorphisms $M_1 \to M_2$ can be expressed as a composition $\psi \cdot \mathrm{Aut}(M_1)$ where $\psi : M_1 \to M_2$ is any isomorphism. Thus, our first result goes hand-in-hand with the following.

Theorem 6.2. *Let M be a map on a surface of genus g. The generators of the automorphism group* $\mathrm{Aut}(M)$ *of M can be computed in time $f(g) \cdot \|M\|$, where f is some computable function and $\|M\|$ denotes the size of the map M.*

Colbourn and Booth [40] proposed a way to modify the Hopcroft-Wong algorithm [94] to compute the generators of the automorphism group of a spherical map. However, they state the following: "*We ... base our automorphism algorithms on the Hopcroft-Wong algorithm. Necessarily, we will only be able to sketch our procedure. A more complete description and a proof of correctness would require a more thorough analysis of the Hopcroft-Wong algorithm than has yet appeared in the literature.*" Sadly, the situation has not changed since, and the only available description of the Hopcroft-Wong algorithm is the extended abstract [94], which contains no proof of correctness and running time.[1] Our contribution also fills in this gap and we obtain much better insight into the Hopcroft-Wong algorithm by solving the problem in a much greater generality; see [99] as well.

Roughly speaking, the key idea of the Hopcroft-Wong algorithm is to try to apply contractions of edges to obtain two smaller isomorphic maps. In order to do this, edges must be chosen canonically, which is not always possible. Since Hopcroft and Wong consider only the spherical case, this situation occurs only in one special case. However, on the surfaces of higher genus, this situation is quite common and requires a completely different, more systematic, approach. As a consquence of considering the problem on the higher genus, our approach turnes out to be much simpler even for planar graphs than the approach originally proposed by Colbourn and Booth [40].

The Hopcroft-Wong algorithm reduces spherical maps to maps having the same degrees of vertices and also the same degrees of faces (e.g. Platonic solids). These maps are then treated separately. We, however, relax this condition and instead reduce our map to a map having the same cyclic vector of face sizes at each vertex (e.g. on sphere these also include Archimedean solids). The number of such maps is bounded for surfaces of genus $g > 1$, and for surfaces of genus $g \leq 1$ we give some special algorithms. This, surprisingly, allows a much more unified method of reducing the map, while preserving its automorphisms and isomorphisms.

Simultaneous conjugation problem. The problems of testing isomorphism of maps and computing the generators of the automorphism group of a map are related to the problem of *simultaneous conjugation*. In the latter problem, the input consists of two sets of permutations $\alpha_1, \dots, \alpha_d$ and $\beta_1, \dots, \beta_d$ on the set $\{1, \dots, n\}$, each of which generates a transitive subgroup of the symmetric group. The goal is to find a permutation γ such that $\gamma\alpha_i\gamma^{-1} = \beta_i$, for $i = 1, \dots, d$. Let us observe that this problem is a generalization of the map isomorphism problem. If α_1 and β_1 are involutions, $d = 2$, and the set $\{1, \dots, n\}$ is identified with the set of darts of a map on a surface (see Section 6.2 for definitions), then this problem is exactly the map isomorphism problem. If further $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$, we get the map automorphism problem.

Since mid 1970s it has been known that the simultaneous conjugation problem can be solved in time $\mathcal{O}(dn^2)$ [68, 90]. A faster algorithm, with running time $\mathcal{O}(n^2 \log d / \log n + dn \log n)$, was found only recently [27]. This implies an $\mathcal{O}(n^2 / \log n)$ algorithm for the isomorphism and automorphism problems for maps

[1]The PhD thesis of Wong also does not bring any new information compared to [94].

of unrestricted genus. In complexity theory, this is not considered to be a "truly subquadratic" algorithm. This motivates the following conjecture.

Conjecture 6.3. *There is no $\varepsilon > 0$ for which there is an algorithm for testing isomorphism of maps of unrestricted genus in time $\mathcal{O}(n^{2-\varepsilon})$.*

An interesting open subproblem is to prove a conditional "truly superlinear" lower bound for any of the mentioned problems. There has been some progress in the direction of providing a lower bound. In particular it is known that the communication complexity of the simultaneous conjugation problem is $\Omega(dn \log(n))$, for $d > 1$, and that under the decision tree model the search version of the simultaneous conjugation problem has lower bound of $\Omega(n \log n)$ [26].

6.2 Preliminaries: maps on surfaces

A *map* M is 2-cell decomposition of a closed connected compact surface S. The 0-cells, 1-cells and 2-cells are called respectively the *vertices*, the *edges* and the *faces* of M. Equivalently, a map is defined by an embedding $\iota\colon X \to S$ of a connected graph X into S such that every connected component of $S \setminus \iota(X)$ is homeomorphic to an open disc. By $V(M)$, $E(M)$, and $F(M)$ we denote the sets of vertices, edges, and faces of M, respectively. We put $v(M) := |V(M)|$, $e(M) := E(M)$, and $f(M) := |F(M)|$.

Recall that closed connected compact surfaces are characterized by two invariants: the orientability and the Euler characteristic χ.

The above invariants of M are related as follows.

Theorem 6.4 (Euler-Poincaré formula)**.** *Let M be a map on a surface S. Then*

$$v(M) - e(M) + f(M) = \chi(S) = \begin{cases} 2 - 2g, & \text{if } S \text{ orientable of genus } g; \\ 2 - \gamma, & \text{if } S \text{ is non-orientable of genus } \gamma. \end{cases}$$

In what follows, we present an algebraic description of a map suitable for investigation of automorphisms and isomorphisms between maps. We follow the description from [97], where the reader can find more details. For some deep reasons we introduce different models for maps on orientable surfaces and for non-orientable maps.

Oriented maps. Even though our main concern is in general maps, a large part of our algorithm deals with maps on orientable surfaces. An *oriented map* is a map on an orientable surface with a fixed global orientation. Every oriented map can be combinatorially described as a triple (D, R, L). Here, D is the set of *darts*. By a dart we mean an oriented edge. Hence, each edge gives rise to two darts. The permutation $R \in \mathrm{Sym}(D)$, called *rotation*, is the product $R = \Pi_{v \in V} R_v$, where each R_v cyclically permutes the darts originating at $v \in V$, following the chosen orientation around v. The *dart-reversing involution* $L \in \mathrm{Sym}(D)$ is an involution of D that, for each edge, swaps the two oppositely directed darts arising from the edge.

Formally, a *combinatorial oriented map* is any triple $M = (D, R, L)$, where D is a finite non-empty set of *darts*, R is any permutation of darts, L is a fixed-point-free involution of D, and the group $\langle R, L \rangle \leq \mathrm{Sym}(D)$ is transitive on D. By the size $\|M\|$ of the map, we mean the number of darts $|D|$.

The group $\langle R, L \rangle$ is called the *monodromy group* of M. The vertices, edges, and faces of M are in one-to-one correspondence with the cycles of the permutations R, L, $R^{-1}L$, respectively. By the phrase "a dart x is incident to a vertex v" we mean that $x \in R_v$. Similarly, "x is incident to a face f" means that x belongs to the boundary walk of f defined by the respective cycle of $R^{-1}L$. By the *degree of a face* we mean the length of its boundary walk. A face of degree d will be called a d-face. Note that each dart is incident to exactly one face. For convenience, we frequently use a shorthand notation $x^{-1} = Lx$, for $x \in D$. The *dual* of an oriented map $M = (D, R, L)$ is the oriented map $M^* = (D, R^{-1}L, L)$.

Apart from standard map theory references, we need to introduce labeled maps. A *planted tree* is a rooted tree embedded in the sphere, i.e., a planted tree is a spherical map having exactly one face. We say that a planted tree is *integer-valued* if to each vertex there is assigned an integer. A *dart-labeling* of an oriented map $M = (D, R, L)$ is a mapping $\ell \colon D \to \mathcal{T}$, where $\mathcal{T}$ is the set of integer-valued planted trees. A *labeled oriented map* M is a 4-tuple (D, R, L, ℓ). The *dual map* is the map M^* defined as $M^* = (D, R^{-1}L, L, \ell)$.

Two labeled oriented maps $M_1 = (D_1, R_1, L_1, \ell_1)$ and $M_2 = (D_2, R_2, L_2, \ell_2)$ are *isomorphic*, in symbols $M_1 \cong M_2$, if there exists a bijection $\psi \colon D_1 \to D_2$, called an *orientation-preserving isomorphism* from M_1 to M_2, such that

$$\psi R_1 = R_2 \psi, \quad \psi L_1 = L_2 \psi, \quad \text{and} \quad \ell_1 = \ell_2 \psi. \tag{6.1}$$

The *set of orientation-preserving isomorphisms* from M_1 to M_2 is denoted by $\mathrm{Iso}^+(M_1, M_2)$. The *orientation-preserving automorphism group* of M is the set $\mathrm{Aut}^+(M) := \mathrm{Iso}^+(M, M)$. Algebraically, the group $\mathrm{Aut}^+(M)$ is the label preserving subgroup of the *centralizer* of the monodromy group $\langle R, L \rangle$ in $\mathrm{Sym}(D)$.

The following statement, well-known for unlabeled maps, extends easily to labeled maps.

Theorem 6.5. *Let M_1 and M_2 be labeled oriented maps with sets of darts D_1 and D_2, respectively. For every $x \in D_1$ and every $y \in D_2$, there exists at most one isomorphism $M_1 \to M_2$ mapping x to y. In particular,* $\mathrm{Aut}^+(M_1)$ *is fixed-point-free on D_1.*

Corollary 6.6. *Let M_1 and M_2 be labeled oriented maps with sets of darts D_1 and D_2, respectively. If $x \in D_1$ and $y \in D_2$, then it can be checked in time $\mathcal{O}(|D_1| + |D_2|)$ whether there is an isomorphism mapping x to y.*

Chirality. The *mirror image* of an oriented map $M = (D, R, L)$ is the oriented map $M^{-1} = (D, R^{-1}, L)$. Similarly, the *mirror image* of labeled oriented map $M = (D, R, L, \ell)$ is the map $M^{-1} = (D, R^{-1}, L, \ell^{-1})$, where $\ell^{-1}(x)$ is the mirror image of $\ell(x)$ for each $x \in D$.

An oriented map M is called *reflexible* if $M \cong M^{-1}$. Otherwise the maps M and M^{-1} form a *chiral pair*. For example, all the Platonic solids are reflexible. The set of *all isomorphisms* from M_1 to M_2 is defined as $\mathrm{Iso}(M_1, M_2) := \mathrm{Iso}^+(M_1, M_2) \cup \mathrm{Iso}^+(M_1, M_2^{-1})$. Similarly, we put $\mathrm{Aut}(M) := \mathrm{Iso}(M, M)$.

Maps on all surfaces. Let M be a map on any, possibly non-orientable, surface. In general, a *combinatorial non-oriented map* is a quadruple (F, λ, ρ, τ), where F is a finite non-empty set of *flags*, and $\lambda, \rho, \tau \in \mathrm{Sym}(F)$ are fixed-point-free[2] involutions such that $\lambda\tau = \tau\lambda$ and the group $\langle \lambda, \rho, \tau \rangle$ acts transitively on F. By the *size* $\|M\|$ of the map M we mean the number of flags $|F|$.

Each flag corresponds uniquely to a vertex-edge-face incidence triple (v, e, f). Geometrically, it can be viewed as the triangle defined by v, the center of e, and the center f. The group $\langle \lambda, \rho, \tau \rangle$ is called the *non-oriented monodromy group* of M. The vertices, edges, and faces of M correspond uniquely to the orbits of $\langle \rho, \tau \rangle$, $\langle \lambda, \tau \rangle$, and $\langle \rho, \lambda \rangle$, respectively. Similarly, an *isomorphism* of two non-oriented maps M_1 and M_2 is a bijection $\psi \colon F \to F$ which commutes with λ, ρ, τ. The even-word subgroup $\langle \rho\tau, \tau\lambda \rangle$ has index at most two in the monodromy group of M. If it is exactly two, the map M is called *orientable*. For every oriented map (D, R, L) it is possible to construct the corresponding non-oriented map (F, λ, ρ, τ). Conversely, from an orientable non-oriented map (F, λ, ρ, τ) it is possible to construct two oriented maps (D, R, L) and (D, R^{-1}, L).

Test of orientability. For a non-oriented map $M = (F, \lambda, \rho, \tau)$, it is possible to test in linear time if M is orientable [85, 129]. The *barycentric subdivision* B of M is constructed by placing a new vertex in the center of every edge and face, and then joining the centers of faces with the incident vertices and with the center of the incident edges. The dual of B is 3-valent map, i.e., every vertex is of degree three.

Theorem 6.7. *A map $M = (F, \lambda, \rho, \tau)$ is orientable if and only if the underlying 3-valent graph of the dual of the barycentric subdivision of M is bipartite.*

Face-normal maps. A map is called *face-normal*, if all its faces are of degree at least three. It is well-known that every face-normal map on the sphere or on the projective plane has a vertex of degree at most 5. Using the Euler-Poincaré formula, this can be generalized for other surfaces.

Theorem 6.8. *Let S be a closed compact surface with Euler characteristic $\chi(S) \leq 0$ and let M be a face-normal map on S. Then there is a vertex of valence at most $6(1 - \chi(S))$.*

Proof. A bound for maximum degree is achieved by a triangulation, thus we may assume that M is a triangulation. We have $f = 2e/3$. By plugging this in the Euler-Poincaré formula and using the Handshaking lemma, we obtain $3v - \overline{d}v/2 = 3\chi(S)$, where $\overline{d}$ is the average degree. By manipulating the equality, we get $\overline{d} - 6 = -6\chi(S)/v$. Since $\chi(S) \leq 0$, the right hand side is maximized for $v = 1$. We conclude that $\overline{d} \leq 6(1 - \chi(S))$. □

The lexicographically minimal representative of the cyclic vector of degrees of faces incident with a vertex v, induced by the chosen global orientation, is called the *local type* of v. A vertex is *light* if its local type is minimal with respect to lexicographic linear ordering, otherwise a vertex is *heavy*. In particular, a light vertex is of minimum degree. Theorem 6.8 implies that, in a face-normal map, the degree of light vertices is bounded by a function of $\chi(S)$.

[2] It is possible to extend the theory to maps on surfaces with boundaries by allowing fixed points of λ, ρ, τ. [28]

Uniform and homogeneous maps. A map is *uniform*[3] if the local types of all vertices are the same. A map is *homogeneous* of type $\{k, \ell\}$ if every vertex is of degree k and every face is of degree ℓ.

A *dipole* is a 2-vertex spherical map which is dual to a spherical cycle. *A bouquet* is a one-vertex map that is a dual of a *planted star* (a tree with at most one vertex of degree > 1).

Example 6.9. The face-normal uniform spherical maps are: the 5 Platonic solids, the 13 Archimedean solids, pseudo-rhombicuboctahedron, prisms, antiprisms, and cycles of length at least 3. It easily follows from Euler's formula that the spherical homogeneous maps are the 5 Platonic solids, cycles, and dipoles.

6.3 Overview of the algorithm

We provide a high-level overview of the whole algorithm determining the automorphism group of a map. The input consists of a non-oriented map given by the quadruple $N = (F, \lambda, \rho, \tau)$.

First, using Theorem 6.7, we test whether N is orientable or not. If the map is orientable, then we know that the underlying surface is orientable and we fix a global orientation of the surface. We construct two oriented maps $M = (D, R, L)$ and $M^{-1} = (D, R^{-1}, L)$ representing N.

We start by determining $\mathrm{Aut}^+(M)$. On the map M, we perform a sequence of elementary local reductions (Section 6.4). There are two types of reductions: normalization and elimination of vertices of minimum degree. The normalization is of the highest priority and its purpose is to ensure that the resulting map is face-normal. In a face-normal map, it is guaranteed by Theorem 6.8 that there is a vertex of small degree. The second elementary reduction replaces a vertex of minimum degree by a polygon connecting its higher-degree neighbours and reconnecting the other incident edges (see Figure 6.3). These two reductions are applied until we are left with a map which has all vertices of degree k. Now, we observe that our reductions do not really depend on the degrees of vertices, but rather on some vertex-labelling (not related to dart labelling) which is linearly ordered. At this stage we can no longer distinguish vertices based on their degreee. We refine the procedure by using the local types instead of degrees. It follows from Theorem 6.8 that the number of local types sufficient to consider is bounded. Thus, our reduction can be applied in the same way, but instead of degrees we use local types. The result is a labeled face-normal uniform oriented map $M' = (D', R', L', \ell')$ with $\mathrm{Aut}^+(M) \cong \mathrm{Aut}^+(M')$ and $D' \subseteq D$; for more details see Section 6.4.

The number of face-normal uniform oriented map M' on a surface of genus $g > 1$ is bounded by a fucntion of g (Proposition 6.13), which means that a brute-force approach is sufficient to determine $\mathrm{Aut}^+(M')$. For the case of sphere and torus, the problem is non-trivial since there are infinite families of face-normal uniform maps and a special treatment is necessary; for more details see Section 6.5. Now, since $\mathrm{Aut}^+(M)$ acts fixed-point-freely on D and $D' \subseteq D$, there is a unique way to extend $\mathrm{Aut}^+(M')$ to $\mathrm{Aut}^+(M)$. Finally, to construct $\mathrm{Aut}(M)$, we run the whole algorithm again to determine $\mathrm{Iso}(M, M^{-1})$.

[3] In [9] Babai uses the term semiregular instead of uniform.

If the map is N is non-orientable, we construct its oriented antipodal double-cover $\widetilde{M} = (D, R, L) = (F, \rho\tau, \tau\lambda)$. We show that $\mathrm{Aut}(N) \leq \mathrm{Aut}^+(\widetilde{M})$, and therefore, we can again apply our algorithm to determine $\mathrm{Aut}^+(\widetilde{M})$. Here, the most difficult part is to determine $\mathrm{Aut}(N)$ within $\mathrm{Aut}^+(\widetilde{M})$. For the case of projective plane and Klein bottle the problem is highly non-trivial and a special treatment is again needed, while for the other cases, again, a brute force approach is sufficient; for more details see Section 6.6.

6.4 From oriented to uniform oriented maps

In this section, we describe in detail a set of elementary reductions defined on labeled oriented maps, given by a quadruple (D, R, L, ℓ). The output of each elementary reduction is always a quadruple (D', R', L', ℓ'), satisfying $D' \subseteq D$, $v(M') + e(M') < v(M) + e(M)$, and $\mathrm{Aut}^+(M') = \mathrm{Aut}^+(M)$. If none of the reductions apply, the map is a uniform oriented map. This defines a function which assigns to a given oriented map M a unique labeled oriented map U with $\mathrm{Aut}^+(M) \cong \mathrm{Aut}^+(U)$. Since the darts of U form a subset of the darts of M, by semiregularity, every generator of $\mathrm{Aut}^+(U)$ can be extended to a generator of $\mathrm{Aut}^+(M)$ in linear time. We deal with the uniform oriented maps in Section 6.5.

After every elementary reduction, to ensure that $\mathrm{Aut}^+(M') = \mathrm{Aut}^+(M)$, we need to define a new labeling ℓ'. To this end, in the whole section, we assume that we have an injective function $\textbf{Label}\colon \mathbb{N} \times \bigcup_{k=1}^{\infty} \mathcal{T}^k \to \mathcal{T}$, where $\mathcal{T}$ is the set of all integer-valued planted trees. Moreover, we assume that the root of $\textbf{Label}(t, T_1, \ldots, T_k)$ contains the integer t, corresponding to the current step of the reduction procedure. After every elementary reduction, this integer is increased by one. In Section 6.7, we show how to evaluate **Label** in linear time.

Normalization. By Theorem 6.8, there is always a light vertex in a face-normal map. The purpose of the following reduction is to remove faces of degree one and two. This reduction is of the highest priority and it is applied until the map is one of the following: (i) face-normal, (ii) bouquet, (iii) dipole. In the cases (ii) and (iii), the whole reduction procedure stops with a uniform map. In the case (i), the reduction procedure continues with further reductions. We describe the reduction formally.

For technical reasons we split the reduction into two parts: deletion of loops, denoted by $\textbf{Loops}(M)$, and replacement of a dipole by an edge, denoted by $\textbf{Dipoles}(M)$.

Reduction Loops. If $M = (D, R, L, \ell)$ with $v(M) > 1$ contains loops, we remove them. Let $\mathcal{L}$ be the list of all maximal sequences of darts of the form $s = \{x_1, x_1^{-1}, \ldots, x_k, x_k^{-1}\}$, where $Rx_i = x_i^{-1}$, for $i = 1, \ldots, k$, $Rx_i^{-1} = x_{i+1}$ for $i = 1, \ldots, k-1$, and $Rx_k^{-1} \neq x_1$. By definition, $R^{-1}Lx_i = x_i$, hence x_i bounds a 1-face, for $i = 1, \ldots, k-1$; see Figure 6.1. Moreover, for each such sequence s, all the darts x_i are incident to the same vertex $v \in V(M)$. We say that the unique vertex v with $R_v = (x_0, x_1, x_1^{-1}, \ldots, x_k, x_k^{-1}, x_{k+1}, \ldots)$ is *incident* to s. We call the darts x_0 and x_{k+1} the *bounding darts* of the sequence s.

The new map $M' = (D'; R', L', \ell') =: \mathbf{Loops}(M)$ is defined as follows. First, we put $D' := D \setminus \bigcup_{s \in \mathcal{L}} s$, and $L' := L_{\restriction D'}$. Let $s = \{x_1, x_1^{-1}, \dots, x_k, x_k^{-1}\} \in \mathcal{L}$ with bounding darts x_0 and x_{k+1}. If v is incident to s, then we put $R'_v := (x_0, x_{k+1}, \dots)$, else we put $R'_v := R_v$. Moreover, we put $\ell'(x_0) := \mathbf{Label}(t, a_0, \dots, a_k)$ and $\ell'(x_{k+1}) := \mathbf{Label}(t, a_{k+1}, b_k, \dots, b_1)$, where t is the current step, $a_i = \ell(x_i)$, for $i = 0, \dots, k+1$, and $b_i = \ell(x_i^{-1})$, for $i = 1, \dots, k$. For every $x \in D'$ which is not a bounding dart in M, we put $\ell'(x) := \ell(x)$. We obtain a well-defined map M' with no faces of valence one; see Figure 6.1.

Lemma 6.10. *Let $M_i = (D_i, R_i, L_i, \ell_i)$, $i = 1, 2$ where $D_1 \cap D_2 = \emptyset$, be labeled oriented maps. Let $M'_1 := \mathbf{Loops}(M_1)$ and $M'_2 := \mathbf{Loops}(M_2)$. Then $\mathrm{Iso}^+(M_1, M_2)_{\restriction D'_1} = \mathrm{Iso}^+(M'_1, M'_2)$. In particular, $\mathrm{Aut}^+(M_1)_{\restriction D'_1} = \mathrm{Aut}^+(M'_1)$.*

Proof. If M has no 1-faces or if M is a bouquet, then $M' = M$ and there is nothing to prove. Otherwise, let $\psi\colon M_1 \to M_2$ be an isomorphism. We prove that $\psi' := \psi_{\restriction D'_1}$ is an isomorphism of M'_1 and M'_2. Since ψ preserves the set of 1-faces, the mapping ψ' is a well-defined bijection. We check the commuting rules (6.1) for ψ'.

By the definition of **Loops**, $L'_i = L_{i \restriction D_i}$, for $i = 1, 2$. Thus, we have $\psi' L'_1 = L'_2 \psi'$. As concerns the permutations R'_1 and R'_2, we need to check the commuting rule only at the darts preceding a sequence of 1-faces. With the above notation, using the definition of M'_1 and M'_2, and the fact that ψ is an isomorphism, we get

$$\psi' R'_1 x_0 = \psi' R_1 (L_1 R_1)^k x_0 = \psi R_1 (L_1 R_1)^k x_0 =$$

$$= R_2 (L_2 R_2)^k \psi x_0 = R_2 (L_2 R_2)^k \psi' x_0 = R'_2 \psi' x_0.$$

Finally, for ℓ'_1 and ℓ'_2, we have, by the definition of **Loops**,

$$\ell'_1(x_0) = \mathbf{Label}(t, \ell_1(x_0), \dots, \ell_1(x_k)) = \mathbf{Label}(t, \ell_2(\psi x_0), \dots, \ell_2(\psi x_k)) = \ell'_2(\psi' x_0)$$

if and only if

$$\ell_1(x_i) = \ell_2(\psi x_i), \text{ for } i = 0, \dots, k,$$

which is satisfied since ψ is an isomorphism. Similarly, $\ell'_1(x_{k+1}) = \ell'_2(\psi x_{k+1})$.

Conversely, let $\psi'\colon M'_1 \to M'_2$ be an isomorphism. With the above notation, we have

$$x_i = R_1 (L_1 R_1)^i x_0 \quad \text{and} \quad x_{k+1} = R'_1 x_0.$$

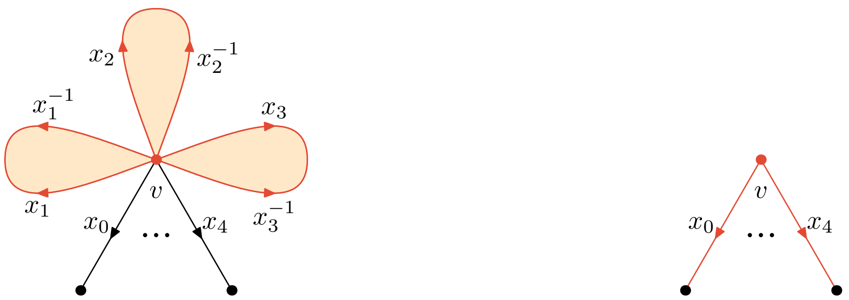


Figure 6.1: A sequence of darts $x_1, x_1^{-1}, x_2, x_2^{-1}, x_3, x_3^{-1}$ with bounding darts x_0 and x_4.

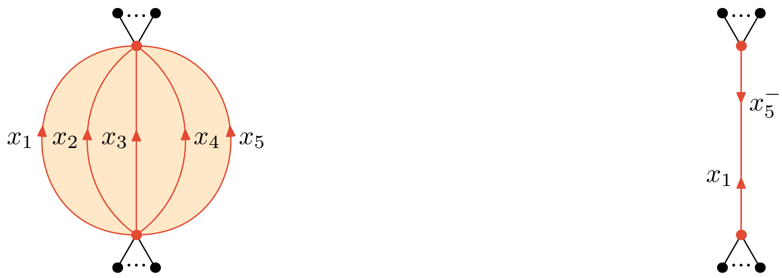


Figure 6.2: A sequence of darts $x_1, \ldots, x_5$ forming a dipole.

Since **Label** is injective, it follows that there are $y_1, \ldots, y_k$ in $D_2 \setminus D'_2$ such that

$$y_i = R_2(L_2R_2)^i\psi' x_0.$$

Here we employ the fact that t is increased after every elementary reduction. This forbids the existence of an isomorphisms $\psi' : M'_1 \to M'_2$ taking a bounding dart to a dart that is not bounding, i.e., ψ' takes the set of bounding darts onto the set of bounding darts. We define an extension ψ of ψ' by setting $\psi x_i = y_i$, for $i = 1, \ldots, k$. It is straightforward to check that $\psi \in \mathrm{Iso}^+(M_1, M_2)$. $\square$

Reduction Dipoles. If $M = (D, R, L, \ell)$ with $v(M) > 2$ contains dipoles. Let $\mathcal{L}$ be the list of all maximal sequences $s = (x_1, \ldots, x_k)$ of darts, $k > 1$, satisfying $Rx_i = x_{i+1}$, $(R^{-1}L)^2 x_i = x_i$, and either $Rx_k \neq x_1$ or $Rx_1^{-1} \neq x_k^{-1}$; see Figure 6.2. Let $s^{-1} := (x_k^{-1}, \ldots, x_1^{-1}) \in \mathcal{L}$ be the *inverse* sequence. There are vertices u and v such that $R_u = (y_1, s, y_2, \ldots)$ and $R_v = (z_1, s^{-1}, z_2, \ldots)$, for some $y_1, y_2, z_1, z_2 \in D$. At least one of the sets $\{y_1, y_2\}$, $\{z_1, z_2\}$ is non-empty since otherwise $v(M) = 2$ and M is a dipole. We say that u, v are *incident* to s, s^{-1}, respectively; see Figure 6.2

The new map $M' = (D', R', L', \ell') =:$ **Dipoles**(M) is defined as follows. First, we put

$$D' := D \setminus \bigcup_{(x_1,\ldots,x_k)\in\mathcal{L}} \{x_2, \ldots, x_k\} \cup \{x_1^{-1}, \ldots, x_{k-1}^{-1}\}.$$

Let $s = (x_1, \ldots, x_k) \in \mathcal{L}$. If u and v are incident to s and s^{-1}, respectively, then we put $R'_u := (y_1, x_1, y_2 \ldots)$ and $R'_v := (z_1, x_k^{-1}, z_2, \ldots)$, else we put $R'_u := R_u$. Next, we put $L'x_1 := x_k^{-1}$, $L'x_k^{-1} := x_1$, and $L'x := Lx$ if $x \notin s \in \mathcal{L}$. Finally, we put $\ell'(x_1) :=$ **Label**$(t, a_1, \ldots, a_k)$ and $\ell'(x_k^{-1}) :=$ **Label**$(t, b_k, \ldots, b_1)$, where t is the current step, $a_i = \ell(x_i)$ and $b_i = \ell(x_i^{-1})$, for $i = 1, \ldots, k$. We put $\ell'(x) := \ell(x)$ for $x \notin s \in \mathcal{L}$. We obtain a well-defined map M' with no 2-faces; see Figure. 6.2.

Lemma 6.11. *Let $M_i = (D_i, R_i, L_i, \ell_i)$, $i = 1, 2$ where $D_1 \cap D_2 = \emptyset$, be labeled oriented maps. Let $M'_1 :=$* **Dipoles**(M_1) *and $M'_2 :=$* **Dipoles**(M_2)*. Then* $\mathrm{Iso}^+(M_1, M_2)_{\restriction D'_1} = \mathrm{Iso}^+(M'_1, M'_2)$*. In particular,* $\mathrm{Aut}^+(M_1)_{\restriction D'_1} = \mathrm{Aut}^+(M'_1)$*.*

Proof. Let $\psi\colon M_1 \to M_2$ be an isomorphism. We prove that $\psi' = \psi_{\restriction D'_1}$ is an isomorphism of M'_1 and M'_2. Since ψ preserves the set of 2-faces, the mapping ψ' is a well-defined bijection. We check the commuting rules (6.1) for ψ'.

We have $L'_1 x_1 = x_k^{-1} = L_1 R_1^{k-1} x_1$ and $L'_1 x_k^{-1} = L_1 R_1^{k-1} x_k^{-1}$ from the definition of **Dipoles**. Further,

$$\psi' L'_1 x_1 = \psi' L_1 R_1^{k-1} x_1 = \psi L_1 R_1^{k-1} x_1 = L_2 R_2^{k-1} \psi x_1 = L'_2 \psi' x_1,$$

and

$$\psi' L_1' x_k^{-1} = \psi' L_1 R_1^{k-1} x_k^{-1} = \psi L_1 R_1^{k-1} x_k^{-1} = L_2 R_2^{k-1} \psi x_k^{-1} = L_2' \psi' x_k^{-1}.$$

It follows that $\psi L_1' = L_2' \psi$.

For R_1' and R_2', it follows that we need to check the commuting rule only at the darts x_1 and x_k^{-1} bounding a sequence of 2-faces in M_1. With the above notation, using the definition of M_1' and M_2', and the fact that ψ is an isomorphism, we get

$$\psi' R_1' x_1 = \psi' R_1^{k-1} x_1 = \psi R_1^{k-1} x_1 = R_2^{k-1} \psi x_1 = R_2' \psi x_1 = R_2' \psi' x_1.$$

For x_k^{-1}, the verification of the commuting rules is similar.

For ℓ_1' and ℓ_2', we have, by the definition of **Dipoles**,

$$\ell_1'(x_1) = \textbf{Label}(s, \ell_1(x_1), \ldots, \ell_1(x_k)) =$$

$$= \textbf{Label}(s, \ell_2(\psi x_1), \ldots, \ell_2(\psi x_k)) = \ell_2'(\psi' x_1)$$

if and only if

$$\ell_1(x_i) = \ell_2(\psi x_i), \text{ for } i = 1, \ldots, k,$$

which is satisfied since ψ is an isomorphism. Similarly, we check that $\ell_1'(x_k^{-1}) = \ell_2'(\psi x_k^{-1})$.

Conversely, let $\psi' \colon M_1' \to M_2'$ be an isomorphism. With the above notation, we have

$$x_i = R_1^{i-1} x_1 \quad \text{and} \quad x_i^{-1} = R^{i-1} x_k^{-1},$$

for $i = 1, \ldots, k$. Since **Label** is injective, it follows that there are $y_2, \ldots, y_k$ in $D_2 \setminus D_2'$ such that

$$y_i = R_2^{i-1} \psi' x_1$$

determines a dipole. We define an extension ψ of ψ' by setting $\psi x_i := y_i$, for $i = 2, \ldots, k$. It is straightforward to check that the extension $\psi \in \mathrm{Iso}^+(M_1, M_2)$. $\square$

Face-normal maps. The input is a labeled face-normal oriented map $M = (D, R, L, \ell)$ and a list $\mathcal{L}$ of all light vertices of degree d which have at least one heavy neighbour. For every vertex $v \in \mathcal{L}$, we denote by $u_0, \ldots, u_{k-1}$, for some $1 \leq k \leq d$, the cyclic sequence of all heavy neighbours of v, following the prescribed orientation of the underlying surface. Denote by $x_0, x_1, \ldots, x_{k-1}$ the darts based at $u_0, u_1, \ldots, u_{k-1}$, joining u_j to v for $j = 0, \ldots, k-1$. Let $R_{u_i} = (y_i, x_i, z_i, \ldots)$, for $i = 0, \ldots, k-1$, and let

$$R_v = (x_0^{-1}, A_0, x_1^{-1}, A_1, \ldots, x_{k-1}^{-1}, A_{k-1}),$$

where each A_i is a (possibly empty) sequence of darts.

The new map $M' = (D', R', L', \ell') =: \textbf{Delete}(M)$ is defined as follows. We set $D' := D$ and $L' := L$. For a heavy vertex w with no light neighbour, we have $R_w' := R_w$. If $v \in \mathcal{L}$, with the above notation, we set $R_{u_i}' := (y_i, A_i, x_i, x_{i-1}^{-1}, z_i, \ldots)$. Moreover, we set $\ell'(x_i) := \textbf{Label}(t, \ell(x_i))$ and $\ell'(x_i^{-1}) := \textbf{Label}(t, \ell(x_i^{-1}))$, where t is the current step number; see Figure 6.3.

Lemma 6.12. *Let $M_i = (D_i, R_i, L_i, \ell_i)$, $i = 1, 2$ where $D_1 \cap D_2 = \emptyset$, be labeled oriented maps. Let $M_1' := \textbf{Delete}(M_1)$ and $M_2' := \textbf{Delete}(M_2)$. Then* $\mathrm{Iso}^+(M_1, M_2) = \mathrm{Iso}^+(M_1', M_2')$. *In particular,* $\mathrm{Aut}^+(M_1) = \mathrm{Aut}^+(M_1')$.

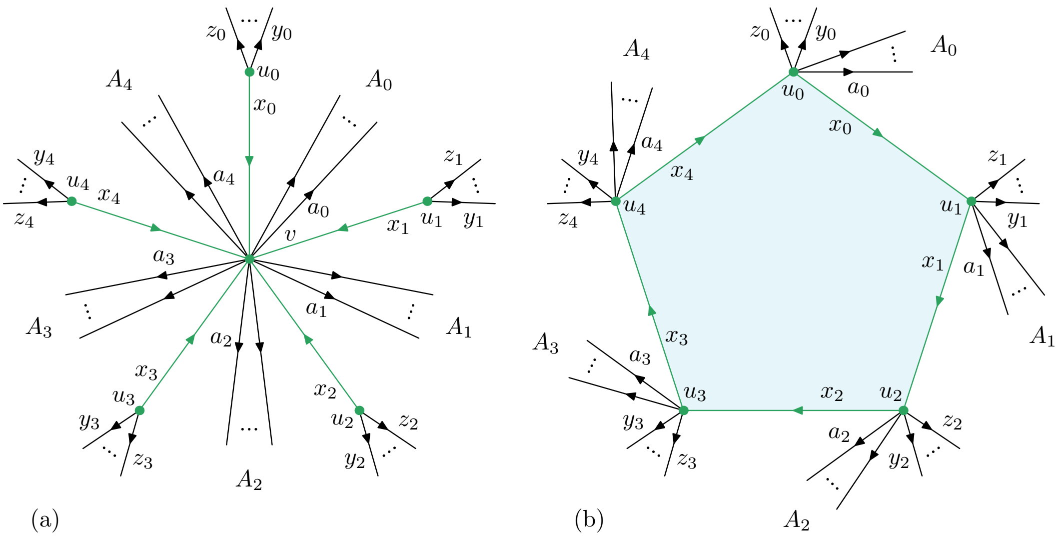


Figure 6.3: An example of the reduction deleting a vertex.

Proof. Let $\psi \colon M_1 \to M_2$ be an isomorphism. We prove that ψ is also an isomorphism of M_1' and M_2'. We check the commuting rules (6.1) for ψ. We have $L_i' = L_i$, for $i = 1, 2$, so $L_1'\psi = \psi L_2'$. For R_1' and R_2', we need to check the commuting rules only at $x_i, x_i^{-1}, y_i, a_i \in D_1'$, for $i = 0, \dots, k-1$, where a_i is the last dart in the sequence A_i. We have

$$\psi R_1' x_i = \psi R_1^{-1} L_1 x_i = R_2^{-1} L_2 \psi x_i = R_2' \psi x_i,$$

$$\psi R_1' x_i^{-1} = \psi R_1 L_1 x_i^{-1} = R_2 L_2 \psi x_i^{-1} = R_2' \psi x_i^{-1}.$$

It remains to check the commuting rules at each y_i and a_i. Note that if A_i is empty there is nothing to check. We have

$$\psi R_1' y_i = \psi R_1 L_1 R_1 y_i = R_2 L_2 R_2 \psi y_i = R_2' \psi y_i.$$

Further, using the relations $R_1' a_i = x_i = L_1 R_1^q a_i$, for some $q > 0$, we get

$$\psi R_1' a_i = \psi x_i = \psi L_1 R_1^q a_i = L_2 R_2^q \psi a_i = R_2' \psi a_i.$$

Putting it together, we proved that $\psi R_1' = R_2' \psi$. Clearly, $\ell_1'(x_i) = \ell_2'(\psi x_i)$ if and only if $\ell_1(x_i) = \ell_2(\psi x_i)$. Similarly for x_i^{-1}.

For the converse, we assume that $\psi R_1' = R_2' \psi$ and $\psi L_1' = L_2' \psi$ and we prove $\psi R_1 = R_2 \psi$ and $\psi L_1 = L_2 \psi$. Similarly as above, we need to check the commuting rules for $x_i, x_i^{-1}, y_i, a_i \in D_1$.

- By the definition of M_1' and M_2', we have $R_1 x_i = z_i = (R_1')^2 x_i$. Since **Label** is injective, we have $R_2 \psi x_i = \psi z_i = (R_2')^2 \psi x_i$. Using these relations, we get

$$\psi R_1 x_i = \psi (R_1')^2 x_i = (R_2')^2 \psi x_i = R_2 \psi x_i.$$

- By the definition of M_1' and M_2', we have $R_1 x_i^{-1} = R_1'^m L_1' x_i^{-1}$, for some m. Since **Label** is injective, we have $R_2 \psi x_i^{-1} = R_2'^m L_2' \psi x_i^{-1}$. Using these relations, we get

$$\psi R_1 x_i^{-1} = \psi R_1'^m L_1 x_i^{-1} = R_1'^m L_2' \psi x_i^{-1} = R_2 \psi x_i^{-1}.$$

- By the definition of M_1' and M_2', we have $R_1 y_i = x_i = R_1'^m y_i$, for some m. Since **Label** is injective, $R_2 \psi y_i = \psi x_i = R_2'^m \psi y_i$. Using these relations, we get
$$\psi R_1 y_i = \psi R_1'^m y_i = R_2'^m \psi y_i = R_2 \psi y_i.$$
- By the definition of M_1' and M_2', we have $R_1 a_i = L_1' R_1'^{-1} L_1' R_1' a_i$. Since **Label** is injective, $R_2 \psi a_i = L_2' R_2'^{-1} L_2' R_2' \psi a_i$. Using these relations, we get
$$\psi R_1 a_i = \psi L_1' R_1'^{-1} L_1' R_1' a_i = L_2' R_2'^{-1} L_2' R_2' \psi a_i = R_2 \psi a_i.$$

Putting it togehter, we proved that $\psi R_1 = R_2 \psi$, which implies that ψ is an isomorphism $M_1 \to M_2$. This completes the proof. □

6.5 Irreducible maps on orientable surfaces

In this section, we provide an algorithm computing the automorphism group of irreducible oriented maps, with fixed Euler characteristic, in linear time. The proof is split into three parts: negative Euler characteristic (Section 6.5.1), sphere (Section 6.5.2), and torus (Section 6.5.3).

6.5.1 Surfaces of negative Euler characteristic

If the Euler characteristic χ is negative, the irreducible maps are exactly all the uniform face-normal maps. We prove that the number of uniform face-normal maps is bounded by a function of χ. Therefore, generators of the automorphism group can be computed by a brute force approach. Note that the following lemma does not require the underlying surface to be orientable, it only requires χ to be negative.

Proposition 6.13. *The number of uniform face-normal maps on a closed compact surface S with Euler characteristic $\chi(S) < 0$ is bounded by a function of $\chi(S)$.*

Proof. Babai noted in [9, Theorem 3.3] that the Hurwitz Theorem (see, e.g. [17] or [85]) implies that the number of vertices of a uniform map M on S is at most $84|\chi(S)|$. By Theorem 6.8, the degree of a vertex of M is bounded by a function of $\chi(S)$ as well. Therefore, the number of edges is also bounded by a function of $\chi(S)$ and the theorem follows. □

Corollary 6.14. *Let $M = (D, R, L)$ be a uniform face-normal map on an orientable surface S with $\chi(S) < 0$. Then generators of* $\mathrm{Aut}(M)$ *can be computed in time $f(\chi(S))|D|$, for some computable function f.*

6.5.2 Sphere

By the definition of the reductions in Section 6.4, the irreducible spherical maps are the two-skeletons of the five Platonic solids, 13 Archimedean solids, pseudo-rhombicuboctahedron, prisms, antiprisms, cycles, dipoles, and bouquets.

In the first three cases, the automorphism group can be computed by a brute force approach. We show that for (labeled) prisms, antiprisms, dipoles and bouquets, the problem can be reduced to computing the automorphism group of a cycle.

Lemma 6.15. *For every labeled map M which is a prism, an antiprism, a dipole or a bouquet, there is a labeled cycle M' such that* $\mathrm{Aut}^+(M) \cong \mathrm{Aut}^+(M')$. *Moreover M' can be constructed in linear time.*

Proof. The idea of the proof is to take the dual M^* of M, if M is not a bouquet, and apply the reductions defined in Section 6.4, following the order defined by the priorities.

Clearly, the dual of a dipole is a cycle. The dual of an n-prism, is an n-bipyramid. It is easy to see that an n-bipyramid, for $n \neq 4$, is reduced to a $3n$-dipole by applying constant number of reductions reduction. Similarly, the dual of an n-antiprism, for $n \neq 3$, is again a reducible map, which is reduced by applying a constant number of reductions to a $2n$-dipole. Every prism and antiprism is therefore transformed to a labeled cycle.

Concerning bouquets, we transform every n-bouquet to an n-cycle based on the same set darts. Formally, let $B_n = (D, R, L, \ell)$ be a bouquet. We set $D' = D$, $L' = L$ and $\ell'(x) = \mathbf{Label}(s, \ell(x))$. By definition the rotation consists of a single cycle of the form $R = (x_0, x_0^{-1}, x_1, x_1^{-1}, \dots, x_{n-1}, x_{n-1}^{-1})$. We set $R' = \prod_{i=0}^{d-1}(x_i^{-1}, x_{i+1})$. □

It remains to show how to test isomorphism of two labeled maps whose underlying graphs are cycles. In order to make the exposition simpler, we transform the labeling of darts $\ell\colon D \to \mathcal{U}$ of a labeled cycle M to a labeling of vertices. For every vertex v of M, there are two darts x, y incident with it, and we have $R_v = (x, y)$. The pair $(\ell(x), \ell(y))$ can be considered as a new label of the vertex v. Thus, the problem reduces to testing isomorphism of two *vertex-labeled* cycles.

Automorphisms and isomorphisms of labeled cycles. In this section we modify the algorithm which was given by Hopcroft and Wong [94] to test isomorphism of cycles. This algorithm is an essential tool, since we apply it several times as a black box in the rest of the text. In particular, we use it in the algorithm for uniform toroidal maps (the next subsection) and in the algorithm for computing the centralizer of a fixed-point-free involution in a certain 2-generated group; see Lemma 6.29. The latter application is necessary to compute the generators of the automorphism group of a map on the projective plane or on the Klein bottle.

Given cycles X_1 and X_2 with vertex-labelings ℓ_1 and ℓ_2, respectively, the following algorithm tests if there is an isomorphism $\psi\colon V(X_1) \to V(X_2)$ such that $\ell_1(v) = \ell_2(\psi(v))$, for every $v \in V(X_1)$. For simplicity, we assume that, at the start, if X_i, for $i = 1, 2$, has k different labels, for some $k \leq |V(X_i)|$, then the labels are the integers $1, \dots, k$ and the same coding is used in X_1 and X_2. Moreover, we fix an orientation of X_1 and X_2, so that for every vertex v its successor $\mathrm{suc}(v)$ is well defined.

- **Step 1:** We find an arbitrary vertex v_1 in X_1, with $\ell_1(v_1) \neq \ell_1(\mathrm{suc}(v_1))$. If no such vertex exists in X_1, then ℓ_1 is constant in which case it is easy to check if $X_1 \cong X_2$. Otherwise, we find $v_2 \in V(X_2)$ with $\ell_1(v_1) = \ell_2(v_2)$ and $\ell_1(\mathrm{suc}(v_1)) = \ell_2(\mathrm{suc}(v_2))$. If no such vertex v_2 exists, then X_1 and X_2 are not isomorphic.

- **Step 2:** For $i = 1, 2$, we construct the set S_i of all vertices u of X_i with $\ell_i(u) = \ell_i(v_i)$ and $\ell_i(\mathrm{suc}(u)) = \ell_i(\mathrm{suc}(v_i))$; see Figure 6.4. The sets S_1 and S_2 form independent sets in X_1 and X_2, respectively. Every isomorphism maps S_1 bijectively to S_2. If $|S_1| \neq |S_2|$, then X_1 and X_2 are not isomorphic.

- **Step 3:** For every $v \in S_i$ ($i = 1, 2$), we join v to $\mathrm{suc}(\mathrm{suc}(v))$ and remove $\mathrm{suc}(v)$. We relabel every vertex in S_i by k, where k is the smallest unused integer; see Figure 6.4.

- **Step 4:** We find an arbitrary vertex $v_1 \in S_1$ with $\ell_1(v_1) \neq \ell_1(\mathrm{suc}(v_1))$. If no such vertex exists, then we have $S_1 = V(X_1)$ and ℓ_1 is constant. It is easy to check if $X_1 \cong X_2$. Otherwise, we find $v_2 \in S_2$ with $\ell_2(v_2) = \ell_1(v)$ and $\ell_2(\mathrm{suc}(v_2)) = \ell_2(\mathrm{suc}(v))$. If no such vertex exists, then X_1 and X_2 are not isomorphic, and we stop.

- **Step 5:** For $i = 1, 2$, we remove from S_i every u with $\ell_i(\mathrm{suc}(u)) \neq \ell_i(\mathrm{suc}(v_i))$. The sets S_1 and S_2 form independent sets in X_1 and X_2, respectively. If $|S_1| \neq |S_2|$, then X_1 and X_2 are not isomorphic and we stop. We go to Step 3.

By X_i^{-1} we denote the labeled cycle X_i ($i = 1, 2$) with the reverse orientation.

Lemma 6.16. *Applying the above algorithm twice for the inputs* (X_1, X_2) *and* (X_1, X_2^{-1}) *with* X_2 *taken with the chosen and the reverse orientation, it is decided in linear time if two labeled cycles* X_1 *and* X_2 *are isomorphic as oriented maps.*

Proof. Let X_1' and X_2' be the graphs obtained from X_1 and X_2 after applying Step 3, respectively. It suffices to show that $X_1 \cong X_2$ if and only if $X_1' \cong X_2'$.

Let T_i be the set of clockwise neighbors of S_i, for $i = 1, 2$. Formally, $T_i = \{u \in V(X_i) : u = \mathrm{suc}(v), \text{ for } v \in S_i\}$. The subgraph of X_i induced by $S_i \cup T_i$ is a matching such that all the vertices in S_i have the same label and all the vertices in T_i have the same label. Every orientation preserving isomorphism $\psi\colon X_1 \to X_2$ satisfies $\psi(S_1) = S_2$ and $\psi(T_1) = T_2$. We have $V(X_i') = V(X_i) \setminus T_i$. Therefore, the restriction of ψ to $V(X_1')$ is an isomorphism from X_1' to X_2'.

On the other hand, if $\psi'\colon X_1' \to X_2'$ is an isomorphism, then let U_i be the set of clockwise neighbors of S_i in X_i'. We have $\psi'(S_1) = S_2$. Note that we assume that S_1 and S_2 are updated before applying Step 4. Since $|S_i| = |T_i|$, we can easily extend ψ' to an isomorphism $\psi\colon X_1 \to X_2$.

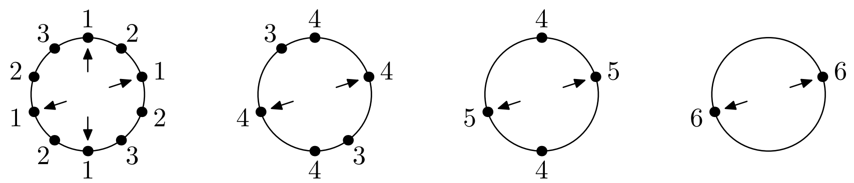


Figure 6.4: Illustration of the reduction procedure for cycles.

We need to execute the algorithm twice to check whether X_1 is isomorphic X_2, or to a 180-degree rotation of X_2. More precisely, $\mathrm{Iso}(X_1, X_2)$ checks map for the existence of map isomorphisms taking the inner face of X_1 onto the inner face X_2, while $\mathrm{Iso}(X_1, X_2^{-1})$ checks the existence of a map isomorphisms taking the inner face of X_1 onto the outer face of X_2. □

Lemma 6.17. *The complexity of the above algorithm is $\mathcal{O}(n)$, where n is the number of vertices of X_1 and X_2.*

Proof. Steps 1–2 take $\mathcal{O}(n)$ time. Each iteration of Steps 3–5 takes $\mathcal{O}(|S_1|+|S_2|)$ time. However, since we remove $|S_1|$ vertices from X_1 and $|S_2|$ vertices from X_2 in each iteration of Step 3, the overall complexity is $\mathcal{O}(n)$. □

Corollary 6.18. *For a labeled cycle X on n vertices, there is a $\mathcal{O}(n)$-time algorithm that computes the generator of the cyclic group of rotations of X.*

The results of this subsection are summarized by the following.

Theorem 6.19. *If $M = (D, R, L)$ is an irreducible spherical map, then the generators of* $\mathrm{Aut}(M)$ *can be computed in time $\mathcal{O}(|D|)$.*

6.5.3 Torus

By definition, the toroidal irreducible maps are uniform face-normal maps. The universal covers of uniform toroidal maps are uniform tilings (infinite maps with finite vertex and face degrees) of the Euclidean plane. There are 12 of such tilings; see [86, page 63]. Their local types are $(3,3,3,3,3,3)$, $(4,4,4,4)$, $(6,6,6)$, $2\times(3,3,3,3,6)$, $(3,3,3,4,4)$, $(3,3,4,3,4)$, $(3,4,6,4)$, $(3,6,3,6)$, $(3,12,12)$, $(4,6,12)$, and $(4,8,8)$. One type occurs in two forms, one is the mirror image of the other. Each of these tilings T gives rise to an infinite family of toroidal uniform maps as follows. It is well-known that $\mathrm{Aut}^+(T)$ is isomorphic either to the triangle group $\Delta(4,4,2)$ or to $\Delta(6,3,2)$. Each of these contains an infinite subgroup H of translations generated by two shifts. Every finite uniform toroidal map of the prescribed local type can be constructed as the quotient T/K, where K is a subgroup of H of finite index.

From uniform face-normal toroidal maps to homogeneous maps. We show that each of the uniform maps can be reduced to one of the two homogeneous types $\{4,4\}$ and $\{6,3\}$, while preserving the automorphism group.

Lemma 6.20. *For every labeled uniform toroidal map M, there is a labeled homogeneous map M' of type $\{4,4\}$ or $\{6,3\}$ such that* $\mathrm{Aut}^+(M) \cong \mathrm{Aut}^+(M')$. *Moreover M' can be constructed in linear time.*

Proof. The idea of the proof is to take the dual M^* of M and apply the reductions defined in Section 6.4. It is straightforward to check by a case-by-case analysis that in each case M^* is reduced to a homogeneous map. □

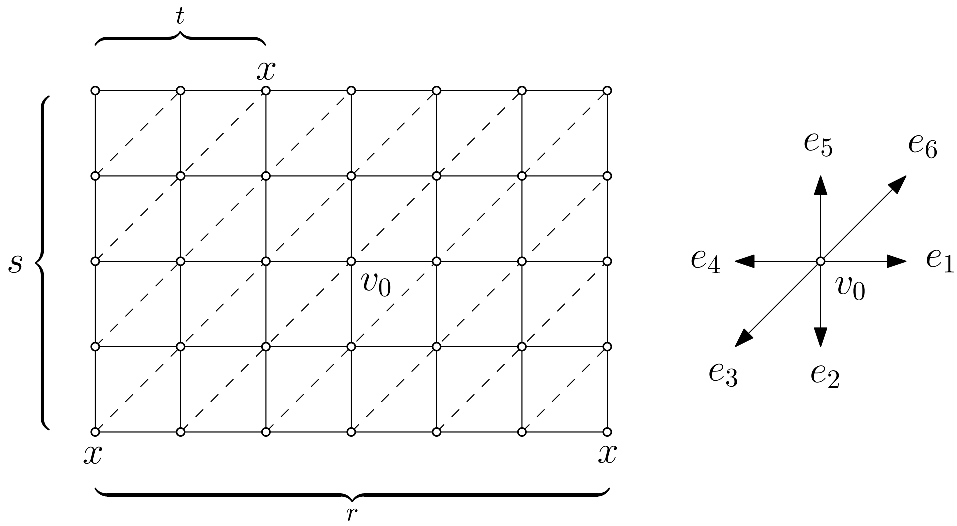


Figure 6.5: The toroidal triangulation $T(6,4,2)$. Its parameter list is the cyclic sequence of parameters (6,4,2), (12,2,6), (12,2,4), (6,4,4), (12,2,6), (12,2,10). The stabilizers of the automorphism group action are trivial and its full automorphism group is isomorphic to $\mathbb{Z}_6 \times \mathbb{Z}_4$.

Homogeneous maps on torus. First, we describe how to compute the generators of the automorphism group of unlabeled homogeneous toroidal maps. There are three possible homogeneous types of such maps: $\{6,3\}$, $\{4,4\}$, and $\{3,6\}$. They admit a simple description in terms of three integer parameters r, s, t where $0 \le t < r$. The 4-regular quadrangulation $Q(r,s,t)$ is obtained from the $(r+1) \times (s+1)$ grid with underlying graph $P_{r+1} \square P_{s+1}$ (the Cartesian product of paths on $r+1$ and $s+1$ vertices) by identifying the "leftmost" path of length s with the "rightmost" one (to obtain a cylinder) and identifying the bottom r-cycle of this cylinder with the top one after rotating the top clockwise for t edges. In other words, the quadrangulation $Q(r,s,t)$ is the quotient of the integer grid $\mathbb{Z} \square \mathbb{Z}$ determined by the equivalence relation generated by all pairs $(x,y) \sim (x+r,y)$ and $(x,y) \sim (x+t,y+s)$.

This classification can be derived by considering appropriate fundamental polygon of the universal cover (which is isomorphic to the tessellation of the plane with unit squares). This structure was known to geometers (Coxeter and Moser [45]). In graph theory, this was observed by Altschuler [3]; several later works do the same (e.g. [154]). Our notation comes from Fisk [66], who only considered 6-regular triangulations. The 6-regular triangulation $T(r,s,t)$ is obtained from $Q(r,s,t)$ by adding all diagonal edges joining (x,y) with $(x+1,y+1)$. And the 3-regular hexangulations $H(r,s,t)$ of the torus are just duals of triangulations $T(r,s,t)$.

The parameters r, s, t depend on the choice of one of the edges incident with a chosen reference vertex v_0. Let us describe the 6-regular case $T(r,s,t)$ first. Let the clockwise order of the edges around v_0 be $e_1, e_2, \ldots, e_6$. We start with e_1 and take the straight-ahead walk (when we arrive to a vertex u using an edge e, we continue the walk with the opposite edge in the local rotation around u). When we come back for the first time at the vertex we have already traversed, it can be shown that this vertex must be v_0 and that we arrive through the edge e_4, which is opposite to the initial edge e_1. This way the straight-ahead walk closes up into a straight-ahead cycle $C = v_0 v_1 \dots v_{r-1} v_0$. We let r be the length of this cycle. Now, let us start a straight-ahead walk from v_0 with the initial edge e_2. Let s be the number of steps on this walk until we reach a vertex, say v_t ($0 \le t < r$), on the cycle C, for the first time. This determines the three parameters r, s, t and it can be shown that the map is isomorphic to the map $T(r, s, t)$ described above. The 4-regular case is the same except that we do not have the directions of the edges e_3 and e_6. See Figure 6.5 for an example. In particular, this proves the well-known fact that $T(r, s, t)$ is vertex-transitive, for the abelian group $\langle a, b \rangle$ such that $ra = 0$ and $ta = sb$.

Labeled homogeneous maps on the torus. We give an algorithm that computes the generators of the automorphism groups of a labeled homogeneous toroidal map M of type $\{4, 4\}$ or $\{3, 6\}$. For technical reasons, we assume that the map M is vertex-labeled instead of dart-labeled. This transformations can be done easily by, for a given vertex, encoding the lables of the outgoing darts into the vertex. The following lemma describes some important properties of $\mathrm{Aut}^+(M)$.

Lemma 6.21 ([152]). *Let M be a toroidal map of type $\{4, 4\}$ or $\{6, 3\}$. The orientation-preserving automorphism group of a labeled map M is a semidirect product $T \rtimes H$, where T is a direct product of two cyclic groups, and $|H| \le 6$. Moreover, the action of T is regular on the vertices of M.*

Since the order of H is bounded by a constant, it takes linear time to check whether every element of H is a label-preserving automorphism. The main difficulty is to find T. The subgroup T is generated by α and β, where α is the horizontal, and β is the vertical shift by the unit distance. Now the meaning of the parameters r, s, t is the following: $|\alpha| = r$, $\alpha^t = \beta^s$, and s is the least power of β such that $\beta^s \in \langle \alpha \rangle$. The following lemma shows that T can always be written as a direct product of two cyclic groups.

Lemma 6.22. *There exists δ and γ such that $T = \langle \delta \rangle \times \langle \gamma \rangle$. Moreover, δ and γ can be computed in time $\mathcal{O}(rs)$.*

Proof. Using the Smith Normal Form, we have that $T \cong \mathbb{Z}_m \times \mathbb{Z}_n$, where $m = \gcd(r, t, s)$ and $n = rs / \gcd(r, t, s)$. Since t divides r, we have $m = \gcd(t, s)$. The respective generators of T can be chosen to be $\delta = \alpha^{\frac{t}{m}} \beta^{\frac{-s}{m}}$ and $\gamma = \beta^{\frac{t}{m}}$. $\square$

Lemma 6.22 can be viewed as a transformation of the shifted grid $\mathcal{G}$ to the orthogonal grid $\mathcal{G}^\perp$. Note that the underlying graph may change, but both $\mathcal{G}$ and $\mathcal{G}^\perp$ are Cayley graphs based on the group T, therefore, the vertex-labeling naturally transfers. Thus, we may assume that $t = 0$ and $T = \langle \alpha \rangle \times \langle \beta \rangle \cong \mathbb{Z}_r \times \mathbb{Z}_s$. We need to compute generators of the label-preserving subgroup of T.

Subgroups of $\mathbb{Z}_r \times \mathbb{Z}_s$. From now on, we assume that we are given a *cyclic orthogonal grid* $\mathcal{G}$ of size rs, which is graph with vertices identified with $(i, j) \in G$, where $G = \mathbb{Z}_r \times \mathbb{Z}_s$. For every (i, j), there is an edge between (i, j) and $(i + 1 \mod r, j)$, and between (i, j) and $(i, j + 1 \mod s)$. Moreover, we are given an integer-labeling ℓ of the vertices of $\mathcal{G}$. Clearly, $\mathcal{G}$ determines the *ℓ-preserving subgroup* H of G, namely

$$H = \{(x, y) : \forall (i, j) \in G, \ell(i, j) = \ell(i + x, j + y)\}.$$

The goal is to find the generators of H in time $\mathcal{O}(rs)$.

We give a description of any subgroup of the direct product of G that is suitable for our algorithm. First, we define four important mappings. The two *projections* $\pi_1 \colon G \to \mathbb{Z}_r$ and $\pi_2 \colon G \to \mathbb{Z}_s$ are defined by $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$, respectively. The two *inclusions* $\iota_1 \colon \mathbb{Z}_r \to G$ and $\iota_2 \colon \mathbb{Z}_s \to G$ are defined by $\iota_1(x) = (x, 0)$ and $\iota_2(y) = (0, y)$, respectively.

Lemma 6.23. *Let $G = \mathbb{Z}_r \times \mathbb{Z}_s$ for $r, s \geq 1$, and let H be a subgroup of G. Then there are $a, c \in \mathbb{Z}_r$ and $b \in \mathbb{Z}_s$ such that*

$$H = \{(ia + jc, jb) : i, j \in \mathbb{Z}\} = \langle (a, 0), (c, b) \rangle,$$

where $\langle a \rangle = \iota_1^{-1}(H)$, $\langle b \rangle = \pi_2(H)$, and $c < a$ is the minimum integer such that $(c, b) \in H$.

Proof. Note that $\iota_1^{-1}(H)$ is a subgroup of $\mathbb{Z}_r$, i.e., there is $a \in \mathbb{Z}_r$ such that $\langle a \rangle = \iota_1^{-1}(H)$. Similarly, $\pi_2(H)$ is a subgroup of $\mathbb{Z}_s$, i.e., there is $b \in \mathbb{Z}_s$ such that $\langle b \rangle = \pi_2(H)$. Finally, let $c \in \mathbb{Z}_r$ be minimum such that $(c, b) \in H$. We prove that $H = \{(ia + jc, jb) : i, j \in \mathbb{Z}\}$.

Clearly, for every $i, j \in \mathbb{Z}$, we have

$$(ia + jc, jb) = (ia, 0) + (jc, jb) = i(a, 0) + j(c, b).$$

By the definition ι_1 and by the definition of c, we have $(a, 0), (c, b) \in H$, and hence $i(a, 0) + j(c, b) \in H$.

On the other hand, let $(x, y) \in H$. Then $\pi_2(x, y) = y \in \mathbb{Z}_s$, hence there is j such that $y = jb$. By the definition of c, we have $(c, b) \in H$, and therefore, $(x - jc, 0) = (x, y) - j(c, b) \in H$. By the definition of ι_1, we have $\iota_1^{-1}(x - jc, 0) = x - jc \in \mathbb{Z}_r$. There exits i such that $x - jc = ia$. We obtain $(x, y) = (ia + jc, jb)$.

Finally, we show that $c < a$. Dividing c by a, we get $c = ka + r$, for some $k \geq 0$ and $0 \leq r < a$. However,

$$(r, b) = (c - ka, b) = (c, b) - k(a, 0) \in H.$$

Thus, by minimality of c, we obtain $c < a$. $\square$

This description suggests an algorithm to find the generators of the given subgroup H of $\mathbb{Z}_r \times \mathbb{Z}_s$. In our setting, the subgroup H is given on the input by a labeling function ℓ, defined on the vertices of the $r \times s$ orthogonal grid. The subgroup H is the ℓ-preserving subgroup of $\mathbb{Z}_r \times \mathbb{Z}_s$.

To compute the generators of H, it suffices, by Lemma 6.23, to determine $a, c \in \mathbb{Z}_r$ and $b \in \mathbb{Z}_s$ such that $\langle a \rangle = \iota_1^{-1}(H)$, $\langle b \rangle = \pi_2(H)$, and c is the smallest integer such that $(c, b) \in H$. Then $H = \langle (a, 0), (c, b) \rangle$.

Lemma 6.24. *There is an $\mathcal{O}(rs)$-time algorithm which computes the integers a, b, c such that $\iota_1^{-1}(H) = \langle a \rangle$, $\pi_2(H) = \langle b \rangle$ and $c < a$ is the smallest integer such that $(c, b) \in H$.*

Proof. First, we compute a in time $\mathcal{O}(rs)$. Let $X_1, \ldots, X_s$ be the horizontal cycles of length r. For each X_j, we compute in time $\mathcal{O}(r)$ the integer r_j such that $\mathrm{Aut}(X_j) \cong \mathbb{Z}_{r_j}$. We put $a := r/r'$, where $r' = \gcd(r_1, \ldots, r_s)$. We need to argue that $\iota_1^{-1}(H) = \langle a \rangle$.

Let $(x, 0) \in H$. There exist integers i_j, for $j = 1, \ldots, s$, such that $x = i_j a_j = i_j r / r_j$. We put $i := \gcd(i_1, \ldots, i_s)$. Then $x = ir/r' = ia \in \langle a \rangle$.

Conversely, let $x = ia$. By the definition of a, we have $(a, 0) \in H$. Then for every x', y', we have

$$\ell(x' + x, y') = \ell(x' + ia, y') = \ell(x' + (i-1)a, y') = \cdots = \ell(x' + a, y') = \ell(x', y'),$$

i.e., $(x, 0) \in H$.

Before, dealing with b and c, we first reduce the problem to a special case, where we have a grid $\mathcal{G}'$ satisfying that the labeling ℓ of any horizontal cycle X'_j of $\mathcal{G}'$ is a rotation of the labeling of the cycle X'_0. We do this as follows. For a horizontal cycle X_j of $\mathcal{G}$ let Σ_j denote the string

$$\ell((0, j)), \ldots, \ell((r-1, j)).$$

We say that X_j and $X_{j'}$ are equivalent if Σ_j is a cyclic rotation of $\Sigma_{j'}$. We assign and integer label to every X_j such that X_j and $X_{j'}$ have the same label if and only if they equivalent. This defines an auxiliary labeled cycle, for which we compute the integer s' such that its automorphism group is isomorphic to $\mathbb{Z}_{s'}$. We define new grids $\mathcal{G}'_i$ consisting of cycles $X_{0+i}, X_{\hat{s}+i}, X_{2\hat{s}'+i}, \ldots, X_{(s'-1)\hat{s}+i}$, for $\hat{s} = s/s'$ and $i = 0, \ldots, s'-1$. Each $\mathcal{G}'_i$ is a grid of size $r\hat{s}$.

Now, for every fixed i, we compute b_i and c_i such that $H_i = \langle (a, 0), (c_i, b_i) \rangle$ is the automorphism group of $\mathcal{G}'_i$ in time $\mathcal{O}(r\hat{s})$. Since $\langle b_i \rangle$ is a subgroup of $\mathbb{Z}_{\hat{s}}$, we may assume that b divides $\hat{s}$. Given b_i, there exits a unique $c_i < a$ such that $\ell(c_i, b_i) = \ell(0, 0)$. Thus, it is possible to identify the set of candidate pairs (c_i, b_i) in time $\mathcal{O}(r\hat{s})$. Finally, the group $H = \bigcap_{i=0}^{s'} H_i$ is the automorphism group of $\mathcal{G}$. To compute b and c such that $H = \langle (c, b) \rangle$, we put $b = \mathrm{lcm}(b_0, \ldots, b_{s'})$ and $c = \mathrm{lcm}(c_0, \ldots, c_{s'})$.

Given (c_i, b_i), we claim that it can be checked in time $\mathcal{O}(\hat{s})$ whether the group $\langle (c_i, b_i) \rangle$ is ℓ-preserving in $\mathcal{G}'_i$. Consider the vectors

$$v_j = (\ell(jc_i, jb_i), \ell(jc_i, jb_i + 1), \ldots, \ell(jc_i, (j+1)b_i - 1)),$$

for $j = 0, \ldots, \hat{s}/b_i$. Now, the group $\langle (c_i, b_i) \rangle$ is ℓ-preserving in $\mathcal{G}'_i$ if and only if all these vectors are equal. To verify this this, we need $\mathcal{O}(b\hat{s}/b) = \mathcal{O}(\hat{s})$ comparisons.

The number of all candidate pairs is at most the number of divisors of $\mathcal{O}(\hat{s})$. Thus the total number of comparisons is

$$\sum_{d|\hat{s}} \mathcal{O}(\hat{s}) = \hat{s} \sum_{d|\hat{s}} \mathcal{O}(1) = \mathcal{O}(\hat{s}^2) = \mathcal{O}(r\hat{s}).$$

The last equality holds if we assume that $s \leq r$, which is always possible without loss of generality. Moreover, we do this for every $i = 0, \ldots, s'-1$, thus, the total complexity is $\mathcal{O}(s' r\hat{s}) = \mathcal{O}(s' rs/s') = \mathcal{O}(rs)$. $\square$

The results of this subsection are summarized by the following.

Theorem 6.25. *If $M = (D, R, L, \ell)$ is a uniform face-normal labeled toroidal map, then the generators of* $\mathrm{Aut}(M)$ *can be computed in time $\mathcal{O}(|D|)$.*

6.6 Non-orientable surfaces

For a map M on a non-orientable surface S, we reduce the problem of computing the generators of $\mathrm{Aut}(M)$ to the problem of computing the generators of $\mathrm{Aut}^+(\widetilde{M})$, for some orientable map $\widetilde{M}$. In particular, the map $\widetilde{M}$ is the antipodal double cover of M.

Given a map $M = (F, \lambda, \rho, \tau)$ on a non-orientable surface of genus γ, we define the antipodal double cover $\widetilde{M} = (D, R, L)$ by setting $D := F$, $R := \rho\tau$, and $L := \tau\lambda$. Since M is non-orientable, we have $\langle R, L\rangle = \langle \lambda, \rho, \tau\rangle$, so $\langle R, L\rangle$ is transitive and $\widetilde{M}$ is well-defined. For more details on this construction see [131]. We note that $\widetilde{\chi} = 2\chi$, where $\widetilde{\chi}$ and χ is the Euler characteristic of the underlying surface of $\widetilde{M}$ and M, respectively.

Lemma 6.26. *We have* $\mathrm{Aut}(M) \le \mathrm{Aut}^+(\widetilde{M})$*.*

Proof. Let $\varphi \in \mathrm{Aut}(M)$. Then we have $R^\varphi = (\rho\tau)^\varphi = \rho^\varphi\tau^\varphi = \rho\tau = R$ and $L^\varphi = (\tau\lambda)^\varphi = \tau^\varphi\lambda^\varphi = \tau\lambda = L$. □

Lemma 6.27. *We have* $\mathrm{Aut}(M) = \{\varphi \in \mathrm{Aut}^+(\widetilde{M}) : \varphi\tau = \tau\varphi\}$*.*

Proof. Let $\varphi \in \mathrm{Aut}^+(\widetilde{M})$. We have $\varphi R\varphi^{-1} = R$ and $\varphi L\varphi^{-1} = L$. By plugging in $R = \rho\tau$ and $L = \tau\lambda$, we obtain

$$\varphi(\rho\tau)\varphi^{-1} = \rho\tau \quad \text{and} \quad \varphi(\tau\lambda)\varphi^{-1} = \lambda\tau.$$

From there, by rearranging the left-hand sides of the equations, we get

$$(\varphi\rho\varphi^{-1})(\varphi\tau\varphi^{-1}) = \varphi(\rho\tau)\varphi^{-1} = \rho\tau \quad \text{and} \quad (\varphi\tau\varphi^{-1})(\varphi\lambda\varphi^{-1}) = \varphi(\tau\lambda)\varphi^{-1} = \tau\lambda.$$

Finally, we obtain

$$\varphi\rho\varphi^{-1} = \rho\tau(\varphi\tau\varphi^{-1}) \quad \text{and} \quad \varphi\lambda\varphi^{-1} = (\varphi\tau\varphi^{-1})\tau\lambda.$$

If $\varphi \in \mathrm{Aut}(M)$, then, in particular, it commutes with τ. On the other hand, if φ commutes with τ, then the last two equations imply that it also must commute with ρ and λ, i.e., $\varphi \in \mathrm{Aut}(M)$. □

The previous lemma suggest an approach for computing the generators of the automorphism group of M.

Lemma 6.28. *Let $M = (F, \lambda, \rho, \tau)$ be a map on a non-orientable surface of genus $\gamma > 2$. Then it is possible to compute the generators of* $\mathrm{Aut}(M)$ *in time* $f(\gamma)|F|$*.*

Proof. First, we construct $\widetilde{M}$ in time $\mathcal{O}(|F|)$. Using the algorithm from Sections 6.4 and 6.5, we construct the associated labeled uniform map $\overline{M}$ and compute the generators of $\mathrm{Aut}^+(\widetilde{M})$.

Suppose that $\gamma > 2$. By Riemann-Hurwitz theorem, we have $|\mathrm{Aut}^+(\overline{M})| \le 84(g-1)$, where $g = \gamma - 1$. For each $\overline{\varphi} \in \mathrm{Aut}^+(\overline{M})$, we construct the unique extension $\varphi \in \mathrm{Aut}^+(\widetilde{M})$ and check whether $\varphi\tau\varphi^{-1} = \tau$ in time $\mathcal{O}(|F|)$. The previous Lemma 6.27 states that $\mathrm{Aut}(M)$ consists exactly of those φ. □

To proceed with maps on the projective plane and Klein bottle, we define the *action diagram* for a permutation group $G \leq \mathrm{Sym}(\Omega)$ with a generating set S. To every generator $g \in S$ we assign a unique color c_g. The action diagram $\mathcal{A}(G)$ of G is an edge-colored oriented graph with the vertex set Ω. There is an oriented edge $x \to y$ of color c_g if and only $gx = y$. We first prove the following technical lemma.

Lemma 6.29. *Let $C \leq \mathrm{Sym}(\Omega)$ be a semiregular cyclic group and let $\tau \in \mathrm{Sym}(\Omega)$ be fixed-point-free involution. Then there is an algorithm which finds the subgroup $L \leq C$ centralizing τ in time $\mathcal{O}(|\Omega|)$.*

Proof. Let $C = \langle \varphi \rangle$, where $|\varphi| = n$. It is sufficient to find the smallest $m > 0$ such that φ^m commutes with τ. Then the group $A = \langle \tau, \varphi^m \rangle$ is abelian of order n/m or $2n/m$. Since both φ and τ are fixed-point-free permutations, the orbits of A are either of size n/m, or of size $2n/m$ and the respective action diagrams are either Möbius ladders, or ladders (prisms).

Step 1. We first determine the largest cyclic subgroup $K \leq \langle \varphi \rangle$ satisfying the property that the orbits of $\langle K, \tau \rangle$ have size $|K|$, or $2|K|$. If this is the case, then τ matches an orbit O of K to a unique orbit $\tau(O)$, where $\tau(O) = O$ may happen.

The group $\langle \varphi \rangle$ acts semiregularly on Ω and hence there are exactly $|\Omega|/n$ orbits $O_1, \dots, O_{|\Omega|/n}$ of size n. We find the smallest $m' > 0$ such that for every $i = 1, \dots, |\Omega|/n$, and every $x \in O_i$, there is j such that $\{\tau(x), \tau\varphi^{m'}(x)\} \subseteq O_j$. Let $X_i := \mathcal{A}(\langle \varphi \rangle, O_i)$, for $i = 1, \dots, |\Omega|/n$. Note that each X_i is an oriented cycle, a Cayley graph of a cyclic group. We further label the vertices of X_i such that $\ell_i(x) = j$ if $\tau x \in O_j$.

For each labeled cycle X_i, we use the algorithm of Section 6.5 to compute the integer k_i such that $\mathrm{Aut}^+(X_i) \cong \mathbb{Z}_{k_i}$. We have $\mathrm{Aut}^+(X_i) = \langle \varphi^{m_i} \rangle$, where $m_i := n/k_i$, for $i = 1, \dots, |\Omega|/n$. Each φ^{m_i} is label-preserving on X_i, i.e., for every $x \in O_i$, $\ell(x) = \ell\varphi^{m_i}(x)$. By the definition of ℓ, the points $\tau(x)$ and $\tau\varphi^{m_i}(x)$ belong to the same orbit of φ. Clearly,

$$\langle \varphi^{m'} \rangle = \bigcap_{i=1}^{|D|/n} \langle \varphi^{m_i} \rangle.$$

This implies that $m' = n/k$, where $k = \gcd(k_1, \dots, k_{|\Omega|/n})$.

Step 2. We find m such that $L = \langle \varphi^m \rangle$ commutes with τ. Clearly, L is a subgroup of $K = \langle \alpha \rangle$, where $\alpha = \varphi^{m'}$. Let $O'_1, \dots, O'_{|\Omega|/k}$ be the orbits of K, and we define $X'_i := \mathcal{A}(\langle \varphi^{m'} \rangle, O'_i)$, for $i = 1, \dots, |D|/k$. From the definition of m', it follows that $\tau(O'_i) = O'_j$, for some j. In other words, τ defines a perfect matching between the points of O'_i and O'_j. We distinguish two cases.

- $\tau(O'_i) = O'_i$. We identify the points of O_i with $\mathbb{Z}_k = \{0, \dots, k-1\}$. For a point $x \in O_i$, with $|x - \tau(x)| = k/2$, we define $\ell(x) := k/2$ and $\ell\tau(x) := k/2$. For a point $x \in O_i$, with $|x - \tau(x)| < k/2$, we define $\ell(x) := +|x - \tau(x)|$ and $\ell\tau(x) := -|x - \tau(x)|$.

- $\tau(O'_i) = O'_j$, for $i \neq j$. We identify the points of $O'_i \cup O'_j$ with $\mathbb{Z}_k \cup \mathbb{Z}_k = \{0, \dots, k-1\} \cup \{0, \dots, k-1\}$ as follows. First, we identify O'_i with $\mathbb{Z}_k$. Then, for $x \in O'_i$ identified with 0, we identify $\tau(x)$ with 0, and extend uniquely using the action of $(\mathbb{Z}_k, +)$. Similarly, as in the previous case we define a labeling ℓ of O_i and O_j. For a point $x \in O_i$, with $|x - \tau(x)| = k/2$, we define $\ell(x) := k/2$ and $\ell\tau(x) := k/2$. For a point $x \in O_i$, with $|x - \tau(x)| < k/2$, we define $\ell(x) := +|x - \tau(x)|$ and $\ell\tau(x) := -|x - \tau(x)|$.

We use the algorithm of Section 6.5 to compute the integer k'_i such that $\mathrm{Aut}^+(X_i) \cong \mathbb{Z}_{k'_i}$, for $i = 1, \ldots, |\Omega|/k$. Let $m'_i := k/k'_i$. By the definition of ℓ, we have, for every $x \in O_i$,

$$\begin{aligned}\ell(x) &= \ell\alpha^{m'_i}(x),\\ \pm|x - \tau(x)| &= \pm|\alpha^{m'_i}(x) - \tau\alpha^{m'_i}(x)|.\end{aligned}$$

This exactly means that $\tau\alpha^{m'_i}(x) = \alpha^{m'_i}\tau(x)$. Finally, $L = \langle\varphi^m\rangle$ is the intersection

$$\bigcap_{i=1}^{|\Omega|/k} \langle\alpha^{m'_i}\rangle.$$

This implies that $m = k/\gcd(k'_1, \ldots, k'_{|\Omega|/k})$.

Both Step 1 and Step 2 can be computed in linear time. In Step 1, we have $|\Omega|/n$ cycles of size n, The time spent by the algorithm of Section 6.5 on each of the cycles is $\mathcal{O}(|\Omega|/n)$. The greatest common divisor of $|\Omega|/n$ numbers in the interval $[1, |\Omega|]$ can be computed in time $\mathcal{O}(|\Omega|)$. Thus the overall complexity of Step 1 is $\mathcal{O}(|\Omega|)$. Note that exactly the same argument works for Step 2. This completes to proof for $\mathrm{Aut}(\widetilde{M}) \cong \mathbb{Z}_n$. □

Lemma 6.30. *For a map $M = (F, \lambda, \rho, \tau)$ on the projective plane, it is possible to compute the generators of* $\mathrm{Aut}(M)$ *in time* $\mathcal{O}(|F|)$.

Proof. Note that in this case $\widetilde{M}$ is a spherical map. If the reduced map $\overline{M}$ from $\widetilde{M}$ does not belong to one of the infinite families, then we may use the same approach as in the case when $\gamma > 2$. Otherwise, $\overline{M}$ is one of the following: bouquet, dipole, cycle, prism, antiprism. We only deal with the cycle, since the other cases are reduced to it. If $\overline{M}$ is a cycle, then we have that either $\mathrm{Aut}^+(\overline{M}) \cong \mathrm{Aut}^+(\widetilde{M}) \cong \mathbb{D}_n$ or $\mathrm{Aut}^+(\overline{M}) \cong \mathrm{Aut}^+(\widetilde{M}) \cong \mathbb{Z}_n$, for some n.

First, suppose that $\mathrm{Aut}^+(\widetilde{M}) \cong \mathbb{Z}_n$. Let φ be the generator of $\mathrm{Aut}^+(\widetilde{M})$, i.e., $\langle\varphi\rangle = \mathrm{Aut}^+(\widetilde{M})$ and $\varphi^n = \mathrm{id}$. By Lemma 6.26, $\mathrm{Aut}(M) \leq \mathrm{Aut}^+(\widetilde{M})$ and therefore, $\mathrm{Aut}(M)$ is also a cyclic group. By Lemma 6.27, it is sufficient to find the smallest $m > 0$ such that φ^m commutes with τ. By Lemma 6.29, this can be done in time $\mathcal{O}(|D|)$.

Suppose that $\mathrm{Aut}(\widetilde{M}) \cong \mathbb{D}_n$. It is known that $\mathbb{D}_n$ can be written as the semidirect product[4] $\mathbb{Z}_n \rtimes \mathbb{Z}_2$, i.e., $\mathbb{D}_n$ has two generators, one of order n and the other of order 2. There are $\varphi, \psi \in \mathrm{Aut}^+(\widetilde{M})$ such that $\varphi^n = \mathrm{id}$, $\psi^2 = \mathrm{id}$, and $\langle\varphi, \psi\rangle = \mathrm{Aut}^+(\widetilde{M})$. By Lemma 6.26, $\mathrm{Aut}(M) \leq \mathrm{Aut}^+(\widetilde{M})$, i.e., there is k dividing n such that $\mathrm{Aut}(M) \cong \mathbb{D}_k$ or $\mathrm{Aut}(M) \cong \mathbb{Z}_k$. To check whether $\psi \in \mathrm{Aut}(M)$, by Lemma 6.27, it suffices to check if $\psi\tau\psi^{-1} = \tau$, which can be done in linear time. To investigate the cyclic subgroup $\langle\psi\rangle$, we proceed as in the cyclic case above. □

Lemma 6.31. *Let $M = (F, \lambda, \rho, \tau)$ be a map on the Klein bottle. Then it is possible to compute the generators of* $\mathrm{Aut}(M)$ *in time* $\mathcal{O}(|F|)$.

[4] A group G is a semidirect product of N and H if $N, H \leq G$, $N \cap H = \{1\}$, and N is a normal subgroup of G. We write $G = N \rtimes H$.

Proof. We form the antipodal double-cover $\widetilde{M} = (D, R, L)$ of M, which is in this case a toroidal map, and compute the generators of $G = \mathrm{Aut}^+(\widetilde{M}) = T \rtimes G_v$, where $T = \langle\varphi\rangle \times \langle\psi\rangle$ and G_v is the vertex-stabilizer with $|G_v| \leq 6$. Further, we assume that $|\varphi| = a$ and $|\varphi| = b$. By Lemma 6.27, we need to determine the subgroup H of G centralized by τ. For $H \cap G_v$, this is done by brute-force, checking the commutativity with τ for every element individually. We show how to find in linear time the generators of $K = H \cap T$. By Lemma 6.29, we find minimal $m > 0$ and $n > 0$ such that φ^m and ψ^n commute with τ.

By definition, every element of $\langle\varphi^m, \psi^n\rangle = \langle\varphi^m\rangle \times \langle\psi^n\rangle$ commutes with τ. Conversely, let $\pi = \varphi^k\psi^\ell \in T$ be such that $\tau\pi\tau^{-1} = \pi$. By plugging in, we get

$$\tau\varphi^k\psi^\ell\tau^{-1} = \tau\varphi\tau^{-1}\tau\psi^\ell\tau^{-1} = \varphi^k\psi^\ell.$$

Since $T = \langle\varphi\rangle \times \langle\psi\rangle$, the last equality holds only if $\tau\varphi^k\tau^{-1} = \varphi^k$ and $\tau\psi^\ell\tau = \psi^\ell$. It follows that $\varphi^k \in \langle\varphi^m\rangle$ and $\psi^\ell \in \langle\psi^n\rangle$, and consequently, $\pi \in \langle\varphi^m, \psi^n\rangle$. □

The results of this subsection are summarized by the following.

Theorem 6.32. *If $M = (F, \lambda, \rho, \tau)$ be a map on a non-orientable surface of genus γ, then the generators of* $\mathrm{Aut}(M)$ *can be computed in time $\mathcal{O}(|F|)$.*

6.7 Complexity of the algorithm and summary

In this section, we investigate the complexity of various parts of our algorithm. We argue that it runs in time linear in the size of the input, i.e., in time $\mathcal{O}(|D_1| + |D_2|)$. We show a representation of the function ℓ such that $\ell(x)$ and $\ell(y)$ can be compared in constant time. We also describe an implementation of the function **Label** that computes the new label in time proportional to the number of its arguments. At the end, we give a summary of the whole algorithm.

Reductions. The data structure to find next reduction is a list of queues, the number of which is bounded by a function of genus. Every time a vicinity of a vertex is modified, it is pushed to the correct queue. At each step we look at the the first non-empty queue.

The only difficulty is with updating the local type for every vertex. If there is a large face f of size $\mathcal{O}(v(M))$ incident to a vertex of small degree, we cannot afford to update the local type of every vertex incident to f, since the degree of f may decrease just by one. To overcome this obstacle we use another trick.

We define the *vertex-face incidence map* $\Gamma(M)$ of M which is a bipartite quadrangular map associated to M. Its vertices are the vertices and centers of faces of M. For every vertex $v \in V(M)$ and face-center $f \in F(M)$ of a face incident to v there is an edge joining v to f. Note that f can be multiply incident to v, for each such incidence there is an edge in $\Gamma(M)$. The map $\Gamma(M)$ can be alternatively obtained as the dual of the medial map. Every reduction easily translates to a reduction in $\Gamma(M)$. We update $\Gamma(M)$ after every elementary reduction. The important property of $\Gamma(M)$ is that if v is a vertex of M, then cyclic vector of degrees of the neighbours of v in $\Gamma(M)$ is exactly the local type of v in M. To update the local type of a vertex after a reduction it suffices to look at its cyclic vector of degrees of its neighbours in Γ.

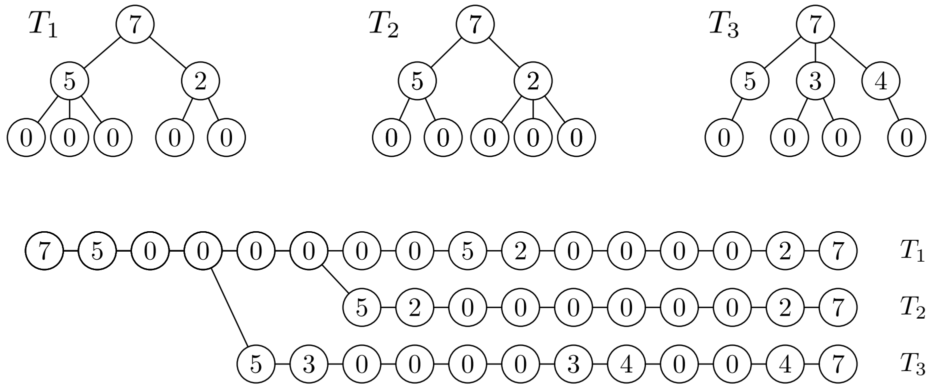


Figure 6.6: Labels represented as planted trees together with the associated prefix tree.

Labels. In Section 6.4, we were using the function ℓ as the labeling of a map M and the injective function $\mathbf{Label}(t, a_1, \dots, a_m)$, where $t \in \mathbb{N}$ denotes the step and every a_i is a label, for constructing new labels.

First, we describe the implementation of labels, i.e., the images of ℓ. Every label is implemented as a *rooted planted tree* with integers assigned to its nodes. A rooted planted tree is a rooted tree embedded in the plane, i.e., by permuting the children of a node we get different trees; see Figure 6.6. Every planted tree with n nodes can be uniquely encoded by a 01-string of length $2n$. Further, we require that the children of every node N have smaller integers than their nodes. This type of tree is also called a *maximum heap*. Such a tree can be uniquely encoded by a string (sequence) of integers.

Now we define **Label**. The integer t represents the current step of the algorithm. At the start, we have $t = 0$ and the map M has constant labeling – every dart is labeled by a one-vertex tree with 0 assigned to its only vertex. Performing a reduction increments s by 1. For labels (rooted planted trees) $a_1, \dots, a_m$, the function $\mathbf{Label}(t, a_1, \dots, a_m)$ constructs a new rooted planted tree with t in the root and the root joined to the roots of $a_1, \dots, a_m$. Clearly this function is injective and can be implemented in the same running time as the corresponding reduction.

Finally, we relabel homogeneous maps by integers. This is necessary mainly for the case when the reductions terminate at cycles since in this case we need to be able to compare labels in constant time. Suppose that we have two homogeneous maps M_1 and M_2 with the corresponding sets of labels $\mathcal{T} = \{T_1, \dots, T_k\}$ and $\mathcal{T}' = \{T'_1, \dots, T'_k\}$. We construct bijections $\sigma\colon \mathcal{T} \to \{1, \dots, k\}$ and $\sigma'\colon \mathcal{T}' \to \{1, \dots, k\}$ such that after replacing T_i by $\sigma(T_i)$ and T'_i by $\sigma'(T'_i)$, we get isomorphic maps. To construct σ and σ', we replace every tree in $\mathcal{T}$ and $\mathcal{T}'$ by a string of integers. Then we find the lexicographic ordering of $\mathcal{T}$ and $\mathcal{T}'$ by constructing two *prefix trees* (sometimes in literature called *trie*); see Figure 6.6. This lexicographic ordering gives the bijections σ and σ'. Finally, we need to check if the pre-images of every i under σ and σ' are the same planted trees, otherwise the maps are not isomorphic. This can be easily implemented in linear time.

Summary of the algorithm. The input of the whole algorithm is a non-oriented map $N = (F, \lambda, \rho, \tau)$. First, we compute its Euler characteristic by performing a breadth-first search. Then, we test whether it is orientable using Theorem 6.7.

If N is orientable, then we construct the associated oriented maps $M = (D, R, L)$ and $M^{-1} = (D, R^{-1}, L)$. We use the algorithms from Section 6.4 and 6.5 to compute $\mathrm{Aut}^+(M)$ and to find any $\varphi \in \mathrm{Iso}^+(M, M^{-1})$. The group $\mathrm{Aut}(N)$ is reconstructed from $\mathrm{Aut}^+(M)$ and φ.

If N is not orientable, then we construct the associated antipodal double-cover $\widetilde{M} = (F.\rho\tau, \tau\lambda)$, which is an oriented map, and use again algorithms from Section 6.4 and 6.5 to compute the generators of $\mathrm{Aut}^+(\widetilde{M})$. Finally, we use the algorithms from Section 6.6 to find the subgroup of $\mathrm{Aut}^+(\widetilde{M})$ which is equal to $\mathrm{Aut}(N)$.

Finding all isomorphisms between two maps. Our algorithm can be easily adapted for the problem of finding all isomorphisms between two maps M_1 and M_2, using the relation $\mathrm{Iso}(M_1, M_2) = \mathrm{Aut}(M_1)\varphi$, where $\varphi\colon M_1 \to M_2$ is an arbitrary isomorphism. To compute φ, we find the reduction for the maps M_1, M_2, and M_2^{-1}. Testing isomorphism of the reduced maps can be done by a brute-force algorithm if they do not belong to an infinite family. Otherwise, the Euler characteristic is non-negative. For the sphere or the projective plane, we apply the algorithm of Section 6.5. For the torus and the Klein bottle, the described algorithms can be easily adapted.

Bibliography

[1] Deniz Agaoglu and Petr Hlinený. Isomorphism problem for s_d-graphs. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 4:1–4:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company, 1974.

[3] A. Altshuler. Construction and enumeration of regular maps on the torus. *Discrete Math.*, 4:201–217, 1973.

[4] Vikraman Arvind, Bireswar Das, Johannes Köbler, and Seinosuke Toda. Colored hypergraph isomorphism is fixed parameter tractable. *Algorithmica*, 71(1):120–138, 2015.

[5] Vikraman Arvind, Roman Nedela, Ilia Ponomarenko, and Peter Zeman. Testing isomorphism of chordal graphs of bounded leafage is fixed-parameter tractable. *CoRR*, abs/2107.10689, 2021.

[6] Albert Atserias and Elitza N. Maneva. Sherali-adams relaxations and indistinguishability in counting logics. *SIAM J. Comput.*, 42(1):112–137, 2013.

[7] Gilles Audemard, Saïd Jabbour, and Lakhdar Sais. Symmetry breaking in quantified boolean formulae. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2262–2267, 2007.

[8] L. Babai. Automorphism groups of planar graphs II. In *Infinite and finite sets (Proc. Conf. Keszthely, Hungary, 1973) Bolyai-North-Holland*, pages 29–84, 1975.

[9] L. Babai. Vertex-transitive graphs and vertex-transitive maps. *Journal of graph theory*, 15(6):587–627, 1991.

[10] L. Babai. Automorphism groups, isomorphism, reconstruction. In *Handbook of combinatorics (vol. 2)*, pages 1447–1540. MIT Press, 1996.

[11] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016.

[12] László Babai. Canonical form for graphs in quasipolynomial time: preliminary report. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1237–1246. ACM, 2019.

[13] László Babai, D. Yu. Grigoryev, and David M. Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 310–324. ACM, 1982.

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

[15] J. Bang-Jensen, J. Huang, and X. Zhu. Completing orientations of partially oriented graphs. *CoRR*, abs/1509.01301, 2015.

[16] D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM Journal on Computing*, 17(1):53–76, 1988.

[17] N. L. Biggs and A. T. White. *Permutation groups and combinatorial structures*, volume 33. Cambridge University Press, 1979.

[18] G. Birkhoff. On groups of automorphisms. *Rev. Un. Mat. Argentina*, 11:155–157, 1946.

[19] Miklós Biro, Mihály Hujter, and Zs Tuza. Precoloring extension. i. interval graphs. *Discrete Mathematics*, 100(1-3):267–279, 1992.

[20] 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.

[21] Kellogg S Booth. Lexicographically least circular substrings. *Information Processing Letters*, 10(4-5):240–242, 1980.

[22] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

[23] A. Bouchet. Reducing prime graphs and recognizing circle graphs. *Combinatorica*, 7(3):243–254, 1987.

[24] A. Bouchet. Unimodularity and circle graphs. *Discrete Mathematics*, 66(1-2):203–208, 1987.

[25] Neal Brand. Isomorphisms of cyclic combinatorial objects. *Discret. Math.*, 78(1-2):73–81, 1989.

[26] A. Brodnik, A. Malnič, and R. Požar. Lower bounds on the simultaneous conjugacy problem in the symmetric group. In *4th annual Mississippi Discrete Mathematics Workshop*, 2015.

[27] A. Brodnik, A. Malnič, and R. Požar. Fast permutation-word multiplication and the simultaneous conjugacy problem. *arXiv preprint arXiv:1907.07889*, 2019.

[28] R. P. Bryant and D. Singerman. Foundations of the theory of maps on surfaces with boundary. *The Quarterly Journal of Mathematics*, 36(1):17–41, 1985.

[29] Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 612–617. IEEE Computer Society, 1989.

[30] P. J. Cameron. *Permutation groups*, volume 45. Cambridge University Press, 1999.

[31] Maurice Chandoo. Deciding circular-arc graph isomorphism in parameterized logspace. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[32] S. Chaplick, S. Felsner, U. Hoffmann, and V. Wiechert. Grid intersection graphs and order dimension. *Order*, 35(2):363–391, 2018.

[33] S. Chaplick, R. Fulek, and P. Klavík. Extending partial representations of circle graphs. *CoRR*, abs/1309.2399, 2015.

[34] Steven Chaplick, Martin Toepfer, Jan Voborník, and Peter Zeman. On h-topological intersection graphs. In Hans L. Bodlaender and Gerhard J. Woeginger, editors, *Graph-Theoretic Concepts in Computer Science - 43rd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers*, volume 10520 of *Lecture Notes in Computer Science*, pages 167–179. Springer, 2017.

[35] Steven Chaplick and Peter Zeman. Combinatorial problems on h-graphs. *Electron. Notes Discret. Math.*, 61:223–229, 2017.

[36] Steven Chaplick and Peter Zeman. Isomorphism-completeness for H-graphs. *https://kam.mff.cuni.cz/ pizet/gic.pdf*, 2021.

[37] Gang Chen and Ilia Ponomarenko. *Lectures on coherent configurations.* 2019.

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

[39] 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.

[40] 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.

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

[42] Charles J. Colbourn and Kellogg S. Booth. Linear time automorphism algorithms for trees, interval graphs, and planar graphs. *SIAM J. Comput.*, 10(1):203–225, 1981.

[43] J. H. Conway, H. Burgiel, and C. Goodman-Strauss. *The symmetries of things.* CRC Press, 2016.

[44] D. G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A. P. Sprague. Simple linear time recognition of unit interval graphs. *Information Processing Letters*, 55(2):99–104, 1995.

[45] H. S. M. Coxeter and W. O. J. Moser. *Generators and relations for discrete groups*, volume 14. Springer Science & Business Media, 2013.

[46] W.H. Cuningham and J. Edmonds. A combinatorial decomposition theory. *Canad. J. Math.*, 32:734–765, 1980.

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

[48] William H Cunningham. Decomposition of directed graphs. *SIAM Journal on Algebraic Discrete Methods*, 3(2):214–228, 1982.

[49] William H Cunningham and Jack Edmonds. A combinatorial decomposition theory. *Canadian Journal of Mathematics*, 32(3):734–765, 1980.

[50] A. R. Curtis, M. C. Lin, R. M. McConnell, Y. Nussbaum, F. J. Soulignac, J. P. Spinrad, and J. L. Szwarcfiter. Isomorphism of graph classes related to the circular-ones property. *Discrete Mathematics and Theoretical Computer Science*, 15(1):157–182, 2013.

[51] E. Dahlhaus. Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition. *Journal of Algorithms*, 36(2):205–240, 1998.

[52] H. de Fraysseix. Local complementation and interlacement graphs. *Discrete Mathematics*, 33(1):29–35, 1981.

[53] H. de Fraysseix and P. O. de Mendez. On a characterization of gauss codes. *Discrete & Computational Geometry*, 22(2):287–295, 1999.

[54] G. Di Battista and R. Tamassia. Incremental planarity testing. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 436–441. IEEE, 1989.

[55] G. Di Battista and R. Tamassia. On-line graph algorithms with SPQR-trees. In *International Colloquium on Automata, Languages, and Programming*, pages 598–611. Springer, 1990.

[56] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM Journal on Computing*, 25(5):956–997, 1996.

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

[58] Zdenek Dvorák. On recognizing graphs by numbers of homomorphisms. *J. Graph Theory*, 64(4):330–342, 2010.

[59] Sergei Evdokimov, Ilia Ponomarenko, and Gottfried Tinhofer. Forestal algebras and algebraic forests (on a new class of weakly compact graphs). *Discrete Mathematics*, 225(1-3):149–172, 2000.

[60] Sergei Evdokimov, Ilia N. Ponomarenko, and Gottfried Tinhofer. Forestal algebras and algebraic forests (on a new class of weakly compact graphs). *Discret. Math.*, 225(1-3):149–172, 2000.

[61] Sergei Evdokimov, Ilia N. Ponomarenko, and Gottfried Tinhofer. Forestal algebras and algebraic forests (on a new class of weakly compact graphs). *Discret. Math.*, 225(1-3):149–172, 2000.

[62] S. Even and A. Itai. Queues, stacks, and graphs. *Theory of Machines and Computation (Z. Kohavi and A. Paz, Eds.)*, pages 71–76, 1971.

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

[64] J. Fiala, P. Klavík, J. Kratochvíl, and R. Nedela. Algorithmic aspects of regular graph covers with applications to planar graphs. In *ICALP 2014*, volume 8572 of *LNCS*, pages 489–501, 2014.

[65] J. Fiala, P. Klavík, J. Kratochvíl, and R. Nedela. 3-connected reduction for regular graph covers. *CoRR*, abs/1503.06556, 2017.

[66] S. Fisk. Geometric coloring theory. *Advances in Math.*, 24(3):298–340, 1977.

[67] Fedor V. Fomin, Petr A. Golovach, and Jean-Florent Raymond. On the tractability of optimization problems on h-graphs. *Algorithmica*, 82(9):2432–2473, 2020.

[68] M. Fontet. Calcul de centralisateur d'un groupe de permutations. *Bull. Soc. Math. France Mém.,(49-50)*, pages 53–63, 1977.

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

[70] C. P. Gabor, K. J. Supowit, and W. Hsu. Recognizing circle graphs in polynomial time. *J. ACM*, 36(3):435–473, 1989.

[71] 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.

[72] T. Gallai. Transitiv orientierbare graphen. *Acta Math. Hungar.*, 18(1):25–66, 1967.

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

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

[75] 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.

[76] E. Gioan, C. Paul, M. Tedder, and D. Corneil. Practical and efficient circle graph recognition. *Algorithmica*, 69(4):759–788, 2014.

[77] E. Gioan, C. Paul, M. Tedder, and D. Corneil. Practical and efficient split decomposition via graph-labelled trees. *Algorithmica*, 69(4):789–843, 2014.

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

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

[80] Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017.

[81] Martin Grohe. *Descriptive complexity, canonisation, and definable graph structure theory*, volume 47. Cambridge University Press, 2017.

[82] Martin Grohe and Sandra Kiefer. A linear upper bound on the Weisfeiler-Leman dimension of graphs of bounded genus. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 117:1–117:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[83] Martin Grohe, Daniel Neuen, and Pascal Schweitzer. A faster isomorphism test for graphs of small degree. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 89–100. IEEE Computer Society, 2018.

[84] Martin Grohe, Daniel Neuen, Pascal Schweitzer, and Daniel Wiebking. An improved isomorphism test for bounded-tree-width graphs. *ACM Trans. Algorithms*, 16(3):34:1–34:31, 2020.

[85] J. L. Gross and T. W. Tucker. *Topological graph theory*. Dover, 2001.

[86] B. Grünbaum and G. C. Shephard. *Tilings and patterns*. Freeman, 1987.

[87] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In *International Symposium on Graph Drawing*, pages 77–90. Springer, 2000.

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

[89] Zdeněk Hedrlín and Aleš Pultr. On full embeddings of categories of algebras. *Illinois Journal of Mathematics*, 10(3):392–406, 1966.

[90] Christoph M Hoffmann. Subcomplete generalizations of graph isomorphism. *Journal of Computer and System Sciences*, 25(3):332–359, 1982.

[91] J. E. Hopcroft and R. E. Tarjan. A V^2 algorithm for determining isomorphism of planar graphs. *Information Processing Letters*, 1(1):32–34, 1971.

[92] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, 1973.

[93] J. E. Hopcroft and R. Endre. Tarjan. A V log V algorithm for isomorphism of triconnected planar graphs. *J. Comput. Syst. Sci.*, 7(3):323–331, 1973.

[94] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism on planar graphs. In *Proc. 6th Annual ACM Symp. on Theory of Computing.*, 1974.

[95] John E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1974, Seattle, Washington, USA*, pages 172–184. ACM, 1974.

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

[97] G. A. Jones and D. Singerman. Theory of maps on orientable surfaces. *Proceedings of the London Mathematical Society*, 3(2):273–307, 1978.

[98] Vít Kalisz, Pavel Klavík, and Peter Zeman. Circle graph isomorphism in almost linear time. 2019.

[99] K. Kawarabayashi, P. Klavík, B. Mohar, R. Nedela, and P. Zeman. Isomorphisms of maps on the sphere. In *Polytopes and Discrete Geometry*, volume 764 of *Contemporary Mathematics*, pages 125–147. American Mathematical Society, 2021.

[100] Ken-ichi Kawarabayashi, Bojan Mohar, Roman Nedela, and Peter Zeman. Automorphisms and isomorphisms of maps in linear time. In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland*, volume 198 of *LIPIcs*, pages 86:1–86:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[101] Sandra Kiefer and Daniel Neuen. The power of the Weisfeiler-Leman algorithm to decompose graphs. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 45:1–45:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[102] Sandra Kiefer, Ilia Ponomarenko, and Pascal Schweitzer. The weisfeiler-leman dimension of planar graphs is at most 3. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[103] Sandra Kiefer, Ilia Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. *J. ACM*, 66(6):44:1–44:31, 2019.

[104] P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskočil. Extending partial representations of proper and unit interval graphs. *Algorithmica*, 77(4):1071–1104, 2017.

[105] P. Klavík, J. Kratochvíl, Y. Otachi, and T. Saitoh. Extending partial representations of subclasses of chordal graphs. *Theoretical Computer Science*, 576:85–101, 2015.

[106] P. Klavík, J. Kratochvíl, Y. Otachi, T. Saitoh, and T. Vyskočil. Extending partial representations of interval graphs. *Algorithmica*, 2016.

[107] P. Klavík and P. Zeman. Automorphism groups of geometrically represented graphs. In *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 540–553, 2015.

[108] Pavel Klavík, Roman Nedela, and Peter Zeman. Jordan-like characterization of automorphism groups of planar graphs. 2021.

[109] Pavel Klavík and Peter Zeman. Automorphism groups of geometrically represented graphs. *submitted*, 2019.

[110] Johannes Köbler, Sebastian Kuhnert, Bastian Laubner, and Oleg Verbitsky. Interval graphs: Canonical representations in logspace. *SIAM Journal on Computing*, 40(5):1292–1315, 2011.

[111] Johannes Köbler, Sebastian Kuhnert, and Oleg Verbitsky. Helly circular-arc graph isomorphism is in logspace. In *International Symposium on Mathematical Foundations of Computer Science*, pages 631–642. Springer, 2013.

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

[113] T. Krawczyk and B. Walczak. Extending partial representations of trapezoid graphs. In *WG 2017*, Lecture Notes in Computer Science, 2017.

[114] In-Jen Lin, Terry A. McKee, and Douglas B. West. The leafage of a chordal graph. *Discuss. Math. Graph Theory*, 18(1):23–48, 1998.

[115] Min Chih Lin, Francisco J Soulignac, and Jayme L Szwarcfiter. A simple linear time algorithm for the isomorphism problem on proper circular-arc graphs. In *Scandinavian Workshop on Algorithm Theory*, pages 355–366. Springer, 2008.

[116] Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Fixed-parameter tractable canonization and isomorphism test for graphs of bounded treewidth. *SIAM J. Comput.*, 46(1):161–189, 2017.

[117] 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.

[118] George S. Lueker and Kellogg S. Booth. A linear time algorithm for deciding interval graph isomorphism. *J. ACM*, 26(2):183–195, 1979.

[119] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982.

[120] Eugene M. Luks. Permutation groups and polynomial-time computation. In *Groups And Computation, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, October 7-10, 1991*, volume 11 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 139–175. DIMACS/AMS, 1991.

[121] S. Mac Lane. A structural characterization of planar combinatorial graphs. *Duke Mathematical Journal*, 3(3):460–472, 1937.

[122] P. Mani. Automorphismen von polyedrischen graphen. *Mathematische Annalen*, 192(4):279–303, 1971.

[123] François Margot. Symmetry in integer linear programming. In *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, pages 647–686. Springer, 2010.

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

[125] Rudolf Mathon. A note on the graph isomorphism counting problem. *Inf. Process. Lett.*, 8(3):131–132, 1979.

[126] R.M. McConnell and J.P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1–3):189–241, 1999.

[127] Gary L. Miller. Graph isomorphism, general remarks. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 143–150. ACM, 1977.

[128] Gary L. Miller. Isomorphism testing for graphs of bounded genus. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 225–235. ACM, 1980.

[129] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, 2001.

[130] W. Naji. *Graphes de Cordes: Une Caracterisation et ses Applications.* PhD thesis, l'Université Scientifique et Médicale de Grenoble, 1985.

[131] R. Nedela and M. Škoviera. Exponents of orientable maps. *Proceedings of the London Mathematical Society*, 75(1):1–31, 1997.

[132] Roman Nedela, Ilia Ponomarenko, and Peter Zeman. Testing isomorphism of circular-arc graphs in polynomial time. *arXiv preprint arXiv:1903.11062*, 2019.

[133] T. Needham. *Visual complex analysis.* Oxford University Press, 1998.

[134] Daniel Neuen. Hypergraph isomorphism for groups with restricted composition factors. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 88:1–88:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[135] Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of robust graph isomorphism, lasserre gaps, and asymmetry of random graphs. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1659–1677. SIAM, 2014.

[136] S. Oum. Rank-width and vertex-minors. *J. Comb. Theory, Ser. B*, 95(1):79–100, 2005.

[137] I. N. Ponomarenko. The isomorphism problem for classes of graphs closed under contraction. *Journal of Soviet Mathematics*, 55(2):1621–1643, 1991.

[138] Louis C. Ray and Russell A. Kirsch. Finding chemical records by digital computers. *Science*, 126(3278):814–819, 1957.

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

[140] R. W. Robinson. Enumeration of non-separable graphs. *Journal of Combinatorial Theory*, 9(4):327–356, 1970.

[141] J. J. Rotman. *An Introduction to the Theory of Groups*. Graduate Texts in Mathematics. Springer, 1994.

[142] T. Saitoh, Y. Otachi, K. Yamanaka, and R. Uehara. Random generation and enumeration of bipartite permutation graphs. *Journal of Discrete Algorithms*, 10:84–97, 2012.

[143] W. Schnyder. Planar graphs and poset dimension. *Order*, 5(4):323–343, 1989.

[144] Ákos Seress and Á Seress. *Permutation group algorithms*. Number 152. Cambridge University Press, 2003.

[145] Yossi Shiloach. Fast canonization of circular strings. *Journal of algorithms*, 2(2):107–121, 1981.

[146] J. Spinrad. On comparability and permutation graphs. *SIAM J. Comput.*, 14(3):658–670, 1985.

[147] J. P. Spinrad. Recognition of circle graphs. *J. of Algorithms*, 16(2):264–282, 1994.

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

[149] Juraj Stacho. On 2-subcolourings of chordal graphs. In Eduardo Sany Laber, Claudson F. Bornstein, Loana Tito Nogueira, and Luérbio Faria, editors, *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium, Búzios, Brazil, April 7-11, 2008, Proceedings*, volume 4957 of *Lecture Notes in Computer Science*, pages 544–554. Springer, 2008.

[150] Ernst Steinitz. *Polyeder und Raumeinteilungen. Encyclopädie der mathematischen Wissenschaften.* BG Teubner Verlag Leipzig, 1922.

[151] J. Stillwell. *Geometry of surfaces.* Springer, 1992.

[152] Ondrej Šuch. Vertex-transitive maps on a torus. *Acta Math. Univ. Comenianae*, 80(1):1–30, 2011.

[153] A. Takaoka. Graph isomorphism completeness for trapezoid graphs. *CoRR*, abs/1503.06929, 2015.

[154] C. Thomassen. Tilings of the torus and the Klein bottle and vertex-transitive graphs on a fixed surface. *Trans. Amer. Math. Soc.*, 323(2):605–635, 1991.

[155] M.C. Thornton. Spaces with given homeomorphism groups. *Proc. Amer. Math. Soc.*, 33:127–131, 1972.

[156] B.A. Trakhtenbrot. Towards a theory of non-repeating contact schemes. *Trudi Mat. Inst. Akad. Nauk SSSR*, 51:226–269, 1958.

[157] W. T. Tutte. *Connectivity in graphs*, volume 15. University of Toronto Press, 1966.

[158] W. T. Tutte. *Graph Theory.* Cambridge University Press, 2001.

[159] R. Uehara. Tractabilities and intractabilities on geometric intersection graphs. *Algorithms*, 6(1):60–83, 2013.

[160] T.R.S. Walsh. Counting unlabeled three-connected and homeomorphically irreducible two-connected graphs. *J. Combin. Theory B*, 32:12–32, 1982.

[161] L. Weinberg. A simple and efficient algorithm for determining isomorphism of planar triply connected graphs. *IEEE Transactions on Circuit Theory*, 13(2):142–148, 1966.

[162] B. Weisfeiler and A.A. Leman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 9:12–16, 1968.

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

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

[165] Hao Zhou, Wei Zhang, Fengqiong Wei, and Yunfang Chen. Analysis of android malware family characteristic based on isomorphism of sensitive api call graph. In *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pages 319–327, 2017.

List of publications

The papers related to this thesis are marked by $\star$.

Journal papers

1. Steven Chaplick, and Peter Zeman: *Combinatorial Problems on H-graphs.* Electronic Notes in Discrete Mathematics. September 2017.

2. Pavel Klavík, Dušan Knop, Peter Zeman: *Graph isomorphism restricted by lists.* Theoretical Computer Science. January 2021.

3. $\star$ Ken-ichi Kawarabayashi, Pavel Klavík, Bojan Mohar, Roman Nedela, and Peter Zeman: *Isomorphism of maps on the sphere.* Volume "Polytopes and Discrete Geometry" of Contemporary Mathematics, American Mathematical Society. January 2021.

4. Steven Chaplick, Fedor V. Fomin, Petr A. Golovach, Dušan Knop, and Peter Zeman: *Kernelization of Graph Hamiltonicity: Proper H-graphs.* SIAM Journal of Discrete Mathematics. April 2021.

5. Steven Chaplick, Martin Töpfer, Jan Voborník, and Peter Zeman: *On H-Topological Intersection Representations of Graphs.* Algorithmica. June 2021.

6. $\star$ Pavel Klavík, and Peter Zeman: *Automorphism Groups of Geometrically Represented Graphs.* Currently being revised in Ars Mathematica Contemporanea.

7. $\star$ Pavel Klavík, Roman Nedela, Peter Zeman: *Jordan-like characterization of automorphism groups of planar graphs.* Currently being revised in Journal of Combinatorial Theory, Series B.

Conference proceedings

1. $\star$ Pavel Klavík, and Peter Zeman: *Automorphism Groups of Geometrically Represented Graphs.* 32nd International Symposium on Theoretical Aspects of Computer Science (STACS). March 2015.

2. Steven Chaplick, Martin Töpfer, Jan Voborník, and Peter Zeman: *On H-Topological Intersection Representations of Graphs.* International Workshop on Graph-Theoretic Concepts in Computer Science (WG). June 2017.

3. Steven Chaplick, Fedor V. Fomin, Petr A. Golovach, Dušan Knop, and Peter Zeman: *Kernelization of Graph Hamiltonicity: Proper H-graphs.* Workshop on Algorithms and Data Structures (WADS). August 2019.

4. Pavel Klavík, Dušan Knop, and Peter Zeman: *Graph Isomorphism Restricted by Lists.* International Workshop on Graph-Theoretic Concepts in Computer Science (WG). June 2020.

5. ⋆ Ken-ichi Kawarabayashi, Bojan Mohar, Roman Nedela, and Peter Zeman: *Automorphisms and Isomorphisms of Maps in Linear Time*. International Colloquium on Automata, Languages and Programming (ICALP). July 2021.

Currently submitted papers

1. ⋆ Vít Kalisz, Pavel Klavík, and Peter Zeman: *Circle Graph Isomorphism in Almost Linear Time*. Submitted.
2. Jiří Fiala, Ignaz Rutter, Peter Stümpf, Peter Zeman: *Extending Partial Representations of Circular-Arc Graphs*. Submitted.
3. Kenta Ozeki, Peter Zeman: *Characterization of extended star graphs by asteroidal k-tuples*. Submitted.
4. ⋆ Peter Zeman: *Automorphism groups of subclasses of planar graphs*. Submitted.
5. ⋆ Steven Chaplick, Peter Zeman: *Isomorphism-completeness for H-graphs*. Submitted.
6. ⋆ Vikraman Arvind, Roman Nedela, Ilia Ponomarenko, and Peter Zeman: *Testing isomorphism of chordal graphs of bounded leafage is fixed-parameter tractable*. Submitted.