# Partial and Simultaneous Representations of Circular Permutation Graphs

Master Thesis of


Miriam Münch

At the Department of Informatics and Mathematics
Chair of Theoretical Computer Science

UNIVERSITÄT PASSAU


| | |
|---|---|
| Reviewers: | Prof. Dr. I. Rutter |
| | Prof. Dr. D. Sudholt |
| Advisor: | Peter Stumpf, M.Sc. |

Time Period: December 2nd 2020 – June 7th 2021

**Statement of Authorship**

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text.

Passau, June 7, 2021

## Abstract

A graph $G = (V, E)$ is a comparability graph if its edges can be oriented transitively, i.e. there exists an orientation $\overrightarrow{E}$ of $G$ such that $\overrightarrow{E}$ containing the directed edges $uv$ and $vw$, implies that $u$ and $w$ are also connected and $uw \in \overrightarrow{E}$ for every $u, v, w \in V$. The class of permutation graphs contains all graphs that are represented by at least one permutation diagram which is a specific linear intersection representation. A graph is a circular permutation graph if it is represented by a specific circular intersection representation called a circular permutation diagram.

A partial representation of a comparability graph is a transitive orientation of a subgraph. Partial representations of permutation or circular permutation graphs are permutation or circular permutation diagrams respectively, representing a subgraph. The partial representation problem for a graph class $\mathcal{C}$ answers the question whether a given partial representation of a $\mathcal{C}$ graph can be extended to a representation of the entire graph. Given $\mathcal{C}$-graphs $G_1, \dots, G_k$, the simultaneous representation problem asks whether there exists representations $R_1, \dots, R_k$ such that for all $1 \le i, j \le k$, $R_i$ represents $G_i$ and $R_i$ and $R_j$ are isomorph on the subgraph shared by $G_i$ and $G_j$. Both problems extend the recognition problem. We present efficient algorithms to solve the partial representation problem and the simultaneous representation problem for the three classes of comparability, permutation and circular permutation graphs. It turns out that the partial representation problem can be solved in linear time for all three considered graph classes. The simultaneous representation problem for comparability graphs is solvable in linear time, if all input graphs pairwise share the same induced subgraph. In this case the problem is also solvable in quadratic time for permutation and circular permutation graphs. If the input graphs pairwise share an arbitrary subgraph however, the simultaneous representation problem is already `NP`-complete for permutation graphs.



## Deutsche Zusammenfassung

Ein Graph $G = (V, E)$ ist ein Vergleichbarkeitsgraph, wenn seine Kanten transitiv orientiert werden können, d.h. es existiert eine Orientierung $\overrightarrow{E}$ von $G$, so dass, falls $\overrightarrow{E}$, die gerichteten Kanten $uv$ und $vw$ enthält, dies impliziert, dass $u$ und $w$ ebenfalls verbunden sind und $uw \in \overrightarrow{E}$ für alle $u, v, w \in V$. Ein Graph $G = (V, E)$ ist ein Permutationsgraph, wenn er von einem Permutationsdiagramm, welches eine spezielle lineare Schnittrepräsentation bezeichnet, repräsentiert wird. Die Klasse der zirkulären Permutationsgraphen enthält alle Graphen, die durch eine spezielle zirkuläre Schnittrepräsentation, sogenannte zirkuläre Permutationsdiagramme, repräsentiert werden.

Eine partielle Repräsentation eines Vergleichbarkeitsgraphen ist eine transitive Orientierung eines Teilgraphen. Partielle Repräsentationen von Permutations- oder zirkulären Permutationsgraphen sind Permutations- bzw. zirkuläre Permutationsdiagramme, die einen Teilgraphen repräsentieren. Das Partialrepräsentationsproblem für eine Graphenklasse $\mathcal{C}$ beantwortet die Frage, ob eine gegebene Partialrepräsentation eines $\mathcal{C}$-Graphen zu einer Repräsentation des gesamten Graphen erweitert werden kann. Gegeben $\mathcal{C}$-Graphen $G_1, \dots, G_k$, fragt das Simultanrepräsentationsproblem, ob es Darstellungen $R_1, \dots, R_k$ gibt, so dass für alle $1 \le i, j \le k$, $G_i$ durch $R_i$ repräsentiert wird und $R_i$ und $R_j$ isomorph auf dem von $G_i$ und $G_j$ geteilten Teilgraphen sind. Beide Probleme sind Erweiterungen des Erkennungsproblems. Wir präsentieren effiziente Algorithmen zur Lösung des partiellen Repräsentationsproblems und des simultanen Repräsentationsproblems für die drei Klassen der Vergleichbarkeit-, Permutations- und zirkulären Permutationsgraphen. Es zeigt sich, dass das partielle Repräsentationsproblem für alle drei betrachteten Graphenklassen in linearer Zeit gelöst werden kann. Das Simultanrepräsentationsproblem kann für Vergleichbarkeitsgraphen in linearer Zeit gelöst werden, sofern alle Eingabegraphen den selben induzierten Subgraphen teilen. In diesem Fall is das Problem auch für Permutations- und zirkuläre Permutationsgraphen in quadratischer Zeit lösbar. Wenn sich die Eingabegraphen jedoch paarweise einen unterschiedlichen Subgraphen teilen, ist das Simultanrepräsentationsproblem bereits für Permutationsgraphen NP-vollständig.

# Contents

# 1. Introduction

In this thesis we examine two problems for the classes of comparability (Comp), permutation (Perm) and circular permutation graphs (CPerm). The first one is the partial representation problem $\textsc{RepExt}(\mathcal{C})$, which given a $\mathcal{C}$-graph $G = (V, E)$ for a graph class $\mathcal{C}$ and a corresponding partial representation $D'$ of a subgraph $H$, asks, whether it is possible to extend $D'$ to a representation $D$ of the entire graph, such that $D$ restricted to $H$ is isomorphic to $D'$.

The second problem we deal with is the simultaneous representation problem $\textsc{SimRep}(\mathcal{C})$, which, given a number of $\mathcal{C}$-graphs $G_1, \ldots, G_r$, asks whether there exist representations for the given graphs, such that for every pair of these graphs, the corresponding representations are the same on the subgraph shared by them.

Both problems are already well-studied. Concerning interval graphs for example it is already known, that the partial representation problem can be solved in linear time [KKV11, BR15] and simultaneous interval graphs can be recognized in $\mathcal{O}(n^2 \log n)$ time [JL10]. For chordal graphs both problems turned out to be NP-complete [KKOS15, JL10].

In this thesis we examine how the two problems can be solved efficiently for comparability, permutation and circular permutation graphs. A comparability graph is a graph whose edges can be oriented transitively, i.e. for every triple of vertices $u, v$ and $w$, $u \to v$ and $v \to w$ implies $u \to w$. The class of permutation graphs contains all graphs that are represented by a permutation diagram consisting of two parallel horizontal lines $L_1$ and $L_2$ and a line segment connecting a point on $L_1$ to a point on $L_2$ for every vertex in the graph. Such a (linear) permutation diagram represents a graph $G = (V, E)$ if and only if for every $u, v \in V$ the corresponding segments intersect if and only if $E$ contains an edge between $u$ and $v$. A circular permutation diagram consists of two concentric circles $C_1$ and $C_2$ with distinct radius, and a chord connecting a point on $C_1$ to a point on $C_2$ for every vertex in the graph. Analogously to the linear case, a circular permutation diagram represents a graph $G = (V, E)$ if and only if for every $u, v \in V$ the corresponding chords intersect if and only if $E$ contains an edge between $u$ and $v$. The graphs represented by a circular permutation diagram are called circular permutation graphs and were first studied by Rotem and Urrutia[RU82].

Klavík et al. already showed that $\textsc{RepExt}(\mathit{Comp})$ can be solved in $\mathcal{O}((n+m)\Delta)$ time for comparability graphs with $n$ vertices, $m$ edges and maximum degree $\Delta$ [KKKW12] (see Chapter 3). For $\textsc{RepExt}(\mathit{Perm})$ they gave an $\mathcal{O}(n^3)$ time algorithm. We show that both problems can be solved in $\mathcal{O}(n+m)$ time. Furthermore we also give an algorithm solving

| | RepExt | SimRep |
|---|---|---|
| Comp | $\mathcal{O}((n+m)\Delta)$ [KKKW12] | $\mathcal{O}(nm)$ [JL10] |
| Perm | $\mathcal{O}(n^3)$ [KKKW12] | $\mathcal{O}(n^3)$ [JL10] |
| CPerm | ? | ? |

(a) known runtimes

| | RepExt | SimRep |
|---|---|---|
| Comp | $\mathcal{O}(n+m)$ | $\mathcal{O}(n+m)$ |
| Perm | $\mathcal{O}(n+m)$ | $\mathcal{O}(n^2)$ |
| CPerm | $\mathcal{O}(n+m)$ | $\mathcal{O}(n^2)$ |

(b) new runtimes

Figure 1.1: Comparison of results

RepExt(*CPerm*) in the same runtime. Concerning the simultaneous representation problem for so called *sunflower* graphs that all share the same subgraph, Jampani and Lubiw gave an $\mathcal{O}(nm)$ algorithm to solve SimRep(*Comp*) and an $\mathcal{O}(n^3)$ algorithm to solve SimRep(*Perm*) [JL10] (see Chapter 3). We show that the problem can be solved in $\mathcal{O}(n+m)$ time for comparability graphs and in $\mathcal{O}(n^2)$ time for permutation and circular permutation graphs. Figure 1.1 summarises the already known and the new results.

All new algorithms presented in this thesis use the *modular decomposition* of a graph which was first described by Gallai [Gal67]. A *module* of an undirected graph $G = (V, E)$ is a subset of vertices $X \subseteq V$ such that every vertex in $V \setminus X$ is either adjacent to every $x \in X$ or to none of them. The modular decomposition of a graph is a canonical linear-space representation of its modules. McConnell and Spinrad presented linear-time algorithms to compute the modular decomposition of a graph and to solve the strongly related *transitive orientation problem* that asks whether the edges of a given graph can be oriented transitively [MS99]. With these base algorithms they were able to show that also many other related problems such as *recognition of permutation graphs and two-dimensional partial orders*, *Recognition of cointerval graphs and interval graphs* and *Recognition of circular permutation graphs* are solvable in $\mathcal{O}(n+m)$ time for graphs with $n$ vertices and $m$ edges.

For the more general non-sunflower case, where the input graphs pairwise share an arbitrary set of vertices and edges induced by them, we proof that for permutation graphs the simultaneous representation problem is NP-complete. Since every permutation graph is also a circular permutation graph and a comparability graph the simultaneous representation problem for these two graph classes is also NP-complete in the non-sunflower case.

In Chapter 2 we give basic definitions and notations used in the following. In Chapter 3 we give linear-time algorithms to solve RepExt(*Comp*) and SimRep(*Comp*) respectively. The linear-time algorithm to solve RepExt(*Perm*) and the algorithm with quadratic runtime for solving SimRep(*Perm*) for the sunflower case we present in Chapter 4. Furthermore, in this chapter we see that the simultaneous representation problem for non-sunflower permutation graphs is NP-complete. In Chapter 5 we give the linear-time algorithm to solve RepExt(*CPerm*) and the algorithm with quadratic runtime for solving SimRep(*CPerm*).

# 2. Preliminaries

In this chapter we introduce basic definitions and notation that we use in the following sections. Let $G = (V, E)$ be an undirected graph and let $U \subseteq V$ be a set of vertices. Then the vertices in $U$ *induce* all edges in $E$ whose endpoints both are in $U$. We denote the graph that consists of the vertices in $U$ and all edges of $G$ induced by them as $[U]_G$.

## 2.1 Comparability Graphs

To define the class of comparability graphs, we first introduce transitive orientations. Figure 2.1 gives an example of a comparability graph and a corresponding transitive orientation. A *transitive orientation* of an undirected graph $G = (V, E)$ is an assignment of direction to each of its edges, such that for every $u, v, w \in V$, if $uv \in E$ is oriented from $u$ to $v$ and $vw \in E$ is oriented from $v$ to $w$, then $uw$ is also an edge in $E$ and is oriented from $u$ to $w$. A graph $G$ is *transitively orientable* or a *comparability graph* if and only if a transitive orientation for its edges does exist. If the complement of $G$ is a comparability graph, $G$ is a *co-comparability graph*. We denote the class of comparability graphs by Comp. The class coComp contains all co-comparability graphs. In the following the term *(partial) representation* for a comparability graph denotes a (partial) transitive orientation of its edges.

It will be useful many times to examine the partial orientation $<$ of the vertices of a comparability graph $G = (V, E)$ induced by a (partial) transitive orientation $\overrightarrow{E}$ of $G$. For two vertices $u$ and $v$ in $V$ we have $u < v$ if $\overrightarrow{E}$ contains the directed edge $\overrightarrow{uv}$. If $uv \notin E$, the vertices $u$ and $v$ are incomparable. The transitive orientation of the graph given in Figure 2.1 induces $1 < 3$, $1 < 4$, $2 < 3$, $2 < 4$ and $2 < 5$. Note that a transitive orientation of a complete graph $G$ induces a total order on the vertices of $G$.

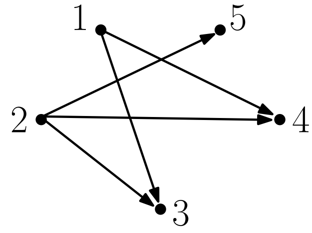



Figure 2.1: A transitively oriented comparability graph.

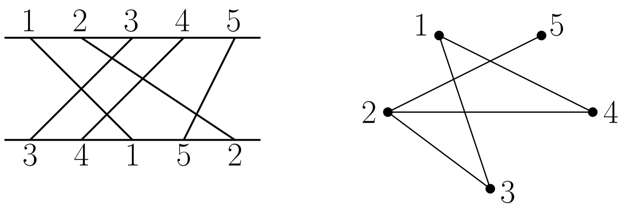



Figure 2.2: A permutation diagram and a corresponding permutation graph.

## 2.2 Permutation and Circular Permutation Graphs

In this section we define the classes of permutation and circular permutation graphs. An example for the following two definitions concerning permutation diagrams and permutation graphs is given in Figure 2.2. Usually permutation diagrams are defined to consist of two parallel lines $L_1$ and $L_2$ labelled by $1, 2, \ldots, n$ and a permutation of $1, 2, \ldots, n$ respectively, from left to right and a set of line segments $\{\bar{1}, \bar{2}, \ldots, \bar{n}\}$ such that for every $1 \leq i \leq n$ segment $\bar{i}$ connects $i$ on $L_1$ with $i$ on $L_2$. Here, we define permutation diagrams in a more general way by using a set of arbitrary labels for the horizontal lines. This will make the description of partial representations easier, since we do not have to rename labels of the diagram or nodes of a corresponding graph when extending a representation.

**Definition 2.1** (Permutation Diagram)**.** *Let $S$ be a set of labels and let $P_1$ and $P_2$ be permutations of the elements in $S$. Let $L_1$ and $L_2$ be two parallel horizontal lines, such that $|S|$ pairwise distinct points on the upper line $L_1$ are labelled with the elements in $P_1$ from left to right, while $|S|$ points on the bottom line $L_2$ are labelled with the elements of $P_2$, again from left to right.*

*A* permutation diagram *$D$ consists of $L_1$ and $L_2$ and a set of line segments $\{\overline{u} \mid u \in S\}$ such that for every $u \in S$, segment $\overline{u}$ connects the point labelled with $u$ on $L_1$ with the point labelled with $u$ on $L_2$. Note that two segments $\overline{x}$ and $\overline{y}$ intersect if and only if the order of the corresponding endpoints along $L_1$ is opposite to their order along $L_2$.*

**Definition 2.2** (Permutation Graph)**.** *Let $S$ be a set of labels and let $D$ be a permutation diagram of the elements in $S$. An undirected graph $G$* is represented by *$D$ if its vertices can be labelled with the elements of $S$, such that segment $\overline{u}$ intersects segment $\overline{v}$ if and only if the vertices labelled with $u$ and $v$ are adjacent in $G$. $G$ is called a* permutation graph*, if there exists at least one permutation diagram $D$ that represents $G$.*

Let Perm be the class of permutation graphs. Then we have $\textsc{Perm} = \textsc{Comp} \cap \textsc{coComp}$ [EPL72].

Let $G = (V, E)$ be a permutation graph. If we have a transitive orientation $\overrightarrow{E}$ of $G$ and a transitive orientation $\overrightarrow{\overline{E}}$ of the complement graph $\overline{G} = (V, \overline{E})$, we get the corresponding permutation diagram $D$ for $G$ as follows [PLE71]:

Consider the complete graph $G' = (V, E \cup \overline{E})$. Then $E'_1 := \overrightarrow{E} \cup \overrightarrow{\overline{E}}$ is a transitive orientation of $G'$ and we can label the vertices of $V$ according to the total order induced by $E'_1$, i.e. every $v \in V$, is labelled with the number of edges oriented towards $v$ incremented by one. Then the points along the bottom line $L_1$ of $D$, are labelled with $1, \ldots, n$ which is the total order induced by $E'_1$. The points along the bottom line $L_2$ of $D$, are labelled with the labels of the vertices according to the total order induced by the transitive orientation $E'_2 := \overleftarrow{E} \cup \overrightarrow{\overline{E}}$ of $G'$, which we obtain from $E'_1$ by reversing all orientations of $\overrightarrow{E}$. This means that $v$ is the direct right neighbour of $u$ if and only if the number of edges oriented towards $v$ is exactly by one greater than the number of edges oriented towards $u$ in $E'_2$.

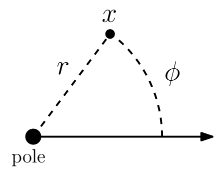



Figure 2.3: A point $x = (r, \phi)$ in a polar coordinate system.

Given a permutation diagram $D$ representing a permutation graph $G = (V, E)$ we directly get a corresponding transitive orientation of $G$ by orienting an edge $e$ in $E$ towards the endpoint that appears to the right of the other on the upper horizontal line of $D$. From PERM = CO ∩ coCO we get that the complement of every permutation graph is a permutation graph itself. We get the corresponding permutation diagram $D'$ of $\overline{G}$ by reversing the order of the nodes along the bottom line of $D$.

Next we will introduce the class of circular permutation graphs. Here we consider them embedded into a polar coordinate system which is a two-dimensional coordinate system in which every point $x = (r, \phi)$ is determined by its distance $r$ from a fixed point called *pole* and its angle *phi* from a fixed direction (see Figure 2.3).

Furthermore, we define a chord in a polar coordinate system to be *monotonous* if the $r$ coordinates of the points along the chord either increases or decreases monotonously and either the $\phi$ coordinate is the same for all points or the coordinates of every two points that are not both endpoints, differ in $\phi$.

**Definition 2.3** (Circular Permutation Diagram)**.** *Let $S$ be a set of labels and let $P_1$ and $P_2$ be permutations of the elements in $S$. Let $C_1$ and $C_2$ be two concentric circles with shared centre $M$, such that $|S|$ points on the inner circle $C_1$ are labelled with the elements of $P_1$ in the counter-clockwise direction, while $|S|$ points on the outer circle $C_2$ are labelled with the elements of $P_2$, also in the counter-clockwise direction.*

*A* circular permutation diagram *$C$ of $P$ consists of $C_1$ and $C_2$ in a polar coordinate system whose pole is located in $M$ and a set of monotonous chords $\{\overline{u} \mid u \in S\}$ such that for every $u \in S$, chord $\overline{u}$ connects the point labelled with $u$ on the inner and the point labelled with $u$ on the outer circle, totally within the annular region contained between $C_1$ and $C_2$. Furthermore here, we demand for every $u \neq v$ that $\overline{u}$ and $\overline{v}$ intersect at most once.*

Note that there may exist different circular permutation diagrams for the same permutation. Furthermore, it is also possible to obtain the same diagram for two different permutations.

**Definition 2.4** (Circular Permutation Graph)**.** *Let $C$ be a circular permutation diagram. An undirected graph $G$* is represented by *$C$ if its vertices can be labelled with the elements of $S$ such that chord $\overline{i}$ intersects chord $\overline{j}$ if and only if the vertices labelled with $i$ and $j$ are adjacent in $G$. $G$ is called a* circular permutation graph*, if there exists at least one circular permutation diagram $C$ that represents $G$.*

An example of a circular permutation graph and its corresponding circular permutation diagram is given in Figure 2.4.

To illustrate the relation between linear and circular permutation diagrams we need to further consider cylindrical coordinate systems which are three-dimensional coordinate systems in which every point $x = (r, \phi, z)$ is determined by its distance $r$ from a reference axis, its angle $\phi$ from a fixed reference direction and its distance $z$ from a fixed plane perpendicular to the chosen axis (see Figure 2.5).

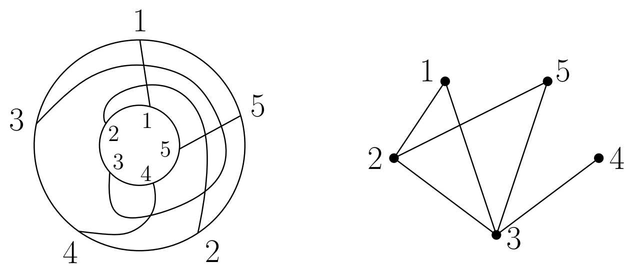



Figure 2.4: A circular permutation diagram and a corresponding circular permutation graph

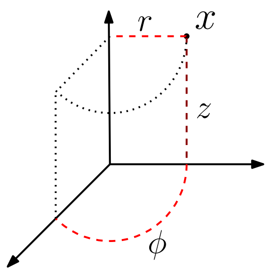



Figure 2.5: A point $x = (r, \phi, z)$ in a cylindrical coordinate system.

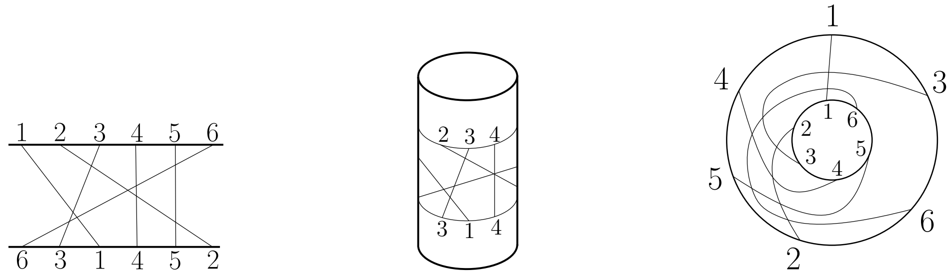
Analogously to the case of a polar coordinate system, we define a chord in a cylindrical coordinate system to be *monotonous* if the $z$ coordinates of the points along the chord either increases or decreases monotonously and either the $\phi$ coordinate is the same for all points or the coordinates of every two points that are not both endpoints, differ in $\phi$.

**Definition 2.5** (Cylindrical Permutation Diagram)**.** *Let $S$ be a set of labels and let $P_1$ and $P_2$ be permutations of the elements in $S$. A* cylindrical permutation diagram *$C$ consists of two parallel lines $L_1$ and $L_2$ on the surface of a cylinder parallel to the bottom base area such that $|S|$ points on the upper line $L_1$ are labelled with the elements in $P_1$ from left to right, while $|S|$ points on the bottom line $L_2$ are labelled with the elements of $P_2$, again from left to right. Furthermore we have a set of monotonous chords $\{\overline{u} \mid u \in S\}$ such that for all points along every chord the $r$ coordinate equals the radius of the considered cylinder and for every $u \in S$, chord $\overline{u}$ connects the point labelled with $u$ on $L_1$ with the point labelled with $u$ on $L_2$ totally within the region between the two parallel lines.*

**Lemma 2.6.** *Every circular permutation diagram $C$ can be mapped isomorphically to a cylindrical permutation diagram $C'$ such that $C$ and $C'$ represent the same circular permutation graphs.*

*Proof.* Consider a circular permutation diagram $C$ and a cylinder whose base area has the same radius $y$ as the inner circle $C_1$ of $C$. Then we map the two circles $C_1$ and $C_2$ to two lines parallel to the bottom base area whose vertical distance equals the difference of the radii of $C_1$ and $C_2$. Now we map every point $x = (r, \phi)$ on every chord of $C$ to the point $x = (y, \phi, r)$. Then every chord $\bar{v}$ in $C$ is mapped to a chord in $C'$ connecting a point on the lower parallel line to a point on the upper parallel line totally within the region between the two lines. The intersection points with the two parallel lines we label with $v$. Then a chord $\bar{v}$ in $C'$ intersects a chord $\bar{u}$ if and only if the two corresponding chords in $C$ intersect, hence $C'$ represents the same circular permutation diagram as $C$.

Now consider a cylindrical permutation diagram $C'$ and let $C_1$ and $C_2$ be two concentric circles around the origin of a polar coordinate system such that the radius of $C_1$ equals




(a) linear permutation diagram

(b) cylinder representation

(c) circular permutation diagram

Figure 2.6: The linear permutation diagram (a) is mapped onto the surface of the cylinder (b) and afterwards transformed into the circular permutation diagram (c)

the radius of the base area of the cylinder and the radius of $C_2$ is greater than the radius of $C_1$ by the vertical distance between the two horizontal lines of $C'$. Now we map every point $x' = (r, \phi, z)$ on every chord of $C'$ to the point $x = (z, \phi)$. Then every chord $\bar{v}$ in $C'$ is mapped to a chord in $C$ connecting a point on $C_1$ to a point on $C_2$ totally within the annular region between the two circles. The intersection points with the two circles we label with $v$. Then a chord $\bar{v}$ in $C$ intersects a chord $\bar{u}$ if and only if the two corresponding chords in $C'$ intersect, hence $C$ represents the same circular permutation diagram as $C'$. □

Hence circular permutation diagrams are equivalent to cylindrical permutation diagrams.

**Lemma 2.7.** *Every permutation diagram $D$ can be transformed into a circular representation $C$.*

*Proof.* Consider a Cartesian coordinate system, with the bottom line $L_2$ of $D$ on the $x$-axis, such that the left endpoint of $L_2$ lies in the origin. Project $D$ onto the surface of a cylinder with a circumference that equals the length of the horizontal lines $L_1$ and $L_2$ of $D$, such that $L_1$ and $L_2$ run parallel to the base areas and the segments are oriented in the same direction as in $D$. To ensure monotony, for the projections of the line segments we demand that for all points $x = (r_1, \phi_1, z_1), y = (r_2, \phi_2, z_2)$ along them, $z_1 \neq z_2$. The points along one segment, except for the two endpoints, either all have the same $\phi$ -value, or their coordinates all pairwise differ in $\phi$. Then we know that there exists an equivalent circular permutation diagram representing the same graphs as $D$. Figure 2.6 illustrates the steps of an example transformation. □

Let $G = (V, E)$ be a circular permutation graph. *Switching* a vertex $v$ in $G$, i.e. connecting it to all vertices it was not adjacent to in $G$ and removing all edges to its former neighbours, gives us the graph $G_v = (V, E_v)$ with $E_v = (E \backslash \{xy \in E \mid x = v\}) \cup \{vx \mid x \in V, vx \notin E\}$.

Let $C$ be a circular permutation diagram representing a permutation graph $G = (V, E)$ and let $v \in V$ be a vertex in $G$. The chord $\overline{v}$ in $C$ can be *switched* to the chord $\overline{v}'$, if $\overline{v}'$ has the same endpoints as $\overline{v}$, but intersects exactly the chords $\overline{v}$ does not intersect in $C$. Hence, this modified circular permutation diagram $C'$ is a representation of $G_v$ [RU82]. Here we will further demand that there exists no intermediate point along $\overline{u}$, such that an intermediate point of $\overline{u}'$ has the same angular coordinate. Notice that for every chord $\overline{u}$ in a circular permutation diagram $C$, there exists at least one chord $\overline{u}'$ such that $\overline{u}$ can be switched to $\overline{u}'$. Most of the time, it suffices to know that there exists a $\overline{u}'$ with the desired properties that $\overline{u}$ can be switched to, hence, we often use the term *switch chord $u$*, without further specifying $\overline{u}'$.

**Definition 2.8.** *Let $D_1$ and $D_2$ be two circular permutation diagrams. Then $D_1$ and $D_2$ are* isomorphic*, if and only if the labels in $S$ appear in the same order respectively along both the outer and the inner circles of the diagrams and for every $u, v \in S$, chord $\overline{u}$ intersects chord $\overline{v}$ in $D_1$ if and only if chord $\overline{u}$ intersects chord $\overline{v}$ in $D_2$.*

## 2.3 Partial and Simultaneous Representations

A *partial representation* of a graph $G = (V, E)$ is a mapping $\phi : R \to \mathcal{S}$ of the induced subgraph $G[R]$ for a set $R \subseteq V$ to a class $\mathcal{S}$ of objects [KKKW12]. The partial representation problem $\textsc{RepExt}(\mathcal{C})$ for a graph class $\mathcal{C}$ represented in a class $\mathcal{S}$ is defined as follows [KKKW12]. Given a graph $G \in \mathcal{C}$ and a partial representation $\phi : R \to \mathcal{S}$ is it possible to extend $\phi$ to a representation $\psi : V \to \mathcal{S}$ of the entire graph $G$, such that $\psi|_R = \phi$?

Let $\mathcal{C}$ be a graph class and let $G_1, \ldots, G_r$ be graphs in $\mathcal{C}$. Then $G_1, \ldots, G_r$ are said to be *simultaneous $\mathcal{C}$ graphs*, if there exist representations $R_1, \ldots, R_r$ such that for each $1 \leq i, j \leq r$, $R_i$ represents $G_i$ and $R_i$ and $R_j$ are the same on the subgraph shared by $G_i$ and $G_j$. The simultaneous $\mathcal{C}$ representation problem $\textsc{SimRep}(\mathcal{C})$ for a graph class $\mathcal{C}$, given $\mathcal{C}$-graphs $G_1, \ldots, G_r$, asks whether they are simultaneous $\mathcal{C}$ graphs [JL10].

Often the simultaneous representation is examined for the special case of so called *sunflower-graphs* where the given graphs $G_1, G_2, \ldots, G_r$ all pairwise share the same induced subgraph $I$.

**Definition 2.9.** *Let $\mathcal{C}$ be a graph class and let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2), \ldots, G_r = (V_r, E_r)$ be graphs in $\mathcal{C}$ sharing some vertices $I$ and the edges induced by $I$ such that for any two distinct $i, j$ with $1 \leq i < j \leq r$, $V_i \cap V_j = I$ and $[I]_{G_i} = [I]_{G_j}$. Then $G_1, G_2, \ldots, G_r$ are called* $r$-sunflower $\mathcal{C}$ graphs.

## 2.4 Modular Decomposition

In this section we introduce the modular decomposition on which our efficient algorithms to solve the partial representation problem and the simultaneous representation problem are based. Let $G = (V, E)$ be an undirected graph for the rest of this section. The overall aim is to find so called quotient graphs, whose vertices are modules, i.e. special subsets of $V$, that are transitively orientable if and only if the subgraph of $G$ induced by the edges represented by the edges in the quotient graph can be oriented transitively. First of all, we need the definition of a *module*.

**Definition 2.10** (Modules)**.** *A set of vertices $M \subseteq V$ is a* module*, if every vertex $u \in V \backslash M$ is either adjacent to all vertices in $M$ or to none of them.*

$V$ and its singleton subsets are called the *trivial modules*. If $G$ has no non-trivial modules, then it is called *prime*. A module $M$ of a graph $G$ that does not overlap with any other module of $G$ is called *strong*. This means, $M$ is strong if for all modules $M'$ of $G$ we either have $M \cap M' = \emptyset$ or $M \subseteq M'$ or $M' \subseteq M$. Let $S$ be a subset of $V$. A module $M$ is *maximal* with respect to $S$, if $M \subset S$ and there exists no module $M'$ such that $M \subset M' \subset S$. Every graph has a uniquely defined *(maximal strong) modular decomposition* that contains all maximal strong modules [HP10]. For the definition of the modular decomposition tree, additionally we have to introduce the $\Gamma$-relation and edge classes. If a graph $G$ is transitively orientable every two edges that are $\Gamma$-related and thus share a common endpoint have to be directed in the same way with respect to this shared endpoint in every transitive orientation of $G$. Moreover in every transitive orientation of $G$ there are only two possible orientations for every edge class where one is the reverse of the other.

**Definition 2.11** (The Γ-relation)**.** *Let $G = (V, E)$ be an undirected graph and let $ab, cd \in E$ with $a, b, c, d \in V$, be two distinct adjacent edges of $G$, i.e. $a = c$ and $b \neq d$. Then $ab\Gamma cd$ if and only if $bd \notin E$.*

*Let $G' = (V, E')$ be the directed graph we obtain from $G$ by replacing every undirected edge $uv$ in $E$ by the two directed edges $\overrightarrow{uv}$ and $\overrightarrow{vu}$. Then for any two edges $ab$ and $bc$ such that $ac \notin E$, no transitive orientation of $H$ can contain $\overrightarrow{ab}$ and $\overrightarrow{bc}$, as well as it can not contain $\overrightarrow{cb}$ and $\overrightarrow{ba}$. We define $\overrightarrow{ab}\Gamma\overrightarrow{cd}$ if and only if $a = c$ and $bd \notin E$, or $b = d$ and $ac \notin E$.*

The Γ relation describes the directly implied dependencies between orientations of two adjacent edges. Edge classes introduced in the following definition describe equivalence classes of edges that can not be oriented independently if we want to receive a transitive orientation.

**Definition 2.12** (Edge classes)**.** *Two distinct edges $ab$ and $cd$ of an undirected graph $G$ belong to the same edge class if and only if there exists a sequence of edges $x_iy_i$ ($i = 1, \ldots, n, n \geq 2$) such that $x_1y_1 = ab$, $x_ny_n = cd$ and $x_jy_j\Gamma x_{j+1}y_{j+1}$ for every $1 \leq j \leq n-1$.*

Now we define the unique modular decomposition tree $T$ for a graph $G = (V, E)$ such that for every node $\mu$ in $T$, the children of $\mu$ in $T$ are exactly the maximal strong modules of $[\mu]_G$.

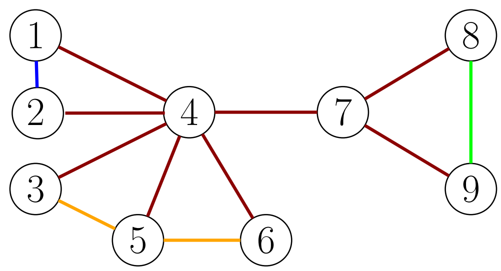
**Definition 2.13** (Modular Decomposition Tree)**.** *For an undirected graph $G = (V, E)$, its* modular decomposition tree *$T$ is recursively uniquely defined as follows:*

*$V$ is the root of $T$. For every node $U$ of $T$ that is not a singleton set, holds:*

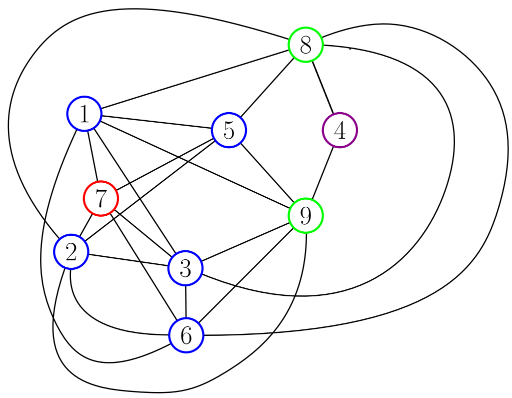
- *If $[U]_G$ is a disconnected graph, then for every connected component of $[U]_G$, the set of its vertices is a child of $U$ in $T$.*
- *If $[U]_{\overline{G}}$ is disconnected, then for every connected component of $[U]_{\overline{G}}$, the set of its vertices is a child of $U$ in $T$.*
- *If $[U]_G$ and $[U]_{\overline{G}}$ are both connected, then the children of $U$ in $T$ are given by the sets contained in the unique proper decomposition $P = \{A_1, \ldots, A_q\}$ of $U$ [Gal67] with the following properties:*
    - *a) For every pair of indices $i, j$ with $(1 \leq i < j \leq q)$: If $[U]_{\overline{G}}$ contains an $A_iA_j$-edge, i.e. an edge where one endpoint belongs to $A_i$ and the other one to $A_j$, then $A_i$ and $A_j$ are fully connected.*
    - *b) All edges in $E$ that connect two distinct sets $A_i$ and $A_j$ for all $i \neq j$ belong to the same edge class $E'$ of $G$. Every vertex in $V$ is incident to at most one edge in $E'$.*
    - *c) The edge classes of $G$, except for $E'$, are the edge classes of the graphs $[A_i]_G = G_i$ $(1 \leq i \leq q)$.*
    - *d) The decomposition $P$ is not a refinement of another proper decomposition of $V$ with the properties a), b) and c).*

The following example illustrates the above definition of the modular decomposition tree.

**Example 2.14.** *Consider the following graph $G = (V, E)$:*

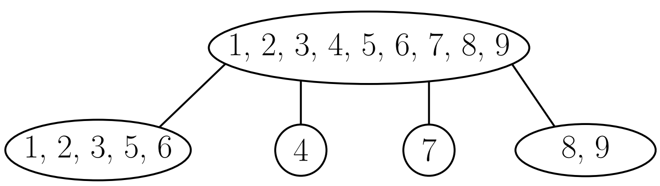



*To get the unique modular decomposition tree $T$ for $G$, we stepwise apply the recursive definition given above. The root of $T$ consists of all vertices in $V$. Since both $G$ and its complement $\overline{G}$ are connected, the children of the root are the sets contained in the unique proper decomposition $P = \{A_1, \ldots, A_n\}$ of $G$ described in Definition 2.13. All edges in $G$ belonging to the same edge class have the same color in Figure 2.14. We can see that the red edges build the edge class $E'$, hence the vertices 4 and 7 both must appear as singleton sets in the decomposition. The set of the remaining nodes is not a module since for example vertex 7 is not connected to every vertex in $\{1, 2, 3, 5, 6, 8, 9\}$ but when splitting this set into vertices adjacent to 7 and vertices adjacent to 4 we get the desired decomposition. In our case $P = \{\{4\}, \{7\}, \{1, 2, 3, 5, 6\}, \{8, 9\}\}$. Property a) is satisfied, since $\overline{G}$ does not contain edges between the pairs of sets $(\{4\}, \{7\})$, $(\{7\}, \{8, 9\})$, and $(\{4\}, \{1, 2, 3, 5, 6\})$ respectively, while the remaining pairs, namely $(\{4\}, \{8, 9\})$, $(\{1, 2, 3, 5, 6\}, \{7\})$ and $(\{1, 2, 3, 5, 6\}, \{8, 9\})$, are all fully connected:*



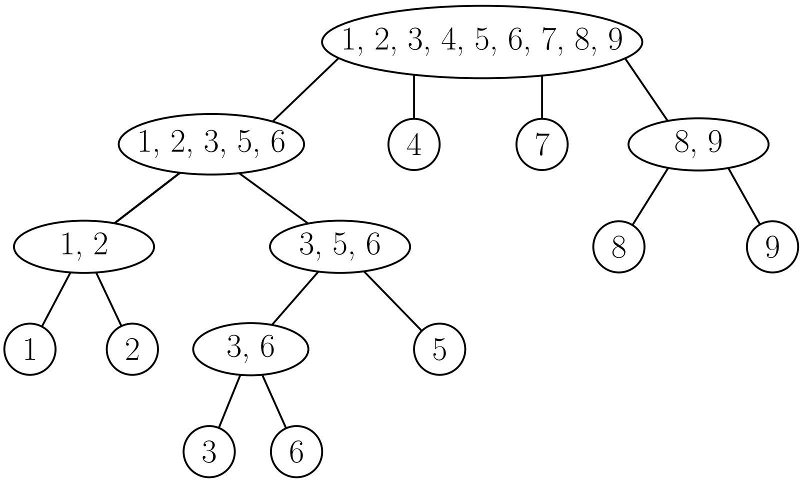

*Property b) is satisfied since $E' = \{14, 24, 34, 45, 46, 47, 78, 79\}$ (the red edges) is indeed an edge class of $G$ and every vertex in $V$ is incident to at most one edge in $E'$. Furthermore, the additional edge classes of $G$ are $\{12\}, \{35, 36\}$ and $\{89\}$, which are exactly the edge classes of $\{1, 2, 3, 5, 6\}$ and $\{8, 9\}$, hence also property c) is fulfilled. Finally, also property d) is satisfied, since $P$ contains all maximal modules of $G$.*

*This leads to the following partial modular decomposition tree:*




*Since the complement of the subgraph $[\{8, 9\}]_G$ is not connected, the children of $\{8, 9\}$ in $T$ are the singleton sets $\{8\}$ and $\{9\}$. $[\{1, 2, 3, 5, 6\}]_G$ also is not connected, hence the vertices of its connected components $\{1, 2\}$ and $\{3, 5, 6\}$ respectively, are the children of $\{1, 2, 3, 5, 6\}$ in $T$ .*

*Now, for $\{1, 2\}$ we get the children $\{1\}$ and $\{2\}$, while $\{3, 6\}$ and $\{5\}$ are the set of vertices of the connected components of $[\{3, 5, 6\}]_G$ and thus the children of $\{3, 5, 6\}$ in $G$. Finally, the children of $\{3, 6\}$ are $\{3\}$ and $\{6\}$, which gives us the following unique modular decomposition tree $T$ for $G$:*




Let $G = (V, E)$ be an undirected graph and let $T$ be the modular decomposition tree of $G$. By $\text{children}_T(\mu)$ we denote the family of children of $\mu$ in $T$ and $[\mu]_G/\text{children}_T(\mu)$ denotes the quotient graph whose vertices are the members of $\text{children}_T(\mu)$, where node $U$ and $Y$ are adjacent if and only if every $u \in U \subseteq V$ is adjacent to every $y \in Y \subseteq V$ in $[\mu]_G$. Note that when referring to a node $\mu$ in $T$ unless stated otherwise we mean the set of all vertices in $G$ that are represented by $\mu$ but it is also reasonable to refer to the quotient graph $[\mu]_G/\text{children}_T(\mu)$. Figure 2.7 illustrates this approach for the modular decomposition tree of Example 2.14.

For every node $\mu$ in $T$ every edge $UW$ in the quotient graph $[\mu]_G/\text{children}_T(\mu)$ represents a set of edges $E' := \{vw \in E \mid v \in U, w \in W\}$. Every edge $e \in E$ is represented by exactly one edge in one of the quotient graphs in $T$, namely in the quotient graph $[\mu]_G/\text{children}_T(\mu)$ such that the endpoints of $e$ appear in two distinct children of $\mu$. Note that $\mu$ is the lowest common ancestor of the two endpoints of $e$ in $T$. If $G$ is a comparability graph, then in

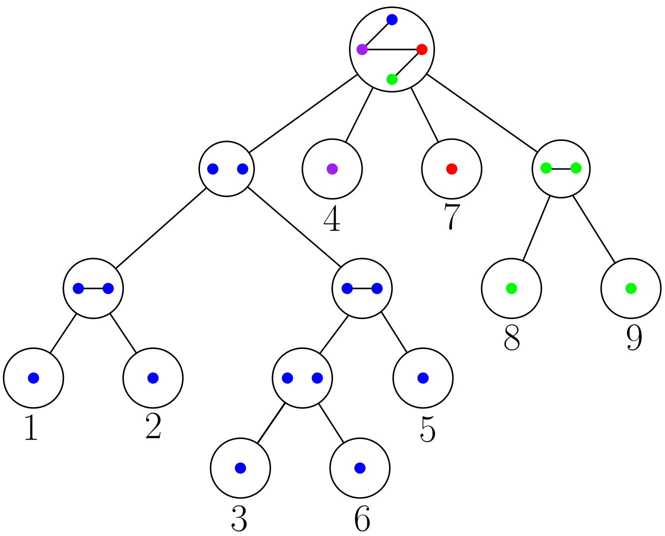
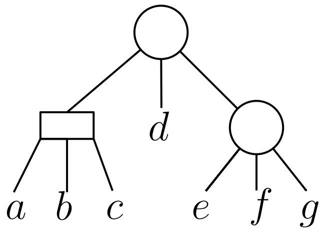



Figure 2.7: A version of the modular decomposition tree from Example 2.14 whose nodes represent quotient graphs.




(a) A $PQ$-tree $T$ over the set $U = \{a, b, c, d, e, f, g, h\}$.

| | | | |
|---|---|---|---|
| $abcdefg$ | $abcefgd$ | $cbadefg$ | $cbaefgd$ |
| $abcdegf$ | $abcegfd$ | $cbadegf$ | $cbaegfd$ |
| $abcdfeg$ | $abcfegd$ | $cbadfeg$ | $cbafegd$ |
| $abcdfge$ | $abcfged$ | $cbadfge$ | $cbafged$ |
| $abcdgfe$ | $abcgfed$ | $cbadgfe$ | $cbagfed$ |
| $abcdgef$ | $abcgefd$ | $cbadgef$ | $cbagefd$ |

(b) All permutations of the elements in $U$ represented by $T$.

Figure 2.8

every transitive orientation of $G$, the edges represented by one edge connecting module $\mu_1$ and $\mu_2$ in a quotient graph have to be oriented either all from the endpoint in $\mu_1$ to the endpoint in $\mu_2$ or all from the endpoint in $\mu_2$ to the endpoint in $\mu_1$. Hence the problem of finding a transitive orientation of $G$ reduces to finding a transitive orientation for the quotient graph $[\mu]_G/\text{children}_T(\mu)$ for every node $\mu$ in the modular decomposition tree $T$ of $G$. An *orientation* $\overrightarrow{T}$ of $T$ is an assignment of directions to all edges in every quotient graph corresponding to a node in $T$. Such an orientation $\overrightarrow{T}$ is transitive, if it is transitive on every quotient graph corresponding to a node in $G$. An orientation of $G$ *induces* an orientation of $T$ if for all nodes $\mu_1, \mu_2$ in $T$ all edges in $E$ with one endpoint in $\mu_1$ and one endpoint in $\mu_2$ are either all oriented towards their endpoint in $\mu_1$ or they are all oriented towards their endpoint in $\mu_2$. If so, in the former case the direction $\overrightarrow{\mu_2\mu_1}$, in the latter one $\overrightarrow{\mu_1\mu_2}$ is induced. An orientation of $T$ also induces an orientation of $G$. For every directed edge $\overrightarrow{\mu_1\mu_2}$ in a quotient graph corresponding to a node in $T$, we orient all edges in $E$ with one endpoint in $\mu_1$ and one endpoint in $\mu_2$ towards their endpoint in $\mu_2$. An orientation of $G$ is transitive if and only if it induces a transitive orientation of $T$. Similarly an orientation of $T$ is transitive if and only if it induces a transitive orientation of $G$ [Gal67].

A graph $G$ is *prime*, if there exist exactly two transitive orientations of $G$ where one is the reverse of the other. A node $\mu$ in a modular decomposition tree $T$ is called *prime*, if the corresponding quotient graph $[\mu]_G/\text{children}_T(\mu)$ is prime. Analogously $\mu$ is called *complete* or *empty* if $[\mu]_G/\text{children}_T(\mu)$ is complete or empty respectively. All quotient graphs in a modular decomposition tree are either prime, complete or empty [MS99]. In the following as a convention we assume that in every modular decomposition tree all prime nodes are labelled with one of the two transitive orientations of the corresponding quotient graph, referred to as *default orientation.* Note that with the algorithm presented by [MS99] we can compute such a default orientation in $\mathcal{O}(n+m)$, hence the time we need to compute a tree in which every prime node is labelled with a default orientation is also not greater than $\mathcal{O}(n+m)$.

## 2.5 PQ-trees

To achieve a linear runtime for the simultaneous representation problem for comparability graphs, we need a data structure called $PQ$-tree that allows us to represents all permissible permutations of the elements of a set $U$ in which certain subsets $S \subset U$ appear consecutively. $PQ$-trees were first introduced by Booth and Lueker [BL76]. An example of a $PQ$-tree over a set $U$ is given in Figure 2.8. Let $U = \{a_1, a_2, \ldots, a_n\}$ be a universal set. Then a $PQ$-tree $T$ over that set is a rooted tree whose leaves are elements of $U$ and whose internal nodes are either so called $P$- or $Q$- nodes. A $P$-node is drawn as a circle, a $Q$-node as a rectangle. Booth and Lueker defined a $PQ$-tree $T$ to be *proper* if it has the following properties.

1) Every element $a_i \in U$ appears exactly once as a leaf of $T$.

2) Every $P$-node has at least two children.

3) Every $Q$-node has at least three children.

The *frontier* $\textsc{Frontier}(T)$ of a $PQ$-tree $T$ is given by the order of its leaves from left to right. Two $PQ$-trees $T$ and $T'$ are *equivalent* ($T \equiv T'$), if and only if $T$ can be transformed into $T'$ by arbitrarily permute the children of arbitrarily many $P$-nodes and reversing the order of arbitrarily many $Q$-nodes. A transformation that transforms $T$ into an equivalent tree $T'$ is called an *equivalence transformation*. The tree $T$ represents exactly those permutations that can be obtained from $T$ by equivalence transformations. The set of all permutations represented by $T$ is denoted by $\textsc{Consistent}(T) = \{\textsc{Frontier}(T') \mid T' \equiv T\}$. The $PQ$-tree that does not have any nodes is called the *null tree*.

# 3. Related Work

## 3.1 Extending Partial Representations of Comparability and Permutation Graphs

Klavík et al. already showed that RepExt($Comp$) can be solved in $\mathcal{O}((n+m)\Delta)$ time for comparability graphs with $n$ vertices, $m$ edges and maximum degree $\Delta$ [KKKW12]. To do so, they modify the following $\mathcal{O}((n+m)\Delta)$ time recognition algorithm of Golumbic [Gol04]. Let $G$ be an input graph. The recognition algorithm in every step picks an arbitrary unoriented edge of $G=(V,E)$ and assigns a direction to it, which may force the orientation of several other edges. If two edges $e = uv$ and $f = vw$ share an endpoint $v$ and $e$ is oriented towards $v$, while $f$ is oriented towards $w$, this implies that the edge $uw$ must be in $E$ and that it has to be oriented toward $w$. Else if $e$ is already oriented, say towards $v$, but $f$ does not have an assigned direction yet and $uw \notin E$, then $f$ has to be oriented towards $v$. If we succeed to orient all edges in $E$ without having to reorient an already directed edge, $G$ is a comparability graph, else it is not. Let now $\overrightarrow{E}'$ be a partial orientation of $G$ and assume that $G$ is a comparability graph. For solving the partial representation problem, we first choose an order $e_1 < e_2 < \cdots < e_m$ of the $m$ edges in $G$ such that the first $k$ edges $e_1, \ldots, e_k$ are preoriented by $\overrightarrow{E}'$. Then we proceed as above with the only difference that instead of arbitrarily choosing the next edge to be oriented in each step, we always pick the first non-oriented edge in the chosen ordering. When choosing an edge $e_i$ with $i \leq k$ we orient it according to $\overrightarrow{E}'$ otherwise we orient it arbitrarily. This modified algorithm stops and returns that $\overrightarrow{E}'$ is not extendible to a transitive orientation of the entire graph $G$ if it is forced to orient an edge $e_i$ with $i \leq k$ in the opposite direction to the one induced by $\overrightarrow{E}'$. In this case, $\overrightarrow{E}'$ is not extendible, since the orientation of $e_i$ contradicting $\overrightarrow{E}'$ was forced by the orientation of $e_1, \ldots, e_{i-1}$. Else the algorithm returns a transitive orientation of $G$ that contains all orientations predefined by $\overrightarrow{E}'$, hence in this case $\overrightarrow{E}'$ is indeed extendible.

For RepExt($Perm$) Klavík et al. presented a $\mathcal{O}(n^3)$ time algorithm. Let $G$ be a permutation graph whose complement we denote by $\overline{G}$ and let $D'$ be a permutation diagram representing a subgraph $H$ of $G$. Then we orient every edge $e = uv$ in $G$ or $\overline{G}$ such that $u$ and $v$ both are vertices in $H$, towards the endpoint whose corresponding label along the upper line of $D'$ appears to the left of the other. This gives us partial orientations of $G$ and $\overline{G}$. Since $G$ is a permutation graph, both, $G$ and its complement are comparability graphs. Hence we can run the algorithm for solving RepExt($Comp$) for $G$ and $\overline{G}$. The algorithm fails if it is not possible to extend one of the two partial orientations to a transitive orientation of $G$ or $\overline{G}$ respectively. Else it returns two transitive orientations $\overrightarrow{E_1}$ of $G$ and $\overrightarrow{E_2}$ of $\overline{G}$ and we can construct a permutation diagram $D$ representing $G$ and extending $D'$ by labelling the upper line with $\overrightarrow{E_1} \cup \overrightarrow{E_2}$ and the bottom line with $\overleftarrow{E_1} \cup \overrightarrow{E_2}$ (where we received $\overleftarrow{E_1}$ by reverting all orientations in $\overrightarrow{E_1}$).

## 3.2 Solving the Simultaneous Representation Problem for Comparability and Permutation Graphs

Let $G_1, G_2, \ldots, G_r$ be $r$-sunflower comparability graphs, i.e. $r$ comparability graphs that pairwise share the same induced subgraph $H$, and let $G := G_1 \cup G_2 \cup \cdots G_r$. Then by $n$ we denote the number of vertices in $G$ and $m$ is the number of edges in $G$. Jampani and Lubiw presented an $\mathcal{O}(nm)$ algorithm to solve SimRep(*Comp*) for $G_1, \ldots, G_r$ [JL10]. To do so, they showed and made use of the following results.

First of all Jampani and Lubiw used the $\Gamma$ relation for directed edges (see Definition 2.11) to define a relation $\Gamma'$ on the directed edges of $G$: $\overrightarrow{e}\Gamma'\overrightarrow{f}$ if $\overrightarrow{e}\Gamma\overrightarrow{f}$ and $e$ and $f$ belong to the same edge class $E_i$ for an $i \in \{1, \ldots, r\}$. They denote the transitive closure of $\Gamma'$ which is an equivalence relation by $\Gamma'_t$ and call the corresponding equivalence classes *composite classes*. If such a composite class $C$ contains an edge of the shared graph $H$, this class is called a *super class*, else $C$ is a *base class*. For a class of directed edges $C$, we denote the set we receive by reverting all orientations of the edges in $C$ by $C^{-1}$.

An $S$-decomposition of $G = G_1 \cup G_2 \cup \cdots \cup G_r$ is a partition of the edge set $\hat{E}(G) = \hat{B}_1 + \hat{B}_2 + \cdots \hat{B}_i + \hat{S}_{i+1} + \hat{S}_{i+2} + \cdots + \hat{S}_j$ such that $B_k$ is a base class of $G - \cup_{1 \le l < k} \hat{B}_l$ for all $k \in \{1, \ldots, i\}$, and for all $k \in \{i+1, \ldots, j\}$, $S_k$ is a super class of $G - \cup_{1 \le l < i} \hat{B}_l - \cup_{i+1 \le l < k} \hat{S}_l$.

**Theorem 3.1.** *[JL10] Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_r = (V_r, E_r)$ be $r$-sunflower comparability graphs sharing some vertices $H$ and the edges induced by $H$. Let $G = G_1 \cup G_2 \cup \cdots \cup G_r$ and $\hat{E}(G) = \hat{B}_1 + \hat{B}_2 + \cdots \hat{B}_i + \hat{S}_{i+1} + \hat{S}_{i+2} + \cdots + \hat{S}_j$ be an $S$-decomposition of $G$. The following statements are equivalent.*

1) *$G_1, G_2, \ldots, G_r$ are simultaneous comparability graphs.*
2) *Every composite class of $G$ is pseudo-transitive, i.e. $C \cap C^{-1} = \emptyset$ for all composite classes $C$ of $G$.*
3) *Every part of the $S$-decomposition is pseudo-transitive, i.e. $B_k \cap B_k^{-1} = \emptyset$ for $k = 1, \ldots, i$ and $S_k \cap S_k^{-1} = \emptyset$ for $k = i+1, \ldots, j$.*

Theorem 3.1 implies that we can determine whether $r$-sunflower graphs $G_1, G_2, \ldots, G_r$ are simultaneous comparability graphs, by checking whether all composite classes of $G = G_1 \cup G_2 \cup \cdots \cup G_r$ are pseudo-transitive. If existent, we can compute a pseudo-transitive orientation, i.e. an orientation of $G$ that is transitive on every $G_i$ for $1 \le i \le r$, by first stepwise removing the base classes from $G$ and recursively orienting the remaining graph until no base class is left and then stepwise removing the super classes and again recursively orienting the remaining graph, until all edges are removed.

To solve SimRep(*Perm*) in $\mathcal{O}(n^3)$ time [JL10], Jampani and Lubiw used that $r$-sunflower permutation graphs are simultaneous permutation graphs if and only if they are simultaneous comparability graphs and simultaneous co-comparability graphs. Hence they apply their algorithm to solve SimRep(*Comp*) to the $r$ input graphs and afterwards to their complements.

# 4. Comparability graphs

In this chapter we give efficient algorithms for solving the *partial representation problem* and the *simultaneous representation problem* for comparability graphs.

## 4.1 Extending Partial Representations

The *partial representation problem* for comparability graphs RepExt(Comp) is to decide for a comparability graph $G = (V, E)$ given a partial orientation $\overrightarrow{W}$, i.e. a transitive orientation of some of its edges, whether it is possible to orient the remaining edges in a way that we get a transitive orientation of the entire graph. As we will see this is the case if and only if for every quotient graph in the modular decomposition tree $T$ of $G$ the partial orientation induced by $\overrightarrow{W}$ can be extended to a transitive orientation of the entire quotient graph. An oriented edge $uv \in \overrightarrow{W}$ induces that the edge between the modules $U$ and $V$ in a quotient graph, such that $u \in U$ and $v \in V$, is oriented from $U$ to $V$. In the following let $G$ be a comparability graph with $n$ vertices and $m$ edges.

If a vertex $v \in V$ is contained in a node $\mu$ in the modular decomposition tree $T$, we denote the child of $\mu$ containing $v$ by $\mathrm{rep}_T^{\mu}(v)$. Every edge $e \in E$ is represented by exactly one edge $e'$ in exactly one quotient graph of $T$. The node corresponding to this quotient graph in $T$ we denote by $\mathrm{rep}_T(e)$. Note that for $e = uv$ with $u, v \in V$, $\mathrm{rep}_T(e)$ is the lowest common ancestor of the leaves corresponding to $u$ and $v$ respectively in $T$. Additionally to the set of represented vertices of $G$ and the corresponding quotient graph, for every node $\mu$ in a modular decomposition tree $T$ we also want to store the set of edges in $E$ that are represented by the edges in $[\mu]_G/\mathrm{children}_T(\mu)$. An edge $e = uv \in E$ with $u, v \in V$ is represented by an edge in the quotient graph $[\mu]_G/\mathrm{children}_T(\mu)$, if and only if $u, v \in \mu$ and $u \in \mu_1 \neq \mu_2 \ni v$ for $\mu_1, \mu_2 \in \mathrm{children}(\mu)$. This means that $\mu$ is the lowest common ancestor of $u$ and $v$ in $T$. Tarjan presented an algorithm using the union-find data structure to compute the lowest common ancestors of all pairs of nodes in a fixed given set [Tar79] and later proves that this can be done in linear time [GT85]. Applied to our case, this means that we can compute the lowest common ancestors of every pair of vertices $u, v \in V$ such that there exists an edge $e \in E$ whose endpoints are $u$ and $v$ (which are exactly $m$ pairs) in $\mathcal{O}(m)$ time. Now we store for every node $\mu \in T$ all pairs of nodes $uv$, i.e. the edges between those two vertices, such that $\mu$ is the lowest common ancestor of $u$ and $v$ in $T$. This allows us to determine $\mathrm{rep}_{\mathrm{T}}(\mathrm{e})$ for every $e \in E$ in constant time, after the described preprocessing that takes $\mathcal{O}(m)$ time.

**Lemma 4.1.** *Let $G$ be a comparability graph and let $\overrightarrow{W}$ be a partial orientation of $G$. Then $\overrightarrow{W}$ can be extended to a transitive orientation of the entire graph $G$ if and only if the induced partial orientations for the prime and complete quotient graphs in the modular decomposition tree for $G$ all can be extended to a transitive orientation of the whole quotient graph.*

*Proof.* Assume that $\overrightarrow{W}$ can be extended to a transitive orientation of the entire graph $G$ and let $\overrightarrow{E}$ denote such a transitive orientation of $G$. Then $\overrightarrow{E}$ induces a transitive orientation $\overrightarrow{T}$ of the modular decomposition tree $T$ for $G$ that consists of transitive orientations for the complete and prime quotient graphs in $T$ containing the predefined orientations induced by $\overrightarrow{W}$ since $\overrightarrow{E}$ is an extension of $\overrightarrow{W}$.

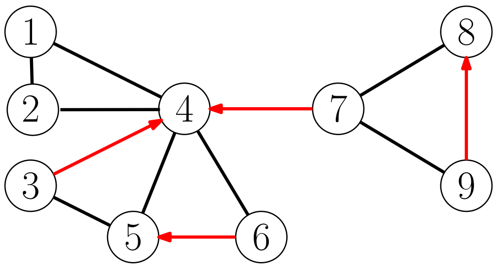
Now assume that $\overrightarrow{W}$ induces partial orientations for the prime and complete quotient graphs in the modular decomposition tree $T$ for $G$ such that all these orientations can be extended to a transitive orientation of the whole quotient graph respectively. Then we receive a transitive orientation $\overrightarrow{T}$ of $T$ by combining the transitive orientations of the prime and complete quotient graphs. Note that we do not need to consider the empty quotient graphs since they do not contain any edge and hence their transitive orientations are also empty. Since $\overrightarrow{T}$ contains the predefined orientations induced by $\overrightarrow{W}$, $\overrightarrow{T}$ induces a transitive orientation of the entire graph $G$ extending $\overrightarrow{W}$. □

**Theorem 4.2.** *The problem REPEXT(COMP) can be solved in $\mathcal{O}(n+m)$ time for permutation graphs with $n$ vertices and $m$ edges.*

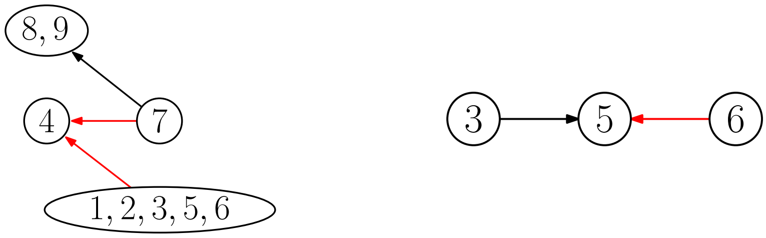
*Proof.* By Lemma 4.1 to solve REPEXT(COMP) for a comparability graph $G$ it suffices to solve REPEXT(COMP) for the prime and complete quotient graphs in the modular decomposition $T$ of $G$. Let $\overrightarrow{W}$ be the given partial orientation of $G$. For a complete quotient graph in $T$ we need to find a linear order $<$ of the nodes of the quotient graph that respects the partial order induced by the edges whose orientation is induced by the given partial orientation of $G$. To do so, we extend the induced partial order in $\mathcal{O}(n_i + m_i)$ time with the method presented by Kahn [Kah62] where $n_i$ and $m_i$ are the number of nodes and edges of the complete quotient graph. Recall that every edge in $G$ is represented by exactly one edge in exactly one quotient graph in $T$. Since the modular decomposition tree $T$ has $n$ leaves and thus at most $2n-1$ nodes, in total we need $\mathcal{O}(n+m)$ time to compute linear orders with the desired properties for all complete quotient graphs. Afterwards we orient the edges in the complete quotient graphs according to the computed linear orders.

A prime quotient graph has exactly two transitive orientations, where one is the reverse of the other. Therefore we first orient the edges that are induced by $\overrightarrow{W}$ and afterwards choose one of the two transitive orientations of the quotient graph. Now we orient the edges according to the chosen orientation. If we have to reverse the orientation of a preoriented edge, we try again with the only other existent transitive orientation of the prime quotient graph. If both transitive orientations are not compatible with the partial orientation, $\overrightarrow{W}$ is not extendible to a transitive orientation of the entire graph. Else, if we can orient all prime quotient graphs according to one of their two transitive orientations without reversing an already oriented edge, the orientations of the quotient graphs together induce a transitive orientation on the entire graph $G$ that is an extension of $\overrightarrow{W}$. Since we can compute the modular decomposition tree $T$ in $\mathcal{O}(n+m)$ time, in total we can decide whether the partial orientation $\overrightarrow{W}$ is extendible and if so find a corresponding transitive orientation of the entire graph in the same time. □

**Example 4.3.** *Consider the following partial orientation $\overrightarrow{W}$ of the comparability graph from Example 2.14.*

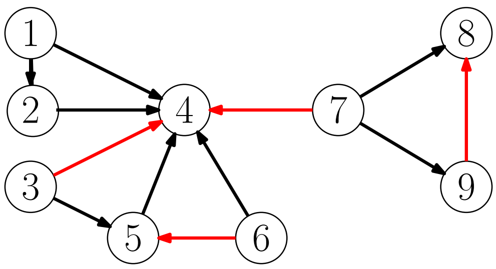



*Note that we regard a quotient graph with only one edge as complete and not as prime. Then for the two prime quotient graphs we receive the following extensions of the edges induced by $\overrightarrow{W}$.*




*One of the two complete quotient graphs in the modular decomposition tree of $T$ is already entirely oriented by $\overrightarrow{W}$. For the other one the induced orientation is empty, hence we may orient the edge between vertex 1 and 2 arbitrarily.*

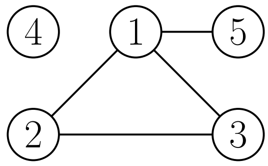

*Hence all by $\overrightarrow{W}$ induced partial orientations of prime or complete quotient graphs can be extended to transitive orientations of the entire quotient graph. Thus $\overrightarrow{W}$ can also be extended to a transitive orientation of $G$, which is induced by the chosen orientations of the quotient graphs.*



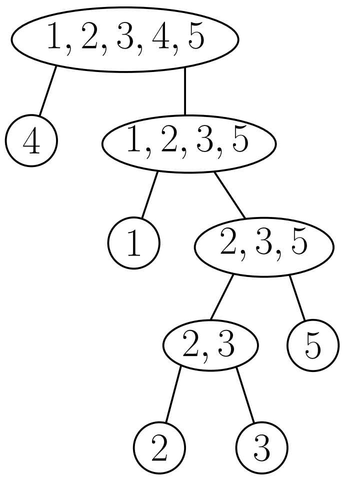

## 4.2 The Simultaneous Representation Problem

Let $G$ be a comparability graph and let $H = (V, E)$ be an induced subgraph of $G$. For the graph $G$ we define a modular decomposition tree $T$ reduced to the vertices in $V$ as follows: Let $B$ be the modular decomposition tree for $G$. Then we get the desired *reduced tree* by first removing all nodes $\mu$ from $B$ that contain only vertices that do not belong to the subgraph $H$, i.e. $\mu \cap V = \emptyset$. If the resulting tree contains nodes that only have one child, we may delete this nodes from the tree and append their children to their parent nodes. In a second step we remove all vertices from the remaining nodes that are not contained in $V$. This means every node $\mu$ is replaced by $\mu \cap V$. For a node $\mu$ in $T$ let $\nu \in B$ be the lowest common ancestor of all vertices contained in $\mu$. Note that by construction $\nu \setminus \mu$ only contains vertices not in $H$. Furthermore the children of $\mu$ are the children of $\nu$ that contain vertices in $H$ reduced to $\nu \cap \mu$. Hence the quotient graph $[\mu]_G/\text{children}_T(\mu)$ is an induced substructure of $[\nu]_G/\text{children}_B(\nu)$. In particular if $[\nu]_G/\text{children}_B(\nu)$ is complete or empty, then the same holds for $[\mu]_G/\text{children}_T(\mu)$. If $[\nu]_G/\text{children}_B(\nu)$ is prime, $[\mu]_G/\text{children}_T(\mu)$ can be an arbitrary graph. All nodes labelled as prime in $B$ are also treated as prime in $T$ and labelled with one of the two transitive orientations of the corresponding quotient graph reduced to the remaining nodes. We also reduce the set that contains all edges represented by the quotient graph of $\mu$ to the edges in $E$ for every $\mu$ in the reduced tree. The resulting tree whose leaves are exactly the singleton sets of the vertices in $V$ we denote by $T$.

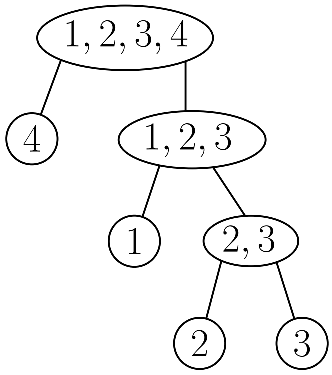
**Example 4.4.** *Let $G$ be the following graph.*




*Then the modular decomposition tree $B$ is given by:*




*The following tree is the modified modular decomposition tree reduced to the subgraph induced by the vertices* 1, 2, 3 *and* 4.



An orientation $\overrightarrow{E}$ of the subgraph $H$ of $G$ is induced by the reduced modular decomposition tree $T$ if and only if there exists a transitive orientation $\overrightarrow{T}$ of $T$ such that for every directed edge $\overrightarrow{uv}$ in $\overrightarrow{E}$, $\overrightarrow{T}$ contains the corresponding directed edge $\overrightarrow{\mathrm{rep}_T^\mu(u)\mathrm{rep}_T^\mu(v)}$ where $\mu = \mathrm{rep}_T(uv)$ is the lowest common ancestor of $u$ and $v$ in $T$.

**Lemma 4.5.** *All orientations of $H$ induced by $T$ are also transitive.*

*Proof.* Let $\overrightarrow{E}$ be an orientation of $H = (V, E)$ induced by $T$. Let $u, v, w$ be three vertices in $V$ such that $uv, vw \in E$. Assume that $\overrightarrow{E}$ contains $\overrightarrow{uv}$ and $\overrightarrow{vw}$. Let $\mu \in T$ be the lowest common ancestor of $u, v$ and $w$. Now we distinguish several cases. First let us assume that $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u}) \neq \mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})$, $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u}) \neq \mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})$ and $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v}) \neq \mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})$. Since $\overrightarrow{E}$ contains $\overrightarrow{uv}$ and $\overrightarrow{vw}$, there exists a transitive orientation $\overrightarrow{\mu}$ of $[\mu]_H/\mathrm{children}_T(\mu)$ that contains the directed edges $\overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})}$ and $\overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})}$. Since $\overrightarrow{\mu}$ is transitive, it contains also the edge $\overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})}$, which implies that also $uw \in E$ and $\overrightarrow{uw} \in \overrightarrow{E}$.

Next we assume $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u}) = \mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})$. This leads to a contradiction since to induce $\overrightarrow{uv}$ and $\overrightarrow{vw}$, the orientation $\overrightarrow{\mu}$ of $\mu$ must contain $\overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})}$ and $\overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})} = \overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u})}$. Hence we consider the case $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u}) \neq \mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})$, $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u}) = \mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})$. To induce $\overrightarrow{vw}$ our transitive orientation of $\mu$ must contain $\overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})} = \overrightarrow{\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u})\mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})}$ which implies that also $uw \in E$ and $\overrightarrow{uw} \in \overrightarrow{E}$. The last case $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{u}) \neq \mathrm{rep}^\mu_\mathrm{T}(\mathrm{w})$, $\mathrm{rep}^\mu_\mathrm{T}(\mathrm{v}) = \mathrm{rep}^\mu_\mathrm{T}(\mathrm{v})$ is analogue to the previous one. □

**Lemma 4.6.** *Let $G$ be a graph with modular decomposition tree $B$ and let $H = (V, E)$ be an induced subgraph of $G$. The reduced modular decomposition tree $T$ induces exactly those transitive orientations of the subgraph $H$, that can be extended to a transitive orientation of the entire graph $G$.*

*Proof.* The modular decomposition tree $B$ of $G$ induces exactly the transitive orientations of $G$. For every node $\mu$ in $T$ let $\nu$ be the lowest common ancestor of the vertices in $\mu$ in $B$. By construction of $T$ we know that the quotient graph $[\mu]_G/\mathrm{children}_T(\mu)$ is an induced substructure of $[\nu]_G/\mathrm{children}_B(\nu)$. We already know that $[\nu]_G/\mathrm{children}_B(\nu)$ can be either prime, complete or empty. In the latter case $[\mu]_G/\mathrm{children}_T(\mu)$ and the corresponding transitive orientation are also empty. Thus we only have to distinguish the cases in which $[\nu]_G/\mathrm{children}_B(\nu)$ is prime or complete. If $[\nu]_G/\mathrm{children}_B(\nu)$ is complete, any transitive orientation of $[\mu]_G/\mathrm{children}_T(\mu)$ can be extended to a transitive orientation of $[\nu]_G/\mathrm{children}_B(\nu)$, since $[\mu]_G/\mathrm{children}_T(\mu)$ is also a clique. In the case when $[\nu]_G/\mathrm{children}_B(\nu)$ is prime, by construction $[\mu]_G/\mathrm{children}_T(\mu)$ is also labelled as prime with the default orientation of $[\nu]_G/\mathrm{children}_B(\nu)$ reduced to $[\mu]_G/\mathrm{children}_T(\mu)$. Hence every transitive orientation of $[\mu]_G/\mathrm{children}_T(\mu)$ induces a partial transitive orientation of $[\nu]_G/\mathrm{children}_B(\nu)$ that is contained in its default orientation and hence this partial orientation can be extended to a transitive orientation of $[\nu]_G/\mathrm{children}_B(\nu)$. Thus every transitive orientation of the quotient graphs in $T$ induces a partial orientation of $B$ that can be extended to a transitive orientation of $B$. On the other hand every transitive orientation of $H$ that can be extended to a transitive orientation of $G$ induces a partial orientation of $B$ that in turn induces a transitive orientation of $T$. □

Let $S$ be the modular decomposition tree for $H$. The aim is now to examine which constraints a transitive orientation of the quotient graphs in $S$ must fulfil in order to induce a transitive orientation on $H$ that can be extended to a transitive orientation of $G$. An orientation of an edge $\nu_1\nu_2$ in a quotient graph of $S$ induces an orientation of all edges in

$H$ that are represented by $\nu_1\nu_2$, i.e. one of their endpoints is contained in $\nu_1$, the other one in $\nu_2$. Recall that a transitive orientation of all quotient graphs corresponding to $S$ induces a transitive orientation of $H$.

**Lemma 4.7.** *Let $\overrightarrow{T}$ be a transitive orientation for $T$. Then $\overrightarrow{T}$ induces a unique transitive orientation of $S$.*

*Proof.* By construction of $T$ we know that $\overrightarrow{T}$ can be extended to a transitive orientation $\overrightarrow{B}$ of the modular decomposition tree $B$ for $G$. The transitive orientation $\overrightarrow{B}$ induces a transitive orientation $\overrightarrow{E_G}$ of $G$ that is in particular transitive on the induced subgraph $H$. Hence $\overrightarrow{E_G}$ reduced to $H$ induces a transitive orientation of $S$. $\square$

Let $\mu_S$ be a complete node in $S$ with $\text{children}_S(\mu_S) = \{\nu_1, \nu_2, \ldots, \nu_k\}$ and let $\mu_T$ be a node in $T$. Now we show that, if $\mu_T$ contains vertices belonging to two different children of $\mu_S$, then $\mu_T$ already contains all vertices in these two children.

**Lemma 4.8.** *Let $\mu_S$ be a complete node in $S$ with* $\text{children}_S(\mu_S) = \{\nu_1, \nu_2, \ldots, \nu_k\}$*. Then for every $\mu_T \in T$ and every $1 \leq i \leq k$, if $\mu_T \cap \nu_i \neq \emptyset$ and there exists a $j \in \{1, 2, \ldots, k\}$ with $j \neq i$ and $\mu_T \cap \nu_j \neq \emptyset$, then $\nu_i \subseteq \mu_T$.*

*Proof.* Let $\mu_S \in S$ with $\nu_1, \nu_2 \in \text{children}(\mu_S)$ and let $u \in \nu_1$ and $v \in \nu_2$ be two vertices of the subgraph $H$. Furthermore let $\mu_T$ be the lowest common ancestor of $u$ and $v$ in $T$. Assume that there exists a vertex $w \in \nu_1$ such that $w \notin \mu_T$. Let $\mu$ be the lowest common ancestor of $u$ and $w$ in $T$. Then we know that $\mu_1 := \text{rep}_T^{\mu}(u) = \text{rep}_T^{\mu}(v) \neq \text{rep}_T^{\mu}(w) =: \mu_2$. Since $S$ is complete $[\mu_T]_H/\text{children}_T(\mu_T)$ contains an edge between $\text{rep}_T^{\mu_T}(u)$ and $\text{rep}_T^{\mu_T}(v)$. Furthermore $[\mu]_H/\text{children}_T(\mu)$ also contains an edge between $\mu_1$ and $\mu_2$. By Lemma 4.7 any transitive orientations of $T$ induces a transitive orientation of $S$. But there exists a transitive orientation of $[\mu_T]_H/\text{children}_T(\mu_T)$ that contains $\overrightarrow{\text{rep}_T^{\mu_T}(u)\text{rep}_T^{\mu_T}(v)}$ and we can transitively orient $[\mu]_H/\mathit{children}_T(\mu)$ such that the edge between $\mu_1$ and $\mu_2$ is directed towards $\mu_2$. This orientation can not be mapped to $\mu_S$ since all edges between vertices in $\nu_1$ and $\nu_2$ either have to be all oriented towards their endpoint in $\nu_1$ or they are all oriented towards their endpoint in $\nu_2$. Hence we have $w \in \mu_T$ for every $w \in \nu_1$. $\square$

Let $\mu_S$ be a complete node in $S$ and let $\mu_T$ be the lowest common ancestor in $T$ of all vertices contained in $\mu_S$. Then we denote the subtree of $T$ with root $\mu_T$ by $T[\mu_T]$. For every inner node $\nu$ in $T[\mu_T]$ let $M_\nu \subseteq \mathit{children}(\mu_S)$ be the set that contains exactly those children $\mu_i$ of $\mu_S$ such that $\mu_i \cap \nu \neq \emptyset$. If $|M_\nu| > 1$, by Lemma 4.8 every $\mu_i \in M_\nu$ is fully contained in $\nu$. We say that a transitive orientation $\overrightarrow{\mu_S}$ of $[\mu_S]_H/\text{children}_S(\mu_S)$ *fulfils the consecutiveness constraints*, if $\overrightarrow{\mu_S}$ induces an orientation of $[\mu_T]_H/\text{children}_T(\mu_T)$ such that for every node $\nu$ in $T[\mu_T]$ the edge between a node $\nu_S \in \text{children}(\mu_S) \setminus M_\nu$ and a node in $M_\nu$ are either all oriented towards their endpoint in $M_\nu$ or they are all oriented towards $\nu_S$. This term comes from the fact that if $\overrightarrow{\mu_S}$ fulfils the consecutiveness constraints, in the total order of the children of $\mu_S$ induced by $\overrightarrow{\mu_S}$, for every node $\nu$ in $T[\mu_T]$ all nodes in $M_\nu$ appear consecutively.

**Lemma 4.9.** *A transitive orientation $\overrightarrow{\mu_S}$ of a complete node $\mu_S$ in $S$ induces a partial orientation of $T$ if and only if $\overrightarrow{\mu_S}$ fulfils the consecutiveness constraints.*

*Proof.* Let $\nu$ be a node in $T[\mu_T]$ and let $\nu_1, \nu_2, \nu_3 \in \text{children}(\mu_S)$ such that $\nu_1, \nu_2 \in M_\nu$ but $\nu_3 \notin M_\nu$. Let $\overrightarrow{\mu_S}$ be a transitive orientation of $[\mu_S]_H/\text{children}_S(\mu_S)$ containing the directed edges $\overrightarrow{\nu_1\nu_3}$ and $\overrightarrow{\nu_3\nu_2}$. Then $\overrightarrow{\mu_S}$ does not fulfil the consecutiveness constraints. Let $u_1 \in \nu_1$, $u_2 \in \nu_2$ and $u_3 \in \nu_3$ be vertices in $H$. Since $\mu_S$ is complete we have $u_1u_2, u_2u_3, u_1u_3 \in E$. The partial orientation $\overrightarrow{E}$ of $H$ induced by $\overrightarrow{\mu_S}$ contains $\overrightarrow{u_1u_3}$ and $\overrightarrow{u_3u_2}$. We already know that $\nu_1, \nu_2 \subseteq \nu$ and $\nu_3 \nsubseteq \nu$. Hence the lowest common ancestor $\nu'$ in $T$ of $\nu_1$ and $\nu_3$ is also the lowest common ancestor of $\nu_2$ and $\nu_3$ and $\mathrm{rep}_T^{\nu'}(u_1) = \mathrm{rep}_T^{\nu'}(u_2)$. Since $u_1u_3 \in E$ the quotient graph $[\nu']_H/\mathrm{children}_T(\nu')$ contains an edge $\mathrm{rep}_T^{\nu'}(u_1)\mathrm{rep}_T^{\nu'}(u_3) = \mathrm{rep}_T^{\nu'}(u_2)\mathrm{rep}_T^{\nu'}(u_3)$. Now $\overrightarrow{E}$ induces both $\overrightarrow{\mathrm{rep}_T^{\nu'}(u_1)\mathrm{rep}_T^{\nu'}(u_3)}$ and $\overleftarrow{\mathrm{rep}_T^{\nu'}(u_1)\mathrm{rep}_T^{\nu'}(u_3)}$. Hence the transitive orientation $\overrightarrow{\mu_S}$ can not be mapped to $T$.

Now assume that $\overrightarrow{\mu_S}$ fulfils the consecutiveness constraints. Let $\nu_1, \nu_2 \in \mathrm{children}(\nu)$ for a node $\nu$ in $T[\mu_T]$, such that the quotient graph $[\nu]_H/\mathrm{children}_T(\nu)$ contains an edge $\nu_1\nu_2$. Then for $i \in \{1, 2\}$ every edge between a node $\mu_i$ not in $M_{\nu_i}$ and the nodes in $M_{\nu_i}$ in $[\mu_S]_H/\mathrm{children}_S(\mu_S)$ are either all oriented towards $\mu_i$ or they are all oriented towards their endpoint in $M_{\nu_i}$. This especially holds for $\mu_1 \in M_{\nu_2}$ and $\mu_2 \in M_{\nu_1}$. Hence in total we get that the edges in $[\mu_S]_H/\mathrm{children}_S(\mu_S)$ between a node in $M_{\nu_1}$ and a node in $M_{\nu_2}$ are either all oriented towards their endpoint in $M_{\nu_1}$ or they are all oriented towards their endpoint in $M_{\nu_2}$. Since this holds for every $\nu_1, \nu_2 \in \mathrm{children}(\nu)$ for a node $\nu$ in $T[\mu_T]$, such that the quotient graph $[\nu]_H/\mathrm{children}_T(\nu)$ contains an edge $\nu_1\nu_2$, the transitive orientation $\overrightarrow{\mu_S}$ induces a partial orientation of $T$. □

Let $\nu$ be a prime node in $T[\mu_T]$ and let $\nu_1$ and $\nu_2$ be two children of a complete node $\mu_S \in S$ with $u_1, v_1 \in \nu_1$, $u_2 \in \nu_2$. Recall that a transitive orientation of $T$ induces a transitive orientation of $S$. All edges in $H$ between vertices in $\nu_1$ and vertices in $\nu_2$ are represented by exactly one edge in $S$. Hence a transitive orientation of the quotient graph of $\nu$ induces a partial orientation of $S$. As a result no transitive orientation of $[\nu]_H/\mathrm{children}_T(\nu)$ can contain both $\overrightarrow{\mathrm{rep}_T^{\nu}(u_1)\mathrm{rep}_T^{\nu}(u_2)}$ and $\overrightarrow{\mathrm{rep}_T^{\nu}(u_2)\mathrm{rep}_T^{\nu}(v_1)}$. In other words all edges between the representative of a vertex in a child $\nu_1$ of $\mu_S$ and the representative of a vertex in another child $\nu_2$ of $\mu_S$ are either all oriented towards the representative of a vertex in $\nu_1$ or they are all oriented towards the representative of a vertex in $\nu_2$. If we choose a representative for an arbitrary vertex in every child of $\mu_S$, they are fully connected in $[\nu]_H/\mathrm{children}_T(\nu)$ and hence the default orientation of the quotient graph induces a total order on them.

**Lemma 4.10.** *Let $\mu_S$ be a prime node in $S$ and let $\mu_T \in T$ be the lowest common ancestor of all vertices in $\mu_S$. Then $\mu_T$ is also prime and for every $\nu \in \mathrm{children}(\mu_S)$ there exists a $\nu' \in \mathrm{children}(\mu_T)$ such that $\nu \subseteq \nu'$.*

*Proof.* Note that $[\mu_S]_H$ and $\overline{[\mu_S]_H}$ are connected. Since $\mu_T$ is the lowest common ancestor of the vertices in $\mu_S$ in $T$, there exists no children $\mu$ of $\mu_T$ such that $\mu_S \subseteq \mu$. Hence also $[\mu_T]_H$ and $\overline{[\mu_T]_H}$ are connected, which already implies that $\mu_T$ is prime. Recall that the edges in $H$ represented by the edges in a prime quotient graph $[\mu_T]_H$ and $\overline{[\mu_T]_H}$ all belong to the same edge class and the remaining edge classes of $[\mu_T]_H$ are exactly the edge classes of the subgraphs induced by the children of $\mu_T$. Since $[\mu_S]_H$ is an induced subgraph of $[\mu_T]_H$, edges belonging to the same edge class in $[\mu_S]_H$ also belong to the same edge class in $[\mu_T]_H$. Hence vertices belonging to the same child of $\mu_S$ also belong to the same child of $\mu_T$. □

Now, we want to label each complete node $\mu_S$ in $S$ with a $PQ$-tree $B_\mu$ whose leaves are the nodes in the quotient graph $[\mu_S]_H/\mathrm{children}_S(\mu_S)$ such that $B_\mu$ represents all total orders inducing partial transitive orientations of $H$ that can be extended to transitive orientations of $G$. Additionally we construct a 2-SAT-formula $\varphi_G$ that synchronizes these $PQ$-trees with each other and the orientations of the prime quotient graphs in $S$. Initially $\varphi_G$

contains the constraint ($\overrightarrow{\nu_T} \Leftrightarrow \overrightarrow{\nu_S}$) for every prime node $\nu_S$ in $S$ such that $\nu_T$ is the lowest common ancestor of the vertices contained in $\nu_S$ in $T$ and $\overrightarrow{\nu_S}$ is the default orientation of the corresponding quotient graph $\nu_S$ is labelled with, while $\overrightarrow{\nu_T}$ is the orientation of $[\nu_T]_H/\text{children}_T(\nu_T)$ containing $\overrightarrow{\nu_S}$. Note that by Lemma 4.10, $\mu_T$ is labelled as prime. In case $\overrightarrow{\nu_T}$ is not the default orientation of $\nu_T$, we label $\nu_T$ with $\overrightarrow{\nu_T}$. Note that to add a constraint of the form ($a \Leftrightarrow b$) to a 2-SAT-formula we add the clauses ($a \vee \neg b$) and ($\neg a \vee b$).

Let $\mu_T$ be the lowest common ancestors in $T$ of the vertices in $\mu_S$. To construct $B_{\mu_S}$ consider $T[\mu_T]$ and modify it as follows. First we remove all nodes $\nu$ from the tree that do not share a represented edge in $H$ with $\mu_S$. Afterwards we append to every leaf in the resulting tree the children of $\mu_S$ that are endpoints of edges in $[\mu_S]_H/\text{children}_T(\mu_S)$ representing an edge that is also represented in the quotient graph of the currently considered node in $T[\mu_T]$. Now we replace all inner nodes whose corresponding quotient graph is complete or empty by a $P$-node. All remaining inner prime nodes $\nu$ of $T[\mu_T]$ become $Q$-nodes $q_\nu$ and their children are appended according to the total order on the elements of $M_\nu$ induced by the default orientation of $\nu$. Additionally we add the constraint ($\overrightarrow{\nu} \Leftrightarrow q_\nu$) to the 2-SAT-formula $\varphi_G$, where the variable $q_\nu$ represents the default order of the children of $q_\nu$.

**Lemma 4.11.** *A transitive orientation $\overrightarrow{S}$ of $S$ induces transitive orientations for all quotient graphs $[\mu_T]_H/\text{children}_T(\mu_T)$ for all prime-labelled nodes $\mu_T$ in $T$ if and only if all prime and complete quotient graphs corresponding to nodes in $S$ are oriented according to a solution of the 2-SAT-formula $\varphi_G$.*

*Proof.* Assume that the transitive orientation $\overrightarrow{S}$ induces transitive orientations for all quotient graphs $[\mu_T]_H/\text{children}_T(\mu_T)$ for all prime-labelled nodes $\mu_T$ in $T$. Then the orientations of the prime and complete quotient graphs corresponding to nodes in $S$ that induce a direction for an edge in such a $[\mu_T]_H/\text{children}_T(\mu_T)$ either all induce directed edges contained in the default orientation of $[\mu_T]_H/\text{children}_T(\mu_T)$ and no edge in the reverse default orientation or the other way round. By construction of $\varphi_G$ the variable assignment induced by $\overrightarrow{S}$ and the resulting orientation for $[\mu_T]_H/\text{children}_T(\mu_T)$ is satisfying.

Now assume that all prime and complete quotient graphs corresponding to nodes in $S$ are oriented according to a solution of the 2-SAT-formula $\varphi_G$. Then by construction of $\varphi_G$ the orientation $\overrightarrow{S}$ induces either the default orientation of $[\mu_T]_H/\text{children}_T(\mu_T)$ or its reverse, which are both transitive. □

**Lemma 4.12.** *Let $\mu_T$ be a complete node in $T$ and let $\mu_S \in S$ be the lowest common ancestor of all vertices in $\mu_T$. Then $\mu_S$ is also complete and for every $\nu \in \text{children}(\mu_T)$ there exists a $\nu' \in \text{children}(\mu_S)$ such that $\nu \subseteq \nu'$.*

*Proof.* Note that $[\mu_T]_H$ is connected but $\overline{[\mu_T]}_H$ is not. Since $[\mu_T]_H$ is a subgraph of $[\mu_S]_H$ and $\mu_S \in S$ is the lowest common ancestor of the vertices in $\mu_T$, we know that $\mu_S$ is not empty. Recall that $\mu_T$ is a module in $H$, hence all vertices not in $\mu_T$ are either adjacent to all vertices in $\mu_T$ or to none of them. If $\mu_S$ would contain a vertex $v$ not adjacent to any vertex in $\mu_T$, then $\mu_S$ would be disconnected and all vertices of $\mu_T$ would belong to the same child of $\mu_S$. This contradicts the fact that $\mu_S$ is the lowest common ancestor of all vertices in $\mu_T$. Hence all vertices in $\mu_S \setminus \mu_T$ must be adjacent to all vertices in $\mu_T$. Thus the nodes of $[\mu_T]_H/\text{children}_T(\mu_T)$ are also connected components of $\overline{[\mu_T]}_H$ and all connected components of $\overline{[\mu_S]}_H$ are fully connected in $[\mu_S]_H$. □

**Theorem 4.13.** *Let $\overrightarrow{S}$ be a transitive orientation of $S$. Then $\overrightarrow{S}$ induces a transitive orientation of the subgraph $H$ that can be extended to a transitive orientation of the entire* *graph $G$, if and only if every complete quotient graph in $S$ is oriented according to a total order induced by the corresponding PQ-tree and the orientation of all non-empty quotient graphs are compatible with a solution of the 2-*Sat*-formula $\varphi_G$.*

*Proof.* Assume that $\overrightarrow{S}$ induces a transitive orientation $\overrightarrow{E_H}$ of the subgraph $H$ that can be extended to a transitive orientation $\overrightarrow{E_G}$ of the entire graph $G$. Then $\overrightarrow{E_H}$ in turn induces a transitive orientation of $T$. In particular $\overrightarrow{S}$ induces a transitive orientation for all prime and all complete quotient graphs corresponding to nodes in $T$. By Lemma 4.9 for every complete node $\mu_S$ in $S$, $\overrightarrow{S}$ reduced to the quotient graph $[\mu_S]_H/\text{children}_S(\mu_S)$ fulfils the consecutiveness constraints. This means that $[\mu_S]_H/\text{children}_S(\mu_S)$ is oriented according to a total order induced by the corresponding $PQ$-tree. By Lemma 4.11 we also know that all prime and complete quotient graphs corresponding to nodes in $S$ are oriented according to a solution of the 2-Sat-formula $\varphi_G$.

For the other direction assume that $\overrightarrow{S}$ contains for every complete quotient graph in $S$ an orientation according to a total order induced by the corresponding $PQ$-tree that is compatible with a solution of the 2-Sat-formula $\varphi_G$. Again let $\overrightarrow{E_H}$ be the transitive orientation of subgraph $H$ induced by $\overrightarrow{S}$. By Lemma 4.12 we know that for every complete quotient graph $[\mu_T]_H/\text{children}_T(\mu_T)$ there exists a complete node $\mu_S$ in $S$ such that for every edge $e$ in $H$ with $\text{rep}_T(e) = \mu_T$, we have $\text{rep}_S(e) = \mu_S$. Hence every transitive orientation of $S$ induces a transitive orientation on the complete quotient graphs of $T$. Since by Lemma 4.9 the transitive orientation $\overrightarrow{S}$ reduced to the prime quotient graphs induces a partial orientations of the quotient graphs in $T$, $\overrightarrow{S}$ induces transitive orientations for all complete quotient graphs corresponding to nodes in $T$. By Lemma 4.11 $\overrightarrow{S}$ induces transitive orientations for all quotient graphs $[\mu_T]_H/\text{children}_T(\mu_T)$ for all prime-labelled nodes $\mu_T$. Thus in total $S$ induces a transitive orientation of $T$ which means that $\overrightarrow{E_H}$ can be extended to a transitive orientation of the entire graph $G$. ☐

Let us now consider $r$-sunflower graphs $G_1, G_2, \ldots, G_r$ and let $H$ be the induced subgraph shared by them. Furthermore let $\varphi_i$ be the 2-Sat-formula corresponding to $G_i$ for every $1 \leq i \leq r$ and let $G := G_1 \cup G_2 \cup \cdots G_r$. With the procedure described above for every complete node $\mu$ in $S$ we get $r$ $PQ$-trees, each representing the constraints a transitive orientation of $\mu$ must fulfil to induce a transitive orientation of $H$ that can be extended to a transitive orientation of $G_i$. Hence we label $\mu$ with the intersection of all these $r$ $PQ$-trees and denote the intersection tree by $B_\mu$. If two $Q$-nodes $q_1$ and $q_2$ appearing in $\varphi_i$'s are merged during the intersection process, the resulting new $Q$-node gets new variable $q$ and every occurrence of $q_1$ and $q_2$ in any $\varphi_i$ is replaced by $q$. Finally, we can define the 2-Sat-formula $\varphi_G := \bigwedge_{1 \leq i \leq r} \varphi_i$.

**Definition 4.14.** *Let $G_1 = (V_1, E_1), \ldots, G_r = (V_r, E_r)$ be $r$-sunflower comparability graphs and let $G = G_1 \cup G_2 \cup \cdots \cup G_r$. Then an orientation $W$ of $G$ is* pseudo-transitive*, if $W$ is transitive on every $G_i$ for $i \in \{1, \ldots, r\}$.*

Note that $r$ sunflower comparability graphs are simultaneous comparability graphs if and only if there exists a pseudo-transitive orientation of $G = G_1 \cup G_2 \cup \cdots \cup G_r$ [JL10].

**Theorem 4.15.** *Let $\overrightarrow{S}$ be a transitive orientation of $S$. Then $\overrightarrow{S}$ induces a transitive orientation of the subgraph $H$ that can be extended to a pseudo-transitive orientation of $G = G_1 \cup G_2 \cup \cdots \cup G_r$, if and only if every complete quotient graph in $S$ is oriented according to a total order induced by the corresponding PQ-tree and the orientation of all non-empty quotient graphs are compatible with a solution of the 2-*Sat*-formula $\varphi_G$.*

*Proof.* Assume that $\overrightarrow{S}$ induces a transitive orientation of the subgraph $H$ that can be extended to a pseudo-transitive orientation of $G = G_1 \cup G_2 \cup \cdots \cup G_r$. Then in particular $\overrightarrow{S}$ induces a transitive orientation of the subgraph $H$ that can be extended to a pseudo-transitive orientation of $G_i$ for every $i \in \{1, \ldots, r\}$ and hence by Theorem 4.13 every complete quotient graph in $S$ is oriented according to a total order induced by the corresponding $PQ$-tree $B_\mu^i$ and the orientation of all non-empty quotient graphs are compatible with a solution of the 2-Sat-formula $\varphi_i$. Since this holds for ever $i \in \{1, \ldots, r\}$, every complete quotient graph in $S$ is oriented according to a total order induced by the intersection of the $B_\mu^i$s, namely $B_\mu$, and the orientation of all non-empty quotient graphs are compatible with $\varphi_G$.

Now assume that every complete quotient graph in $S$ is oriented according to a total order induced by the corresponding $PQ$-tree and the orientation of all non-empty quotient graphs are compatible with a solution of the 2-Sat-formula $\varphi_G$. Then in particular for every $i \in \{1, \ldots, r\}$ every complete quotient graph in $S$ is oriented according to a total order induced by the corresponding $PQ$-tree $B_\mu^i$ and the orientation of all non-empty quotient graphs are compatible with a solution of the 2-Sat-formula $\varphi_i$. Hence by Theorem 4.13 $\overrightarrow{S}$ induces a transitive orientation of the subgraph $H$ that can be extended to a pseudo-transitive orientation of every $G_i$. Thus $\overrightarrow{S}$ induces a transitive orientation of the subgraph $H$ that can be extended to a pseudo-transitive orientation of $G = G_1 \cup G_2 \cup \cdots \cup G_r$. □

Theorem 4.15 directly implies the following corollary.

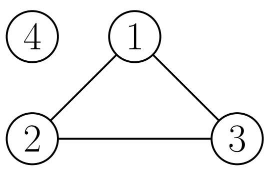
**Corollary 4.16.** *$G_1, G_2, \ldots, G_r$ are simultaneous comparability graphs if and only if for every complete node $\mu$ in $S$ the $PQ$-tree $B_\mu$ is not the null tree and the 2-Sat-formula $\varphi_G$ is satisfiable.*

Let $G_1, G_2, \ldots, G_r$ be simultaneous comparability graphs. Then we choose an arbitrary solution of $\varphi$ and orient every complete quotient graph of $S$ according to a total order induced by the corresponding $PQ$-tree that is compatible with the chosen solution of $\varphi$ (i.e. the children of every $Q$-node are ordered according to the solution of $\varphi$). For a prime quotient graph of node $\mu$ we choose the default orientation if the chosen solution of $\varphi_G$ contains $\overrightarrow{\mu}$ and the reverse orientation otherwise. Together, all these orientations of quotient graphs in $S$ induce a transitive orientation on $H$ and by applying the algorithm from Chapter 4 to solve RepExt(*Comp*) to every input graph, we get a pseudo-transitive orientation of $G = G_1 \cup G_2 \cup \cdots \cup G_r$.

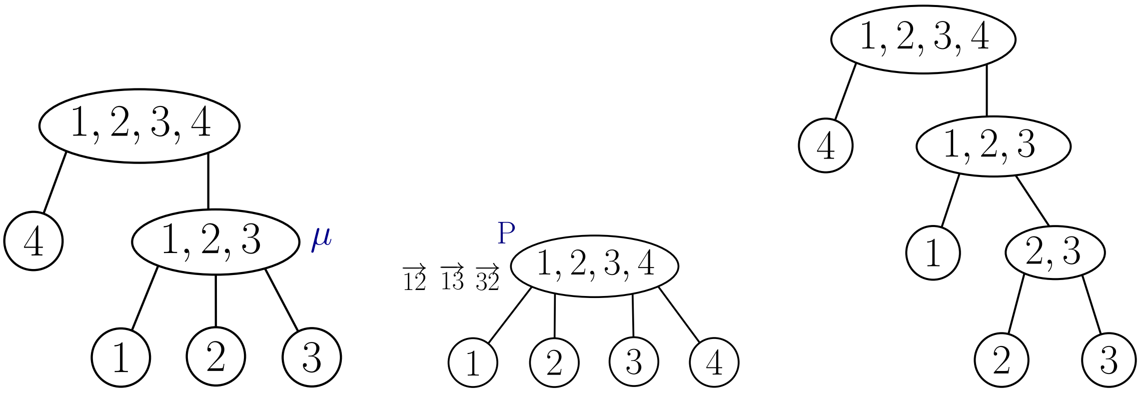
**Example 4.17.** *Consider the following graphs $G_1$ and $G_2$.*


(a) $G_1$ (b) $G_2$

*The two input graphs share the following induced subgraph $H$.*

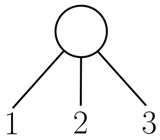


*Now we construct the (reduced) modular decomposition trees $S$, $T_1$ and $T_2$.*

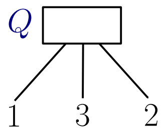



(a) $S$ (b) $T_1$ (c) $T_2$

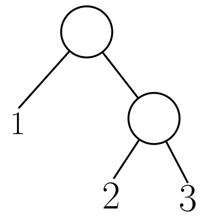
*We first consider $T_1$ and compute the corresponding PQ-tree $B^1_\mu$ for the only complete node in $S$ and the 2-*SAT*-formula $\varphi_1$. Initially $B^1_\mu$ is the following tree.*




*The lowest common ancestor of the vertices in $\mu$ in tree $T_1$ is the prime root $P$ with $M_P = \{1, 2, 3\}$ and the default orientation induces $1 < 3 < 2$. Hence we get the following tree for $B_\mu$.*




*Additionally we add the constraint $(\overrightarrow{P} \Leftrightarrow Q)$ to $\varphi_1$, i.e. $\varphi_1 = (\overrightarrow{P} \vee \neg Q) \wedge (\neg \overrightarrow{P} \vee Q)$. The decomposition tree $T_2$ gives us the following $PQ-tree$ for $\mu$.*




*Intersecting the two PQ-trees gives us the PQ-tree we received from $T_1$. Since $\varphi_2 = \emptyset$, $\varphi_G = \varphi_1$. A satisfying solution is for example $\{\overrightarrow{P}, Q\}$. Since $B_\mu$ is not the null tree and $\varphi_G$ is satisfiable we already know that $G_1$ and $G_2$ are simultaneous comparability graphs. The default orientation of $P$ directly gives us a transitive orientation of $H$ that is compatible with $B_\mu$ and a solution of $\varphi_G$. Extending this gives us the following simultaneous transitive orientations for $G_1$ and $G_2$.*


(a) $G_1$ (b) $G_2$

**Theorem 4.18.** *SimRep(Comp) can be solved in $\mathcal{O}(n+m)$ time for $r$-sunflower comparability graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_r = (V_r, E_r)$ where $n = \sum_{i=1}^{r} |V_i|$ and $m = \sum_{i=1}^{r} |E_i|$.*

*Proof.* Let $n_i := |V_i|$ and $m_i := |E_i|$ for all $1 \leq i \leq r$. We have seen that to solve SimRep(Comp) first we need to compute the modular decomposition trees of the input graphs $G_1, G_2, \ldots, G_r$ and the shared graph $H$. This can be done in $\mathcal{O}(n_i+m_i)$ time for every input graph $G_i$ [MS99] for $1 \leq i \leq r$. Hence in total we need $\sum_{i=1}^{r} \mathcal{O}(n_i + m_i) = \mathcal{O}(n+m)$ time to construct all $r$ modular decomposition trees. Additionally we have to construct the modular decomposition tree $S$ of the shared graph $H$ which also takes $\mathcal{O}(n+m)$ time since $H$ has at most $n$ vertices and $m$ edges. To reduce a modular decomposition tree $B_i$ to the vertices of $H$ we have to process every node $\mu$ in $B_i$ once to remove all vertices not in $H$ from $\mu$. We visit the nodes from bottom to top such that all children of $\mu$ are processed before $\mu$ to be able to remove empty and degree-one nodes directly from the tree. Since every $B_i$ has $n_i$ leaves and thus in total at most $2n_i - 1$ nodes, the reduction of all $r$ modular decomposition trees can be done in $\mathcal{O}(\sum_{i=1}^{r} n_i) = \mathcal{O}(n)$ time.

To compute the $PQ$-tree $B_\mu^i$ for a complete node $\mu$ in $S$ we have to consider at most as many nodes in $T_i$ as the number of edges in $H$ represented by edges in $[\mu]_H/\text{children}_S(\mu)$. Hence the computation of all $PQ$-trees for every complete node in $S$ with regard to $T_i$ can be done in $\mathcal{O}(m_i + n_i)$ time and thus we need $\mathcal{O}(n+m)$ time to construct all $r$ $PQ$-trees for all complete nodes in $S$. Every $PQ$-tree $B_\mu^i$ for a complete node $\mu$ in $S$ is a $PQ$-tree on the set children($\mu$) and hence their intersection can be computed in $\mathcal{O}(|\text{children}(\mu)|)$ time [Boo75] which means that all needed intersections can be computed in total in $\mathcal{O}(n)$ time.

The construction of the 2-Sat-formula $\varphi_i$ and their combination $\varphi_G$ takes constant time. By [APT79] we can solve $\varphi_G$ in $\mathcal{O}(n' + m')$ time where $n'$ is the number of variables in $\varphi_G$ and $m'$ is the number of clauses in the formula. Every $\varphi_i$ contains one variable for every prime node in $T_i$ and $S$ which are less than $4n_i$ and one variable for every $Q$-node in a $PQ$-tree $B_\mu^i$ for a complete node $\mu$ in $S$. Note that we only get $Q$-nodes in $B_\mu^i$ if there exists a prime node $\mu'$ in $T_i$ such that there exists an edge in $[\mu']_H/\text{children}_T(\mu')$ that represents an edge of $H$ that is also represented by an edge in $[\mu]_H/\text{children}_S(\mu)$. Hence in total all $B^i$s can contain at most $m_i$ $Q$-nodes. By construction $\varphi_G$ contains not more variables than all $\varphi_i$s together. Thus $\varphi_G$ contains at most $4n + m$ variables. Furthermore we know that every $\varphi_i$ contains exactly two clauses for every $Q$-node and exactly two clauses for every prime node in $S$, hence $\mathcal{O}(n_i + m_i)$ in total. Thus by construction $\varphi_G$ contains $\mathcal{O}(n+m)$ clauses and can be solved in $\mathcal{O}(n+m)$ time.

□

# 5. Permutation graphs

In this chapter we give efficient algorithms for solving the *partial representation problem* and the *simultaneous representation problem* for permutation graphs.

## 5.1 Extending Partial Representations

Applied to the class of permutation graphs, the problem RepExt(Perm) asks, whether given a permutation graph $G = (V, E)$ and a permutation diagram $D'$, representing an induced subgraph $H$ of $G$, it is possible to extend $D'$ to a representation of the entire graph $G$, such that $D$ restricted to the vertices in $H$ is isomorphic to $D'$.

RepExt(Perm) was already shown to be solvable in cubic time [KKKW12]. The algorithm presented by Klavík et al. applies their $\mathcal{O}((n + m)\Delta)$ algorithm for solving the partial representation problem for comparability graphs to both, the input graph and its complement, and if existent constructs the desired extended permutation diagram out of the two extended transitive orientations. Here we will give an algorithm that solves RepExt(Perm) in linear time using modular decomposition. To do so, we need the following definitions and observations:

Let $\sigma$ be a permutation of a set $S$ of size $n$. A subset $H \subseteq S$ is an *interval* of $\sigma$, if the elements of $H$ occur consecutively in $\sigma$ [HP10].

**Definition 5.1.** *A set $S$ of elements is a* common interval *of a set of permutations $\Sigma$, if in each permutation $\sigma \in \Sigma$, the elements of $S$ form an interval of $\sigma$.*

*$S$ is called a* strong *common interval, if for every distinct common interval $S'$ of $\Sigma$, we either have $S \subseteq S'$, $S' \subseteq S$ or $S \cap S' = \emptyset$.*

**Lemma 5.2** ([Mon03])**.** *Let $G = (V, E)$ be a permutation graph and $D$ be a permutation diagram representing $G$. We denote the upper horizontal line of $D$ by $L_1$ and the lower line by $L_2$. Furthermore, let $L_i(V)$ denote the order of vertices induced by the labelling of $L_i$ for $i \in \{1, 2\}$. A set of vertices $M \subseteq V$ is a strong module if and only if $M$ is a strong common interval of $L_1(V)$ and $L_2(V)$.*

**Corollary 5.3.** *Let $G$ be an undirected permutation graph and let $T$ be its modular decomposition tree. Then for every $\mu \in T$, the quotient graph $[\mu]_G/\mathrm{children}_T(\mu)$ is also a permutation graph.*

*Proof.* Let $R$ be a representation of $G$ and let $\mu \in T$. Since all children $X$ of $\mu$ in $T$ are strong modules by definition of the modular decomposition tree, the vertices of $G$ they contain, appear as strong common intervals along both, the upper and the lower line of $R$. Hence, to construct a permutation diagram for $[\mu]_G/\text{children}_T(\mu)$, we first delete all labels and chords in $R$ that do not belong to vertices in $\mu$ and afterwards, along both parallel lines, replace for every node $X \in \text{children}_T(\mu)$ the labels corresponding to vertices contained in $X$ (which form a strong common interval) by a single new label for $X$. As a last step, we connect all labels along $L_1$ to their corresponding label along $L_2$. □

In the following, let $G = (V, E)$ be a permutation graph with $n$ vertices and $m$ edges and let $D'$ be a corresponding partial representation of a subgraph $H = (V', E)$ of $G$. Furthermore, let $T$ be the modular decomposition tree for $G$ and let $\mu \in T$ be a module. Then a permutation diagram $D_\mu$ of the quotient graph $[\mu]_G/\text{children}_T(\mu)$ *respects* the order of the vertices induced by $D'$, if for every pair of distinct vertices $i, j \in \mu \cap V'$, they either both belong to the same child of $\mu$ or the labels corresponding to the children that contain $i$ and $j$, appear in the same order along both horizontal lines of $D_\mu$ as $i$ and $j$ appear along both horizontal lines of $D'$. We shall see with the next Lemma that every permutation diagram for a permutation graph $G$ induces a permutation diagram for every quotient graph $[\mu]_G/\text{children}_T(\mu)$.

**Lemma 5.4.** *Let $G = (V, E)$ be a permutation graph with $n$ vertices and $m$ edges and let $D'$ be a corresponding partial representation of a subgraph $H$ of $G$. Furthermore, let $T$ be the modular decomposition tree for $G$. Then $D'$ is extendible to a permutation diagram $D$ representing the entire graph $G$ if and only if for every inner node $\mu$ in $T$, there exists a permutation diagram $D_\mu$ for the quotient graph $[\mu]_G/\text{children}_T(\mu)$ that respects the order of the vertices induced by $D'$.*

*Proof.* Assume that $D'$ is extendible to a permutation diagram $D$ representing the entire graph $G$, such that $D$ restricted to $H$ is isomorphic to $D'$. Then for every inner node $\mu$ in the modular decomposition tree $T$, we get the permutation diagram $D_\mu$ for the quotient graph $[\mu]_G/\text{children}_T(\mu)$ that respects the order of the vertices induced by $D'$, by ordering the endpoints of the segments corresponding to vertices in $\text{children}_T(\mu)$ along both horizontal lines according to their order in $D$. This is well- defined, since all modules $\mu \in T$ are strong by definition and by Lemma 5.2, we know that for every strong module, all labels corresponding to vertices contained in $\mu$ appear as consecutive sequence along both the upper and the lower horizontal line of every representation $D$ for $G$.

Now, assume that for every inner node $\mu$ in $T$, there exists a permutation diagram $D_\mu$ for the quotient graph $[\mu]_G/\text{children}_T(\mu)$ that respects the order of the vertices induced by $D'$. Then we can use all the $D_\mu$'s to construct a permutation diagram $D$ for $G$ that extends $D'$. We start with the representation of $G/\text{children}_T(G)$ and replace for every $X \in \text{children}_T(G)$ the corresponding endpoints along the two horizontal lines with the order of the vertices of $[X]_G/\text{children}_T(X)$ induced by $D_X$. Then we connect all pairs of labels corresponding to the same node and proceed with the same procedure for every $X' \in \text{children}_T(X)$, until every segment of the representation corresponds to a single vertex of $G$. □

The construction described in the proof above can be done in linear time: Assume that for every permutation diagram $D$ we store two double linked lists that contain the labels in the order they appear along $L_1$ and $L_2$ respectively. Then we may start with the list for $L_1$ of a representation for $G/\text{children}_T(G)$ and traverse it from left to right. If we meet a label corresponding to a node $\mu$ in $T$, we replace the current entry with the linked list corresponding to the upper horizontal line of a representation $D_\mu$ of $[\mu]_G/\text{children}_T(\mu)$ and afterwards proceed our traversing at the leftmost entry of the inserted sublist. We continue in the same way until we reach the right most entry of the list and no further replacements are necessary. Then we proceed similarly for the list storing the labels of $L_2$. It remains to show that the total number of list entries we have to look at is linear in n, denoting the number of nodes in $G$. Therefore we first have to prove the following assumption:

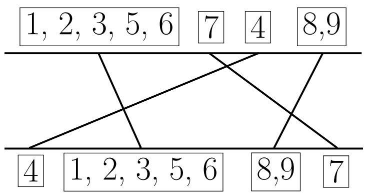
Assume we reach a list entry $\mu$ that contains a set of $k$ nodes and has a right neighbour $\nu$. If we replace it as described above, we visit at most $2k-1$ entries (including $\mu$) until we reach the former right neighbour $\nu$.

*Proof.* For $k = 1$ the claim holds, since in this case we do not replace anything and reach $\nu$ directly in the next step. If $k = 2$ we replace $\mu$ by two new list entries both containing singleton sets that we do not have to replace themselves. This means we have to look at $\mu$ and the two new entries before reaching $\nu$, which makes $3 = 2 \cdot 2 - 1$ visited entries in total. Now for $k > 2$, we divide $\mu$ into $2 \leq j \leq k$ disjoint subsets $l_1, \ldots, l_j$ such that $\sum_{i=1}^{j} |l_i| = k$. Then we know that to completely dissolve $l_i$ into singleton sets we need to visit at most $2|l_i| - 1$ list entries. Hence, in total, to dissolve $\mu$ we look at at most $\sum_{i=1}^{j} 2|l_i| - 1 = (\sum_{i=1}^{j} 2|l_i|) - j = 2(\sum_{i=1}^{j} |l_i|) - j = 2k - j < 2k - 1$ list entries. □

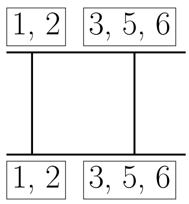
Now, let $n$ be the number of nodes in $G$ and let $n'$ be the number of nodes in $G/\text{children}_T(G)$. Then the original lists corresponding to the horizontal lines of $D_G$ representing $G/\text{children}_T(G)$ each have $n' < n$ entries $k_1, \ldots, k_{n'}$ with $\sum_{i=1}^{n'} |k_i| = n$. The above assumption implies that for every horizontal line of $D_G$ we need to visit at most $\sum_{i=1}^{n'} 2|k_i| - 1 = 2(\sum_{i=1}^{n'} |k_i|) - n' = 2n - n' \leq 2n$ list entries to get a list that only contains singleton sets.

The following example illustrates the construction of $D$ given $D'$.

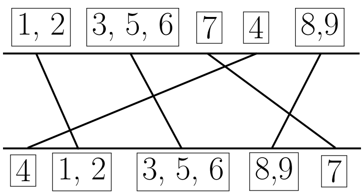
**Example 5.5.** *In this example we will see, how we may construct a representation for the graph $G$ in Example 2.14 from the representations of its quotient graphs. To do so, we start with a representation $D_G$ for $G/\text{children}_T(G)$:*



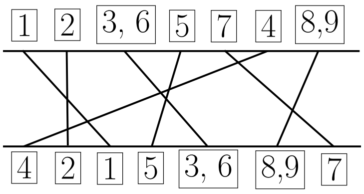

*For the next step, we choose $\mu = \{1, 2, 3, 5, 6\}$, which is a child of $V$ in the modular decomposition tree $T$. The following permutation diagram $D_\mu$ represents $[\mu]_G/\text{children}_T(\mu)$:*



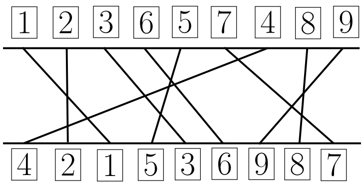

*Now, we replace in $D_G$ the label corresponding to $X$ along the upper line by the labels appearing in $D_\mu$ along the upper line. The labels along the lower line of $D_\mu$ replace the label corresponding to $\mu$ along the lower line of $D_G$ and afterwards each pair of matching labels is connected via a new segment:*




*In the next step we proceed similarly for the node $\{1, 2\}$ in $T$ and get the following modified permutation diagram:*




*After replacing the labels $\{3, 6\}$ and $\{8, 9\}$ with the new labels induced by the representations of their corresponding quotient graphs and connecting the matching labels, we get the following linear representation of the entire graph $G$:*




Now we are able to prove that the presented algorithm has a linear runtime.

**Theorem 5.6.** *The problem REPEXT(PG) can be solved in $\mathcal{O}(n + m)$ time for permutation graphs with $n$ vertices and $m$ edges.*

*Proof.* Let $G = (V, E)$ be a permutation graph with $n$ vertices and $m$ edges and let $D'$ be a corresponding partial representation of a subgraph $H = (V', E)$ of $G$. By McConnell and Spinrad [MS99], it is possible to compute the modular decomposition tree $T$ of $G$ in $\mathcal{O}(n + m)$ time.

Then by Lemma 5.4 $D'$ is extendible to a permutation diagram $D$ representing the entire graph $G$, such that $D$ restricted to $X$ is isomorphic to $D'$, if and only if for every inner node $\mu$ in $T$, there exists a permutation diagram $D_\mu$ for the quotient graph $[\mu]_G/\text{children}_T(\mu)$ that respects the order of the vertices induced by $D'$.

Let $\mu \in T$ be a node of the modular decomposition tree. It is already known that the quotient graph $[\mu]_G/\text{children}_T(\mu)$ is either, complete, empty or prime [MS99] and by definition, $\mu$ and all its children represent a subset of vertices of $G$ that are a strong module of the graph. In the first two cases, we can choose an arbitrary total order of the children of $\mu$ that respects the order induced by the upper line of $D'$, for the upper line $L_1$ and reverse this order along the lower line, to represent a complete graph, or choose the same order for $L_2$, to represent an empty graph. Such an order always exists, since for every child $\nu$ of $\mu$, we know that $\nu \cap \mu$ is a strong module of $H$, and hence, by Lemma 5.2, $\nu \cap \mu$ is a strong common interval on both, the lower and the upper line of $D'$.

Since by Corollary 5.3 every quotient graph is a permutation graph, if existent, we may compute the representation with the desired properties of a prime quotient graph

$[\mu]_G/\text{children}_T(\mu)$ in time that is linear in the sum of the sizes of nodes and edges of $[\mu]_G/\text{children}_T(\mu)$[MS99]. For every prime graph, there exist exactly two transitive orientations, where each one is the reverse of the other [Gol04]. Thus, if the computed orientation does not respect the order of the vertices induced by the upper horizontal line of $D'$, we reverse it and check again. If the new transitive orientation still does not have the desired property, we know that $D'$ is not extendible to a representation $D$ for the entire graph $G$.

To obtain a representation $D$ that represents the entire graph $G$ and extends $D'$, if existent, we proceed as in the proof of Lemma 5.4, i.e. starting with the representation of $G/\text{children}_T(G)$ we stepwise replace every non-singleton module $\mu$ by its corresponding representation $D_\mu$. We have seen that this can be done in linear time.

□

## 5.2 The Simultaneous Representation Problem

### 5.2.1 Simultaneous Representations for Sunflower Permutation Graphs

The simultaneous representation problem for $r$-sunflower permutation graph $G_1, G_2, \dots, G_r$ sharing an induced subgraph $H$ deals with the question whether there exist permutation diagrams $D_1, D_2, \dots, D_r$ such that $D_i$ represents $G_i$ for every $1 \leq i \leq r$ and all $D_i$'s are isomorphic on the shared vertices of $H$. In this section we show that SimRep(Perm) can be solved in quadratic time using modular decomposition and the property, that every permutation graph and its complement are comparability graphs. To do so, we first need to prove the following lemma.

**Lemma 5.7.** *Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \dots, G_r = (V_r, E_r)$ be permutation graphs. Then $G_1, \dots, G_r$ are simultaneous permutation graphs if and only if they are simultaneous comparability graphs and simultaneous co-comparability graphs.*

*Proof.* Assume that $G_1, \dots, G_r$ are simultaneous permutation graphs. Then there exist permutation diagrams $D_1, \dots, D_r$ such that for every $i \in \{1, \dots r\}$ $D_i$ represents $G_i$ and all $D_i$'s are the same on the shared subgraph $H$, i.e. the vertices of $H$ appear in the same order along the upper and the lower line of every $D_i$. Let $<_i$ be the total order of the vertices in $E_i$ induced by their order along the upper line $L_1$ of $D_i$ such that for two vertices $u$ and $v$ we have $u <_i v$ if and only if $u$ appears to the left of $v$ along $L_1$. By Pneuli et al. [PLE71] we get transitive orientations $\overrightarrow{E_i}$ of $G_i$ and $\overrightarrow{E_i}^C$ of the complement $\overline{G_i}$ if we orient every edge towards the greater vertex according to $<_i$. Since for all $G_i$'s the vertices in $H$ appear in the same order along the upper horizontal line of $D_i$, the edges induced by them are oriented in the same way in every orientation $\overrightarrow{E_i}$. Since the same holds also for $\overrightarrow{E_i}^C$, $G_1, \dots, G_r$ are simultaneous comparability graphs and simultaneous co-comparability graphs.

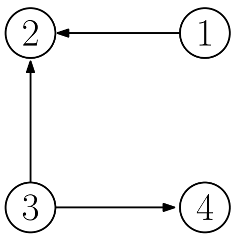
For the other direction we assume that $G_1, \dots, G_r$ are simultaneous comparability and co-comparability graphs. Then there exists a pseudo-transitive orientation $\overrightarrow{W}$ of $G = G_1 \cup G_2 \cup \dots \cup G_r$ such that $\overrightarrow{W}$ is transitive on every $G_i$. There also exists a pseudo-transitive orientation $\overrightarrow{W'}$ of $G' = \overline{G_1} \cup \overline{G_2} \cup \dots \cup \overline{G_r}$ such that $\overrightarrow{W'}$ is transitive on every $\overline{G_i}$. By orienting every $G_i$ with the directions induced by $\overrightarrow{W}$ and $\overline{G_i}$ with the directions induced by $\overrightarrow{W'}$, and considering the resulting directed complete graph, we receive an induced total order $<_i$ of the vertices in $V_i$. Since the subgraph $H$ and its complement $\overline{I}$ are oriented simultaneously in every $G_i$ and $\overline{G_i}$ respectively, the total order $<_i$ reduced to the vertices in $H$ is the same for every $1 \leq i \leq r$. Now we construct a simultaneous permutation diagram $D_i$ for every $G_i$ such that they are the same on the shared subgraph $H$ as follows. First we label the upper line $L_1$ of $D_i$ with the vertices in $V_i$ in ascending order according to $<_i$. Then we orient the complete graph that contains all vertices in $V_i$ according to $\overrightarrow{W'}$ and the reverse of $\overrightarrow{W}$ which we denote by $\overleftarrow{W}$. The total order induces by this orientation we denote by $<_i'$. Again we know that $<_i'$ reduced to the vertices in $H$ is the same for every $1 \leq i \leq r$. Hence if we label the bottom line $L_2$ of $D_i$ with the vertices in $V_i$ in ascending order according to $<_i'$ and connect the vertices on $L_1$ to their counterpart on $L_2$ with a line segment, we receive a permutation diagram representing $G_i$. By construction all $D_i$'s are the same on the shared subgraph $H$, hence $G_1, G_2, \ldots, G_r$ are simultaneous permutation graphs. □

Now we are able to show the desired quadratic runtime to solve SimRep(Perm).

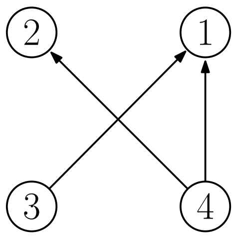
**Corollary 5.8.** *The problem SimRep(Perm) can be solved in $\mathcal{O}(n^2)$ time for $r$-sunflower permutation graphs $G_1, G_2, \ldots, G_r$, where $n$ is the number of vertices in $G := G_1 \cup G_2 \cup \cdots \cup G_r$.*

*Proof.* By Lemma 5.7 to solve SimRep(Perm) for $G_1, G_2, \ldots, G_r$ it suffices to solve SimRep(Comp) for $G_1, G_2, \ldots, G_r$ and also for the complement graphs $\overline{G_1}, \overline{G_2}, \ldots, \overline{G_r}$. In Section 4.2 we have seen that SimRep(Comp) can be solved in $\mathcal{O}(n+m)$ time for $r$-sunflower comparability graphs $G_1, G_2, \ldots, G_r$ with $G := G_1 \cup G_2 \cup \cdots \cup G_r$, where $n$ is the number of vertices in $G$ and $m$ denotes the number of edges in $G$. The union of the complements $G' := \overline{G_1} \cup \overline{G_2} \cup \cdots \cup \overline{G_r}$ has also $n$ nodes. The number of edges in $G'$ we denote by $m'$. Hence SimRep(Comp) for $\overline{G_1}, \overline{G_2}, \ldots, \overline{G_r}$ can be solved in $\mathcal{O}(n+m')$ time. Since $m + m' = n^2$ to solve SimRep(Perm) in total we need $\mathcal{O}(n^2)$ time. The corresponding simultaneous representations we get as described in the proof of Lemma 5.7. □

**Example 5.9.** *This Example shall illustrate how we get the simultaneous permutation diagrams from the simultaneous transitive orientations of the input graphs and their complements of a SimRep(Perm)-instance. Consider an instance of the simultaneous representation problem for $r$-sunflower permutation graphs $G_1, \ldots, G_r$ such that the input graphs are simultaneous comparability and simultaneous co-comparability graphs. Assume that we get the following orientations for $G_1$ after applying the algorithm presented in Section 4.2 to solve SimRep(Comp).*

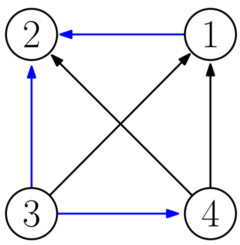



(a) $G_1$

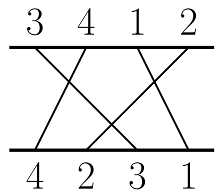



(b) $\overline{G_1}$

*Then these two orientations induce a the following transitive on the complete graph $G_1 \cup \overline{G_1}$ that in turn induces the total order $3 < 4 < 1 < 2$.*



*After reversing all orientations of the edges in $G_1$ the new orientation of $G_1 \cup \overline{G_1}$ induces the total order $4 < 2 < 3 < 1$. Hence we receive the following permutation diagram representing $G_1$.*




### 5.2.2 Simultaneous Representations for general Permutation Graphs

Now we come back to the general case where the $r$ graphs $G_1, G_2, \ldots, G_r$ pairwise share an arbitrary set of vertices and edges induced by them. Then for permutation graphs $G_1, \ldots, G_r$ the simultaneous representation problem turns out to be NP-complete.

Let $G_1 = (V_1, E_1), \ldots G_k = (V_k, E_k)$ be arbitrary permutation graphs. The problem SimRep(Perm) is to decide whether there exist linear permutation diagrams $R_1, \ldots, R_k$, such that for every $1 \leq i \leq k$, $R_i$ represents $G_i$ and for every $j, l \in \{1, \ldots k\}$ the order of the labels corresponding to vertices in $V_j \cap V_l$ along the upper and lower line respectively is the same in both $R_j$ and $R_l$.

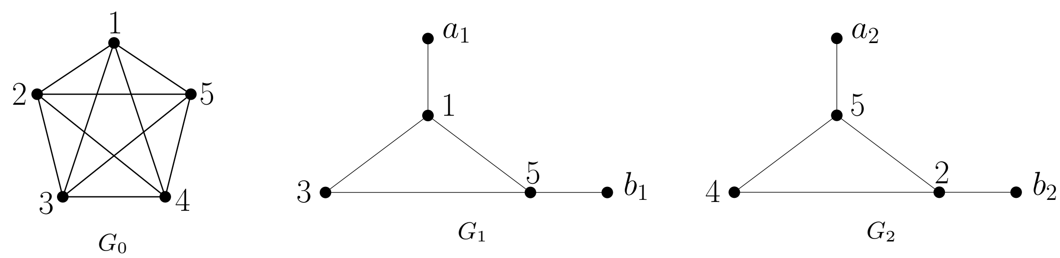
**Theorem 5.10.** *SimRep(Perm) for k permutation graphs where k is not fixed is NP-complete.*

*Proof.* Let $n = \max\{|V_1|, |V_2|, \ldots, |V_k|\}$. Given representation $R_i$ of $G_i$ for every $1 \leq i \leq k$, to check whether $R_1, \ldots, R_k$ are a solution of SimRep(Perm), we have to check for every pair of graphs, whether their common vertices appear in the same order along the upper and lower line respectively in the corresponding representations. Since such a check can be done in linear time, in total we can decide in $\mathcal{O}(k^2 n)$ time whether $R_1, \ldots, R_k$ are valid simultaneous representations of $G_1, \ldots G_k$. Hence, the problem SimRep(Perm) is in NP.

To show the NP-completeness, we reduce the problem TotalOrdering, which is known to be NP-complete [Opa79], to SimRep(Perm). TotalOrdering is defined as follows: Given a finite set $S$ and a finite set $T$ of triples of elements in $S$, decide whether there exists a total ordering $<$ of $S$ such that for all triples $(x, y, z) \in T$ either $x < y < z$ or $x > y > z$. Let $H_T$ be an instance of TotalOrdering with $s = |S|$ and $t = |T|$. Furthermore we number the triples in $T$ with $1, \ldots, t$ and denote the $i-th$ triple by $(x_i, y_i, z_i)$. We construct an instance $H_S$ of SimRep(Perm), consisting of undirected graphs $G_0, G_1, \ldots, G_t$ as follows:

- $G_0 := (S, E_0)$ is the complete graph with one vertex for each element in $S$.
- $G_i := (V_i, E_i)$ for $1 \leq i \leq t$ are the graphs with $V_i = \{x_i, y_i, z_i, a_i, b_i\}$, $E_i = \{x_i a_i, x_i y_i, x_i z_i, y_i z_i, z_i b_i\}$ with $a_i, b_i \notin S$ (see Figure 5.2)

**Example 5.11.** *Consider the TotalOrdering instance $T = \{(1,3,5),(5,4,2)\}$ with $S := \{1,2,3,4,5\}$. Then the corresponding SimRep(Perm) instance contains the following three permutation graphs:*



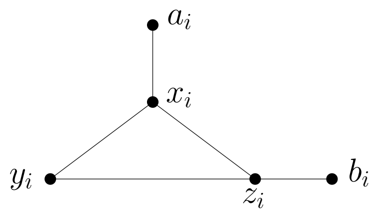



Figure 5.2: The graph $G_i$ corresponding to the $i-th$ triple $(x_i, y_i, z_i)$ of a TotalOrdering instance

Since $G_0$ is a complete graph, it is a permutation graph. For every $1 \leq i \leq t$, we have exactly two transitive orientations of $G_i$, namely $\{\overrightarrow{x_i a_i}, \overrightarrow{x_i y_i}, \overrightarrow{x_i z_i}, \overrightarrow{y_i z_i}, \overrightarrow{b_i z_i}\}$ and its reversal. Since the complement graph $\overline{G_i}$ is isomorphic to $G_i$ (only the labels of the vertices have switched), the complement of every $G_i$ is a comparability graph as well and hence $G_1, \ldots, G_t$ are permutation graphs. We now have to show that $H_T$ has a solution if and only if $G_0, G_1, \ldots, G_t$ are simultaneous permutation graphs.

First, assume that $G_0, \ldots, G_t$ are simultaneous permutation graphs. Then we know that these graphs are simultaneous comparability and simultaneous co-comparability graphs. Hence there exist orientations $T_0, \ldots, T_t$ and $R_0, \ldots, R_t$ such that for every $0 \leq i \leq t$, $T_i$ is an orientation of $G_i$ and $R_i$ is an orientation of $\overline{G_i}$ with the following property: For every $j, k \in \{0, \ldots t\}$ every edge in $E_j \cap E_k$ is oriented in the same way in both $T_j$ and $T_k$ and every edge in $\overline{E_j} \cap \overline{E_k}$ is oriented in the same way in both $R_j$ and $R_k$. Then the orientation of the complete graph $G_0$ implies a total order on the elements of $T$, where $u < v$ if and only if the edge $uv$ is oriented from $u$ to $v$. By construction, there are only two valid transitive orientations for $G_i$. The one given above implies $x_i < y_i < z_i$ and for the reverse orientation we get $z_i < y_i < x_i$. Hence the received total order satisfies that for every triple $(x, y, z) \in T$ we have either $x < y < z$ or $x > y > z$.

Now assume that $H_T$ has a solution. Then there exists a total order $<$ such that for all triples $(x, y, z) \in T$ either $x < y < z$ or $x > y > z$ holds. We get a transitive orientation $R_i$ of $G_i$ for $1 \leq i \leq t$ if we orient all edges between the vertices $x_i, y_i$ and $z_i$ towards the greater element according to the order "$<$". If $x_i$ is the smallest element of the triple, then we choose $\overrightarrow{x_i a_i}$ and $\overrightarrow{b_i z_i}$, else $z_i$ is the smallest element and we choose $\overrightarrow{a_i x_i}$ and $\overrightarrow{z_i b_i}$. Finally, we orient the edges in $G_0$ also towards the pairwise greater element according to "$<$". This gives us orientations $T_0, \ldots, T_t$ of $G_0, \ldots, G_t$, with the property that for every $j, k \in \{0, \ldots t\}$ every edge in $E_j \cap E_k$ is oriented in the same way in both $T_j$ and $T_k$. Hence $G_0, \ldots, G_t$ are simultaneous comparability graphs. Furthermore, $\overline{E_0} = \emptyset$ and for every $1 \leq i \leq t$ for every edge $uv \in \overline{E_i}$ at least one of the endpoints $u$ and $v$ is in $\{a_i, b_i\}$. Hence, $\overline{G_0}, \ldots, \overline{G_t}$ pairwise do not share any edges and thus they are also simultaneous comparability graphs. Now Lemma 5.7 implies that $G_0, \ldots, G_t$ are simultaneous permutation graphs.

Hence the instance $H_T$ of TotalOrdering has a solution if and only if the instance $H_S$ of SimRep(Perm) is simultaneous and thus SimRep(Perm) is NP-complete. $\square$

# 6. Circular permutation graphs

In this chapter we give efficient algorithms for solving the *partial representation problem* and the *simultaneous representation problem* for circular permutation graphs.

## 6.1 Extending Partial Representations

The partial representation problem for circular permutation graphs deals with the question whether it is possible to extend a given circular permutation diagram $D'$, of a subgraph $H$ of a circular permutation diagram $G = (V, E)$ to a representation $D$ of the entire graph $G$. In this section we show that RepExt(CPerm) can be solved via a reduction to RepExt(Perm).

**Lemma 6.1.** *Let $G$ be a permutation graph with an isolated vertex $v$ and let $C_p$ be a circular permutation diagram representing an induced subgraph $H$ of $G$ containing vertex $v$. Let $D_p$ be the linear permutation diagram we receive by opening $C_p$ along the chord $\bar{v}$ corresponding to vertex $v$ which does not intersect any other chord in $C_p$. Then $C_p$ can be extended to a circular permutation diagram $C$ representing the entire graph $G$ such that $C$ and $C_p$ are isomorphic on the vertices of $H$ if and only if $D_p$ can be extended to a permutation diagram $D$ representing the entire graph $G$ such that $D$ and $D_p$ are isomorphic on the vertices of $H$.*

*Proof.* Assume that $C_p$ can be extended to a circular permutation diagram $C$ representing the entire graph $G$ such that $C$ and $C_p$ are isomorphic on the vertices of $H$. Then chord $\bar{v}$ does not intersect any other chord in $C$ and thus we can open $C$ along $\bar{v}$ and receive a linear permutation diagram representing $G$. Since $C$ and $C_p$ are isomorphic on the vertices of $H$, the same holds for $D$ and $D_p$, hence $D$ is an extension of $D_p$ with the desired properties.

Now assume that $D_p$ can be extended to a permutation diagram $D$ representing the entire graph $G$ such that $D$ and $D_p$ are isomorphic on the vertices of $H$. Analogously to the other direction we receive an extension of $C_p$ with the desired properties by transforming $D$ into a circular permutation diagram $C$. □

**Lemma 6.2.** *Let $G$ be a circular permutation graph and let $C_p$ be a circular permutation diagram representing an induced subgraph $H$ of $G$. Let $v$ be a vertex in $H$ and let $G'$ denote the graph we receive from $G$ by switching all neighbours of $v$. Furthermore let $C'_p$ be the* *circular permutation diagram representing the subgraph of $G'$ induced by the vertices in $H$ which we receive by switching all chords corresponding to neighbours of $v$ in $C_p$. Then $C_p$ can be extended to a circular permutation diagram $C$ representing the entire graph $G$ such that $C$ and $C_p$ are isomorphic on the vertices of $H$ if and only if $C'_p$ can be extended to a circular permutation diagram $C'$ representing the entire graph $G'$ such that $C'$ and $C'_p$ are isomorphic on the vertices of $H$.*

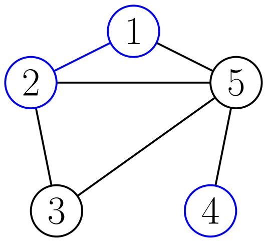
*Proof.* Assume that $C_p$ can be extended to a circular permutation diagram $C$ representing the entire graph $G$ such that $C$ and $C_p$ are isomorphic on the vertices of $H$. Let $C'$ be the circular permutation diagram we obtain by switching all chords corresponding to neighbours of $v$ in $C$. Then $C'$ represents $G'$. Note that the switch-operation does not change the order of the vertices along the inner and the outer circle of a circular permutation diagram and in both $C'$ and $C'_p$ the chords corresponding to vertices not adjacent to $v$ are bent in the same direction as in $C_p$, while the chords corresponding to vertices not adjacent to $v$ are bent in the opposite direction. Hence $C'$ and $C'_p$ are isomorphic on the vertices in $H$. Since the switch-operation is self-inverse, the other direction follows analogously. □

**Corollary 6.3.** *The problem REPEXT(CPERM) can be solved in $\mathcal{O}(n+m)$ time for circular permutation graphs with $n$ vertices and $m$ edges.*

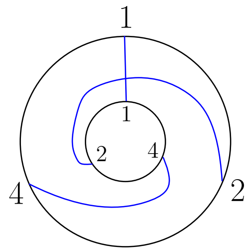
*Proof.* Given a circular permutation graph $G$ and a corresponding partial representation $C_p$ containing vertex $v$, we obtain a partial representation $C'_p$ of $G'$ by switching chord $\overline{x}$ for all $x$ that are adjacent to $v$ in $G$, one at a time in an arbitrary order. Note that by Lemma 6.2 $C_p$ is extendible if and only if $C'_p$ is extendible. By definition of the switch operation, vertex $v$ is isolated in $G'$ and hence the chord $\overline{v}$ does not intersect any other chord of $C'_p$. Thus we can open $C'_p$ along the chord $\overline{v}$, which means that we receive a linear permutation diagram $D'_p$, where the vertices of the inner circle of $C'_p$ are distributed along the upper and the vertices of the outer circle along the lower line. The leftmost vertex along both horizontal lines is $v$, followed by the remaining vertices according to their counter clockwise order in $C'_p$. By Lemma 6.1 $C'_p$ is extendible if and only if $D'_p$ is extendible. Since $G'$ is a permutation graph and $D'_p$ is the corresponding partial permutation diagram, if existent, we get a representation $D'$ of the entire graph $G'$ by applying the algorithm to solve REPEXT(Perm) presented in Section 5.1. Now, we can transform $D'$ back into a circular permutation diagram $C'$. Finally, to obtain a circular permutation diagram $C$ of $G$ extending $C_p$, we have to switch all the chords in $C'$, belonging to vertices that $v$ is adjacent to in $G$.

By Theorem 5.6, REPEXT(Perm) can be solved in $\mathcal{O}(n+m)$ time for permutation graphs with $n$ vertices and $m$ edges. For solving REPEXT(CPerm), additionally we need to switch the chords belonging to the vertices in the neighbourhood of vertex $v$ twice, which can be done in constant time [Sri96], and transform a circular permutation diagram into a linear permutation diagram and the other way round, which can be done in linear time. Hence in total, we need $\mathcal{O}(n+m)$ time to solve REPEXT(CPerm). □

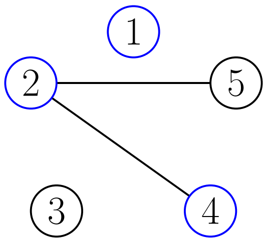
**Example 6.4.** *This Example shall illustrate the described reduction of REPEXT(CPerm) to REPEXT(Perm). Consider the following graph $G$ and the subgraph $H$ induced by the blue vertices $1, 2$ and $4$.*



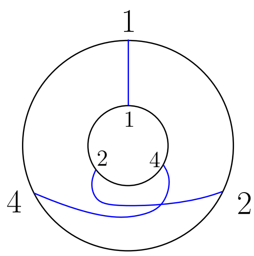

*Given the following circular permutation diagram $C_p$ representing $H$ we want to examine whether $C_p$ can be extended to a circular permutation diagram $C$ representing the entire graph $G$ such that $C$ is isomorphic to $C_p$ on the vertices of the subgraph $H$.*



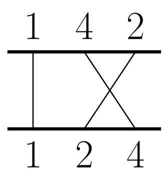

*Let $G'$ be the following permutation graph which we receive by switching all neighbours of vertex 1 in $G$, namely 2 and 5 one at a time. The blue subgraph induced by the vertices of $H$ we denote by $H'$.*



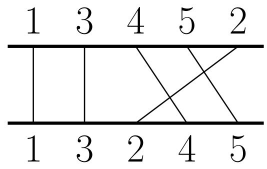

*We get a circular permutation diagram $C'_p$ representing $H'$ by switching all chords in $C_p$ corresponding to vertices adjacent to 1 in $H$, which is in our case only vertex 2.*



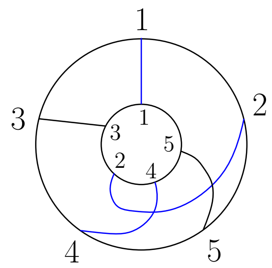

*Then $C_p$ can be extended to a circular permutation diagram $C$ representing $G$ with the desired properties if and only if $C'_p$ can be extended to a circular permutation diagram $C'$ representing $G'$ such that $C'$ is isomorphic to $C'_p$ on the vertices of the subgraph $H'$. To receive such a circular permutation diagram $C'$ we first open $C'_p$ along the chord corresponding to vertex 1 and thus receive a linear permutation diagram $D'_p$ representing $H'$.*



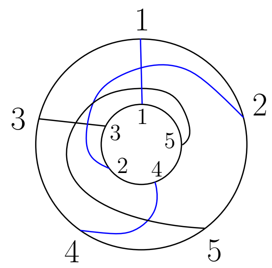
*By applying the algorithm from Section 5.1 we receive a permutation diagram $D'$ representing $G'$ that is isomorphic to $D'_p$ on the subgraph $H$.*




*Now by transforming $D'$ back into a circular permutation diagram we receive the circular permutation diagram $C'$ representing $G'$ such that $C'$ and $C'_p$ are isomorphic on the vertices of $H$.*



*Hence $C_p$ is extendible to a circular permutation diagram $C$ with the desired properties and we get $C$ by switching all chord in $C'$ corresponding to vertices adjacent to 1 in $G$, namely 2 and 5.*



## 6.2 The Simultaneous Representation Problem

The simultaneous representation problem for $r$-sunflower circular permutation graphs $G_1, G_2, \dots, G_r$ sharing an induced subgraph $H$ deals with the question whether there exist circular permutation diagrams $C_1, C_2, \dots, C_r$ such that $C_i$ represents $G_i$ for every $1 \leq i \leq r$ and all $C_i$s are isomorphic on the shared vertices of $H$. This means that the order in which the vertices of $H$ appear along the inner and the outer circle respectively is the same for every $C_i$ and the chord $\bar{v}$ corresponding to $v$ is bent in the same direction in every $C_i$. In this section we show that SimRep(CPerm) can be solved via a reduction to SimRep(Perm).

**Lemma 6.5.** *Let $G_1, G_2, \dots, G_r$ be $r$-sunflower circular permutation graphs sharing an induced subgraph $H$. Let $v$ be a vertex in $H$ and let $G'_i$ be the graph we receive by switching all neighbours of vertex $v$ in $G_i$ for $i \in \{1, \dots, r\}$. Then $G_1, G_2, \dots, G_r$ are simultaneous circular permutation graphs if and only if $G'_1, G'_2, \dots, G'_r$ are simultaneous permutation graphs.*

*Proof.* We already know that $G'_1, G'_2, \ldots, G'_r$ are indeed permutation graphs. Now assume that $G_1, G_2, \ldots, G_r$ are simultaneous circular permutation graphs. Then there exist circular permutation diagrams $C_1, C_2, \ldots, C_r$ such that for every $1 \leq i \leq r$, $C_i$ represents $G_i$ and all $C_i$s are isomorphic on the shared subgraph $H$. We get a circular permutation diagram $C'_i$ representing $G_i$ by switching all chords corresponding to neighbours of vertex $v$ in $C_i$. Since the order of the vertices along the inner and outer circle is not affected by the switch-operation we know that over all $C'_i$s the order in which the vertices of $H$ appear along the outer and the inner circle respectively is the same. We also know that the bending direction of every chord corresponding to a vertex in $H$ is the same in all $C'_i$s since the bending direction of chords corresponding to neighbours of $v$ in $H$ is changed in all $C'_i$s. Chords corresponding to vertices not adjacent to $v$ in $H$ are not switched. Recall that after switching all neighbours of chord $\bar{v}$ in a circular permutation diagram no chord is intersecting $\bar{v}$ any more and hence we receive a permutation diagram $D'_i$ representing $G'_i$ by opening $C'_i$ along the chord $\bar{v}$. Then the $D'_i$s are also isomorphic on $H$ and hence $G'_1, G'_2, \ldots, G'_r$ are simultaneous permutation graphs.

For the other direction assume that $G'_1, G'_2, \ldots, G'_r$ are simultaneous permutation graphs. Then there exist permutation graphs $D'_1, D'_2, \ldots, D'_r$ such that for every $1 \in \{1, \ldots, r\}$, $D'_i$ represents $G'_i$ and all $D'_i$s are isomorphic on the shared subgraph $H$. Recall that we can transform every linear permutation diagram $D'_i$ into a circular permutation diagram $C'_i$ representing $G'_i$. Note that the $C'_i$s are also the same on $H$. Now we receive a circular permutation diagram $C_i$ representing $G_i$ by switching all chords that correspond to vertices adjacent to $v$. Analogously to the other direction the $C_i$s are still isomorphic on $H$ since the switch operation does not change the order of the vertices along the outer or the inner circle of a circular permutation diagram and chords corresponding to vertices in $H$ are either switched in every $C_i$ or in none of them. $\square$

**Corollary 6.6.** *The problem* SimRep(CPerm) *can be solved in $\mathcal{O}(n^2)$ time for $r$-sunflower circular permutation graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_r = (V_r, E_r)$, where $n = \sum_{i=1}^{r} |V_i|$.*

*Proof.* Let $v$ be a vertex in $H$ and let $G'_i$ be the graph we receive by switching all neighbours of vertex $v$ in $G_i$ for $i \in \{1, \ldots, r\}$. By Lemma 6.5 to solve SimRep(CPerm) for $G_1, G_2, \ldots, G_r$ it suffices to solve SimRep(Perm) for $G'_1, G'_2, \ldots, G'_r$. In Section 5.2.1 we have seen that SimRep(perm) can be solved in $\mathcal{O}(n^2)$ time for $r$-sunflower permutation graphs $G'_1 = (V_1, E'_1), G'_2 = (V_2, E'_2), \ldots, G'_r = (V_r, E'_r)$ where $n = \sum_{i=1}^{r} |V_i|$. As described in the proof of Lemma 6.5, if $G_1, G_2, \ldots, G_r$ are simultaneous circular permutation graphs we get corresponding simultaneous representation by transforming simultaneous linear permutation diagrams representing $G'_1, G'_2, \ldots, G'_r$ into circular permutation diagrams and switching all chords corresponding to a neighbour of vertex $v$ in $H$. Since this can be done in quadratic time and for every $G_i$ we can compute $G'_i$ in $\mathcal{O}(n^2)$ time, in total we also need $\mathcal{O}(n^2)$ time to solve SimRep(CPerm) for $G_1, G_2, \ldots, G_r$. $\square$

# 7. Conclusion

In this thesis we examined two already well-studied problems, namely the partial representation problem and the simultaneous representation problem, for the three graph classes of comparability, permutation and circular permutation graphs respectively. The problem whether a given partial orientation of a comparability graph can be extended to transitive orientation of the entire graph has already been shown to be solvable in $\mathcal{O}((n+m)\Delta)$ time for comparability graphs with $n$ nodes, $m$ edges and maximum degree $\Delta$ [KKKW12]. The best know runtime so far to decide whether a given permutation diagram $D'$ representing a subgraph of a permutation graph can be extended to a permutation diagram $D$ representing the entire graph was $\mathcal{O}(n^3)$ [KKKW12]. We presented $\mathcal{O}(n+m)$ algorithm based on the concept of modular decomposition for both problems and showed that even the partial representation problem for circular permutation graphs that has not been studied so far, is solvable in $\mathcal{O}(n+m)$ time.

Jampani and Lubiw showed that we can decide in $\mathcal{O}(nm)$ time whether for given $r$-sunflower comparability graphs sharing an induced subgraph $H$ there exist representations that are all isomorphic on $H$ [JL10]. To solve the simultaneous representation problem for permutation graphs they gave an $\mathcal{O}(n^3)$ algorithm [JL10]. Again with the concept of modular decomposition we were able to show that the simultaneous representation for comparability graphs can be solved in $\mathcal{O}(n+m)$ time for $r$-sunflower comparability graphs where $n$ is the sum of the number of vertices of every input graph. Based on this result we gave $\mathcal{O}(n^2)$ algorithms to solve the simultaneous representation problem for $r$-sunflower permutation and circular permutation graphs, where $n$ is the sum of the number of vertices of every input graph. For the non-sunflower case we showed that it is an `NP`-complete problem to decide whether for $r$ permutation graphs there exist representations that are all pairwise isomorphic on the pairwise shared subgraph.

It remains an open problem whether the simultaneous representation problem for $r$-sunflower permutation and circular permutation graphs can be solved in less than quadratic time. Furthermore it would be interesting to examine whether we can solve the partial representation and the simultaneous representation problem with the concept of modular decomposition also for other graph classes and thereby maybe achieve better runtimes than the ones known so far. There may be also other related problems that can be solved for comparability, permutation and circular permutation graphs with the concept of the modular decomposition.

# Bibliography


[APT79] Bengt Aspvall, Michael F Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[BL76] Kellogg S Booth and George S Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of computer and system sciences*, 13(3):335–379, 1976.

[Boo75] Kellogg Speed Booth. *PQ-tree algorithms.* PhD thesis, University of California, Berkeley, 1975.

[BR15] Thomas Bläsius and Ignaz Rutter. Simultaneous pq-ordering with applications to constrained embedding problems. *ACM Transactions on Algorithms (TALG)*, 12(2):1–46, 2015.

[EPL72] Shimon Even, Amir Pnueli, and Abraham Lempel. Permutation graphs and transitive graphs. *Journal of the ACM (JACM)*, 19(3):400–410, 1972.

[Gal67] Tibor Gallai. Transitiv orientierbare graphen. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(1-2):25–66, 1967.

[Gol04] Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs.* Elsevier, 2004.

[GT85] Harold N Gabow and Robert Endre Tarjan. A linear-time algorithm for a special case of disjoint set union. *Journal of computer and system sciences*, 30(2):209–221, 1985.

[HP10] Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.

[JL10] Krishnam Raju Jampani and Anna Lubiw. Simultaneous interval graphs. In *International Symposium on Algorithms and Computation*, pages 206–217. Springer, 2010.

[Kah62] Arthur B Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562, 1962.

[KKKW12] Pavel Klavík, Jan Kratochvíl, Tomasz Krawczyk, and Bartosz Walczak. Extending partial representations of function graphs and permutation graphs. In *European Symposium on Algorithms*, pages 671–682. Springer, 2012.

[KKOS15] Pavel Klavík, Jan Kratochvíl, Yota Otachi, and Toshiki Saitoh. Extending partial representations of subclasses of chordal graphs. *Theoretical Computer Science*, 576:85–101, 2015.

[KKV11] Pavel Klavík, Jan Kratochvíl, and Tomáš Vyskočil. Extending partial representations of interval graphs. In *International Conference on Theory and Applications of Models of Computation*, pages 276–285. Springer, 2011.

[Mon03] Fabien de Montgolfier. *Décomposition modulaire des graphes: théorie, extensions et algorithmes*. PhD thesis, Montpellier 2, 2003.

[MS99] Ross M McConnell and Jeremy P Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1-3):189–241, 1999.

[Opa79] Jaroslav Opatrny. Total ordering problem. *SIAM Journal on Computing*, 8(1):111–114, 1979.

[PLE71] Amir Pnueli, Abraham Lempel, and Shimon Even. Transitive orientation of graphs and identification of permutation graphs. *Canadian Journal of Mathematics*, 23(1):160–175, 1971.

[RU82] Doron Rotem and Jorge Urrutia. Circular permutation graphs. *Networks*, 12(4):429–437, 1982.

[Sri96] R Sritharan. A linear time algorithm to recognize circular permutation graphs. *Networks: An International Journal*, 27(3):171–174, 1996.

[Tar79] Robert Endre Tarjan. Applications of path compression on balanced trees. *Journal of the ACM (JACM)*, 26(4):690–715, 1979.