WORD-REPRESENTABILITY OF GRAPHS VIA SPLIT DECOMPOSITION

TITHI DWARY

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, INDIA

MARCH 2026

Word-Representability of Graphs via Split Decomposition

by

Tithi Dwary

Department of Mathematics

submitted in fulfillment of the requirements of the degree of Doctor of Philosophy

to the

Indian Institute of Technology Guwahati
Guwahati - 781039, India

March 2026

To

My Parents

&

My Gurus

Certificate

This is to certify that the thesis entitled *Word-Representability of Graphs via Split Decomposition* submitted by *Ms. Tithi Dwary* to the Indian Institute of Technology Guwahati, for the award of the Degree of Doctor of Philosophy, is a record of the original bona fide research work carried out by her under my supervision and guidance. The thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Guwahati	Dr. K. V. Krishna
March 2026	Supervisor

Acknowledgements

I would like to take this opportunity to express my sincere appreciation to all those who have supported and assisted me, directly or indirectly, in reaching this stage and completing this thesis.

First and foremost, I would like to express my deepest gratitude to my thesis supervisor, Prof. K. V. Krishna, for his excellent guidance and unwavering support in various ways throughout my Ph.D. journey. His clarity of thought and the interesting questions he raised during our discussions greatly helped me to shape this thesis in the way I had envisioned. His dedication to work, patience, and strong work ethic have been a constant source of inspiration to me. I am especially grateful to him for carefully reviewing each of my manuscripts and the thesis line by line, and for helping me present them in their best possible form. It has truly been a pleasure working with him, and I consider myself very fortunate to have him as my supervisor.

I sincerely acknowledge the members of my Doctoral Committee – Prof. Anupam Saikia, Prof. Rupam Barman, and Dr. P. A. S. Sree Krishna – for carefully reviewing my work and for their valuable suggestions and guidance. I am also thankful to Dr. H. Ramesh and Dr. Sriram Bhyravarapu for attending all my seminars and for providing insightful feedback and constructive suggestions on my research work. I would like to thank all the faculty members of the Department of Mathematics for their assistance in various ways throughout my research period.

I am grateful to the Indian Institute of Technology Guwahati for providing financial assistance and the necessary facilities to carry out and complete my research work. I also thank the office and technical staff of the Department of Mathematics for their assistance and cooperation during my research tenure.

I sincerely acknowledge all my teachers from school, college, and university for their continuous encouragement and guidance throughout my academic journey.

I express my profound gratitude to my parents and my sister for their unconditional love, unwavering support, and firm belief in my abilities. The sacrifices and efforts my parents have made since my childhood to enable me to reach this stage have been a constant source of strength and motivation throughout my journey.

I feel truly fortunate to have had such wonderful colleagues and friends - Anuran, Khyodeno, Sagar, Soni, Sakshi, Anuj, Anjali, Achyuta, Puspanjali, Supratim, and Sachin - during my time at IIT Guwahati. The many memorable moments I shared with them made this journey enjoyable. Anuj, Anjali, and Achyuta made my years on campus especially memorable. I spent the most beautiful and cherished moments with them, and I am grateful from the bottom of my heart for their constant support through both the good and the difficult times.

Finally, I would like to take this opportunity to thank the Almighty God for the blessings and for all the good things that have come into my life thus far.

IIT Guwahati Tithi Dwary

Abstract

The theory of word-representable graphs lies at the intersection of graph theory and combinatorics on words. This thesis uses a graph decomposition technique, viz., split decomposition, to study the theory of word-representable graphs. Further, the thesis constructs minimum-word-representants for certain classes of graphs.

In this thesis, a characterization of word-representable graphs with respect to split decomposition is obtained. Accordingly, the representation number of a word-representable graph is investigated in terms of the representation numbers of its split components. As a consequence, a characterization of word-representable graphs using modular decomposition is presented. In this context, a complete answer to the open problem posed by Kitaev and Lozin on the word-representability of the lexicographical product of graphs is obtained. Further, using split decomposition, the word-representability for specific subclasses of perfect graphs, viz., parity graphs, and well-partitioned chordal (WPC) graphs is established. In this connection, while it is shown that all parity graphs are word-representable, a characterization and a polynomial-time recognition algorithm for the word-representability of WPC graphs are presented. Moreover, the representation numbers of word-representable WPC graphs and that of split graphs are determined. In addition, a characterization of the class of (double-)arborescences in terms of their minimal split-decomposition trees is obtained and their minimum-word-representants are constructed. It is an open problem in the literature to find the classes of word-representable graphs whose minimum-word-representants are of length $2n-\kappa$, where n is the number of vertices of the graph and κ is its clique number. Contributing to the open problem, the class of treelike permutation graphs is shown as the first example satisfying the criterion of the aforementioned open problem, for an arbitrary κ.

Contents

Introduction

The theory of word-representable graphs has emerged as a promising area at the intersection of graph theory and combinatorics on words, offering ways to encode certain graphs by words. The word-representable graphs can be seen as a natural generalization of well-known class of comparability graphs, while these graphs were introduced to study the word problem of Perkins semigroups by Kitaev and Seif [2008]. This thesis investigates the word-representability of graphs through a graph decomposition technique, namely split decomposition. Further, the thesis constructs word-representants of minimum length for specific classes of graphs, addressing an open problem in the literature.

A graph is said to be a comparability graph if it admits a transitive orientation. Comparability graphs are extensively studied in the literature and are of particular interest from the perspective of partially ordered sets (posets). Every comparability graph induces a poset, based on its transitive orientation. While there can be multiple transitive orientations of a comparability graph, the posets induced by a comparability graph have the same dimension [Trotter et al., 1976]. The concept of transitive orientation of a graph was generalized to the concept of semi-transitive orientation in [Halldórsson et al., 2016]. Moreover, in [Halldórsson et al., 2016], it was proved that the graphs which admit semi-transitive orientations are precisely the word-representable graphs. Thus, every comparability graph is a word-representable graph.

A graph $G = (V, E)$ is called a word-representable graph, if there exists a word w over V such that for all $a, b \in V$, a and b are adjacent in G if and only if a and b alternate in w. In this case, we say w is a word-representant of G. If no such word exists, then the graph is said to be non-word-representable. The class of word-representable graphs is interesting from a graph-theoretical point of view, as it contains several important classes of graphs such as circle graphs, 3-colorable graphs, comparability graphs and cover graphs. Moreover, it was observed that the class of triangle-free word-representable graphs is precisely the class of cover graphs of posets [Kitaev and Lozin, 2015; Pretzel, 1985]. Thus, recognizing whether a given graph is word-representable is an NP-complete problem, as recognizing cover graphs of posets is NP-complete [Brightwell, 1993; Kitaev and Lozin, 2015]. However, certain computational problems on graphs that are hard, in general, could be solved in polynomial time for word-representable graphs, e.g., maximum clique problem [Halldórsson et al., 2011]. It was observed in [Halldórsson et al., 2011] that this class of graphs has connection to scheduling problems.

A word-representable graph G is said to be k-word-representable if there is a word-representant of G in which every letter appears exactly k times; such words are called k-uniform words. In [Kitaev and Pyatkin, 2008], it was proved that every word-representable graph is k-word-representable, for some k. The smallest number k such that a graph G is k-word-representable is called the representation number of G and it is denoted by $\mathcal{R}(G)$. It is evident that $\mathcal{R}(G) = 1$ if and only if G is a complete graph. Further, in [Halldórsson et al., 2011], the class of word-representable graphs with representation number at most two is characterized by the class of circle graphs. In general, the problem of determining the representation number of a word-representable graph is computationally hard [Halldórsson et al., 2011]. Nonetheless, for certain graph classes, the representation numbers were obtained in the literature. For example, in [Glen et al., 2018], the representation number of a crown graph was determined. In contrast to the word-representable graphs with representation number two, not many graphs with representation number three are known in the literature. The Cartesian product of K_m and K_2 [Broere and Zantema, 2019], prisms [Kitaev, 2013], the Petersen graph [Kitaev and Lozin, 2015], stacked book graphs, and melon graphs [Mozhui, 2025; Mozhui and Krishna, 2025a, 2026] are known to have the representation number at most three. Recently, the representation number of a k-cube was investigated in [Hefty et al., 2024].

A graph G is said to be permutationally representable if it can be represented by a word that is concatenation of some number of permutations on the vertices of G. If G is represented by $p_1 p_2 \cdots p_k$, where each p_i is a permutation on the vertices of G, then we say G is a permutationally k-representable graph. It was shown in [Kitaev and Seif, 2008] that a graph is permutationally representable if and only if it is a comparability graph. The permutation-representation number (in short, *prn*) of a comparability graph G, denoted by $\mathcal{R}^p(G)$, is the smallest number k such that G is permutationally k-representable. It is evident that for a comparability graph G, $\mathcal{R}(G) \leq \mathcal{R}^p(G)$. Moreover, it was observed that the *prn* of a comparability graph is same as the dimension of its induced poset [Halldórsson et al., 2011; Mozhui and Krishna, 2025b]. Whereas the class of comparability graphs with *prn* one is the class of complete graphs, the class of graphs with *prn* at most two is characterized as the class of permutation graphs (cf. [Mozhui and Krishna, 2023]). While there is no classification available for the class of graphs with *prn* more than two, it is known that the problem of determining the *prn* of a comparability graph is computationally hard [Yannakakis, 1982].

The word-representability of various graph classes was characterized using semi-transitive orientations in the literature, e.g., Kneser graphs [Kitaev and Saito, 2020], triangle-free graphs [Kitaev and Pyatkin, 2023], (extended-)Mycielski graphs [Hameed, 2024; Kitaev and Pyatkin, 2025]. In contrast, neither a characterization nor a recognition algorithm for the word-representability of perfect graphs is known in the literature. In fact, these problems remain open even for chordal graphs, a well-known subclass of perfect graphs. However, these problems were completely settled for one of the subclasses of chordal graphs, namely split graphs, through a series of articles [Chen et al., 2022; Kitaev et al., 2021; Kitaev and Pyatkin, 2024]. Despite this progress, the representation number of word-representable split graphs was not known. Starting from the split graphs, there is a hierarchy of well-known subclasses of chordal graphs as per the following [Ahn et al., 2022]:

$$\text{split graphs} \subset \text{starlike graphs} \subset \text{WPC graphs} \subset \text{chordal graphs}$$

Ahn et al. [2022] proposed well-partitioned chordal (WPC) graphs as a framework to bridge complexity gaps in problems that remain hard on chordal graphs but become tractable on split graphs. Several problems, e.g., tree 3-spanner problem, transversal of longest paths and cycles, geodetic set problem which are either hard or open on chordal graphs were proved to be polynomial-time solvable on WPC graphs [Ahn et al., 2022; Chakraborty et al., 2024]. It has been recently established that WPC graphs are multi-word-representable (a generalized notion of word-representability) using two words [Kenkireth et al., 2025]. However, characterizing and recognizing word-representability, and determining representation number were open for WPC graphs.

In addition to uniform word-representants, minimum-word-representants of a word-representable graph have also received attention in the literature. A minimum-word-representant of a word-representable graph G is a word-representant which has the shortest length among all the word-representants of G, and its length is denoted by $\ell(G)$. Let G be a word-representable graph on n vertices. Then, it is evident that $\ell(G) \geq n$. Since the class of circle graphs characterizes the 2-word-representable graphs [Halldórsson et al., 2011], if G is a circle graph, we have $\ell(G) \leq 2n$. In the seminal work on minimum-word-representants, Gaetz and Ji [2020] considered the subclasses of circle graphs, namely cycles and trees, and provided explicit formulae for both the length and the number of their minimum-word-representants.

Srinivasan and Hariharasubramanian [2024] proved that there is no circle graph G with $\ell(G) = 2n$ and an edgeless graph G is the only circle graph having $\ell(G) = 2n-1$. Also, they showed that $\ell(G) = 2n-2$ for a non-complete triangle-free circle graph G containing at least one edge. For a word-representable graph G with clique number κ, while they observed that $\ell(G)$ need not be $2n-\kappa$, they proved that $\ell(G) \geq 2n-\kappa$. In this connection, they posed an open problem to find classes of word-representable graphs G such that $\ell(G) = 2n - \kappa$.

A graph is called a treelike comparability graph if it admits a transitive orientation such that the underlying graph of its transitive reduction is a tree; in that case, the corresponding orientation is called a treelike orientation. It was established in [Cornelsen and Di Stefano, 2009] that a treelike orientation of a comparability graph, if exists, is unique (up to isomorphism of the corresponding oriented graphs or of the graphs by reversing the whole orientation). If a treelike comparability graph is also a permutation graph, it is called a treelike permutation graph. A treelike comparability graph G with an all-adjacent vertex is called a double-arborescence, i.e., there is a vertex r (called a root of G) which is adjacent to all other vertices of G. In addition, under the treelike orientation, if the root r is a source (or a sink), then G is called an arborescence. A treelike comparability graph G is called a path of k double-arborescences (or simply, a path of double-arborescences) if G consists of k double-arborescences and a path (ignoring the directions) connecting their roots. Arborescences were first studied by Wolk [1962, 1965] and were characterized as (C_4, P_4)-free graphs, where C_n is a cycle on n vertices and P_n is a path on n vertices. Further, Golumbic [1978] characterized the arborescences as trivially perfect graphs. Based on the aforementioned characterizations, linear-time algorithms for recognizing the arborescences were obtained in the literature (see [Chu, 2008; Yan et al., 1996]). It was proved in [Cornelsen and Di Stefano, 2009] that a graph is a treelike permutation graph if and only if it is a path of double-arborescences. Treelike comparability graphs were also characterized in [Cornelsen and Di Stefano, 2009]. However, no characterization was available for double-arborescences in the literature. Further, being a subclass of distance-hereditary graphs, treelike comparability graphs are circle graphs and hence, are 2-word-representable. Thus, for a treelike comparability graph G on n vertices, $\ell(G) \leq 2n$. However, the exact value of $\ell(G)$ for treelike comparability graphs or for any of their above-mentioned subclasses was not known.

Graph decomposition techniques are powerful tools in graph theory. They are widely used to characterize and recognize various classes of graphs. For example, the concept of modular decomposition of a graph was first introduced by Gallai [1967] and subsequently employed to characterize comparability graphs. Further, a linear-time recognition algorithm for comparability graphs was obtained using modular decomposition in [McConnell and Spinrad, 1999]. Modular decomposition was useful in characterizing several other graph classes, including cographs, permutation graphs, and interval graphs (cf. [Golumbic, 2004]). The concept of split decomposition, introduced by Cunningham [1982], is a more general graph decomposition technique and it was useful for solving some NP-hard problems [Rao, 2004, 2008]. It was also used for characterizing and recognizing certain classes of graphs such as distance-hereditary graphs [Di Stefano, 2012; Gavoille and Paul, 2003], treelike comparability graphs [Cornelsen and Di Stefano, 2009], circle graphs [Bouchet, 1987; Gabor et al., 1989; Spinrad, 1994], and parity graphs [Cicerone and Di Stefano, 1999; Dahlhaus, 2000].

This thesis focuses on the study of the word-representability of graphs using split decomposition and further constructs minimum-word-representants for certain graph classes. In fact, it characterizes word-representable graphs and studies their representation numbers with respect to split decomposition. As a consequence, the thesis also presents a characterization of word-representable graphs using modular decomposition. In this context, the thesis provides a complete answer to the open problem posed by Kitaev and Lozin [2015] on the word-representability of the lexicographical product of graphs. Further, using split decomposition, the thesis establishes the word-representability for specific subclasses of perfect graphs, namely parity graphs and well-partitioned chordal graphs. While all parity graphs are shown to be word-representable, the thesis provides a characterization and presents a polynomial-time recognition algorithm for the word-representability of WPC graphs. The thesis also determines the representation numbers of word-representable WPC graphs and that of split graphs. In addition, this thesis characterizes the class of (double-)arborescences in terms of their minimal split-decomposition trees and constructs their minimum-word-representants. Addressing an open problem posed in [Srinivasan and Hariharasubramanian, 2024], the thesis proves that the length of a minimum-word-representant of any treelike permutation graph is $2n - \kappa$, where n is the number of vertices of the graph and κ is its clique number. Notably, the class of treelike permutation graphs is the first example satisfying the criterion of the aforementioned open problem, for an arbitrary κ.

In this thesis, the necessary background material and related results are presented in Chapter 1. Starting from Chapter 2, the contributions of the thesis have been organized into four chapters.

Chapter 2: Split Decomposition

Chapter 3: Modular Decomposition

Chapter 4: Well-Partitioned Chordal Graphs

Chapter 5: Minimum-Word-Representants

Chapter 2. In this chapter, we show that the class of word-representable graphs is closed under split recomposition and obtain a characterization of word-representable graphs with respect to split decomposition. We also determine the representation number of a word-representable graph in terms of its split components. Accordingly, we establish the word-representability of a subclass of perfect graphs, known as parity graphs. The class of parity graphs includes both bipartite graphs and distance-hereditary graphs. While the class of comparability graphs is not closed under split recomposition, we obtain a characteristic property by which the recomposition of comparability graphs is a comparability graph. Consequently, we establish the *prn* of the resulting comparability graph. Further, we introduce a subclass of comparability graphs, which we call *prn*-irreducible graphs, and study their split recomposition. We provide criteria such that the split recomposition of two *prn*-irreducible graphs is a comparability graph and accordingly determine the *prn* of the resulting graph. We also observe that the split recomposition of *prn*-irreducible graphs is not *prn*-irreducible. The work contained in this chapter has been published in the journal *Discrete Applied Mathematics* [Dwary and Krishna, 2024].

Chapter 3. In this chapter, we characterize word-representable graphs with respect to the modular decomposition. We also determine the representation number of a word-representable graph in terms of the permutation-representation numbers of the subgraphs induced by modules and the representation number of the associated quotient graph. Furthermore, utilizing these results, we establish the word-representability of 3-leaf power graphs and obtained their *prn* and representation numbers. Additionally, we provide a necessary and sufficient condition under which the lexicographical product of two word-representable graphs is word-representable, thereby offering a complete answer to the open problem posed in [Kitaev and Lozin, 2015, Chapter 7]. The work contained in this chapter was presented at the International Conference on Graph Theory and its Applications (ICGTA 2024), Bengaluru, 20-22 June 2024; and subsequently accepted for publication in the journal *Communications in Combinatorics and Optimization* [Dwary and Krishna, 2025].

Chapter 4. In this chapter, we study the word-representability of well-partitioned chordal (WPC) graphs. We show that every split component of the minimal split decomposition of a WPC graph is a split graph, which gives us a characterization of word-representable WPC graphs. Based on this characterization, we provide a polynomial-time algorithm for recognizing word-representable WPC graphs. We further determine the representation number of word-representable WPC graphs. Through an algorithmic procedure, we first establish that the representation number of a word-representable split graph is at most three, and subsequently extend this result to word-representable WPC graphs. We obtain a minimal forbidden induced subgraph characterization of circle graphs restricted to WPC graphs and use it to classify the WPC graphs as well as the split graphs with respect to their representation numbers. Further, we characterize WPC comparability graphs and conclude that the permutation-representation number of a WPC comparability graph is at most four. Although it remains open whether this bound is tight, we focus on a subclass of WPC graphs, namely starlike graphs. The starlike graphs include split graphs. We prove that the permutation-representation number of a starlike comparability graph is at most three, and the bound is tight. Moreover, we provide a characterization of starlike comparability graphs. The part of this chapter concerning the representation number of word-representable split graphs is to appear in the journal *Siberian Advances in Mathematics* [Dwary et al., 2026]. The remaining results on the word-representability of WPC graphs have been submitted to a journal [Dwary and Krishna, 2026c].

Chapter 5. In this chapter, we employ the notion of split-decomposition trees and characterize double-arborescences as well as arborescences. In addition, we also construct minimum-word-representants of treelike permutation graphs. We first characterize double-arborescences as P_4-free treelike comparability graphs. Next, we introduce a notion called s-leaf-path in the minimal split-decomposition tree of a distance-hereditary graph. Using the concept of s-leaf-path, we provide yet another characterization for the well-known class of cographs. Further, using split decomposition, we obtain an alternative proof for the characterization of arborescences (see [Wolk, 1965, Theorem 3] and [Wolk, 1962]) that arborescences are precisely

(C_4, P_4)-free graphs. Consequently, we obtain characterizations of arborescences and double-arborescences with respect to their minimal split-decomposition trees. In the context of word-representable graphs, it is an open problem in the literature to find the classes of word-representable graphs whose minimum-word-representants are of length $2n - \kappa$, where n is the number of vertices of the graph and κ is its clique number. Contributing to the open problem, we prove that the class of tree-like permutation graphs is one such class. To establish this result, we first construct minimum-word-representants for double-arborescences in polynomial time and then extend this construction to treelike permutation graphs. The part of this chapter on double-arborescences has been published in the journal *Discrete Applied Mathematics* [Dwary and Krishna, 2026a]. The remaining work on the treelike permutation graphs was presented at the 51st International Conference on Current trends in Theory and Practice of Computer Science (SOFSEM 2026), Kraków, Poland, February 09-13, 2026; and has been published in the Lecture Notes in Computer Science (LNCS) series by Springer [Dwary and Krishna, 2026b].

Epilogue. Every contributory chapter is concluded with future directions based on the work embedded in the chapter.

1

Preliminaries

This chapter provides the necessary background material for the thesis and fixes the notation and terminology used throughout. To make the thesis as self-contained as possible, we present all required definitions and state the relevant results used in the thesis. We begin by recalling the fundamental concepts of graphs. We then define partially ordered sets and describe their relationship with comparability graphs. Next, we present the concept of word-representable graphs and summarize their key properties. We proceed with the split decomposition of graphs and provide the essential background for its application. We also present the concept of modular decomposition of graphs. Finally, we review certain graph classes and related results that are utilized in this thesis.

1.1 Graphs

In this section, we provide some basic concepts and examples of graphs that are often used in this thesis. For the concepts on graphs that are not defined in the thesis, one may refer to [West, 1996].

Definition 1.1.1 (Graph)**.** A *graph* G is a pair (V, E), where V is a finite set and E is a set of 2-element subsets of V. The elements of V and E are called *vertices* and *edges* of the graph G, respectively. For $a, b \in V$, if $\{a, b\} \in E$, we say a and b are *adjacent* in G; otherwise, a and b are said to be *non-adjacent* in G. In this thesis, an edge $\{a, b\} \in E$ is denoted by $\overline{ab}$.

Definition 1.1.2 (Induced subgraph)**.** Let $G = (V, E)$ be a graph. A graph $H = (V', E')$ is said to be a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. A subgraph (V', E') of G is called an *induced subgraph* (induced by its vertex set V'), denoted by $G[V']$, if E' is the set of all 2-element subsets of V' that are in E.

Definition 1.1.3 (Neighborhood)**.** Let $H = (V', E')$ be a subgraph of a graph G. For $a \in V'$, the *neighborhood* of a in H, denoted by $N_H(a)$, is the set of all vertices adjacent to a in H. When the context is clear with respect to the subgraph under consideration, we simply use the notation $N(a)$ for the neighborhood of a. For $A \subseteq V'$, the neighborhood of A, $N_H(A) = \bigcup\limits_{a \in A} N_H(a) \setminus A$.

Definition 1.1.4 (All-adjacent vertex)**.** A vertex of a graph is said to be an *all-adjacent vertex*, also known as *universal vertex*, if it is adjacent to all other vertices of the graph.

Definition 1.1.5 ($\mathcal{A}$-free graph)**.** For a set of graphs $\mathcal{A}$, a graph G is said to be $\mathcal{A}$*-free* if G does not contain any induced subgraph isomorphic to a graph in $\mathcal{A}$.

Definition 1.1.6 (Orientation)**.** An *orientation* of a graph $G = (V, E)$ is an assignment of direction to each edge, i.e., fixing an order for the vertices of each edge.

Thus, the resulting edge set, say O, is a subset of $V \times V$, is an orientation of G. If an orientation O is assigned to a graph $G = (V, E)$, then the resulting graph is denoted by $\hat{G} = (V, O)$ and it is called an *oriented graph*. For $\overline{ab} \in E$, if $(a, b) \in O$, we write $\overrightarrow{ab}$ indicating the orientation/direction assigned to the edge.

Definition 1.1.7 (Path and Cycle)**.** A *path* in a graph $G = (V, E)$ is a sequence of vertices $a_1, a_2, \ldots, a_k$, written as $a_1 - a_2 - \cdots - a_k$, such that $\overline{a_i a_{i+1}} \in E$ for $1 \leq i < k$. If $a_1 = a_k$, the path is called a *cycle*. If an n-vertex graph is itself a path or a cycle, the graph is denoted by P_n or C_n, respectively.

If G is oriented, we also indicate the direction of each edge in the path. For example, $a_1 \rightarrow a_2 \leftarrow a_3 \rightarrow a_4$ is an oriented path between a_1 and a_4. If all edges on a path are in the same direction, then we say it is a *directed path*. For example, $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4$ is a directed path from a_1 to a_4.

Definition 1.1.8 (Distance)**.** Let $G = (V, E)$ be a graph and $a, b \in V$. If there is a path between a and b in G, the distance form a to b, denoted by $d_G(a, b)$, is the least length of a path (i.e., the number of edges in the path) from a to b in G.

In this thesis, unless otherwise stated, the graphs are *connected graphs*, i.e., the graphs in which there is a path between any pair of vertices.

Definition 1.1.9 (Complete graph)**.** A graph is said to be a *complete graph* if every pair of vertices are adjacent in the graph. A complete graph on n vertices is denoted by K_n.

Definition 1.1.10 (Clique and Clique number)**.** A *clique* in a graph $G = (V, E)$ is a subset of the vertex set V such that it induces a complete subgraph. The size of a maximum clique in G is said to be the *clique number* of G. Depending on the context, we use clique to refer either a set of vertices or to the complete subgraph induced by that set.

Definition 1.1.11 (Edgeless graph and Independent set)**.** A graph is said to be an *edgeless graph* if every pair of vertices are non-adjacent in the graph. A set of vertices in a graph is said to be an *independent set* if it induces an edgeless subgraph.

Definition 1.1.12 (Bipartite graph)**.** A graph $G = (V, E)$ is said to be a *bipartite graph* if there exists a partition $\{A, B\}$ of V such that A and B are independent sets in G. If each vertex of A is adjacent to every vertex of B in G, then G is called a *complete bipartite graph*, and it is denoted by $K_{m,n}$, where $m = |A|$ and $n = |B|$ are the number of elements of A and B, respectively. A complete bipartite graph $K_{1,k}$ is called a *star* and it is denoted by S_k.

Notation 1.1.13. Let $G = (V, E)$ be a graph and A, B be two disjoint subsets of V. The bipartite graph with the vertex set $A \cup B$ and the edge set $\{\overline{ab} \in E \mid a \in A, b \in B\}$ is denoted by $G[A, B]$. We say A *is complete to* B in G if each vertex of A is adjacent to every vertex of B in G.

Definition 1.1.14 (Permutation graph)**.** A graph is called a *permutation graph* if it is an intersection graph[1] of line segments whose endpoints lie on two parallel lines. An example of a permutation graph and its lines representation are given in Figure 1.1.

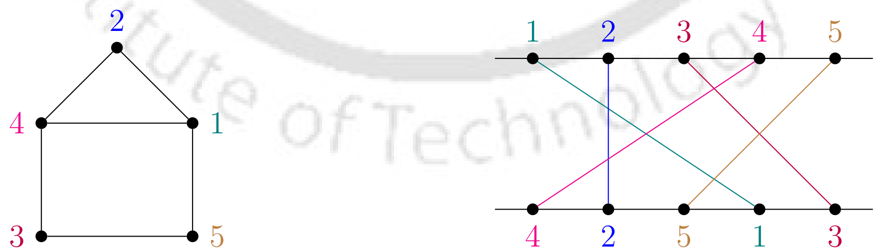


FIGURE 1.1: An example of a permutation graph

[1]An intersection graph is a graph defined over a family of sets/objects as its vertex set in which two vertices are adjacent if and only if the corresponding objects intersect.

Definition 1.1.15 (Circle graph)**.** A graph is called a *circle graph* if it is an intersection graph of chords of a circle. An example of a circle graph and its circle representation are given in Figure 1.2.

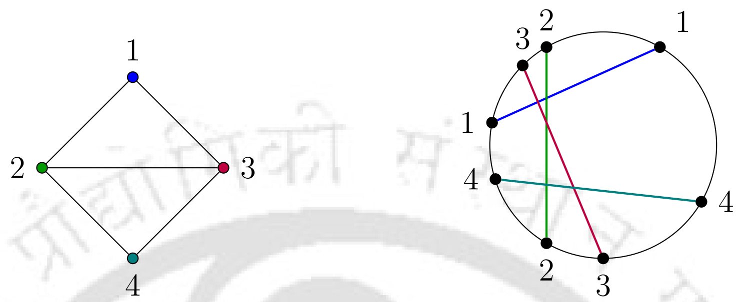


Figure 1.2: An example of a circle graph

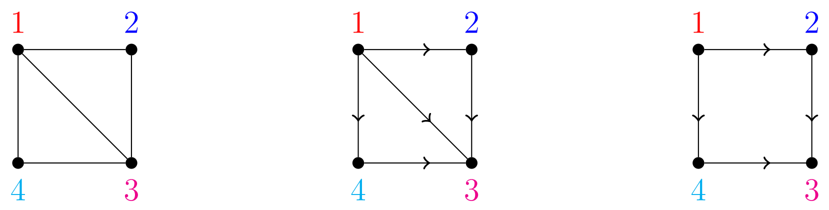
Remark 1.1.16. The class of permutation graphs is a subclass of the class of circle graphs.

1.2 Posets and Comparability Graphs

In this section, we define the concepts of partially ordered sets and comparability graphs and describe the relationship between them. For further details, one may refer to [Trotter, 1992].

Definition 1.2.1 (Poset)**.** A *partially ordered set* (in short, *poset*) is a pair $(P, \prec)$, where P is a nonempty set and $\prec$ is a partial order on P. A poset is often denoted by its underlying set.

Definition 1.2.2 (Linear order and Chain)**.** The elements a, b in a poset $(P, \prec)$ are said to be *comparable* if $a \prec b$ or $b \prec a$; otherwise, we say they are *incomparable*. A partial order on a set P is said to be a *linear order* if any two elements of P are comparable. If a partial order $\prec$ on a set P is a linear order, the poset $(P, \prec)$ is called a *chain*.



$G = (V, E)$ $\hat{G} = (V, O)$ $\tilde{G} = (V, \tilde{O})$

FIGURE 1.3: A transitive orientation O of G and its transitive reduction $\tilde{G}$

Definition 1.2.3 (Realizer and Dimension)**.** A *realizer* of a poset $(P, \prec)$ is a collection of linear orders $\{L_1, L_2, \ldots, L_k\}$ on P such that, for every $a, b \in P$, $a \prec b$ if and only if $a\ L_i\ b$, for all $1 \le i \le k$; i.e., $\prec$ equals $\bigcap_{i=1}^{k} L_i$. The *dimension* of a poset P, denoted by $\dim(P)$, is the smallest positive integer k such that P has a realizer of size k.

Remark 1.2.4 ([Trotter, 1992])**.** A poset has dimension one if and only if it is a chain.

Definition 1.2.5 (Maximal element and Minimal element)**.** An element a in a poset P is called a *maximal element* if there is no element b in P such that $a \prec b$. Similarly, an element a is called *minimal* in P if there is no element b such that $b \prec a$ in P.

We now present the concept of comparability graphs and their connection with posets.

Definition 1.2.6 (Comparability graph and its transitive reduction)**.** A graph $G = (V, E)$ is said to be a *comparability graph* if it admits a *transitive orientation*, i.e., an orientation O in which if $\overrightarrow{ab} \in O$ and $\overrightarrow{bc} \in O$, then the transitive edge $\overrightarrow{ac} \in O$, for all $a, b, c \in V$. The *transitive reduction* of a comparability graph G with respect to a transitive orientation O is the oriented graph $\tilde{G} = (V, \tilde{O})$, where the edge set $\tilde{O}$ is obtained by deleting the transitive edges from O.

Example 1.2.7. An example of a comparability graph $G = (V, E)$, along with a transitive orientation O and its transitive reduction $\tilde{G} = (V, \tilde{O})$, is shown in Figure 1.3.

Remark 1.2.8. Let $G = (V, E)$ be a comparability graph with the transitive reduction $\tilde{G} = (V, \tilde{O})$ of a transitive orientation O of G. For any $a, b \in V$, there is a directed path from a to b in $\tilde{G}$ if and only if $\overrightarrow{ab} \in O$.

Remark 1.2.9. Not all graphs are comparability graphs, e.g., for each $n \geq 2$, C_{2n+1} is not a comparability graph. Also the graph given in Figure 1.4 is not a comparability graph (cf. [Golumbic, 2004, Chapter 5]).

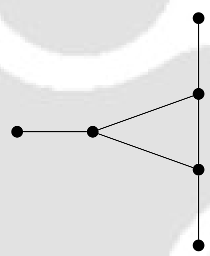


Figure 1.4: A non-comparability graph

The following theorem establishes the computational complexity of the recognition problem for comparability graphs.

Theorem 1.2.10 ([McConnell and Spinrad, 1999])**.** *Recognizing whether a graph is a comparability graph can be done in linear time.*

As stated in the following remark, posets and comparability graphs are related to each other.

Remark 1.2.11. For every poset P there is a comparability graph whose vertices are the elements of P and two distinct vertices are adjacent if and only if they are comparable in P. Conversely, if G is a comparability graph with a transitive orientation O, then G induces a poset, say $P_G = (V, \prec)$, such that $a \prec b$ in P_G if and only if $\overrightarrow{ab} \in O$.

Remark 1.2.12. A comparability graph can induce more than one poset, based on its transitive orientations.

However, we have the following important result in this context.

Theorem 1.2.13 ([Trotter et al., 1976]). *If P and Q are two posets induced by the same comparability graph, then* $\dim(P) = \dim(Q)$.

Further, posets with dimension at most two are characterized in terms of their underlying comparability graphs.

Theorem 1.2.14 ([Baker et al., 1972]). *A poset has dimension at most two if and only if its corresponding comparability graph is a permutation graph.*

The problem of determining the dimension of a given poset is computationally hard as per the following.

Theorem 1.2.15 ([Yannakakis, 1982]). *Let P be a poset. For $k \geq 3$, the problem of determining whether* $\dim(P) \leq k$ *is NP-complete.*

1.3 Word-Representable Graphs

This section is devoted to the theory of word-representable graphs, which studies the interplay between words and graphs. We refer to the monograph [Kitaev and Lozin, 2015] for a comprehensive introduction to this topic.

1.3.1 Words

In this subsection, we present some basic terminology and concepts related to words that are used throughout the thesis.

Definition 1.3.1 (Word)**.** Let X be a finite set (of letters). A *word* over X is a finite sequence of letters of X written by juxtaposing them. The *empty word* corresponding to the empty sequence is denoted by ε. The *length of a word* w, denoted by $|w|$, is the number of letters in the sequence w. Clearly, $|\varepsilon| = 0$.

Definition 1.3.2 (Subword and Factor)**.** A *subword* u of a word w is a subsequence of the sequence w and it is denoted by $u \ll w$. A subword u is a *factor* of w if $w = xuy$, for some words x and y, possibly empty words, denoted by $u \ll_f w$.

Example 1.3.3. Over the set $\{a, b, c\}$, $w = ababbcac$ is a word, $aabcc \ll w$ and $abbc \ll_f w$. Note that the subword $aabcc$ of w is not a factor of w.

Notation 1.3.4. Let w be a word over a set X.

1. For $Y \subseteq X$, the subword of w restricted to the letters of Y is denoted by $w|_Y$, i.e., the subword of w obtained by deleting all the letters belonging to $X \setminus Y$ from w. For example, if $w = aacbabca$, $w|_{\{a,b\}} = aababa$.

2. If $w = a_1a_2 \cdots a_k$ (for $a_i \in X$), the reversal of w is $w^R = a_k a_{k-1} \cdots a_1$.

3. For some positive integer k, x^k denotes the word $xx \cdots x$, where x is repeated k times.

Definition 1.3.5 (Alternation of letters)**.** Let a and b be two distinct letters that appear in a word w. We say a and b *alternate* in w if $w|_{\{a,b\}}$ is in one of the following forms: $(ab)^k$, $(ba)^k$, $a(ba)^k$ or $b(ab)^k$ for some positive integer $k \geq 1$. Otherwise, we say a and b do not alternate in w.

Example 1.3.6. Consider the word $w = aabcbacbc$. The letters b and c alternate in w as $w|_{\{b,c\}} = bcbcbc$, while the letters a and b do not alternate in w as $w|_{\{a,b\}} = aabbab$.

Definition 1.3.7 (k-uniform word)**.** A word w is called a *k-uniform word* if every letter appears exactly k times in w. A 1-uniform word w is a *permutation* on the set of letters of w.

Example 1.3.8. The word *aabcbacbc* is a 3-uniform word and the word *acbde* is a permutation on the set $\{a, b, c, d, e\}$.

Definition 1.3.9 (Cyclic shift)**.** A *cyclic shift* of a word $w = uv$ is a word vu, where u and v are two (possibly empty) words.

Example 1.3.10. The words *ccabbcddbb* and *abbcddbbcc* are cyclic shifts of the word *bbccabbcdd*.

1.3.2 Definition and Properties

We now introduce the concept of word-representable graphs and present some of their properties.

Definition 1.3.11 (Word-representable graph)**.** A graph $G = (V, E)$ is said to be a *word-representable graph* if there exists a word w over the vertex set V of G such that for all $a, b \in V$, a and b are adjacent in G if and only if a and b alternate in w. Such a word w is called a *word-representant* of G and we say w represents G. An example of a word-representable graph is given in Figure 1.5.

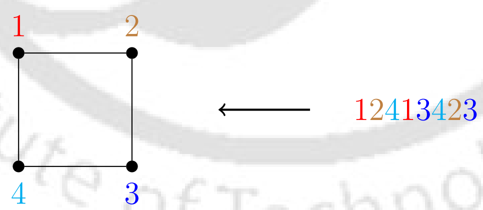


FIGURE 1.5: A word-representable graph

Example 1.3.12 ([Kitaev and Lozin, 2015])**.** Complete graphs, cycles, bipartite graphs are word-representable.

Remark 1.3.13 ([Kitaev and Lozin, 2015])**.** Word-representable graphs form a hereditary class, i.e., every induced subgraph of a word-representable graph is a word-representable graph.

Remark 1.3.14 ([Kitaev and Lozin, 2015])**.** Not all graphs are word-representable. For example, the wheel graph W_5 shown in Figure 1.6 is a non-word-representable graph.

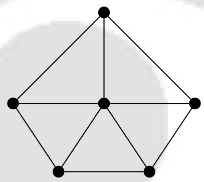


Figure 1.6: The wheel graph W_5

The following theorem establishes the computational complexity of the recognition problem for word-representable graphs.

Theorem 1.3.15 ([Kitaev and Lozin, 2015])**.** *It is an NP-complete problem to recognize whether a given graph is word-representable.*

Word-representable graphs are characterized using the notion of semi-transitive orientation of a graph in [Halldórsson et al., 2016].

Definition 1.3.16 (Semi-transitive orientation)**.** An orientation of a graph is said to be *semi-transitive* if it is acyclic, and for any directed path $a_1 \to a_2 \to \cdots \to a_k$ either there is no edge between a_1 and a_k, or there is a directed edge $\overrightarrow{a_i a_j}$, for every pair with $1 \leq i < j \leq k$. A graph is said to be a *semi-transitively orientable graph* if it admits a semi-transitive orientation. An example of a semi-transitively oriented graph is depicted in Figure 1.7.

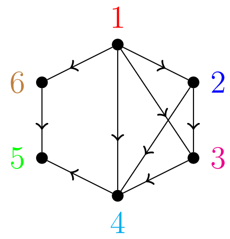


FIGURE 1.7: A semi-transitively oriented graph

Theorem 1.3.17 ([Halldórsson et al., 2016])**.** *A graph is word-representable if and only if it is a semi-transitively orientable graph.*

Remark 1.3.18. From Definition 1.3.16, it is evident that a transitive orientation of a graph is also a semi-transitive orientation. Thus, every comparability graph is a word-representable graph.

In the following, we summarize some of the properties of word-representants of a word-representable graph.

Proposition 1.3.19 ([Kitaev and Lozin, 2015])**.** *If a word w represents a graph G, then the reversal w^R of w also represents G.*

Definition 1.3.20 (k-word-representable graph)**.** A graph G is said to be *k-word-representable* if there exists a k-uniform word representing G.

Proposition 1.3.21 ([Kitaev and Pyatkin, 2008])**.** *A cyclic shift of a k-uniform word representing a graph also represents the graph.*

Theorem 1.3.22 ([Kitaev and Pyatkin, 2008])**.** *A graph is word-representable if and only if it is k-word-representable for some k.*

Proposition 1.3.23 ([Kitaev and Pyatkin, 2008])**.** *For every $l > k$, a k-word-representable graph is also l-word-representable. Thus, every word-representable graph has infinitely many word-representants.*

1.3.3 Representation Number

In this subsection, we define the concept of representation number of a word-representable graph and present some related results.

Definition 1.3.24 (Representation number)**.** The *representation number* of a word-representable graph G, denoted by $\mathcal{R}(G)$, is the smallest positive integer k such that G is k-word-representable.

Remark 1.3.25. For a graph G, $\mathcal{R}(G) = 1$ if and only if G is a complete graph.

The representation numbers of trees and cycles are established as per the following results.

Theorem 1.3.26 ([Kitaev and Lozin, 2015])**.** *For a tree T, $\mathcal{R}(T) \leq 2$.*

Theorem 1.3.27 ([Kitaev and Lozin, 2015])**.** *For $n \geq 4$, $\mathcal{R}(C_n) = 2$.*

Further, the class of graphs with representation number at most two is characterized as per the following result.

Theorem 1.3.28 ([Halldórsson et al., 2011])**.** *For a graph G, $\mathcal{R}(G) \leq 2$ if and only if G is a circle graph.*

Further, in [Fleischmann et al., 2024], the class of graphs with representation number k, for $k \geq 3$, is characterized through the notion of k-circle graphs. In general, the problem of determining the representation number of a word-representable graph is computationally hard.

Theorem 1.3.29 ([Halldórsson et al., 2011])**.** *It is an NP-complete problem to decide whether a given graph is k-word-representable, for $3 \leq k \leq \lceil \frac{n}{2} \rceil$.*

1.3.4 Minimum-Word-Representants

In addition to uniform word-representants, minimum-word-representants of word-representable graphs have also been studied in the literature.

Definition 1.3.30 (Minimum-word-representant)**.** A *minimum-word-representant* of a word-representable graph G is a word-representant which has the shortest length among all the word-representants of G. The length of a minimum-word-representant of G is denoted by $\ell(G)$.

Remark 1.3.31. A minimum-word-representant of a word-representable graph need not be uniform. Further, if G is a word-representable graph on n vertices, $\ell(G) \geq n$.

Moreover, we have the following result on the lower bound of $\ell(G)$.

Theorem 1.3.32 ([Srinivasan and Hariharasubramanian, 2024])**.** *Let G be a word-representable graph on n vertices with clique number κ. Then $\ell(G) \geq 2n - \kappa$.*

Remark 1.3.33. Since circle graphs are 2-word-representable (by Theorem 1.3.28), if G is a circle graph on n vertices, we have $\ell(G) \leq 2n$.

Moreover, we have the following results related to circle graphs.

Theorem 1.3.34 ([Srinivasan and Hariharasubramanian, 2024])**.** *Let n be the number of vertices of a graph. Then, we have the following:*

1. *There is no circle graph G with $\ell(G) = 2n$.*
2. *An edgeless graph G is the only circle graph with $\ell(G) = 2n - 1$.*
3. *For a triangle-free circle graph G containing at least one edge, $\ell(G) = 2n - 2$.*

1.3.5 Permutationally Representable Graphs

In this subsection, we present a special subclass of word-representable graphs, namely permutationally representable graphs and review some of their properties.

Definition 1.3.35 (Permutationally representable graph)**.** A graph $G = (V, E)$ is said to be *permutationally representable* if it is represented by a word w of the form $p_1 p_2 \cdots p_k$, for some positive integer k, where each p_i is a permutation on the vertex set V. In this case, since w is concatenation of k permutations, the graph G is called a *permutationally k-representable graph*. An example of a permutationally 2-representable graph is given in Figure 1.8.

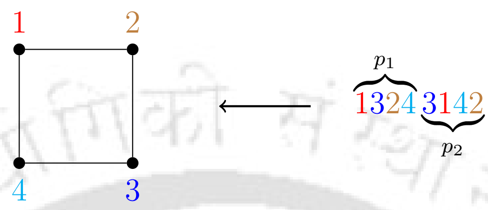


Figure 1.8: A permutationally representable graph

Permutationally representable graphs are characterized as per the following theorem.

Theorem 1.3.36 ([Kitaev and Seif, 2008])**.** *A graph is a permutationally representable graph if and only if it is a comparability graph.*

Throughout this thesis, we use the terms "permutationally representable graphs" and "comparability graphs" interchangeably, as they refer to the same class of graphs. Comparability graphs play a vital role in the theory of word-representable graphs.

Theorem 1.3.37 ([Kitaev and Pyatkin, 2008])**.** *If G is a word-representable graph, then the subgraph induced by $N_G(a)$ is a comparability graph, for every vertex a of G.*

Remark 1.3.38. The converse of Theorem 1.3.37 is not true in general. A counter example can be found in the monograph [Kitaev and Lozin, 2015, Chapter 3].

Moreover, we have the following result connecting the k-word-representability with the permutationally k-representability of graphs.

Theorem 1.3.39 ([Halldórsson et al., 2011]). *Suppose a graph G is obtained from a graph H by adding an all-adjacent vertex. Then G is k-word-representable if and only if H is permutationally k-representable.*

1.3.6 Permutation-Representation Number

We now present the notion of permutation-representation number of a comparability graph, which coincides with the dimension of its induced poset.

Definition 1.3.40 (Permutation-representation number). For a comparability graph G, the *permutation-representation number* (in short, *prn*) of G, denoted by $\mathcal{R}^p(G)$, is defined as the smallest positive integer k such that G is permutationally k-representable.

Remark 1.3.41. For a comparability graph G, it is evident that $\mathcal{R}(G) \leq \mathcal{R}^p(G)$.

Remark 1.3.42. For a graph G, $\mathcal{R}^p(G) = 1$ if and only if G is a complete graph.

Moreover, we have the following result relating the *prn* of a comparability graph with the dimension of its induced poset.

Theorem 1.3.43 ([Halldórsson et al., 2011; Mozhui and Krishna, 2025b]). *For a comparability graph G, $\mathcal{R}^p(G) = \dim(P_G)$, where P_G is an induced poset of G.*

Remark 1.3.44. Although a comparability graph can induce more than one poset, in view of Theorem 1.2.13, it is enough to consider any poset induced by a transitive orientation in Theorem 1.3.43.

Thus, in view of Theorem 1.2.14, we have the following characterization for the class of comparability graphs with *prn* at most two. However, no characterization is available for comparability graphs with *prn* k, where $k \geq 3$.

Theorem 1.3.45. *For a comparability graph G, $\mathcal{R}^p(G) \leq 2$ if and only if G is a permutation graph.*

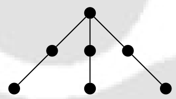
The permutation-representation numbers of even cycles, paths and trees are established as per the following result. However, in view of Theorem 1.2.15, it is evident that determining the *prn* of a comparability graph is computationally hard in general.

Theorem 1.3.46 ([Trotter, 1992])**.**

(i) $\mathcal{R}^p(C_4) = 2$ *and, for* $n \geq 3$, $\mathcal{R}^p(C_{2n}) = 3$.

(ii) *For* $n \geq 3$, $\mathcal{R}^p(P_n) = 2$.

(iii) *For a tree* T, $\mathcal{R}^p(T) \leq 3$. *Moreover,* $\mathcal{R}^p(T) = 3$ *if and only if the tree* S *shown in Figure 1.9 is isomorphic to an induced subgraph of* T.



S

Figure 1.9: A tree with *prn* 3

1.4 Split Decomposition

We now present the concept of split decomposition of a graph which is used as a fundamental tool in this thesis. The literature has several variants of split decomposition and its inverse operation, namely split recomposition. In this thesis, we use the version defined by Bouchet [1988]. For more details, one may refer to [Cornelsen and Di Stefano, 2009].

Definition 1.4.1 (Split)**.** A *split* of a connected graph $G = (V, E)$ is a bipartition $\{V_1, V_2\}$ of V (i.e., $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \varnothing$) satisfying the following: (i) $|V_1| \geq 2$ and $|V_2| \geq 2$, (ii) $N_G(V_1)$ is complete to $N_G(V_2)$ in G, i.e., each $a \in N_G(V_1)$ is adjacent to every $b \in N_G(V_2)$ in G. If a graph has no split, then it is said to be a *prime graph*; otherwise, it is said to be *decomposable*.

Remark 1.4.2. If $|V| \leq 3$, then G is a prime graph and G is either K_1, K_2, K_3 or the star S_2.

Definition 1.4.3 (Simple decomposition)**.** A *simple decomposition* of a connected graph G with a split $\{V_1, V_2\}$ is represented as a disjoint union of the induced subgraphs $G[V_1]$ and $G[V_2]$ along with an edge $e = \overline{v_1 v_2}$ (called *marked edge*), where v_1 and v_2 are two new vertices (called *marked vertices*) such that v_1 and v_2 are adjacent to every vertex of $N_G(V_2)$ and $N_G(V_1)$, respectively. By ignoring the edge e, we see two components, say G_1 and G_2, over the vertex sets $V_1 \cup \{v_1\}$ and $V_2 \cup \{v_2\}$, called the *split components*.

Definition 1.4.4 (Split decomposition)**.** A *split decomposition* of a connected graph G is defined recursively as follows.

1. The graph G is itself a split decomposition of G. In this case, G is considered as a split component of itself.

2. If $\mathcal{H}$ is a split decomposition of G with a split component H, the graph obtained from $\mathcal{H}$ by replacing H with a simple decomposition of H is also a split decomposition of G.

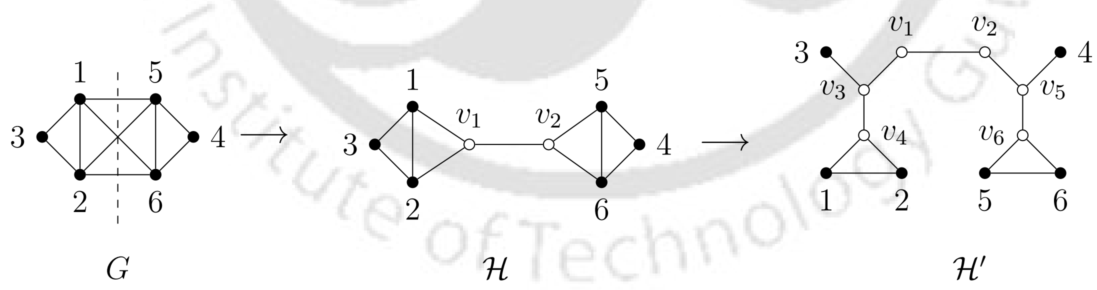


FIGURE 1.10: Split decompositions of a graph

Remark 1.4.5 (cf. [Cicerone and Di Stefano, 1999; Rao, 2008])**.** The split components of a split decomposition of G are connected and each split component of G is isomorphic to an induced subgraph of G.

Definition 1.4.6 (Minimal split decomposition)**.** A *minimal split decomposition* of a graph is a split decomposition whose split components can be cliques, stars and prime graphs such that the number of split components is minimized.

Example 1.4.7. Figure 1.10 shows various split decompositions of the graph G, among which $\mathcal{H}'$ is the minimal split decomposition of G. The marked vertices are indicated as unfilled vertices.

Theorem 1.4.8 ([Cunningham, 1982; Cunningham and Edmonds, 1980])**.** *Every connected graph has a unique minimal split decomposition.*

Theorem 1.4.9 ([Charbit et al., 2012; Dahlhaus, 1994])**.** *The minimal split decomposition of a graph can be computed in linear time.*

Remark 1.4.10 ([Di Stefano, 2012])**.** For a graph $G = (V, E)$, the number of split components of the minimal split decomposition of G is at most $|V| - 2$.

We now present the inverse operation of split decomposition, namely split recomposition of two graphs.

Definition 1.4.11 (Split recomposition)**.** Let $G = (V \cup \{m\}, E)$ and $G' = (V' \cup \{m'\}, E')$ be two connected graphs with disjoint vertex sets, where V and V' are nonempty. Then the *split recomposition* of G and G' with respect to the vertices m and m' (called the *marked vertices*), denoted by $G\ {}_{m}\circledast_{m'}\ G'$, is defined by $G\,{}_{m}\circledast_{m'} G' = (V'', E'')$, where the vertex set $V'' = V \cup V'$ and edge set E'' consist of the edges of $G[V]$, the edges of $G'[V']$ and $\{\overline{ab} \mid a \in N_G(m), b \in N_{G'}(m')\}$.

Example 1.4.12. An example of a split recomposition of the graphs G and G' with respect to the marked vertices m and m' is given in Figure 1.11.

Remark 1.4.13. In Definition 1.4.11, if $|V| = 1$ and $|V'| = 1$, the graph $G\,{}_{m}\circledast_{m'} G'$ is K_2. If $|V| \geq 2$ and $|V'| \geq 2$, the graph $G\ {}_{m}\circledast_{m'}\ G'$ contains a split, namely $\{V, V'\}$, and hence, can be decomposed. Further, if $|V| \geq 2$ and $|V'| = 1$, $G\,{}_{m}\circledast_{m'} G'$ is isomorphic to G.

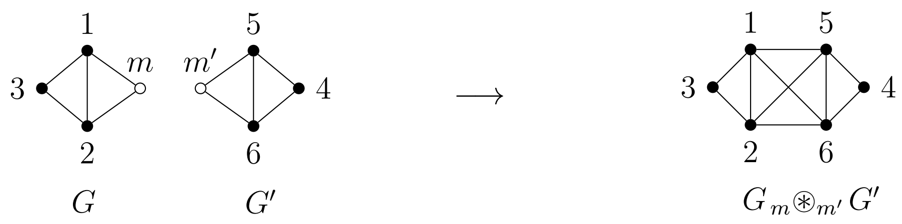


FIGURE 1.11: Split recomposition of two graphs

Remark 1.4.14. Let $G = (V \cup \{m\}, E)$ and $G' = (V' \cup \{m'\}, E')$ be two graphs. Then G and G' can be treated as induced subgraphs of $G\,_m\circledast_{m'}\,G'$. For instance, if $b \in N_G(m)$ and $b' \in N_{G'}(m')$, then $(G\,_m\circledast_{m'}\,G')[V \cup \{b'\}]$ is isomorphic to G, and G' is isomorphic to $(G\,_m\circledast_{m'}\,G')[V' \cup \{b\}]$.

Theorem 1.4.15 ([Bixby, 1984])**.** *The class of perfect graphs*[2] *is closed under split recomposition.*

1.4.1 Graph-Labeled Tree

The concept of split decomposition can be viewed in a more general combinatorial structure, namely graph-labeled tree, introduced by Gioan and Paul [2012]. It was shown in [Gioan and Paul, 2012] that split decomposition when reformulated in terms of graph-labeled tree, yields new combinatorial and algorithmic results as well as simpler proofs for previously known results. We now present the concept of graph-labeled tree and its relation with a split decomposition of a graph.

Definition 1.4.16 (Graph-labeled tree)**.** Let T be a tree and $\mathcal{F}$ be a set of vertex disjoint graphs. A *graph-labeled tree*, denoted by $T_{\mathcal{F}}$, is the graph obtained from T by labeling (in fact, by inserting) every internal vertex α of degree k (≥ 2) by a graph $G_\alpha \in \mathcal{F}$ on k vertices such that there is a bijection from the edges of T incident to α to the vertices of G_α (and by replacing the end point α of a tree edge with the corresponding vertex of G_α).

[2] A graph G is called a perfect graph if for every induced subgraph H of G, the chromatic number of H is same as the clique number of H.

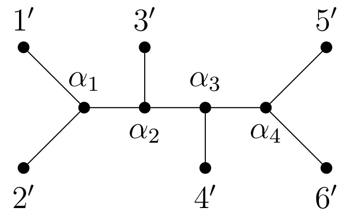


Figure 1.12: A tree

Remark 1.4.17. The set of pendant (degree one) vertices of $T_{\mathcal{F}}$ is precisely the set of leaves of T. Given a graph-labeled tree $T_{\mathcal{F}}$ and the set $\mathcal{F}$, we can determine the underlying tree structure T. In the depiction of a graph-labeled tree, we encircle the graphs G_α in $T_{\mathcal{F}}$ that are replacing the internal vertices α of T. For example, for the tree given in Figure 1.12, a graph-labeled tree is depicted in Figure 1.13(b).

Definition 1.4.18 (Maximal alternated path)**.** The edges of $T_{\mathcal{F}}$ in the encircled portions are called *$\mathcal{F}$-edges* and the remaining edges are called *T-edges*, as they correspond to the tree edges of T. A path in $T_{\mathcal{F}}$ is called an *alternated path* if it alternates between T-edges and $\mathcal{F}$-edges. A *maximal alternated path* is an alternated path that cannot be extended by adding more edges while maintaining the alternatedness.

Definition 1.4.19 (Accessibility graph)**.** The *accessibility graph* of a graph-labeled tree $T_{\mathcal{F}}$ is the graph, denoted by $T_{\mathcal{F}}^{A}$, in which the vertex set is the set of pendant vertices in $T_{\mathcal{F}}$ and any two vertices a and b in $T_{\mathcal{F}}^{A}$ are adjacent if and only if there is an alternated path between a and b in $T_{\mathcal{F}}$.

We now relate split decomposition with the concept of graph-labeled tree and define a split-decomposition tree of a graph.

Definition 1.4.20 (Split-decomposition tree)**.** Let $\mathcal{H}$ be a split decomposition of a graph G. A *split-decomposition tree* of G is a graph-labeled tree $T_{\mathcal{F}}$ obtained by extending $\mathcal{H}$ by adding one new pendant vertex a' for each non-marked vertex a of $\mathcal{H}$

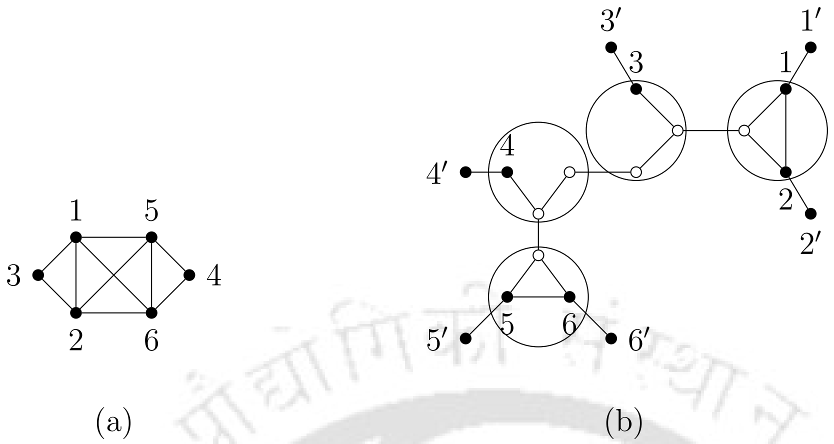


FIGURE 1.13: (a) The graph G and (b) its minimal split-decomposition tree

such that a and a' are adjacent, and considering $\mathcal{F}$ as the set of the split components of $\mathcal{H}$. If the split components of a split-decomposition tree are cliques (called *clique components*) and stars (called *star components*), then it is called a *clique-star tree.*

We rewrite the uniqueness result by Cunningham (Theorem 1.4.8) in terms of graph-labeled trees in the following theorem.

Theorem 1.4.21 ([Bahrani and Lumbroso, 2018; Cunningham, 1982; Gioan and Paul, 2012]). *For every connected graph G, there exists a unique split-decomposition tree $T_{\mathcal{F}}$ of G such that*

(i) *the accessibility graph $T_{\mathcal{F}}^{A}$ is isomorphic to the graph G,*

(ii) *every split component is a clique, a star, or a prime graph,*

(iii) *every split component has at least three vertices,*

(iv) *there is no marked edge with end points in two clique components, and*

(v) *there is no marked edge between the center of a star component and a leaf of another star component.*

Definition 1.4.22 (Minimal split-decomposition tree)**.** For a connected graph G, the unique split-decomposition tree described in Theorem 1.4.21 is called the *minimal split-decomposition tree* of G. For example, refer to Figure 1.13 for the minimal split-decomposition tree of the graph G.

1.5 Modular Decomposition

In this section, we present the preliminaries of modular decomposition. For more details, one may refer to the surveys [Habib and Paul, 2010; Möhring and Radermacher, 1984], while it is worth noting that [Gallai, 1967] is the seminal paper on modular decomposition.

Definition 1.5.1 (Module)**.** Let $G = (V, E)$ be a graph. A set of vertices $A \subseteq V$ is called a *module* of G if for any $b \in V \setminus A$, either $N_G(b) \cap A = \varnothing$ or $A \subseteq N_G(b)$. Note that the set $\{2, 3\}$ is a module of the graph G depicted in Figure 1.14(a).

Definition 1.5.2 (M-prime and M-decomposable graph)**.** A graph is said to be an *M-prime graph* if it contains only *trivial modules*, namely the singletons and the whole set V. Any module other than the trivial ones is called a *non-trivial module*. If a graph contains at least one non-trivial module, the graph is said to be *M-decomposable*.

Definition 1.5.3 (Modular partition)**.** A *modular partition* of a graph $G = (V, E)$ is a partition of V into modules of G.

Definition 1.5.4 (Quotient graph)**.** For some positive integer k, let $\mathcal{P} = \{A_1, \ldots, A_k\}$ be a modular partition of $G = (V, E)$, i.e., A_i is a module of G for each $1 \leq i \leq k$. Choose exactly one vertex a_i from each A_i and let $V' = \{a_1, a_2, \ldots, a_k\}$. The *quotient graph* associated to $\mathcal{P}$ of G, denoted by $G/\mathcal{P}$, is the graph (V', E'), where $\overline{a_i a_j} \in E'$ if and only if $\overline{ab} \in E$, for all $a \in A_i$ and for all $b \in A_j$.

Example 1.5.5. The graph G depicted in Figure 1.14(a) is M-decomposable and the partition $\mathcal{P} = \{\{1\}, \{2, 3, 4\}, \{5\}, \{6, 7\}, \{8, 9, 10, 11\}\}$ of the vertex set of G is a modular partition of G. Further, the graph shown in Figure 1.14(b) is the quotient graph $G/\mathcal{P}$ associated to the modular partition $\mathcal{P}$ of G. Observe that $G/\mathcal{P}$ is an M-prime graph.

Definition 1.5.6 (Modular decomposition)**.** A modular partition $\mathcal{P}$ of G and its quotient graph $G/\mathcal{P}$ constitute a *modular decomposition* of G.

Remark 1.5.7. For some positive integer k, let $\mathcal{P} = \{A_1, A_2, \ldots, A_k\}$ be a modular partition of a graph G. It is evident that $G/\mathcal{P}$ is isomorphic to an induced subgraph of G. Moreover, the original graph G can be reconstructed from $G/\mathcal{P}$ by replacing each vertex of $G/\mathcal{P}$ with the corresponding induced subgraph $G[A_i]$, for $1 \le i \le k$.

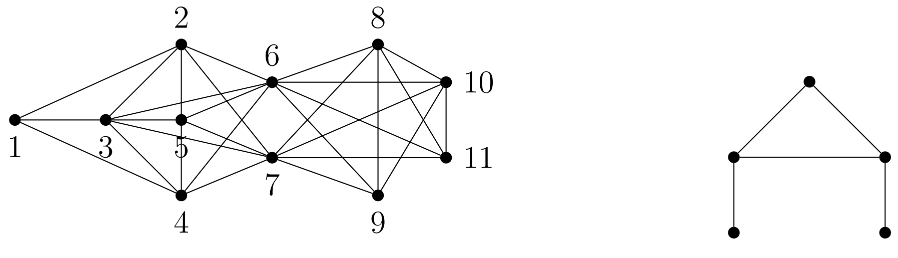
Definition 1.5.8 (Strong module and Maximal module)**.** A module A in a graph $G = (V, E)$ is said to be *strong* if A does not intersect with any other module of G or whenever $A \cap A' \neq \varnothing$ for some module A' of G then one is contained in the other. Note that the whole set V and the singletons are strong modules of G. A module A is called *maximal* if there is no module A' of G such that $A \subset A' \subset V$.

Definition 1.5.9 (Maximal modular partition)**.** A modular partition of a graph containing only maximal strong modules is called a *maximal modular partition.*

Theorem 1.5.10 ([Habib and Paul, 2010])**.** *Every graph has a unique maximal modular partition.*

Example 1.5.11. The maximal modular partition of the graph G, depicted in Figure 1.14(a), is the partition $\mathcal{P}$, as described in Example 1.5.5. However, the partition $\mathcal{Q} = \{\{1\}, \{2, 4\}, \{3\}, \{5\}, \{6, 7\}, \{8, 9, 11\}, \{10\}\}$ of the vertex set of G is a modular partition but not a maximal one.

Remark 1.5.12 ([McConnell and Spinrad, 1999])**.** All modules of a graph G can be obtained through the notion called modular decomposition tree of G, whose nodes



(a) (b)

Figure 1.14: A graph G in (a) and its quotient graph $G/\mathcal{P}$ in (b)

are the strong modules of G. The modular decomposition tree of G has V as the root and the children of a node A in the tree are the parts of the maximal modular partition of the induced subgraph $G[A]$.

Further, we have the following result on the computational complexity of determining the modular decomposition tree of a graph.

Theorem 1.5.13 ([McConnell and Spinrad, 1999]). *The modular decomposition tree of a graph G and modular decomposition of G can be computed in $O(n+m)$ time, where n and m are the number of vertices and the number of edges of G, respectively.*

We now present a characterization of comparability graphs with respect to a modular decomposition.

Theorem 1.5.14 ([Golumbic, 2004; Möhring, 1985]). *For some positive integer k, let $\mathcal{P} = \{A_1, A_2, \ldots, A_k\}$ be a modular partition of an M-decomposable graph G. Then G is a comparability graph if and only if $G/\mathcal{P}$ and each of the induced subgraphs $G[A_i]$ are comparability graphs.*

1.6 Graph Classes

In this section, we present the graph classes, namely distance-hereditary graphs, treelike comparability graphs, and well-partitioned chordal graphs, and recall their

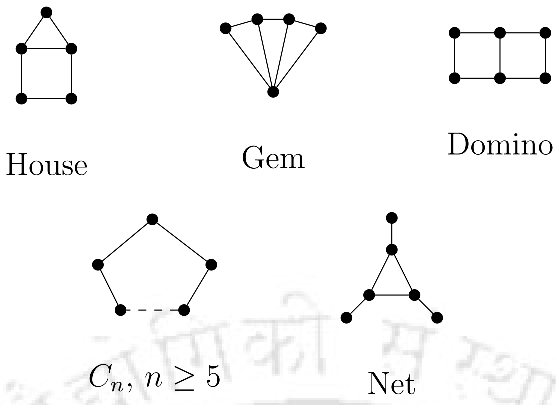


FIGURE 1.15: Forbidden induced subgraphs for distance-hereditary comparability graphs

properties that are used in this thesis.

1.6.1 Distance-Hereditary Graphs

Definition 1.6.1 (Distance-hereditary graph)**.** A graph G is called a *distance-hereditary graph* if the distance between any two vertices in any connected induced subgraph of G is the same as the distance in G.

The literature has various characterizations for distance-hereditary graphs (cf. [Bandelt and Mulder, 1986] and [Bouchet, 1988]). In the following, we present the characterization obtained using split decomposition.

Theorem 1.6.2 ([Bouchet, 1988])**.** *A graph G is a distance-hereditary graph if and only if the minimal split-decomposition tree of G is a clique-star tree.*

Theorem 1.6.3 ([Di Stefano, 2012])**.** *A graph G is a distance-hereditary comparability graph if and only if the house, domino, gem, net and cycles of length at least five (depicted in Figure 1.15) are not induced subgraphs of G.*

Further, certain subclasses of distance-hereditary graphs were characterized in terms of their minimal split-decomposition trees in [Bahrani and Lumbroso, 2018].

In particular, C_4-free distance-hereditary graphs were characterized through a notion called center-center path, which is defined below.

Definition 1.6.4 (Center-center path)**.** Let G be a distance-hereditary graph with the minimal split-decomposition tree $T_{\mathcal{F}}$. A *center-center path* in $T_{\mathcal{F}}$ is an alternated path whose endpoints are the centers of star components, and it does not contain any edge from either of star components.

Theorem 1.6.5 ([Bahrani and Lumbroso, 2018])**.** *Let G be a distance-hereditary graph with the minimal split-decomposition tree $T_{\mathcal{F}}$. Then G is C_4-free if and only if $T_{\mathcal{F}}$ does not have any center-center path.*

We now introduce one of the well-studied subclasses of distance-hereditary graphs, namely cographs.

Definition 1.6.6 (Cograph)**.** A *cograph* is defined recursively as follows:

(i) A graph on a single vertex is a cograph.

(ii) If $G_1, \ldots, G_k$ are cographs, then so is their union.

(iii) If G is a cograph, then so is its complement.

Cographs are characterized through various parameters. In the following, we present the forbidden induced subgraph characterization of cographs.

Theorem 1.6.7 ([Corneil et al., 1981])**.** *A graph G is a cograph if and only if G is P_4-free.*

In this thesis, we use the terms "cographs" and "P_4-free graphs" interchangeably.

Theorem 1.6.8 ([Bandelt and Mulder, 1986])**.** *P_4-free graphs are distance-hereditary graphs.*

Theorem 1.6.9 ([Bose et al., 1998])**.** *P_4-free graphs are permutation graphs.*

In Chapter 5, we characterize the class of P_4-free graphs in terms of their minimal split-decomposition trees (see Corollary 5.1.5). In this connection, we need the following lemmas on maximal alternated paths in the minimal split-decomposition tree of a distance-hereditary graph.

Lemma 1.6.10 ([Bahrani and Lumbroso, 2018])**.** *Let G be a distance-hereditary graph with the minimal split-decomposition tree $T_{\mathcal{F}}$. Any maximal alternated path starting from any split component of $T_{\mathcal{F}}$ ends in a pendant vertex of $T_{\mathcal{F}}$.*

Lemma 1.6.11 ([Bahrani and Lumbroso, 2018])**.** *Let G be a distance-hereditary graph with the minimal split-decomposition tree $T_{\mathcal{F}}$. If α is an internal vertex of the underlying tree T of $T_{\mathcal{F}}$, then any two maximal alternated paths P and Q that start at distinct vertices of G_α but contain no interior edges from G_α end at distinct pendant vertices of $T_{\mathcal{F}}$.*

1.6.2 Treelike Comparability Graphs

In this subsection, we present the concept of treelike comparability graphs. We also define certain subclasses of treelike comparability graphs, namely arborescences, double-arborescences, path of double-arborescences, and state some relevant results.

Definition 1.6.12 (Treelike comparability graph)**.** A graph is called a *treelike comparability graph* if it admits a transitive orientation such that the underlying graph of its transitive reduction is a tree. In this case, the corresponding orientation is called a *treelike orientation.*

Example 1.6.13. For the graph $G = (V, E)$ shown in Figure 1.16, O is a treelike orientation of G, as the underlying graph of its transitive reduction $\tilde{G}$ is a tree.

Remark 1.6.14. Not all comparability graphs admit treelike orientation, e.g., C_6 does not have any treelike orientation.

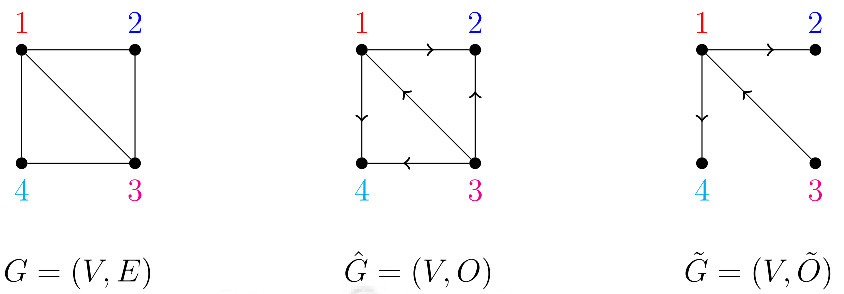


Figure 1.16: A treelike comparability graph with its treelike orientation and reduction

Treelike comparability graphs were characterized using split decomposition in [Cornelsen and Di Stefano, 2009, Theorem 4]. In view of the algorithm provided in [Cornelsen and Di Stefano, 2009, Theorem 5], we rewrite the characterization of treelike comparability graphs in the setting of graph-labeled trees.

Theorem 1.6.15 ([Cornelsen and Di Stefano, 2009])**.** *Let G be a graph and $T_{\mathcal{F}}$ be the minimal split-decomposition tree of G. Then G is a treelike comparability graph if and only if*

(i) *$T_{\mathcal{F}}$ is a clique-star tree,*

(ii) *each clique component has at most two marked vertices, and*

(iii) *there is no marked edge between the centers of two star components.*

Remark 1.6.16. In view of Theorems 1.6.2 and 1.6.15, it is evident that every treelike comparability graph is a distance-hereditary graph.

Theorem 1.6.17 ([Cornelsen and Di Stefano, 2009])**.** *A treelike orientation of a comparability graph, if exists, is unique up to isomorphism and reversing the whole orientation.*

Theorem 1.6.18 ([Cornelsen and Di Stefano, 2009])**.** *It can be tested in linear time whether a graph is a treelike comparability graph. Moreover, for a treelike comparability graph G, the treelike orientation of G as well as the transitive reduction can be found in linear time.*

Definition 1.6.19 (Treelike permutation graph)**.** A graph is called a *treelike permutation graph* if it is a treelike comparability graph as well as a permutation graph.

In [Cornelsen and Di Stefano, 2009], a treelike permutation graph is characterized as a path of double-arborescences. The concepts of double-arborescences and path of double-arborescences are defined below.

Definition 1.6.20 (Arborescence and Double-arborescence)**.** A treelike comparability graph G with an all-adjacent vertex is called a *double-arborescence*, i.e., there is a vertex r such that $V = \{r\} \cup N_G(r)$. We consider r as the root of G. In addition, under the treelike orientation, if the root r is a source (or a sink), i.e., the indegree (respectively, outdegree) of r is zero, then G is called an *arborescence*. The treelike orientation of a (double-)arborescence is called the *(double-)arborescence orientation*.

Remark 1.6.21. Note that every arborescence is a double-arborescence but not conversely.

Definition 1.6.22 (Strict-double-arborescence)**.** A double-arborescence is called a *strict-double-arborescence* if it is not an arborescence.

Example 1.6.23. Some examples of double-arborescences in terms of their transitive reductions are given in Figure 1.17. Note that the graph shown in Figure 1.17(b) is, in fact, a strict-double-arborescence.

(a) Arborescences (b) Double-arborescence

FIGURE 1.17: Examples of double-arborescences in terms of transitive reductions

The arborescences were first studied by Wolk [1962, 1965] and they were characterized as per the following result.

Theorem 1.6.24 ([Wolk, 1962, 1965])**.** *A graph is an arborescence if and only if it is* (C_4, P_4)*-free.*

Definition 1.6.25 (Path of double-arborescences)**.** A treelike comparability graph G is called a *path of* k *double-arborescences* (or simply, a *path of double-arborescences*) if G (precisely) consists of k double-arborescences and a path (ignoring the directions) connecting their roots; such a path is called a *root-path* of G.

Example 1.6.26. An example of path of 4 double-arborescences in terms of its transitive reduction is given in Figure 1.18.

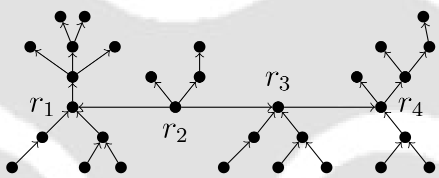


Figure 1.18: A path of double-arborescences

Theorem 1.6.27 ([Cornelsen and Di Stefano, 2009])**.** *A graph is a treelike permutation graph if and only if it is a path of double-arborescences.*

Theorem 1.6.28 ([Cornelsen and Di Stefano, 2009])**.** *Let* G *be a treelike permutation graph. The minimum* k *for which* G *is a path of* k *double-arborescences can be determined in linear time.*

1.6.3 Well-Partitioned Chordal Graphs

In this section, we present the concept of well-partitioned chordal graphs, a subclass of chordal graphs – graphs in which every cycle on at least four vertices has a

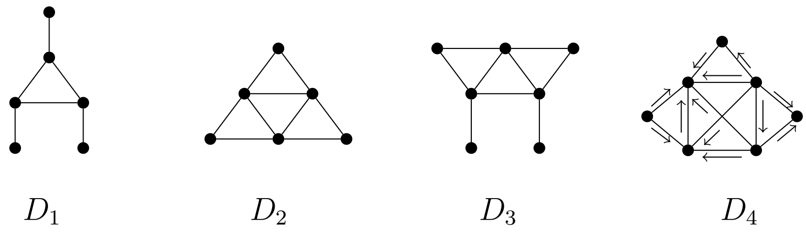


FIGURE 1.19: The set of graphs $\mathcal{D}$

chord. Well-partitioned chordal graphs were introduced in [Ahn et al., 2022] as a generalization of split graphs. We first introduce the concept of split graphs and present some results related to them.

Definition 1.6.29 (Split graph)**.** A graph G is called a *split graph* if the vertex set of G can be partitioned into a clique and an independent set. Refer to Figure 1.19 for some examples of split graphs.

The following theorems provide us the forbidden induced subgraph characterizations of comparability graphs, permutation graphs and circle graphs, within split graphs.

Theorem 1.6.30 ([Golumbic, 2004])**.** *Let G be a split graph. Then, G is a comparability graph if and only if G does not contain any induced subgraph isomorphic to D_1, D_2 or D_3 given in Figure 1.19.*

Theorem 1.6.31 ([Bonomo-Braberman et al., 2022])**.** *Let G be a split graph. Then, G is a permutation graph if and only if G is $\mathcal{D}$-free, where $\mathcal{D}$ is the set of graphs given in Figure 1.19.*

Theorem 1.6.32 ([Bonomo-Braberman et al., 2022])**.** *Let G be a split graph. Then, G is a circle graph if and only if G is a $\mathcal{C}$-free graph, where $\mathcal{C}$ is the set of graphs given in Figure 1.20. Further, each graph belonging to the set $\mathcal{C}$ is a minimally non-circle[3] graph.*

[3] A graph G is minimally non-circle if G is not a circle graph but every proper induced subgraph of G is a circle graph.

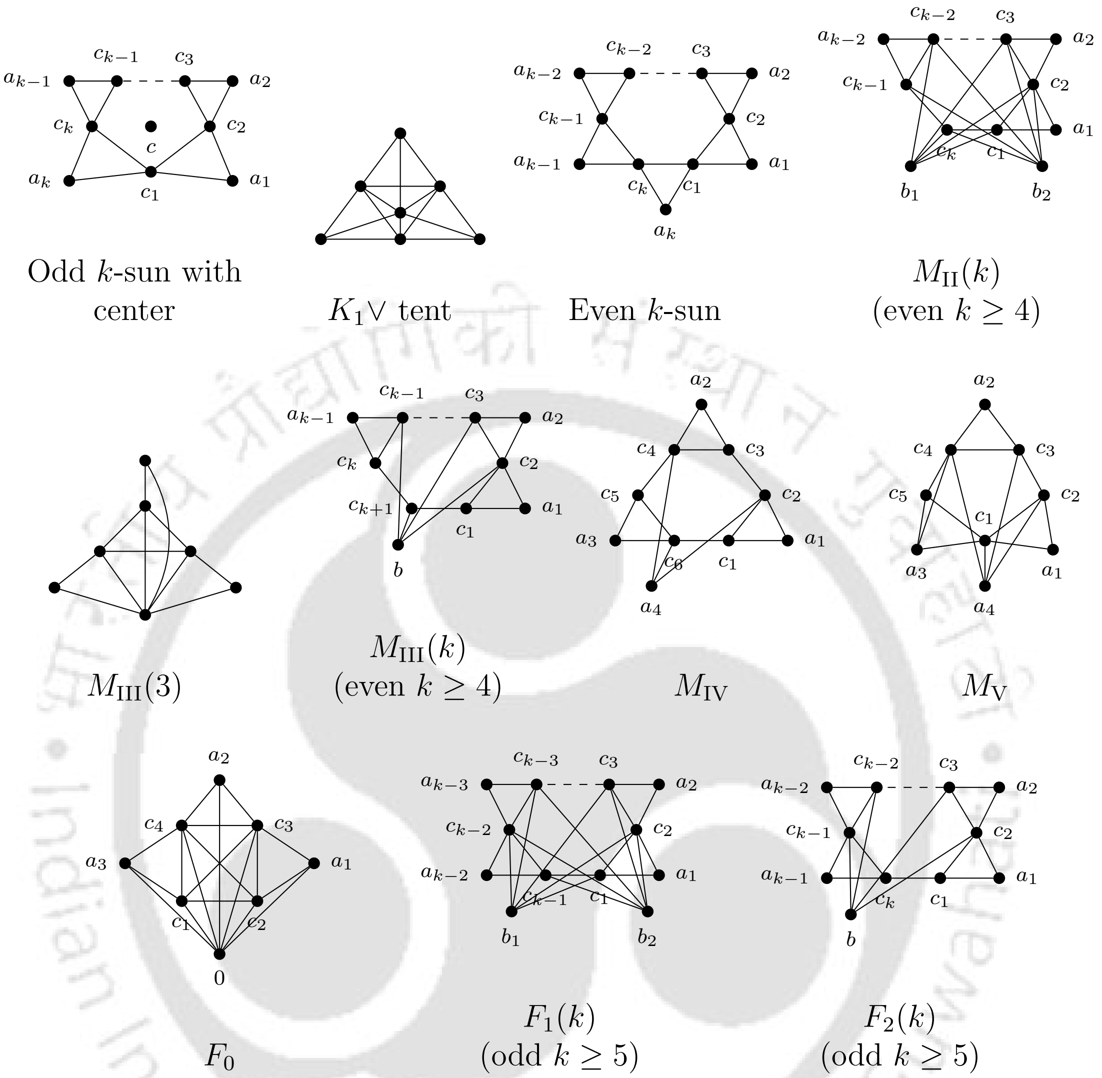


Figure 1.20: The set $\mathcal{C}$ of split graphs. For convenience, except in F_0, the edges of the clique over the vertices $c, c_1, c_2, \ldots$ (wherever labeled) are not drawn completely in each graph.

In the following, we present some results on the word-representability of split graphs.

Remark 1.6.33 ([Kitaev et al., 2021]). Not all split graphs are word-representable, e.g., for $k \geq 3$, an odd k-sun with center depicted in Figure 1.20 is not word-representable.

Notation 1.6.34. In what follows, the graph $(I \cup C, E)$ denotes a split graph, where C is a clique and I is an independent set in the graph. Further, for a split graph $(I \cup C, E)$, we consider C as inclusion wise maximal, i.e., no vertex of I is adjacent to all vertices of C. Further, for any two integers $m \leq n$, we denote the set of integers $\{m, m+1, \ldots, n\}$ by $[m, n]$.

Theorem 1.6.35 ([Kitaev et al., 2021; Kitaev and Pyatkin, 2024])**.** *Suppose $G = (I \cup C, E)$ is a split graph. Then, G is word-representable if and only if the vertices of C can be labeled from 1 to $k = |C|$ in such a way that for each $a, b \in I$ the following hold.*

(i) *Either $N(a) = [1, m] \cup [n, k]$, for $m < n$, or $N(a) = [l, r]$, for $l \leq r$.*

(ii) *If $N(a) = [1, m] \cup [n, k]$ and $N(b) = [l, r]$, for $m < n$ and $l \leq r$, then $l > m$ or $r < n$.*

(iii) *If $N(a) = [1, m] \cup [n, k]$ and $N(b) = [1, m'] \cup [n', k]$, for $m < n$ and $m' < n'$, then $m' < n$ and $m < n'$.*

Theorem 1.6.36 ([Kitaev and Pyatkin, 2024])**.** *The problem of recognizing word-representability of split graphs can be done in polynomial time.*

Let $G = (C \cup I, E)$ be a split graph. Consider the partition $\{C\} \cup \{\{a\} \mid a \in I\}$ of the vertex set of G in which each part is a clique. The partition forms a structure of a star with C as the center and the cliques of size one as its leaves.

Well-partitioned chordal graphs are defined by generalizing the aforementioned structure of split graphs. First, it allows the parts of a partition of the vertex set to be arranged in a tree structure instead of a star. Second, it permits a clique of arbitrary size in each part, satisfying certain conditions. In the following, we present the formal definition.

Definition 1.6.37 (Well-partitioned chordal graph)**.** A connected graph $G = (V, E)$ is said to be a *well-partitioned chordal graph* (in short, WPC graph) if there exist a partition $\mathcal{P}$ of V and a tree $\mathcal{T} = (\mathcal{P}, E')$ such that the following hold:

1. Each part $A \in \mathcal{P}$ is a clique in G.
2. For $A, B \in \mathcal{P}$, if $\overline{AB} \in E'$, there exist subsets $A' \subseteq A$ and $B' \subseteq B$ such that the edge set of the bipartite graph $G[A, B]$ is $\{\overline{ab} \mid a \in A', b \in B'\}$.
3. For $A, B \in \mathcal{P}$, if $\overline{AB} \notin E'$, the edge set of the bipartite graph $G[A, B]$ is empty.

Notation 1.6.38. The tree $\mathcal{T}$ in Definition 1.6.37 is called a *partition tree* of G, and the elements of $\mathcal{P}$ are called its *bags*. A bag B in a partition tree $\mathcal{T}$ is called a *leaf bag* if the degree of B in $\mathcal{T}$ is one; otherwise it is called an *internal bag*.

Example 1.6.39. An example of a WPC graph is depicted in Figure 1.21, where the bags of the partition of the vertex set are marked within dashed circles.

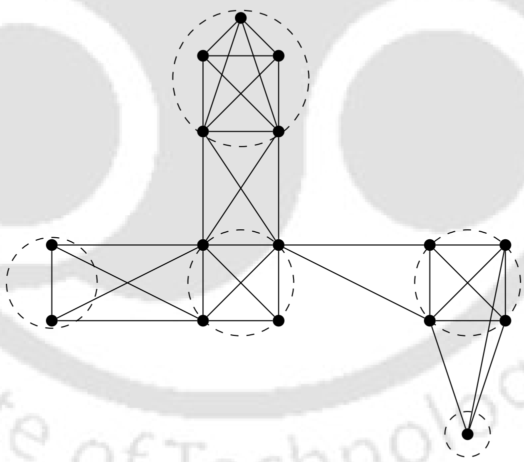
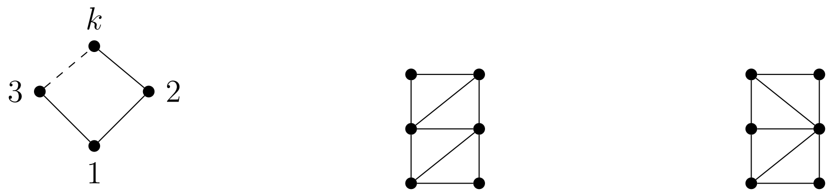


FIGURE 1.21: A WPC graph

Remark 1.6.40. A connected WPC graph can have more than one partition tree.

The class of WPC graphs includes the class of split graphs as stated in the following remark.



$C_k, k \geq 4$ $\qquad$ O_1 $\qquad$ O_2

FIGURE 1.22: Some graphs from the set $\mathbb{O}$

Remark 1.6.41. A connected graph G is a split graph if and only if G is a WPC graph and there exists a partition tree $\mathcal{T}$ associated to G such that $\mathcal{T}$ is a star centered at a clique and each leaf bag is a clique of size one.

Definition 1.6.42 (Boundary)**.** Let $G = (V, E)$ be a WPC graph with a partition tree $\mathcal{T} = (\mathcal{P}, E')$. For $A, B \in \mathcal{P}$ with $\overline{AB} \in E'$, the *boundary of A with respect to B* is defined as $\{a \in A \mid N_G(a) \cap B \neq \varnothing\}$ and is denoted by $bd(A, B)$.

In view of Definition 1.6.37, we have the following remarks.

Remark 1.6.43. If two bags A and B are adjacent in $\mathcal{T}$, then $bd(A, B)$ is complete to $bd(B, A)$ in G.

Remark 1.6.44. Let $\overline{AB} \in E'$ for some bags A and B of $\mathcal{P}$. Consider the graph $\mathcal{T} \setminus \overline{AB}$ obtained by deleting the edge $\overline{AB}$ from $\mathcal{T}$ and let $\mathcal{T}^A$ and $\mathcal{T}^B$ be the two components of $\mathcal{T} \setminus \overline{AB}$ containing A and B, respectively. Then $V = V^A \cup V^B$, where V^A and V^B consist of the vertices appeared in $\mathcal{T}^A$ and $\mathcal{T}^B$, respectively. Moreover, $V^A \cap V^B = \varnothing$, and $N_G(V^A) = bd(B, A)$ and $N_G(V^B) = bd(A, B)$.

Proposition 1.6.45 ([Ahn et al., 2022])**.** *Every induced subgraph of a WPC graph is also a WPC graph.*

The following theorem establishes that the set of graphs $\mathbb{O}$, given in [Ahn et al., 2022, Figure 2], is the set of minimal forbidden induced subgraphs for WPC graphs.

In Figure 1.22, we present some of the graphs from the set $\mathbb{O}$ that are used in our work.

Theorem 1.6.46 ([Ahn et al., 2022])**.** *A graph G is a WPC graph if and only if G is a $\mathbb{O}$-free graph.*

2

Split Decomposition

In this chapter, we characterize word-representable graphs with respect to split decomposition. Consequently, we determine the representation number of a word-representable graph in terms of the representation numbers of its split components. Accordingly, we show that parity graphs are word-representable. Further, we obtain a characteristic property by which the recomposition of two comparability graphs is a comparability graph. Consequently, we establish the permutation-representation number (prn) of the resulting comparability graph. We also introduce a subclass of comparability graphs, called prn-irreducible graphs. We provide a criterion such that the split recomposition of two prn-irreducible graphs is a comparability graph and determine the prn of the resulting graph.

2.1 Word-Representable Graphs

First we observe in Remark 2.1.1 that word-representability is preserved by split decomposition. Further, we show that split recomposition also preserves the word-representability of graphs. Consequently, we observe that the class of parity graphs is word-representable. We also relate the representation number of the graph obtained by the recomposition with its components.

Remark 2.1.1. If a graph G is word-representable, then its split components are word-representable, as they are isomorphic to certain induced subgraphs of G (by Remark 1.4.5) and every induced subgraph of a word-representable graph is word-representable (by Remark 1.3.13).

Theorem 2.1.2. *If $G = (V \cup \{m\}, E)$ and $G' = (V' \cup \{m'\}, E')$ are two k-word-representable graphs, then so is the split recomposition $G\,_m\circledast_{m'}\,G'$.*

Proof. Let w and w' be two k-uniform words representing the graphs G and G', respectively. In view of Proposition 1.3.21, note that a cyclic shift of a k-uniform word representing a graph also represents the graph. Accordingly, assume $w = w_1mw_2m\cdots w_km$ and $w' = m'w'_1m'w'_2\cdots m'w'_k$. Note that $w|_V = w_1w_2\cdots w_k$ and $w'|_{V'} = w'_1w'_2\cdots w'_k$. Moreover, $w|_V$ and $w'|_{V'}$ are k-uniform words. We will show that the k-uniform word $u = w_1w'_1w_2w'_2\cdots w_kw'_k$ represents the graph $G\,_m\circledast_{m'}\,G'$.

Note that the induced subgraphs $G[V]$ and $G'[V']$ are induced subgraphs of $G\,_m\circledast_{m'}\,G'$. Further, the k-uniform words $w|_V$ and $w'|_{V'}$ represent $G[V]$ and $G'[V']$, respectively. Accordingly, since $u|_V = w|_V$ and $u|_{V'} = w'|_{V'}$, any two vertices of $G[V]$ or of $G'[V']$ are adjacent if and only if they alternate in the word u.

For $a \in V$ and $b \in V'$, note that a and b are adjacent in $G\,_m\circledast_{m'}\,G'$ if and only if $a \in N_G(m)$ and $b \in N_{G'}(m')$. Accordingly, we consider the following cases to complete the proof.

Let $a \in N_G(m)$ and $b \in N_{G'}(m')$. Note that $w|_{\{a,m\}} = (am)^k$ (i.e., the word obtained by concatenating am for k times) and $w'|_{\{b,m'\}} = (m'b)^k$. Accordingly, for all $1 \leq i \leq k$, w_i and w'_i contain exactly one copy of a and b, respectively. Hence, $u|_{\{a,b\}} = (ab)^k$ so that a and b alternate in u.

Let $a \notin N_G(m)$. Since the k-uniform word w represents G, the vertices a and m do not alternate in w. Hence, aa is a substring of w_t, for some t. For any $b \in V'$, note that b does not appear in w_t. According to the construction of u, the subword $u|_{\{a,b\}}$ has aa as a factor. Hence, a and b do not alternate in u. Similarly, when $b \notin N_{G'}(m')$, we can observe that b does not alternate with any $a \in V$.

Hence, the k-uniform word u represents $G\,{}_m\circledast_{m'} G'$. □

Corollary 2.1.3. *If $G = (V \cup \{m\}, E)$ and $G' = (V' \cup \{m'\}, E')$ are two word-representable graphs such that $\mathcal{R}(G) = k$ and $\mathcal{R}(G') = k'$, then $\mathcal{R}(G\,{}_m\circledast_{m'} G') = \max\{k, k'\}$.*

Proof. Let $t = \max\{k, k'\}$. Using Proposition 1.3.23, both G and G' are t-word-representable. Consequently, by Theorem 2.1.2, $G\,{}_m\circledast_{m'} G'$ is t-word-representable so that $\mathcal{R}(G\,{}_m\circledast_{m'} G') \leq t$. But, by Remark 1.4.14, since G and G' are isomorphic to certain induced subgraphs of $G\,{}_m\circledast_{m'} G'$, we have $\mathcal{R}(G\,{}_m\circledast_{m'} G') \geq t$. Hence, $\mathcal{R}(G\,{}_m\circledast_{m'} G') = \max\{k, k'\}$. □

We now summarize the characterization of word-representability of a graph with respect to its split decomposition in the following theorem.

Theorem 2.1.4. *Let H_i $(1 \leq i \leq s)$ be the components of a split decomposition of a graph G. Then G is word-representable if and only if H_i is word-representable for each $1 \leq i \leq s$. Further, $\mathcal{R}(G) = \max\limits_{1\leq i\leq s} \mathcal{R}(H_i)$.*

In view of Theorem 1.3.28, since circle graphs are characterized as 2-word-representable graphs, we have the following corollary of Theorem 2.1.4.

Corollary 2.1.5. *Let H_i $(1 \leq i \leq s)$ be the components of a split decomposition of a graph G. Then G is a circle graph if and only if each H_i is a circle graph.*

Since the components of the minimal split decomposition of a graph are cliques, stars and prime graphs, we have the following corollary of Theorem 2.1.4 on the non-word-representability of perfect graphs, while a characterization for word-representable perfect graphs is an open problem.

Corollary 2.1.6. *A perfect graph is not word-representable if and only if at least one of its prime components is a non-word-representable perfect graph.*

Since the class of perfect graphs is closed under split recomposition (by Theorem 1.4.15), non-word-representable perfect graphs are completely determined by the prime perfect graphs that are not word-representable.

Uisng Theorem 2.1.4, we now establish a class of word-representable graphs, namely parity graphs, within perfect graphs which includes bipartite graphs as well as distance-hereditary graphs.

Definition 2.1.7 (Parity graph)**.** A graph is called a *parity graph* if the lengths of any two induced paths between a pair of vertices are of the same parity.

Theorem 2.1.8 ([Sachs, 1970])**.** *The class of parity graphs is a subclass of perfect graphs.*

Theorem 2.1.9 ([Cicerone and Di Stefano, 1999])**.** *A graph G is a parity graph if and only if every component in its minimal split decomposition is a clique or a bipartite graph.*

Since cliques and bipartite graphs are word-representable (cf. Example 1.3.12), in view of Theorem 2.1.9, we have the following corollary of Theorem 2.1.4.

Corollary 2.1.10. *The class of parity graphs is word-representable.*

2.2 Comparability Graphs

In this section, we study the split recomposition of comparability graphs. First we observe that the class of comparability graphs is not closed under split recomposition.

In Subsection 2.2.1, we give a sufficient condition so that a recomposition of comparability graphs is a comparability graph. Further, in Subsection 2.2.2, we investigate the *prn* of a recomposition of comparability graphs. Finally, in Subsection 2.2.3, we provide a characteristic property by which a recomposition of comparability graphs is a comparability graph.

Remark 2.2.1. Let G be a comparability graph. Then every split component of G is a comparability graph, as it is isomorphic to an induced subgraph of G and every induced subgraph of a comparability graph is a comparability graph.

However, as shown in the following example, split recomposition of two comparability graphs is not always a comparability graph.

Example 2.2.2. The graphs G and G' shown in Figure 2.1 are comparability graphs. However, the split recomposition $G\,{}_m\circledast_{m'}\,G'$ is not a comparability graph (cf. Remark 1.2.9).

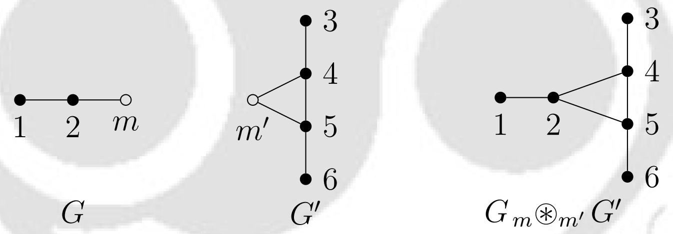


Figure 2.1: Recomposition of two comparability graphs

Notation 2.2.3. For the rest of this chapter, let $G = (V \cup \{m\}, E)$ and $G' = (V' \cup \{m'\}, E')$ be two graphs with disjoint vertex sets. Consider the graph

$$H = (V \cup V' \cup \{m, m'\}, E \cup E' \cup \{\overline{mm'}\}),$$

which is obtained by adding the edge $\overline{mm'}$ between G and G'. This graph operation is useful for investigating a sufficient condition and the *prn* of $G\,{}_m\circledast_{m'}\,G'$.

2.2.1 Sufficient Condition

In this subsection, we give a necessary and sufficient condition for H to be a comparability graph. Further, we also discuss the conditions under which the recomposition $G\ {}_{m}\circledast_{m'}\ G'$ is a comparability graph. As a consequence, we also establish that a recomposition of bipartite graphs is a bipartite graph.

Theorem 2.2.4. *The graph H is a comparability graph if and only if there exist transitive orientations O and O' of G and G', respectively, such that m is a source in O and m' is a sink in O', or vice versa.*

Proof. Suppose O and O' are transitive orientations of G and G', respectively, such that m is a source in O and m' is a sink in O'. We extend O and O' to an orientation O_H of H by orienting the edge $\overline{mm'}$ as $\overrightarrow{mm'}$. Similarly, if m is a sink in O and m' is a source in O', then assign $\overrightarrow{m'm}$ as the orientation to the edge $\overline{mm'}$ in O_H. In both cases, clearly, O_H is a transitive orientation of H so that H is a comparability graph.

Conversely, suppose H is a comparability graph. Let O_H be a transitive orientation of H in which the edge $\overline{mm'}$ is oriented as $\overrightarrow{mm'}$. We claim that m is a source and m' is a sink with respect to O_H. On the contrary, we have the following cases:

- m is not a source: Then, for some $a \in V$, we have $\overrightarrow{am}$ in O_H. By transitivity of O_H, $\overrightarrow{am'}$ exists in O_H; which is a contradiction as $\overline{am'}$ is not an edge in H.

- m' is not a sink: Then, for some $b \in V'$, we have $\overrightarrow{m'b}$ in O_H. By transitivity of O_H, $\overrightarrow{mb}$ exists in O_H; which is a contradiction as $\overline{mb}$ is not an edge in H.

This proves our claim. Note that $H[V \cup \{m\}] = G$ and $H[V' \cup \{m'\}] = G'$. Further, let O and O' be the restrictions of O_H to G and G', respectively. Clearly, O and O' are transitive orientations of G and G', respectively. Moreover, m is a source in O and m' is a sink in O'.

Similarly, if the edge $\overline{mm'}$ of H is oriented as $\overrightarrow{m'm}$ in O_H, then we can show that there exist transitive orientations O and O' of G and G', respectively, such that m is a sink of O and m' is a source of O'. □

We now present a condition under which a recomposition of comparability graphs is a comparability graph.

Theorem 2.2.5. *If G and G' are comparability graphs such that m is a source of G and m' is a sink of G' with respect to some transitive orientations of G and G', then $G\,_m\circledast_{m'} G'$ is a comparability graph.*

Proof. By Theorem 2.2.4, H is a comparability graph in which m is a source and m' is a sink. In view of the discussion in [Cornelsen and Di Stefano, 2009, Section 3], we give a transitive orientation to $G\,_m\circledast_{m'} G'$ from a transitive orientation of H by deleting m and m' and by orienting $\overrightarrow{ba}$ for all adjacent vertices $a(\neq m')$ of m and $b(\neq m)$ of m'. □

Remark 2.2.6. In Theorem 2.2.5, if m is a sink and m' is a source with respect to some transitive orientations of G and G', respectively, then also a transitive orientation can be assigned to $G\,_m\circledast_{m'} G'$. Hence, in the rest of this chapter, we consider one of the combinations to state the corresponding results.

Remark 2.2.7. Note that every bipartite graph is a comparability graph. For instance, if $(A \cup B, E)$ is a bipartite graph with bipartition $\{A, B\}$, then assign the transitive orientation to the edges in E as $\overrightarrow{ab}$ for $a \in A$ and $b \in B$. In this orientation, every vertex of A is a source and every vertex of B is a sink. Accordingly, in view of Theorem 2.2.5, if G and G' are two bipartite graphs, then the split recomposition $G\,_m\circledast_{m'} G'$ is a comparability graph for any choice of vertices m and m'.

Moreover, we have the following stronger result for the bipartite graphs.

Proposition 2.2.8. *If G and G' are two bipartite graphs, then for any choice of vertices m and m', the split recomposition $G\,_m\circledast_{m'} G'$ is a bipartite graph.*

Proof. Let $\{A, B\}$ be a bipartition of G and $\{A', B'\}$ be a bipartition of G'. Without loss of generality, suppose $m \in A$ and $m' \in A'$. We prove that $\{(A \setminus \{m\}) \cup B', (A' \setminus \{m'\}) \cup B\}$ is a bipartition of the vertices of $G\,{}_m\circledast_{m'}\,G'$. Consider an edge $\overline{ab}$ of $G\,{}_m\circledast_{m'}\,G'$. Then $\overline{ab}$ is in one of the induced subgraphs $G[V]$ and $G'[V']$, or it is an edge between $N_G(m)$ and $N_{G'}(m')$. In any case, it can be observed that if $a \in (A \setminus \{m\}) \cup B'$ then $b \in (A' \setminus \{m'\}) \cup B$, or vice versa. □

2.2.2 Permutation-Representation Number

In this subsection, we study the *prn* of the graph H and obtain a partial result on the *prn* of $G\,{}_m\circledast_{m'}\,G'$. A complete result on the *prn* of $G\,{}_m\circledast_{m'}\,G'$ is given in Subsection 2.2.3, subsequent to its characterization.

Lemma 2.2.9. *Let G be a comparability graph with a transitive orientation O. If m is a source with respect to O, then m is a minimal element of the induced poset P_G. Similarly, if m is a sink, then m is a maximal element of P_G.*

Proof. Let us assume that m is not a minimal element of P_G. Then there exists an element a in P_G such that $a \prec m$ in P_G which implies there is an edge $\overrightarrow{am}$ in G with respect to O. This is a contradiction to m being a source. Similarly, the result can be proved when m is a sink. □

The following remark produces a word to represent a comparability graph based on a realizer of its induced poset.

Remark 2.2.10. Let C be a comparability graph and P_C be the poset induced by a transitive orientation of C. Let $\{L_1, L_2, \ldots, L_k\}$ be a realizer of P_C. For $1 \le i \le k$, construct permutations p_i from (P_C, L_i) on the vertices of C such that a occurs before b in the permutation p_i if aL_ib. Then, in view of [Halldórsson et al., 2011, Lemma 4], the word $p_1p_2\cdots p_k$ represents C.

In the following proposition, we present a way of extending the permutations representing $G[V]$ to the permutations representing a comparability graph G with m as a source (or a sink).

Proposition 2.2.11. *Let G be a comparability graph on the vertex set $V \cup \{m\}$ such that m is a source (or a sink) with respect to some transitive orientation of G. Then there exist k and permutations p_i on V, for $1 \leq i \leq k$, which can be extended to permutations q_i on $V \cup \{m\}$, for $1 \leq i \leq k+1$, such that $p_1 \cdots p_k$ represents $G[V]$ and $q_1 \cdots q_k q_{k+1}$ represents G.*

Proof. Consider the poset P_G induced by the transitive orientation of G for which m is a source. Note that m is a minimal element of P_G. Now consider the subposet of P_G with the elements of V and note that the corresponding comparability graph is $G[V]$. Accordingly, we denote the subposet by $P_{G[V]}$.

Consider a realizer $L = \{L_1, L_2, \ldots, L_k\}$ of $P_{G[V]}$. Using the construction in [Hiraguti, 1955, Theorem 4.2] (see also [Trotter, 1992, Theorem 9.3 in Chapter 1] and [Bogart, 1973, Theorem 2.11]), we extend L to a realizer $M = \{M_1, \ldots, M_k, M_{k+1}\}$ of P_G as per the following: Let $U(m) = \{a \in V \mid m \prec a \text{ in } P_G\}$ and set

$$M_i = m \prec L_i, \quad (1 \leq i \leq k);$$
$$M_{k+1} = L_k(V \setminus U(m)) \prec m \prec L_k(U(m)),$$

where, for $X \subseteq V$, $L_k(X)$ is the linear order on X obtained from L_k by restricting it to X.

For $1 \leq i \leq k$, construct permutations p_i from $(P_{G[V]}, L_i)$ such that a occurs before b in p_i if aL_ib. Similarly, for $1 \leq i \leq k+1$, construct permutations q_i from (P_G, M_i). Note that, as m is a minimal element of P_G, we have $N_G(m) = U(m)$. Accordingly, note that

$$q_i = mp_i, \quad (1 \leq i \leq k);$$

$$q_{k+1} = p_k|_{V\setminus N_G(m)} m p_k|_{N_G(m)}.$$

Hence, by Remark 2.2.10, the words $p_1p_2\cdots p_k$ and $q_1q_2\cdots q_kq_{k+1}$ represent the graphs $G[V]$ and G, respectively.

Similarly, if m is a sink, there exist permutations p_i on V, for $1 \le i \le k$, and permutations q_i on $V \cup \{m\}$, for $1 \le i \le k+1$, such that the words $p_1p_2\cdots p_k$ and $q_1q_2\cdots q_kq_{k+1}$ represent $G[V]$ and G, respectively, where the permutations q_i are as per the following:

$$q_i = p_i m, \quad (1 \le i \le k);$$
$$q_{k+1} = p_k|_{N_G(m)} m p_k|_{V\setminus N_G(m)}.$$

□

Note that $\mathcal{R}^p(G) = k$ implies $\dim(P_G) = k$ so that $\dim(P_{G[V]}) \le k$. Accordingly, Proposition 2.2.11 is rephrased in the following lemma for the graphs G and G'. The lemma is useful for obtaining the *prn* of their recomposition and also that of H.

Lemma 2.2.12. *Suppose G and G' are comparability graphs such that m is a source of G and m' is a sink of G' with respect to some transitive orientations of G and G'. Let $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$. Then there exist permutations p_i $(1 \le i \le k)$ on V, p'_i $(1 \le i \le k')$ on V' which can be extended to the permutations q_i $(1 \le i \le k+1)$ on the vertices of G and permutations q'_i $(1 \le i \le k'+1)$ on the vertices of G' such that the words $p_1p_2\cdots p_k$ and $p'_1p'_2\cdots p'_{k'}$ represent $G[V]$ and $G'[V']$, respectively, and the words $w = q_1q_2\cdots q_kq_{k+1}$ and $w' = q'_1q'_2\cdots q'_{k'}q'_{k'+1}$ represent the graphs G and G', respectively, where*

$$\begin{aligned} q_i &= mp_i, \quad (1 \le i \le k); & q'_i &= p'_i m', \quad (1 \le i \le k'); \\ q_{k+1} &= p_k|_{V\setminus N_G(m)} m p_k|_{N_G(m)}; & q'_{k'+1} &= p'_{k'}|_{N_{G'}(m')} m' p'_{k'}|_{V'\setminus N_{G'}(m')}. \end{aligned}$$

Theorem 2.2.13. *Let G and G' be comparability graphs such that $\mathcal{R}^p(G) = k$ and*

$\mathcal{R}^p(G') = k'$. *If the graph H is a comparability graph, then $\mathcal{R}^p(H) = \max\{k, k'\}$ or $1 + \max\{k, k'\}$.*

Proof. Suppose H is a comparability graph. By Theorem 2.2.4, m is a source and m' is a sink with respect to some transitive orientations of G and G', respectively.

As $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$, using Lemma 2.2.12, consider the permutations p_i $(1 \le i \le k)$ on V, p'_i $(1 \le i \le k')$ on V', q_i $(1 \le i \le k+1)$ on the vertices of G, and q'_i $(1 \le i \le k'+1)$ on the vertices of G' such that the words $w = q_1q_2\cdots q_kq_{k+1}$ and $w' = q'_1q'_2\cdots q'_{k'}q'_{k'+1}$ represent the graphs G and G', respectively.

Suppose $k' < k$. We construct $k+1$ permutations u_i $(1 \le i \le k+1)$ on the vertices of H as per the following:

$$\begin{aligned}
u_1 &= p'_1mp_1m',\\
u_i &= p'_imm'p_i, \quad (2 \le i \le k')\\
u_j &= p'_{k'}mm'p_j, \quad (k'+1 \le j \le k)\\
u_{k+1} &= q_{k+1}q'_{k'+1}\\
&= p_k|_{V\setminus N_G(m)}mp_k|_{N_G(m)}p'_{k'}|_{N_{G'}(m')}m'p'_{k'}|_{V'\setminus N_{G'}(m')}
\end{aligned}$$

We now show that the word $u = u_1u_2\cdots u_ku_{k+1}$ represents the graph H. Note that $H[V\cup\{m\}] = G$ and $H[V'\cup\{m'\}] = G'$. Further, since

$$\begin{aligned}
u|_{V\cup\{m\}} &= q_1q_2 \ \cdots \ q_kq_{k+1} \text{ and}\\
u|_{V'\cup\{m'\}} &= q'_1q'_2 \ \cdots \ q'_{k'-1}\underbrace{q'_{k'} \ \cdots \ q'_{k'}}_{k-k'+1 \text{ times}}q'_{k'+1},
\end{aligned}$$

we have $u|_{V\cup\{m\}}$ and $u|_{V'\cup\{m'\}}$ represent the graphs G and G', respectively. Hence, any two vertices of G or of G' are adjacent if and only if they alternate in the word u. Also, corresponding to the edge $\overline{mm'}$ in H, it is easy to see that m and m' alternate in u, as $mm' \ll u_i$ for all $1 \le i \le k+1$.

Note that there are no edges between the graphs G and G' except $\overline{mm'}$; accordingly, we show that the corresponding vertices do not alternate in u. For $a \in V$ and $a' \in V' \cup \{m'\}$, as $a'a \ll u_2$ but $aa' \ll u_{k+1}$, it is evident that a and a' do not alternate in u. Also, for $a \in V \cup \{m\}$ and $a' \in V'$, we have a and a' do not alternate in u as $a'a \ll u_1$ but $aa' \ll u_{k+1}$. Hence, the word u represents the graph H.

Similarly, when $k \leq k'$, we can show that the word $u = u_1u_2 \cdots u_{k'}u_{k'+1}$ represents the graph H, where

$$
\begin{aligned}
u_1 &= p_1'mp_1m'; \\
u_i &= p_i'mm'p_i, \quad (2 \leq i \leq k); \\
u_j &= p_j'mm'p_k, \quad (k+1 \leq j \leq k'); \\
u_{k'+1} &= q_{k+1}q_{k'+1}' \\
&= p_k|_{V \setminus N_G(m)} m p_k|_{N_G(m)} p_{k'}'|_{N_{G'}(m')} m' p_{k'}'|_{V' \setminus N_{G'}(m')}.
\end{aligned}
$$

In any case, since u is concatenation of $1 + \max\{k, k'\}$ permutations, we have $\mathcal{R}^p(H) \leq 1 + \max\{k, k'\}$. Also, since G and G' are induced subgraphs of H, we have $\mathcal{R}^p(H) \geq \max\{k, k'\}$. Hence, $\mathcal{R}^p(H) = \max\{k, k'\}$ or $1 + \max\{k, k'\}$. □

Theorem 2.2.14. *Let G and G' be comparability graphs such that m is a source of G and m' is a sink of G' with respect to some transitive orientations of G and G'. If $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$, then $\mathcal{R}^p(G\,{}_m{\circledast}_{m'}\,G') = \max\{k, k'\}$ or $1 + \max\{k, k'\}$.*

Proof. By Theorem 2.2.5, note that $G\,{}_m{\circledast}_{m'}\,G'$ is a comparability graph and hence permutationally representable. As $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$, using Lemma 2.2.12, consider the permutations p_i $(1 \leq i \leq k)$ on V, p_i' $(1 \leq i \leq k')$ on V', q_i $(1 \leq i \leq k+1)$ on the vertices of G, and q_i' $(1 \leq i \leq k'+1)$ on the vertices of G' such that the words $w = q_1q_2 \cdots q_kq_{k+1}$ and $w' = q_1'q_2' \cdots q_{k'}'q_{k'+1}'$ represent the graphs G and G', respectively.

Suppose $k < k'$. We construct $k' + 1$ permutations v_i $(1 \leq i \leq k' + 1)$ on the vertices of $G\,{}_m\circledast_{m'}\,G'$ as per the following:

$$\begin{aligned} v_i &= p'_i p_i, \quad (1 \le i \le k) \\ v_j &= p'_j p_k, \quad (k+1 \le j \le k') \\ v_{k'+1} &= p_k|_{V\setminus N_G(m)} p'_{k'}|_{N_{G'}(m')} p_k|_{N_G(m)} p'_{k'}|_{V'\setminus N_{G'}(m')} \end{aligned}$$

We show that the word $v = v_1 v_2 \cdots v_{k'} v_{k'+1}$ represents the graph $G\,{}_m\circledast_{m'}\,G'$. Note that $G[V]$ and $G'[V']$ are induced subgraphs of $G\,{}_m\circledast_{m'}\,G'$ represented by

$$\begin{aligned} w|_V &= p_1 p_2 \;\cdots\; p_k p_k|_{V\setminus N_G(m)} p_k|_{N_G(m)}, \text{ and} \\ w'|_{V'} &= p'_1 p'_2 \;\cdots\; p'_{k'} p'_{k'}|_{N_{G'}(m')} p'_{k'}|_{V'\setminus N_{G'}(m')}, \end{aligned}$$

respectively. Further, note that

$$\begin{aligned} v|_V &= p_1 p_2 \;\cdots\; p_{k-1} \underbrace{p_k \;\cdots\; p_k}_{k'-k+1 \text{ times}} p_k|_{V\setminus N_G(m)} p_k|_{N_G(m)}, \text{ and} \\ v|_{V'} &= p'_1 p'_2 \;\cdots\; p'_{k'} p'_{k'}|_{N_{G'}(m')} p'_{k'}|_{V'\setminus N_{G'}(m')}. \end{aligned}$$

Thus, any two vertices of $G[V]$ or of $G'[V']$ are adjacent if and only if they alternate in the word v.

For $a \in V$ and $a' \in V'$, note that a and a' are adjacent in $G\,{}_m\circledast_{m'}\,G'$ if and only if $a \in N_G(m)$ and $a' \in N_{G'}(m')$. Accordingly, we consider the following cases to show that a and a' are adjacent in $G\,{}_m\circledast_{m'}\,G'$ if and only if they alternate in v. If $a \in N_G(m)$ and $a' \in N_{G'}(m')$, then from the construction of v, we have $a'a \ll v_i$ for all $1 \le i \le k'+1$, so that a and a' alternate in v. If $a \notin N_G(m)$, then for any $a' \in V'$, we have $a'a \ll v_1$ but $aa' \ll v_{k'+1}$. Similarly, if $a' \notin N_{G'}(m')$, then for any vertex $a \in V$, we have $a'a \ll v_1$ but $aa' \ll v_{k'+1}$. Hence, a and a' do not alternate in v, if $a \notin N_G(m)$ or $a' \notin N_{G'}(m')$. Hence the word v represents $G\,{}_m\circledast_{m'}\,G'$ permutationally.

Similarly, when $k' \leq k$, we can show that the word $v = v_1 v_2 \cdots v_k v_{k+1}$ represents the graph $G\,{}_m\circledast_{m'}\,G'$ permutationally, where

$$\begin{aligned} v_i &= p'_i p_i, \quad (1 \leq i \leq k'); \\ v_j &= p'_{k'} p_j, \quad (k'+1 \leq j \leq k); \\ v_{k+1} &= p_k|_{V \setminus N_G(m)} p'_{k'}|_{N_{G'}(m')} p_k|_{N_G(m)} p'_{k'}|_{V' \setminus N_{G'}(m')}. \end{aligned}$$

In any case, since v is concatenation of $1 + \max\{k, k'\}$ permutations, we have $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') \leq 1 + \max\{k, k'\}$. Also, since G and G' are isomorphic to certain induced subgraphs of $G\,{}_m\circledast_{m'}\,G'$ (cf. Remark 1.4.14), we have $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') \geq \max\{k, k'\}$. Hence, $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = \max\{k, k'\}$ or $1 + \max\{k, k'\}$. $\square$

In view of Proposition 2.2.8, we have the following corollary of Theorem 2.2.14.

Corollary 2.2.15. *If G and G' are bipartite graphs with $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$, then $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = \max\{k, k'\}$ or $1 + \max\{k, k'\}$.*

2.2.3 Characterization

In this subsection, we provide a sufficient condition (see Theorem 2.2.17) and a necessary condition (see Theorem 2.2.20) for $G\,{}_m\circledast_{m'}\,G'$ to be a comparability graph. Consequently, we determine the *prn* of $G\,{}_m\circledast_{m'}\,G'$.

Recall from Theorem 2.2.5 that recomposition of two comparability graphs is a comparability graph when the marked vertices are a source and a sink with respect to their transitive orientations. However, the marked vertices need not be a source or a sink to give a transitive orientation to the recomposition, as shown in the following example.

Example 2.2.16. Consider the comparability graphs G_1 and G_2 given in Figure 2.2 in which a and b are the marked vertices. Note that b is not a source or a sink under any transitive orientation given to G_2. However, $G_1 \ {}_a\circledast_b\ G_2$ is a comparability graph.

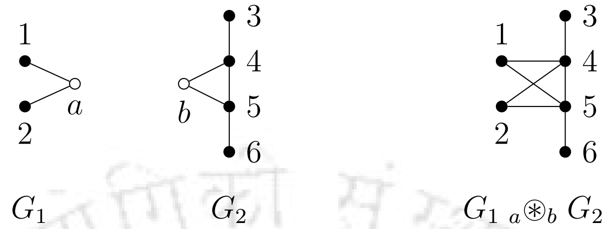


Figure 2.2: Recomposition of comparability graphs

Note that in Example 2.2.16, the vertex a is an all-adjacent vertex, i.e., a is adjacent to all other vertices of G_1. In connection to this observation, we prove the following theorem.

Theorem 2.2.17. *Let G and G' be two comparability graphs such that at least one of m and m' is an all-adjacent vertex. Then $G\,{}_m\circledast_{m'}\,G'$ is a comparability graph. Moreover, if $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$, then $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = \max\{k, k'\}$.*

Proof. We prove the result by constructing requisite number of permutations on the vertices of $G\,{}_m\circledast_{m'}\,G'$ whose concatenation represents $G\,{}_m\circledast_{m'}\,G'$.

Let the words $p_1p_2\cdots p_k$ and $p'_1p'_2\cdots p'_{k'}$ represent G and G', respectively, where each p_i $(1 \le i \le k)$ is a permutation on the vertices of G and each p'_i $(1 \le i \le k')$ is a permutation on the vertices of G'.

Suppose $\max\{k, k'\} = t$. Without loss of generality let $k' \le k$. Set $p'_j = p'_{k'}$ for all $k'+1 \le j \le t$ and note that $p'_1p'_2\cdots p'_t$ represents G'. Then, the words $w = p_1p_2\cdots p_t$ and $w' = p'_1p'_2\cdots p'_t$ represent the graphs G and G', respectively. For $1 \le i \le t$, let $p_i = r_ims_i$ and $p'_i = r'_im's'_i$ so that $p_i|_V = r_is_i$ and $p'_i|_{V'} = r'_is'_i$.

Suppose m is an all-adjacent vertex of G. Let $u_i = r'_ir_is_is'_i$, for all $1 \le i \le t$. Note that each u_i is a permutation on the vertices of $G\,{}_m\circledast_{m'}\,G'$. We show that the word $u = u_1u_2\cdots u_t$ represents the graph $G\,{}_m\circledast_{m'}\,G'$ permutationally.

Note that the induced subgraphs $G[V]$ and $G'[V']$ are induced subgraphs of $G\,{}_m\circledast_{m'}\,G'$. Further, since $u|_V = p_1|_V p_2|_V \cdots p_t|_V = w|_V$ and $u|_{V'} = p'_1|_{V'} p'_2|_{V'} \cdots p'_t|_{V'}$ $= w'|_{V'}$, the subwords $u|_V$ and $u|_{V'}$ of u represent the graphs $G[V]$ and $G'[V']$, respectively. Thus, any two vertices of $G[V]$ or of $G'[V']$ are adjacent if and only if they alternate in the word u.

For $a \in V$ and $a' \in V'$, note that a and a' are adjacent in $G\,{}_m\circledast_{m'}\,G'$ if and only if $a \in N_G(m)$ and $a' \in N_{G'}(m')$. However, note that $V = N_G(m)$ as m is an all-adjacent vertex of G. Accordingly, we consider the two cases based on $a' \in N_{G'}(m')$ or $a' \notin N_{G'}(m')$ to show that a and a' are adjacent in $G\,{}_m\circledast_{m'}\,G'$ if and only if they alternate in u so that u represents the graph $G\,{}_m\circledast_{m'}\,G'$.

- Case 1: $a' \in N_{G'}(m')$. Since w' represents G', we have $a'm' \ll p'_i$ for all i, or $m'a' \ll p'_i$ for all i, $1 \le i \le t$. Observe that each u_i $(1 \le i \le t)$ is constructed from p'_i by replacing m' with the permutation $p_i|_V$. Hence, if $a'm' \ll p'_i$ for all i, then $a'a \ll u_i$ for all i. If $m'a' \ll p'_i$ for all i, then $aa' \ll u_i$ for all i. In any case, a and a' alternate in u.

- Case 2: $a' \notin N_{G'}(m')$. Then a' and m' do not alternate in w'. Accordingly, there exist i and j $(1 \le i, j \le t)$ such that $a'm' \ll p'_i$ and $m'a' \ll p'_j$. Again, as per the construction of u_i and u_j, we have $a'a \ll u_i$ and $aa' \ll u_j$. Hence, a and a' do not alternate in u.

Similarly, if m' is an all-adjacent vertex of G', we can show that the word $u = u_1u_2\cdots u_t$ represents the graph $G\,{}_m\circledast_{m'}\,G'$ permutationally, where $u_i = r_i r'_i s'_i s_i$, for $1 \le i \le t$.

Therefore, we have $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') \le t = \max\{k, k'\}$. But, by Remark 1.4.14, since G and G' are isomorphic to certain induced subgraphs of $G\,{}_m\circledast_{m'}\,G'$, we have $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') \ge t$. Hence, $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = \max\{k, k'\}$. $\square$

Notation 2.2.18. Let a be a vertex of a graph C. Then the graph obtained from C by adding a new vertex a' adjacent only to the vertex a of C is denoted by C^a.

We now recall a theorem that characterizes a source of a comparability graph.

Theorem 2.2.19 ([Gimbel, 1992; Olariu, 1992]). *Let C be a comparability graph and a be a vertex of C. Then a is a source with respect to some transitive orientation of C if and only if C^a is a comparability graph.*

Now, using Theorem 2.2.19, we find a necessary condition for $G\,_m\circledast_{m'}\,G'$ to be a comparability graph, when one of m and m' is not a source (or a sink) with respect to any transitive orientation of G or G'.

Theorem 2.2.20. *Let G and G' be two comparability graphs such that m is not a source with respect to any transitive orientation of G. If $G\,_m\circledast_{m'}\,G'$ is a comparability graph, then m' is an all-adjacent vertex of G'.*

Proof. Let $G\,_m\circledast_{m'}\,G'$ be a comparability graph. Assume m' is not an all-adjacent vertex of G'. Then there exists a vertex $c' \in V'$ such that $c' \notin N_{G'}(m')$ but $\overline{b'c'} \in E'$ for some vertex $b' \in N_{G'}(m')$. The connectedness of the graph G' ensures the existence of such vertex c'.

Consider the induced subgraph $(G\,_m\circledast_{m'}\,G')[V \cup \{b'\}]$, say ${}^mG_{b'}$. Note that b' is adjacent to all vertices of $N_G(m)$ in $G\,_m\circledast_{m'}\,G'$. Hence, ${}^mG_{b'}$ is isomorphic to G as it can be obtained from G by replacing m with b'. While ${}^mG_{b'}$ is a comparability graph, b' cannot be a source with respect to any transitive orientation of ${}^mG_{b'}$, since m is not a source by the assumption in our statement of the theorem.

Now consider the induced subgraph $(G\,_m\circledast_{m'}\,G')[V \cup \{b', c'\}]$ of $G\,_m\circledast_{m'}\,G'$. Note that $(G\,_m\circledast_{m'}\,G')[V \cup \{b', c'\}] = ({}^mG_{b'})^{b'}$, the graph obtained from ${}^mG_{b'}$ by making c' adjacent only to b'. Since b' cannot be a source, $({}^mG_{b'})^{b'}$ is not a comparability graph, by Theorem 2.2.19. However, $({}^mG_{b'})^{b'}$ is a comparability graph as it is an induced subgraph of the comparability graph $G\,_m\circledast_{m'}\,G'$; a contradiction. Hence, m' should be an all-adjacent vertex of G'. $\square$

We summarize the results of this section in the following theorem, which provides us a characterization for $G\,_m\circledast_{m'}\,G'$ to be a comparability graph.

Theorem 2.2.21. *Let $G = (V \cup \{m\}, E)$ and $G' = (V' \cup \{m'\}, E')$ be two comparability graphs with disjoint vertex sets. Suppose $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$. Then we have the following:*

1. *The split recomposition $G\,{}_m\circledast_{m'}\,G'$ is a comparability graph if and only if one of the following holds:*

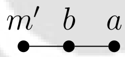
   (i) The vertices m and m' are a source and a sink, or vice versa, with respect to some transitive orientations on G and G'.

   (ii) At least one of the vertices m and m' is an all-adjacent vertex.

2. *Moreover, if the split recomposition $G\,{}_m\circledast_{m'}\,G'$ is a comparability graph, then $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = \max\{k, k'\}$ or $1 + \max\{k, k'\}$.*

As a consequence, we also provide the *prn* of the graph construction G^m in the following corollary.

Corollary 2.2.22. *Let G be a comparability graph such that $\mathcal{R}^p(G) = k$. If G^m is a comparability graph, then $\mathcal{R}^p(G^m) = k$ or $1 + k$.*

Proof. If G^m is a comparability graph, then by Theorem 2.2.19, m is a source of G. Consider a path P_3: 

and note that $\mathcal{R}^p(P_3) = 2$, as the word $bam'bm'a$ represents P_3. Further, it can be observed that G^m is isomorphic to $G\,{}_m\circledast_{m'}\,P_3$.

If $k = 1$, then G is a complete graph so that m is an all-adjacent vertex of G. Accordingly, by Theorem 2.2.17, $\mathcal{R}^p(G^m) = \mathcal{R}^p(G\,{}_m\circledast_{m'}\,P_3) = \max\{1, 2\} = 2$. Otherwise, for $k \geq 2$, $\max\{k, 2\} = k$. Then, by Theorem 2.2.21, $\mathcal{R}^p(G^m) = k$ or $1 + k$. Hence, in any case, we have $\mathcal{R}^p(G^m) = k$ or $1 + k$. □

2.3 *prn*-Irreducible Graphs

Following the notion of irreducible posets in order theory (cf. [Trotter, 1992]), in this section, we define the concept of *prn*-irreducible comparability graphs and study them with respect to split recomposition. In this connection, we characterize *prn*-irreducible graphs such that their recomposition is a comparability graph. Accordingly, we determine the *prn* of the resulting comparability graph based on the *prn*s of its split components. We also observe that the split recomposition of *prn*-irreducible graphs is not *prn*-irreducible.

Definition 2.3.1 (*prn*-irreducible graph)**.** A comparability graph $C = (V, E)$ is said to be *k-prn-irreducible* for some $k \geq 2$, if $\mathcal{R}^p(C) = k$ and $\mathcal{R}^p(C[V \setminus \{a\}]) = k - 1$ for every $a \in V$. Further, a comparability graph is said to be *prn-irreducible* if it is k-*prn*-irreducible for some $k \geq 2$.

Remark 2.3.2. A comparability graph is 2-*prn*-irreducible if and only if it is an edgeless graph on two vertices. For instance, any connected graph C on at least 3 vertices with $\mathcal{R}^p(C) = 2$ has a vertex a that is not adjacent to all other vertices. If we remove any vertex adjacent to a from C, the *prn* of the resulting graph is not one, as it is not a complete graph.

Example 2.3.3. For $n \geq 3$, consider the crown graph $H_{n,n}$ (a graph which is obtained by deleting a perfect matching from the complete bipartite graph $K_{n,n}$). From [Halldórsson et al., 2011], recall that $\mathcal{R}^p(H_{n,n}) = n$. Let H' be the bipartite graph obtained by deleting a vertex from $H_{n,n}$. Note that $\mathcal{R}^p(H') \leq n - 1$ as $n - 1$ is the size of one of the partite sets of H' [Mozhui and Krishna, 2025b, Corollary 6], and $n - 1 \leq \mathcal{R}^p(H')$ as $H_{n-1,n-1}$ is an induced subgraph of H' (cf. [Mozhui and Krishna, 2025b]). Hence, $\mathcal{R}^p(H') = n - 1$ so that $H_{n,n}$ is n-*prn*-irreducible.

Example 2.3.4. For $n \geq 3$, the cycle C_{2n} is 3-*prn*-irreducible. Note that by deleting a vertex from C_{2n} we get the path P_{2n-1} with $2n-1$ vertices. For $n \geq 3$, $\mathcal{R}^p(C_{2n}) = 3$ and $\mathcal{R}^p(P_n) = 2$ (by Theorem 1.3.46(ii)).

Trotter and Moore [1976], as well as Kelly [1977] independently and simultaneously obtained the complete list of 3-irreducible posets. A complete list of 3-*prn*-irreducible comparability graphs can be derived from the corresponding posets.

Proposition 2.3.5. *For $k \geq 2$, any k-prn-irreducible comparability graph does not have an all-adjacent vertex.*

Proof. On the contrary, suppose a is an all-adjacent vertex of a k-*prn*-irreducible comparability graph $C = (V, E)$. As $\mathcal{R}^p(C[V \setminus \{a\}]) = k - 1$, let $p_1 p_2 \cdots p_{k-1}$ represents the induced subgraph $C[V \setminus \{a\}]$, where each p_i $(1 \leq i \leq k-1)$ is a permutation on $V \setminus \{a\}$. Then consider the permutations $q_i = ap_i$, for all $1 \leq i \leq k-1$, on V and note that the word $q_1 q_2 \cdots q_{k-1}$ represents the graph C. Hence, $\mathcal{R}^p(C) = k - 1$; a contradiction. □

In view of Proposition 2.3.5, we have the following corollary of Theorem 2.2.21.

Corollary 2.3.6. *Let G and G' be comparability graphs such that G is k-prn-irreducible and G' is k'-prn-irreducible. Then $G\,{}_m\circledast_{m'}\,G'$ is a comparability graph if and only if m is a source of G and m' is a sink of G' with respect to some transitive orientations of G and G'.*

We now obtain the permutation-representation number of the split recomposition of two *prn*-irreducible comparability graphs in the following theorem.

Theorem 2.3.7. *For $k, k' \geq 3$, let G and G' be k-prn-irreducible and k'-prn-irreducible comparability graphs, respectively. If $G\,{}_m\circledast_{m'}\,G'$ is a comparability graph, then $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = \max\{k, k'\}$.*

Proof. Suppose $G\,{}_m\circledast_{m'}\,G'$ is a comparability graph. In view of Corollary 2.3.6, let m be a source of G and m' be a sink of G' with respect to some transitive orientations of G and G'. Note that $\mathcal{R}^p(G) = k$, $\mathcal{R}^p(G') = k'$ and $\mathcal{R}^p(G[V]) = k-1$, $\mathcal{R}^p(G'[V']) = k'-1$. Along the lines of Lemma 2.2.12, consider the permutations p_i $(1 \leq i \leq k-1)$ on V, p'_i $(1 \leq i \leq k'-1)$ on V'.

Suppose $k' < k$. We construct k permutations v_i $(1 \leq i \leq k)$ on the vertices of $G\,{}_m\circledast_{m'}\,G'$ as per the following and show that the word $v = v_1 v_2 \cdots v_k$ represents the graph $G_m \circledast_{m'} G'$.

$$
\begin{aligned}
v_i &= p'_i p_i, \quad (1 \le i \le k'-1) \\
v_j &= p'_{k'-1} p_j, \quad (k' \le j \le k-1) \\
v_k &= p_{k-1}|_{V \setminus N_G(m)} p'_{k'-1}|_{N_{G'}(m')} p_{k-1}|_{N_G(m)} p'_{k'-1}|_{V' \setminus N_{G'}(m')}
\end{aligned}
$$

Note that

$$
\begin{aligned}
v|_V &= p_1 p_2 \ \cdots \ p_{k-1} p_{k-1}|_{V \setminus N_G(m)} p_{k-1}|_{N_G(m)} \text{ and} \\
v|_{V'} &= p'_1 p'_2 \ \cdots \ p'_{k'-2} \underbrace{p'_{k'-1} \ \cdots \ p'_{k'-1}}_{k-k'+1 \quad \text{times}} p'_{k'-1}|_{N_{G'}(m')} p'_{k'-1}|_{V' \setminus N_{G'}(m')}.
\end{aligned}
$$

Thus, any two vertices of $G[V]$ or of $G'[V']$ are adjacent if and only if they alternate in the word v.

For $a \in V$ and $a' \in V'$, note that a and a' are adjacent in $G_m \circledast_{m'} G'$ if and only if $a \in N_G(m)$ and $a' \in N_{G'}(m')$. If $a \in N_G(m)$ and $a' \in N_{G'}(m')$, then from the construction of v, we have $a'a \ll v_i$ for all $1 \le i \le k$, so that a and a' alternate in v. If $a \notin N_G(m)$, then for any $a' \in V'$, we have $a'a \ll v_1$ but $aa' \ll v_k$. Similarly, if $a' \notin N_{G'}(m')$, then for any vertex $a \in V$, we have $a'a \ll v_1$ but $aa' \ll v_k$. Hence, a and a' do not alternate in v, if $a \notin N_G(m)$ or $a' \notin N_{G'}(m')$. Hence the word v represents $G_m \circledast_{m'} G'$ permutationally.

Similarly, when $k \le k'$, we can show that the word $v = v_1 v_2 \cdots v_{k'-1} v_{k'}$ represents the graph $G_m \circledast_{m'} G'$ permutationally, where

$$
\begin{aligned}
v_i &= p'_i p_i, \quad (1 \le i \le k-1); \\
v_j &= p'_j p_{k-1}, \quad (k \le j \le k'-1); \\
v_{k'} &= p_{k-1}|_{V \setminus N_G(m)} p'_{k'-1}|_{N_{G'}(m')} p_{k-1}|_{N_G(m)} p'_{k'-1}|_{V' \setminus N_{G'}(m')}.
\end{aligned}
$$

In any case, since v is concatenation of $\max\{k, k'\}$ permutations, we have

$\mathcal{R}^p(G\ {}_m\circledast_{m'} G') \leq \max\{k, k'\}$. Also, since G and G' are isomorphic to certain induced subgraphs of $G\ {}_m\circledast_{m'} G'$ (cf. Remark 1.4.14), we have $\mathcal{R}^p(G\ {}_m\circledast_{m'} G') \geq \max\{k, k'\}$. Hence, $\mathcal{R}^p(G\ {}_m\circledast_{m'} G') = \max\{k, k'\}$. $\square$

In view of Theorem 2.3.7, we have the *prn* of the following bipartite graphs, which are obtained from the recomposition of *prn*-irreducible graphs.

Example 2.3.8. For $k, k' \geq 3$, the *prn* of the recomposition of crown graphs $H_{k,k}$ and $H_{k',k'}$ (with respect to any two vertices of $H_{k,k}$ and $H_{k',k'}$) is $\max\{k, k'\}$.

Example 2.3.9. For $k, k' \geq 3$, the *prn* of the recomposition of even cycles C_{2k} and $C_{2k'}$ (with respect to any two vertices of C_{2k} and $C_{2k'}$) is 3.

Theorem 2.3.10. *For $k, k' \geq 3$, if G and G' are k-prn-irreducible and k'-prn-irreducible comparability graphs, respectively, then $G\ {}_m\circledast_{m'} G'$ is not prn-irreducible.*

Proof. Suppose $G\ {}_m\circledast_{m'} G'$ is a comparability graph. By Proposition 2.3.5, m and m' are not all-adjacent vertices. Also, by Theorem 2.3.7, we have $\mathcal{R}^p(G\ {}_m\circledast_{m'} G') = \max\{k, k'\} = k$, say, without loss of generality.

Since m' is not an all-adjacent vertex of G', there exists $c' \in V'$ such that $c' \notin N_{G'}(m')$. Consider the induced subgraph $L = (G\ {}_m\circledast_{m'} G')[V \cup (V' \setminus \{c'\})]$. Clearly, $\mathcal{R}^p(L) \leq k$. We show that $\mathcal{R}^p(L) = k$ so that $G\ {}_m\circledast_{m'} G'$ is not *prn*-irreducible.

For $b' \in N_{G'}(m')$, note that the induced subgraph $(G\ {}_m\circledast_{m'} G')[V \cup \{b'\}]$ is isomorphic to G and it is also an induced subgraph of L. Thus, $\mathcal{R}^p(L) \geq k$ and hence, $\mathcal{R}^p(L) = k$. $\square$

2.4 Conclusion and Future Work

The work in this chapter paves a way, in general, to study the word-representability of graphs using split decomposition and recomposition. As the problem of deciding whether a given graph is word-representable is NP-complete, using split decomposition and recomposition, the problem can be reduced to its prime components. For example, the parity graphs are established as word-representable, because their prime components are word-representable (see Corollary 2.1.10). In view of Corollary 2.1.6, the open problem of characterizing word-representable perfect graphs can be approached through the following problem: What are the prime perfect graphs that are not word-representable? Similar to bipartite graphs, it is also interesting to find the subclasses of comparability graphs which are closed under recomposition. If one of the marked vertices is an all-adjacent vertex in a general case, or in the case when G and G' are *prn*-irreducible graphs, we proved that $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = k$, where $k = \max\{\mathcal{R}^p(G), \mathcal{R}^p(G')\}$. However, there are two possibilities of *prn* for the recomposed comparability graph, i.e., $\mathcal{R}^p(G\,{}_m\circledast_{m'}\,G') = k$ or $1 + k$. Accordingly, one may target to classify the graphs with respect to the *prn* of their recomposition amongst the two categories.

3

Modular Decomposition

In this chapter, we study the word-representability of graphs using modular decomposition. In this connection, we consider a graph operation which replaces a module in place of a vertex in a graph and provide a characteristic property for the word-representability of the resulting graph. Consequently, we characterize the word-representable graphs with respect to modular decomposition. We also determine the representation number of a word-representable graph in terms of the permutation-representation numbers of the subgraphs induced by modules and the representation number of the associated quotient graph. In this context, we obtain a complete answer to the open problem posed by Kitaev and Lozin on the word-representability of the lexicographical product of graphs.

3.1 Characterizations

To study modular decomposition, first we consider the following operation which replaces a module in place of a vertex in a graph.

Definition 3.1.1. Let $G = (V \cup \{a\}, E)$, where $a \notin V$, and $H = (V', E')$ be two graphs. The graph $G_a[H] = (V'', E'')$ is defined by $V'' = V \cup V'$ and the edge set E'' consist of the edges of $G[V]$, edges of E' and $\left\{\overline{bc} \mid b \in V', c \in N_G(a)\right\}$. We say $G_a[H]$ is obtained from G by replacing the vertex a of G with the module H.

Remark 3.1.2. In Definition 3.1.1, note that V' is a module in the graph $G_a[H]$. For instance, let $b \in V'' \setminus V' = V$. Thus, both a and b are vertices of G, and we have either $b \notin N_G(a)$ or $b \in N_G(a)$. Thus, by definition of $G_a[H]$, either $N_{G_a[H]}(b) \cap V' = \varnothing$ or $V' \subseteq N_{G_a[H]}(b)$.

Remark 3.1.3. In Definition 3.1.1, if $V = \varnothing$, i.e., $V \cup \{a\} = \{a\}$, then $G_a[H]$ is nothing but H. Further, if $V \neq \varnothing$, then $G_a[H] = G\ {}_a\circledast_b\ H'$, where the graph H' is obtained by adding an all-adjacent vertex b to H.

Notation 3.1.4. For the rest of this chapter, G and H denote the graphs $G = (V \cup \{a\}, E)$ with $V \neq \varnothing, a \notin V$ and $H = (V', E')$, unless stated otherwise.

Remark 3.1.5. For two word-representable graphs G and H, the graph $G_a[H]$ is not always a word-representable graph. For instance, consider the graph obtained by replacing one of the two vertices of the complete graph K_2 with the cycle C_5. This graph is nothing but the wheel graph W_5 (see Figure 3.1), which is not a word-representable graph (cf. Remark 1.3.14).

The following theorem gives us a sufficient condition for the word-representability of $G_a[H]$.

Theorem 3.1.6 ([Kitaev, 2013])**.** *Suppose G is a word-representable graph. If H is a comparability graph, then $G_a[H]$ is word-representable.*

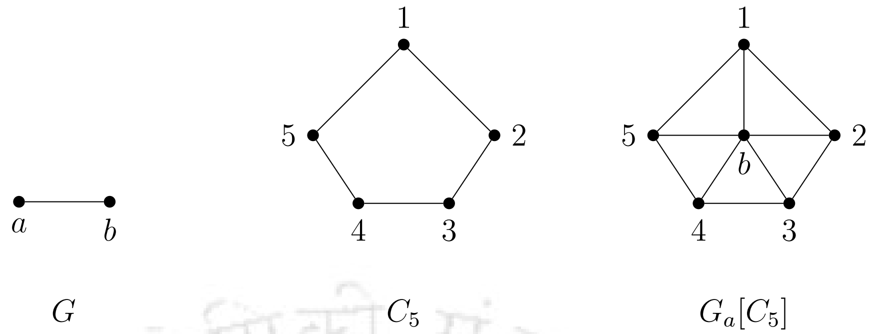


FIGURE 3.1: Replacing the vertex a of G with the module C_5

Further, on the representation number of $G_a[H]$, it was stated in [Kitaev, 2013] that if $\mathcal{R}(G) = k$ and $\mathcal{R}(H) = k'$, then $\mathcal{R}(G_a[H]) = \max\{k, k'\}$. Although it was stated that k' is the representation number of the graph H, the proof in [Kitaev, 2013] uses the concept of *prn* of H. If $k' = \mathcal{R}(H)$, then the statement does not hold in some cases, as shown in the following counterexample.

Example 3.1.7. Consider the graph obtained by replacing a vertex, say a, of $G = K_2$ with the cycle C_6. Note that $\mathcal{R}(G) = 1$ (by Remark 1.3.25) and $\mathcal{R}(C_6) = 2$ (by Theorem 1.3.27). But $\mathcal{R}(G_a[C_6]) \neq 2$. In fact, $G_a[C_6]$ is the wheel W_6 and its representation number is 3 (by Theorem 1.3.39), as $\mathcal{R}^p(C_6) = 3$ (by Theorem 1.3.46(i)).

Based on the original proof in [Kitaev, 2013], we state the correct version of the theorem on the representation number of $G_a[H]$ in the following.

Theorem 3.1.8. *Let G be a word-representable graph and H be a comparability graph. If $\mathcal{R}(G) = k$ and $\mathcal{R}^p(H) = k'$, then $\mathcal{R}(G_a[H]) = \max\{k, k'\}$.*

Corollary 3.1.9. *Suppose the graph H' is obtained from a comparability graph H by adding an all-adjacent vertex. Then, H' is word-representable and $\mathcal{R}(H') = \mathcal{R}^p(H)$.*

Proof. It can be observed that the graph H' is obtained from the complete graph K_2 by replacing one of its two vertices with the module H. Since complete graphs are word-representable, by Theorem 3.1.6, we have H' is a word-representable graph. Further, since $\mathcal{R}(K_2) = 1$, by Theorem 3.1.8, we have $\mathcal{R}(H') = \mathcal{R}^p(H)$. □

In fact, the graph H' in Corollary 3.1.9 is a comparability graph and $\mathcal{R}^p(H') = \mathcal{R}^p(H)$, as described in the following remark.

Remark 3.1.10. Let the graph H' be obtained from the comparability graph H by adding the all-adjacent vertex b and $\mathcal{R}^p(H) = k$. Suppose the word $p_1p_2\cdots p_k$ represents the graph H, where each p_i $(1 \le i \le k)$ is a permutation on the vertices of H. Then, the word $bp_1bp_2\cdots bp_k$ permutationally represents the graph H' so that H' is a comparability graph and $\mathcal{R}^p(H') \le k$. Since H is an induced subgraph of H' and $\mathcal{R}^p(H) = k$, we have $\mathcal{R}^p(H') \ge k$ so that $\mathcal{R}^p(H') = k$.

Using split recomposition, we now show that the sufficient condition given in Theorem 3.1.6 by Kitaev [2013] is indeed a characteristic property for the word-representability of the graph $G_a[H]$.

Theorem 3.1.11. *The graph $G_a[H]$ is word-representable if and only if G is a word-representable graph and H is a comparability graph.*

Proof. Note that by Remark 3.1.3, we have $G_a[H] = G\ {}_a\circledast_b\ H'$, where the graph H' is obtained from H by adding an all-adjacent vertex b. Thus, by Remark 1.4.14 and Theorem 2.1.2, we have $G_a[H]$ is word-representable if and only if both G and H' are word-representable graphs. Further, by Theorem 1.3.39, H' is word-representable if and only if H is a comparability graph. Thus, the result follows. □

In the following theorem, we provide a characterization for $G_a[H]$ to be a comparability graph and give its *prn*.

Theorem 3.1.12. *The graph $G_a[H]$ is a comparability graph if and only if G and H are comparability graphs. Moreover, if $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(H) = k'$, then $\mathcal{R}^p(G_a[H]) = \max\{k, k'\}$.*

Proof. Suppose $G_a[H]$ is a comparability graph. Since G and H are isomorphic to certain induced subgraphs of $G_a[H]$, G and H are comparability graphs. Conversely, suppose G and H are comparability graphs. Note that by Remark 3.1.3, $G_a[H] = G\ {}_a\circledast_b\ H'$, where the graph H' is obtained by adding an all-adjacent vertex b to H. Since H is a comparability graph, by Remark 3.1.10, H' is also a comparability graph and $\mathcal{R}^p(H') = \mathcal{R}^p(H)$. As $G_a[H] = G\ {}_a\circledast_b\ H'$ and b is an all-adjacent vertex of H', in view of Theorem 2.2.17, we have $G_a[H]$ is a comparability graph and $\mathcal{R}^p(G_a[H]) = \max\{k, k'\}$. □

Alternative proofs for Theorem 3.1.11 and Theorem 3.1.12 can be found in [Dwary and Krishna, 2025]. It can be observed that, using split decomposition, we are able to provide shorter proofs of these theorems here. Using these results, we now determine the representation number as well as the *prn* of 3-leaf power graphs.

Definition 3.1.13 (3-leaf power graph)**.** A graph $G = (V, E)$ is called a 3-*leaf power graph* if there is a tree T with V as its leaves such that for all $a, b \in V$, $\overline{ab} \in E$ if and only if their distance in T is at most three.

These graphs were introduced in [Nishimura et al., 2002], and a forbidden induced subgraph characterization for this class of graphs was obtained in [Dom et al., 2004]. Further, we have the following characterization of 3-leaf power graphs, which we will use in the sequel.

Theorem 3.1.14 ([Brandstädt and Le, 2006])**.** *A connected graph G is a 3-leaf power graph if and only if it is obtained from a suitable tree, say T_G, by replacing each vertex of T_G with a clique. The tree T_G is called an associated tree of G.*

Remark 3.1.15. Since both trees and cliques are comparability graphs, in view of Theorems 3.1.12 and 3.1.14, it is evident that 3-leaf power graphs are comparability graphs, and hence they are word-representable.

Furthermore, we have the following theorem on the representation number as well as on the *prn* of a 3-leaf power graph.

Theorem 3.1.16. *If G is a 3-leaf power graph, then $\mathcal{R}(G) \leq 2$ and $\mathcal{R}^p(G) \leq 3$. Moreover, $\mathcal{R}^p(G) = 3$ if and only if the graph S (depicted in Figure 1.9) is isomorphic to an induced subgraph of G.*

Proof. Note that by Theorem 3.1.14, a 3-leaf power graph G is obtained from an associated tree T_G by replacing iteratively each vertex of T_G with a clique. Let $a_1, a_2, \ldots, a_n$ be the vertices of T_G and $M_1, M_2, \ldots, M_n$ be the cliques replacing $a_1, a_2, \ldots, a_n$, respectively, to obtain G. Recall that $\mathcal{R}^p(M_i) = 1$, for all $1 \leq i \leq n$ (by Remark 1.3.42). Hence, in view of Theorem 3.1.8, we have

$$\mathcal{R}(G) = \max\{\mathcal{R}(T_G), \mathcal{R}^p(M_1), \mathcal{R}^p(M_2), \ldots, \mathcal{R}^p(M_n)\} = \mathcal{R}(T_G).$$

Similarly, in view of Theorem 3.1.12, we have $\mathcal{R}^p(G) = \mathcal{R}^p(T_G)$. It is known that the representation number and the *prn* of trees are at most 2 and 3, respectively (by Theorem 1.3.26 and Theorem 1.3.46(iii)). Hence, $\mathcal{R}(T_G) \leq 2$ and $\mathcal{R}^p(T_G) \leq 3$ so that $\mathcal{R}(G) \leq 2$ and $\mathcal{R}^p(G) \leq 3$.

Suppose $\mathcal{R}^p(G) = 3$ so that we have $\mathcal{R}^p(T_G) = 3$. In view of Theorem 1.3.46(iii), the graph S is isomorphic to an induced subgraph of T_G. Further, since T_G is an induced subgraph of G, we have S is isomorphic to an induced subgraph of G. The converse part holds as $\mathcal{R}^p(S) = 3$ (by Theorem 1.3.46(iii)). □

As a consequence of the above results, we now study the word-representability of the lexicographical product of graphs.

Definition 3.1.17 (Lexicographical product of graphs)**.** Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. The *lexicographical product of G and G'*, denoted by $G \circ G'$, defined by $G \circ G' = (V'', E'')$, where the vertex set $V'' = V \times V'$ and the edge set

$$E'' = \left\{\overline{(a, a')(b, b')} \mid \overline{ab} \in E \text{ or } (a = b,\ \overline{a'b'} \in E')\right\}.$$

The word-representability of lexicographical product of graphs is an open problem posed in [Kitaev and Lozin, 2015, Chapter 7]. In connection to this problem, the following result was shown in [Choi et al., 2019, Section 6] by constructing an explicit example.

Proposition 3.1.18 ([Choi et al., 2019])**.** *The class of word-representable graphs is not closed under the lexicographical product.*

First, we observe the relation between lexicographical product and the operation of module replacing a vertex as per the following remark.

Remark 3.1.19. The lexicographical product $G \circ G'$ is nothing but replacing each vertex of G by the module G'. Hence, both G and G' are induced subgraphs of $G \circ G'$.

Accordingly, in the following theorem, we provide a necessary and sufficient condition for $G \circ G'$ to be a word-representable graph. Thus, we settle the above-mentioned open problem.

Theorem 3.1.20. *Let G and G' be two graphs. The lexicographical product $G \circ G'$ is word-representable if and only if G is word-representable and G' is a comparability graph. Moreover, if $\mathcal{R}(G) = k$ and $\mathcal{R}^p(G') = k'$, then $\mathcal{R}(G \circ G') = \max\{k, k'\}$.*

Proof. The proof follows from Remark 3.1.19, Theorem 3.1.8 and Theorem 3.1.11. □

We further state a characteristic property for $G \circ G'$ to be a comparability graph.

Theorem 3.1.21. *Let G and G' be two graphs. The lexicographical product $G \circ G'$ is a comparability graph if and only if G and G' are comparability graphs. Moreover, if $\mathcal{R}^p(G) = k$ and $\mathcal{R}^p(G') = k'$, then $\mathcal{R}^p(G \circ G') = \max\{k, k'\}$.*

We are now ready to present the main result of this chapter on characterization of word-representable graphs with respect to the modular decomposition, through the following lemma.

Lemma 3.1.22. *Let G be a word-representable graph and A be any non-trivial module of G. Then the induced subgraph $G[A]$ is a comparability graph.*

Proof. Consider the graph G' which is obtained from G by replacing the module A by a new vertex, say a. Thus, G' is word-representable as it is isomorphic to an induced subgraph of G. Note that G can be reconstructed from G' by replacing the vertex a with the module $G[A]$. Hence, by Theorem 3.1.11, we have $G[A]$ is a comparability graph. □

Theorem 3.1.23. *Let G be an M-decomposable graph and let $\mathcal{P} = \{A_1, A_2, \ldots, A_k\}$ be a modular partition of G, for some positive integer $k \geq 2$. Then, we have the following:*

1. *G is word-representable if and only if for each $1 \leq i \leq k$, $G[A_i]$ is a comparability graph and $G/\mathcal{P}$ is a word-representable graph.*

2. *If G is word-representable, then*

$$\mathcal{R}(G) = \max\{\mathcal{R}(G/\mathcal{P}), \mathcal{R}^p(G[A_1]), \ldots, \mathcal{R}^p(G[A_k])\}.$$

3. *If G is a comparability graph, then*

$$\mathcal{R}^p(G) = \max\{\mathcal{R}^p(G/\mathcal{P}), \mathcal{R}^p(G[A_1]), \ldots, \mathcal{R}^p(G[A_k])\}.$$

Proof.

1. Suppose G is a word-representable graph. As $G/\mathcal{P}$ is an induced subgraph of G, it is a word-representable graph. Further, from Lemma 3.1.22, we have $G[A_i]$ is a comparability graph, for each $1 \leq i \leq k$. Note that the converse is evident from Remark 1.5.7 and Theorem 3.1.11.

2. If G is a word-representable graph, then from part (1) we have each $G[A_i]$ is a comparability graph and $G/\mathcal{P}$ is a word-representable graph. Hence, the result follows from Remark 1.5.7 and Theorem 3.1.8.

3. The result is evident from Remark 1.5.7, Theorem 1.5.14 and Theorem 3.1.12.

□

3.2 Conclusion

In this chapter, we characterized word-representable graphs with respect to the modular decomposition. Furthermore, we have established the word-representability of 3-leaf power graphs and lexicographical product of graphs and obtained their representation numbers and also their *prn*. It is interesting to find further special classes of word-representable graphs using modular decomposition.

While recognizing word-representability of graphs is NP-complete [Halldórsson et al., 2011], it is interesting to study methods to prove the non-word-representability of a graph. To explore more in this direction, one may refer to [Kitaev and Sun, 2024]. In [McConnell and Spinrad, 1999], it was shown that the recognition of comparability graphs as well as the computation of modular decomposition of a graph can be done in $O(n+m)$ time, where n and m are the number of vertices and the number of edges of the graph, respectively. Hence, in view of Theorem 3.1.23, we have a polynomial (in $m+n$) time test for non-word-representability of a graph G: Find the maximal modular partition of G in $O(m+n)$ time. Note that there are at most n modules in the partition. Then check the subgraph induced by each of these modules in $O(m+n)$ time whether they are comparability graphs. If a non-comparability module is found, G is not a word-representable graph. However, if all modules are comparability graphs, then the test gives no information. In that case, the problem is reduced to the problem of recognizing the word-representability of the associated quotient graph.

4

Well-Partitioned Chordal Graphs

In this chapter, we show that every component of the minimal split decomposition of a well-partitioned chordal (WPC) graph is a split graph, which leads to a characterization of word-representable WPC graphs. Based on this characterization, we show that the recognition of word-representability of WPC graphs can be done in polynomial time. Furthermore, we determine the representation number of a word-representable WPC graph as well as that of a word-representable split graph. We also obtain a minimal forbidden induced subgraph characterization of circle graphs restricted to WPC graphs, and use it to completely classify the word-representable WPC graphs with respect to their representation numbers. Moreover, we characterize WPC comparability graphs and study their permutation-representation numbers.

4.1 Characterizations

In this section, we characterize word-representable WPC graphs. Further, we provide characterizations for circle graphs and comparability graphs, within WPC graphs.

4.1.1 Word-Representable Graphs

In this subsection, we first observe that not all WPC graphs are word-representable and accordingly, we provide a characterization for word-representable WPC graphs.

Remark 4.1.1. Since the class of split graphs is a subclass of the class of WPC graphs (cf. Remark 1.6.41) and not all split graphs are word-representable (cf. Remark 1.6.33), it is evident that not all WPC graphs are word-representable.

We characterize word-representable WPC graphs through the following lemma.

Lemma 4.1.2. *Let G be a WPC graph and $\mathcal{H}$ be its minimal split decomposition. Then every split component of $\mathcal{H}$ is a split graph.*

Proof. First note that each split component in $\mathcal{H}$ is connected (cf. Remark 1.4.5). Since $\mathcal{H}$ is the minimal split decomposition of G, the split components in $\mathcal{H}$ are cliques, stars, and prime graphs. As stars and cliques are split graphs, it is sufficient to prove that every prime component of $\mathcal{H}$ is a split graph. Let $H = (V, E)$ be a prime component of $\mathcal{H}$ such that it is neither a clique nor a star. Then, by Remark 1.4.2, $|V| \geq 4$. Since H is isomorphic to an induced subgraph of G, by Proposition 1.6.45, we have H is also a WPC graph. Thus, by Definition 1.6.37, there exist a partition $\mathcal{P}$ of V and a partition tree $\mathcal{T} = (\mathcal{P}, E')$.

Suppose each bag of the partition $\mathcal{P}$ is of size one. Then, the WPC graph H must be a tree. Since H is not a star, there is an edge, say e, in H with the endpoints a_1 and a_2 such that neither a_1 nor a_2 is a leaf. Consider the graph $H \setminus \{e\}$ obtained by removing the edge e from H and let H_1 and H_2 be two components of $H \setminus \{e\}$

containing a_1 and a_2, respectively. Thus, the vertex set V can be partitioned into V_1 and V_2 consisting of the vertices appearing in H_1 and H_2, respectively. It can be observed that $|V_1| \geq 2$, $|V_2| \geq 2$, $N_H(V_1) = \{a_2\}$ and $N_H(V_2) = \{a_1\}$. Hence, the partition $\{V_1, V_2\}$ forms a split in H. This contradicts the assumption that H is a prime graph. Thus, there exists a bag B in $\mathcal{P}$ of size at least two.

We now observe that the bag B cannot be adjacent to any bag of size at least two in $\mathcal{T}$. Assume that $\overline{BB'} \in E'$ for some bag B' of size at least two. Consider the graph $\mathcal{T} \setminus \overline{BB'}$ obtained by deleting the edge $\overline{BB'}$ from $\mathcal{T}$ and let $\mathcal{T}^B$ and $\mathcal{T}^{B'}$ be the two components of $\mathcal{T} \setminus \overline{BB'}$ containing B and B', respectively. Thus, the vertex set V could be partitioned into V^B and $V^{B'}$ consisting of the vertices appeared in $\mathcal{T}^B$ and $\mathcal{T}^{B'}$, respectively. Since both B and B' have at least two vertices, we have $|V^B| \geq 2$ and $|V^{B'}| \geq 2$. Further, by Remark 1.6.44, $N_H(V^B) = bd(B', B)$ and $N_H(V^{B'}) = bd(B, B')$. In view of Remark 1.6.43, we have $N_H(V^B)$ is complete to $N_H(V^{B'})$ in H so that the partition $\{V^B, V^{B'}\}$ forms a split of H. This leads to a contradiction, as H is a prime graph. Thus, any bag adjacent to B in $\mathcal{T}$ is of size one.

We conclude the proof by showing that $\mathcal{T}$ is a star centered at B so that H is a split graph by Remark 1.6.41. We achieve this by proving that any bag B' adjacent to B in $\mathcal{T}$ is a leaf of $\mathcal{T}$. Suppose there is another bag B'' adjacent to B' in $\mathcal{T}$. As done in the previous paragraph, partition the vertex set V into V^B and $V^{B'}$ consisting of the vertices appeared in the components of $\mathcal{T} \setminus \overline{BB'}$ containing B and B', respectively. Note that $|V^B| \geq 2$ and, as $B' \cup B'' \subseteq V^{B'}$, we have $|V^{B'}| \geq 2$. Thus, in view of Remarks 1.6.43 and 1.6.44, $\{V^B, V^{B'}\}$ forms a split of H. Again this leads to a contradiction, as H is a prime graph. Thus, B' is a leaf of $\mathcal{T}$. $\square$

Corollary 4.1.3. *If a WPC graph G is a prime graph, then G is a split graph.*

Remark 4.1.4. Note that the converse of Lemma 4.1.2 is not true. For instance, the minimal split decomposition of the cycle C_4 has two split components and both split components are the star S_2 such that the centers of the stars are the marked vertices. However, C_4 is not a chordal graph, and hence it is not a WPC graph.

Theorem 4.1.5. *Let G be a WPC graph and $\mathcal{H}$ be its minimal split decomposition. Then, G is word-representable if and only if all components of $\mathcal{H}$ are word-representable split graphs.*

Proof. In view of Theorem 2.1.4, G is word-representable if and only if all components of $\mathcal{H}$ are word-representable. Further, by Lemma 4.1.2, all components of $\mathcal{H}$ are split graphs. Hence, the theorem follows. □

Thus, Theorem 4.1.5 in conjunction with Theorem 1.6.35 provides us a complete characterization of word-representable WPC graphs.

4.1.2 Circle Graphs

The forbidden induced subgraph characterization of circle graphs is an open problem in the literature. However, in [Bonomo-Braberman et al., 2022], circle graphs when restricted to split graphs are characterized in terms of the forbidden induced subgraphs. In this subsection, we prove that the set of forbidden induced subgraphs of circle graphs, when restricted to split graphs, remains the same when restricted to WPC graphs. This is useful for classification of WPC graphs with respect to their representation numbers in Section 4.3. Recall that, by Theorem 1.6.32, the set $\mathcal{C}$ (given in Figure 1.20) consists of forbidden induced subgraphs for circle graphs within split graphs. We now prove the following theorem for WPC graphs.

Theorem 4.1.6. *Let G be a WPC graph. Then, G is a circle graph if and only if G is a $\mathcal{C}$-free graph.*

Proof. Note that by Theorem 1.6.32, each graph belonging to the set $\mathcal{C}$ is not a circle graph. Since every induced subgraph of a circle graph is a circle graph, if G is a circle graph, then G is $\mathcal{C}$-free. Conversely, suppose that the WPC graph G is a $\mathcal{C}$-free graph. We prove that G is a circle graph. On the contrary, suppose that G is not a circle graph. Let H_i $(1 \leq i \leq k)$ be the components of the minimal split decomposition of G. Then, by Corollary 2.1.5, there exists at least one component, say H_t, such that H_t is not a circle graph. Further, since H_t is a split graph (by Lemma 4.1.2), and H_t is not a circle graph, in view of Theorem 1.6.32, H_t must contains at least one graph, say F, from the set $\mathcal{C}$ as an induced subgraph. Since H_t is an induced subgraph of the graph G, we have F is an induced subgraph of G. This contradicts the assumption that G is a $\mathcal{C}$-free graph. Hence, G is a circle graph. □

Remark 4.1.7. Since each graph belonging to the set $\mathcal{C}$ is a minimally non-circle graph (see Theorem 1.6.32), Theorem 4.1.6 provides us a minimal forbidden induced subgraph characterization of circle graphs restricted to WPC graphs.

Proposition 4.1.8. *Each graph belonging to the set $\mathcal{C}$ is a prime graph.*

Proof. For $G \in \mathcal{C}$, if G is not a prime graph, then G can be decomposed using split decomposition. Let H_i, $2 \leq i \leq k$, be the components of a split decomposition of G. Note that each H_i is a proper induced subgraph of G. Thus, by Theorem 1.6.32, each H_i is a circle graph as G is a minimally non-circle graph. Then, by Corollary 2.1.5, we have G is a circle graph; a contradiction. □

4.1.3 Comparability Graphs

In this subsection, we provide a forbidden induced subgraph characterization of WPC comparability graphs. The minimal forbidden induced subgraph characterization for the comparability graphs was provided by Gallai [1967] as per the following result.

Theorem 4.1.9 ([Gallai, 1967])**.** *A graph G is a comparability graph if and only if G is $\mathcal{N}$-free, where the set of graphs $\mathcal{N}$ is given in two parts through Figure 4.1 and*

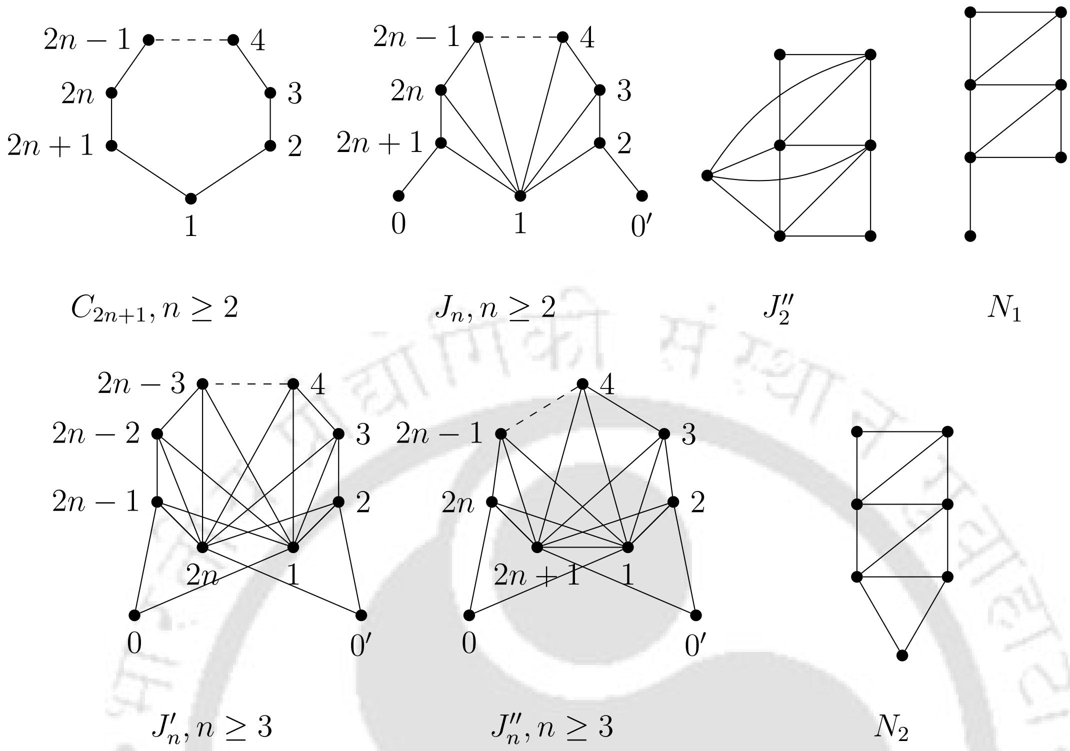


FIGURE 4.1: Part 1 of $\mathcal{N}$

Figure 4.2 as described in the following. One part of $\mathcal{N}$ is given in Figure 4.1, and the complements of the rest of the graphs of $\mathcal{N}$ are given in Figure 4.2.

Notation 4.1.10. Let $\mathcal{W}$ denote the set of graphs consisting of J_2, $\overline{M_2}$, $\overline{L_1}$ and $\overline{L_2}$ from $\mathcal{N}$, where $\overline{G}$ denotes the complement of a graph G.

Lemma 4.1.11. *The set of WPC graphs in $\mathcal{N}$ is $\mathcal{W}$.*

Proof. Note that the graphs C_{2n+1}, for $n \geq 2$, belong to the set $\mathbb{O}$ (see Figure 1.22), where $\mathbb{O}$ is the set of forbidden induced subgraphs of WPC graphs. Further, the graphs J_2'', N_1, and N_2 (depicted in Figure 4.1) contain the graph O_1 (see Figure 1.22) as an induced subgraph. Thus, in view of Theorem 1.6.46, the graphs $C_{2n+1}(n \geq 2)$, J_2'', N_1 and N_2 are not WPC graphs.

Observe that the graph J_2 is a WPC graph with the bags $\{1, 3, 4\}, \{0', 2\}$, and $\{0, 5\}$ of its partition tree. However, for $n \geq 3$, the graph J_n contains O_2 (see

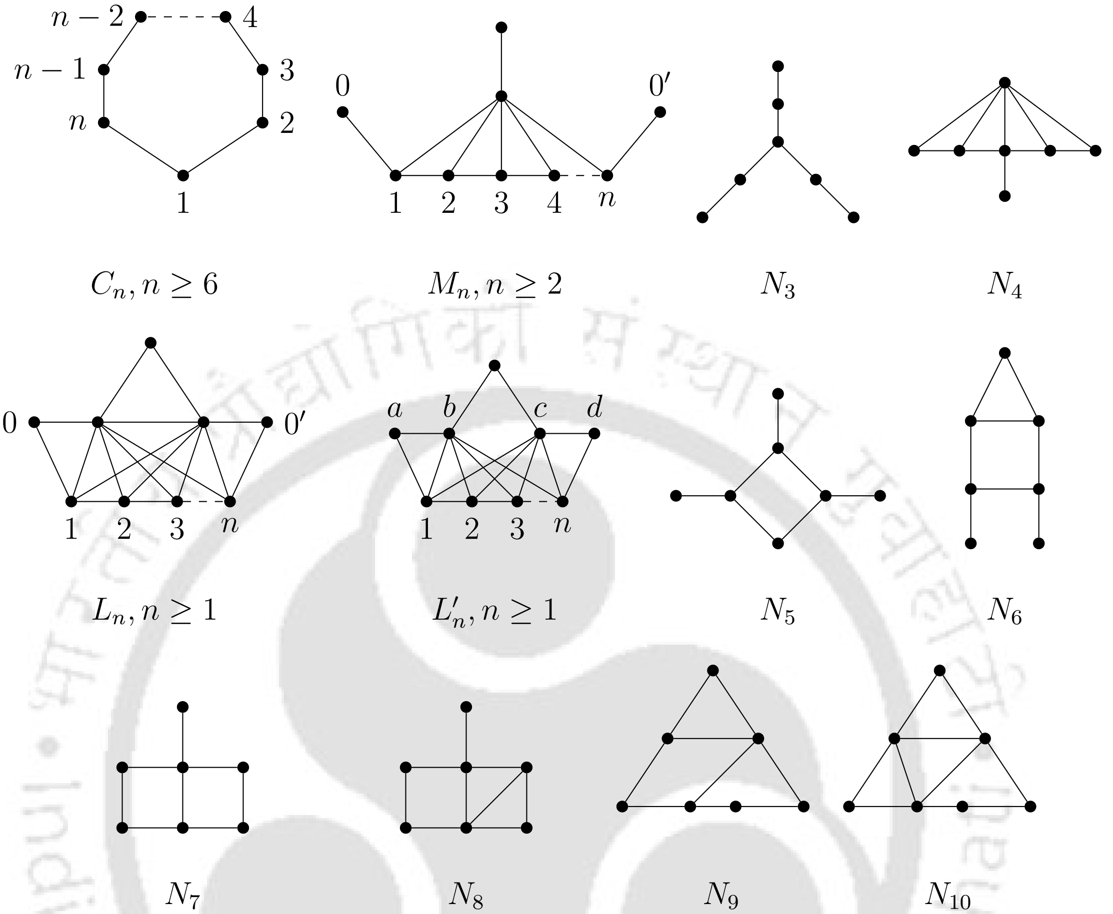
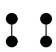


FIGURE 4.2: The complements of Part 2 of $\mathcal{N}$

Figure 1.22) as an induced subgraph induced by the vertex set $\{1, 3, 4, 5, 6, 7\}$. Thus, in view of Theorem 1.6.46, J_n is not a WPC graph for any $n \geq 3$.

For $n \geq 3$, the graphs J'_n and J''_n contain O_2 as an induced subgraph induced by the sets $\{2, 3, 4, 5, 0', 2n\}$ and $\{2, 3, 4, 5, 0', 2n+1\}$, respectively. Thus, for $n \geq 3$, J'_n and J''_n are not WPC graphs.

Note that the complement of M_2, L_1 and L_2 are nothing but the graphs D_2, D_1 and D_3 (see Figure 1.19), respectively. Since D_1, D_2, D_3 are split graphs, they are WPC graphs. However, for $n \geq 3$, the graphs M_n and L_n contain $2K_2$, whose depiction is , as induced subgraphs induced by the vertex set $\{0, 1, n, 0'\}$. Thus, the complement of each of these graphs contains C_4 (which is not a chordal graph) as an induced subgraph so that it is not a WPC graph.

Further, it can be observed that all the remaining graphs, namely C_n (for $n \geq 6$), L'_n (for $n \geq 1$), $N_3, N_4, N_5, N_6, N_7, N_8, N_9$ and N_{10} (see Figure 4.2) contain $2K_2$ as an induced subgraph. Thus, their complements are not WPC graphs. □

Theorem 4.1.12. *Let G be a WPC graph. Then, G is a comparability graph if and only if G is $\mathcal{W}$-free.*

Proof. If G is a comparability graph, then G is $\mathcal{W}$-free, because each graph in the set $\mathcal{W}$ is not a comparability graph and every induced subgraph of a comparability graph is a comparability graph. Conversely, suppose G is $\mathcal{W}$-free. On the contrary, if G is not a comparability graph, then G contains one of the graphs from the set $\mathcal{N}$ as an induced subgraph. Since every induced subgraph of a WPC graph is also a WPC graph and any WPC graph in $\mathcal{N}$ must belong to $\mathcal{W}$ (by Lemma 4.1.11), G cannot be $\mathcal{W}$-free; a contradiction. Hence, G is a comparability graph. □

4.2 Recognition Algorithm

In this section, using Theorem 4.1.5, we give a polynomial-time algorithm for the following recognition problem.

Problem

Given a WPC graph G, is G word-representable?

Algorithm

We propose Algorithm 1 based on the algorithms for split decomposition (see [Charbit et al., 2012; Dahlhaus, 1994]) and recognition of word-representable split graphs (see [Kitaev and Pyatkin, 2024]). In view of the characterization given in Theorem 4.1.5, in Algorithm 1, we decompose a given WPC graph G using split decomposition and verify whether all the split components are word-representable or not. If one of the components is not word-representable, then G is not word-representable.

Algorithm 1: Recognizing word-representability of WPC graphs.

Input: A WPC graph G.
Output: Yes, if G is word-representable; otherwise, No.

```
1 Compute the minimal split decomposition 𝓗 of G. Let H_i, 1 ≤ i ≤ k, be
    the components of 𝓗.
2 Flag = Yes.
3 for each H_i ∈ 𝓗 do
4     if H_i is not a word-representable graph then
5         Flag = No.
6     end
7 end
8 return Flag
```

Time Complexity

In Algorithm 1, the minimal split decomposition can be computed in linear time as per Theorem 1.4.9 (step 1). Further, in view of Remark 1.4.10, the number of components of the minimal split decomposition are polynomially bounded with respect to the size of the input graph. Since every H_i is a split graph (by Lemma 4.1.2), by Theorem 1.6.36, the word-representability of each H_i can be verified in polynomial time (step 4) on the size of H_i. Thus, testing all the components (step 3) takes polynomial time on the size of the input graph. Hence, we have the following theorem.

Theorem 4.2.1. *The word-representability of a WPC graph can be decided in polynomial time.*

4.3 Representation Number

In this section, we focus on determining the representation number of a word-representable WPC graph. In view of Theorem 2.1.4, note that the representation number of a word-representable WPC graph is the maximum of the representation numbers of the split components of its minimal split decomposition. Further, by Lemma 4.1.2, it is evident that these split components are nothing but word-representable split graphs. In this connection, in Subsection 4.3.1, we first prove that the representation number of a word-representable split graph is at most three. Accordingly, we obtain the representation number of a word-representable WPC graph. Further, in Subsection 4.3.2, we completely classify split graphs as well as WPC graphs with respect to their representation numbers.

4.3.1 Construction of Word-Representant

In this subsection, first we devise a polynomial-time algorithm to construct a 3-uniform word-representant of a word-representable split graph so that its representation number is at most three. The algorithm is based on the vertex labeling of word-representable split graphs given in Theorem 1.6.35. Consequently, we prove that the representation number of a word-representable WPC graph is at most three.

Notation 4.3.1. Let $G = (I \cup C, E)$ be a word-representable split graph. The vertices of C are labeled from 1 to $k = |C|$ such that the labeling satisfies the properties given in Theorem 1.6.35. We now consider the following sets:

$$A = \{a \in I \mid N(a) = [1, m] \cup [n, k], \text{ for some } m < n\}$$
$$B = \{a \in I \mid N(a) = [l, r], \text{ for some } l \leq r\}$$

Remark 4.3.2 ([Kitaev and Pyatkin, 2024]). The above-mentioned labeling of the vertices of C can be found in polynomial time.

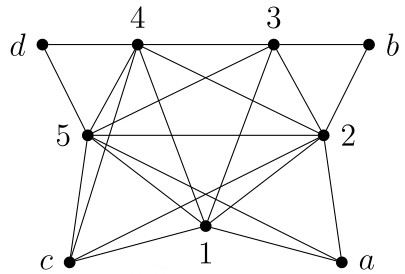


Figure 4.3: A word-representable split graph

Notation 4.3.3. By Theorem 1.6.35, we have $A \cap B = \varnothing$ and $I = A \cup B$. For $a \in A$, let $N(a) = [1, m_a] \cup [n_a, k]$ and for $a \in B$, let $N(a) = [l_a, r_a]$.

Remark 4.3.4. By Theorem 1.6.35 (iii), since $m_a < n_b$ for any $a, b \in A$, we have

$$\max\{m_a \mid a \in A\} < \min\{n_a \mid a \in A\}.$$

Given a split graph $G = (I \cup C, E)$ with a semi-transitive orientation (and hence, a labeling of the vertices of C), Algorithm 2 constructs a 3-uniform word-representant of G. In the following example, we employ Algorithm 2 on a word-representable split graph and present the output.

Example 4.3.5. Consider the word-representable split graph shown in Figure 4.3, say $G = (A \cup B \cup C, E)$, where $A = \{a, c\}$, $B = \{b, d\}$ and $C = \{1, 2, 3, 4, 5\}$. Note that the vertices of C are labeled such that the labeling satisfies the criteria of Theorem 1.6.35. Given G as input, Algorithm 2 produces the 3-uniform word $1b2ac3d45db123bc4a5d12ca345$ as the output.

Remark 4.3.6. Algorithm 2 runs in polynomial time.

Notation 4.3.7. For the rest of the chapter, w refers to the output of Algorithm 2.

Algorithm 2: Constructing a 3-uniform word-representant of a split graph.

Input: A word-representable split graph $G = (I \cup C, E)$ with $1, \ldots, k$ as the labels of the vertices of C.

Output: A 3-uniform word w representing G.

Initialize p_1, p_2 and p_3 with the permutation $12 \cdots k$ and update them with the vertices of I, as per the following.
Initialize $d = 1$.
for $a \in I$ **do**
    if $a \in A$ **then**
        Let $N(a) = [1, m_a] \cup [n_a, k]$.
        if $m_a > d$ **then**
            Assign $d = m_a$.
        end
        Replace m_a in p_1 with $m_a a$.
        Replace n_a in p_2 with $a n_a$.
    end
    else
        Let $N(a) = [l_a, r_a]$.
        Replace l_a in p_1 with $a l_a$.
        Replace r_a in p_2 with $r_a a$.
    end
end
Replace d in p_3 with $d(p_1|_A)^R$. ▷ Note that $d = \max\{m_a \mid a \in A\}$
return the 3-uniform word $p_1(p_1|_B)^R p_2 p_3$

Remark 4.3.8. In Algorithm 2, note that the order of the vertices of C, i.e., $12 \cdots k$, in the words p_1, p_2, and p_3 do not alter when they are updated. Hence, we have $w|_C = 12 \cdots k 12 \cdots k 12 \cdots k$.

Remark 4.3.9. At the end of the algorithm, both the words p_1 and p_2 are permutations on the set V, while the word p_3 is a permutation only on $C \cup A$. Observe that the elements of B do not occur in p_3.

Through the following lemmas, we prove that the word w is a word-representant of the word-representable split graph G.

Lemma 4.3.10. *If $a \in A$, then $m_a a n_a$ is a subword of each of p_1, p_2, and p_3.*

Proof. Note that $N(a) = [1, m_a] \cup [n_a, k]$, for $m_a < n_a$. Thus, as per line 1 of Algorithm 2, we have $m_a n_a \ll p_1$, $m_a n_a \ll p_2$, and $m_a n_a \ll p_3$. As $a \in A$, replacing m_a in p_1 with $m_a a$ (see line 9) and replacing n_a in p_2 with $a n_a$ (see line 10), we have $m_a a n_a \ll p_1$ and $m_a a n_a \ll p_2$, respectively. Further, in view of Remark 4.3.4, we have $m_a \leq d < n_a$ so that we have $m_a d n_a \ll p_3$. Now, as per line 18 of Algorithm 2, we have $m_a a n_a \ll m_a d a n_a \ll p_3$. □

Lemma 4.3.11. *For $a, b \in V$, if a and b are adjacent in G, then a and b alternate in w.*

Proof. As I is an independent set in G, note that both a and b cannot be in I. Suppose both $a, b \in C$. As the vertices of C are labeled from 1 to k, we have $a = i$ and $b = j$, for some $1 \leq i, j \leq k$. Thus, in view of Remark 4.3.8, we have $w|_{\{a,b\}} = ababab$, if $i < j$, or $w|_{\{a,b\}} = bababa$, if $j < i$. Hence, in this case, a alternates with b in w. If $a \in I$ and $b \in C$, then we consider the following cases:

- Case 1: $a \in A$. Note that $b \in N(a) = [1, m_a] \cup [n_a, k]$, for $m_a < n_a$. If $b \in [1, m_a]$, as per line 1 of Algorithm 2, we have $b m_a \ll p_1$, $b m_a \ll p_2$, and $b m_a \ll p_3$. Hence, in view of Lemma 4.3.10, we have $b m_a a n_a \ll p_1$, $b m_a a n_a \ll p_2$, and $b m_a a n_a \ll p_3$. As $p_1 p_2 p_3 \ll w$ and w is a 3-uniform word, we have $w|_{\{a,b\}} = bababa$. Thus, a and b alternate in w. Similarly, if $b \in [n_a, k]$, we can observe that $m_a a n_a b \ll p_1$, $m_a a n_a b \ll p_2$, and $m_a a n_a b \ll p_3$. Hence, $w|_{\{a,b\}} = ababab$ so that a and b alternate in w.

- Case 2: $a \in B$. Note that $b \in N(a) = [l_a, r_a]$, for $l_a \leq r_a$. Thus, by line 1 of Algorithm 2, we have $l_a b r_a \ll p_1$, $l_a b r_a \ll p_2$, and $l_a b r_a \ll p_3$. As $a \in B$, replacing l_a in p_1 with $a l_a$ (see line 14), and replacing r_a in p_2 with $r_a a$ (see line 15), we have $a l_a b r_a \ll p_1$ and $l_a b r_a a \ll p_2$, respectively. Finally, as $w = p_1 (p_1|_B)^R p_2 p_3$, we have $w|_{\{a,b\}} = ababab$. Thus, a and b alternate in w.

Thus, in any case, a and b alternate in w. □

Lemma 4.3.12. *If $a, b \in I$, then a and b do not alternate in w.*

Proof. Since $I = A \cup B$, we consider the following cases.

- Case 1: $a, b \in A$. Since both a and b appear in p_1, without loss of generality, suppose $ab \ll p_1$. Since $(p_1|_A)^R \ll p_3$ (by line 18 of Algorithm 2), we have $ba \ll p_3$. Thus, if $ab \ll p_2$, we have $ababba \ll p_1p_2p_3 \ll w$, and if $ba \ll p_2$, we have $abbaba \ll p_1p_2p_3 \ll w$. As w is a 3-uniform word, in any case, we can see that a and b do not alternate in w.

- Case 2: $a, b \in B$. Since both a and b appear in p_1, without loss of generality, suppose $ab \ll p_1$. Then, we have $ba \ll (p_1|_B)^R$. Since $w = p_1(p_1|_B)^R p_2p_3$, in view of Remark 4.3.9, $abba$ is a factor of $w|_{\{a,b\}}$. Hence, a and b do not alternate in w.

- Case 3: $a \in A$ and $b \in B$. Note that $N(a) = [1, m_a] \cup [n_a, k]$, for $m_a < n_a$, and $N(b) = [l_b, r_b]$, for $l_b \leq r_b$. Then, from Theorem 1.6.35 (ii), we have $l_b > m_a$ or $r_b < n_a$. Accordingly, we consider the following cases.

- Subcase 3.1: $l_b > m_a$. From line 1 of Algorithm 2, we have $m_al_b \ll p_1$. As $a \in A$, replacing m_a in p_1 with m_aa we have, $m_aal_b \ll p_1$ (see line 9). Further, as $b \in B$, replacing l_b in p_1 with bl_b we have, $m_aabl_b \ll p_1$ (see line 14) so that $ab \ll p_1$. Since $w = p_1(p_1|_B)^R p_2p_3$, we have $abbp_2p_3 \ll w$. Thus, in view of Remark 4.3.9, abb is a factor of $w|_{\{a,b\}}$. Hence, a and b do not alternate in w.

- Subcase 3.2: $r_b < n_a$. From line 1 of Algorithm 2, we have $r_bn_a \ll p_2$. As $a \in A$, replacing n_a in p_2 with an_a, we have $r_ban_a \ll p_2$ (see line 10). Further, as $b \in B$, replacing r_b in p_2 with r_bb, we have $r_bban_a \ll p_2$ (see line 15) so that $ba \ll p_2$. Since $w = p_1(p_1|_B)^R p_2p_3$, in view of Remark 4.3.9, baa is a factor of $w|_{\{a,b\}}$. Hence, a and b do not alternate in w.

Thus, in any case, a and b do not alternate in w. □

Lemma 4.3.13. *For $a \in I$ and $b \in C$, if a and b are not adjacent in G, then a and b do not alternate in w.*

Proof. As $I = A \cup B$, we consider the following cases.

- Case 1: $a \in A$. Note that $N(a) = [1, m_a] \cup [n_a, k]$, for $m_a < n_a$. As $b \in C$ and $b \notin N(a)$, we have $m_a < b < n_a$. Thus, $m_a b n_a \ll p_1$, $m_a b n_a \ll p_2$ and $m_a b n_a \ll p_3$. Accordingly, replacing m_a in p_1 with $m_a a$ (see line 9), and replacing n_a in p_2 with $a n_a$ (see line 10), we have $m_a a b n_a \ll p_1$ and $m_a b a n_a \ll p_2$, respectively. Thus, $ab \ll p_1$ and $ba \ll p_2$ so that we have $abba \ll p_1 p_2$. Further, since $w = p_1 (p_1|_B)^R p_2 p_3$, in view of Remark 4.3.9, it is evident that $abba$ is a factor of $w|_{\{a,b\}}$. Hence, a and b do not alternate in w.

- Case 2: $a \in B$. Note that $N(a) = [l_a, r_a]$, for $l_a \leq r_a$. As $b \in C$ and $b \notin N(a)$, we have $b < l_a$ or $b > r_a$. Suppose $b < l_a$. Then, we have $b l_a r_a \ll p_1$, $b l_a r_a \ll p_2$, and $b l_a r_a \ll p_3$. Accordingly, replacing l_a in p_1 with $a l_a$ (by line 14), and replacing r_a in p_2 with $r_a a$ (by line 15), we have $b a l_a r_a \ll p_1$ and $b l_a r_a a \ll p_2$, respectively. Thus, $ba \ll p_1$ and $ba \ll p_2$. Further, since $w = p_1 (p_1|_B)^R p_2 p_3$, and $a \ll (p_1|_B)^R$, $b \ll p_3$, we have $baabab \ll w$ so that a and b do not alternate in w. Similarly, if $b > r_a$, we can observe that $abaabb \ll w$ so that a and b do not alternate in w.

Thus, in any case, a and b do not alternate in w. □

Thus, we have the following result on the representation number of a word-representable split graph.

Theorem 4.3.14. *Let $G = (I \cup C, E)$ be a word-representable split graph. Then, $\mathcal{R}(G) \leq 3$.*

Proof. Note that, if a and b are non-adjacent vertices of G, then either both a, b are in I or one of them belongs to I. Thus, by Lemmas 4.3.11, 4.3.12, and 4.3.13, the 3-uniform word w represents G. Hence, $\mathcal{R}(G) \leq 3$. □

Consequently, we obtain the representation number of a word-representable WPC graph as per the following result.

Theorem 4.3.15. *If G is a word-representable WPC graph, then $\mathcal{R}(G) \leq 3$.*

Proof. Let H_i $(1 \leq i \leq k)$ be the components of the minimal split decomposition of G. Since G is a word-representable WPC graph, by Theorem 4.1.5, each of H_i is a word-representable split graph. Further, in view of Theorem 2.1.4, we have $\mathcal{R}(G) = \max\limits_{1 \leq i \leq k} \mathcal{R}(H_i)$. Since the representation number of a word-representable split graph is at most three (by Theorem 4.3.14), we have $\mathcal{R}(H_i) \leq 3$, for all $1 \leq i \leq k$, so that $\mathcal{R}(G) \leq 3$. $\square$

4.3.2 Classifications

In this subsection, we consider the word-representable graphs within the classes of split graphs, WPC graphs, split comparability graphs, and WPC comparability graphs, and completely classify each class as per the representation numbers.

In view of Theorems 1.6.32 and 4.1.6, we investigate the word-representability as well as the transitive orientability of each graph from the set $\mathcal{C}$ given in Figure 1.20, in the sequel.

Definition 4.3.16 (Even k-sun)**.** For an even integer $k \geq 4$, an *even k-sun* is the split graph $(I \cup C, E)$ with $I = \{a_1, \ldots, a_k\}$ and $C = \{c_1, \ldots, c_k\}$ such that $N(a_i) = \{c_i, c_{i+1}\}$, for each $1 \leq i \leq k-1$, and $N(a_k) = \{c_1, c_k\}$.

We now recall the result [Kitaev et al., 2021, Theorem 9] on word-representability of an even k-sun.

Lemma 4.3.17 ([Kitaev et al., 2021])**.** *For $k \geq 4$, an even k-sun is a word-representable graph.*

Definition 4.3.18 (Odd k-sun with center)**.** For an odd integer $k \geq 3$, an *odd k-sun with center* is the split graph $(I \cup C, E)$ with $I = \{a_1, \ldots, a_k\}$ and $C = \{c, c_1, \ldots, c_k\}$

such that $N(a_i) = \{c_i, c_{i+1}\}$, for each $1 \leq i \leq k-1$ and $N(a_k) = \{c_1, c_k\}$. The vertex c is called the center.

We now recall the result [Kitaev et al., 2021, Theorem 11] on word-representability of an odd k-sun with center.

Lemma 4.3.19 ([Kitaev et al., 2021])**.** *For $k \geq 3$, an odd k-sun with center is a non-word-representable graph.*

Note that the graph M_{V} from the set $\mathcal{C}$ is isomorphic to the graph T_8 depicted in [Chen et al., 2022, Figure 3]. Accordingly, the result [Chen et al., 2022, Theorem 19] can be stated as per the following.

Lemma 4.3.20 ([Chen et al., 2022])**.** *The graph M_{V} is non-word-representable.*

Lemma 4.3.21. *The graph F_0 is word-representable.*

Proof. The graph F_0 is obtained from the graph D_4, given in Figure 1.19, by adding an all-adjacent vertex. In view of Theorem 1.3.39, F_0 is a word-representable graph if and only if D_4 is a comparability graph. Note that the split graph D_4 is a comparability graph; for instance, see the transitive orientation marked along its edges in Figure 1.19. Hence, the result follows. □

Lemma 4.3.22. *The graphs $M_{\mathrm{III}}(3)$ and $K_1\vee$ tent are non-word-representable graphs.*

Proof. Note that the graphs $M_{\mathrm{III}}(3)$ and $K_1\vee$ tent are obtained from the graphs D_1 and D_2 (see Figure 1.19), respectively, by adding an all-adjacent vertex. In a word-representable graph, the neighborhood of every vertex induces a comparability graph (see Theorem 1.3.37 or Theorem 1.3.39). However, since none of D_1 and D_2 is a comparability graph (by Theorem 1.6.30), $M_{\mathrm{III}}(3)$ and $K_1\vee$ tent are not word-representable graphs. □

Definition 4.3.23 ($M_{\mathrm{II}}(k)$)**.** For an even integer $k \geq 4$, the graph $M_{\mathrm{II}}(k)$ is the split graph $(I \cup C, E)$ with $I = \{b_1, b_2, a_1, \ldots, a_{k-2}\}$ and $C = \{c_1, \ldots, c_k\}$ such that

$N(b_1) = \{c_1, \ldots, c_{k-2}, c_k\}$, $N(b_2) = \{c_2, \ldots, c_k\}$ and, for $1 \le i \le k-2$, $N(a_i) = \{c_i, c_{i+1}\}$.

Lemma 4.3.24. *For $k \ge 4$, the graph $M_{\rm II}(k)$ is non-word-representable.*

Proof. On the contrary, suppose $M_{\rm II}(k)$ is a word-representable graph. Then, in view of Theorem 1.6.35, the vertices of C can be labeled from 1 to $k = |C|$ such that, for $a \in I$, either $a \in A$, i.e., $N(a) = [1, m] \cup [n, k]$ (with $m < n$) or $a \in B$, i.e., $N(a) = [l, r]$ (with $l \le r$). We derive a contradiction in each of the following cases, by considering the possibilities of b_1, b_2 belonging to the sets A, B.

- Case 1: $b_1, b_2 \in A$. Note that $\deg(b_1) = k - 1 = \deg(b_2)$ and $N(b_1) \ne N(b_2)$. In view of Theorem 1.6.35(iii), without loss of generality, for some $m \ge 1$ with $m + 2 < k$, let $N(b_1) = [1, m] \cup [m + 2, k]$ and $N(b_2) = [1, m + 1] \cup [m + 3, k]$. Accordingly, the vertices c_1 and c_{k-1} are labeled as $m + 2$ and $m + 1$, respectively. Further, for each $1 \le i \le k - 2$, since the degree of a_i is two, we have $N(a_i) = [j, j + 1]$, for some $1 \le j < k$, or $N(a_i) = [1, 1] \cup [k, k]$. Note that $N(a_1) = \{c_1, c_2\}$ and c_1 is labeled as $m + 2$ $(< k)$. Since $m + 1$ is already used as the label of c_{k-1}, the vertex c_2 should be labeled as $m + 3$. Continuing with this argument, the vertices $c_3, \ldots, c_{k-m-1}$ should be labeled as $m + 4, \ldots, k$, respectively. On the other hand, since $N(a_{k-2}) = \{c_{k-2}, c_{k-1}\}$ and $m + 2$ is used as the label of c_1, the vertex c_{k-2} should be labeled as m. Again continuing with this argument, the vertices $c_{k-3}, \ldots, c_{k-m}$ are labeled as $m - 1, \ldots, 2$, respectively. This shows that $N(a_{k-m-1}) = \{c_{k-m-1}, c_{k-m}\} = \{k, 2\}$, a contradiction to Theorem 1.6.35(i), as $k \ge 4$.

- Case 2: $b_1, b_2 \in B$. Since $\deg(b_1) = \deg(b_2) = k - 1$, and $N(b_1) \ne N(b_2)$, one of $N(b_1)$ and $N(b_2)$ is $[2, k]$, and the other one is $[1, k - 1]$. Without loss of generality, suppose that $N(b_1) = [2, k]$ and $N(b_2) = [1, k - 1]$. Thus, c_1 and c_{k-1} are labeled as k and 1, respectively. Further, for each $1 \le i \le k - 2$, since the degree of a_i is two, we have $N(a_i) = [j, j + 1]$, for some $1 \le j < k$, or $N(a_i) = [1,1] \cup [k,k]$. Since $N(a_{k-2}) = \{c_{k-2}, c_{k-1}\}$ and k is already used as a label of c_1, the vertex c_{k-2} should be labeled as 2. Continuing the same argument, $3, \ldots, k-2$ must be the labels of the vertices $c_{k-3}, \ldots, c_2$, respectively. Thus, we have $N(a_1) = \{c_1, c_2\} = \{k, k-2\}$, a contradiction to Theorem 1.6.35(i), as $k \geq 4$.

- Case 3: $b_1 \in A$ and $b_2 \in B$. Since $\deg(b_2) = k-1$, we have $N(b_2) = [1, k-1]$ or $[2, k]$. Suppose that $N(b_2) = [1, k-1]$. Since $\deg(b_1) = k-1$, in view of Theorem 1.6.35(ii), we have $N(b_1) = [1, k-2] \cup [k, k]$. Thus, k and $k-1$ must be the labels of c_1 and c_{k-1}, respectively. Since $N(a_{k-2}) = \{c_{k-2}, c_{k-1}\}$, c_{k-2} should be labeled as $k-2$. Note that as $N(a_i) = \{c_i, c_{i+1}\}$, for each $2 \leq i \leq k-3$, continuing the same argument, the vertices $c_{k-3}, \ldots, c_2$ must receive the labels $k-3, \ldots, 2$, respectively. Thus, $N(a_1) = \{c_1, c_2\} = \{k, 2\}$, a contradiction to Theorem 1.6.35(i). Similarly, if $N(b_2) = [2, k]$, we will arrive at a contradiction.

Hence, the graph $M_{\mathrm{II}}(k)$ is non-word-representable. □

Definition 4.3.25 ($M_{\mathrm{III}}(k)$)**.** For an even integer $k \geq 4$, the graph $M_{\mathrm{III}}(k)$ is the split graph $(I \cup C, E)$ with $I = \{b, a_1, \ldots, a_{k-1}\}$ and $C = \{c_1, \ldots, c_{k+1}\}$ such that $N(b) = \{c_2, \ldots, c_{k-1}, c_{k+1}\}$ and, for $1 \leq i \leq k-1$, $N(a_i) = \{c_i, c_{i+1}\}$.

Lemma 4.3.26. *For $k \geq 4$, the graph $M_{\mathrm{III}}(k)$ is non-word-representable.*

Proof. On the contrary, suppose $M_{\mathrm{III}}(k)$ is a word-representable graph. Then, in view of Theorem 1.6.35, the vertices of C can be labeled from 1 to $k+1 = |C|$ such that, for $a \in I$, either $a \in A$, i.e., $N(a) = [1, m] \cup [n, k+1]$ (with $m < n$) or $a \in B$, i.e., $N(a) = [l, r]$ (with $l \leq r$). We derive a contradiction in each of the following cases, by considering the possibilities of b belonging to the sets A, B.

- Case 1: $b \in A$. Since $\deg(b) = k-1$, we have $N(b) = [1, m] \cup [m+3, k+1]$. Thus, one of c_1 and c_k is labeled as $m+1$, and the other one is labeled as $m+2$.

In view of the symmetry, suppose c_1 and c_k are labeled as $m+2$ and $m+1$, respectively. Since $N(a_1) = \{c_1, c_2\}$ and $m+1$ is already used as the label of c_k, the vertex c_2 should be labeled as $m+3$. Continuing the same argument, we conclude that the vertices $c_3, \ldots, c_{k-m}$ are labeled as $m+4, \ldots, k+1$, respectively. On the other hand, as $N(a_{k-1}) = \{c_{k-1}, c_k\}$, the vertex c_{k-1} should be labeled as m. Accordingly, the vertices $c_{k-2}, \ldots, c_{k-m+1}$ are labeled as $m-1, \ldots, 2$, respectively. Thus, $N(a_{k-m}) = \{c_{k-m}, c_{k-m+1}\} = \{k+1, 2\}$, a contradiction to Theorem 1.6.35(i), as $k \geq 4$.

- Case 2: $b \in B$. Since $\deg(b) = k-1$, we have $N(b) = [1, k-1]$ or $[2, k]$, or $[3, k+1]$. Suppose $N(b) = [1, k-1]$. Thus, one of c_1 and c_k is labeled as k, and the other one is labeled as $k+1$. Without loss of generality, suppose c_1 and c_k are labeled as k and $k+1$, respectively. Since $N(a_1) = \{c_1, c_2\}$, the vertex c_2 should be labeled as $k-1$. Accordingly, the vertices $c_3, \ldots, c_{k-1}$ are labeled as $k-2, \ldots, 2$, respectively. Thus, $N(a_{k-1}) = \{c_{k-1}, c_k\} = \{2, k+1\}$, a contradiction to Theorem 1.6.35(i). Similarly, if $N(b) = [2, k]$ or $[3, k+1]$, we will arrive at a contradiction.

Hence, there is no labeling of the vertices of C which satisfies Theorem 1.6.35 so that the graph $M_{\mathrm{III}}(k)$ is non-word-representable. □

Definition 4.3.27 (M_{IV})**.** The graph M_{IV} is the split graph $(I \cup C, E)$ with $I = \{a_1, a_2, a_3, a_4\}$ and $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ such that $N(a_1) = \{c_1, c_2\}, N(a_2) = \{c_3, c_4\}, N(a_3) = \{c_5, c_6\}$, and $N(a_4) = \{c_2, c_4, c_6\}$.

Lemma 4.3.28. *The graph M_{IV} is a non-word-representable graph.*

Proof. On the contrary, suppose M_{IV} is word-representable. Then, in view of Theorem 1.6.35, the vertices of C can be labeled from 1 to $6 = |C|$ such that, for $a \in I$, either $a \in A$, i.e., $N(a) = [1, m] \cup [n, 6]$ (with $m < n$) or $a \in B$, i.e., $N(a) = [l, r]$ (with $l \leq r$).

For $1 \leq i \leq 3$, since degree of a_i is two, we have $N(a_i) = \{1, 6\}$, or $\{j, j+1\}$, for some $1 \leq j \leq 5$. Further, since the degree of a_4 is three, and $N(a_4)$ contains exactly one vertex from each of $N(a_i)$, for $1 \leq i \leq 3$, and $N(a_i) \cap N(a_j) = \varnothing$ $(1 \leq i \neq j \leq 3)$, we see that $N(a_4)$ cannot be consecutive, i.e., $a_4 \notin B$. We now show that $a_4 \notin A$, as well; thus, we arrive at a contradiction.

Suppose $a_4 \in A$. Then the only possibilities are $N(a_4) = [1, 1] \cup [5, 6]$ or $[1, 2] \cup [6, 6]$. Suppose that $N(a_4) = [1, 1] \cup [5, 6]$. As $N(a_4)$ contains exactly one vertex from each of $N(a_i)$, for $1 \leq i \leq 3$, without loss of generality, suppose that $1 \in N(a_1), 5 \in N(a_2), 6 \in N(a_3)$. For $1 \leq i \neq j \leq 3$, since $N(a_i) \cap N(a_j) = \varnothing$, we must have $N(a_1) = \{1, 2\}$ and $N(a_2) = \{4, 5\}$. This implies that $N(a_3) = \{3, 6\}$; a contradiction to Theorem 1.6.35(i). Similarly, it can be shown that $N(a_4)$ cannot be $[1, 2] \cup [6, 6]$. □

Remark 4.3.29. While odd k-sun with center and M_V are known to be non-word-representable, in Lemmas 4.3.22, 4.3.24, 4.3.26 and 4.3.28, we established that the split graphs $M_{\text{III}}(3)$, $K_1 \vee$ tent, $M_{\text{II}}(k)$ (even $k \geq 4$), $M_{\text{III}}(k)$ (even $k \geq 4$) and M_{IV} are also non-word-representable. Moreover, in view of Theorem 1.6.32, they are minimal non-word-representable graphs.

Definition 4.3.30 ($F_1(k)$)**.** For an odd integer $k \geq 5$, the graph $F_1(k)$ is the split graph $(I \cup C, E)$ with $I = \{b_1, b_2, a_1, \ldots, a_{k-2}\}$ and $C = \{c_1, \ldots, c_{k-1}\}$ such that $N(b_1) = \{c_1, \ldots, c_{k-2}\}$, $N(b_2) = \{c_2, \ldots, c_{k-1}\}$ and, for $1 \leq i \leq k-2$, $N(a_i) = \{c_i, c_{i+1}\}$.

Lemma 4.3.31. *For $k \geq 5$, the graph $F_1(k)$ is word-representable.*

Proof. For each $1 \leq i \leq k-1$, we label c_i as i. Observe that under this labeling of the vertices of C, we have $N(b_1) = [1, k-2], N(b_2) = [2, k-1]$ and, for each $1 \leq i \leq k-2$, $N(a_i) = [i, i+1]$. Hence, in view of Theorem 1.6.35, the graph $F_1(k)$ is word-representable. □

Definition 4.3.32 ($F_2(k)$)**.** For an odd integer $k \geq 5$, the graph $F_2(k)$ is the split graph $(I \cup C, E)$ with $I = \{b, a_1, \ldots, a_{k-1}\}$ and $C = \{c_1, \ldots, c_k\}$ such that $N(b) = \{c_2, \ldots, c_{k-1}\}$ and, for $1 \leq i \leq k-1$, $N(a_i) = \{c_i, c_{i+1}\}$.

Lemma 4.3.33. *For $k \geq 5$, the graph $F_2(k)$ is word-representable.*

Proof. For each $1 \leq i \leq k$, we label c_i as i. Observe that under this labeling of the vertices of C, we have $N(b) = [2, k-1]$ and, for each $1 \leq i \leq k-1$, $N(a_i) = [i, i+1]$. Hence, in view of Theorem 1.6.35, the graph $F_2(k)$ is word-representable. □

Notation 4.3.34. Let $\mathcal{C}_3$ denote the subset of $\mathcal{C}$ consisting of the following graphs: (i) F_0, (ii) for even $k \geq 4$, even-k-sun, and (iii) for odd $k \geq 5$, $F_1(k)$ and $F_2(k)$.

In view of the lemmas and remarks established between Lemma 4.3.17 and Lemma 4.3.33, we have the following result.

Lemma 4.3.35. *The class of word-representable graphs in $\mathcal{C}$ is $\mathcal{C}_3$.*

We now provide the characterization of the class of word-representable split graphs and that of the class of word-representable WPC graphs having representation number exactly three.

Theorem 4.3.36. *Let G be a word-representable split graph. Then, $\mathcal{R}(G) = 3$ if and only if G contains at least one graph from $\mathcal{C}_3$ as an induced subgraph.*

Proof. From Theorem 4.3.14, it is evident that $\mathcal{R}(G) \leq 3$. Suppose $\mathcal{R}(G) = 3$. Since circle graphs are characterized as 2-word-representable graphs (by Theorem 1.3.28), G is not a circle graph. Then, from Theorem 1.6.32, G contains at least one graph from the set $\mathcal{C}$ as an induced subgraph. Further, as G is a word-representable split graph, in view of Lemma 4.3.35, G must have an induced subgraph from $\mathcal{C}_3$.

For converse, note that every graph in $\mathcal{C}_3$ is not a circle graph (cf. Theorem 1.6.32). Hence, $\mathcal{R}(G) = 3$. □

Along lines similar to those of Theorem 4.3.36, the following characterization can be obtained.

Theorem 4.3.37. *Let G be a word-representable WPC graph. Then, $\mathcal{R}(G) = 3$ if and only if G contains at least one graph from $\mathcal{C}_3$ as an induced subgraph.*

We now investigate the transitive orientability of each graph from the set $\mathcal{C}$ to characterize the split comparability graphs and WPC comparability graphs with representation number three.

Remark 4.3.38. The graphs belonging to $\mathcal{C} \setminus \mathcal{C}_3$ are not comparability graphs, as they are not word-representable (by Lemma 4.3.35).

Hence, it is sufficient to verify the graphs in $\mathcal{C}_3$. To prove $F_1(5)$ is a comparability graph, we need the following characterization of split comparability graphs from [Ortiz and Villanueva, 1996]. The characterization is rewritten in terms of the vertex labeling, along the lines of Theorem 1.6.35. For an alternative proof of the following result, one can refer to [Dwary et al., 2025].

Theorem 4.3.39 ([Ortiz and Villanueva, 1996])**.** *Let $G = (I \cup C, E)$ be a split graph. Then, G is transitively orientable if and only if the vertices of C can be labeled from 1 to $k = |C|$ such that the following properties* (i) – (v) *hold: For $a, b \in I$,*

(i) *The neighborhood $N(a)$ has one of the following forms: $[1, m] \cup [n, k]$ for $m < n$, $[1, r]$ for $r < k$, or $[l, k]$ for $l > 1$.*

(ii) *If $N(a) = [1, r]$ and $N(b) = [l, k]$, for $r < k$ and $l > 1$, then $r < l$.*

(iii) *If $N(a) = [1, m] \cup [n, k]$ and $N(b) = [1, r]$, for $r < k$ and $m < n$, then $r < n$.*

(iv) *If $N(a) = [1, m] \cup [n, k]$ and $N(b) = [l, k]$, for $l > 1$ and $m < n$, then $m < l$.*

(v) *If $N(a) = [1, m] \cup [n, k]$ and $N(b) = [1, m'] \cup [n', k]$, for $m < n$ and $m' < n'$, then $m < n'$ and $m' < n$.*

Theorem 4.3.40. *The graphs F_0 and $F_1(5)$ are the only comparability graphs in $\mathcal{C}_3$.*

Proof. Consider the vertex sets $\{c_1, c_2, c_3, a_1, a_2, a_3, a_k\}$ in even-k-sun (for $k \geq 4$), $\{c_2, c_3, c_4, a_1, a_2, a_3, a_4\}$ in $F_1(k)$ (for $k \geq 7$) and $\{c_2, c_3, c_4, a_1, a_2, a_3, a_4\}$ in $F_2(k)$ (for $k \geq 5$). Note that the subgraphs induced by these vertex sets in the respective graphs are isomorphic to D_3 (see Figure 1.19). Since D_3 is not a comparability graph (cf. Theorem 1.6.30), none of even-k-sun (for $k \geq 4$), $F_1(k)$ (for $k \geq 7$) and $F_2(k)$ (for $k \geq 5$) are comparability graphs.

Recall that the graph F_0 is obtained from the split comparability graph D_4 by adding the all-adjacent vertex 0 (see Figure 1.20). Thus, by Remark 3.1.10, F_0 is a comparability graph.

Note that $F_1(5)$ is the split graph $(I \cup C, E)$ with $I = \{a_1, a_2, a_3, b_1, b_2\}$ and $C = \{c_1, c_2, c_3, c_4\}$ such that $N(b_1) = \{c_1, c_2, c_3\}$, $N(b_2) = \{c_2, c_3, c_4\}$ and, for $1 \leq i \leq 3$, $N(a_i) = \{c_i, c_{i+1}\}$. We assign the labels $3, 4, 1$ and 2 to the vertices c_1, c_2, c_3 and c_4, respectively. Under this labeling of the vertices of C, we have $N(a_1) = [3, 4]$, $N(a_2) = [1, 1] \cup [4, 4]$, $N(a_3) = [1, 2]$, $N(b_1) = [1, 1] \cup [3, 4]$, and $N(b_2) = [1, 2] \cup [4, 4]$. Observe that this labeling satisfies all the conditions given in Theorem 4.3.39 so that $F_1(5)$ is a comparability graph. □

The following characterizations of split comparability graphs and WPC comparability graphs with representation number three can be concluded directly from Theorems 4.3.14, 4.3.36, 4.3.37 and 4.3.40.

Theorem 4.3.41. *Let G be a split comparability graph. Then, $\mathcal{R}(G) = 3$ if and only if G contains F_0 or $F_1(5)$ as an induced subgraph.*

Theorem 4.3.42. *Let G be a WPC comparability graph. Then, $\mathcal{R}(G) = 3$ if and only if G contains F_0 or $F_1(5)$ as an induced subgraph.*

Remark 4.3.43. Since the complete graphs are the graphs with representation number one, the characterizations given in Theorems 4.3.36, 4.3.37, 4.3.41 and 4.3.42

completely classify the word-representable split graphs, word-representable WPC graphs, split comparability graphs and WPC comparability graphs, respectively, with respect to their representation numbers.

4.4 Permutation-Representation Number

In this section, we study the *prn*s of WPC comparability graphs. From an existing result in the dimension theory, we first conclude that the *prn* of a WPC comparability graph is at most four. Further, we determine the *prn* of a starlike comparability graph. The class of starlike comparability graphs includes split comparability graphs and is a subclass of WPC comparability graphs.

4.4.1 Split Graphs and WPC Graphs

In this subsection, we recall the results on the dimension of a split order as well as that of a cycle-free poset. Accordingly, we present the corresponding results on the *prn* of a split comparability graph and that of a WPC comparability graph.

Definition 4.4.1 (Split order)**.** A poset is said to be a *split order* if its corresponding comparability graph is a split graph.

Theorem 4.4.2 ([Guenver and Rampon, 2004])**.** *Every split order has dimension at most three, and the bound is tight.*

Definition 4.4.3 (Cycle-free poset)**.** A poset is said to be *cycle-free* if its corresponding comparability graph is a chordal graph.

Theorem 4.4.4 ([Ma and Spinrad, 1991])**.** *Every cycle-free poset has dimension at most four.*

Thus, in view of Theorems 1.3.43, 4.4.2 and 4.4.4, we have the following results on the *prn* of a split comparability graph and that of a WPC comparability graph.

Theorem 4.4.5. *If G is a split comparability graph, then $\mathcal{R}^p(G) \leq 3$, and the bound is tight.*

Theorem 4.4.6. *If G is a WPC comparability graph, then $\mathcal{R}^p(G) \leq 4$.*

However, we do not have an example to show that the upper bound given in Theorem 4.4.6 is tight.

4.4.2 Starlike Graphs

In the following, we consider the starlike graphs in the hierarchy

$$\text{split graphs} \subset \text{starlike graphs} \subset \text{WPC graphs} \subset \text{chordal graphs}$$

and obtain the minimal forbidden induced subgraph characterization for starlike comparability graphs along with their *prn*.

Definition 4.4.7 (Starlike graph)**.** A graph is said to be a *starlike graph* if it is an intersection graph of substars of a star.

Remark 4.4.8 ([Ahn et al., 2022])**.** For a starlike graph G, there exists a partition tree $\mathcal{T}$ of G which is a star. Let $\mathcal{T}$ be centered at a clique C and B be a leaf bag of $\mathcal{T}$. For $a \in C$, the vertex a is adjacent to either all vertices of B or no vertex of B. Thus, each leaf bag of $\mathcal{T}$ is a module in G.

Remark 4.4.9. Let $B_1, \ldots, B_k$ be the leaf bags of $\mathcal{T}$ given in Remark 4.4.8. Consider new vertices d_i corresponding to each bag B_i, for $1 \leq i \leq k$. Construct the split graph $G' = (C \cup I, E)$ from G by replacing each bag B_i with the vertex b_i, where $I = \{b_i \mid 1 \leq i \leq k\}$ is an independent set in G'. Note that, since each B_i is a module in G, one can reconstruct G from the split graph G' by replacing each vertex b_i with the corresponding clique B_i.

Thus, we have the following result.

Lemma 4.4.10. *Every starlike graph can be obtained from a split graph* $(I \cup C, E)$ *by replacing each vertex of the independent set* I *with some cliques.*

We now provide a forbidden induced subgraph characterization of starlike comparability graphs and show that the *prn* of a starlike comparability graph is at most three.

Theorem 4.4.11. *Let* G *be a starlike graph. Then the following hold.*

1. G *is a comparability graph if and only if* G *does not contain any of* D_1, D_2 *and* D_3 *(see Figure 1.19) as an induced subgraph.*
2. *If* G *is a comparability graph, then* $\mathcal{R}^p(G) \leq 3$. *Moreover,* $\mathcal{R}^p(G) = 3$ *if and only if* G *contains* D_4 *(see Figure 1.19) as an induced subgraph.*

Proof. In view of Lemma 4.4.10, suppose G is obtained from the split graph $G' = (I \cup C, E)$ by replacing each $b_i \in I$ with some clique B_i, for $1 \leq i \leq k = |I|$. Then, in view of Theorem 3.1.12, we have G is a comparability graph if and only if G' and each B_i are comparability graphs. Since cliques are comparability graphs, it is evident that G is a comparability graph if and only if the split graph G' is a comparability graph. Thus, part (1) follows from Theorem 1.6.30.

If G is a comparability graph, then from the previous paragraph, we have the split graph G' is a comparability graph. In view of Theorem 3.1.12, we have $\mathcal{R}^p(G) = \max\{\mathcal{R}^p(G'), \mathcal{R}^p(B_1), \ldots, \mathcal{R}^p(B_k)\}$. Since the *prn* of a clique is one, we have $\mathcal{R}^p(G) = \mathcal{R}^p(G')$. Thus, $\mathcal{R}^p(G) \leq 3$ as $\mathcal{R}^p(G') \leq 3$ (by Theorem 4.4.5).

First note that the split graph D_4 is a comparability graph; for instance, see the transitive orientation marked along its edges in Figure 1.19. Thus, $\mathcal{R}^p(D_4) \leq 3$. Now we recall from Theorem 1.6.31 that a split graph is a permutation graph if and only if it is $\mathcal{D}$-free (see Figure 1.19). Further, in view of Theorem 1.3.45, note that permutation graphs are characterized as permutationally 2-representable graphs. Hence, being in $\mathcal{D}$, the graph D_4 is not permutationally 2-representable so that

$\mathcal{R}^p(D_4) = 3$. Thus, if G contains D_4 as an induced subgraph, then $\mathcal{R}^p(G) = 3$. Conversely, suppose $\mathcal{R}^p(G) = 3$. Then, we have $\mathcal{R}^p(G') = 3$ so that the split graph G' is not permutationally 2-representable. Thus, G' contains at least one graph from $\mathcal{D}$. But, by Theorem 1.6.30, D_4 is the only comparability graph from $\mathcal{D}$. Thus, G' must contain D_4 as an induced subgraph. Hence, D_4 is an induced subgraph of G, as G' is an induced subgraph of G. $\square$

4.5 Conclusion and Future Work

This work demonstrates that split decomposition has emerged as a powerful and efficient tool for analyzing the word-representability of WPC graphs. We proved that the components of the minimal split decomposition of a WPC graph are split graphs (Lemma 4.1.2) and observed that the converse is not true (Remark 4.1.4). If the split recomposition of split graphs can be specified to characterize WPC graphs, it may lead to a linear-time algorithm for recognizing WPC graphs. At present, only a polynomial-time recognition of WPC graphs is available [Ahn et al., 2022]. A forbidden induced subgraph characterization of word-representable WPC graphs could be obtained along the lines of proof of Theorem 4.1.6, using the forbidden induced subgraph characterization for word-representable split graphs from [Srinivasan and Hariharasubramanian, 2025]. In view of Theorem 4.4.6, it is interesting to determine whether a WPC comparability graph with *prn* equal to four exists.

5

Minimum-Word-Representants

In this chapter, we characterize double-arborescences as P_4-free treelike comparability graphs. We also characterize P_4-free graphs (known as cographs) using the split-decomposition trees. Consequently, using split-decomposition trees, we characterize double-arborescences and one of its subclasses, namely arborescences. Moreover, we obtain an alternative proof for the well-known characterization of arborescences given in [Wolk, 1965] using split decomposition. Further, we devise an algorithm to construct a minimum-word-representant for a given arborescence. The algorithm is then extended to obtain a minimum-word-representant of length $2n - \kappa$ for a double-arborescence and more generally, for a treelike permutation graph, where n is the number of vertices and κ is the clique number of the graph.

5.1 Double-Arborescences

In this section, we characterize double-arborescences and prove that the length of the minimum-word-representants of a double-arborescence on n vertices with clique number κ is $2n - \kappa$.

5.1.1 Characterizations

In this subsection, we characterize double-arborescences as P_4-free treelike comparability graphs. We also obtain another characterization for double-arborescences by providing a necessary and sufficient condition for P_4-free graphs using the notion of s-leaf-path in their split-decomposition trees. Further, we obtain an alternative proof for the well-known characterization of arborescences given in [Wolk, 1965, Theorem 3].

Lemma 5.1.1. *Let $G = (V, E)$ be a path of k double-arborescences with smallest possible k. If $k \geq 2$, then G contains P_4 as an induced subgraph.*

Proof. Let O be the treelike orientation of G and let $\tilde{G} = (V, \tilde{O})$ be the transitive reduction of O. Let G_i with root r_i, $1 \leq i \leq k$, be the k double-arborescences in G such that $r_1 - r_2 - \cdots - r_k$ is the root-path in $\tilde{G}$.

Suppose the edge $\overline{r_1 r_2}$ is oriented as $\overrightarrow{r_1 r_2}$ in $\tilde{O}$. We claim that there exists a vertex a_1 in G_1 such that $\overrightarrow{r_1 a_1}$ is in $\tilde{O}$. On the contrary, suppose $\overrightarrow{a r_1}$ in O for all vertices a in G_1. Then, we have $\overrightarrow{a r_2}$ in O for all vertices a in G_1. Then, G is a path of $k-1$ double-arborescences, namely $G_1 \cup G_2, G_3, \ldots, G_k$ with the root-path $r_2 - r_3 - \cdots - r_k$ in $\tilde{G}$; a contradiction to the minimality of k. Further, there exist a vertex a_2 in $\bigcup\limits_{2 \leq i \leq k} G_i$ such that $\overrightarrow{a_2 r_t}$ is in $\tilde{O}$, for some $t \geq 2$, as shown in the following cases:

- Case 1: $\overrightarrow{r_{i+1} r_i}$ in $\tilde{O}$ for some $i \geq 2$. Let t be the least such that $\overrightarrow{r_{t+1} r_t}$ in $\tilde{O}$. Then we choose a_2 to be r_{t+1}.

- Case 2: $\overrightarrow{r_ir_{i+1}}$ for all $i < k$ in $\tilde{O}$. For each $i \geq 2$, if $\overrightarrow{r_ia}$ in O for all vertices a in G_i, then $\overrightarrow{r_1a}$ in O for all vertices a in $\bigcup\limits_{2\leq i\leq k} G_i$ as O is a transitive orientation. In which case, G is a double-arborescence with the root r_1; a contradiction to the minimality of k. Hence, there exists a_2 in G_t, for some $t \geq 2$, such that $\overrightarrow{a_2r_t}$ is in $\tilde{O}$.

The path $a_1 - r_1 - r_t - a_2$ is an induced P_4 in G. Indeed, other than these three edges, there will not be any more edges between a_1, r_1, r_t and a_2 in G as there is no directed path between a_1 and r_t; a_1 and a_2; or r_1 and a_2 in $\tilde{G}$.

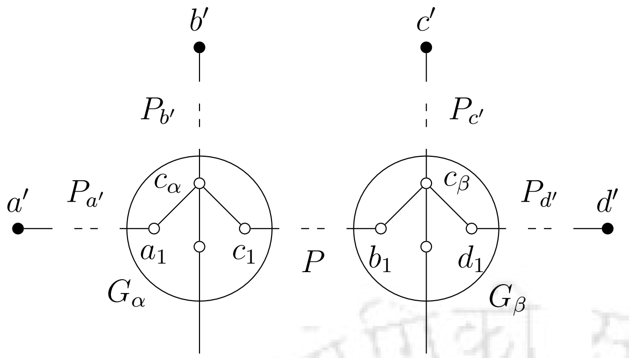
Similarly, one can observe that if $\overline{r_1r_2}$ is oriented as $\overrightarrow{r_2r_1}$ in $\tilde{O}$, then there exist vertices a_1 in G_1 and a_2 in $\bigcup\limits_{2\leq i\leq k} G_i$ such that $\overrightarrow{a_1r_1}$ and $\overrightarrow{r_ta_2}$ are in $\tilde{O}$ for some $t \geq 2$ so that $\{a_1, r_1, r_t, a_2\}$ induces a P_4 in G. □

We now prove that the class of double-arborescences is the intersection of the class of P_4-free graphs and the class of treelike comparability graphs.

Theorem 5.1.2. *A graph G is a double-arborescence if and only if G is a P_4-free treelike comparability graph.*

Proof. Suppose G is a double-arborescence with a root r. Since G is a treelike comparability graph, by Remark 1.6.16, G is distance-hereditary. If G contains a P_4 induced by, say, $\{a_1, a_2, a_3, a_4\}$, then clearly $r \neq a_i$, for all $1 \leq i \leq 4$, as r is an all-adjacent vertex of G. Note that the graph induced by $\{r, a_1, a_2, a_3, a_4\}$ is the complement of $K_1 \cup P_4$ (see gem in Figure 1.15). But, by Theorem 1.6.3, it is a forbidden induced subgraph for a distance-hereditary comparability graph. Hence, G is P_4-free.

Conversely, suppose a treelike comparability graph G is P_4-free. Note that P_4-free graphs are permutation graphs (cf. Theorem 1.6.9). Thus, by Theorem 1.6.27, G is a path of k double-arborescences such that k is the smallest number of double-arborescences in G. If $k \geq 2$, then by Lemma 5.1.1, G induces a P_4; which is not possible. Hence, $k = 1$ so that G is a double-arborescence. □



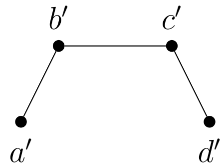


FIGURE 5.1: A portion of some $T_\mathcal{F}$ with an s-leaf-path P and its accessibility graph P_4

We now characterize the class of P_4-free graphs (in Corollary 5.1.5) through the notion of s-leaf-paths in their minimal split-decomposition trees. Recall from Theorem 1.6.2 that the minimal split-decomposition tree of a distance-hereditary graph is a clique-star tree.

Definition 5.1.3 (s-leaf-path)**.** Let G be a distance-hereditary graph with the minimal split-decomposition tree $T_\mathcal{F}$. We call a leaf of a star component in $T_\mathcal{F}$ as *s-leaf*. An *s-leaf-path* in $T_\mathcal{F}$ is an alternated path P such that the endpoints of P are s-leaves of two star components and P does not contain any edge from either of star components (e.g., refer Figure 5.1).

Theorem 5.1.4. *Let G be a distance-hereditary graph and $T_\mathcal{F}$ be the minimal split-decomposition tree of G. Then G contains an induced P_4 if and only if there exists an s-leaf-path in $T_\mathcal{F}$.*

Proof. Note that by Theorem 1.6.2, $T_\mathcal{F}$ is a clique-star tree. Since G is isomorphic to the accessibility graph $T_\mathcal{F}^A$ (see Theorem 1.4.21(i)), we prove that $T_\mathcal{F}^A$ contains an induced P_4 if and only if there exists an s-leaf-path in $T_\mathcal{F}$. Let P be an s-leaf-path between two star components G_α and G_β in $T_\mathcal{F}$. Let $c_1 \in G_\alpha$, $b_1 \in G_\beta$ be the endpoints of P. Since $T_\mathcal{F}$ is the minimal split-decomposition tree, both G_α and G_β have at least three vertices each (see Theorem 1.4.21(iii)). Thus, there are at least two s-leaves in each of the star components G_α and G_β. Then, there are at least two maximal alternated paths from G_α that do not use any edge of G_α and, by Lemma 1.6.11, these maximal alternated paths end at distinct pendant vertices of $T_{\mathcal{F}}$. Of these paths, suppose one path, say $P_{a'}$, is from an s-leaf a_1 in G_α to a pendant vertex a' of $T_{\mathcal{F}}$, and the other path, say $P_{b'}$, is from the center c_α of G_α to a pendant vertex b' of $T_{\mathcal{F}}$. Similarly, there are at least two maximal alternated paths from G_β one, say $P_{c'}$, is from its center c_β to a pendant vertex c' of $T_{\mathcal{F}}$, and the other, say $P_{d'}$, is from an s-leaf, say d_1, in G_β to a pendant vertex d' of $T_{\mathcal{F}}$, as shown in Figure 5.1.

We show that $a' - b' - c' - d'$ is an induced path in $T_{\mathcal{F}}^A$. As shown in Figure 5.1, note that the path consisting of $P_{a'}$ followed by the edge $\overline{a_1 c_\alpha}$ and then $P_{b'}$ is an alternated path from a' to b' in $T_{\mathcal{F}}$ so that a' and b' are adjacent in $T_{\mathcal{F}}^A$. Further, the path consisting of $P_{b'}$ followed by the edge $\overline{c_\alpha c_1}$, the path P, the edge $\overline{b_1 c_\beta}$, then the path $P_{c'}$ is an alternated path from b' to c' in $T_{\mathcal{F}}$. Thus, b' and c' are adjacent in $T_{\mathcal{F}}^A$. Similarly, the vertices c' and d' are adjacent in $T_{\mathcal{F}}^A$. Consider the underlying tree T of $T_{\mathcal{F}}$ and note that there is a unique path between a' and c' in T that should pass through the vertices α and β of T. Thus, any path between a' and c' in $T_{\mathcal{F}}$ should pass through G_α and G_β, and by construction, that path must use two edges of G_α so that it cannot be an alternated path. This shows that a' and c' are not adjacent in $T_{\mathcal{F}}^A$. Similarly, one can show that $\overline{b'd'}$ and $\overline{a'd'}$ are not in $T_{\mathcal{F}}^A$. Hence, $T_{\mathcal{F}}^A$ contains an induced P_4.

Conversely, suppose $T_{\mathcal{F}}^A$ has an induced P_4, say $a' - b' - c' - d'$. Since $\overline{b'a'}$, $\overline{b'c'}$ are edges of $T_{\mathcal{F}}^A$, there exist alternated paths, say $P_{b',a'}$ and $P_{b',c'}$, which begin at the pendant vertex b' of $T_{\mathcal{F}}$ and end at the pendant vertices a' and c' of $T_{\mathcal{F}}$, respectively. Similarly, there exists an alternated path $P_{c',d'}$ between the pendant vertices c' and d' of $T_{\mathcal{F}}$.

Let G_α be the split component of $T_{\mathcal{F}}$ until which the paths $P_{b',a'}$ and $P_{b',c'}$ have the common segment and they split thereafter. Let $c_\alpha \in G_\alpha$ be the vertex at which the common segment ends, and a_1 and c_1 be the vertices of G_α at which the paths

$P_{b',a'}$ and $P_{b',c'}$ exit G_α, respectively. We now show that a_1 and c_1 are not adjacent in G_α so that G_α is a star component. Otherwise, we can have an alternated path between a' and c', namely P_{a',a_1} followed by the edge $\overline{a_1c_1}$ of G_α and then $P_{c_1,c'}$, where P_{a',a_1} is the segment of $P_{b',a'}$ between a_1 and a', and $P_{c_1,c'}$ is the segment of $P_{b',c'}$ between c_1 and c'. Thus, a' and c' are adjacent in $T^A_{\mathcal{F}}$; a contradiction. Hence, G_α must be a star component. It is evident that c_α is its center and a_1, c_1 are its s-leaves.

Let G_β be the split component of $T_{\mathcal{F}}$ until which the paths $P_{c',d'}$ and $P_{b',c'}$ have the common segment and they split thereafter. Let $c_\beta \in G_\beta$ be the vertex at which the common segment ends, and b_1 and d_1 be the vertices of G_β at which the paths $P_{b',c'}$ and $P_{c',d'}$ exit G_β, respectively. As shown above, observe that G_β is a star component, in which c_β is the center and b_1, d_1 are s-leaves.

Note that the alternated path $P_{b',c'}$ passes through the split components G_α and G_β. We observe that the pendant vertex b' is nearer to G_α than G_β on the path $P_{b',c'}$. On the contrary, suppose G_β is nearer to b'. Then, clearly, G_α would be nearer to c'. Since the path $P_{b',c'}$ exits G_α at c_1 towards the pendant vertex c', the vertex c_1 is closest in G_α to c'. Similarly, the vertex b_1 is closest in G_β to b'. Accordingly, the vertices $b', c', c_1, b_1, c_\alpha, c_\beta$ will appear in the following sequence on the path $P_{b',c'}$:

$$b', b_1, c_\beta, c_\alpha, c_1, c'$$

Thus, the segment of $P_{b',c'}$ between c_α and c_β is a center-center path in $T_{\mathcal{F}}$, as shown in Figure 5.2. Then along the lines of the proof of Theorem 1.6.5, as given in [Bahrani and Lumbroso, 2018], it can be observed that $a' - b' - c' - d' - a'$ forms a C_4 in $T^A_{\mathcal{F}}$; a contradiction to induced P_4 over these vertices. Hence, the pendant vertex b' is nearer to G_α than G_β on the path $P_{b',c'}$ so that the vertices $b', c', c_1, b_1, c_\alpha, c_\beta$ will appear in the following sequence on the path $P_{b',c'}$:

$$b', c_\alpha, c_1, b_1, c_\beta, c'$$

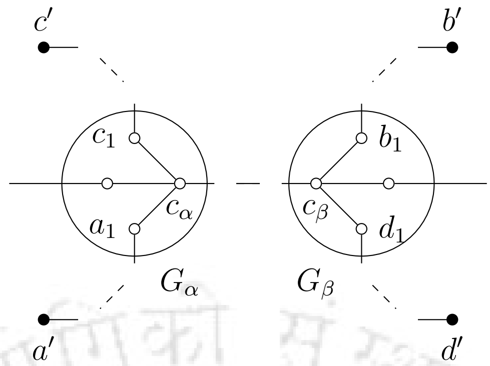


FIGURE 5.2: A center-center path

Evidently, the segment of $P_{b',c'}$ between c_1 and b_1 is an s-leaf-path between G_α and G_β in $T_{\mathcal{F}}$. □

Since P_4-free graphs are distance-hereditary (see Theorem 1.6.8), and the distance-hereditary graphs are characterized as the graphs whose minimal split-decomposition trees are clique-star trees (see Theorem 1.6.2), we have the following corollary of Theorem 5.1.4.

Corollary 5.1.5. *Let $T_{\mathcal{F}}$ be the minimal split-decomposition tree of a graph G. Then G is P_4-free if and only if $T_{\mathcal{F}}$ is a clique-star tree and $T_{\mathcal{F}}$ does not have any s-leaf-path.*

We now consolidate the following characterization of double-arborescences in terms of their minimal split-decomposition trees.

Corollary 5.1.6. *Let G be a graph and $T_{\mathcal{F}}$ be its minimal split-decomposition tree. Then G is a double-arborescence if and only if the following statements hold.*

(i) *$T_{\mathcal{F}}$ is a clique-star tree.*

(ii) *Each clique component has at most two marked vertices.*

(iii) *There is no marked edge between the centers of two star components.*

(iv) *$T_{\mathcal{F}}$ does not have any s-leaf-path.*

Proof. First note that the statements (i), (ii) and (iii) characterize treelike comparability graphs, by Theorem 1.6.15. Further, by Corollary 5.1.5, the statements (i) and (iv) characterize P_4-free graphs. Hence, the result follows in view of Theorem 5.1.2 that P_4-free treelike comparability graphs are precisely double-arborescences. $\square$

We now continue to provide an alternative proof for the characterization of arborescences as (C_4, P_4)-free graphs using split decomposition. First we characterize the strict-double-arborescences and consequently show that arborescences are precisely (C_4, P_4)-free treelike comparability graphs. Further, we prove that (C_4, P_4)-free graphs are treelike comparability graphs to conclude the alternative proof.

Lemma 5.1.7. *Suppose G is a strict-double-arborescence with a root r and O is the treelike orientation of G. There exist two non-adjacent vertices a_1 and a_2 in G such that $\overrightarrow{ra_1}$ and $\overrightarrow{ra_2}$ in O. Similarly, there exist two non-adjacent vertices a_3 and a_4 in G such that $\overrightarrow{a_3r}$ and $\overrightarrow{a_4r}$ in O.*

Proof. Suppose $G = (V, E)$ is a strict-double-arborescence with a root r. Let $\tilde{G} = (V, \tilde{O})$ be the transitive reduction of O. Let $A_1 = \{a \in V \mid \overrightarrow{ra} \text{ in } O\}$ and $A_2 = \{a \in V \mid \overrightarrow{ar} \text{ in } O\}$. Note that by definition of strict-double-arborescence, we have $A_1 \neq \varnothing$, $A_2 \neq \varnothing$, and $V = \{r\} \cup A_1 \cup A_2$. Moreover, if $a, b \in A_1$ are adjacent in G, then they are on a directed path from r in $\tilde{G}$. Accordingly, if the vertices of A_1 form a clique in G, then the vertices of A_1 are on a directed path from r in $\tilde{G}$. In that case, G can be seen as an arborescence whose root would be the end point, say a', of the above mentioned directed path, i.e., $V = \{a'\} \cup \{a \in V \mid \overrightarrow{aa'} \text{ in } O\}$; a contradiction to G is a strict-double-arborescence. Hence, there exist two vertices $a_1, a_2 \in A_1$ such that a_1 and a_2 are not adjacent in G. Similarly, there exist two non-adjacent vertices in A_2. $\square$

Theorem 5.1.8. *Let G be a double-arborescence. Then, G is a strict-double-arborescence if and only if G contains C_4 as its induced subgraph.*

Proof. Let O be the treelike orientation of $G = (V, E)$ and $\tilde{G} = (V, \tilde{O})$ be the transitive reduction of O.

Suppose G is a strict-double-arborescence with root r. Then, by Lemma 5.1.7, there exist two pairs of non-adjacent vertices a_1, b_1 and a_2, b_2 in G such that $\overrightarrow{ra_1}$, $\overrightarrow{rb_1}$, $\overrightarrow{a_2r}$ and $\overrightarrow{b_2r}$ exist in O. Then $\{a_1, b_1, a_2, b_2\}$ induces a C_4, namely $a_1 - a_2 - b_1 - b_2 - a_1$, in G.

Conversely, suppose G contains a $C_4 = a'_1 - a'_2 - b'_1 - b'_2 - a'_1$ (say) as its induced subgraph. We show that the double-arborescence G is not an arborescence so that G is a strict-double-arborescence. On the contrary, let G be an arborescence with root r such that r is a source in O. Without loss of generality, suppose $\overrightarrow{a'_1a'_2} \in O$. Since b'_1 is adjacent to a'_2 but non-adjacent to a'_1 in G, we must have $\overrightarrow{b'_1a'_2} \in O$. Thus, there are two directed paths, one from a'_1 to a'_2 and the other from b'_1 to a'_2 in $\tilde{G}$. Further, since r is an all-adjacent vertex in G, we must have directed paths from r to a'_1 and from r to b'_1 in $\tilde{G}$. These four paths together form a cycle in the underlying undirected graph of $\tilde{G}$, but as it is a tree, we have a contradiction. Hence, G is not an arborescence. □

In view of Theorem 5.1.2, since double-arborescences are precisely the P_4-free treelike comparability graphs, we have the following corollary of Theorem 5.1.8.

Corollary 5.1.9. *A graph G is an arborescence if and only if it is a (C_4, P_4)-free treelike comparability graph.*

Using the split decomposition, we now present an alternative proof for the well-known characterization of arborescences given in [Wolk, 1965, Theorem 3] (also see [Wolk, 1962]).

Theorem 5.1.10. *A graph G is an arborescence if and only if G is (C_4, P_4)-free.*

Proof. Suppose G is an arborescence. Then, by Corollary 5.1.9, G is a (C_4, P_4)-free graph.

Conversely, suppose G is a (C_4, P_4)-free graph. We show that G is a treelike comparability graph so that it is an arborescence (by Corollary 5.1.9). For that, we use the characterization given in Theorem 1.6.15. Since G is P_4-free, G is a distance-hereditary graph (by Theorem 1.6.8) so that the minimal split-decomposition tree $T_{\mathcal{F}}$ of G is a clique-star tree (by Theorem 1.6.2). Further, as G is (C_4, P_4)-free, note that $T_{\mathcal{F}}$ has neither a center-center path (by Theorem 1.6.5) nor an s-leaf-path (by Corollary 5.1.5). In particular, there is no marked edge between the centers of two star components; otherwise, we will have a center-center path in $T_{\mathcal{F}}$.

Finally, we claim that each clique component of $T_{\mathcal{F}}$ has at most two marked vertices. On the contrary, suppose there is a clique component in $T_{\mathcal{F}}$ with three marked vertices say a_1, a_2 and a_3. For $1 \leq i \leq 3$, let e_i be the marked edge in $T_{\mathcal{F}}$ with one end point a_i and the other end point, say, b_i. Since $T_{\mathcal{F}}$ is the minimal split-decomposition tree of G, by statement (iv) of Theorem 1.4.21, each b_i $(1 \leq i \leq 3)$ is a vertex of a star component. Note that not all b_i's can be centers of the respective star components; otherwise, $T_{\mathcal{F}}$ will have a center-center path, e.g., $b_1 - a_1 - a_2 - b_2$. Also, not all b_i's can be s-leaves of the respective star components; otherwise, $T_{\mathcal{F}}$ will have an s-leaf-path, e.g., $b_1 - a_1 - a_2 - b_2$. Hence, one of b_i's will be the center (or a s-leaf), without loss of generality assume that it is b_1, and the other two will be s-leaves (or the centers, respectively) of the respective star components. In any case, $b_2 - a_2 - a_3 - b_3$ is an s-leaf-path or center-center path in $T_{\mathcal{F}}$, which is not possible in $T_{\mathcal{F}}$. Thus, there will be at most two marked vertices in any clique component of $T_{\mathcal{F}}$. Hence, by Theorem 1.6.15, G is a treelike comparability graph. □

We reconcile the following characterization of arborescences in terms of their minimal split-decomposition trees.

Corollary 5.1.11. *Let G be a graph and $T_{\mathcal{F}}$ be its minimal split-decomposition tree. Then G is an arborescence if and only if the following statements hold.*

(i) *$T_{\mathcal{F}}$ is a clique-star tree.*

(ii) $T_{\mathcal{F}}$ *does not have any center-center path.*

(iii) $T_{\mathcal{F}}$ *does not have any s-leaf-path.*

Proof. Suppose G is an arborescence. Then, by Theorem 5.1.10, we have G is a (C_4, P_4)-free graph. Since G is P_4-free, the statements (i) and (iii) hold (by Corollary 5.1.5). Further, the statement (ii) holds by Theorem 1.6.5.

For proof of the converse, first note that the statement (i) implies that G is distance-hereditary. Thus, using statements (i) and (ii), G is C_4-free (by Theorem 1.6.5) and using statements (i) and (iii), we have G is P_4-free (by Corollary 5.1.5). Hence, G is a (C_4, P_4)-free graph so that G is an arborescence (by Theorem 5.1.10). $\square$

5.1.2 Taxonomy

Figure 5.3 depicts the taxonomy of various classes of graphs in the context of distance-hereditary graphs discussed in this work, namely treelike comparability graphs, P_4-free graphs (known as cographs), C_4-free graphs, arborescences (marked by X in the figure) and strict-double-arborescences (marked by Y in the figure). Observe that X ∪ Y is the class of double-arborescences.

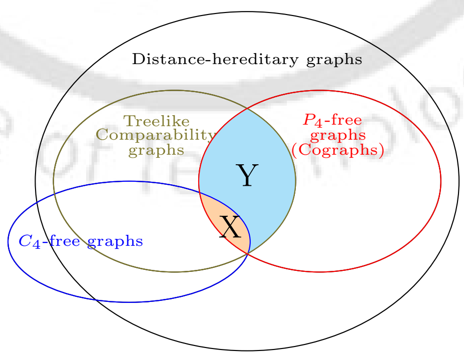


Figure 5.3: Taxonomy of various graph classes

5.1.3 Minimum-Word-Representants

In this subsection, we devise an algorithm to find minimum-word-representants of arborescences and extend it to double-arborescences. Moreover, we prove that for a double-arborescence G on n vertices with clique number κ, the length of a minimum-word-representant $\ell(G) = 2n - \kappa$.

Remark 5.1.12. In [Bouchet, 1988], it was established that the class of distance-hereditary graphs are circle graphs. Further, by Theorem 1.3.28, we have circle graphs are 2-word-representable. Thus, being a subclass of distance-hereditary graphs, the double-arborescences are 2-word-representable. Hence, if G is a double-arborescence on n vertices, it is evident that $\ell(G) \leq 2n$.

Notation 5.1.13. In what follows, for a comparability graph $G = (V, E)$ with a transitive orientation O, we use the following notations. For $a \in V$, the upper set of a, $U(a) = \{b \in V \mid \overrightarrow{ab} \in O\}$ and $p_{\overline{a}}, p'_{\overline{a}}, q_{\overline{a}}, q'_{\overline{a}}$ denote permutations on $U(a)$. Further, the lower set of a, $L(a) = \{b \in V \mid \overrightarrow{ba} \in O\}$ and $p_{\underline{a}}, p'_{\underline{a}}, q_{\underline{a}}, q'_{\underline{a}}$ denote permutations on $L(a)$. A permutation on the empty set is considered to be the empty word.

Algorithm

Let $G = (V, E)$ be an arborescence on n vertices with clique number κ. Note that a maximum size clique in a comparability graph can be found in quadratic time [Even et al., 1972]. Suppose C is a maximum clique in G. As G is a treelike comparability graph, by Theorem 1.6.18, the treelike orientation of G as well as the transitive reduction can be found in linear time. Suppose $\tilde{G} = (V, \tilde{O})$ is the transitive reduction of the treelike orientation O of G. Recall that $\tilde{G}$ is a directed rooted tree and the root r is the source or the sink of O. Note that the elements of C form a longest directed path in $\tilde{G}$ and $r \in C$.

In Algorithm 3, we give a procedure based on breadth-first search (BFS) on the underlying tree of $\tilde{G}$ to construct a word of minimum length representing G. For

BFS, one may refer to [Cormen et al., 2009].

Algorithm 3: Constructing minimum-word-representant of an arborescence.

Input: The transitive reduction $\tilde{G}$ of an arborescence G and a longest directed path C in $\tilde{G}$.
Output: A word w representing G.

Initialize w with the root r.
Append r in Q (where Q is the queue used in BFS)
while Q *is not empty* **do**
  Remove the first element of Q, say a.
  if a *is not a leaf* **then**
    Let $c_1, c_2, \ldots, c_t$ be the children of a and append them in Q.
    if $a \in C$ **then**
      Without loss of generality let $c_1 \in C$ and update w by $c_1c_2 \cdots c_t w c_t c_{t-1} \cdots c_2$.
    end
    else
      Replace the first occurrence of a in w by $c_1c_2 \cdots c_t a$ and the second occurrence of a in w by $c_t c_{t-1} \cdots c_1 a$.
    end
  end
end
return w.

Notation 5.1.14. In what follows, w refers to the output of Algorithm 3.

We have the following properties of w.

Remark 5.1.15. If G is an arborescence on n vertices with clique number κ, the length of the word w is $2n - \kappa$, as each element of C appears exactly once and the remaining elements of V appear twice in w.

Remark 5.1.16.

(i) If r is the source in O, then $V = \{r\} \cup U(r)$ and the word w is of the form $p_{\overline{r}} r u$, where u is a subword of w such that no element of C appears in u, and each element not in C has one occurrence in $p_{\overline{r}}$ and the other occurrence in u. Moreover, for any vertex $a \in U(r)$, $p_{\overline{a}} a \ll_f p_{\overline{r}} \ll_f w$.

(ii) If r is the sink in O, then $V = \{r\} \cup L(r)$ and the word w is of the form $q_{\underline{r}} r v$, where v is a subword of w such that no element of C appears in v, and each element not in C has one occurrence in $q_{\underline{r}}$ and the other occurrence in v. Moreover, for any vertex $a \in L(r)$, $q_{\underline{a}} a \ll_f q_{\underline{r}} \ll_f w$.

Proposition 5.1.17. *Let r be the source in O and $(r =)a_1 \to a_2 \to \cdots \to a_s$ be a directed path in $\tilde{G}$, for some integer $s \geq 2$. Then, for all $1 \leq i \leq s-1$, $p_{\overline{a_{i+1}}} a_{i+1} \ll_f p_{\overline{a_i}} \ll_f p_{\overline{a_1}} \ll_f w$.*

Proof. In view of Remark 5.1.16(i), we have $p_{\overline{a_2}} a_2 \ll_f p_{\overline{a_1}} \ll_f w$. Thus, the result is true for $i = 1$. Suppose for $2 \leq i = t < s-1$, we have $p_{\overline{a_{t+1}}} a_{t+1} \ll_f p_{\overline{a_t}} \ll_f p_{\overline{a_1}} \ll_f w$. We show that the result is true for $i = t+1$. Since $\{a_{t+2}\} \cup U(a_{t+2}) \subseteq U(a_{t+1})$, exactly one occurrence of each element of $U(a_{t+2}) \cup \{a_{t+2}\}$ appears in $p_{\overline{a_{t+1}}}$. Further, since $p_{\overline{a_{t+2}}} a_{t+2} \ll_f p_{\overline{a_1}}$ (by Remark 5.1.16(i)) and $p_{\overline{a_{t+1}}} \ll_f p_{\overline{a_1}}$, we have $p_{\overline{a_{t+2}}} a_{t+2} \ll_f p_{\overline{a_{t+1}}} \ll_f p_{\overline{a_1}}$. Thus, the result is true for $i = t+1$. Hence, for all $1 \leq i \leq s-1$, the result follows. □

Similarly, if r is the sink in O, we have the following result.

Proposition 5.1.18. *Let r be the sink in O and $a_s \to a_{s-1} \to \cdots a_2 \to a_1(= r)$ be a directed path in $\tilde{G}$, for some integer $s \geq 2$. Then, for all $1 \leq i \leq s-1$, $q_{\underline{a_{i+1}}} a_{i+1} \ll_f q_{\underline{a_i}} \ll_f q_{\underline{a_1}} \ll_f w$.*

Proof. The result follows from the proof of Proposition 5.1.17 upon replacing $p_{\overline{a_i}}$ with $q_{\underline{a_i}}$ and $U(a_i)$ with $L(a_i)$ throughout the argument. □

Correctness

We now prove that the word w represents the arborescence G through the following lemmas.

Lemma 5.1.19. *For $a, b \in V$, if a and b are adjacent in G, then a and b alternate in w.*

Proof. Without loss of generality, suppose $\overrightarrow{ba} \in O$. Let $b = a_s \to a_{s-1} \to \cdots \to a_0 = a$ be the path from b to a in $\tilde{G}$. Suppose the root r is the sink in O (if r is the source in O, the proof will be similar). Thus, according to the construction of w, a is visited first then $a_1, a_2, \ldots, a_{s-1}$ and lastly b is visited.

- Case 1: Suppose $a \notin C$. Then $a_i \notin C$ for all $0 \leq i \leq s$ and consequently all of them occur twice in w. For $1 \leq t \leq s$, using induction on t, we show that $w|_{\{a,a_t\}} = a_t a a_t a$. Hence, in particular for $t = s$, $w|_{\{a,b\}} = baba$ so that a and b alternate in w.

  For $t = 1$, since a_1 is a child of a, as per line 11 of Algorithm 3, we have $a_1 a a_1 a \ll w$ and both a and a_1 appear exactly twice in w. Hence, $w|_{\{a,a_1\}} = a_1 a a_1 a$. For $t = p$ suppose $w|_{\{a,a_p\}} = a_p a a_p a$. For $t = p + 1$, since a_{p+1} is a child of a_p, both the occurrences of a_p in w are replaced by a word containing $a_{p+1}a_p$ as a subword. Hence, $a_{p+1}a_p a a_{p+1} a_p a \ll w$ so that $w|_{\{a,a_{p+1}\}} = a_{p+1} a a_{p+1} a$.

- Case 2: Suppose $a \in C$. If b is also in C, then both a and b occur exactly once in w so that they alternate in w. If $b \notin C$, then let m be the smallest possible index with $1 \leq m \leq s$ such that $a_m \notin C$. For $m \leq t \leq s$, using induction on t, we show that $w|_{\{a,a_t\}} = a_t a a_t$. Hence, for $t = s$, we have a and b alternate in w.

  Since $a_{m-1} \in C$, note that a_{m-1} appears exactly once in w. If $m = 1$, clearly $a_m a a_m \ll w$. For $m > 1$, note that $w|_{\{a,a_{m-1}\}} = a_{m-1}a$. Since a_m is a child of a_{m-1}, in view of line 7 of Algorithm 3, a_m appears twice in w such that $a_m a_{m-1} a a_m \ll w$. Thus, for $t = m$, $w|_{\{a,a_m\}} = a_m a a_m$. For $t = m + p$ suppose $w|_{\{a,a_{m+p}\}} = a_{m+p} a a_{m+p}$. For $t = m + p + 1$, since a_{m+p+1} is a child of a_{m+p}, both the occurrences of a_{m+p} in w are replaced by a word containing $a_{m+p+1}a_{m+p}$ as a subword. Hence, $a_{m+p+1}a_{m+p} a a_{m+p+1} a_{m+p} \ll w$ so that $w|_{\{a,a_{m+p+1}\}} = a_{m+p+1} a a_{m+p+1}$.

Hence, in any case, a and b alternate in w. □

Lemma 5.1.20. *For $a, b \in V$, if a and b are not adjacent in G, then they do not alternate in w.*

Proof. Suppose the root r is the sink in O (if r is the source in O, the proof will be similar). Since a and b are non-adjacent in $\tilde{G}$ and r is the sink in $\tilde{O}$, there exists c in G such that there are two disjoint directed paths from a to c and from b to c in $\tilde{G}$. Let $a = a_s \to a_{s-1} \to \cdots \to a_1 \to c$ and $b = b_{s'} \to b_{s'-1} \to \cdots \to b_1 \to c$ be the directed paths in $\tilde{G}$. Hence, c is visited first and then a_i, b_j are visited in Algorithm 3. Note that a_1, b_1 are children of c and let a_1 is visited before b_1 in the algorithm.

- Case 1: One of a and b is in C but the other is not; say, $a \in C$ and $b \notin C$. Then $c \in C$, $a_i \in C$, for all $1 \leq i \leq s$, and $b_j \notin C$, for all $1 \leq j \leq s'$. For $1 \leq t \leq s$, using induction on t, we first show that $w|_{\{a_t, b_1\}} = a_t b_1 b_1$.

  For $t = 1$, since a_1, b_1 are children of c and $a_1 \in C$, as per the algorithm, we have $a_1 b_1 c b_1 \ll w$. Hence, $w|_{\{a_1, b_1\}} = a_1 b_1 b_1$. For $t = p$, suppose $w|_{\{a_p, b_1\}} = a_p b_1 b_1$. For $t = p + 1$, since a_{p+1} is a child of a_p and both a_p and $a_{p+1} \in C$, as per the algorithm, we have $a_{p+1} a_p b_1 b_1 \ll w$ so that $w|_{\{a_{p+1}, b_1\}} = a_{p+1} b_1 b_1$. Hence, in particular for $t = s$, we have $w|_{\{a, b_1\}} = a b_1 b_1$.

  Now for $1 \leq t \leq s'$, using induction on t, we show that $w|_{\{a, b_t\}} = a b_t b_t$. For $t = 1$, we have $w|_{\{a, b_1\}} = a b_1 b_1$. For $t = p$, suppose $w|_{\{a, b_p\}} = a b_p b_p$. For $t = p+1$, since b_{p+1} is a child of b_p, both the occurrences of b_p in w are replaced by a word containing $b_{p+1} b_p$ as a subword. Thus, we have $a b_{p+1} b_p b_{p+1} b_p \ll w$ so that $w|_{\{a, b_{p+1}\}} = a b_{p+1} b_{p+1}$. Hence, in particular for $t = s'$, we have $w|_{\{a, b\}} = abb$ so that a and b do not alternate in w.

- Case 2: Both a and b are not in C. Then we have the following cases.

- Subcase 2.1: Suppose $a_1 \notin C$ and $b_1 \notin C$. Then note that $a_i \notin C$ for all $1 \leq i \leq s$ and $b_j \notin C$ for all $1 \leq j \leq s'$.

  For $1 \leq t \leq s$, using induction on t, we first show that $w|_{\{a_t,b_1\}} = a_tb_1b_1a_t$. Note that a_1, b_1 are children of c and both a_1 and b_1 are not in C. Hence, if $c \in C$, as per the algorithm, we have $a_1b_1cb_1a_1 \ll w$ and if $c \notin C$, as per the algorithm, we have $a_1b_1cb_1a_1c \ll w$. Hence, in any case, for $t = 1$, we have $w|_{\{a_1,b_1\}} = a_1b_1b_1a_1$. For $t = p$, suppose $w|_{\{a_p,b_1\}} = a_pb_1b_1a_p$. For $t = p+1$, since a_{p+1} is a child of a_p, both the occurrences of a_p in w are replaced by a word containing $a_{p+1}a_p$ as a subword. Thus, we have $a_{p+1}a_pb_1b_1a_{p+1}a_p \ll w$ so that $w|_{\{a_{p+1},b_1\}} = a_{p+1}b_1b_1a_{p+1}$. Hence, in particular for $t = s$, we have $w|_{\{a,b_1\}} = ab_1b_1a$.

  Now for $1 \leq t \leq s'$, using induction on t, we show that $w|_{\{a,b_t\}} = ab_tb_ta$. For $t = 1$, we have $w|_{\{a,b_1\}} = ab_1b_1a$. For $t = p$, suppose $w|_{\{a,b_p\}} = ab_pb_pa$. For $t = p+1$, since b_{p+1} is a child of b_p and both the occurrences of b_p in w are replaced by a word containing $b_{p+1}b_p$ as a subword, we have $ab_{p+1}b_pb_{p+1}b_pa \ll w$ so that $w|_{\{a,b_{p+1}\}} = ab_{p+1}b_{p+1}a$. Hence, in particular for $t = s'$, we have $w|_{\{a,b\}} = abba$ so that a and b do not alternate in w.

- Subcase 2.2: One of a_1 and b_1 is in C but not the other; say, $a_1 \in C$ and $b_1 \notin C$. Then $c \in C$ and $b_j \notin C$ for all $1 \leq j \leq s'$. Let $m(> 1)$ be the smallest positive integer such that $a_m \notin C$. Therefore $a_i \notin C$ for all $m \leq i \leq s$.

  For $m \leq t \leq s$, using induction on t, we show that $w|_{\{a_t,b\}} = a_tbba_t$. Hence, in particular for $t = s$, we have $w|_{\{a,b\}} = abba$. As shown in Case 1, we can have $w|_{\{a_{m-1},b\}} = a_{m-1}bb$. For $t = m$, since a_m is a child of a_{m-1} but $a_m \notin C$, as per the algorithm, we have $a_ma_{m-1}bba_m \ll w$. Thus $w|_{\{a_m,b\}} = a_mbba_m$. For $t = m+p$, suppose $w|_{\{a_{m+p},b\}} = a_{m+p}bba_{m+p}$. For $t = m+p+1$, as a_{m+p+1} is a child of a_{m+p} and both the occurrences of a_{m+p} in w are replaced by a word containing $a_{m+p+1}a_{m+p}$ as a subword, we have $a_{m+p+1}a_{m+p}bba_{m+p+1}a_{m+p} \ll w$.

Thus $w|_{\{a_{m+p+1},b\}} = a_{m+p+1}bba_{m+p+1}$. Hence in particular for $t = s$, we have $w|_{\{a,b\}} = abba$.

Hence in any case a and b do not alternate in w. $\square$

We now conclude in the following theorem that the word w produced by Algorithm 3 on an arborescence G is a minimum-word-representant of G.

Theorem 5.1.21. *If G is an arborescence on n vertices with clique number κ, then $\ell(G) = 2n - \kappa$.*

Proof. Let w be the word obtained by Algorithm 3 on an arborescence G. From Lemmas 5.1.19 and 5.1.20, it is evident that w represents G. In view of Remark 5.1.15, the length of the word w is $2n - \kappa$ so that $\ell(G) \leq 2n - \kappa$. Further, as the size of a maximum clique in G is κ, by Theorem 1.3.32, we have $2n - \kappa \leq \ell(G)$. Hence, w is a minimum-word-representant of G and $\ell(G) = 2n - \kappa$. $\square$

We now prove a more general theorem for the minimum-word-representants of double-arborescences.

Theorem 5.1.22. *Suppose G is a double-arborescence on n vertices with clique number κ. Then $\ell(G) = 2n - \kappa$.*

Proof. Let r be a root of a double-arborescence $G = (V, E)$ and $\tilde{G} = (V, \tilde{O})$ be the transitive reduction of the double-arborescence orientation O of G. If C is a longest directed path in $\tilde{G}$, then note that the size of C is κ.

Let $A = \{r\} \cup L(r)$ and $B = \{r\} \cup U(r)$ so that $A \cap B = \{r\}$ and $A \cup B = V$. Consider the restrictions O_A, O_B of O and $\tilde{O}_A, \tilde{O}_B$ of $\tilde{O}$ on the vertex sets A and B, respectively. Then both $\tilde{G}_A = (A, \tilde{O}_A)$ and $\tilde{G}_B = (B, \tilde{O}_B)$ are directed rooted trees with root r. Note that r is the sink in $\tilde{O}_A$ and it is the source in $\tilde{O}_B$. Let G_A and G_B be the induced subgraphs of G on the vertex sets A and B, respectively. Thus, both G_A and G_B are arborescences.

Let $C^A = C \cap A$ and $C^B = C \cap B$ so that C^A and C^B are longest directed paths in $\tilde{G}_A$ and $\tilde{G}_B$, respectively. We run Algorithm 3 separately on $\tilde{G}_A$ and $\tilde{G}_B$ by considering the respective longest directed paths C^A and C^B. Let u and v be minimum-word-representants of G_A and G_B, respectively, obtained from Algorithm 3. Suppose that $u = q_{\underline{r}} r u_1$ and $v = p_{\overline{r}} r v_1$. Note that the reversal of u, $u^R = u_1^R r q_{\underline{r}}^R$ also represents G_A (cf. Proposition 1.3.19). We show that the word $w' = u_1^R p_{\overline{r}} r q_{\underline{r}}^R v_1$ represents the graph G.

First note that $w'|_A = u_1^R r q_{\underline{r}}^R$. Hence any two vertices a and a' of G_A are adjacent in G if and only if a and a' alternate in w'. Also $w'|_B = p_{\overline{r}} r v_1$ so that any two vertices b and b' of G_B are adjacent in G if and only if they alternate in w'. We now show that r alternates with any $a \in V \setminus \{r\}$ in w' in the following cases:

- $a \in A \setminus \{r\}$ such that $a \in C^A$: Then a occurs exactly once in u^R as well as in w'. Moreover, since r occurs exactly once in w', we have a and r alternate in w'. Similarly, if $a \in B \setminus \{r\}$ such that $a \in C^B$, then both a and r occur exactly once in w' so that they alternate in w'.

- $a \in A \setminus \{r\}$ such that $a \notin C^A$: Then a occurs exactly twice in u^R as well as in w'. Further, in view of Remark 5.1.16, we have $w'|_{\{a,r\}} = ara$. Similarly, when $a \in B \setminus \{r\}$ but $a \notin C^B$, then $w'|_{\{a,r\}} = ara$.

Finally, let $a \in A \setminus \{r\}$ and $b \in B \setminus \{r\}$. In view of Remark 5.1.16, the positions of a's in $q_{\underline{r}}$ and u_1 and the positions of b's in $p_{\overline{r}}$ and v_1 are evident as shown in the following cases so that a and b alternate in w'.

- If $a \in C^A$ and $b \in C^B$, then $w'|_{\{a,b\}} = ba$.

- If $a \in C^A$ and $b \notin C^B$, then $w'|_{\{a,b\}} = bab$.

- If $a \notin C^A$ and $b \in C^B$, then $w'|_{\{a,b\}} = aba$.

- If $a \notin C^A$ and $b \notin C^B$, then $w'|_{\{a,b\}} = abab$.

Hence, the word w' represents the graph G. Note that the elements of C appear exactly once in w' while the rest of the elements of V appear twice in w' so that the length of the word w' is $2n - \kappa$. Further, since the size of a maximum clique in G is κ, we have $\ell(G) = 2n - \kappa$. □

Further, based on the construction, we present some structural properties of the minimum-word-representant $w' = u_1^R p_{\bar{r}} r q_{\underline{r}}^R v_1$ of the double-arborescence G. We use these properties in Section 5.2 to construct minimum-word-representants for treelike permutation graphs.

Remark 5.1.23. The word w' is of the form $v_2 p_{\bar{r}} r p_{\underline{r}} v_1$, where $v_2 = u_1^R$ and $p_{\underline{r}} = q_{\underline{r}}^R$ is a permutation on $L(r)$. Moreover, we have the following properties of $p_{\bar{r}}$ and $p_{\underline{r}}$.

(i) Since $p_{\bar{r}} r v_1$ is a minimum-word-representant of the arborescence G_B, by Remark 5.1.16(i), if $a \in U(r)$, then $p_{\bar{a}} a \ll_f p_{\bar{r}} \ll_f w'$.

(ii) Since $q_{\underline{r}} r u_1$ is a minimum-word-representant of the arborescence G_A, by Remark 5.1.16(ii), if $a \in L(r)$, then $q_{\underline{a}} a \ll_f q_{\underline{r}}$ so that $a p_{\underline{a}} \ll_f p_{\underline{r}} \ll_f w'$, where $(q_{\underline{a}} a)^R = a p_{\underline{a}}$.

Remark 5.1.24. Let a and b be two distinct vertices of the double-arborescence G such that $\overrightarrow{ra} \in \tilde{O}$ and $\overrightarrow{rb} \in \tilde{O}$. Thus, a and b are non-adjacent in G and $a, b \in U(r)$. Then, in view of Remark 5.1.23(i), we have $p_{\bar{a}} a \ll_f p_{\bar{r}}$ and $p_{\bar{b}} b \ll_f p_{\bar{r}}$. Since $U(a) \cap U(b) = \varnothing$, $p_{\bar{r}} = x_1 p_{\bar{a}} a x_2 p_{\bar{b}} b x_3$ (if a appears before b in $p_{\bar{r}}$) or $p_{\bar{r}} = x_1 p_{\bar{b}} b x_2 p_{\bar{a}} a x_3$ (if b appears before a in $p_{\bar{r}}$), for some words x_1, x_2, x_3.

Remark 5.1.25. Let a and b be two distinct elements of the double-arborescence G such that $\overrightarrow{ar} \in \tilde{O}$ and $\overrightarrow{br} \in \tilde{O}$. Then, in view of Remark 5.1.23(ii), we have $p_{\underline{r}} = y_1 a p_{\underline{a}} y_2 b p_{\underline{b}} y_3$ (if a appears before b in $p_{\underline{r}}$) or $p_{\underline{r}} = y_1 b p_{\underline{b}} y_2 a p_{\underline{a}} y_3$ (if b appears before a in $p_{\underline{r}}$), for some words y_1, y_2, y_3.

Remark 5.1.26. In view of Propositions 1 and 2, and Remark 5.1.23, we have the following properties of the word w'.

(i) Let $(r =)a_1 \to a_2 \to \cdots \to a_s$ be a directed path in $\tilde{G}$, for some integer $s \geq 2$. Then, for all $1 \leq i \leq s-1$, $p_{\overline{a_{i+1}}} a_{i+1} \ll_f p_{\overline{a_i}} \ll_f p_{\overline{a_1}} \ll_f w'$.

(ii) Let $a_s \to a_{s-1} \cdots \to a_2 \to a_1(= r)$ be a directed path in $\tilde{G}$, for some integer $s \geq 2$. Then, for all $1 \leq i \leq s-1$, $a_{i+1} p_{\underline{a_{i+1}}} \ll_f p_{\underline{a_i}} \ll_f p_{\underline{a_1}} \ll_f w'$.

5.2 Treelike Permutation Graphs

This section consists of two subsections. In Subsection 5.2.2, we give a polynomial-time algorithm to construct a minimum-word-representant of a treelike permutation graph, which uses permutations representing arborescences. While the literature has various methods for constructing permutations representing a permutation graph (cf. [Golumbic, 2004; McConnell and Spinrad, 1999]), we construct a specific type of permutations for arborescences in Subsection 5.2.1, which are useful in Subsection 5.2.2.

5.2.1 Permutations for an Arborescence

Let $G = (V, E)$ be an arborescence with root r in the treelike orientation O of G. Let $\tilde{G} = (V, \tilde{O})$ be the transitive reduction of O. As G is a treelike comparability graph, by Theorem 1.6.18, both O and $\tilde{G}$ can be found in linear time. In Algorithm 4, we give a procedure based on breadth-first search (BFS) on the underlying tree of $\tilde{G}$ to construct two permutations q and q' such that the word qq' represents G.

Remark 5.2.1. Algorithm 4 runs in $O(n^2)$ time, where n is the number of vertices of G. Note that, in each iteration of total n iterations, testing condition in step 4 and updating q and q' in step 6 can be done in $O(n)$ time.

Remark 5.2.2. We have the following properties of q and q':

(i) If r is the sink in O, the permutations q and q' are of the form $q = q_{\underline{r}} r$ and $q' = q'_{\underline{r}} r$ such that the word $q_{\underline{r}} q'_{\underline{r}}$ represents the induced subgraph $G[L(r)]$.

Algorithm 4: Constructing two permutations for an arborescence.

Input: The transitive reduction $\tilde{G} = (V, \tilde{O})$ of the treelike orientation O of an arborescence $G = (V, E)$.
Output: Two permutations q and q' on V.

Initialize $q = r$, $q' = r$ and Q with r (where Q is a queue as in BFS).
while *Q is not empty* **do**
    Remove the first element of Q, say a.
    if *a is not a leaf* **then**
        Let $c_1, c_2, \ldots, c_m$ be the children of a and append them in Q.
        Replace a in q by $c_1c_2\cdots c_m a$ and in q' by $c_m c_{m-1}\cdots c_1 a$.
    end
end
return q and q'.

(ii) Moreover, if $a \in L(r)$, then $p_{\underline{a}}a \ll_f q_{\underline{r}}$ and $p'_{\underline{a}}a \ll_f q'_{\underline{r}}$ such that $p_{\underline{a}}p'_{\underline{a}}$ represents the induced subgraph $G[L(a)]$.

(iii) If r is the source in O, the permutations q and q' are of the form $q = q_{\overline{r}}r$ and $q' = q'_{\overline{r}}r$, and for $a \in U(r)$, we have $p_{\overline{a}}a \ll_f q_{\overline{r}}$ and $p'_{\overline{a}}a \ll_f q'_{\overline{r}}$ such that $q_{\overline{r}}q'_{\overline{r}}$ and $p_{\overline{a}}p'_{\overline{a}}$ represent $G[U(r)]$ and $G[U(a)]$, respectively.

Remark 5.2.3. In view of Remark 5.2.2, we further observe the following.

(i) Let r be the sink in O and $a_s \to a_{s-1} \to \cdots a_2 \to a_1 (= r)$ be a directed path in $\tilde{G}$, for some integer $s \geq 2$. Then, for all $1 \leq i \leq s-1$, $p_{\underline{a_{i+1}}}a_{i+1} \ll_f p_{\underline{a_i}} \ll_f q_{\underline{a_1}}$ and $p'_{\underline{a_{i+1}}}a_{i+1} \ll_f p'_{\underline{a_i}} \ll_f q'_{\underline{a_1}}$, where for $1 \leq i \leq s$, the word $p_{\underline{a_i}}p'_{\underline{a_i}}$ represents the induced subgraph $G[L(a_i)]$.

(ii) Let r be the source in O and $(r =)a_1 \to a_2 \to \cdots \to a_s$ be a directed path in $\tilde{G}$, for some integer $s \geq 2$. Then, for all $1 \leq i \leq s-1$, $p_{\overline{a_{i+1}}}a_{i+1} \ll_f p_{\overline{a_i}} \ll_f q_{\overline{a_1}}$ and $p'_{\overline{a_{i+1}}}a_{i+1} \ll_f p'_{\overline{a_i}} \ll_f q'_{\overline{a_1}}$, where for $1 \leq i \leq s$, the word $p_{\overline{a_i}}p'_{\overline{a_i}}$ represents the induced subgraph $G[U(a_i)]$.

Lemma 5.2.4. *For $a, b \in V$, if a and b are adjacent in G, then a and b alternate in the word qq'.*

Proof. Without loss of generality, suppose $\overrightarrow{ba} \in O$. Let $(b =)a_s \rightarrow a_{s-1} \rightarrow \cdots \rightarrow a_0(= a)$ be the directed path from b to a in $\tilde{G}$. Suppose the root r is the sink in O (if r is the source in O, the proof will be similar). As per Algorithm 4, a is visited first, then $a_1, a_2, \ldots, a_{s-1}$ and lastly b is visited. For $1 \leq i \leq s$, using induction on i, we show that $a_i a \ll q$ and $a_i a \ll q'$. Then, in particular for $i = s$, we have $ba \ll q$ and $ba \ll q'$ so that a and b alternate in qq'.

For $i = 1$, since a_1 is a child of a, as per line 6, we have $a_1 a \ll q$ and $a_1 a \ll q'$. Hence, a_1 and a alternate in the word qq'. For $i = t$ suppose $a_t a \ll q$ and $a_t a \ll q'$. For $i = t + 1$, since a_{t+1} is a child of a_t, a_t is replaced in both q and q' by a word containing $a_{t+1}a_t$ as a subword. Thus, we have $a_{t+1}a_t a \ll q$ and $a_{t+1}a_t a \ll q'$ so that $a_{t+1}a \ll q$ and $a_{t+1}a \ll q'$. Thus, $qq'|_{\{a,b\}} = baba$ so that a and b alternate in qq'. □

Lemma 5.2.5. *For $a, b \in V$, if a and b are not adjacent in G, then they do not alternate in the word qq'.*

Proof. Suppose the root r is the sink in O (if r is the source in O, the proof will be similar). Since a and b are non-adjacent in $\tilde{G}$, and r is the sink in $\tilde{O}$, there exists a vertex c in G such that there are two disjoint directed paths from a to c, and b to c in $\tilde{G}$. Let $(a =)a_s \rightarrow a_{s-1} \rightarrow \cdots \rightarrow a_1 \rightarrow c$, and $(b =)b_{s'} \rightarrow b_{s'-1} \rightarrow \cdots \rightarrow b_1 \rightarrow c$ be the directed paths in $\tilde{G}$. Note that a_1 and b_1 are children of c.

Without loss of generality, suppose $a_1 b_1 \ll q$. For $1 \leq i \leq s$, using induction on i, we show that $a_t b_1 \ll q$ and $b_1 a_t \ll q'$. For $i = 1$, since a_1, b_1 are children of c, by line 6, we have $a_1 b_1 \ll a_1 b_1 c \ll q$ and $b_1 a_1 \ll b_1 a_1 c \ll q'$. For $i = t$, suppose $a_t b_1 \ll q$ and $b_1 a_t \ll q'$. For $i = t + 1$, since a_{t+1} is a child of a_t, a_t is replaced in both q and q' by a word containing $a_{t+1}a_t$ as a subword. Thus, we have $a_{t+1}a_t b_1 \ll q$ and $b_1 a_{t+1} a_t \ll q'$. Hence, $a_{t+1}b_1 \ll q$ and $b_1 a_{t+1} \ll q'$ so that in particular for $i = s$ we have, $ab_1 \ll q$ and $b_1 a \ll q'$.

Now for $1 \leq i \leq s'$, using induction on i, we show that $ab_t \ll q$ and $b_t a \ll q'$. For $i = 1$, we have $ab_1 \ll q$ and $b_1 a \ll q'$. For $i = t$, suppose $ab_t \ll q$ and $b_t a \ll q'$. For

$i = t+1$, since b_{t+1} is a child of b_t, b_t is replaced in both q and q' by a word containing $b_{t+1}b_t$ as a subword. Thus, we have $ab_{t+1} \ll ab_{t+1}b_t \ll q$ and $b_{t+1}a \ll b_{t+1}b_t a \ll q'$. Hence, in particular for $i = s'$, we have $ab \ll q$ and $ba \ll q'$ so that a and b do not alternate in qq'. $\square$

Thus, in views of Lemmas 5.2.4 and 5.2.5, we have the following theorem establishing the correctness of Algorithm 4.

Theorem 5.2.6. *The word qq' represents the arborescence G.*

5.2.2 Minimum-Word-Representants

Let $G = (V, E)$ be a treelike permutation graph (hence, a path of double-arborescences, by Theorem 1.6.27) and O be the treelike orientation of G. Suppose C is a longest directed path in the transitive reduction $\tilde{G} = (V, \tilde{O})$ of O. Note that C can be found in quadratic time [Even et al., 1972]. In this subsection, we construct a minimum-word-representant of G through Algorithm 5.

Algorithm

In Algorithm 5, we consider a smallest possible root-path in $\tilde{G}$ and the double-arborescence $G_{r_\tau} = G[\{r_\tau\} \cup U(r_\tau) \cup L(r_\tau)]$, where r_τ is the first amongst the roots that appear in C. Using the construction given in Theorem 5.1.22, we first construct a minimum-word-representant of G_{r_τ} and extend it to that of G iteratively. In each iteration, in a specific order of the roots, we include two permutations on the lower or on the upper set of each root that is already included in the word. Similar to the notation of permutations on $U(a)$ and $L(a)$, in Algorithm 5, we denote the permutations on $U^e(b)$ by $q_{\overline{b^e}}$ or $q'_{\overline{b^e}}$, and the permutations on $L^e(b)$ by $q_{\underline{b^e}}$ or $q'_{\underline{b^e}}$. A demonstration of Algorithm 5 is given in Table 5.1, in which we provide stepwise results by indicating the line number from Algorithm 5.

Algorithm 5: Construction of minimum-word-representant of a treelike permutation graph.

Input: A treelike permutation graph $G = (V, E)$, the transitive reduction $\tilde{G} = (V, \tilde{O})$ of the treelike orientation O of G, and a longest directed path C in $\tilde{G}$.

Output: A minimum-word-representant of G.

Let $r_1 - r_2 - \cdots - r_m$ be the root-path in $\tilde{G}$ such that m is the smallest possible, and r_τ be the first amongst the roots that appear in C.
Let $G_{r_\tau} = (V_{r_\tau}, E_{r_\tau})$ be the double-arborescence $G[\{r_\tau\} \cup U(r_\tau) \cup L(r_\tau)]$.
Let $w_{r_\tau} = uv_{r_\tau}u'$ be the minimum-word-representant of G_{r_τ}, constructed using C and the restriction of $\tilde{G}$ to V_{r_τ}, where $v_{r_\tau} = p_{\overline{r_\tau}} r_\tau p_{\underline{r_\tau}}$.
Set $a = r_\tau$ and consider a queue $Q = \langle r_{\tau-1}, r_{\tau-2}, \ldots, r_1, r_{\tau+1}, r_{\tau+2}, \ldots, r_m \rangle$.
while *Q is not empty* **do**
  Let $b = \text{head}(Q)$. If $b = r_{\tau+1}$, then set $a = r_\tau$, $v_a = v_{r_1}$ and $V_a = V_{r_1}$.
  if $\overrightarrow{ab} \in \tilde{O}$ **then**
    Let $L^e(b) = L(b) \setminus (L(a) \cup \{a\})$.
    Using Algorithm 4, construct two permutations q and q' for the arborescence $G[\{b\} \cup L^e(b)]$ with $\tilde{G}$ restricted to the arborescence.
    Let $q_{\underline{b^e}}$ and $q'_{\underline{b^e}}$ be the permutations obtained from q and q' by deleting b so that $q_{\underline{b^e}} q'_{\underline{b^e}}$ represents $G[L^e(b)]$.
    Let v_b be the word obtained by replacing a factor $p_{\bar{b}} b$ in v_a with the word $q_{\underline{b^e}} p_{\bar{b}} b q'_{\underline{b^e}}$.
    Set $w_b = uv_b u'$ and $V_b = V_a \cup L^e(b)$.
  end
  else
    Let $U^e(b) = U(b) \setminus (U(a) \cup \{a\})$.
    Using Algorithm 4, construct two permutations q and q' for the arborescence $G[\{b\} \cup U^e(b)]$ with $\tilde{G}$ restricted to the arborescence.
    Let $q_{\overline{b^e}}$ and $q'_{\overline{b^e}}$ be the permutations obtained from q and q' by deleting b so that $q_{\overline{b^e}} q'_{\overline{b^e}}$ represents $G[U^e(b)]$.
    Let v_b be the word obtained by replacing a factor $p_{\underline{b}} b$ or $b p_{\underline{b}}$ (whichever is available) in v_a with the word $q_{\overline{b^e}} p_{\underline{b}} b q'_{\overline{b^e}}$ or $q_{\overline{b^e}} b p_{\underline{b}} q'_{\overline{b^e}}$, respectively, as appropriate.
    Set $w_b = uv_b u'$ and $V_b = V_a \cup U^e(b)$.
  end
  Remove head(Q) from Q; and update a by b.
end
return w_{r_m}

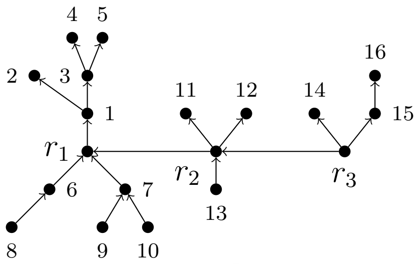


FIGURE 5.4: A path of double-arborescences G

Input: The transitive reduction $\tilde{G} = (V, \tilde{O})$ depicted in Figure 5.4 of a graph $G = (V, E)$ and a longest directed path $C : 13 \to r_2 \to r_1 \to 1 \to 3 \to 4$ in $\tilde{G}$.
1. $r_1 \leftarrow r_2 \leftarrow r_3$ is the root-path in $\tilde{G}$ and $r_\tau = r_2$. 2. $G_{r_2} = G[\{r_2\} \cup U(r_2) \cup L(r_2)]$. 3. $w_{r_2} = uv_{r_2}u'$, where $u = r_3$, $u' = (12)(11)25$ and $v_{r_2} = p_{\overline{r_2}} r_2 p_{\underline{r_2}}$, where $p_{\overline{r_2}} = 45321r_1(11)(12)$ and $p_{\underline{r_2}} = r_3(13)$. 4. $a = r_2$ and $Q = \langle r_1, r_3 \rangle$.
First iteration
6. $b = r_1$ 7. $\overrightarrow{r_2r_1} \in \tilde{O}$ 8. $L^e(r_1) = \{6, 7, 8, 9, 10\}$ 9. $q = 869(10)7r_1$, $q' = (10)9786r_1$ 10. $q_{\underline{r_1^e}} = 869(10)7$, $q'_{\underline{r_1^e}} = (10)9786$ 11. $v_{r_1} = 869(10)745321r_1(10)9786\ (11)(12)r_2r_3(13)$ 12. $w_{r_1} = r_3v_{r_1}(12)(11)25$
Second iteration
6. $b = r_3$, $a = r_2$, $v_a = v_{r_1}$ 14. $\overrightarrow{r_3r_2} \in \tilde{O}$ 15. $U^e(r_3) = \{14, 15, 16\}$ 16. $q = (14)(16)(15)r_3$, $q' = (16)(15)(14)r_3$ 17. $q_{\overline{r_3^e}} = (14)(16)(15)$, $q'_{\overline{r_3^e}} = (16)(15)(14)$ 18. $v_{r_3} = 869(10)745321r_1(10)9786(11)(12)\ r_2(14)(16)(15)r_3(16)(15)(14)(13)$ 19. Output $w_{r_3} = r_3v_{r_3}(12)(11)25$

Table 5.1: Stepwise details of implementation of Algorithm 5 on Figure 5.4

Correctness

We establish the correctness of Algorithm 5 through a series of results, as per the following. In the first iteration of Algorithm 5, the availability of the factor $p_{\overline{r_{\tau-1}}} r_{\tau-1}$

(in line 11) or the factor $r_{\tau-1}p_{\underline{r_{\tau-1}}}$ (in line 18) of the word v_{r_τ} is ensured by the following remark.

Remark 5.2.7. If $\overrightarrow{r_\tau r_{\tau-1}} \in \tilde{O}$, by Remark 5.1.23(i), we have $p_{\overline{r_{\tau-1}}}r_{\tau-1} \ll_f p_{\overline{r_\tau}} \ll_f v_{r_\tau}$. Also, in view of Remark 5.1.23(ii), if $\overrightarrow{r_{\tau-1}r_\tau} \in \tilde{O}$, then $r_{\tau-1}p_{\underline{r_{\tau-1}}} \ll_f p_{\underline{r_\tau}} \ll_f v_{r_\tau}$.

Further, from the second iteration onwards, the presence of the factors $p_{\bar{b}}b$, $p_{\underline{b}}b$, or $bp_{\underline{b}}$ in the word v_a, if $b \neq r_{\tau+1}$, is ensured by the following propositions.

Proposition 5.2.8. *Suppose v_a is obtained at the end of the i^{th} iteration and $b = head(Q)$ in the $(i+1)^{th}$ iteration such that $b \neq r_{\tau+1}$. If $\overrightarrow{ab} \in \tilde{O}$, then $p_{\bar{b}}b \ll_f v_a$.*

Proof. Note that $b \in U(a)$ and $a \neq r_\tau$. We prove that $p_{\bar{b}}b \ll_f v_a$ in various cases as per the following. If $a \in V_{r_\tau} \setminus \{r_\tau\}$, we consider the following cases.

- Case 1: $a \in U(r_\tau)$. Then, we have the portion of the root-path $r_\tau \to \cdots \to a' \to a \to b$ in $\tilde{G}$. Hence, as per line 11 of Algorithm 5, in the i^{th} iteration, the word v_a is obtained from the word $v_{a'}$ (or from v_{r_1}, if $a = r_{\tau+1}$) by replacing the factor $p_{\bar{a}}a$ of $v_{a'}$ (or of v_{r_1}) with the word $q_{\underline{a^e}}p_{\bar{a}}aq'_{\underline{a^e}}$. Thus, $p_{\bar{a}} \ll_f v_a$. Further, in view of Remark 5.1.26(i), we have $p_{\bar{b}}b \ll_f p_{\bar{a}} \ll_f v_a$.

- Case 2: $a \in L(r_\tau)$. Then, we have the portion of the root-path $b \leftarrow a \to a' \to \cdots \to r_\tau$ in $\tilde{G}$. Hence, as per line 18 of Algorithm 5, in the i^{th} iteration, the word v_a is obtained from the word $v_{a'}$ (or from v_{r_1}, if $a = r_{\tau+1}$) by replacing the factor $ap_{\underline{a}}$ or $p_{\underline{a}}a$ of $v_{a'}$ (or of v_{r_1}) with the word $q_{\overline{a^e}}ap_{\underline{a}}q'_{\overline{a^e}}$ or $q_{\overline{a^e}}p_{\underline{a}}aq'_{\overline{a^e}}$, respectively, as appropriate. Thus, in any case, $q_{\overline{a^e}} \ll_f v_a$. Further, since $\{b\} \cup U(b) \subseteq U^e(a)$, in view of Remark 5.2.2(iii), we have $p_{\bar{b}}b \ll_f q_{\overline{a^e}} \ll_f v_a$.

Suppose $a \notin V_{r_\tau}$. Then $a \notin \{r_{\tau-1}, r_{\tau+1}\}$; otherwise, a would belong to either $U(r_\tau)$ or $L(r_\tau)$. Also, note that $a \notin \{r_1, r_m\}$. Further, observe that $a \in L^e(r_s)$ or $a \in U^e(r_s)$, where $s \notin \{1, \tau, m\}$ and r_s appears before a in the queue Q. We consider the following cases.

- Case 1: $a \in L^e(r_s)$. Then, we have the portion of the root-path $b \leftarrow a \rightarrow a' \rightarrow \cdots \rightarrow r_s$ in $\tilde{G}$. Since r_s appears before a in Q, r_s is visited first, then a', a are visited in the algorithm. Hence, as per line 18 of Algorithm 5, in the i^{th} iteration, the word v_{a} is obtained from the word $v_{a'}$ by replacing the factor $p_{\underline{a}}a$ or $ap_{\underline{a}}$ of $v_{a'}$ with the word $q_{\overline{a^e}}p_{\underline{a}}aq'_{\overline{a^e}}$ or $q_{\overline{a^e}}ap_{\underline{a}}q'_{\overline{a^e}}$, respectively. Thus, in any case, $q_{\overline{a^e}} \ll_f v_{a}$. Since $\{b\} \cup U(b) \subseteq U^e(a)$, in view of Remark 5.2.2(iii), we have $p_{\bar{b}}b \ll_f q_{\overline{a^e}} \ll_f v_{a}$.

- Case 2: $a \in U^e(r_s)$. Then, we have the portion of the root-path $r_s \rightarrow \cdots \rightarrow a' \rightarrow a \rightarrow b$ in $\tilde{G}$. Thus, as per line 11 of Algorithm 5, in the i^{th} iteration, the word v_{a} is obtained from the word $v_{a'}$ by replacing the factor $p_{\bar{a}}a$ of $v_{a'}$ with the word $q_{\underline{a^e}}p_{\bar{a}}aq'_{\underline{a^e}}$. In view of Remark 5.2.3(ii), we have $p_{\bar{b}}b \ll_f p_{\bar{a}} \ll_f v_{a}$.

Hence, in any case, we have $p_{\bar{b}}b \ll_f v_{a}$. ☐

Proposition 5.2.9. *Suppose v_{a} is obtained at the end of the i^{th} iteration. If $(r_{\tau+1} \neq)$ $b = head(Q)$ in the $(i+1)^{th}$ iteration and $\overrightarrow{ba} \in \tilde{O}$, then $p_{\underline{b}}b \ll_f v_{a}$ or $bp_{\underline{b}} \ll_f v_{a}$.*

Proof. Note that $b \in L(a)$ and $a \neq r_\tau$. We prove that either $p_{\underline{b}}b$ or $bp_{\underline{b}}$ is a factor of v_{a} in various cases. If $a \in V_{r_\tau} \setminus \{r_\tau\}$, we consider the following cases.

- Case 1: $a \in U(r_\tau)$. Then, we have the portion of the root-path $r_\tau \rightarrow \cdots \rightarrow a' \rightarrow a \leftarrow b$ in $\tilde{G}$. Thus, as per line 11 of Algorithm 5, in the i^{th} iteration, the word v_{a} is obtained from the word $v_{a'}$ (or from v_{r_1}, if $a = r_{\tau+1}$) by replacing the factor $p_{\bar{a}}a$ of $v_{a'}$ (or of v_{r_1}) with the word $q_{\underline{a^e}}p_{\bar{a}}aq'_{\underline{a^e}}$ so that $q_{\underline{a^e}} \ll_f v_{a}$. Since $\{b\} \cup L(b) \subseteq L^e(a)$, by Remark 5.2.2(ii), we have $p_{\underline{b}}b \ll_f q_{\underline{a^e}} \ll_f v_{a}$.

- Case 2: $a \in L(r_\tau)$. Then, we have the portion of the root-path $b \rightarrow a \rightarrow a' \rightarrow \cdots \rightarrow r_\tau$ in $\tilde{G}$. Hence, as per line 18 of Algorithm 5, in the i^{th} iteration, the word v_{a} is obtained from the word $v_{a'}$ (or from v_{r_1}, if $a = r_{\tau+1}$) by replacing the factor $ap_{\underline{a}}$ or $p_{\underline{a}}a$ of $v_{a'}$ (or of v_{r_1}) with the word $q_{\overline{a^e}}ap_{\underline{a}}q'_{\overline{a^e}}$ or $q_{\overline{a^e}}p_{\underline{a}}aq'_{\overline{a^e}}$, respectively, as appropriate. Note that in any case, $p_{\underline{a}} \ll_f v_{a}$. Further, in view of Remark 5.1.26(ii), $bp_{\underline{b}} \ll_f p_{\underline{a}} \ll_f v_{a}$.

Suppose $a \notin V_{r_\tau}$. Then $a \notin \{r_{\tau-1}, r_{\tau+1}\}$; otherwise, a would belong to either $U(r_\tau)$ or $L(r_\tau)$. Also, note that $a \notin \{r_1, r_m\}$. Further, observe that $a \in L^e(r_s)$ or $a \in U^e(r_s)$, where $s \notin \{1, \tau, m\}$ and r_s appears before a in the queue Q. We now consider the following cases.

- Case 1: $a \in L^e(r_s)$. Then, we have the portion of the root-path $b \to a \to a' \to \cdots \to r_s$ in $\tilde{G}$. Since r_s appears before a in Q, r_s is visited first, then a', a are visited in the algorithm. Hence, as per line 18 of Algorithm 5, in the i^{th} iteration, the word v_{a} is obtained from the word $v_{a'}$ by replacing the factor $p_{\underline{a}}a$ or $ap_{\underline{a}}$ of $v_{a'}$ with the word $q_{\overline{a^e}}p_{\underline{a}}aq'_{\overline{a^e}}$ or $q_{\overline{a^e}}ap_{\underline{a}}q'_{\overline{a^e}}$, respectively, as appropriate. Further, in view of Remark 5.2.3(i), we have $p_{\underline{b}}b \ll_f p_{\underline{a}} \ll_f v_{a}$.

- Case 2: $a \in U^e(r_s)$. Then, we have the portion of the root-path $r_s \to \cdots \to a' \to a \leftarrow b$ in $\tilde{G}$. Thus, as per line 11 of Algorithm 5, in the i^{th} iteration, the word v_{a} is obtained from the word $v_{a'}$ by replacing the factor $p_{\overline{a}}a$ of $v_{a'}$ with the word $q_{\underline{a^e}}p_{\overline{a}}aq'_{\underline{a^e}}$. Since $\{b\} \cup L(b) \subseteq L^e(a)$, by Remark 5.2.2, $p_{\underline{b}}b \ll_f q_{\underline{a^e}} \ll_f v_{a}$.

$\square$

If $b = r_{\tau+1}$, we show that either $p_{\overline{r_{\tau+1}}}r_{\tau+1}$ or $r_{\tau+1}p_{\underline{r_{\tau+1}}}$ is a factor of v_{r_1}.

Proposition 5.2.10. *If $\overrightarrow{r_\tau r_{\tau+1}} \in \tilde{O}$, then $p_{\overline{r_{\tau+1}}}r_{\tau+1}$ is a factor of v_{r_1}. Otherwise, $r_{\tau+1}p_{\underline{r_{\tau+1}}}$ is a factor of v_{r_1}.*

Proof. Suppose $\overrightarrow{r_\tau r_{\tau+1}} \in \tilde{O}$. Recall that $v_{r_\tau} = p_{\overline{r_\tau}}r_\tau p_{\underline{r_\tau}}$. If $\overrightarrow{r_\tau r_{\tau-1}} \in \tilde{O}$, then both $r_{\tau-1}, r_{\tau+1} \in U(r_\tau)$. Thus, in view of Remark 5.1.23(i), $p_{\overline{r_{\tau-1}}}r_{\tau-1}$ and $p_{\overline{r_{\tau+1}}}r_{\tau+1}$ are factors of $p_{\overline{r_\tau}}$. Since $U(r_{\tau-1}) \cap U(r_{\tau+1}) = \varnothing$, no element of $U(r_{\tau-1})$ appears in $p_{\overline{r_{\tau+1}}}$ and no element of $U(r_{\tau+1})$ appears in $p_{\overline{r_{\tau-1}}}$. Hence, if $r_{\tau-1}$ appears before

$r_{\tau+1}$ in $p_{\overline{r_\tau}}$, then $v_{r_\tau} = u_1 p_{\overline{r_{\tau-1}}} r_{\tau-1} u_2 p_{\overline{r_{\tau+1}}} r_{\tau+1} u_3 r_\tau p_{\underline{r_\tau}}$; and if $r_{\tau+1} r_{\tau-1} \ll p_{\overline{r_\tau}}$, then $v_{r_\tau} = u_1 p_{\overline{r_{\tau+1}}} r_{\tau+1} u_2 p_{\overline{r_{\tau-1}}} r_{\tau-1} u_3 r_\tau p_{\underline{r_\tau}}$. In any case, $p_{\overline{r_{\tau+1}}} r_{\tau+1} \ll_f v_{r_\tau}$.

Recall that the algorithm obtains v_{r_1} by iteratively updating v_{r_i} to $v_{r_{i-1}}$ (for $\tau \geq i \geq 2$) through concatenation of two permutations on the either side of the factor $p_{\overline{r_{i-1}}} r_{i-1}$ or $r_{i-1} p_{\underline{r_{i-1}}}$ or $p_{\underline{r_{i-1}}} r_{i-1}$. Observe that the factor that is updated in any step is either $p_{\overline{r_{\tau-1}}} r_{\tau-1}$ or it appears in $p_{\overline{r_{\tau-1}}}$ or it appears in the permutations that are included in one of the previous iterations. Thus, $p_{\overline{r_{\tau+1}}} r_{\tau+1}$ remains a factor of v_{r_τ} in any iteration so that we have $p_{\overline{r_{\tau+1}}} r_{\tau+1} \ll_f v_{r_1}$.

If $\overrightarrow{r_{\tau-1} r_\tau} \in \tilde{O}$, then $r_{\tau-1} \in L(r_\tau)$ and $r_{\tau+1} \in U(r_\tau)$. Thus, by Remark 5.1.23, $v_{r_\tau} = x_1 p_{\overline{r_{\tau+1}}} r_{\tau+1} x_2 r_\tau x_3 r_{\tau-1} p_{\overline{r_{\tau-1}}} x_4$, for some words x_i, so that $p_{\overline{r_{\tau+1}}} r_{\tau+1}$ is a factor of v_{r_τ}. Hence, as in the previous case, we have $p_{\overline{r_{\tau+1}}} r_{\tau+1} \ll_f v_{r_1}$.

Along the similar lines, one can prove that $r_{\tau+1} p_{\underline{r_{\tau+1}}}$ is a factor of v_{r_1} if $\overrightarrow{r_{\tau+1} r_\tau} \in \tilde{O}$. □

We now prove that w_{r_m} represents the graph G using the following lemmas.

Lemma 5.2.11. *For $i \geq 0$, if $\overrightarrow{ab} \in \tilde{O}$, let V_a and $V_b = V_a \cup L^e(b)$ be the subsets of V obtained at the end of i^{th} and $(i+1)^{th}$ iterations of Algorithm 5, respectively. For $a_1 \in L^e(b)$ and $b_1 \in V_a$, a_1 and b_1 are adjacent in $G[V_b]$ if and only if either $b_1 = b$ or $b_1 \in U(b)$.*

Proof. Note that for $i = 0$, $a = r_\tau$ and $V_a = V_{r_\tau}$. Further, note that $\{b\} \cup U(b) \subseteq V_a$. If $b_1 = b$ or $b_1 \in U(b)$, then clearly b_1 is adjacent to a_1. For the converse, suppose a_1 and b_1 are adjacent in $G[V_b]$. Since $a_1 \in L^e(b)$, we have $\overrightarrow{a_1 b} \in O$. Thus, if $b_1 = b$, we are done. Suppose $b_1 \neq b$. Note that $\overrightarrow{a_1 b_1} \in O$; otherwise, $b_1 \in L(a_1) \subseteq L^e(b)$, a contradiction that $b_1 \in V_a$. Thus, there is a directed path from a_1 to b_1 in $\tilde{G}$. Also, since $\overrightarrow{a_1 b} \in O$, there is a directed path from a_1 to b in $\tilde{G}$. Moreover, observe that the directed path from a_1 to b_1 is via b in $\tilde{G}$. Hence, $\overrightarrow{b b_1} \in O$ so that $b_1 \in U(b)$. Note that the above argument also works if $b = r_{\tau+1}$, $a = r_\tau, v_a = v_{r_1}$ and $V_a = V_{r_1}$. □

Lemma 5.2.12. *For $i \geq 0$, if $\overrightarrow{ba} \in \tilde{O}$, suppose V_a and $V_b = V_a \cup U^e(b)$ are the subsets of V obtained at the end of i^{th} and $(i+1)^{th}$ iterations of Algorithm 5, respectively. For $a_1 \in U^e(b)$ and $b_1 \in V_a$, a_1 and b_1 are adjacent in $G[V_b]$ if and only if either $b_1 = b$ or $b_1 \in L(b)$.*

Proof. Note that for $i = 0$, $a = r_\tau$ and $V_a = V_{r_\tau}$. Further, note that $\{b\} \cup L(b) \subseteq V_a$. If $b_1 = b$ or $b_1 \in L(b)$, then clearly b_1 is adjacent to a_1. For the converse, suppose a_1 and b_1 are adjacent in $G[V_b]$. Since $a_1 \in U^e(b)$, we have $\overrightarrow{ba_1} \in O$. Thus, if $b_1 = b$, we are done. Suppose $b_1 \neq b$. Note that $\overrightarrow{b_1a_1} \in O$; otherwise, $b_1 \in U(a_1) \subseteq U^e(b)$, a contradiction that $b_1 \in V_a$. Thus, there is a directed path from b_1 to a_1 in $\tilde{G}$. Also, since $\overrightarrow{ba_1} \in O$, there is a directed path from b to a_1 in $\tilde{G}$. Moreover, observe that the directed path from b_1 to a_1 is via b in $\tilde{G}$. Hence, $\overrightarrow{b_1b} \in O$ so that $b_1 \in L(b)$. Note that the above argument also works if $b = r_{\tau+1}$, $a = r_\tau, v_a = v_{r_1}$ and $V_a = V_{r_1}$. $\square$

Lemma 5.2.13. *Let A and B be two disjoint subsets of a set X, p_A be a permutation on A and q_B, q'_B be permutations on B. For some words x_1 and x_2, let $w = x_1 q_B p_A q'_B x_2$ be a word over X such that each letter of X appears at most twice in w. Then, $b_1 \in B$ alternates with $b_2 \in X \setminus B$ in w if and only if $b_2 \in A$.*

Proof. Suppose an element $b_1 \in B$ alternates with an element $b_2 \in X \setminus B$ in w. Note that if $b_2 \notin A$, b_2 cannot appear in p_A. Further, since $b_2 \notin B$ and b_1 appears in q_B as well as in q'_B, $w|_{\{b_1,b_2\}}$ has a factor b_1b_1, a contradiction.

Conversely, let $b_2 \in A$. Then, by the construction of w if b_2 occurs exactly once in w, we have $w|_{\{b_1,b_2\}} = b_1b_2b_1$, and if b_2 occurs exactly twice in w, we have $w|_{\{b_1,b_2\}} = b_1b_2b_1b_2$ or $b_2b_1b_2b_1$. Thus, b_1 and b_2 alternate in w. $\square$

Theorem 5.2.14. *For $i \geq 0$, let w_a and w_b be the words obtained at the end of i^{th} and $(i+1)^{th}$ iterations of Algorithm 5, respectively. If w_a represents $G[V_a]$, then w_b represents $G[V_b]$. Hence, the output w_{r_m} represents G.*

Proof. Note that, for $i = 0$, $w_a = w_{r_\tau}$ represents G_{r_τ} by Theorem 5.1.22. Further, observe that $w_a = uv_au'$. As per line 6 of Algorithm 5, if $b = r_{\tau+1}$, we consider

$a = r_\tau$, $w_a = w_{r_1}$, $v_a = v_{r_1}$ and $V_a = V_{r_1}$ in the following argument. We prove the result in the following cases.

- Case 1: $\overrightarrow{ab} \in \tilde{O}$. Then, as per line 12 of Algorithm 5, we have $V_b = V_a \cup L^e(b)$ and $w_b = uv_bu'$, where the word v_b is obtained from the word v_a by replacing the factor $p_{\bar{b}}b$ of v_a with the word $q_{\underline{b}^e}p_{\bar{b}}bq'_{\underline{b}^e}$. Observe that $w_b = xq_{\underline{b}^e}p_{\bar{b}}bq'_{\underline{b}^e}x'$, for some words x, x' over V_b, in which $w_a = xp_{\bar{b}}bx'$. Thus, $w_b|_{V_a} = w_a$ and $w_b|_{L^e(b)} = q_{\underline{b}^e}q'_{\underline{b}^e}$. Since the subwords w_a and $q_{\underline{b}^e}q'_{\underline{b}^e}$ of w_b represent $G[V_a]$ and $G[L^e(b)]$, respectively, any two vertices of V_a (or any two vertices of $L^e(b)$) are adjacent in $G[V_b]$ if and only if they alternate in the word w_b. Suppose $a_1 \in L^e(b)$ and $b_1 \in V_a$. Then, by Lemma 5.2.11, a_1 and b_1 are adjacent in $G[V_b]$ if and only if either $b_1 = b$ or $b_1 \in U(b)$. Further, in view of Lemma 5.2.13, a_1 alternates with b_1 in w_b if and only if either $b_1 = b$ or $b_1 \in U(b)$. Thus, w_b represents $G[V_b]$.

- Case 2: $\overrightarrow{ba} \in \tilde{O}$. Then, as per line 19 of Algorithm 5, we have $V_b = V_a \cup U^e(b)$ and $w_b = uv_bu'$, where v_b is obtained from v_a by replacing the factor $p_{\underline{b}}b$ or $bp_{\underline{b}}$ of v_a with the word $q_{\overline{b}^e}p_{\underline{b}}bq'_{\overline{b}^e}$ or $q_{\overline{b}^e}bp_{\underline{b}}q'_{\overline{b}^e}$, respectively. Observe that $w_b = yq_{\overline{b}^e}p_{\underline{b}}bq'_{\overline{b}^e}y'$ or $yq_{\overline{b}^e}bp_{\underline{b}}q'_{\overline{b}^e}y'$, for some words y, y' over V_b. Also, $w_b|_{V_a} = w_a$ and $w_b|_{U^e(b)} = q_{\overline{b}^e}q'_{\overline{b}^e}$. Since the subwords w_a and $q_{\overline{b}^e}q'_{\overline{b}^e}$ of w_b represent $G[V_a]$ and $G[U^e(b)]$, respectively, any two vertices of V_a (or any two vertices of $U^e(b)$) are adjacent in $G[V_b]$ if and only if they alternate in w_b. Suppose $a_1 \in U^e(b)$ and $b_1 \in V_a$. Then, by Lemma 5.2.12, a_1 and b_1 are adjacent in $G[V_b]$ if and only if either $b_1 = b$ or $b_1 \in L(b)$. Further, by Lemma 5.2.13, a_1 alternates with b_1 in w_b if and only if either $b_1 = b$ or $b_1 \in L(b)$. Thus, w_b represents $G[V_b]$.

Thus, by induction, we have w_{r_m} represents G. $\square$

We now conclude that the output w_{r_m} of Algorithm 5 is a minimum-word-representant of G.

Theorem 5.2.15. *If G is a treelike permutation graph on n vertices with clique number κ, then $\ell(G) = 2n - \kappa$.*

Proof. Note that the number of vertices in the longest directed path C is κ, as κ is the clique number of G. Since w_{r_τ} is a minimum-word-representant of the double-arborescence G_{r_τ}, by Theorem 5.1.22, the length of w_{r_τ} is $2|V_{r_\tau}| - \kappa$. In each iteration of Algorithm 5, exactly two vertices of $L^e(b)$ or of $U^e(b)$ are inserted in the word w_a. Thus, the length of the output w_{r_m} (a word-representant of G, by Theorem 5.2.14) is $2|V| - \kappa$ so that $\ell(G) \leq 2n - \kappa$. Further, in view of Theorem 1.3.32, we have $2n - \kappa \leq \ell(G)$. Hence, w_{r_m} is a minimum-word-representant of G and $\ell(G) = 2n - \kappa$. □

Remark 5.2.16. The time complexity of Algorithm 5 is a polynomial in the size of the graph. For instance, note that the root-path in step 1 can be found in linear time (by Theorem 1.6.28). Further, step 3 can be done in $O(n^2)$ time, where n is the number of vertices of G. Also, construction of permutations in steps 9 and 16 representing an arborescence can be performed in $O(n^2)$ time (by Remark 5.2.1). It can be observed that each step inside the while loop can be executed in $O(n^2)$ time. Hence, Algorithm 5 runs in $O(n^3)$ time, as the number of iterations is at most n.

5.3 Conclusion and Future Work

The characterizations of (double-)arborescences given in this work may be useful in addressing various enumeration related problems of double-arborescences. In [Castelo and Wormald, 2003], the arborescences were enumerated both exactly and asymptotically. However, no such results are available for double-arborescences. Considering the characterizations based on split decomposition for certain subclasses of distance-hereditary graphs, some grammars enumerating the subclasses were produced in [Bahrani and Lumbroso, 2018]. As our characterizations given in Corollaries 5.1.6 and 5.1.11 for double-arborescences and arborescences are using the split-decomposition trees, taking a clue from the work in [Bahrani and Lumbroso, 2018], one may produce grammars to enumerate the (double-)arborescences. Moreover, the recognization algorithm for treelike comparability graphs given in [Cornelsen and Di Stefano, 2009, Theorem 5] may be customized to both double-arborescences and arborescences, using the characterizations given in this work.

Considering the open problem given in [Srinivasan and Hariharasubramanian, 2024], we established that the length of a minimum-word-representant of a treelike permutation graph on n vertices with clique number κ is $2n - \kappa$. In this context, it is natural to ask the following question. Can this work be extended to a more general class of treelike comparability graphs? Note that a minimum-word-representant of a treelike permutation graph need not be unique. In fact, the minimum-word-representant produced by Algorithm 5 on a treelike permutation graph differs depending on the maximum size clique chosen. It is also interesting to enumerate all minimum-word-representants of treelike permutation graphs, in line with the work in [Gaetz and Ji, 2020].

Bibliography

Ahn, J., Jaffke, L., Kwon, O.-j. and Lima, P. T.: 2022, Well-partitioned chordal graphs, *Discrete Math.* **345**(10), Paper No. 112985, 23.

Bahrani, M. and Lumbroso, J.: 2018, Enumerations, forbidden subgraph characterizations, and the split-decomposition, *Electron. J. Combin.* **25**(4), Paper No. 4.47, 45pp.

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

Bandelt, H.-J. and Mulder, H. M.: 1986, Distance-hereditary graphs, *J. Combin. Theory Ser. B* **41**(2), 182–208.

Bixby, R. E.: 1984, A composition for perfect graphs, *Topics on perfect graphs*, Vol. 88 of *North-Holland Math. Stud.*, North-Holland, Amsterdam, pp. 221–224.

Bogart, K. P.: 1973, Maximal dimensional partially ordered sets. I. Hiraguchi's theorem, *Discrete Math.* **5**, 21–31.

Bonomo-Braberman, F., Durán, G., Pardal, N. and Safe, M. D.: 2022, Forbidden induced subgraph characterization of circle graphs within split graphs, *Discrete Appl. Math.* **323**, 43–75.

Bose, P., Buss, J. F. and Lubiw, A.: 1998, Pattern matching for permutations, *Inform. Process. Lett.* **65**(5), 277–283.

Bouchet, A.: 1987, Reducing prime graphs and recognizing circle graphs, *Combinatorica* **7**(3), 243–254.

Bouchet, A.: 1988, Transforming trees by successive local complementations, *J. Graph Theory* **12**(2), 195–207.

Brandstädt, A. and Le, V. B.: 2006, Structure and linear time recognition of 3-leaf powers, *Inform. Process. Lett.* **98**(4), 133–138.

Brightwell, G.: 1993, On the complexity of diagram testing, *Order* **10**.

Broere, B. and Zantema, H.: 2019, The k-dimensional cube is k-representable, *J. Autom. Lang. Comb.* **24**(1), 3–12.

Castelo, R. and Wormald, N.: 2003, Enumeration of P_4-free chordal graphs, *Graphs Combin.* **19**(4), 467–474.

Chakraborty, D., Chandran, L. S., Padinhatteeri, S. and Pillai, R. R.: 2024, s-club cluster vertex deletion on interval and well-partitioned chordal graphs, *Discrete Appl. Math.* **345**, 170–189.

Charbit, P., de Montgolfier, F. and Raffinot, M.: 2012, Linear time split decomposition revisited, *SIAM J. Discrete Math.* **26**(2), 499–514.

Chen, H. Z. Q., Kitaev, S. and Saito, A.: 2022, Representing split graphs by words, *Discuss. Math. Graph Theory* **42**(4), 1263–1280.

Choi, I., Kim, J. and Kim, M.: 2019, On operations preserving semi-transitive orientability of graphs, *J. Comb. Optim.* **37**(4), 1351–1366.

Chu, F. P. M.: 2008, A simple linear time certifying LBFS-based algorithm for recognizing trivially perfect graphs and their complements, *Inform. Process. Lett.* **107**(1), 7–12.

Cicerone, S. and Di Stefano, G.: 1999, On the extension of bipartite to parity graphs, *Discrete Appl. Math.* **95**(1-3), 181–195.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C.: 2009, *Introduction to algorithms*, third edn, MIT Press, Cambridge, MA.

Corneil, D. G., Lerchs, H. and Burlingham, L. S.: 1981, Complement reducible graphs, *Discrete Appl. Math.* **3**(3), 163–174.

Cornelsen, S. and Di Stefano, G.: 2009, Treelike comparability graphs, *Discrete Appl. Math.* **157**(8), 1711–1722.

Cunningham, W. H.: 1982, Decomposition of directed graphs, *SIAM J. Algebraic Discrete Methods* **3**(2), 214–228.

Cunningham, W. H. and Edmonds, J.: 1980, A combinatorial decomposition theory, *Canadian J. Math.* **32**(3), 734–765.

Dahlhaus, E.: 1994, Efficient parallel and linear time sequential split decomposition, *Foundation of Software Technology and Theoretical Computer Science: 14th Conference Madras, India, December 15–17, 1994 Proceedings 14*, Springer, pp. 171–180.

Dahlhaus, E.: 2000, Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition, *J. Algorithms* **36**(2), 205–240.

de Figueiredo, C. M. H., Klein, S. and Vušković, K.: 2002, The graph sandwich problem for 1-join composition is NP-complete, *Discrete Appl. Math.* **121**(1-3), 73–82.

Di Stefano, G.: 2012, Distance-hereditary comparability graphs, *Discrete Appl. Math.* **160**(18), 2669–2680.

Dom, M., Guo, J., Hüffner, F. and Niedermeier, R.: 2004, Error compensation in leaf root problems, *Algorithms and computation*, Vol. 3341 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, pp. 389–401.

Dwary, T. and Krishna, K. V.: 2024, Word-representability of graphs with respect to split recomposition, *Discrete Appl. Math.* **357**, 310–321.

Dwary, T. and Krishna, K. V.: 2025, Characterization of word-representable graphs using modular decomposition, *Communications in Combinatorics and Optimization, To appear* .

Dwary, T. and Krishna, K. V.: 2026a, Characterization of double-arborescences and their minimum length word-representants, *Discrete Appl. Math.* **380**, 599–611.

Dwary, T. and Krishna, K. V.: 2026b, Minimum length word-representants of tree-like permutation graphs, *SOFSEM 2026: Theory and Practice of Computer Science*, Vol. 16448 of *Lecture Notes in Computer Science, Springer*, pp. 90–103.

Dwary, T. and Krishna, K. V.: 2026c, Word-representability of well-partitioned chordal graphs. Communicated (arXiv:2504.04256).

Dwary, T., Mozhui, K. and Krishna, K. V.: 2025, Characterization of split comparability graphs. (arXiv:2504.19167).

Dwary, T., Mozhui, K. and Krishna, K. V.: 2026, Representation number of word-representable split graphs, *Siberian Advances in Mathematics* . To appear (arXiv:2502.00872).

Even, S., Pnueli, A. and Lempel, A.: 1972, Permutation graphs and transitive graphs, *J. Assoc. Comput. Mach.* **19**, 400–410.

Fleischmann, P., Haschke, L., Löck, T. and Nowotka, D.: 2024, Word-representable graphs from a word's perspective, *Acta Inform.* **61**(4), 383–400.

Gabor, C. P., Supowit, K. J. and Hsu, W. L.: 1989, Recognizing circle graphs in polynomial time, *J. Assoc. Comput. Mach.* **36**(3), 435–473.

Gaetz, M. and Ji, C.: 2020, Enumeration and extensions of word-representants, *Discrete Appl. Math.* **284**, 423–433.

Gallai, T.: 1967, Transitiv orientierbare Graphen, *Acta Math. Acad. Sci. Hungar.* **18**, 25–66.

Gavoille, C. and Paul, C.: 2003, Distance labeling scheme and split decomposition, *Discrete Math.* **273**(1-3), 115–130.

Gimbel, J.: 1992, Sources in posets and comparability graphs, *Order* **9**(4), 361–365.

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

Glen, M., Kitaev, S. and Pyatkin, A.: 2018, On the representation number of a crown graph, *Discrete Appl. Math.* **244**, 89–93.

Golumbic, M. C.: 1978, Trivially perfect graphs, *Discrete Math.* **24**(1), 105–107.

Golumbic, M. C.: 2004, *Algorithmic graph theory and perfect graphs*, Vol. 57 of *Annals of Discrete Mathematics*, second edn, Elsevier Science B.V., Amsterdam.

Guenver, G.-B. and Rampon, J.-X.: 2004, Split orders, *Discrete Math.* **276**(1-3), 249–267.

Habib, M. and Paul, C.: 2010, A survey of the algorithmic aspects of modular decomposition, *Computer Science Review* **4**(1), 41–59.

Halldórsson, M. M., Kitaev, S. and Pyatkin, A.: 2011, Alternation graphs, *Graph-theoretic concepts in computer science*, Vol. 6986 of *Lecture Notes in Comput. Sci.*, Springer, Heidelberg, pp. 191–202.

Halldórsson, M. M., Kitaev, S. and Pyatkin, A.: 2016, Semi-transitive orientations and word-representable graphs, *Discrete Appl. Math.* **201**, 164–171.

Hameed, H.: 2024, On semi-transitivity of (extended) Mycielski graphs, *Discrete Appl. Math.* **359**, 83–88.

Hefty, Z., Horn, P., Muir, C. and Owens, A.: 2024, Word-representable graphs: orientations, posets, and bounds, *Electron. J. Combin.* **31**(4), Paper No. 4.2, 26.

Hiraguti, T.: 1955, On the dimension of orders, *Sci. Rep. Kanazawa Univ.* **4**(1), 1–20.

Kelly, D.: 1977, The 3-irreducible partially ordered sets, *Canadian J. Math.* **29**(2), 367–383.

Kenkireth, B. G., Sajith, G. and Sasidharan, S.: 2025, Multi-word-representability of graphs, *Discrete Appl. Math.* **374**, 61–74.

Kitaev, S.: 2013, On graphs with representation number 3, *J. Autom. Lang. Comb.* **18**(2), 97–112.

Kitaev, S., Long, Y., Ma, J. and Wu, H.: 2021, Word-representability of split graphs, *J. Comb.* **12**(4), 725–746.

Kitaev, S. and Lozin, V.: 2015, *Words and graphs*, Monographs in Theoretical Computer Science. An EATCS Series, Springer, Cham.

Kitaev, S. and Pyatkin, A.: 2008, On representable graphs, *J. Autom. Lang. Comb.* **13**(1), 45–54.

Kitaev, S. and Pyatkin, A.: 2024, On semi-transitive orientability of split graphs, *Inform. Process. Lett.* **184**, Paper No. 106435, 4.

Kitaev, S. and Pyatkin, A. V.: 2023, On semi-transitive orientability of triangle-free graphs, *Discuss. Math. Graph Theory* **43**(2), 533–547.

Kitaev, S. and Pyatkin, A. V.: 2025, A note on Hameed's conjecture on the semi-transitivity of Mycielski graphs, *Discuss. Math. Graph Theory* **45**(4), 1157–1162.

Kitaev, S. and Saito, A.: 2020, On semi-transitive orientability of Kneser graphs and their complements, *Discrete Math.* **343**(8), 111909, 7.

Kitaev, S. and Seif, S.: 2008, Word problem of the Perkins semigroup via directed acyclic graphs, *Order* **25**(3), 177–194.

Kitaev, S. and Sun, H.: 2024, Human-verifiable proofs in the theory of word-representable graphs, *RAIRO-Theor. Inf. Appl.* **58**, 9.

Ma, T. H. and Spinrad, J. P.: 1991, Cycle-free partial orders and chordal comparability graphs, *Order* **8**(1), 49–61.

McConnell, R. M. and Spinrad, J. P.: 1999, Modular decomposition and transitive orientation, *Discrete Math.* **201**(1-3), 189–241.

Möhring, R. H.: 1985, Algorithmic aspects of comparability graphs and interval graphs, *Graphs and order (Banff, Alta., 1984)*, Vol. 147 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, Reidel, Dordrecht, pp. 41–101.

Möhring, R. H. and Radermacher, F. J.: 1984, Substitution decomposition for discrete structures and connections with combinatorial optimization, *Algebraic and combinatorial methods in operations research*, Vol. 95 of *North-Holland Math. Stud.*, North-Holland, Amsterdam, pp. 257–355.

Mozhui, K.: 2025, *Representation Number of Word-Representable Graphs*, PhD thesis, IIT Guwahati, India.

Mozhui, K. and Dwary, T.: 2026, Boxicity of a class of split graphs using words, *CALDAM 2026: Algorithms and Discrete Applied Mathematics*, Vol. 16445 of *Lecture Notes in Computer Science, Springer*, pp. 321–335.

Mozhui, K. and Krishna, K. V.: 2023, Words for the graphs with permutation-representation number at most three. (arXiv:2307.00301).

Mozhui, K. and Krishna, K. V.: 2025a, The representation number and *prn* of stacked book graphs, *Current Progress in Interdisciplinary Research. Select Papers of RIC 2024*, Vol. 3, Springer, pp. 337–346.

Mozhui, K. and Krishna, K. V.: 2025b, An upper bound for the permutation-representation number of bipartite graphs, *Journal of Information Processing* **33**, 1033–1041.

Mozhui, K. and Krishna, K. V.: 2026, Word-representation of melon graphs, *Graphs Combin.* **42**(2), Paper No. 36, 23.

Nishimura, N., Ragde, P. and Thilikos, D. M.: 2002, On graph powers for leaf-labeled trees, *J. Algorithms* **42**(1), 69–108.

Olariu, S.: 1992, On sources in comparability graphs, with applications, *Discrete Math.* **110**(1-3), 289–292.

Ortiz, C. and Villanueva, M.: 1996, On split-comparability graphs, *Proc. II ALIO-EURO Workshop on Practical Combinatorial Optimization, Chile*, pp. 91–105.

Pretzel, O.: 1985, On graphs that can be oriented as diagrams of ordered sets, *Order* **2**(1), 25–40.

Rao, M.: 2004, Coloring a graph using split decomposition, *Graph-theoretic concepts in computer science*, Vol. 3353 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, pp. 129–141.

Rao, M.: 2008, Solving some NP-complete problems using split decomposition, *Discrete Appl. Math.* **156**(14), 2768–2780.

Sachs, H.: 1970, On the Berge conjecture concerning perfect graphs, *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, Gordon and Breach, New York-London-Paris, pp. 377–384.

Spinrad, J.: 1994, Recognition of circle graphs, *J. Algorithms* **16**(2), 264–282.

Srinivasan, E. and Hariharasubramanian, R.: 2024, Minimum length word-representants of word-representable graphs, *Discrete Appl. Math.* **343**, 149–158.

Srinivasan, E. and Hariharasubramanian, R.: 2025, Forbidden induced subgraph characterization of word-representable split graphs, *arXiv:2512.12259* .

Trotter, Jr., W. T., Moore, Jr., J. I. and Sumner, D. P.: 1976, The dimension of a comparability graph, *Proc. Amer. Math. Soc.* **60**, 35–38 (1977).

Trotter, W. T.: 1992, *Combinatorics and partially ordered sets: Dimension theory*, Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD.

Trotter, W. T. and Moore, J. I.: 1976, Characterization problems for graphs, partially ordered sets, lattices, and families of sets, *Discrete Math.* **16**(4), 361–381.

West, D. B.: 1996, *Introduction to graph theory*, Prentice Hall, Inc., Upper Saddle River, NJ.

Wolk, E. S.: 1962, The comparability graph of a tree, *Proc. Amer. Math. Soc.* **13**, 789–795.

Wolk, E. S.: 1965, A note on "The comparability graph of a tree", *Proc. Amer. Math. Soc.* **16**, 17–20.

Yan, J.-H., Chen, J.-J. and Chang, G. J.: 1996, Quasi-threshold graphs, *Discrete Appl. Math.* **69**(3), 247–255.

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

Index

Bio-Data

Full Name	:	Tithi Dwary
<u>Education</u>		
Present	:	Research Scholar, Department of Mathematics Indian Institute of Technology Guwahati, Guwahati.
2020	:	M. Sc. in Pure Mathematics Calcutta University, Ballygunge, Kolkata.
2018	:	B. Sc. in Mathematics (Hons.) Bethune College, Kolkata.
<u>Teaching Experience</u>		
October, 2022 – Till Date	:	Teaching Assistant, Department of Mathematics, IIT Guwahati, Guwahati.

<u>Publications/Manuscripts</u>

1. Tithi Dwary and K. V. Krishna. Word-Representability of Graphs with respect to Split Recomposition. *Discrete Applied Mathematics*, 357, 310-321, 2024.

2. Tithi Dwary and K. V. Krishna. Characterization of Double-Arborescences and their Minimum Length Word-Representants. *Discrete Applied Mathematics*, 380, 599-611, 2026.

3. Tithi Dwary and K. V. Krishna. Characterization of Word-Representable Graphs using Modular Decomposition. *Communications in Combinatorics and Optimization*, To appear, 2026.

4. Tithi Dwary, Khyodeno Mozhui and K. V. Krishna. Representation number of word-representable split graphs. *Siberian Advances in Mathematics*, To appear, 2026.

5. Tithi Dwary and K. V. Krishna. Minimum length word-representants of tree-like permutation graphs. In: *SOFSEM 2026: Theory and Practice of Computer Science*, Lecture Notes in Computer Science, Springer Nature Switzerland, 16448, 90-103, 2026.

6. Khyodeno Mozhui and Tithi Dwary. Boxicity of a Class of Split Graphs Using Words. In: *CALDAM 2026: Algorithms and Discrete Applied Mathematics*, Lecture Notes in Computer Science, Springer, 16445, 321-335, 2026.

7. Khyodeno Mozhui, Tithi Dwary and K. V. Krishna. Line graphs of non-word-representable graphs are not always non-word-representable. arXiv:2509.03339, 2025. (*Communicated*)

8. Tithi Dwary and K. V. Krishna. Word-representability of well-partitioned chordal graphs. arXiv:2504.04256, 2026. (*Communicated*)

Conference Talks:

1. On the Split Decomposition and Word-Representable Graphs at *International Seminar on Topology, Algebra and Applications (ISTAA)*, Department of Mathematics, North Bengal University, March 12-14, 2024.

2. On Operations Preserving Word-Representability of Graphs at *the 9th Gdańsk Conference on Graph Theory – GCGT 2024* (presented online), Gdańsk, Poland, June 16-21, 2024.

3. Characterization of Word-Representable Graphs using Modular Decomposition at *the International Conference on Graph Theory and its Applications (ICGTA) 2024*, Presidency University, Bengaluru, June 20-22, 2024.

4. Minimum length word-representants of treelike permutation graphs at *the 51st International Conference on Current trends in Theory and Practice of Computer Science – SOFSEM 2026*, Kraków, Poland, February 09-13, 2026.

Workshops:

1. *Annual Foundation School-I* organized by National Centre for Mathematics, December 2021, *Topics:* Linear Algebra and Group Theory, Complex Analysis, Point set Topology.

2. *Advance Instructional School* organized by National Centre for Mathematics, May 2022, *Topic:* Geometric Group Theory.

3. *Annual Foundation School-III* organized by National Centre for Mathematics, June 2022, *Topics:* Algebraic Topology, Galois Theory, Functional Analysis.

4. *Advanced Instructional School* organized by National Centre for Mathematics, December 2025, *Topic:* Advanced Combinatorics.