Combining Vertex Orderings

Master's Thesis of

Moritz Bär

At the Department of Informatics
Institute of Theoretical Informatics (ITI)

Reviewer:	PD Dr. Torsten Ueckerdt
Second reviewer:	T.T.-Prof. Dr. Thomas Bläsius
Advisor:	PD Dr. Torsten Ueckerdt

30. November 2025 – 30. May 2025

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, 30. May 2025

(Moritz Bär)

Abstract

It is a central problem in graph theory to determine the values of graph parameters for a given graph or to bound them for a graph class. These parameters range from describing superficial properties, such as clique- or independence number, over applied structure like chromatic number or -index to very abstract structural properties like stack number or bandwidth. For many of these parameters, vertex orderings that avoid certain patterns can be witnesses for the parameters' values. By witness we mean any piece of information by which a parameter value can be certified in polynomial time. Pattern-free vertex orderings satisfy this time-constraint, as the trivial algorithm testing a given vertex ordering on whether it avoids a pattern has polynomial complexity. In some cases, prominently with perfect elimination schemes for chordal graphs, there also exist polynomial or even linear algorithms for computing such witness orderings. We introduce the concept of "combinability" of graph parameters, a measure of how accurate a single vertex ordering can be as a witness for two parameters. More formally, two parameters A and B are (f, g)-combinable for functions f, g if for every graph G which has value n_A for parameter A and value n_B for parameter B, there exists a vertex ordering that is a witness both for value $f(n_A, n_B)$ for A and for value $g(n_A, n_B)$ for B. We define this concept, put it into context regarding existing work on forbidden patterns, and take a handful of well-known graph parameters to study their combinability.

Zusammenfassung

Eine zentrale Frage in der Graphentheorie ist es, Parameterwerte für gegebene Graphen zu bestimmen oder für Graphklassen zu beschränken. Solche Parameter gehen von oberflächlichen Eigenschaften, wie Cliquen- oder Unabhängigkeitszahl, über angewandte Strukturen wie Färbungszahl oder -index bis hin zu sehr abstrakten strukturellen Eigenschaften wie Stack Number oder Bandweite. Für viele solcher Parameter können Knotenordnungen, die bestimmte Muster vermeiden, als Zeugen für Parameterwerte dienen. Als Zeuge bezeichnen wir eine Information, anhand derer ein Parameterwert in Polynomialzeit verifiziert werden kann. Knotenordnungen, die bestimmte Muster vermeiden, erfüllen diese Laufzeitbeschränkung, da der triviale Algorithmus für Tests auf das Vorhandensein eines bestimmten Musters polynomielle Komplexität hat. In einigen Fällen gibt es auch polynomielle Algorithmen zur Berechnung solcher Zeugenordnungen. Ein bekanntes Beispiel hierfür sind perfekte Eliminationsschemata für chordale Graphen, die sogar in Linearzeit berechnet werden können. Wir führen das Konzept der "Kombinierbarkeit" von Graphparametern ein, die eine Metrik für die maximale Genauigkeit einer einzelnen Knotenordnung als Zeuge für zwei Parameter darstellt. Formell sind zwei Parameter A und B (f, g)-kombinierbar für Funktionen f und g, wenn für jeden Graphen G, der den Wert n_A für den Parameter A und den Wert n_B für den Parameter B hat, eine Knotenordnung existiert, die gleichzeitig den Wert $f(n_A, n_B)$ für A und den Wert $g(n_A, n_B)$ für B bezeugt. Dieses Konzept wird von uns definiert und in den Kontext bestehender Arbeiten zu verbotenen Mustern eingeordnet. Weiterhin untersuchen wir eine Handvoll bekannter Graphparameter auf deren Kombinierbarkeit.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Related Work 2

1.3 Outline 2

2 Preliminaries **5**

2.1 Graph Theory 5

2.1.1 Vertex Orderings 6

2.2 Patterns and Combinability 7

2.3 Graph Properties 9

2.4 Well-Known Graph Parameters and Their Associated Patterns 10

3 Combinability of Graph Parameters **13**

3.1 Degeneration Number 13

3.1.1 Degeneration Number and Chordality 13

3.1.2 Degeneration Number and Chromatic Number 16

3.1.3 Degeneration Number and Queue Number 19

3.1.4 Degeneration Number and Stack Number 24

3.1.5 General Framework 24

3.2 Bandwidth Number 27

3.2.1 Sufficient Property 27

3.2.2 Bandwidth Number with Queue Number and Stack Number 27

3.3 Further Results 29

4 Combinability of Individual Patterns **33**

4.1 Single Decided Edges 35

4.2 Permutations of Stars 38

4.3 Split Graphs and Their Mirrors 42

5 Conclusion **49**

5.1 Future Work 49

Bibliography **51**

1 Introduction

Forbidden patterns in vertex orderings have been studied as a method to characterize graph classes. This study has recently lead to a question: When does forbidding two patterns at once characterize the intersection of the classes characterized by each pattern. We find a lot of examples where this is not the case and extend the question to graph parameters.

1.1 Motivation

There are a large number of parameters that can be used to describe the properties of graphs, such as the chromatic number, the independence number, the maximum degree, and many more. Determining these parameters for individual graphs or bounding them for graph classes can be interesting for structural or computational reasons, but it is often an $\mathcal{NP}$-hard problem to determine a parameter's value for a given graph. Many such parameters can be characterized by the existence of a vertex ordering avoiding certain patterns or families of patterns, so vertex orderings can be used to prove properties of a graph. A well-known example of this is the Gallai-Hasse-Roy-Vitaver-theorem, which states that a graph with a vertex ordering in which the longest path going in only one direction is of length k has chromatic number at most k [Gal68, Has65, Roy67, Vit62]. This makes vertex orderings witnesses for graph properties, and for a given vertex ordering it is possible to test for the occurrence of a specific pattern in polynomial time, trivially in $\mathcal{O}(n^k)$ for a graph G on n and pattern P on k vertices. In practice, much faster algorithms have been found for recognizing many patterns, often linear-time or almost linear-time; and for small patterns algorithms have been found that even construct vertex orderings without the pattern in (almost-)linear-time [FH21].

For these reasons, it is an interesting problem to find vertex orderings that avoid certain patterns. A fair amount of research has gone into studying which graph classes can be characterized by forbidding finite families of patterns and into developing algorithms for finding and recognizing vertex orderings that avoid these patterns. In their 2021 survey, Feuilloley and Habib introduce the "union-intersection-property" for patterns: Patterns P and Q have the union-intersection-property if any graph that permits both a P-free and a Q-free vertex ordering also permits a $\{P, Q\}$-free vertex ordering [FH21]. They suggest that a better understanding of when the union-intersection-property holds could yield some interesting structural results.

This thesis explores a generalization of the union-intersection-property that we call the "combinability of graph parameters". Intuitively, we call two graph parameters that can be characterized by forbidden patterns "combinable" if for any graph which has certain parameter values in both, there exists a vertex ordering that shows (larger) values for both parameters at the same time. We additionally require that the shown values overestimate the actual values by at most a function depending only on the actual values. We take a number of common graph parameters that have a pattern-based characterization and investigate their pairwise combinability. In doing this, we find a handful of general approaches to demonstrate or disprove combinability with some of them.

1.2 Related Work

Forbidden patterns as a way of characterizing graph classes have been studied since 1982, when Skrien noticed a common way to describe some well-known graph classes such as chordal-, comparability- and interval graphs by forbidding directed subgraphs on three vertices [Skr82]. Damaschke expanded on this result by formalizing the idea of characterizing graph classes by forbidden ordered subgraphs in orientations and observing that this characterization could lead to fast recognition algorithms [Dam90]. More recently, Feuilloley and Habib have done an exhaustive survey on the graph classes that can be characterized by forbidding sets of patterns on three vertices [FH21], and find linear-time recognition algorithms for most of them.

Even before patterns as such were studied, there had been some results that can be interpreted as characterizing graph parameters by forbidden patterns – a famous example is the identity between chromatic number and length of directed paths discovered separately by Gallai, Hasse, Roy and Vitaver [Gal68, Has65, Roy67, Vit62]. But in recent years, many more graph parameters related to linear layouts have been studied; these include bandwidth number [CCDG82, CS89], cutwidth number [HLMP11, CS89], book thickness or stack number [BK79, Str23, DW04] and queue number [Wie16, Str23, DW04]. Stack number and queue number in particular are often studied together, as they provide an interesting perspective on the relative power of the nominative data structures in the question whether one parameter bounds the other [DW04, KKPU24, Pem92, Duj+22]. To our knowledge, we are the first to investigate the combinability of graph parameters.

1.3 Outline

We begin in Chapter 2 with a brief overview of the basic definitions and notations that are commonly used in graph theory and which we adhere to in this thesis. We extend these common definitions by introducing the concept of *vertex orderings* and giving a directionality to adjacency in Section 2.1.1. With these basics set, we move on to introducing the concept of forbidden patterns as a framework to characterize graph classes in Section 2.2. We define what we mean by patterns and parametrized pattern families, define some operations on patterns and observe a few basic properties of these operations. Here we also give a formal definition for the notion of combinability of pattern families, which we extend to graph properties and graph parameters in Section 2.3. We end in Section 2.4 by giving the list of properties and parameters whose pairwise combinability we explore in this thesis.

Once all the necessary concepts have been introduced, we begin to investigate the main question of this thesis in Chapter 3. We first look at pairings with degeneration number in Section 3.1 and find most of the parameters introduced previously to not be combinable with it. From the constructions we use to show non-combinability, we derive a framework that can be used to show non-combinability of any pattern with degeneration number more easily. We demonstrate the application of this framework by giving simpler proofs for the non-combinability of degeneration number with the other parameters already discussed, as well as finding a sufficient condition for non-combinability with degeneration number. In Section 3.2 we observe a very simple criterion for a family of patterns that guarantees combinability with bandwidth number. We then investigate the tightness of the bounds given by this criterion and find a much stronger statement for the combinability of stack number and queue number with bandwidth number, but only in the restricted case of caterpillars.

Finally, we show the combinability of queue number with chromatic number, demonstrate that the same approach cannot work for stack number and chromatic number, and show that stack number and queue number are not combinable in Section 3.3.

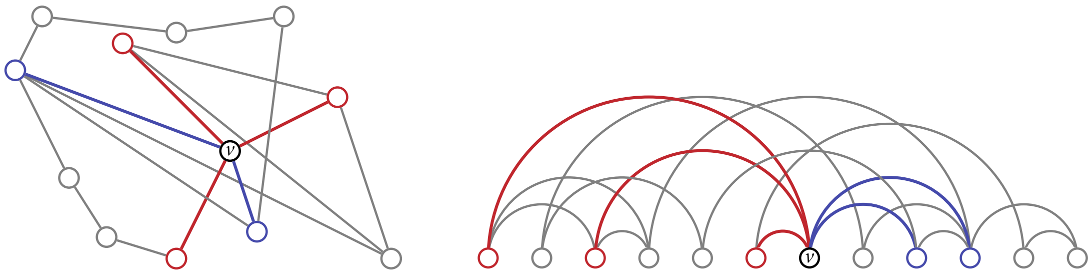
In Chapter 4, we leave behind the general question of combinability for graph parameters and turn to the narrower topic of the combinability (or union-intersection-property) for individual patterns. We begin by conjecturing that no patterns that are different permutations of each other are combinable, and show this in several restricted cases: We start in Section 4.1 by looking at patterns with a single decided edge between consecutive vertices and show that permutations of them are not combinable. However, we also find an infinite family of patterns that are non-trivially combinable here. Next, in Section 4.2 we look at patterns without non-edges whose edges induce a star on their vertex set and show that our conjecture holds here: No two permutations of the same star are combinable. We finish with "separated split graphs" in Section 4.3, the most complicated class of patterns we study. Here we show the weaker result that permutations of separated split graphs with their cliques on different sides are not combinable.

2 Preliminaries

Before we begin our investigation, we want to lay the groundwork by introducing some common concepts of graph theory in Section 2.1, with special emphasis on the vertex orderings this thesis is concerned with in Section 2.1.1. Next, in Section 2.2, we define patterns as trigraphs with a total order on their vertex set and give a set of basic operations on patterns. We then explore the concept of vertex orderings and discuss how graphs can be characterized by whether they permit a vertex ordering without certain patterns, which we refine in Section 2.2 to characterizing graph properties and graph parameters by forbidden patterns. From these concepts we derive the notion of combinability of patterns and graph parameters that will be explored throughout this thesis.

2.1 Graph Theory

An undirected graph is a pair $G = (V, E)$ with vertex set $V(G) = V$ and edge set $E(G) = E \subseteq \binom{V}{2}$. We assume graphs to be finite, simple and undirected unless stated otherwise. Edges $\{u, v\} \in E$ are abbreviated as uv. Two vertices $u, v \in V$ are called *adjacent* if $uv \in E$. The set of vertices adjacent to $v \in V$, called the *neighbourhood* of v is $N(v) := \{u \in V \mid uv \in E\}$. The *degree* of v is $\deg(v) := |N(v)|$. A vertex v with $\deg(v) = 1$ is a leaf. The complement of G is $\overline{G} := \left(V, \binom{V}{2} - E\right)$. By kG we denote the graph obtained by taking k disjoint copies of G. A *directed graph* is a graph whose edges have a unique direction, so $\overrightarrow{uv}$ goes from u to v, but not from v to u. On directed graphs, we use the notation $N^+(v)$ (respectively $N^-(v)$) for the out-neighbourhood (respectively in-neighbourhood) of v, that is the vertices to which v has outgoing (respectively incoming) edges. A graph $H = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. A subgraph is called *induced* if $E' = \{uv \in E \mid u, v \in V'\}$, denoted as $H \subseteq_{\mathrm{IND}} G$ For $V' \subseteq V$ we denote the subgraph of G induced by V' as $G[V']$. We denote by $[n] := \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ the set of natural numbers up to n. Important classes of graphs are the *complete graphs* $K_n := \left([n], \binom{[n]}{2}\right)$, the complete bipartite graphs $K_{n,m} := \left(\{a_i \mid i \in [n]\} \cup \{b_j \mid j \in [m]\}, \{a_i b_j \mid i \in [n], j \in [m]\}\right)$ (especially the stars $T_S(n) := K_{n,1}$), the empty graphs $E_n := ([n], \emptyset)$, the *paths* $P_n := \left([n], \{\{i, i+1\} \mid i \in [n-1]\}\right)$ and, for $n \geq 3$, the *cycles* $C_n := \left([n], \{\{i, i+1\} \mid i \in [n-1]\} \cup \{\{1, n\}\}\right)$. For $n \geq 2$, the unique vertex v with $\deg(v) \geq 2$ in $T_S(n)$ is called the centre of the star. We alternatively denote P_n, C_n by the n-tuple $(v_1, \ldots, v_n)$, especially in the context of finding paths or cycles as subgraphs. The edges are then between consecutive vertices and, in the case of cycles, the first and last vertex. A set $V' \subseteq V$ is *independent* if $G[V']$ is empty, and a *clique* if $G[V']$ is complete. If for every edge $uv \in E$ we have $u \in V'$ or $v \in V'$, V' is a *vertex cover* of G. A graph G is connected if for every pair of vertices $u, v \in V(G)$ there exists a *u, v-path* $(u = v_1, \ldots, v_k = v) \subseteq G$. A tree is a connected graph without a cycle as a subgraph. A caterpillar is a path (the spine) with possibly some leaves attached to each vertex (the hairs).



(a) The graph G. **(b)** The vertex ordering σ of G.

Figure 2.1: A graph G with a vertex ordering σ. $N_l(v)$ is marked in red, $N_r(v)$ in blue.

A graph $G = (V, E)$ is a split graph if V can be partitioned into a clique C and an independent set I. If every vertex in I is adjacent to every vertex in C, we call G a complete split graph. If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two graphs, the carthesian product of G_1 and G_2 is

$$G_1 \square G_2 \coloneqq \left(V_1 \times V_2, \left\{ \{(u_1, y), (v_1, y)\} \,\middle|\, u_1 v_1 \in E_1, y \in V_2 \right\} \cup \left\{ \{(x, u_2), (x, v_2)\} \,\middle|\, u_2 v_2 \in E_2, x \in V_1 \right\} \right).$$

2.1.1 Vertex Orderings

Definition 2.1: *A* vertex ordering *of a graph G is a total order $\prec$ on the vertex set $V(G)$, or equivalently a permutation σ of $V(G)$.*

Vertex orderings can also be interpreted as drawings of G where all vertices are placed on a line segment with no two vertices occupying the same position. Such drawings are often referred to as *linear layouts* (see for example [DW04]). Throughout the thesis, we will be interested in finding vertex orderings of graphs that avoid certain patterns, or in showing that no such orderings exist.

Definition 2.2: *Let G be a graph with vertex ordering σ and $v \in V(G)$. We define the left (respectively right) neighbourhood of v with respect to σ as $N_{l,\sigma}(v)$ (respectively $N_{r,\sigma}(v)$) as*

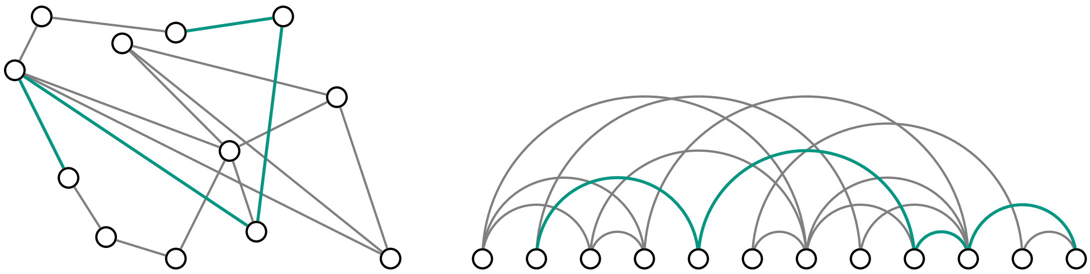
$$\begin{aligned} N_{l,\sigma}(v) &\coloneqq N(v) \cap \left\{ u \in V(G) \,\middle|\, u \prec v \right\} \\ N_{r,\sigma}(v) &\coloneqq N(v) \cap \left\{ u \in V(G) \,\middle|\, v \prec u \right\}. \end{aligned}$$

We also define the left (respectively right) degree of v as $\deg_{l,\sigma}(v) \coloneqq |N_{l,\sigma}(v)|$ *(respectively* $\deg_{r,\sigma}(v) \coloneqq |N_{r,\sigma}(v)|$*).*

Where it is apparent from the context which vertex ordering is meant, we omit the subscript σ for left/right neighbourhoods and degrees and write $N_l, N_r, \deg_l, \deg_r$ instead of $N_{l,\sigma}, N_{r,\sigma}, \deg_{l,\sigma}, \deg_{r,\sigma}$. See Figure 2.1 for an example.

Definition 2.3: *Let G be a graph and σ a vertex ordering of G. A path $P = (v_1, \ldots, v_k) \subseteq G$ is called* σ-monotone *if $v_1 \prec_\sigma v_2 \prec_\sigma \cdots \prec_\sigma v_k$.*

When it is obvious which vertex ordering we mean, we again omit the σ and speak of monotone paths instead of σ-monotone paths. A monotone path is depicted in Figure 2.2.



(a) The graph G. **(b)** The vertex ordering σ of G.

Figure 2.2: A graph G with a vertex ordering σ. The green edges form a longest monotone path.

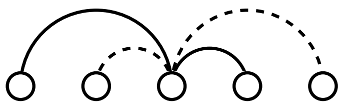


Figure 2.3: A pattern. Non-edges are depicted as dashed lines; where no line is drawn, the edge is undecided.

2.2 Patterns and Combinability

Here we introduce the concept of patterns, their relevance for characterizing graph classes and the notion of combinability of patterns and parametrized families of patterns. The definitions for patterns and the union-intersection-property as well as the operations and notations introduced for patterns are based on those introduced in a survey on forbidden patterns in the characterization of graph classes by Feuilloley and Habib [FH21].

Definition 2.4: *A* pattern *P is an ordered trigraph, that is a quintuple $(V, E, N, U, \prec)$ with a set V of vertices ordered by $\prec$, a set E of edges, a set N of nonedges and a set U of undecided edges. For all pairs $u, v \in V$ with $v \neq u$, $\{u, v\}$ is in exactly one of E, N or U.*

Since $U = \binom{V}{2} - E - N$, we write $P = (V, E, N, \prec)$ instead of $P = (V, E, N, U, \prec)$ unless U has some structure we want to emphasize. Figure 2.3 depicts a pattern on five vertices.

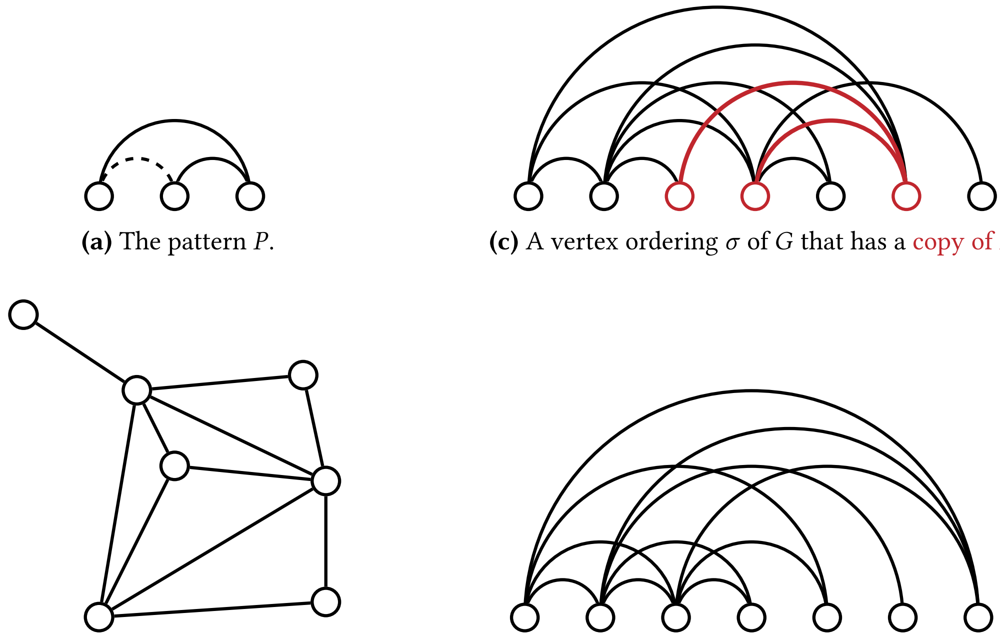
We say a vertex ordering σ of a graph G *has a copy of* or *contains* a pattern $P = \left(V = \{v_1, \ldots, v_k\}, E, N, \prec\right)$ if we can find a subset $\left\{v_{i_1}, \ldots, v_{i_k}\right\} \subseteq V(G)$ of vertices that 'match' the pattern's vertices. $v_{i_1}, \ldots, v_{i_k} \in V(G)$ match $v_1, \ldots, v_k \in V(P)$ if for all $l, r \in [k]$, we get

$$v_{i_l} \prec_\sigma v_{i_r} \iff v_l \prec v_r$$

and

$$\begin{aligned} v_l v_r \in E &\implies v_{i_l} v_{i_r} \in E(G), \\ v_l v_r \in N &\implies v_{i_l} v_{i_r} \notin E(G). \end{aligned}$$

A vertex ordering σ that has no copy of P we call *P-free*. If a graph G permits a P-free vertex ordering σ, we also say that G is P-free. If σ/G are P-free, we often say that σ/G *avoid* P. An example for a pattern P and graph G which can be ordered to contain P or to avoid it is shown in Figure 2.4.



(a) The pattern P.

(c) A vertex ordering σ of G that has a copy of P.

(b) The graph G.

(d) A P-free vertex ordering σ' of G.

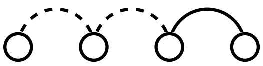
Figure 2.4: A pattern P and a graph G which has a vertex ordering σ that contains P and a P-free vertex ordering σ'. Since G is not a trigraph and so has no undecided edges, we simply omit the non-edges when drawing vertex orderings.

Feuilloley and Habib [FH21] define some operations and notation on patterns. Figure 2.5 shows the effect of these operations on an exemplary pattern. They proceed to make some observations about how the characterized graph class changes under the following operations.

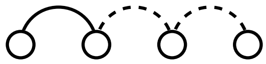
- The *mirror* of pattern $P = (V, E, N, U, \prec)$ is the pattern $\overleftarrow{P} = (V, E, N, U, \succ)$ where $\succ$ inverts the order given by $\prec$.
- The *complement* of pattern $P = (V, E, N, U, \prec)$ is the pattern $\overline{P} = (V, N, E, U, \prec)$.
- A pattern $P_2 = (V_2, E_2, N_2, U_2, \prec_2)$ *extends* the pattern $P_1 = (V_1, E_1, N_1, U_1, \prec_1)$ if $V_1 \subseteq V_2$, $E_1 \subseteq E_2$, $N_1 \subseteq N_2$, $U_2|_{V_1} \subseteq U_1$ and for $u, v \in V_1$ with $u \prec_1 v$, we also have $u \prec_2 v$, so the relative order of V_1 is maintained. Intuitively, this means P_2 can be obtained from P_1 by adding vertices (with all incident edges undecided) and deciding undecided edges.
- A pattern $P = (V, E, N, U, \prec)$ *splits* into patterns P_1 and P_2 (denoted as $P = P_1 \& P_2$) if $P_1 = (V, E + e, N, U - e), P_2 = (V, E, N + e, U - e)$ for some $e \in U$.
- For a pattern P we denote by $\mathcal{C}_P$ the class of P-free graphs. The class of graphs that can avoid a set $\mathcal{P}$ of patterns at the same time is denoted as $\mathcal{C}_{\mathcal{P}}$.

Observation 2.5: *Let P be a pattern and G a graph with a P-free vertex ordering σ. Then each of the following statements hold:*

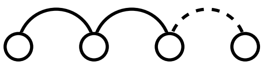
1. *G also avoids $\overleftarrow{P}$ with $\overleftarrow{\sigma}$ (the reverse of σ); hence we have $\mathcal{C}_P = \mathcal{C}_{\overleftarrow{P}}$.*
2. *$\overline{G}$ avoids $\overline{P}$ with σ; hence we have $\mathcal{C}_P = \overline{\mathcal{C}_{\overline{P}}}$.*
3. *If a pattern P' extends P, then G also avoids P' with σ; hence we have $\mathcal{C}_P \subseteq \mathcal{C}_{P'}$.*



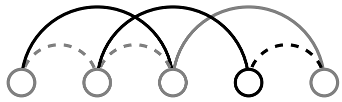
(a) The pattern P.



(b) The pattern $\overleftarrow{P}$.



(c) The pattern $\overline{P}$.



(d) The pattern P'. The edges and vertices that were already present in P are indicated by lower saturation.

Figure 2.5: A pattern P, its mirror $\overleftarrow{P}$ and complement $\overline{P}$, as well as a pattern P' that extends P.

The last relevant definition from [FH21] is the union-intersection-property of sets of patterns, which we here give in the special case that the sets each consist of a single pattern.

Definition 2.6: *Two patterns P and Q have the* union-intersection-property *if* $\mathcal{C}_{\{P,Q\}} = \mathcal{C}_P \cap \mathcal{C}_Q$.

Having defined patterns, we now come to the central definition of this thesis, which builds on patterns and vertex orderings to define the combinability of pattern families.

Definition 2.7: *Let $\mathcal{P}_1, \mathcal{P}_2$ be two parametrized families of patterns $\mathcal{P}_i := \{P_{i,k} \mid k \in \mathbb{N}\}$. We say that $\mathcal{P}_1, \mathcal{P}_2$ are* (f, g)-combinable *if there exist functions $f, g : \mathbb{N}^2 \to \mathbb{N}$ such that for all graphs G and $n_1, n_2 \in \mathbb{N}$ where G avoids P_{1,n_1} and G avoids P_{2,n_2}, there is a vertex ordering σ of G which avoids both $P_{1,f(n_1,n_2)}$ and $P_{2,g(n_1,n_2)}$.*

If $f(n_1, n_2) = c \cdot n_1,\ g(n_1, n_2) = c \cdot n_2$ for some $c \in \mathbb{R}$, we say that $\mathcal{P}_1, \mathcal{P}_2$ are c-combinable*,* perfectly combinable *if $c = 1$. If one or both of $\mathcal{P}_1, \mathcal{P}_2$ consist of a single pattern, the notation is analogous – in particular, two combinable* single-pattern *or* unparametrized *families are always perfectly combinable.*

For single patterns P and Q, the notions of combinability and satisfying the union-intersection-property (UIP) are equivalent: If P and Q have the UIP, any graph that avoids P and Q individually also avoids P and Q simultaneously per definition of the UIP, so P and Q are combinable. For the other direction, let P and Q be combinable. That means we can find a vertex ordering σ of a graph G that avoids both P and Q for all graphs G that are both P-free and Q-free. This exactly matches the definition of the UIP, so P and Q have the UIP. For the sake of consistency, we speak of P and Q being combinable in all such cases.

2.3 Graph Properties

We introduce a definition for graph properties and graph parameters, as well as the properties used later.

Definition 2.8: *A* graph property *is a class $\mathfrak{P}$ of graphs. If $G \in \mathfrak{P}$, we say that G has property $\mathfrak{P}$. If for every graph $G \in \mathfrak{P}$ all induced subgraphs $H \subseteq G$ also have $\mathfrak{P}$, we say that $\mathfrak{P}$ is* hereditary.

A graph parameter $\mathfrak{Q}$ *is a parametrized family of properties $\mathfrak{P}_i$ with $\mathfrak{P}_i \subseteq \mathfrak{P}_{i+1}$ for all i such that for every graph G there exists $i \in \mathbb{N}$ with $G \in \mathfrak{P}_i$. We define the $\mathfrak{Q}$-number of a graph G as* $\mathfrak{Q}(G) := \min\{i \in \mathbb{N} \mid G \in \mathfrak{P}_i\}$.

Hereditary properties can always be characterized by excluding certain subgraphs or families of subgraphs, trivially the family of all graphs without the property. The more interesting question is finding minimal families of forbidden subgraphs, which have been found in many cases. A prominent example is the class of perfect graphs that consists of exactly the graphs with no induced C_{2k-1} or $\overline{C_{2k-1}}$ for any $k \in \mathbb{N}$ [CRST06]. There are also cases where it is more efficient to characterize a property by excluding certain patterns, such as in the case of chordal graphs: A graph G is chordal if and only if it contains no induced cycle of length at least four, giving an infinite minimal family of forbidden subgraphs. An equivalent condition for chordal graphs is this: A graph is chordal if and only if it permits a vertex ordering without . In this case, forbidding a single pattern is sufficient to characterize a property that otherwise requires an infinite family of forbidden subgraphs. There are also parametrized properties that can be characterized by forbidding a single pattern for every parameter instance, such as the class of k-degenerate graphs, for which the forbidden pattern is a k-star with its centre as the last vertex and no non-edges. We now give a general definition for properties and parameters being characterized by patterns.

Definition 2.9: *A graph property $\mathfrak{P}$ is* characterized *by a pattern P if $\mathfrak{P} = \mathcal{C}_P$. We say that $\mathfrak{P} = \mathfrak{P}_P$.*

A graph parameter $\mathfrak{Q}$ is characterized *by a parametrized family of patterns $\mathcal{P}$ if $\mathfrak{P}_i = \mathcal{C}_{P_i}$ for all $i \in \mathbb{N}$. We say that $\mathfrak{Q} = \mathfrak{Q}_\mathcal{P}$.*

In this thesis we restrict our attention to graph parameters $\mathfrak{Q}_\mathcal{P}$ with $\mathcal{P} = \{P_i \mid i \in \mathbb{N}\}$ where for any i the pattern P_{i+1} extends P_i. From Item 3 of Observation 2.5 we can see that for any parametrized family of patterns $\mathcal{P} = \{P_i \mid i \in \mathbb{N}\}$ where P_{i+1} extends P_i and $P_{i+1} \neq P_i$ for all $i \in \mathbb{N}$, the (parametrized) family of properties $\mathfrak{Q}_\mathcal{P}$ is a graph parameter: As P_{i+1} extends P_i, $\mathfrak{Q}_\mathcal{P}$ fulfils $\mathfrak{P}_i = \mathfrak{P}_{P_i} \subseteq \mathfrak{P}_{P_{i+1}} = \mathfrak{P}_{i+1}$. Additionally, since each P_i has to be different from P_{i+1}, the number of vertices in P_i grows at least with the square root of i. This means that any graph G certainly has a $P_{|G|^2}$-free vertex ordering, so there is an i for which $G \in \mathfrak{P}_i$. We sometimes refer to the $\mathcal{P}$-number rather than the $\mathfrak{Q}_\mathcal{P}$-number out of convenience.

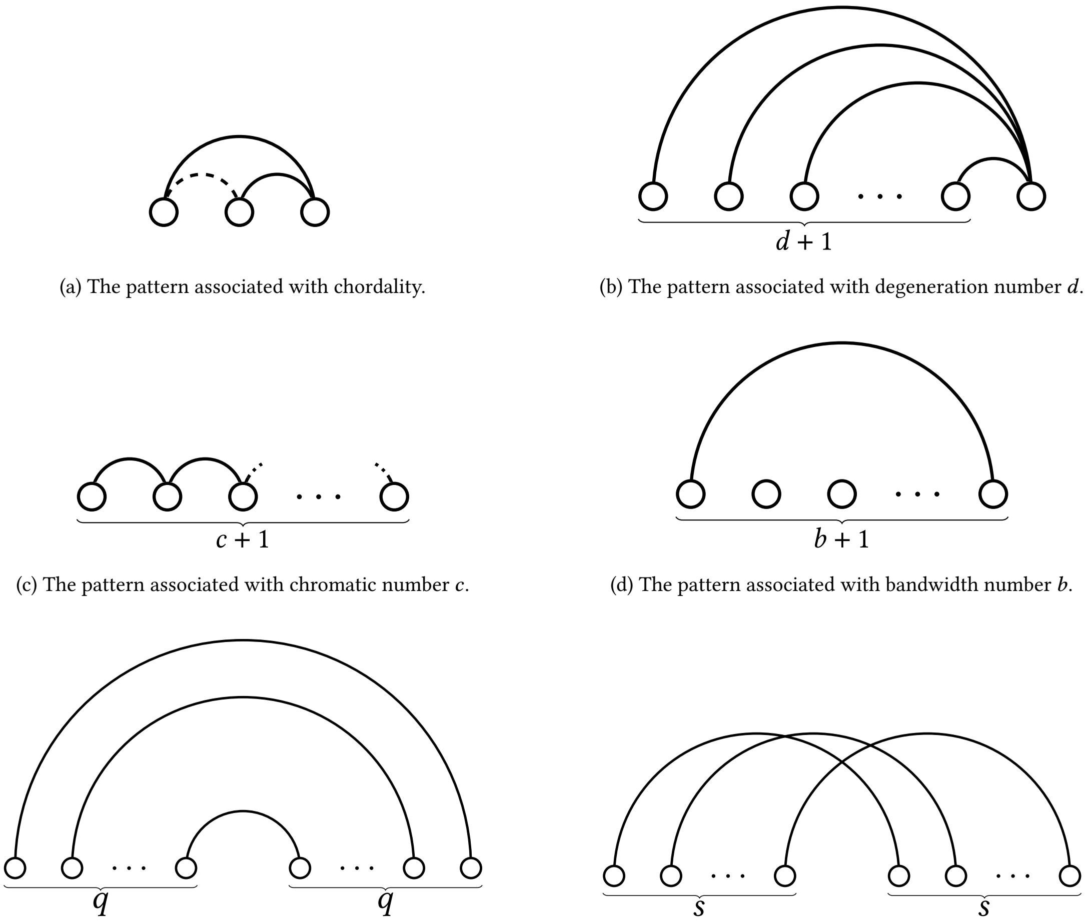
Finally, we extend the notion of combinability to graph properties and parameters: Parameters $\mathfrak{Q}_{\mathcal{P}_1}$ and $\mathfrak{Q}_{\mathcal{P}_2}$ with parametrized families $\mathcal{P}_1$ and $\mathcal{P}_2$ of forbidden patterns are (f, g)-combinable if $\mathcal{P}_1$ and $\mathcal{P}_2$ are (f, g)-combinable.

2.4 Well-Known Graph Parameters and Their Associated Patterns

We now introduce the set of graph parameters that we investigate in this thesis alongside their associated pattern families. We also mention some graph properties that are known to have a strong association to vertex orderings avoiding certain patterns.

Chordality A graph is *chordal* if it is -free. A -free vertex ordering is called a *perfect elimination scheme (PES)*. This characterizes exactly the class of graphs without an induced cycle of length at least four [Ros70].

Degeneration number For $d \in \mathbb{N}$ we look at the pattern P_d that is a star with its centre as the rightmost vertex, formally $P_d \coloneqq ([d+2], \{\{i, d+2\} \mid i \in [d+1]\}, \emptyset, <)$. We call a vertex ordering σ d-degenerate if it avoids P_d. If a graph G has a d-degenerate vertex ordering σ, we also say that G is d-degenerate. The *degeneracy number* dn of G is the minimal $d \in \mathbb{N}$ such that G is d-degenerate.



(a) The pattern associated with chordality.

(b) The pattern associated with degeneration number d.

(c) The pattern associated with chromatic number c.

(d) The pattern associated with bandwidth number b.

(e) The pattern associated with queue number q.

(f) The pattern associated with stack number s.

Figure 2.6: The patterns associated with relevant graph parameters.

An alternative definition for the degeneracy number is $\mathrm{dn}(G) = \max_{H \subseteq G} \delta(H)$, where $\delta(H)$ is the minimum degree of H.

Chromatic number For $c \in \mathbb{N}$ we look at the pattern P_c that is a monotone path, formally $P_c := ([c+1], \{\{i, i+1\} \mid i \in [c]\}, \emptyset, <)$. We will refer to this pattern as $\overrightarrow{P_c}$ throughout the thesis. The *chromatic number* χ of a graph G is the minimum $c \in \mathbb{N}$ such that G avoids P_c.

The better-known alternative definition of chromatic number of G is the minimum number of colours needed to colour the vertices of G such that no two adjacent vertices have the same colour.

Bandwidth number For $b \in \mathbb{N}$ we look at the pattern P_b that is an edge of length b, formally $P_b := ([b+1], \{\{1, b+1\}\}, \emptyset, <)$. The *bandwidth number* bw of a graph G is the minimum $b \in \mathbb{N}$ such that G avoids P_b. This definition is inspired by the bandwidth-problem for sparse matrices.

Queue number For $q \in \mathbb{N}$ we look at the pattern P_q that is a q-rainbow, formally $P_q := ([2q], \{\{i, 2q-i+1\} \mid i \in [q]\}, \emptyset, <)$. The *queue number* qn of a graph G is the minimum $q \in \mathbb{N}$ such that G avoids P_q.

An alternative definition for the queue number of a graph G is the minimum over all vertex orderings σ of the number of queues needed to represent G. A queue is here a set of edges in σ such that no two edges *nest*[1] – that is, a FIFO-ordering of the edges.

Stack number For $s \in \mathbb{N}$ we look at the pattern P_s that is an s-twist, formally $P_s := ([2s], \{\{i, s+i\} \mid i \in [s]\}, \emptyset, <)$. The *stack number* sn of a graph G is the minimum $s \in \mathbb{N}$ such that G avoids P_s.

An alternative definition for the stack number of a graph G is the minimum over all vertex orderings σ of the number of stacks needed to represent G. A stack is here a set of edges in σ such that no two edges *cross*[2] – that is, a LIFO-ordering of the edges.

The patterns associated with these parameters are depicted in Figure 2.6. Since the patterns associated with chordality and degeneration number are asymmetric, we have made an arbitrary choice by placing the star's centre on the rightmost position in the pattern. It is important to make the same choice in both cases, as otherwise chordality and degeneration number would not be combinable even on stars. In fact, even our choice of pattern for degeneration number and its mirror are non-combinable.

[1] Two edges u_1v_1 and u_2v_2 nest if $u_1 \prec_\sigma u_2 \prec_\sigma v_2 \prec_\sigma v_1$

[2] Two edges u_1v_1 and u_2v_2 cross if $u_1 \prec_\sigma u_2 \prec_\sigma v_1 \prec_\sigma v_2$

3 Combinability of Graph Parameters

Here, we discuss the main question of this thesis, namely which pairs of graph parameters are combinable. Figure 3.1 gives an overview of our results. We begin in Section 3.1 by considering the combinations where one parameter is the degeneration number. Here we find an example for perfect combinability and a number of non-combinable parameters. From observations about the latter, we extract a general framework for showing non- combinability with degeneration number. In Section 3.2 we look at bandwidth number and find a general property of families of patterns that guarantees combinability with bandwidth number. With this property established, we give some examples of graph parameters that have the property and investigate whether the bound we find is tight. We end by giving a few pairs of parameters that are not covered by the previous two sections in Section 3.3.

3.1 Degeneration Number

The first property we consider is chordality, since the associated patterns of degeneration number and chordality are very similar. Here, we find an example of perfect combinability – though one of the properties in question is not parametrized, and the generalization natural to the lens of forbidden patterns is not perfectly combinable with degeneration number. We then look at chromatic number (Section 3.1.2), queue number (Section 3.1.3) and stack number (Section 3.1.4), all of which we find to be non-combinable with degeneration number. From the proofs of non-combinability we extract a general framework for showing non-combinability with degeneration number in Section 3.1.5. We end by giving an exemplary application of this framework.

3.1.1 Degeneration Number and Chordality

We show the strongest form of combinability for chordality and degeneration number. This result at first seems unsurprising as the pattern associated with chordality () extends the pattern associated with degeneration number 1 (). We go on, however, to prove that the class of patterns obtained by taking the patterns for degeneration number and making all undecided edges into non-edges is not perfectly combinable with degeneration number.

Proposition 3.1: *Chordality and degeneration number are perfectly combinable.*

For the special case of 1-degenerate graphs, this immediately follows from Item 3 of Observation 2.5. In the general case, direct proof is required. This necessity becomes clear when we consider the patterns of the parametrized generalization of chordality we introduce in Definition 3.2: These patterns are also extensions of the patterns associated with degeneration number. All the same, We find in Proposition 3.3 that they are not perfectly combinable.

Proof of Proposition 3.1. Let G be a chordal, d-degenerate graph, σ a PES of G, and $v \in V(G)$ a vertex of G. Let $k := \deg_l(v)$. Since σ is a PES, we know that $N_l(v)$ forms a clique of size k. Then $C := N_l(v) + v$ is a clique of size $k + 1$ in G. The rightmost vertex of C in any vertex

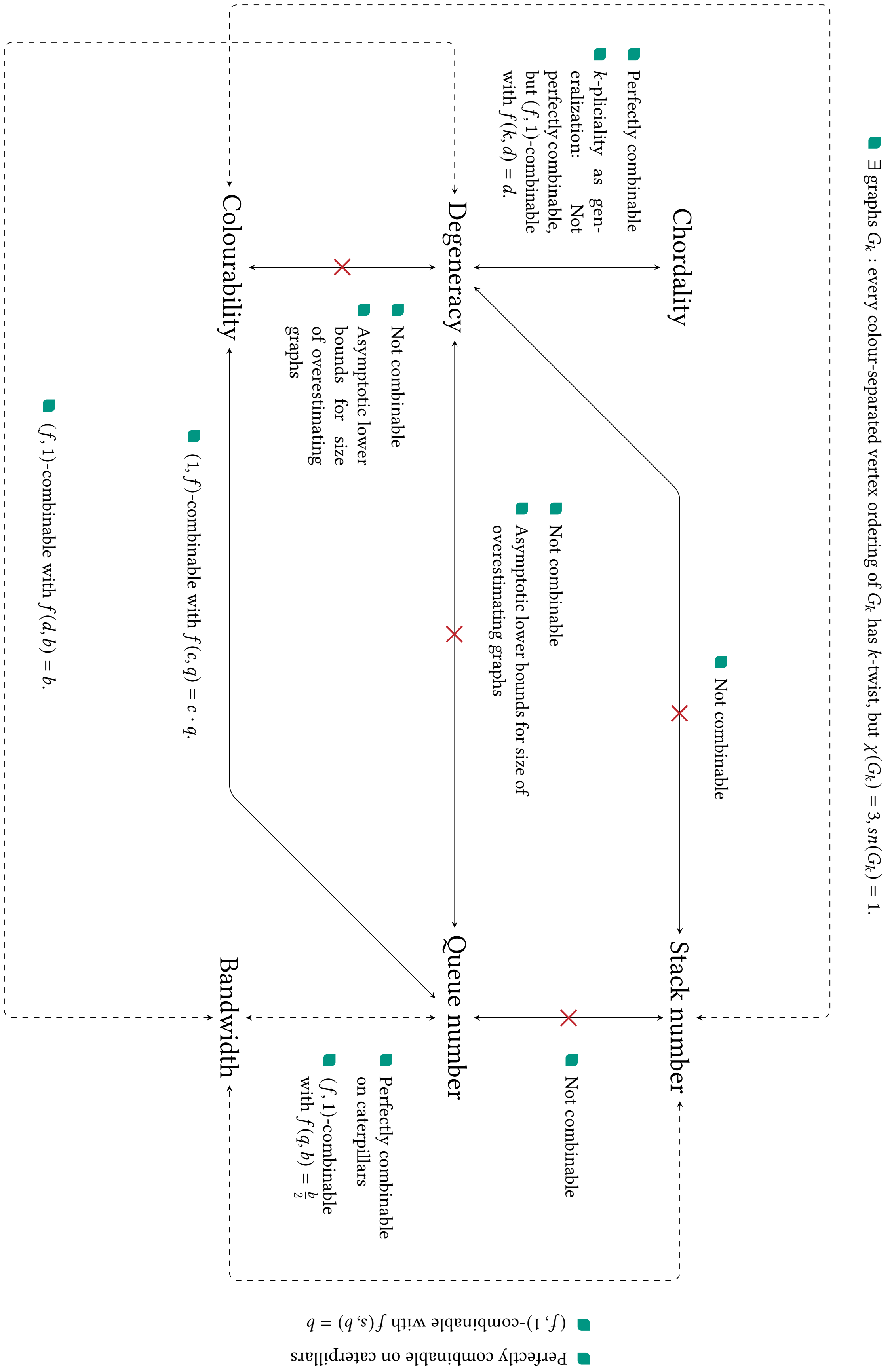
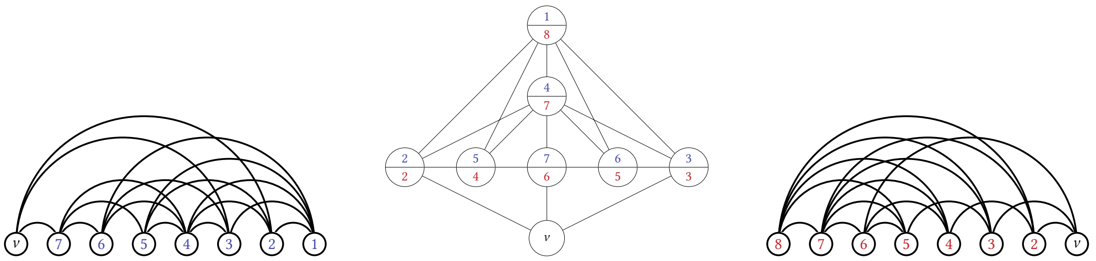


Figure 3.1: Explored relations between graph parameters. The dashed lines mean there are some open questions.



(a) A 2-plicial vertex ordering of G.

(b) The graph G.

(c) A 3-degenerate vertex ordering of G.

Figure 3.2: A 2-plicial, 3-degenerate graph G without a 2-plicial, 3-degenerate vertex ordering. The top labels give the order of a 2-plicial decomposition, the bottom labels give the order of a 3-degenerate decomposition.

ordering has left degree k. Since G is d-degenerate, $k \leq d$. Taking all this together, v was chosen arbitrarily and has left degree at most $k \leq d$. Therefore σ is a d-degenerate vertex ordering of G. ■

Definition 3.2: *Consider the family of patterns $\mathcal{P} = \{P_k \mid k \in \mathbb{N}\}$ with*

$$P_k := \left(V_k := [k+2], E_k := \left\{ \{i, k+2\} \,\middle|\, i \in [k+1] \right\}, \binom{V_k}{2} - E_k, < \right).$$

A graph G is p-plicial if p is its $\mathcal{P}$-number.

Intuitively, the pattern P_k is a star with its centre as the rightmost vertex and all non-edges between the leaves. We observe that the 1-plicial (or simplicial) graphs are exactly the chordal graphs. In fact, pliciality is a natural generalization of chordality when viewed through the lens of forbidden patterns. Observe, too, that the pattern associated with k-pliciality is an extension of the pattern associated with k-degeneracy. With Item 3 of Observation 2.5 we find that $\mathcal{P}$-number and degeneration number are $(f, 1)$-combinable with $f(p, d) = d$. However, $\mathcal{P}$-number and degeneration number are not perfectly combinable, as we can see in the following example:

Proposition 3.3: *$\mathcal{P}$-number and degeneration number are not perfectly combinable.*

Proof. Any d-degenerate graph is d-plicial by Item 3 of Observation 2.5. To show degeneration number and $\mathcal{P}$-number are not perfectly combinable, we must therefore find a d-degenerate, p-plicial graph with $p < d$. The graph G depicted in Figure 3.2 is 2-plicial and 3-degenerate (see the vertex orderings given in Figure 3.2). But G does not have a 2-plicial, 3-degenerate vertex ordering: Any 3-degenerate vertex ordering σ of G must have v as the rightmost vertex, since it is the only vertex of degree at most 3; but the three neighbours of v form an independent set. ■

3.1.2 Degeneration Number and Chromatic Number

Chromatic number begins our series of graph parameters that are not combinable with degeneration number. The construction we use to show this result in the proof of Theorem 3.4 is simply a large tree, so little preliminary work is required. We show in Proposition 3.7 that a large somewhat tree-like structure must be contained in any graph that shows non-combinability for these two parameters.

Theorem 3.4: *Degeneration number and chromatic number are not combinable.*

Proof. Let $k, d \in \mathbb{N}$. We construct a graph where any vertex ordering that overestimates the degeneration number of G by a factor of at most d has a monotone path of length at least k: Let G be a complete d-ary tree of height k and let σ be a vertex ordering of G. G is 1-degenerate and bipartite. Additionally, $|G| \leq d^{k+1}$ and we get $d \geq \sqrt[k]{|G|}$, $k \geq \log_d |G| - 1$.

If any inner vertex v of G is the right end of an inclusion-maximal σ-monotone path, it must have $d + 1$ left neighbours in σ. Therefore σ is not d-degenerate. Otherwise, every inclusion-maximal σ-monotone path has a leaf at its right end. But then there must be a σ-monotone path of length k. ∎

Note that the proof of Theorem 3.4 only forces very slightly superconstant overestimation. However, as we show in Proposition 3.7, this logarithmic/polynomial factor is the largest asymptotically possible. To prove this result, we first define what it means for a vertex to be (d, k)-path- enforcing. Our aim is that any d-degenerate vertex ordering of a graph with a (d, k)-path- enforcing vertex has a monotone path of length at least k, hence the name. We show that every graph G which proves non-combinability of degeneration number and chromatic number has a (d, k)-path-enforcing vertex v for some large values of d and k. Then we use v as the sink in a levelled DAG and finally show that the levels' sizes increase in the same exponential way as those of an r-ary tree for some r.

Definition 3.5: *Let $G = (V, E)$ be a graph, $v \in V$ and $d, k \in \mathbb{N}$. Then v is* (d, k)-path-enforcing *($\mathcal{P}_{d,k}(v)$) if $\left|N(v) \cap \left\{u \in V \,\middle|\, \mathcal{P}_{d,k-1}(u)\right\}\right| \geq d + 1$ or $k = 0$. We additionally define $p_d(v) := \max\left\{k \,\middle|\, \mathcal{P}_{d,k}(v)\right\}$.*

It is not obvious that p_d is well-defined, but we show that it is for d-degenerate graphs in Lemma 3.6. Before we come to that, we give some easy observations to familiarize ourselves with what it means for a vertex to be (d, k)-path-enforcing. Firstly, every vertex of every graph is $(d, 0)$-path-enforcing for any $d \in \mathbb{N}$. If a vertex is (d, k)-path-enforcing, it is also (d, k')-path-enforcing for all $k' < k$. If we need to distinguish between different graphs G and H, we write $p_d^{(G)}(v)$ and $p_d^{(H)}(v)$. Figure 3.3 shows a 2-degenerate graph G with the values of p_2 inscribed in the vertices.

The following observation justifies the name of 'path-enforcing': A d-degeneration σ of a graph G with a (d, k)-path-enforcing vertex v has a σ-monotone path of length k going right from v. We sketch a quick proof of this statement. For $d \in \mathbb{N}, k \geq 1$, a (d, k)-path-enforcing vertex v has at least one $(d, k - 1)$-path-enforcing right neighbour in any d-degeneration. By following this chain, we find a monotone path of length at least k.

Lemma 3.6: *$p_d^{(G)}$ is well-defined for any d-degenerate graph $G = (V, E)$.*

Proof. We proceed in two steps. First we show for that if p_d is well-defined for a graph G, the value of p_d for a single vertex increases by at most one when we add a vertex with at most d edges. Then we use this result to show the statement of Lemma 3.6.

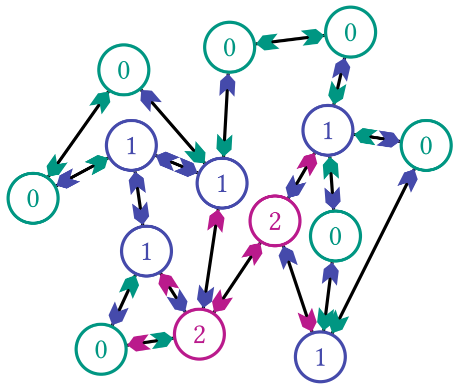


Figure 3.3: A graph G with the values of p_2 inscribed in the vertices. The arrows coming into a vertex indicate the p_2-value of the neighbours.

Let G be a graph where p_d is well-defined. Let G' be a graph obtained from G by adding a vertex v with at most d edges. Then $p_d^{(G')}(v) = 0$ by definition of p_d. We show that $p_d^{(G')}(u) \in \left\{p_d^{(G)}(u), p_d^{(G)}(u) + 1\right\}$ for $u \in V(G)$ by induction on $p_d^{(G)}(u)$.

Base ($p_d^{(G)}(u) = 0$): By definition of p_d we get $\deg_G(u) \leq d$. Since $p_d^{(G')}(v) = 0$, no more than d neighbours of u can be $(d, 1)$-path-enforcing in G', so $p_d^{(G')}(u) \in \{0, 1\}$.

Step: Let $k + 1 := p_d^{(G)}(u)$. By definition of p_d we know that u has at most d neighbours in G that are $(d, k + 1)$-path-enforcing in G. In particular, u has at most d neighbours in G that are $(d, k + 2)$-path-enforcing. By induction hypothesis, $p_d^{(G')}(w) \leq k + 1$ for all neighbours $w \in N(u)$ with $p_d^{(G)}(w) \leq k$. Therefore, $p_d^{(G')}(u) \in \{k + 1, k + 2\}$.

To show that p_d is well-defined on d-degenerate graphs, we use induction on $|G|$:

Base ($|G| \leq d + 1$): By definition we know that $p_d(v) = 0$ for all $v \in V$ with $\deg_G(v) \leq d$.

Step: Let $v \in V$ with $\deg_G(v) \leq d$. By induction hypothesis, $p_d^{(G-v)}$ is well-defined. We know that $p_d^{(G)}(v) = 0$ since $\deg_G(v) \leq d$. But then for $u \neq v$ we already showed that $p_d^{(G)}(u) \in \left\{p_d^{(G-v)}(u), p_d^{(G-v)}(u) + 1\right\}$, so $p_d^{(G)}$ is well-defined.

∎

The proof relies heavily on G being d-degenerate. This condition is indeed necessary as, for example, every vertex of K_{d+2} is (d, k)-path-enforcing for every $k \in \mathbb{N}$. Therefore p_d has no well-defined value for vertices of K_{d+2}. This fits the intuitive property we want from p_d, however: We want to use it to guarantee long monotone paths in a d-degeneration, and K_{d+2} has no d-degeneration, so the premise does not apply in the first place.

Proposition 3.7: *Let $d, k, d', k' \in \mathbb{N}$, $d' > 2d$. Further, let $G_{d',k'}$ be a d-degenerate, k-colourable graph such that every d'-degenerate vertex ordering of $G_{d',k'}$ has a monotone k'-path. Then* $\left|G_{d',k'}\right| \geq \left(\frac{d'-d}{d}\right)^{\frac{k'+1}{k+1}}$.

Proof. To prove this lower bound, we give a d'-degenerate vertex ordering of $G_{d',k'}$ by partitioning the vertices into layers by their p_d-value. We then use the k-colourability of $G_{d',k'}$ to obtain a lower bound on the maximum p_d-value among all vertices. Afterwards, we use a d-degeneration to construct a tree whose root is a vertex with maximum p_d-value and where the children of any vertex are copies of its right neighbourhood with respect to the d-degeneration. We find a lower bound for the size of this tree and an upper bound for the number of copies of each vertex, from which we get a lower bound on the size of $G_{d',k'}$.

We sort the vertices of $G_{d',k'}$ into layers $L_i := \left\{v \in V \,\middle|\, p_{d'}(v) = i\right\}$. The vertices of each layer induce a subgraph of $G_{d',k'}$, which is also k-colourable. Additionally, by definition of $p_{d'}$, a vertex in L_i has at most d' neighbours in $\bigcup_{j\geq i} L_j$. This means that any vertex ordering σ that orders the vertices of $G_{d',k'}$ by $p_{d'}$ in descending order is d'-degenerate. $G[L_i]$is k-colourable for each $i \in \mathbb{N}$. Therefore, $G[L_i]$ permits a vertex ordering with no monotone path of length $k + 1$. We now order the L_i internally in such a way as to avoid monotone paths of length greater than k and externally by i in descending order. The result is a d'-degenerate vertex ordering with a longest monotone path of length at most $r \cdot k + r - 1$, where $r = \max_{v\in V} p_{d'}(v)$. The summand $r - 1$ derives from the $r - 1$ edges that can connect the r monotone paths we found within each layer. By our choice of $G_{d',k'}$, we know that $r \cdot k + r - 1 \geq k'$. Thus, $r \geq \frac{k'+1}{k+1}$.

Now let σ_d be a d-degeneration of $G_{d',k'}$. We orient the edges $E := E(G_{d',k'})$ according to σ_d; that is $\overrightarrow{E} := \left\{\overrightarrow{uv} \,\middle|\, uv \in E, v \prec_{\sigma_d} u\right\}$. We construct the tree T by choosing a copy of some $v_r \in L_r$ as the root. For each vertex v of T that is a copy of some $u \in V\left(\overrightarrow{G_{d',k'}}\right)$, T also contains unique copies of all $w \in N_-(u)$ as children of v. Since we used a vertex ordering to orient the edges, $\overrightarrow{G_{d',k'}}$ is a DAG and so T has finite height. Since two vertices may share a σ_d-right neighbour in $G_{d',k'}$, T may contain multiple copies of the same vertex $v \in V(G_{d',k'})$. Figure 3.4 schematically depicts how T is constructed from v_r and σ_d.

We now group all vertices v that have the same distance δ to the root into levels Λ_δ. Let $v \in V(G_{d',k'})$ with $p_{d'}(v) = i$ for some $i > 0$. v has d' neighbours u with $p_{d'}(u) \geq i - 1$ by definition of $p_{d'}$. But v has at most $d < d'$ outgoing edges, since σ_d was a d-degeneration. Therefore, v must have at least $d' - d$ incoming edges from vertices u_i with $p_{d'}(u_i) \geq i - 1$. From this we get $\left|\left\{v \in \Lambda_i \,\middle|\, p_{d'}(v) = r - i\right\}\right| \geq (d' - d)^i$ for $i \leq r$. But since every vertex v with $p_{d'}(v) > 0$ has incoming edges, T has at least r layers. This immediately gives $|T| \geq (d' - d)^r$. In order to get a lower bound on $\left|G_{d',k'}\right|$ from this, we bound how many times an individual vertex $v \in V(G_{d',k'})$ may appear in T. We show by induction on i that any vertex $v \in V(G_{d',k'})$ can appear at most d^i times in Λ_i:

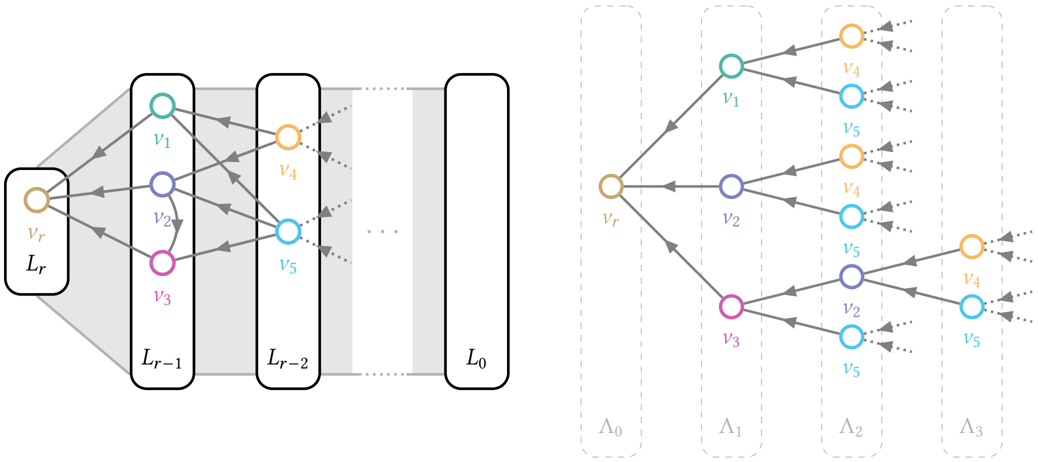
Base: For $i = 0$, we have $|\Lambda_0| = 1$, so v_r appears exactly once.

Step: Consider Λ_{i+1}. We know that v appears exactly once for every $u \in \Lambda_i$ such that v has an outgoing edge to u in $G_{d',k'}$. By induction hypothesis, any such u appears at most d^i times in Λ_i. Since σ_d is a d-degeneration, v has at most d outgoing edges in $G_{d',k'}$. Together, we find that v appears at most $d \cdot d^i = d^{i+1}$ times in Λ_{i+1}.

With this, we get

$$\left|G_{d',k'}\right| \geq \frac{|\Lambda_r|}{d^r} \geq \left(\frac{d' - d}{d}\right)^r = \left(\frac{d' - d}{d}\right)^{\frac{k'+1}{k+1}},$$

which is polynomial in $\frac{d'}{d}$ and exponential in $\frac{k'}{k}$. The condition $d' > 2d$ is needed to guarantee $\frac{d'-d}{d} > 1$, which is necessary to get exponential growth in r. ∎



(a) The layers L_i of $G_{d',k'}$ with $v_r \in L_r$. The edges are oriented according to a d-decomposition σ_d.

(b) The tree constructed by copying vertices as often as needed with levels Λ_i.

Figure 3.4: The p_d-layers of $G_{d',k'}$ and the tree T rooted in v_r.

We have required $d' > 2d$ in our proof. This is likely not a tight condition; in fact we expect $d' > (1+\varepsilon)d$ to be sufficient for $\varepsilon > 0$. We want to point out, however, that the condition $d' > d$ is necessary: For $d' = d$, $G_{1,k'} = P_{2k'}$ has $2k'$ vertices and is $(k = 2)$-colourable, so $|G_{1,k'}|$ is linear in $\frac{k'}{k}$. $P_{2k'}$ is a valid choice for $G_{1,k'}$ since any 1-degenerate vertex ordering has at most two monotone paths. By pigeonhole principle, one of these paths must be of length at least k'.

3.1.3 Degeneration Number and Queue Number

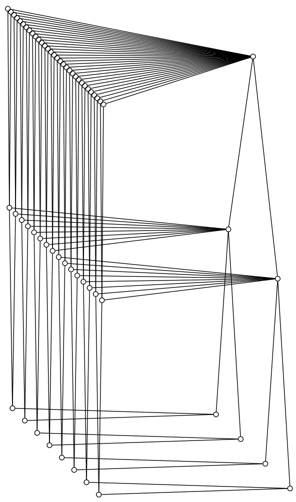
In this section, we show that queue number and degeneration number are not combinable. To this end, we want to construct a family of 2-degenerate graphs $G_{d,k}$ with queue number 2 with k sets of edges such that in any d-degenerate vertex ordering of $G_{d,k}$, at least one edge from each of the k sets nests inside an edge from each other set. Our approach is to start with a grid with 2 columns and k rows and modify it such that every d-degenerate vertex ordering has to have vertices first from one column and then from the other, but the orders of the vertices from rows of the first and second columns are mirrored. The modification to the grid consists of copying each vertex many times, so that each vertex has d neighbours in the set we want to immediately precede it in the ordering for the grid.

Construction 3.8: *For all k, l we define $G_{k,l}$ recursively: $G_{k,1} := K_{k,1}$ with leaves L_1 and centre v_1. For $l \geq 1$ we define $G_{k,l+1} := kG_{k,l} + K_{k^{2l+1},1} + E_{l+1}$. We again denote the leaves of $K_{k^{2l+1},1}$ by L_{l+1} and the centre by v_{l+1}. E is a set of edges that connect v_{l+1} to all copies of v_l, as well as connecting each vertex of L_l to exactly k unique vertices of L_{l+1}.*

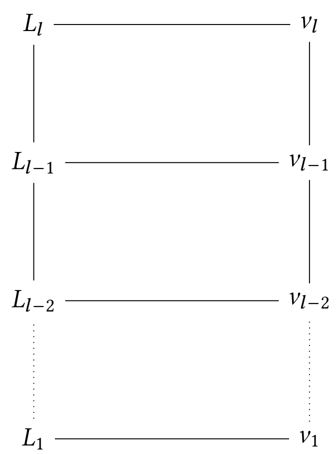
Lemma 3.9: *$G_{k,l}$ of Construction 3.8 has queue number 1 and degeneration number 2.*

Proof. We prove these stronger statements by induction on l:

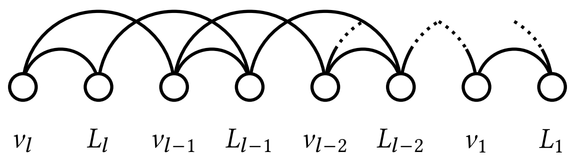
1. $G_{k,l}$ has a 2-degenerate vertex ordering σ with v_l as the leftmost vertex in σ.
2. $G_{k,l}$ has a 2-rainbow-free vertex ordering σ' with v_l as the leftmost vertex, followed immediately by the vertices of L_l.



(a) $G_{2,3}$ with L_i on the left side, v_i on the right and i in descending order from top to bottom.



(b) The grid-like structure formed by $G_{k,l}$. All edges are actually disjoint unions of stars.



(c) The 2-rainbow-free ordering of the grid-like structure of $G_{k,l}$. The edges are again disjoint unions of stars, which have an internal 2-rainbow-free ordering.

Figure 3.5: $G_{2,3}$, the lattice-like structure of $G_{k,l}$ and the 2-rainbow-free ordering of the lattice.

Base: $G_{k,1} = K_{k,1}$ has a vertex ordering with v_1 as the leftmost vertex and left degree 1 for every other vertex. As $G_{k,1}$ is a star, any vertex ordering is 2-rainbow-free, including the one with v_1 as the leftmost vertex.

Step: Let σ_l be the 2-degenerate vertex ordering of $G_{k,l}$ described in Item 1. We construct a 2-degenerate vertex ordering σ_{l+1} of $G_{k,l+1}$ as follows: We place v_{l+1} at the leftmost position. Right of that, we put all vertices of the copies of $G_{k,l}$ in the order given by σ_l. The order between copies is chosen so that all copies of one vertex are placed in an interval. We then place all vertices of L_{l+1} to the right. Since the vertices of L_{l+1} have degree 2, they also have left degree at most 2. By induction hypothesis, all vertices of the copies of $G_{k,l}$ have left degree at most 2. v_{l+1} being the leftmost vertex has left degree 0.

Let σ'_l be the 2-rainbow-free vertex ordering described in Item 2. Then a 2-rainbow-free vertex ordering σ'_{l+1} of $G_{k,l+1}$ can be obtained as follows: v_{l+1} is placed leftmost, immediately followed by the vertices of L_{l+1}. The vertices of L_{l+1} are ordered internally in the same order as the vertices of the copies of L_l they are adjacent to. They are followed by the vertices of the copies of $G_{k,l}$ in the order given by σ'_l, where the copies of any one vertex again form an interval and are always ordered with the same internal order.

We now show that σ'_{l+1} is 2-rainbow-free: Because L_{l+1} and v_{l+1} induce a star and by induction hypothesis, any 2-rainbow would have to be of the form

$$x_1 <_{\sigma'_{l+1}} x_2 <_{\sigma'_{l+1}} v_l \leq_{\sigma'_{l+1}} y_1 <_{\sigma'_{l+1}} y_2.$$

But the only left neighbour of any copy of v_l is v_{l+1}, the leftmost vertex. Therefore, y_1 cannot be a copy of v_l, since that would leave no candidates for x_1. Neither can $x_1 = v_{l+1}$: That would leave only the copies of v_l as candidates for y_2. But since $v_l \leq_{\sigma'_{l+1}} y_1 <_{\sigma'_{l+1}} y_2 = v_l$, the only remaining candidates for y_1 are copies of v_l, which we have already excluded. Finally, by our internal ordering of the vertices of L_{l+1}, there can be no 2-rainbow between L_{l+1} and the copies of L_l.

∎

Theorem 3.10: *Queue number and degeneration number are not combinable.*

Proof. $G_{k+1,l}$ of Construction 3.8 has queue number 1 and degeneration number 2 by Lemma 3.9, but any k-degenerate vertex ordering of $G_{k+1,l}$ has an l-rainbow under some edge of $K_{k+1,l}$, as we show by induction on l:

Base: $G_{k+1,1} = K_{k+1,1}$ has a 1-rainbow, as it has an edge.

Step: In any k-degenerate vertex ordering of $G_{k+1,l+1}$, some vertices of L_{l+1} must be right of every vertex from a copy of L_l. Additionally, some copies of v_l must be right of v_{l+1}. But since there is an edge between each vertex of L_{l+1} and v_{l+1}, the above claim follows.

∎

The graphs $G_{k,l}$ of Construction 3.8 once again rapidly grow very large. However, in Proposition 3.13 we show an exponential lower bound to the order of a graph overestimating the rainbow number, similar to Proposition 3.7 for chromatic number. To do this, we use the Riffle Lemma by Katheder et al. [KKPU24] and the observation that at least half the vertices of a graph G must have at most average degree. We here also give a useful corollary to the Riffle Lemma, which had been known much longer; it was first proved by Pemmaraju in their PhD thesis [Pem92].

Lemma 3.11 (Riffle Lemma): *Let G be a graph with a vertex ordering σ such that σ has no $(r+1)$-rainbow, and let $V_1, \ldots, V_k$ be a partition of $V(G)$. Let σ' be a vertex ordering of G such that for vertices $u, v \in V_i$ we have $u \prec_{\sigma'} v$ whenever $u \prec_\sigma v$. Then σ' has no $(k \cdot r + 1)$-rainbow. If G is bipartite with parts A and B and there exists an l such that $A = \bigcup_{i=1}^{l} V_i$, then σ' has no $(2 \cdot l \cdot (k-l) \cdot r)$-rainbow.*

Corollary 3.12: *Let G be a bipartite graph with parts A and B, let $r \in \mathbb{N}$ and let σ be a vertex ordering of G such that σ has no $(r+1)$-rainbow. Then G has a separated vertex ordering σ' (i.e. $a \prec_{\sigma'} b$ for all $a \in A, b \in B$) such that σ' has no $(2 \cdot r + 1)$-rainbow.*

Proposition 3.13: *Let $d, q, d', q' \in \mathbb{N}$, $d' > 2d$ and let $G_{d',q'}$ be a d-degenerate graph with queue number q such that every d'-degenerate vertex ordering of $G_{d',q'}$ has a $(q'+1)$-rainbow. Then $\left|G_{d',q'}\right| \geq c^{\frac{q'}{q}}$ for some $c > 1$.*

Proof. Let σ be a vertex ordering of $G_{d',k'}$ with no $(q+1)$-rainbow. We now define the partition $V_1, \ldots, V_k$ of $V(G_{d',q'})$: V_1 is the set of vertices of degree at most $2d$ in $G_1 := G_{d',q'}$ and V_i is the set of vertices of degree at most $2d$ in $G_i := G_{i-1} - V_i$ for $i > 1$. Since G_i is d-degenerate for all $i \in \mathbb{N}$, the average degree of G_i is at most $2d$. By pigeonhole principle, at least half of the vertices of G_i are in V_i. Therefore V_i is empty for $i > \log_2(|G_{d',q'}|)$.

Consider the vertex ordering σ' of G which orders the vertices of V_i in the same order as σ and which orders the V_i among each other in decreasing order. By Lemma 3.11, σ' has no $(k \cdot r + 1)$-rainbow. But by our choice of the V_i, σ' is $2d$-degenerate. By definition of $G_{d',q'}$, we know $\frac{q'}{q} \leq k \cdot r$. This gives us $|G_{d',q'}| \geq \left(2^{\frac{1}{q}}\right)^{\frac{q'}{q}}$ and $2^{\frac{1}{q}} > 1$. ∎

Excursion: Queue Number in Bipartite Graphs We end with a little excursion: It is known by a counting argument that there are 3-regular (and necessarily 3-degenerate) graphs with unbounded queue number, however no examples have so far been found [Woo08, HLR92b]. In fact, together with Lemma 11 of [DW05] we find that there are bipartite graphs with maximum degree 3 and unbounded queue number. Our contribution to this search are a family of 3-degenerate bipartite graphs with unbounded queue number in any 3-degenerate vertex ordering, as well as a bipartite graph with unbounded queue number and one 2-regular part. The latter claim builds on Theorem 3.19, a well-known combinatorial result by Erdös and Szekeres [ES35].

Theorem 3.14 (Theorem 1 of [Woo08]): *For all $\Delta \geq 3$ and for all sufficiently large $n > n(\Delta)$, there exists a simple Δ-regular graph G with queue number at least $c\sqrt{\Delta} n^{\frac{1}{2} - \frac{1}{\Delta}}$ for some absolute constant Δ.*

Lemma 3.15 (Lemma 10 of [DW05]): *Let G be a graph, D a subdivision of G with at most one subdividing vertex per edge. If D has queue number q, then G has queue number at most $2q(q+1)$.*

Lemma 3.16: *For every $q \in \mathbb{N}$ there exists a bipartite graph with one 2-regular part and one 3-regular part with queue number at least q.*

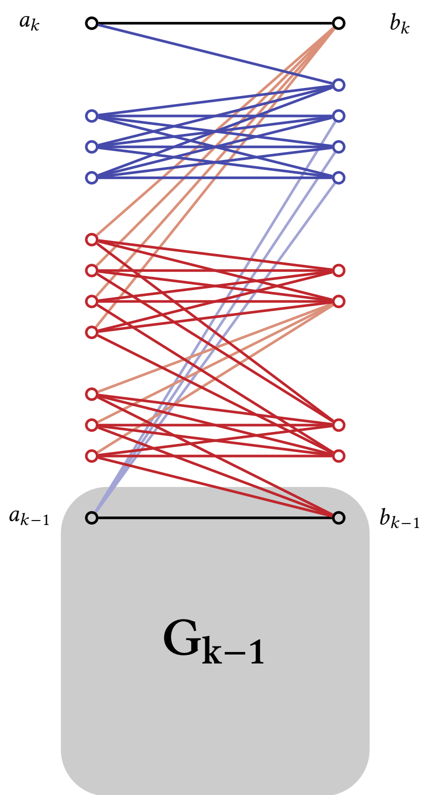
Proof. Let H be a 3-regular graph with queue number at least $2q(q+1)$. H exists by Theorem 3.14. Then G which we obtain from H by subdividing every edge of H once is bipartite with one 2-regular part and one 3-regular part. Additionally, by Lemma 3.15, G has queue number at least q. ∎

Proposition 3.17: *For every $k \in \mathbb{N}$, there exists a 3-degenerate bipartite graph G_k such that for every 3-degenerate vertex ordering σ of G_k, σ induces a k-rainbow.*

Proof. The construction depicted on the right guarantees a k-rainbow under $a_k b_k$. We prove this by induction on k:

Base ($k = 1$): $G_1 = K_2$ is trivially 3-degenerate and bipartite and has a 1-rainbow.

Step: The construction to the right ensures that a_k must be the rightmost vertex in any 3-degenerate vertex ordering of G_k. In particular, a_k must be right of a_{k-1}. The construction also ensures that b_k must be left of b_{k-1}. By induction hypothesis, G_{k-1} has a $k-1$-rainbow R_{k-1} under $a_{k-1}b_{k-1}$, so $R_k = R_{k-1} + a_k b_k$ is a k-rainbow under $a_k b_k$. ∎



Corollary 3.18: *This already shows that queue number and degeneration number are not perfectly combinable even on bipartite graphs.*

Theorem 3.19 (Erdős-Szekeres theorem, [ES35])**:** *Let $l, r \in \mathbb{N}$ and let $a_1, a_2, \ldots, a_k$ be a sequence of $k = (l-1)(r-1)+1$ distinct integers. Then there exists a monotonically increasing subsequence of length at least l or a monotonically decreasing subsequence of length at least r.*

Proposition 3.20: *For every $k \in \mathbb{N}$, there exists a bipartite graph G_k where one part has maximum degree 2 and every separated vertex ordering σ of G_k induces a k-rainbow.*

Proof. We define $G_k = (A_k \cup B_k, E_k)$ with

$$A_k := \left[(2k-1)^2 + 1\right],$$
$$B_k := \left\{\left(i, (2k-1)^2 + 1 - (i-1)\right) \,\middle|\, i \in A_k\right\},$$
$$ab \in E_k \iff a \in b.$$

Let σ be a separated vertex ordering of G_k. Without loss of generality the vertices of A_k are to the right of those of B_k. Further, we can assume that the vertices of A_k are in ascending order, otherwise we relabel them and their correspondents in B_k.

Denote by $(b_i)_{i \in A_k}$ the sequence of the vertices $b_i = (i, (2k-1)^2 + 1 - (i-1)) \in B_k$. By Theorem 3.19, there exists a subsequence (b'_j) of (b_i) of length $2k$ that is either monotonically increasing or monotonically decreasing in the first entry. If monotonically increasing, the edges $\left\{r_j b'_j \,\middle|\, b'_j = (l_j, r_j)\right\} \in E_k$ form a k-rainbow in σ. Otherwise, the edges $\left\{l_j b'_j \,\middle|\, b'_j = (l_j, r_j)\right\} \in E_k$ form a k-rainbow in σ. ∎

Corollary 3.21: *This implies that there exist bipartite graphs with maximum degree 2 in one part and arbitrarily high queue number, since by Corollary 3.12 any vertex ordering of a bipartite graph can be separated while at most doubling the rainbow number.*

3.1.4 Degeneration Number and Stack Number

The approach we use to that degeneration number is not combinable with stack number is very similar to the one we used for queue number: We again use a modified grid, though this time we also add a diagonal to enforce that one column comes before the other. To generate a large twist, we want to ensure that the rows of both columns are ordered in the same way.

Construction 3.22: *We define $G_{k,l} \coloneqq \bigcup_{i=1}^{l} S_i$ where $S_i = k^i K_{k^{l-1},1}$. The leaves A_i of S_i are each adjacent to exactly k unique vertices in A_{i+1}. The centres B_i of S_i are each adjacent to exactly k unique vertices of B_{i+1}. Finally, each vertex in B_l is adjacent to exactly k unique vertices of A_1.*

Lemma 3.23: *$G_{k,l}$ of Construction 3.22 has stack number 2 and degeneration number 2.*

Proof. A vertex ordering σ of $G_{l,k}$ with $A_i <_\sigma A_j$ for $i > j$, $B_i <_\sigma B_j$ for $i > j$ and $A_i <_\sigma B_j$ for all i, j is 2-degenerate. To show 3-twist-freeness, we observe that the S_i have a 2-twist-free ordering. We use this ordering internally for the S_i. We further order A_i, B_i as $A_j <_\sigma A_i$ for $i < j$, $B_i <_\sigma B_j$ for $i < j$ and $A_i <_\sigma B_j$ for all i, j. The vertex ordering of $G_{k,l}$ we obtain in this way is 3-twist-free. ∎

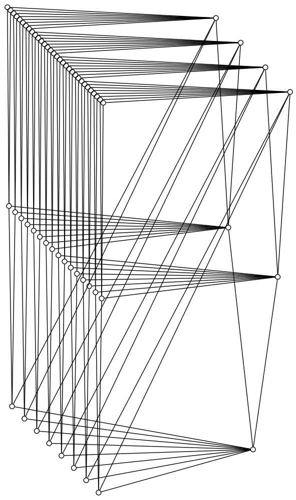
Theorem 3.24: *Stack number and degeneration number are not combinable.*

Proof. $G_{k+1,l}$ of Construction 3.22 has stack number 2 and degeneration number 2 by Lemma 3.23. But any k-degenerate vertex ordering σ of $G_{k+1,l}$ must have some vertex of A_{i+1} right of some vertex of A_i. σ must also have some vertex of B_{i+1} right of some vertex of B_i. Finally, σ mus have some vertex of A_1 right of some vertex of B_l. Together, these vertices with the edges from A_i to B_i induce an l-twist. ∎

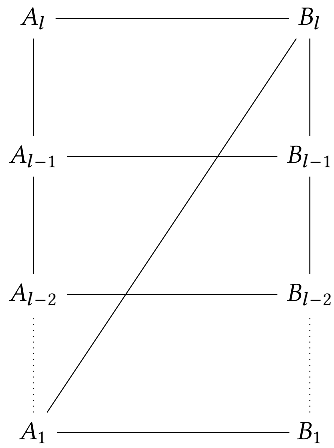
3.1.5 General Framework

Upon closer investigation, the methods we used to show non-combinability in Sections 3.1.2 to 3.1.4 share a common core: The constructions in all three cases take the forbidden structure of the parameter we want to combine with degeneration number, add a path between all adjacent vertices and then blow that path up into a tree to enforce a certain ordering. We here formalize this common core into a framework for showing non-combinability with degeneration number. This approach only works for families of patterns $\mathcal{P}$ with bounded degree and no non-edges between consecutive vertices. We now give a formal definition and a label to the set of pattern families we use for our general framework.

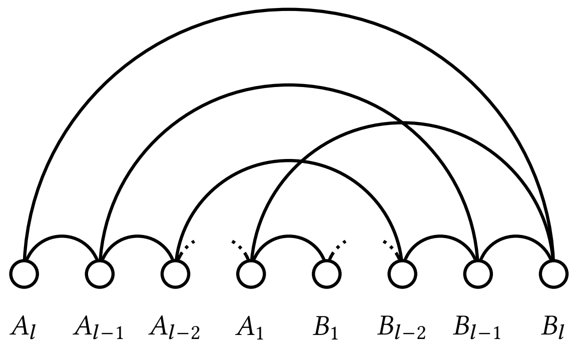
Definition 3.25: *Let $\Delta \in \mathbb{N}$. We denote the set of Δ-bounded left-degree, f-increasing, not consecutive non-edge pattern families by $\Pi_{\Delta,f}$. Consider a family of patterns $\mathcal{P} = \{P_k \mid k \in \mathbb{N}\}$ where N_k denotes the set of non-edges of P_k. Then $\mathcal{P}$ is in $\Pi_{\Delta,f}$ if*



(a) $G_{2,3}$ with A_i on the left side, B_i on the right and i in descending order from top to bottom.



(b) The grid-like structure formed by $G_{k,l}$. All edges are actually disjoint unions of stars.



(c) The 3-twist-free ordering of the grid-like structure of $G_{k,l}$. The edges are again disjoint unions of stars, which have an internal 2-twist-free ordering.

Figure 3.6: $G_{2,3}$, the lattice-like structure of $G_{k,l}$ and the 3-twist-free ordering of the lattice.

- P_{k+1} *extends* P_k,
- P_k *has bounded left degree* Δ *and*
- $\{v_i v_{i+1} \in N_k \mid i \in [f(k) - 1]\} = \emptyset$.

Construction 3.26: *Let* $f : \mathbb{N} \to \mathbb{N}$ *be a function and let* $\mathcal{P} = \{P_k \mid k \in \mathbb{N}\}$ *be a family of patterns. Denote by* P'_k *the pattern obtained from* P_k *by making all edges between successors undecided. Then* $G_{k,d}$ *is constructed as follows: Begin with a* d*-ary tree of height* $f(k)$*. Then evenly connect the vertices in layers* i *and* j *if* $v_i v_j \in E(P_k)$ *for* $i < j$, $i, j \in \mathbb{N}$*. The vertices are connected evenly if every vertex in layer* i *has exactly* d^{j-i} *neighbours in layer* j *and each vertex in layer* j *has exactly one neighbour in layer* i*.*

Lemma 3.27: *Let* $\Delta \in \mathbb{N}$, $\mathcal{P} = \{P_k \mid k \in \mathbb{N}\} \in \Pi_\Delta$*. If there exists an* $l \in \mathbb{N}$ *such that* $G_{k,d}$ *from Construction 3.26 avoids* P_l *for all* $k, d \in \mathbb{N}$*, then* $\mathcal{P}$*-number and degeneration number are not combinable.*

Proof. Since P'_k has maximum left degree Δ, we observe that $G_{k,d+1}$ is $(\Delta + 1)$- degenerate: In any vertex ordering of $G_{k,d+1}$ where the vertices are ordered layer by layer from the underlying tree's root on the left to the leaves on the right, the maximum left degree of any vertex is at most $\Delta + 1$. Further, any $d + 1$-degenerate vertex ordering σ of G must have copies of each of the vertices of P_k in the order they have in P_k. These vertices form a copy of P_k in σ by our construction of $G_{k,d+1}$. Now suppose there exists some $l \in \mathbb{N}$ such that $G_{k,d}$ avoids P_l for all $k, d \in \mathbb{N}$. This means the $\mathcal{P}$-number of $G_{k,d}$ is at most l. Then $G_{k,d}$ is $(\Delta + 1)$-degenerate and has $\mathcal{P}$-number at most l, but any d-degenerate vertex ordering of $G_{k,d}$ has a copy of P_k. Since k, d were chosen arbitrarily, this means that degeneration number and $\mathcal{P}$- number are not combinable. ∎

Applications Using this framework, we can considerably simplify the proofs of Theorems 3.4, 3.10 and 3.24. It is easy to see that their associated pattern families are in Π_1. Then to apply Lemma 3.27, we only need to find an l such that $G_{k,d}$ constructed in Construction 3.26 has chromatic/queue/stack number at most l.

Alternate proof of Theorem 3.4. The resulting $G_{k,d}$ is a complete d-ary tree of height h. Since trees are bipartite, all $G_{k,d}$ avoid $\overrightarrow{P_3}$ and so the claim follows with Lemma 3.27. ∎

Alternate proof of Theorem 3.10. The resulting $G_{k,d}$ permits a 2-rainbow-free vertex ordering σ; indeed such a σ can be obtained from a breadth first search (BFS) starting on the tree's root. The claim then follows from Lemma 3.27. ∎

Alternate proof of Theorem 3.24. Let σ be a BFS ordering of the vertices of $G_{k,d}$, starting at the root. From this we obtain a vertex ordering σ' of $G_{k,d}$ by reversing σ on the BFS-levels $k + 1$ through $2k$. As σ was 3-rainbow-free, σ' is 3-twist-free. The claim then follows from Lemma 3.27. ∎

Finally, we can apply Lemma 3.27 to find an infinite family of patterns that are not combinable with degeneration number.

Proposition 3.28: *Let* $\mathcal{P}$ *be a family of patterns as described in Construction 3.26 and let* Δ *be the maximum left degree. If we can find* $l \in \mathbb{N}$ *such that* P_l *(and therefore all* P_k, $k \geq l$*) has a path of length at least* $\Delta + 2$*, then* $\mathcal{P}$*-number and degeneration number are not combinable.*

Proof. We use the framework introduced by Lemma 3.27: Let $l \in \mathbb{N}$ such that P_l has a path of length at least Δ+2, $k \geq l$. Then we know that $G_{d,k}$ is $(\Delta$+1$)$-degenerate and $(\Delta$+2$)$-colourable. Therefore $G_{d,k}$ admits a vertex ordering σ without a monotone path of length $\Delta + 2$. This vertex ordering σ then avoids P_l. ∎

3.2 Bandwidth Number

Looking at bandwidth number, a sufficient property for combinability is evident, and we document this in Section 3.2.1. We continue in Section 3.2.2 by taking two parameters which have said sufficient property and attempt to improve the bounds from Section 3.2.1, in which we succeed for the heavily restricted case of caterpillars. A general difficulty we encounter in this section is that calculating the bandwidth number of even quite simple graphs, such as caterpillars with hairs of length at most three, is NP-hard [Mon86]. This limits the number of approaches for testing combinability.

3.2.1 Sufficient Property

Let $\mathcal{P}_B = \left\{ P_k^B \,\middle|\, k \in \mathbb{N} \right\}$ be the family of patterns associated with bandwidth number. We observe that a pattern P extends P_k^B if and only if P has an edge of length at least k. Then Item 3 of Observation 2.5 gives us the following result:

Observation 3.29: *Let $\mathcal{P}$ be a family of patterns, $l_{\mathcal{P}} : \mathbb{N} \to \mathbb{N} \in \omega(1)$ a function such that for every $k \in \mathbb{N}$ there exists an edge e in P_k that has length at least $l_{\mathcal{P}}(k)$. Then $\mathcal{P}$-number and bandwidth number are $f, 1$-combinable for $f(p, b) = \min \left\{ k \in \mathbb{N} \,\middle|\, l_{\mathcal{P}}(k) \geq b \right\}$.*

From this, it immediately follows that queue number, stack number, degeneration number and pliciality number as defined in Definition 3.2, as well as a large family of other patterns, are combinable with bandwidth number.

3.2.2 Bandwidth Number with Queue Number and Stack Number

Since bandwidth number is unbounded on trees, which have queue and stack numbers of 1, the question of whether the function we get from Observation 3.29 is optimal arises. While we have been unable to answer this in the general case, we find that both stack number and queue number are perfectly combinable with bandwidth number on caterpillars. Though this may not seem like a strong result, it is not a trivial one either, as degeneration number and bandwidth number are not perfectly combinable even on stars: A star $T_S(2n)$ with $2n$ leaves has degeneration number 1 and bandwidth number $n - 1$, but every 1-degenerate vertex ordering of $T_S(2n)$ has an edge of length $2n - 1$.

Proposition 3.30: *Let G be a caterpillar. G admits a bandwidth-optimal, 2-twist-free and 2-rainbow-free vertex ordering.*

To prove this proposition, we introduce some notation: If σ is a vertex ordering of a caterpillar G, let σ_s be the vertex ordering induced by σ on the spine of G. An *inversion* in σ is any  or  in σ_s. For any leaf of G, we say that it is *estranged* if there is a spinal vertex between it and its parent. See Figure 3.7 for an example of a caterpillar with an inversion and an estranged leaf.

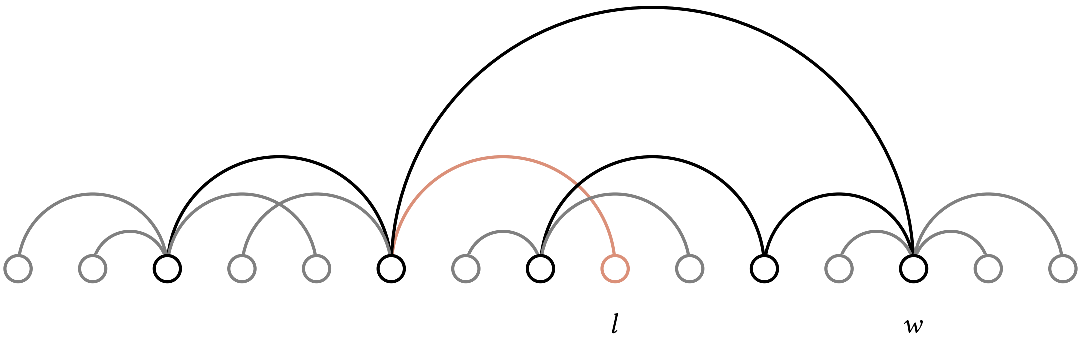


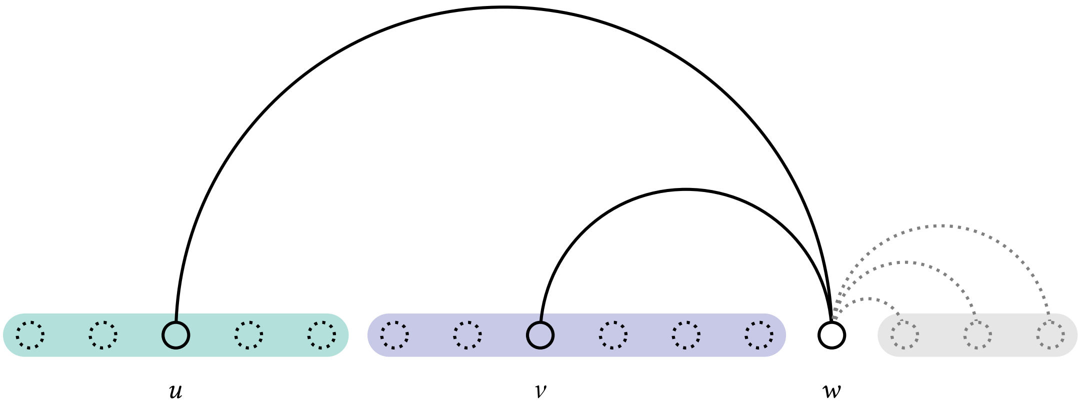
Figure 3.7: A caterpillar with an inversion at w. l is an estranged leaf. The spinal vertices and edges are marked by higher saturation.

Proof of Proposition 3.30. Let G be a caterpillar and let σ be a bandwidth-optimal vertex ordering of G with the least number of inversions. Suppose σ has an inversion. Then there must be an outer inversion, that is an inversion where no spinal edge spans the entirety of the inversion. Without loss of generality it is a rightmost left inversion (). Name the inversion's vertices u, v and w from left to right. There are two cases:

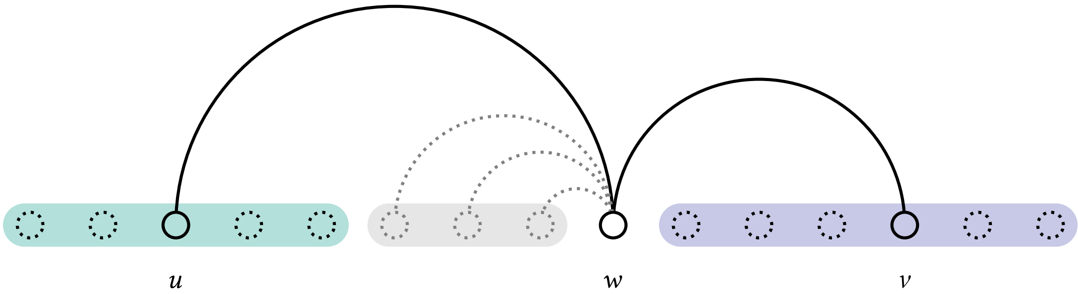
Case 1: There are no spinal vertices right of w. This case is depicted in Figure 3.8. To build a bandwidth-optimal vertex ordering σ' with one fewer inversion, we take the entire subtree under v, mirror it and place it right of all other vertices of σ. If there are leaves to the right of w in σ, it is possible that this stretches vw. If this stretching is due to leaves of the subtree under v, the length of vw cannot increase beyond that of the longest edge to one of these leaves in σ. Otherwise, we can move as many leaves adjacent to the subtree under u or to w as there were vertices between v and w in σ to the left side of w (if there are enough). If there are no more such leaves between v and w, vw has the same length as in σ. Otherwise the length of vw in σ' is at most the length of the longest edge to one of these leaves in σ. In all cases, vw is the only edge whose length increases, and the length of the longest edge in σ is at least the length of vw in σ'.

Case 2: There are spinal vertices right of w. In this case, there is also a leftmost outer right inversion (), whose vertices we label x, y, z from left to right. To obtain a bandwidth-optimal vertex ordering σ' with two fewer inversions, we take the entire subtree S between w and x, mirror it and place it between the other two subtrees. The only edges whose lengths can increase are uw and xz. The largest number of vertices of S under xz and uw in σ' cannot increase. But all vertices of $G - S$ that were not between u and w in σ but are between u and w in σ' were right of w in σ. Therefore the maximum edge length does not increase because of these vertices either. The same holds for x and z.

There is then a bandwidth-optimal, inversion-free vertex ordering σ of G. Without loss of generality, σ minimizes the number of estranged leaves. Suppose σ has an estranged leaf l. Then there must be a spinal vertex w between l and its parent v. By moving l to the near side of v, the only edge that increases in length is uv for some spinal neighbour u of v. But the length of this edge increases only by 1, while the length of vl decreases by at least 1; and in σ, vl was longer than uv.



(a) A caterpillar with an outer single inversion.



(b) The same caterpillar after the inversion has been resolved. Some leaves adjacent to w had to be moved to ensure that vw does not increase in length.

Figure 3.8: Exemplary resolution of a single inversion.

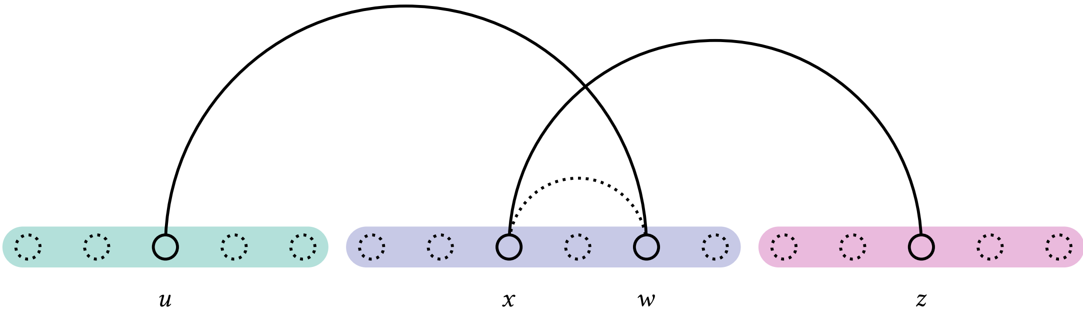
There is then a bandwidth-optimal, inversion-free vertex ordering σ of G with no estranged leaves. σ is already 2-rainbow-free, and the only large twists are between edges connecting leaves adjacent to neighbouring spinal vertices, which can be untangled without increasing the longest edge's length. ∎

3.3 Further Results

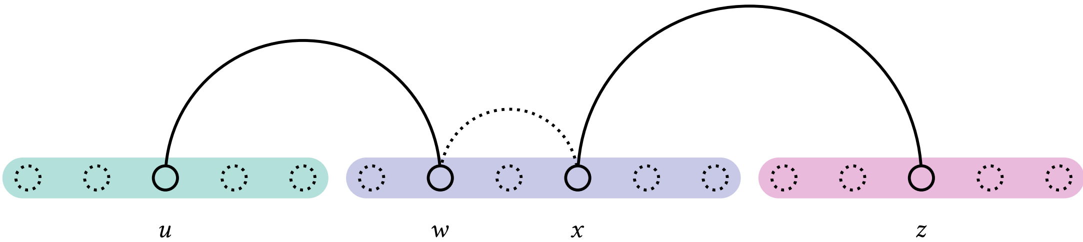
We end this chapter by presenting some rather isolated results. The first of these is another application of the Riffle lemma, showing that queue number and chromatic number are combinable. Although queue number and stack number seem very similar, we next observe in Observation 3.34 that the same approach cannot work to show combinability of chromatic number and stack number. We finish by showing that queue number and stack number are not combinable even on trees in Theorem 3.36.

Proposition 3.31: *Queue number and chromatic number are $(f, 1)$-combinable with $f(q, c) = c^2 \cdot q$.*

Proof. Let G be a graph with $\mathrm{qn}(G) = q$, $\chi(G) = c$. Let σ be a $(q + 1)$-rainbow-free vertex ordering of G and let $V_1, \dots, V_c$ bethe colour classes of G. By Lemma 3.11, we can transform σ into a vertex ordering σ' such that σ' is $(c^2 \cdot q + 1)$-rainbow-free and for all pairs $u \in V_i, v \in V_j, i < j$ we get $u <_{\sigma'} v$. ∎



(a) A caterpillar with an outer double inversion.



(b) The same caterpillar after the inversion has been resolved.

Figure 3.9: Exemplary resolution of a double inversion.

The bound of Proposition 3.31 can be improved to $f(q, c) = c \cdot q$ by using Corollary 3.12:

Proposition 3.32: *Queue number and chromatic number are $(f, 1)$-combinable with $f(q, c) = c \cdot q$.*

Proof. Let G be a graph with queue number q and chromatic number c. We will show that G has a $(c \cdot q + 1)$-rainbow-free vertex ordering σ with no monotone path of length $c + 1$. To do this, we take a $(q + 1)$-rainbow-free vertex ordering σ_q of G and separate the colour classes of V_i of G ($i \in [c]$). The vertices within each colour class are ordered as by σ_q. By Corollary 3.12, we know that the rainbow number rn of any two colour classes V_i, V_j in σ is at most twice the queue number of $G[V_i \cup V_j]$. But since no more than $\frac{c}{2}$ rainbows between colour classes can nest inside one another, we get

$$\mathrm{rn}(\sigma) \leq \frac{c}{2} \cdot \max_{i,j} \overline{\mathrm{qn}}(G[V_i \cup V_j]) \leq \frac{c}{2} \cdot (2\,\mathrm{qn}(G[V_i \cup V_j])) \leq c \cdot q.$$

∎

Before we can prove that this approach cannot work for stack number, we need to introduce a corollary to Theorem 3.19 that connects it to twists and rainbows in vertex orderings.

Corollary 3.33 (to Theorem 3.19)**:** *Let G be a graph and σ a vertex ordering of G. Let $t, r \in \mathbb{N}$. If $M \subseteq E(G)$ is a separated matching of size $(t - 1)(r - 1) + 1$ in σ, then there exists a t-twist or an r-rainbow in σ.*

Proof. Number the vertices of one part of M from left to right. We assign the vertices in the second part of M the same number as their neighbour in M. Then any monotonously increasing subsequence of t numbers in the vertices of the second part of M gives us a t-twist. Conversely, any monotonously decreasing subsequence of r numbers gives us an r-rainbow. The claim then follows from Theorem 3.19. ∎

Observation 3.34: *There is a family of graphs G_k such that stack number and chromatic number of G_k are constant, but any colour-separated vertex ordering of G_k has a k-twist.*

Proof. Consider $G_k := \big((k-1)^4+1\big)K_3$ where the i-th copy of K_3 has vertices $v_{i,1}, v_{i,2}, v_{i,3}$. Obviously, G_k has stack number 1 and chromatic number 3. We now show that any colour-separated layout σ of G has a k-twist. Let V_1, V_2, V_3 be the colour classes of G_k. V_1, V_2 induce a separated matching in σ. Therefore they have either a $(k-1)^2+1$-twist or a $(k-1)^2+1$-rainbow by Corollary 3.33. In the former case we are done. In the latter case we restrict our attention to the triangles involved in the rainbow. Let $R \subseteq E(G_k)$ be the edges of the largest rainbow between V_1 and V_2 in σ. Define

$$\begin{aligned} T' &:= \left\{ i \in \left[\left((k-1)^2+1 \right)^2 + 1 \right] \,\middle|\, v_{i,1}v_{i,2} \in R \right\}, \\ V_i' &:= \left\{ v_{j,i} \in V_i \,\middle|\, j \in T' \right\}. \end{aligned}$$

Applying Corollary 3.33 again, we get a k-twist or a k-rainbow between V_1' and V_3'. Again, in the first case we are done. In the last case, let $R' \subseteq E(G_k[V_1' \cup V_3'])$ be the edges of the largest rainbow between V_1' and V_3' in σ, $T'' := \left\{ i \in T' \,\middle|\, v_{i,1}v_{i,3} \in R' \right\}, V_i'' := \left\{ v_{j,i} \in V_i' \,\middle|\, j \in T'' \right\}$. But since V_1'' and V_2'' also form a k-rainbow, V_3'' and V_2'' must form a k-twist. ∎

It thus remains unclear whether chromatic number and stack number are $(1, f)$-combinable for some function f. The authors' attempts at disproving this by finding graphs G_k with bounded stack number $s \in \mathbb{N}$ and chromatic number $c \in \mathbb{N}$ whose $\overrightarrow{P_{c+1}}$-free orderings have a k-twist have been unsuccessful. The only insight we have gained is that such a witness G_k is unlikely to be bipartite, as a bipartite witness would solve the open question whether queue number is bounded by stack number in the negative. The question was first posed by Heath, Leighton, and Rosenberg [HLR92a], for an overview of the current work on this question see [KKPU24].

Proposition 3.35: *Let G_k be a family of bipartite graphs with bounded stack number $s \in \mathbb{N}$ such that every $\overrightarrow{P_3}$-free vertex ordering of G_k has a k-twist. Then queue number is not bounded by stack number.*

Proof. A separated vertex ordering of G_k is $\overrightarrow{P_3}$-free, and therefore has a k-twist. But then the separated stack number $\overline{\text{sn}}$ is at least k. We can obtain a separated $(k+1)$-rainbow-free vertex ordering from a separated $(k+1)$- twist free vertex ordering by reversing one of the parts [KKPU24]. Therefore separated stack number is the same as separated queue number $\overline{\text{qn}}$. By Corollary 3.12, the separated queue number of G_k is at most twice the queue number of G_k. More precisely, we get

$$k \leq \overline{\text{sn}}(G_k) = \overline{\text{qn}}(G_k) \leq 2 \cdot \text{qn}(G_k),$$

so $\text{qn}(G_k) \geq \frac{k}{2}$ for a graph with bounded stack number s. ∎

Theorem 3.36: *Stack number and queue number are not combinable.*

To show Theorem 3.36, we construct a tree $G_{r,t}$ for all pairs $r, t \in \mathbb{N}$ such that any vertex ordering σ of $V(G_{r,t})$ has either an r-rainbow or a t-twist. Since trees have queue number and stack number 1, the existence of such a tree $G_{r,t}$ proves Theorem 3.36.

We will show that the complete $(2((r-1)(t-1)+2))$-ary tree on r layers is a possible choice for $G_{r,t}$. To do this, we introduce some notation: The *outermost neighbour* u of a vertex v is the one maximizing

$$\left|\left\{x \in N(v) \,\middle|\, u \prec_\sigma x \prec_\sigma v \vee v \prec_\sigma x \prec_\sigma u\right\}\right| =: s(v,u).$$

We now fix some vertex ordering σ and show the following statement:

Lemma 3.37: *The complete* $(2((r-1)(t-1)+2))$*-ary tree on* l *layers* ($T_{(2((r-1)(t-1)+2)),l}$) *has one of the following:*

- *An* r*-rainbow*
- *A* t*-twist*
- *An* l*-rainbow under* vu*, where* v *is the root of* $T_{(2((r-1)(t-1)+2)),l}$ *and* u *is* v*'s outermost neighbour.*

Proof. We prove Lemma 3.37 by induction on l:

Base: Let v be the root, u its outermost neighbour. vu is an edge, and therefore a 1-rainbow, in $T_{(4((r-1)(t-1)+2)),1}$.

Step: Let v be the root of $T_{(2((r-1)(t-1)+2)),l+1}$, u its outermost neighbour. Let x_i be the neighbours of v between them numbered in ascending order from v to u. Without loss of generality $v \prec_\sigma u$. Suppose that for every x_i there is an edge $x_i x_i'$ with $u \prec_\sigma x_i'$ or $x_i' \prec_\sigma x$. Then these edges going out to one side form a separated matching on $\frac{s(v,u)}{2}$ edges. By choice of u, $s(v,u) \geq (r-1)(t-1)+1$, so by Corollary 3.33, there is an r-rainbow or a t-twist in σ and we are done.

Otherwise, there exists an $i \in [s(v,u)]$ such that all neighbours of x_i lie under vu. The tree rooted in x_i is $T_{(2((r-1)(t-1)+2)),l}$, so by induction hypothesis it has an r-rainbow or a t-twist or an l-rainbow under $x_i u_i$ (where u_i is x_i's outermost neighbour). In the first two cases, we are done. But if the subtree under x_i has an l-rainbow under $x_i u_i$, this together with vu forms an $(l+1)$-rainbow under vu in $T_{(2((r-1)(t-1)+2)),l+1}$.

∎

4 Combinability of Individual Patterns

This chapter investigates the combinability of individual patterns. It is concerned mostly with the following conjecture, which arose from the realization that queue number and stack number are not combinable (see Theorem 3.36), and was strengthened since we found it to hold true for patterns on three vertices as documented in Proposition 4.2.

Conjecture 4.1: *If $P = (V, E, N, U, \prec)$ is a pattern and $P' = (V, E, N, U, \prec')$ is a non-isomorphic permutation of P, then P and P' are not combinable.*

Proposition 4.2: *Conjecture 4.1 holds for all patterns on three vertices.*

To prove this proposition, we first make some observations on simple sufficient conditions for the combinability of patterns.

Observation 4.3: *Let P, Q be patterns.*

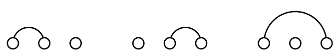
1. *If P is invariant under permutation of the vertices*[1]*, then P is combinable with Q.*
2. *If Q is an extension of P, then P and Q are combinable.*
3. *If P and Q are combinable, so are $\overline{P}$ and $\overline{Q}$, as well as $\overleftarrow{P}$ and $\overleftarrow{Q}$.*

Proof. Item 1 is not immediately obvious, so we give a short proof.

1. Let P be a pattern invariant under permutation of the vertices and let $G = (V, E)$ be a graph with vertex ordering σ. Suppose $V' \subseteq V$ induces a copy of P in σ. Then by permutation invariance, V' induces a copy of P in every vertex ordering σ' of G.
2. Follows directly from Item 3 of Observation 2.5
3. Follows directly from Items 1 and 2 of Observation 2.5

∎

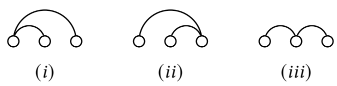
Proof of Proposition 4.2. We group the patterns on three vertices by their underlying trigraph and briefly argue the pairwise non-combinability in each group.



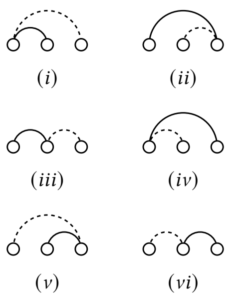
(i) (ii) (iii)

The path P_3 can avoid (i) (respectively (ii)) by placing the central vertex of P_3 to the right (respectively left) of all other vertices, but in no other way. By ordering P_3 as $\overrightarrow{P_3}$, (iii) is avoided. From this we see that (i) and (ii) are not combinable with each other or with (iii).

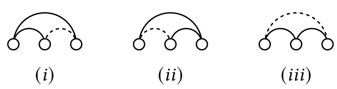
[1] Examples are patterns where two of E, N, U are empty.



Non-combinability of $(i)/(ii)$ and (iii) follows directly from the proof of Theorem 3.4 and Item 1 of Observation 2.5. For (i) and (ii) we consider a star with three leaves and see that it only avoids (i) when two of its leaves are to the left of its centre, but any such ordering has (ii).



Consider the vertex orderings of P_4. Only two of them avoid (i): The first (i)-free vertex ordering has the inner vertices of P_4 on the outside and the leaf edges crossing. This ordering has $(ii) - (iv)$ and (vi). The second (i)-free vertex ordering has a leaf l to the left, followed by the inner vertex v of P_4 not adjacent to l, then the other leaf and the last remaining vertex. This ordering has all of $(ii) - (vi)$. This already shows that (i) is not combinable with $(ii) - (iv)$ and (vi). But (v) is not combinable with (i) either, as we see by considering the vertex orderings of $G = K_3 + K_1$: To avoid (i), the isolated vertex of G must be left of two of the vertices of K_3, but then it has (v).
Any vertex ordering of P_4 without (ii) must have one leaf as the leftmost vertex, immediately followed by its neighbour. The other vertices' order is unrestricted. This already guarantees (iii), (v) and (vi) and so they are not combinable with (ii). To see that (iv) is also not combinable with (ii), we look again at a star with three leaves. This star must have two leaves left of its centre to avoid (ii), but then it has (iv).
In order to avoid (iii), the central two vertices of P_4 must be right of its leaves. But in any such ordering, all remaining patterns $(iv) - (vi)$ can be found.
All the remaining pairs are mirrors of non-combinable pairs. Therefore, none of them are not combinable by Item 1 of Observation 2.5.



We look at a star with three leaves. To avoid (i), it must have two leaves to the left of its centre, but then it has (ii). (i) and (ii), then, are not combinable. We see that (i) and (iii) are not combinable because any perfect elimination scheme of P_6 has (iii). Further, the pair $\big((ii),(iii)\big)$ is the mirror of the pair $\big((i),(iii)\big)$. Therefore (ii) and (iii) also are not combinable by Item 1 of Observation 2.5.

The remaining groups either contain only a single pattern or are the complement of some of the groups discussed above, so their pairs of patterns cannot be combinable by Item 2 of Observation 2.5. ∎

While we have no proof for Conjecture 4.1 in its most general form, we treat the special case where $N = \emptyset$ and (V, E) is a star in Section 4.2, as well as the case where $|E| + |N| = 1$ and the unique edge in $E \cup N$ is not elongated by the permutation in Section 4.1. For the more complex family of patterns that we call *separated split graphs*, we find that two distinct permutations of such a pattern are not combinable if their cliques are on different sides.

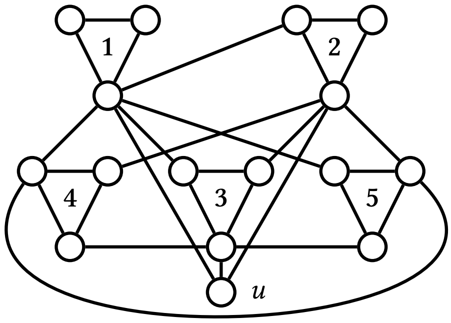


Figure 4.1: A duplicial and co-duplicial graph without a vertex ordering that shows both properties.

We have found no general rules beyond those laid out in Observation 4.3 for generating combinable patterns. From studying patterns on three vertices, one might get the idea that a pattern P and its mirror-complement $\overleftarrow{\overline{P}}$ are always combinable (as they are for and), but this is not even true for the generalization of chordality we discussed in Section 3.1.1. We write duplicial instead of 2-plicial.

Proposition 4.4: *and are not combinable.*

Proof. Consider G depicted in Figure 4.1. To show that G is duplicial and co-duplicial, we convince ourselves that $G - u$ permits an ordering that avoids both and : Take the bottom vertex of triangles 1,2,3 in that order, then the left vertex of triangle four and the right vertex of triangle 5. Since these are the only triangles, this ensures our final vertex order does not contain , and none of these vertices has more than two left neighbours. None of the remaining vertices have three independent neighbours, so any ordering of them will retain -freeness. Taking this ordering and placing u to the left of all other vertices retains -freeness. Placing u to the right retains -freeness.

To see that G has no ordering avoiding both, first observe that in any - free vertex ordering, the leftmost vertex v must fulfil

$$\forall \triangle xyz : \left\{vx, vy, vz\right\} \cap E(G) \neq \emptyset.$$

Note that this also covers the case $v \in \left\{x, y, z\right\}$. It is then easy to see that the bottom vertices of triangles 1,2,3 must be the first three vertices in any -free ordering. But together with u to their right they form . ∎

Analogous constructions can be used for larger stars and their mirrored complements, so p-pliciality and co-p-pliciality are not combinable for $p \geq 2$.

4.1 Single Decided Edges

Since our attempt at finding a family of combinable patterns by taking the mirrored complement failed, we want to see for the special case of symmetric patterns whether they are combinable with their complements. We find that the answer to this question is also no, even in the simple case where only a single edge of each pattern P and Q is decided. That means we study patterns of the form $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$ with special emphasis on the case $k_1 = k_2$. Formally, for $k_1, k_2 \in \mathbb{N}$, the pattern $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$ is defined as $\Big([k_1 + k_2 + 2], \big\{\{k_1 + 1, k_1 + 2\}\big\}, \emptyset, <\Big)$. $\overset{k_1}{\overline{\circ\;\circ}}\ \overset{\cdots}{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$ is defined analogously as $\Big([k_1 + k_2 + 2], \emptyset, \big\{\{k_1 + 1, k_1 + 2\}\big\}, <\Big)$. We begin by some observations about these patterns and the graphs that avoid them. During this study, we encounter some evidence for Conjecture 4.1 in Proposition 4.7.

Observation 4.5: *Let $k_1, k_2 \in \mathbb{N}$. The graphs that avoid $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$ are exactly the graphs with a vertex cover of size at most $k_1 + k_2$.*

Proof. First we show that the class of graphs that avoid $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$ is characterized fully by $k_1 + k_2 =: k$. To this effect, we convince ourselves that every $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$-free graph is $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$-free and vice versa: Let G be a graph on n vertices, σ a vertex ordering of G without $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$. We modify this to

$$\sigma'(v) := \begin{cases} n - \sigma(v) + 1, & \sigma(v) \leq k_2 \\ \sigma(v) - k_2 & \text{otherwise} \end{cases}$$

and see that σ' avoids $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$: Otherwise there is an edge with endpoints u, v such that $n - k_2 > \sigma'(v) > \sigma'(u) > k_1$. But then $\sigma(u), \sigma(v) > k$, contradicting the $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$-freeness of σ. The other direction follows from an analogous argument. The transformation from σ to σ' is depicted in Figure 4.2.

To show the original claim, it then suffices to show that the $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$-free graphs are precisely those that have a vertex cover of size k.

$\Longrightarrow$ Let G be $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$-free, σ a vertex ordering avoiding $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$. This means there is no edge uv with $\sigma(u) > k$ and $\sigma(v) > k$. Then the first k vertices of σ are a vertex cover.

$\Longleftarrow$ Let G have a vertex cover C of size at most k. Then a vertex ordering σ of G that satisfies $\sigma(c) < \sigma(v)$ for all $c \in C, v \in V(G) - C$ is $\overset{k}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}$-free.

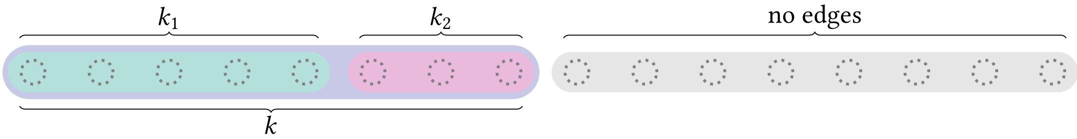
∎

Lemma 4.6: *Let $k_1, k_2, n \in \mathbb{N},\ k := k_1 + k_2$. The complete split graph $G := (C \cup I, E)$ with $|C| = k,\ |G| = n$ permits exactly one $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$-free vertex ordering (up to isomorphism).*

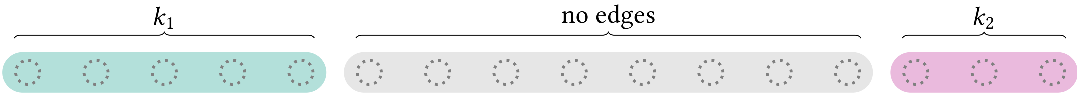
Proof. First we observe a sufficient condition for a vertex ordering σ of G to avoid $\overset{k_1}{\overline{\circ\;\circ}}\ \widehat{\circ\circ}\ \overset{k_2}{\overline{\circ\;\circ}}$:

$$\sigma(v) \in [k_1 + 1, n - k_2] \iff v \in I \tag{$\star$}$$

To verify that Condition ($\star$) is sufficient, observe that no edge can have both endpoints in $[k_1 + 1, n - k_2]$ if $\sigma^{-1}\big([k_1 + 1, n - k_2]\big) = I$ since I is independent. If G is a clique ($|I| \leq 1$), then G only permits a single ordering up to isomorphism. Suppose G is not a clique, so $|I| > 1$. We will now show that in this case, Condition ($\star$) is also necessary. Suppose that there is a single vertex $v \in C$ such that $\sigma(v) \in [k_1 + 1, n - k_2]$. Then there also exists a second vertex $u \in V(G)$ with $\sigma(u) \in [k_1 + 1, n - k_2]$. This u exists because $\big|\sigma^{-1}\big([k_1 + 1, n - k_2]\big)\big| = n - k_1 - k_2 = |I| > 1$. Again, uv is an edge with both endpoints in $[k_1 + 1, n - k_2]$. ∎



(a) A $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}$-free vertex ordering σ of a graph G.



(b) The $\overset{k_1}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2}{\circ\;\circ}$-free vertex ordering σ' of G.

Figure 4.2: Transforming a $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}$-free vertex ordering σ of a graph G into a $\overset{k_1}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2}{\circ\;\circ}$-free vertex ordering σ'.

Proposition 4.7: *Let $k_1, k_1', k_2, k_2' \in \mathbb{N}$ with $k_1 + k_2 = k_1' + k_2'$, $k_1 \neq k_1'$. Then $\overset{k_1}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2}{\circ\;\circ}$ and $\overset{k_1'}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2'}{\circ\;\circ}$ are not combinable.*

Proof. Without loss of generality $k_1 > k_1'$. Let $G := (C \cup I, E)$ be the complete split graph with $|C| = k_1 + k_2$, $|I| = k_2' - k_2 + 1$. By Lemma 4.6, G permits exactly one $\overset{k_1}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2}{\circ\;\circ}$- free ordering σ. It then suffices to show that this ordering contains $\overset{k_1'}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2'}{\circ\;\circ}$. This can be seen by looking at $u := \sigma^{-1}(k_1' + 1), v := \sigma^{-1}(k_1' + 2)$. We know $u \in C$, so $uv \in E$. Additionally, $k_1' + 2 \leq n - k_2 - (|I| - 1) = n - k_2'$, so we find the forbidden edge. Figure 4.3 shows how we find $\overset{k_1'}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2'}{\circ\;\circ}$ in σ. ∎

Proposition 4.8: *Let $k \in \mathbb{N}$. Then $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k}{\circ\;\circ}$ and $\overset{k}{\circ\;\circ}\;\overset{\cdots}{\circ\circ}\;\overset{k}{\circ\;\circ}$ are not combinable.*

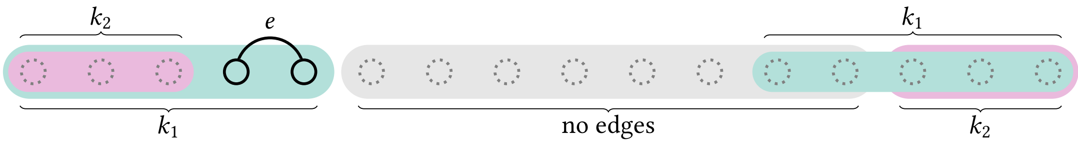
Proof. We observe that the ($\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k}{\circ\;\circ}$,$\overset{k}{\circ\;\circ}\;\overset{\cdots}{\circ\circ}\;\overset{k}{\circ\;\circ}$)-free graphs are exactly the split graphs $G = (C \cup I, E)$ with $|C|, |I| \leq 2k$: From Observation 4.5 we know that the graphs G that avoid $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k}{\circ\;\circ}$ are exactly those with a vertex cover of size at most $2k$. This is equivalent to saying that $G - X$ is independent for some $X \subseteq V(G)$ with $|X| \leq 2k$. But if G is also $\overset{k}{\circ\;\circ}\;\overset{\cdots}{\circ\circ}\;\overset{k}{\circ\;\circ}$-free, the same argument gives us $X' \subseteq V(G)$ of size at most $2k$ $\overset{k}{\circ\;\circ}\;\overset{\cdots}{\circ\circ}\;\overset{k}{\circ\;\circ}$ so that $\overline{G} - X'$ is independent. Therefore, $G - X'$ is a clique. Together, we find that G can be split into an independent set I and clique C such that $I \subseteq X'$ and $C \subseteq X$ with at most $2k$ vertices each.

We know from Lemma 4.6 that the complete split graph G_C with clique and independent set of size $2k$ has a unique vertex ordering σ that avoids $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k}{\circ\;\circ}$. It thus suffices to find a non-edge between any $u, v \in \sigma^{-1}([k + 1, 3k])$. But since $\sigma^{-1}([k + 1, 3k])$ are exactly the vertices of I, such a non-edge exists. ∎

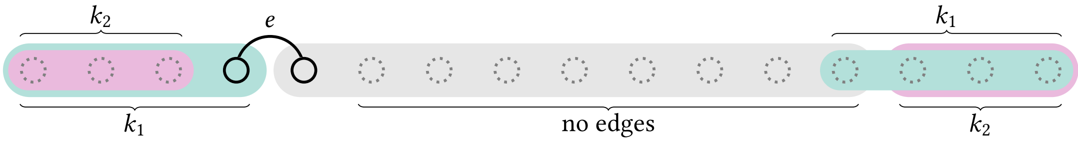
We conclude this section by giving an infinite family of non-trivially combinable pairs of patterns, as well as a strictly stronger statement than Proposition 4.8.

Proposition 4.9: *Let $k_1, k_2 \in \mathbb{N}$. $\overset{k_1}{\circ\;\circ}\;\overset{\frown}{\circ\circ}\;\overset{k_2}{\circ\;\circ}$ and $\overset{k_2}{\circ\;\circ}\;\overset{\cdots}{\circ\circ}\;\overset{k_1}{\circ\;\circ}$ are combinable if and only if $0 \in \{k_1, k_2\}$.*

Proof. Let $k = k_1 + k_2$. First, we show that $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}$ and $\overset{\cdots}{\circ\circ}\;\overset{k}{\circ\;\circ}$ are combinable: We already know that the only graphs in question are split graphs with $|C|, |I| \leq k$. But then any vertex ordering of G that has the vertices of C left of those of I avoids $\overset{k}{\circ\;\circ}\;\overset{\frown}{\circ\circ}$ and $\overset{\cdots}{\circ\circ}\;\overset{k}{\circ\;\circ}$.



(a) Case 1: $k_1 - k_1' > 1$



(b) Case 2: $k_1 - k_1' = 1$

Figure 4.3: The $\overbrace{\circ\ \circ}^{k_1}\ \circ\!\frown\!\circ\ \overbrace{\circ\ \circ}^{k_2}$-free vertex ordering σ of a complete split graph G with $|C| = k_1 + k_2 = |I|$ with a copy of $\overbrace{\circ\ \circ}^{k_2}\ \circ\overset{\cdots}{\ }\circ\ \overbrace{\circ\ \circ}^{k_1}$ around e.

For the other direction, let $k_1, k_2 \neq 0$. We know that the complete split graph G with $|C| = k_1 + k_2 = |I|$ has a unique vertex ordering σ that avoids $\overbrace{\circ\ \circ}^{k_1}\ \circ\!\frown\!\circ\ \overbrace{\circ\ \circ}^{k_2}$. This vertex ordering has $\sigma^{-1}([k_1+1, n-k_2]) = I$ with $n := 2(k_1+k_2)$. If $\left|\sigma^{-1}([k_2+1, n-k_1]) \cap I\right| \geq 2$, we find $\overbrace{\circ\ \circ}^{k_2}\ \circ\overset{\cdots}{\ }\circ\ \overbrace{\circ\ \circ}^{k_1}$. Let $k_1 \geq k_2$ without loss of generality. Then

$$\begin{aligned}\left|\sigma^{-1}([k_2+1, n-k_1]) \cap I\right| &= \left|[k_2+1, n-k_1] \cap \sigma(I)\right| \\ &= \left|[k_2+1, n-k_1] \cap [k_1+1, n-k_2]\right| \\ &= \left|[k_1+1, n-k_1]\right| \\ &= 2(k_1+k_2) - 2k_1 \\ &\geq 2\end{aligned}$$

∎

4.2 Permutations of Stars

We now turn our attention to pairs of patterns P, Q whose underlying graph is the same star. In Proposition 4.12 we show such pairs are non-combinable if one of P, Q has more than half its leaves to the left of the centre and the other has more than half to the right. We finish by proving Conjecture 4.1 for all P, Q with the same underlying star in Theorem 4.16: No two patterns that are permutations of the same star are combinable.

Definition 4.10: *For $l, r \in \mathbb{N}$, $n = l + r$ we denote the star with n leaves by $T_S(n)$. The pattern obtained by ordering $T_S(n)$ with l vertices left and r vertices right of the centre (with no non-edges) we denote by $\overrightarrow{T_S}(l, r)$.*

Lemma 4.11: *Any ordering of $G = C_5 \square E_{2n-1}$ contains $\overrightarrow{T_S}(n, n)$, but each of $\overrightarrow{T_S}(n, n+1)$ and $\overrightarrow{T_S}(n+1, n)$ can be avoided.*

Proof. To show that G avoids $\overrightarrow{T_S}(n, n+1)$ and $\overrightarrow{T_S}(n+1, n)$, we define the notion of a block ordering of G, and then give such an ordering that avoids $\overrightarrow{T_S}(n, n+1)$. The vertices of G can be partitioned into five independent sets according to which of the vertices of C_5 they are a copy of. If $v\ inV(C_5 \square E_{2n-1})$ is a copy of $x \in V(C_5)$, we say that x is the *type* of v. By a block ordering

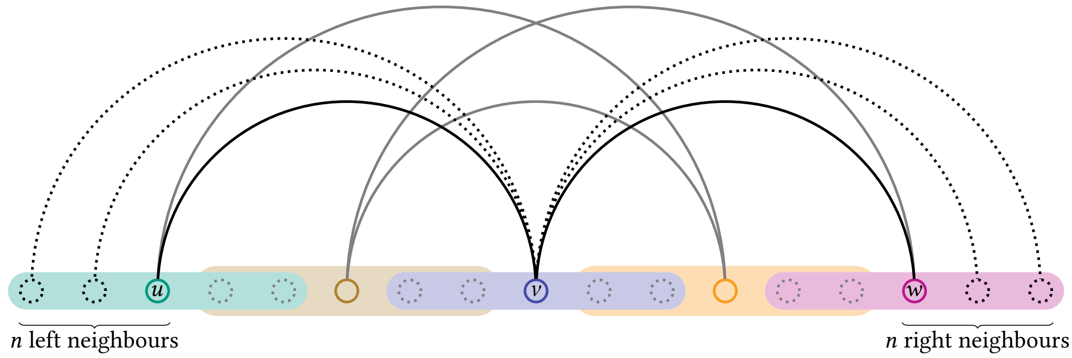


Figure 4.4: The copy of $\overrightarrow{T_S}(n,n)$ in $C_5 \square E_{2n-1}$.

of G we mean a tuple $\mathfrak{B} = ((\mathcal{T}_1, \mathcal{N}_1), \ldots, (\mathcal{T}_k, \mathcal{N}_k))$ with $\mathcal{T}_i \in V(C_5)$, $\mathcal{N}_i \in \mathbb{N}$, $\sum_{i=1}^{k} \mathcal{N}_i = |G|$. A vertex ordering σ of G realizes $\mathfrak{B}$ if for all $i \in [k]$, the vertices in $\sigma^{-1}\left(\left[\sum_{j=1}^{i-1} \mathcal{N}_j, \sum_{j=1}^{i} \mathcal{N}_j\right] \cap \mathbb{N}\right)$ all have type $\mathcal{T}_i$. The key observation is that all vertices inside one block 'behave' in the same way. This means their left and right degrees, as well as the blocks they have neighbours in, are identical. If for a pattern P, all vertex orderings σ of G that realize $\mathfrak{B}$ avoid P, we also say that $\mathfrak{B}$ avoids P. The following block ordering of G avoids $\overrightarrow{T_S}(n, n+1)$, the reverse of this ordering naturally avoids $\overrightarrow{T_S}(n+1, n)$. This can be verified by looking at the left and right degrees of all vertices.

Vertex type $\mathcal{T}$	5	4	2	3	1	5
Block size $\mathcal{N}$	$n-1$	$2n-1$	$2n-1$	$2n-1$	$2n-1$	n
Left degree	0	$n-1$	0	$4n-2$	$3n-2$	$4n-2$
Right degree	$4n-2$	$3n-1$	$4n-2$	0	n	0

Now let σ be an arbitrary vertex ordering of G. For each $i \in [5]$, consider the copy of i that is exactly in the middle of all copies of i. These five vertices induce a copy of C_5, so there must be an ordered path (u, v, w) among them. But then v has n left neighbours (the copies of u left of u and u itself) and n right neighbours (the copies of w right of w and w itself), so we find $\overrightarrow{T_S}(n,n)$. Figure 4.4 shows the copy of $\overrightarrow{T_S}(n,n)$ in $C_5 \square E_{2n-1}$. ∎

Proposition 4.12: *Let $l, l', r, r' \in \mathbb{N}$ with $l < r$, $l' > r'$, $l + r = l' + r'$. Then $\overrightarrow{T_S}(l,r)$ and $\overrightarrow{T_S}(l',r')$ are not combinable.*

Proof. Without loss of generality let $l \geq r'$. Consider $G := C_5 \square E_{2l-1} + L$. Here, L is a set of leaves such that for every vertex $v \in V(C_5 \square E_{2l-1})$, there are exactly $2(l' - r')$ leaves of L attached to v. Then any vertex ordering of G contains $\overrightarrow{T_S}(l,l)$ with centre v by Lemma 4.11. By pigeonhole principle, $l' - r'$ of the leaves attached to v must be either left or right of v. There are then two cases: If we get $l' - r'$ leaves left of v, v has $l + l' - r' \geq l'$ independent left neighbours and $l \geq r'$ independent right neighbours. In this case we find $T_S(l', r')$. Otherwise, v has l independent left neighbours and $r' + l' - r' = l' \geq r$ independent right neighbours. This yields $T_S(l,r)$.

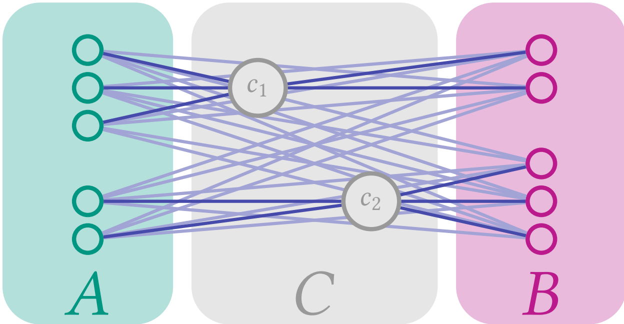


Figure 4.5: The $(2,3)$-star-enforcing graph $G_{FS}(2,3)$ with exemplary vertices $c_1 \in C_1$ and $c_2 \in C_2$ and their neighbours.

To avoid $\overrightarrow{T_S}(l', r')$, choose a vertex ordering of $C_5 \square E_{2l-1}$ that avoids $\overrightarrow{T_S}(l+1, l)$. Then place all leaves adjacent to $v \in V(C_5 \square E_{2l-1})$ immediately left of v if $\deg_l(v) > \deg_r(v)$ and immediately right of v otherwise. We say we place the leaves on the side of largest degree. To avoid $\overrightarrow{T_S}(l, r)$ choose an ordering of $C_5 \square E_{2l-1}$ that avoids $\overrightarrow{T_S}(l, l+1)$ and again place all leaves to the side of largest degree. ∎

Corollary 4.13: *Since $C_5 \square E_{2l-1}$ is triangle-free and we only add leaves to it in our construction of G, the result of Proposition 4.12 also holds for extensions of $\overrightarrow{T_S}(l, r)$ that have some (or all) undecided edges as non-edges.*

To remove the restriction that one star must have more leaves left of the centre and the other must have more leaves right of the centre, we construct a family of graphs that can enforce $\overrightarrow{T_S}(l, r)$ for $l \neq r$, to which we again add leaves in order to find one of our non-combinable patterns.

Construction 4.14: *Let $l, r \in \mathbb{N},\ l, r > 0$. We define the (l, r)-star-enforcing graph $G_{FS}(l, r)$ as $G_{FS}(l, r) := (A \cup B \cup \underbrace{C_1 \cup C_2}_{=:C}, E_{AB} \cup E_C)$ where $|A| = |B| = 2(l + r)$,*

$$\begin{aligned}
C_1 &= \binom{A}{l} \times \binom{B}{r} \\
C_2 &= \binom{A}{r} \times \binom{B}{l} \\
E_{AB} &= \{ab \mid a \in A, b \in B\} \\
E_C &= \left\{ac \,\middle|\, c = (\alpha, \beta) \in C, a \in \alpha\right\} \\
&\quad \cup \left\{cb \,\middle|\, c = (\alpha, \beta) \in C, b \in \beta\right\}.
\end{aligned}$$

Figure 4.5 schematically depicts $G_{FS}(2,3)$. For $n \in \mathbb{N},\ n > 0$ and the special case of stars with all their leaves to one side, we define $G_{FS}(0, n) := K_{n,n} =: G_{FS}(n, 0)$.

Lemma 4.15: *Let $l, r \in \mathbb{N},\ l + r > 0$. Then $G_{FS}(l, r)$ has the following two properties:*

1. *There is a vertex ordering of $G_{FS}(l, r)$ that avoids both $\overrightarrow{T_S}(l, r+1)$ and $\overrightarrow{T_S}(l+1, r)$.*
2. *Any vertex ordering of $G_{FS}(l, r)$ contains a copy of $\overrightarrow{T_S}(l, r)$.*

Proof. We fix $l, r \in \mathbb{N}$. We treat the case $0 \in \{l, r\}$ separately. Without loss of generality let $l = 0$.

1 Every vertex of $G_{FS}(0, r) = K_{r,r}$ has degree r. Therefore there is no vertex with enough neighbours to be the centre of $\overrightarrow{T_S}(1, r)$ or $\overrightarrow{T_S}(0, r + 1)$.

2 Let σ be a vertex ordering of $G_{FS}(0, r)$. The leftmost vertex in σ has r right neighbours, so we find $\overrightarrow{T_S}(0, r)$.

Now we consider the case $l, r \geq 1$.

1 Let σ be a vertex ordering of $G_{FS}(l, r)$ where all vertices of A are left of all vertices of C, which in turn are left of all vertices of B. Then σ avoids $\overrightarrow{T_S}(l, r + 1)$ and $\overrightarrow{T_S}(l + 1, r)$: The vertices of A and B only have edges to one side, so they cannot be the centres of $\overrightarrow{T_S}(l, r + 1)$ or $\overrightarrow{T_S}(l + 1, r)$. But by construction, the vertices of C have degree $l + r$, so they cannot be the centres of $\overrightarrow{T_S}(l, r + 1)$ or $\overrightarrow{T_S}(l + 1, r)$ either.

2 Let σ be an arbitrary vertex ordering of $G_{FS}(l, r)$. Suppose there is a vertex $a \in A$ such that $\left|\{b \in B \,\middle|\, \sigma(b) < \sigma(a)\}\right| \geq l$, $\left|\{b \in B \,\middle|\, \sigma(b) > \sigma(a)\}\right| \geq r$. Then a is the centre of $\overrightarrow{T_S}(l, r)$ with leaves in B.

Otherwise, the intervals containing vertices from A and those containing vertices from B can overlap by at most $l + r$ vertices in A and B each. By discarding the vertices from A and B that are in the overlapping region, we get separated parts A' and B' of size at least $l + r$ each. Without loss of generality A' is left of B'. Suppose any $c \in C_1' := \binom{A'}{l} \times \binom{B'}{r} \subseteq C_1$ is between A' and B'. Then by construction c and its neighbourhood form $\overrightarrow{T_S}(l, r)$.

Otherwise, let a be the rightmost vertex of A' and let b be the leftmost vertex of B'. If $l = r = 1$, let $a' \in A' - a$ and $b' \in B' - b$. Then $c := (a, b) \in C_1'$ is either left of a or right of b. In the first case, cab' forms $\overrightarrow{T_S}(1, 1)$. In the last case, $\overrightarrow{T_S}(1, 1)$ is found in $a'bc$. Now we consider the case where one of l, r is at least 2. There are

$$\begin{aligned}
\binom{|A'|}{l-1}\binom{|B'|}{r-1} &\geq \binom{l+r}{l-1}\binom{l+r}{r-1} \\
&\geq \binom{l+r}{0}\binom{l+r}{1} \\
&= (l+r)
\end{aligned}$$

vertices $C_{a,b} \subseteq C_1'$ adjacent to both a and b. Then, by pigeonhole principle, either l vertices of $C_{a,b}$ are left of a or r vertices of $C_{a,b}$ are right of b. In both cases, we find $\overrightarrow{T_S}(l, r)$.

∎

Theorem 4.16: *Let $n \in \mathbb{N}$ and let $\overrightarrow{T_S}(l, r)$, $\overrightarrow{T_S}(l', r')$, $l' < l$ be two different patterns obtained from of $T_S(n)$ like in Definition 4.10. Then $\overrightarrow{T_S}(l, r)$ and $\overrightarrow{T_S}(l', r')$ are not combinable.*

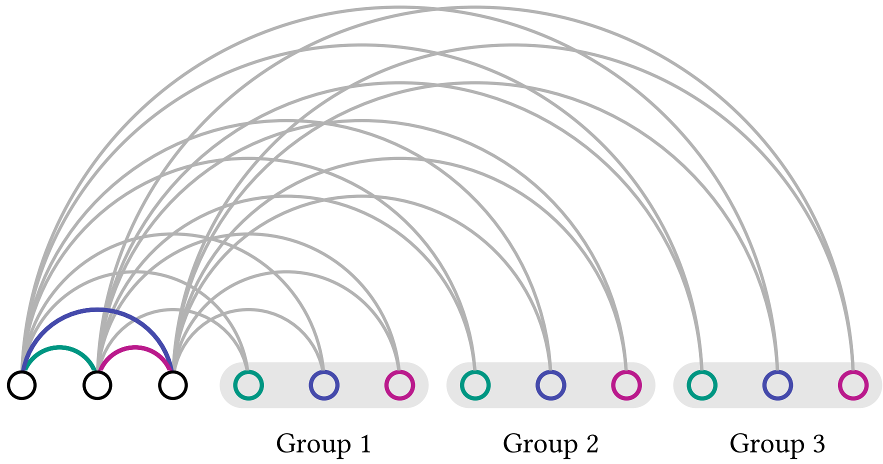


Figure 4.6: A possible group ordering of $S(3, 2, 3)$. The 2-subsets of C are exactly the edges, and the satellites are given the same colours as their bases.

Proof. Consider $G_{FS}(l', r) + L$. Here, L is a set of added leaves such that for every vertex $v \in V\big(G_{FS}(l', r)\big)$, there are exactly $2(l - l')$ leaves of L attached to v. To avoid $\overrightarrow{T_S}(l, r)$, choose a vertex ordering Of $G_{FS}(l', r)$ that avoids $\overrightarrow{T_S}(l' + 1, r)$. Such a vertex ordering exists by Lemma 4.15. By placing all leaves right of their parent, we avoid $\overrightarrow{T_S}(l, r)$. To avoid $\overrightarrow{T_S}(l', r')$, take a vertex ordering of $G_{FS}(l', r)$ that avoids $\overrightarrow{T_S}(l', r + 1)$ and place all leaves left of their parent.

We now show that every vertex ordering σ of $G_{FS}(l', r) + L$ contains $\overrightarrow{T_S}(l, r)$ or $\overrightarrow{T_S}(l', r')$. By Lemma 4.15 we can find $\overrightarrow{T_S}(l', r)$ among the vertices of $G_{FS}(l', r)$ in σ. Let v be the centre of this $\overrightarrow{T_S}(l', r)$. By pigeon hole principle, $(l - l')$ leaves adjacent to v must be either left or right of v. In the first case, v has $(l - l') + l' = l$ left and $r' > r$ right neighbours. In the last case, v has l' left and $r' + (l - l') > r'$ right neighbours. ∎

4.3 Split Graphs and Their Mirrors

If $G = (C \cup I, E)$ is a split graph with clique C and independent set I, we call a pattern $P = \left(C \cup I, E, \binom{V}{2} - E, \prec\right)$ *separated split graph* if $c \prec i$ for all $c \in C,\ i \in I$. We observe that if P is a separated split graph with clique and independent set of size at least 2, and G is a split graph, then G is P-free: By placing the clique of G right of the independent set of G, G avoids P. To show that a separated split graph P is not combinable with its mirror $\overleftarrow{P}$, we first define a family of split graphs in which we can find one of P or $\overleftarrow{P}$ under certain conditions.

Definition 4.17: *Let $n, l, k \in \mathbb{N},\ l \leq n$. The* satellite graph *$S(n, l, k)$ is the split graph with a clique C of size n and exactly k unique vertices adjacent to each l-subset of C. The l-subsets of C we call* bases, *their corresponding vertices* satellites.

By a group-ordered *satellite graph $\overrightarrow{S}(n, l, k)$, we denote any vertex ordering of $S(n, l, k)$ where the vertices of C are to the left and the satellites form k consecutive groups. Each group contains a satellite for each l-base. See Figure 4.6 for an example.*

We will make extensive use of satellite graphs in the proof of Theorem 4.18.

Theorem 4.18: *Let P be a separated split graph. Then P and its mirror are not perfectly combinable.*

To arrive at this result, we show that for any separated split graph P we can find a group-ordered satellite graph $\overrightarrow{S}(n,l,k)$ that contains in Lemma 4.19. In Corollary 4.22 we explore how to find $\overrightarrow{S}(n,l,k)$ in a larger satellite graph $\overrightarrow{S}(n',l',1)$. Then there is a still larger satellite graph $S(n'',l'',1)$ such that we can find $\overrightarrow{S}(n',l',1)$ or its mirror in any vertex ordering of $S(n'',l'',1)$, as we discuss in Lemma 4.20.

Lemma 4.19: *Let $P = \left(C \cup I, E, \binom{V}{2} - E, \prec\right)$ be a separated split graph with clique C of size n and independent set I of size m. Then P is contained in $\overrightarrow{S}(2n,n,m)$.*

Proof. We fix a group ordering $\overrightarrow{S}(2n,n,m)$ of $S(2n,n,m)$ and want to find a copy of P in it. To this end, we interpret the clique of $S(2n,n,m)$ as the disjoint union of C and a copy C' of C with a bijection $\phi : C \to C'$. Let $v_i \in I,\ i \in [m]$ be ordered according to $\prec$. We can choose the i-th group whose base is $N^{(P)}(v_i) \cup \phi\left(C - N^{(P)}(v_i)\right)$ for v_i. In this way we find copies of v_i in $\overrightarrow{S}(2n,n,m)$ in the order given by $\prec$. ∎

Lemma 4.20: *For all $n, l \in \mathbb{N}$ with $l \leq n$, any vertex ordering σ of $S(2n,2l,1)$ contains $\overrightarrow{S}(n,l,1)$ or its mirror.*

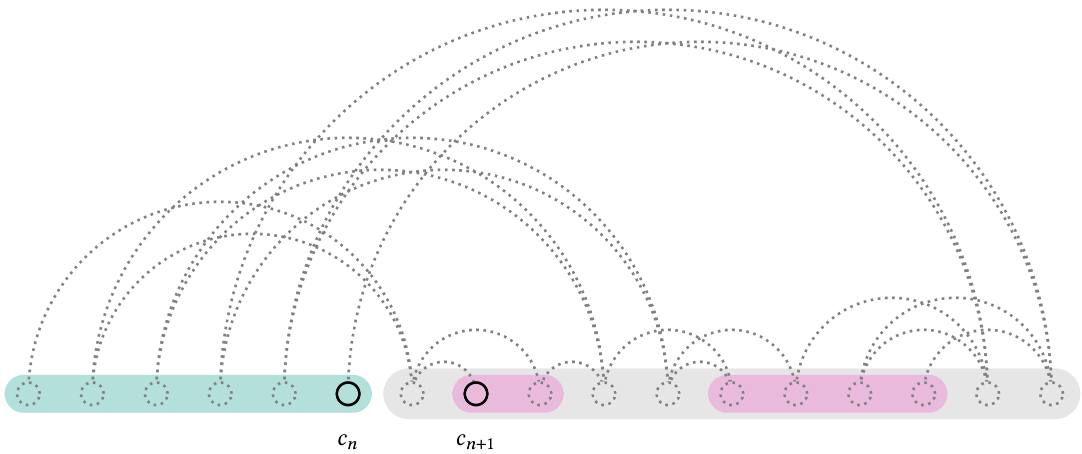
Proof. For $i \in [2n]$ let c_i denote the i-th vertex of the clique of $S(2n,2l,1)$ in σ. We restrict our attention to the satellites whose bases have l vertices from $\{c_1,\ldots,c_n\}$ and l vertices from $\{c_{n+1},\ldots,c_{2n}\}$. Consider all l-bases of $\{c_1,\ldots,c_n\}$. If they all have satellites in the restricted set right of c_n, we are done. Otherwise, there is an l-base B of $\{c_1,\ldots,c_n\}$ whose shared satellites with l-bases of $\{c_{n+1},\ldots,c_{2n}\}$ are all left of c_n (in particular left of c_{n+1}). Then the satellites of B form the satellite set for the mirror of $\overrightarrow{S}(n,l,1)$ with clique $\{c_{n+1},\ldots,c_{2n}\}$. Figure 4.7 depicts these two cases. ∎

Lemma 4.21: $\overrightarrow{S}(2n+1,2l,1)$ *contains* $\overrightarrow{S}(n,l,2)$.

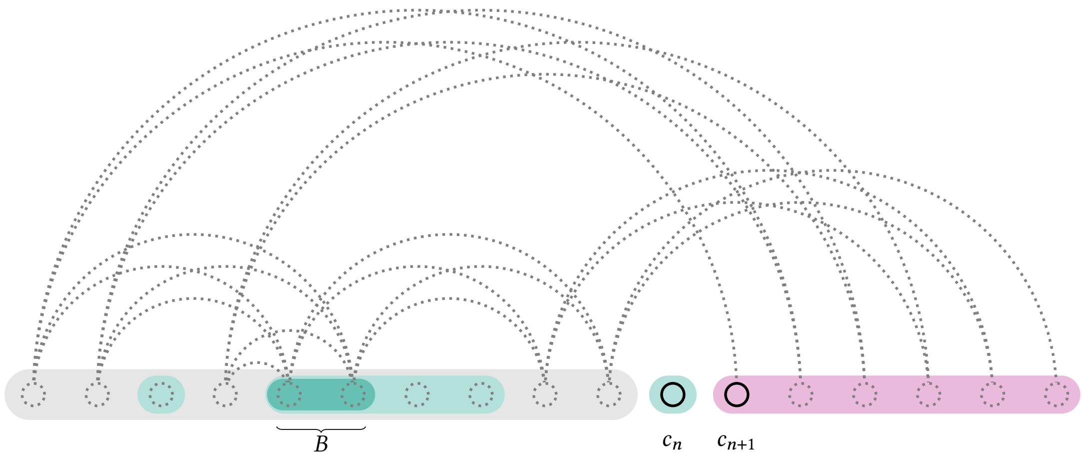
Proof. Let the clique of $\overrightarrow{S}(2n+1,2l,1)$ be partitioned into C_1, C_2 with $|C_1| = n+1, |C_2| = n$. In the following, we will only consider satellites whose bases have the form $B_1 \cup B_2$ where $B_i \in \binom{C_i}{l}$ for $i \in \{1,2\}$. We denote by r_B the index of the rightmost such satellite of $B \in \binom{C_1}{l}$, and choose B such that r_B is minimal. Note that every $B_2 \in \binom{C_2}{l}$ has a satellite at or to the left of r_B: By definition of r_B, the satellite adjacent to $B \cup B_2$ is left of or at r_B. Additionally, every $B_1 \in \binom{C_1}{l}$ has a satellite to the right of or at r_B by choice of B. Then there are two cases:

1. Every $B_2 \in \binom{C_2}{l}$ has a satellite to the right of r_B: Then the restriction of $\overrightarrow{S}(2n+1,2l,1)$ to the clique C_2 and to the satellites adjacent to l-bases of C_2 contains $\overrightarrow{S}(n,l,2)$.
2. There is some $B_2 \in \binom{C_2}{l}$ whose satellites are all left of or at r_B. But then each $B_1 \in \binom{C_1}{l}$ has a satellite at or to the right of r_B. These are the satellites adjacent to $B_2 \cup B_1$. Thus, by choosing $b \in B$, every $B_1 \in \binom{C_1 - b}{l}$ has a satellite to the left of r_B as well as to the right.

The choice of B and the two cases are visualized in Figure 4.8. ∎



(a) All l-bases of $\{c_1, \dots, c_n\}$ (green) share satellites right of c_n with some l-bases of $\{c_{n+1}, \dots, c_{2n}\}$ (purple).



(b) All l-bases of $\{c_{n+1}, \dots, c_{2n}\}$ (purple) share satellites left of c_{n+1} with some l-base B of $\{c_1, \dots, c_n\}$ (green).

Figure 4.7: The cases for finding $\overrightarrow{S}(n, l, 1)$ or its mirror in an arbitrary vertex ordering σ of $S(2n, 2l, 1)$ as described in the proof of Lemma 4.20.

Corollary 4.22: *Let $n, l, k \in \mathbb{N}$ with $l \leq n$. We define $N(k) := 2^{k-1}n + 2^{k-1} - 1$ and $L(k) := 2^{k-1}l$ Then $\overrightarrow{S}\big(N(k), L(k), 1\big)$ contains $\overrightarrow{S}(n, l, k)$.*

Proof. First observe that

$$N(k+1) = 2^k n + 2^k - 1 = 2\Big(2^{k-1}n + 2^{k-1} - 1\Big) + 1 = 2\big(N(k)\big) - 1 \quad \text{and}$$
$$L(k+1) = 2^k l = 2\Big(2^{k-1}l\Big) = 2L(k).$$

We prove this corollary by induction on k:

Base ($k = 1$): $N(1) = 2^{1-1}n + 2^{1-1} - 1 = n$, $L(1) = 2^{1-1}l = l$, so $\overrightarrow{S}\big(N(k), L(k), 1\big) = \overrightarrow{S}(n, l, k)$ in this case.

Step: By Lemma 4.21, $\overrightarrow{S}\big(N(k+1), L(k+1), 1\big)$ contains $\overrightarrow{S}\big(N(k), L(k), 2\big)$. By simply ignoring the second satellite group, we find $\overrightarrow{S}\big(N(k), L(k), 1\big)$. By induction hypothesis, this $\overrightarrow{S}\big(N(k), L(k), 1\big)$ contains $\overrightarrow{S}(n, l, k)$ with clique C'.

The $(k+1)$-st satellite group needed for the copy of $\overrightarrow{S}(n, l, k+1)$ we find in the second satellite group of $\overrightarrow{S}\big(N(k), L(k), 1\big)$: Let C be the clique of $\overrightarrow{S}\big(N(k), L(k), 2\big)$. We choose some $B \in \binom{C - C'}{2^{k-1}l - l}$. Then for every l-base $B' \in \binom{C'}{l}$, we get $|B \cup B'| = 2^{k-1}l$. By definition there is a satellite adjacent to $B \cup B'$ in the second satellite group of $\overrightarrow{S}(2^{k-1}n + 2^{k-1} - 1, 2^{k-1}l, 2)$. This satellite has no edges to any vertices of $C' - B'$, so it is a valid satellite for $\overrightarrow{S}(n, l, k+1)$. Since B' was chosen arbitrarily, we find the entire $(k+1)$-st satellite group in this way. Figure 4.9 depicts how the $(k+1)$-st satellite group is found.

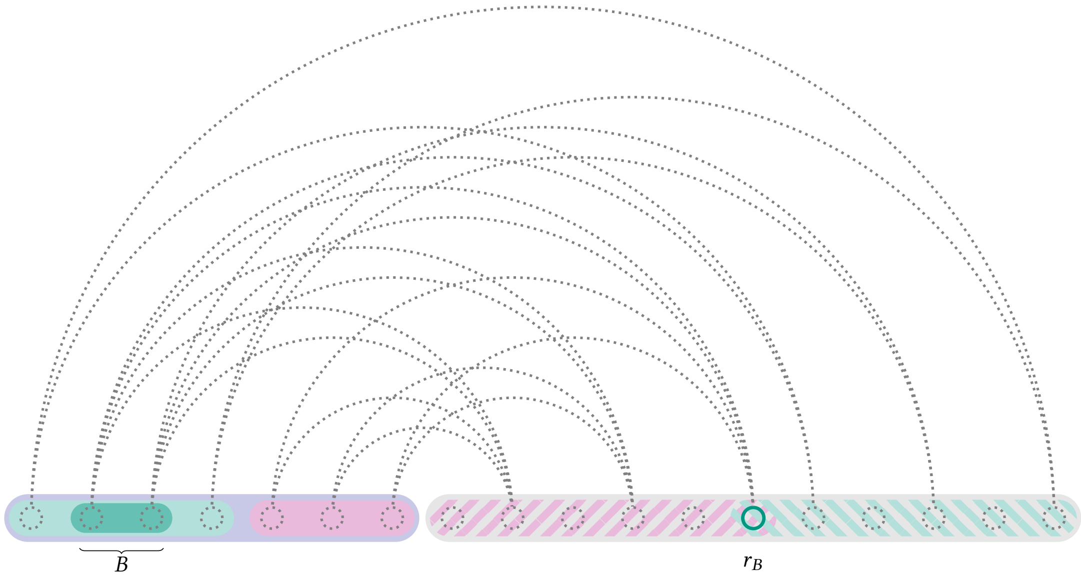
■

We now have assembled all the tools we need to prove Theorem 4.18.

Proof of Theorem 4.18. Let P be a separated split graph with clique of size n and independent set of size k. Then there exist $N, L \in \mathbb{N}$ such that $S(N, L, 1)$ has a P-free ordering (place clique to the right), but any ordering of $S(N, L, 1)$ contains P or its mirror: Choose $N = 2^{k+1}n + 2^{k+1} - 4$, $L = 2^k n$. By Lemma 4.20, $S(N, L, 1)$ contains $\overrightarrow{S}(2^k n + 2^k - 2, 2^{k-1}n, 1)$ or its mirror, without loss of generality not the mirror. By Corollary 4.22, we find $\overrightarrow{S}(2n, n, k)$. Finally, Lemma 4.19 guarantees us a copy of P. ■

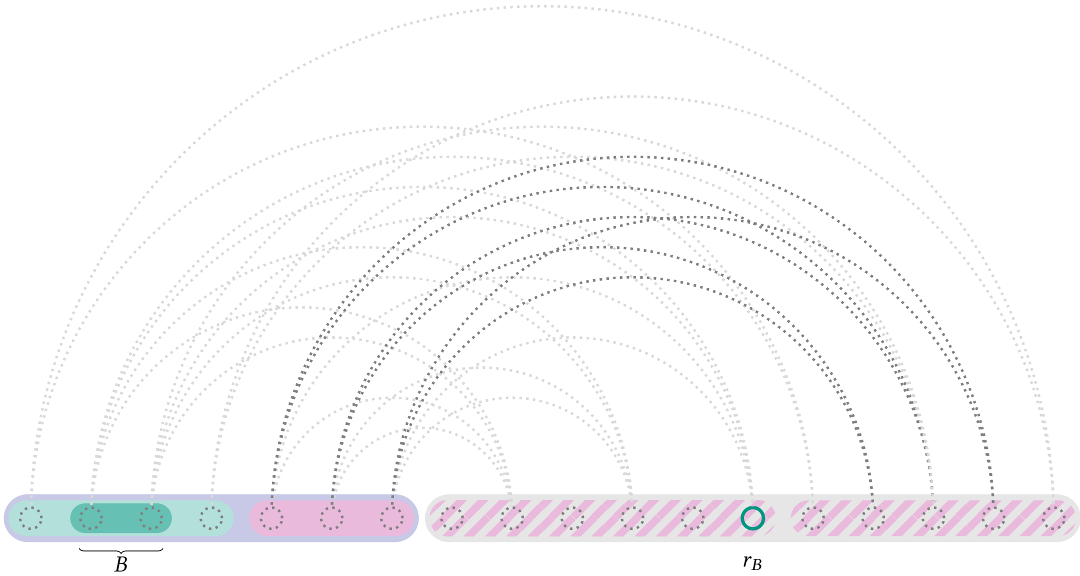
Corollary 4.23: *In fact, the proof of Theorem 4.18 can be easily adapted to show that if P_1, P_2 are separated split graphs, then P_1 and $\overleftarrow{P_2}$ are not perfectly combinable.*

Proof. Choose $n = \max\{n_1, n_2\}$, $k = \max\{k_1, k_2\}$. Proceed as in the proof above to determine $N, L \in \mathbb{N}$ that allow us to find $\overrightarrow{S}(2n, n, k)$ or its mirror in $S(N, L, 1)$. If we find $\overrightarrow{S}(2n, n, k)$, reduce the clique to n_1 vertices. If we find the mirror, reduce the clique to n_2 vertices. Then drop all satellites incident to now incomplete or removed bases. Finally, ignore the unnecessary satellite groups to find P_1 or $\overleftarrow{P_2}$ as in the proof of Lemma 4.19. ■

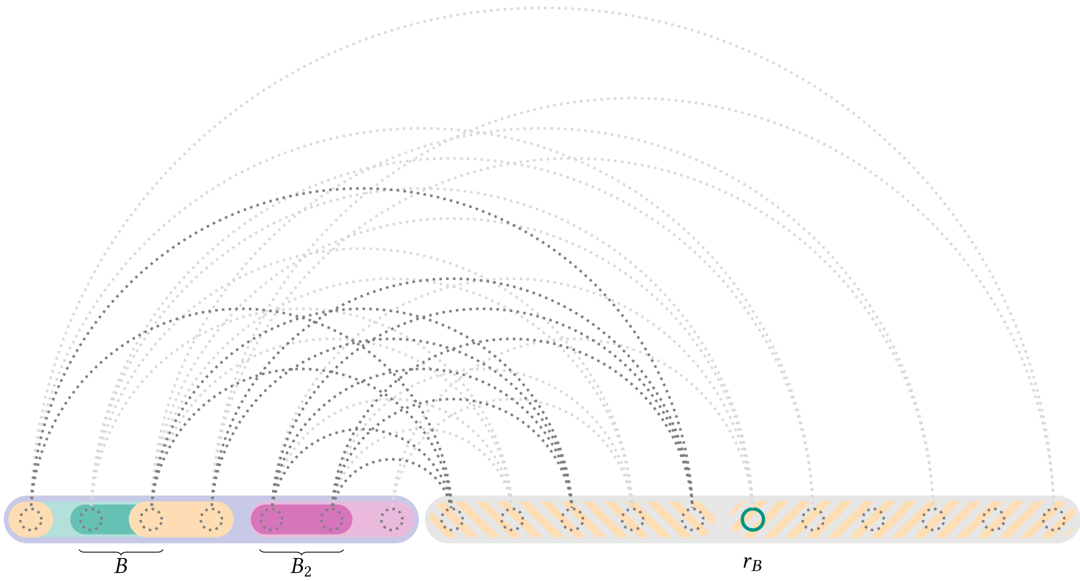


(a) The choice of B in the proof of Lemma 4.21. C_1 is marked in green and C_2 in purple. The striped areas mark where all bases of some set with the same colour have a satellite.

Figure 4.8: Finding $\overrightarrow{S}(n, l, 2)$ in $\overrightarrow{S}(2n+1, 2l, 1)$. Continued in Figure 4.8b.



(b) Case 1 in the proof of Lemma 4.21: All l-bases of C_2 have satellites to the right of r_B. C_1 is marked in green and C_2 in purple. The striped areas mark where all bases of some set with the same colour have a satellite.



(c) Case 2 in the proof of Lemma 4.21: There is an l-base B_2 of C_2 without satellites to the right of r_B. C_1 is marked in green and C_2 in purple. Orange marks the clique $C_1 - b$ of $S(n, l, 2)$. The striped areas mark where all bases of some set with the same colour have a satellite.

Figure 4.8: (Continued from Figure 4.8a.) Finding $\overrightarrow{S}(n, l, 2)$ in $\overrightarrow{S}(2n + 1, 2l, 1)$.

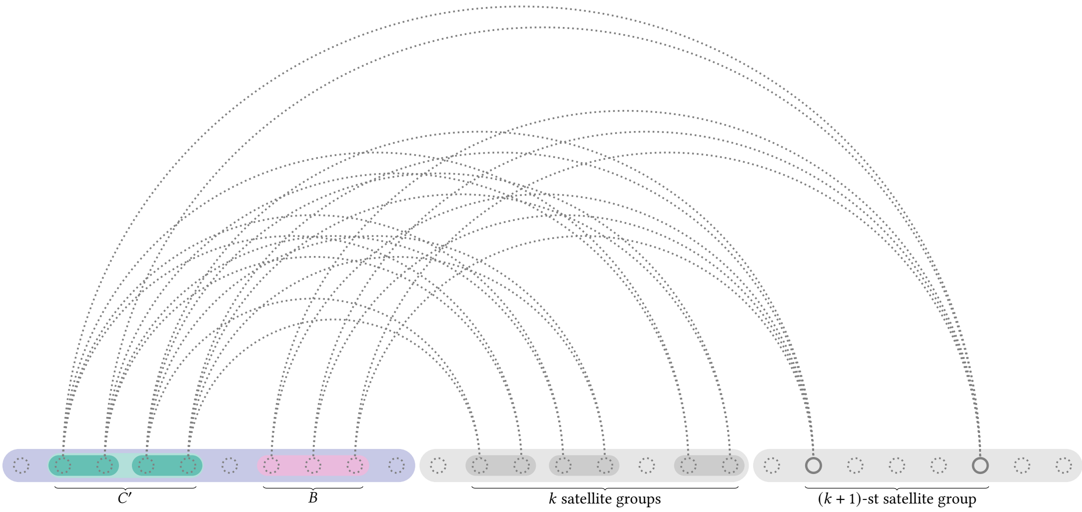


Figure 4.9: Recursively finding $\overrightarrow{S}(n, l, k+1)$ in $\overrightarrow{S}(2^k n + 2(k), 2^k l, 2)$.

5 Conclusion

In this thesis, we have introduced the notion of combinability of graph parameters as a generalization of the union-intersection-property of sets of patterns. We have explored the combinability of a number of well-known graph parameters in Chapter 3 and found some of them to be combinable, while others are not. We have additionally shown that there are infinitely many pairs of parameters that are not combinable (Lemma 3.27) and infinitely many pairs that are combinable (Observation 3.29).

When we turned to investigate the combinability, or union-intersection-property, of single patterns in Chapter 4, we found that it is precisely the patterns with the same underlying graph where it seemed very obvious that the union-intersection-property does not hold. From this observation, we conjectured that two distinct permutations of a pattern are never combinable, for which we gathered some evidence by proving it in several restricted cases.

5.1 Future Work

This thesis does not explore all pairwise combinations even of the graph parameters we have considered, so a natural extension of the results we have presented would be to complete the graph in Figure 3.1. Even for parameter pairs that we have some results for, we do not always know that they are tight, such as with the $(f, 1)$-combinability of queue number and chromatic number. But even if it turns out that the function f we gave in Proposition 3.32 is (asymptotically) tight, it might be possible to find an (asymptotically) smaller function f' so that queue number and chromatic number are (f', g)-combinable for some (preferably small) function g. There are, of course, more graph parameters and graph properties that can be characterized by forbidden patterns, such as transitive orientability, and which could be tested for combinability with each other and with the parameters we have already discussed.

In our definition, we have restricted the notion of combinability to graph parameters for which each value can be characterized by a single forbidden pattern. It would be interesting to see if this restriction could be softened to allow two forbidden patterns per parameter value, sets of forbidden patterns of bounded size, or even all graph parameters that can be characterized by forbidden patterns. In the most general case, the notion of combinability would then apply to all hereditary graph parameters [1]. Two patterns (so-called 'thick patterns') have recently been shown to suffice for bounding mixed number in graphs with bounded maximum degree [HMP24], which would make for interesting combinations with queue number and stack number. Unfortunately, Haun, Merker, and Pupyrev also show no finite family of forbidden patterns can suffice to characterize mixed number even on matchings, so at best, combinability with 'thick number' could be investigated. Another possible generalization is to consider k-tuples of parameters or patterns instead of pairs. Here, a natural question would be whether this extended combinability has something akin to the Helly property. The property we mean that if a set of parameters or patterns are pairwise combinable, the entire

[1] As Feuilloley and Habib discuss in their introduction to [FH21]

set is combinable, too. It is easy to see that a stronger property does not hold; namely there are three patterns and a graph that can avoid them pairwise, but not all three at once: Take all vertex orderings of $K_2 + K_1$ as the three patterns. Then any vertex ordering of $K_2 + K_1$ avoids two of the patterns but contains the third.

This thesis has focused on the properties of graph parameters that make the parameters combinable, but the other direction has not been explored at all: Combinability seems to give a measure of structural similarity of graph parameters, but what does this mean? Searching for applications of combinability might lead to interesting structural results.

In their survey on forbidden patterns on three vertices, Feuilloley and Habib found that the union-intersection-property seemed to hold for many cases where the patterns in question are the split of some third pattern. Since non-trivially combinable larger patterns seem to be quite rare, we suggest that this impression is derived from the uncharacteristically large proportion of permutation-invariant patterns (where two of E, N, U are empty) among those Feuilloley and Habib looked at. Nonetheless, splits might give an interesting weaker form of the Helly property mentioned above: For patterns $P_1 \& P_2 =: P$ and Q, if P_1 and P_2 are combinable with Q, is it true that P is combinable with Q? What if P_1 and P_2 are also combinable with each other?

Aside from this question, it would of course be natural to further investigate Conjecture 4.1 by generalizing the results of Proposition 4.7 and Theorem 4.18 or by considering further classes of patterns.

Bibliography

[BK79] Frank Bernhart and Paul C Kainen. “The book thickness of a graph”. In: *Journal of Combinatorial Theory, Series B* Volume 27 (1979), pp. 320–331.

[CCDG82] Phyllis Z Chinn, Jarmila Chvátalová, Alexander K Dewdney, and Norman E Gibbs. “The bandwidth problem for graphs and matrices—a survey”. In: *Journal of Graph Theory* Volume 6 (1982), pp. 223–254.

[CRST06] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. “The strong perfect graph theorem”. In: *Annals of mathematics* (2006), pp. 51–229.

[CS89] Fan RK Chung and Paul D Seymour. “Graphs with small bandwidth and cutwidth”. In: *Discrete Mathematics* Volume 75 (1989), pp. 113–119.

[Dam90] Peter Damaschke. “Forbidden ordered subgraphs”. In: *Topics in Combinatorics and Graph Theory: Essays in Honour of Gerhard Ringel* (1990), pp. 219–229.

[Duj+22] Vida Dujmović, David Eppstein, Robert Hickingbotham, Pat Morin, and David R Wood. “Stack-number is not bounded by queue-number”. In: *Combinatorica* Volume 42 (2022), pp. 151–164.

[DW04] Vida Dujmović and David R Wood. “On linear layouts of graphs”. In: *Discrete Mathematics & Theoretical Computer Science* Volume 6 (2004).

[DW05] Vida Dujmović and David R. Wood. “Stacks, Queues and Tracks: Layouts of Graph Subdivisions”. English. In: *Discrete Mathematics & Theoretical Computer Science* Volume Vol. 7 (Jan. 2005). ISSN: 1365-8050. DOI: *10.46298/dmtcs.346*.

[ES35] Paul Erdös and George Szekeres. “A combinatorial problem in geometry”. In: *Compositio mathematica* Volume 2 (1935), pp. 463–470.

[FH21] Laurent Feuilloley and Michel Habib. “Graph classes and forbidden patterns on three vertices”. In: *SIAM Journal on Discrete Mathematics* Volume 35 (2021), pp. 55–90.

[Gal68] Tibor Gallai. *On directed paths and circuits, Theory of Graphs (Proc. Colloq., Tihany, 1966)*. 1968.

[Has65] Maria Hasse. “Zur algebraischen begründung der graphentheorie. i”. In: *Mathematische Nachrichten* Volume 28 (1965), pp. 275–290.

[HLMP11] Pinar Heggernes, Daniel Lokshtanov, Rodica Mihai, and Charis Papadopoulos. “Cutwidth of split graphs and threshold graphs”. In: *SIAM Journal on Discrete Mathematics* Volume 25 (2011), pp. 1418–1437.

[HLR92a] Lenwood S Heath, Frank Thomson Leighton, and Arnold L Rosenberg. “Comparing queues and stacks as machines for laying out graphs”. In: *SIAM journal on discrete mathematics* Volume 5 (1992), pp. 398–412.

[HLR92b] Lenwood S. Heath, Frank Thomson Leighton, and Arnold L. Rosenberg. "Comparing Queues and Stacks As Machines for Laying Out Graphs". In: *SIAM Journal on Discrete Mathematics* Volume 5 (1992), pp. 398–412. eprint: *https://doi.org/10.1137/0405031.*

[HMP24] Deborah Haun, Laura Merker, and Sergey Pupyrev. "Forbidden Patterns in Mixed Linear Layouts". In: *arXiv preprint arXiv:2412.12786* (2024).

[KKPU24] Julia Katheder, Michael Kaufmann, Sergey Pupyrev, and Torsten Ueckerdt. "Transforming Stacks into Queues: Mixed and Separated Layouts of Graphs". In: *arXiv preprint arXiv:2409.17776* (2024).

[Mon86] Burkhard Monien. "The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete". In: *SIAM Journal on Algebraic Discrete Methods* Volume 7 (1986), pp. 505–512.

[Pem92] Sriram Venkata Pemmaraju. *Exploring the powers of stacks and queues via graph layouts.* Virginia Polytechnic Institute and State University, 1992.

[Ros70] Donald J Rose. "Triangulated graphs and the elimination process". In: *Journal of Mathematical Analysis and Applications* Volume 32 (1970), pp. 597–609.

[Roy67] Bernard Roy. "Nombre chromatique et plus longs chemins d'un graphe". In: *Revue française d'informatique et de recherche opérationnelle* Volume 1 (1967), pp. 129–132.

[Skr82] Dale J Skrien. "A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs". In: *Journal of graph Theory* Volume 6 (1982), pp. 309–316.

[Str23] Adam Straka. "Stack number and queue number of graphs". In: *arXiv preprint arXiv:2305.09700* (2023).

[Vit62] LM Vitaver. "Determination of minimal coloring of vertices of a graph by means of Boolean powers of the incidence matrix". In: *Doklady Akademii Nauk*. Vol. 147. Russian Academy of Sciences. 1962, pp. 758–759.

[Wie16] Veit Wiechert. "On the queue-number of graphs with bounded tree-width". In: *arXiv preprint arXiv:1608.06091* (2016).

[Woo08] David R Wood. "Bounded-degree graphs have arbitrarily large queue-number". In: *Discrete Mathematics & Theoretical Computer Science* Volume 10 (2008).