Twin-Width, logical and combinatorial characterisations

Colin Geniet

Thèse

en vue de l'obtention du grade de Docteur, délivré par

l'École Normale Supérieure de Lyon

École Doctorale N°512
École Doctorale en Informatique et Mathématiques de Lyon

Discipline : Informatique

Soutenue publiquement le 5 juillet 2024 par

Colin Geniet

Twin-Width
Caractérisations Logiques et Combinatoires

Devant le jury composé de :

Victor Chepoi	Professeur des universités, Aix Marseille Université	Rapporteur
Martin Grohe	Professeur, Université RWTH Aachen	Rapporteur
Frédérique Bassino	Professeure des universités, Université Paris 13	Examinatrice
Mamadou Moustapha Kanté	Professeur des universités, Université Clermont Auvergne	Examinateur
Alantha Newman	Chargée de recherche, Université Grenoble Alpes	Examinatrice
Ioan Todinca	Professeur des universités, Université d'Orléans	Examinateur
Stéphan Thomassé	Professeur des universités, ENS de Lyon	Directeur de thèse

À celle qui m'apprit les mathématiques, et à les aimer

Acknowledgments

My three years in Lyon come to a close as I finish writing this manuscript. During these three years, I met several persons who helped and accompanied me, scientifically and personally, and whom I would like to thank.

Early in our first discussion, Stéphan Thomassé described to me the 'interesting notion'—quite the understatement!—he was working on. I had little idea then of the journey this would become, as Stéphan became my advisor, and this notion my thesis. Five years later, I can appreciate the chance I had to benefit not only from his invaluable knowledge and insight, but also from the tremendous time he spent working with me, and indeed all of his students, in all circumstances. For all this time, advice, questions, and ideas, I owe you my greatest gratitude.

Reviewing a manuscript is no small task, and for this I am extremely grateful to Victor Chepoi and Martin Grohe, whose comments were invaluable in bringing this manuscript to its current form. I also want to thank Frédérique Bassino, Mamadou Kanté, Alantha Newman, and Ioan Todinca, for accepting to join them as part of my jury.

Coming to the LIP at ENS Lyon throughout these three years has always been a great experience and excellent environment; for this I want to give my greatest thanks to our truly amazing administrative staff, and particularly to Laure and Marie for their infinite patience with our team. In these three years, I shared an office with Pegah, Hugues, Julien, and Malory: to each of you, thank you for this great time, your good humour, and your wildly varied characters which kept this office joyful! And as the world does not stop at the office door, this 'thank you' extends to all my colleagues in the MC2 team, in LIP, and also in LIRIS, with a special mention for Laurent who works so hard to keep the bond between our two teams.

I had the great pleasure to work and discuss on twin-width with many people, without whom this work would never have been what it is. I want to thank all of you, and particularly my coauthors Édouard, Eun Jung, Rémi, Romain B., and Romain T. And since thinking about nothing but twin-width would have made for a sad PhD, I shall not forget those of you with whom I worked on other questions: Alexandra, Claire, Louis, Marthe, and Nicolas. Finally, I want to give special thanks to some persons who paved my way to this PhD: Stéphane who gave me a taste of graph theory—and a good one!—during my very first internship; Mikołaj who welcomed me for a great year in Warsaw and taught me so much about logic; and Michał and Szymon whose unfaltering interest and excellent ideas have been an amazing motivation.

Science alone is not enough to be a researcher, much less a person, and thus I also want to express my deepest gratitude to those who supported me personally and emotionally during these years: my family, and particularly my parents, who guided me in the path to mathematics, and whom I will never be able to thank enough for their unending support. But also the friends I met in Lyon through music, who helped me keep joy in sad days and motivation in boring ones with Mozart, Shostakovich, and Tchaikovsky: Basile, Julien, Nicolas, and my very dear friend, Sasha.

Résumé

Un graphe est composé d'un ensemble de sommets reliés par des arêtes. Les graphes sont des structures versatiles, couramment utilisées pour représenter des réseaux de transports, de communication, ou d'individus. Cette versatilité a un prix : beaucoup de problèmes naturels — e.g. trouver un nombre maximum de sommets non reliés — sont difficiles. Plutôt qu'essayer de les résoudre en toute généralité, on peut se restreindre à des graphes satisfaisant certaines conditions pouvant simplifier la tâche, comme par exemple les graphes planaires, i.e. les graphes pouvant être dessinés sur le plan sans croisement.

C'est dans cet esprit que Bonnet, Kim, Thomassé et Watrigant ont introduit en 2020 la *twin-width*, (lit. *largeur de jumeaux*) inspirée par une notion de *largeur de permutations* de Guillemot et Marx. Elle est définie par des *suites de contractions*, au cours desquelles on fusionne des paires de sommets jusqu'à avoir réduit le graphe à un seul sommet, tout en mesurant une notion d'erreurs.

De nombreuses classes de graphes sont de twin-width bornée : par exemple les graphes planaires, ou plus généralement ceux évitant un mineur fixé, ainsi que les graphes de tree-width ou clique-width bornée. Dans une telle classe $\mathcal{C}$, les suites de contractions ont des applications remarquables. On peut par exemple les utiliser pour obtenir un algorithme efficace (au sens de la *complexité paramétrée*) pour tout problème exprimé en logique du premier ordre ; des résultats de coloriage montrant que $\mathcal{C}$ est *χ-bornée* ; et une borne sur le nombre de graphes d'une taille donnée dans $\mathcal{C}$, qui est une *petite classe*. Mentionnons cependant une limite de la notion : trouver rapidement de bonnes suites de contractions est un problème ouvert en général, bien que des algorithmes soient connus pour tous les exemples précédents de classes de twin-width bornée.

Après une présentation détaillée de ces notions et résultats connus ainsi que des techniques impliquées, cette thèse s'intéresse à la question suivante : les propriétés précédentes (algorithme pour les propriétés du premier ordre, coloriage, petitesse) sont elles exclusivement vérifiées par les classes de twin-width bornée ? En général ce n'est pas le cas : pour les deux premières propriétés, les graphes de degré borné sont un contre-exemple, et nous construisons une petite classe de twin-width non bornée, en passant par les groupes et les graphes de Cayley.

Néanmoins, on peut re-poser cette question pour d'autres structures que les graphes. En effet la définition de twin-width s'adapte aisément à toute structure composée de relations binaires. Ainsi, dans les *graphes ordonnés* (un graphe muni d'un ordre total sur les sommets), Bonnet, Giocanti, Ossona de Mendez, Simon, Thomassé et Toruńczyk ont montré que twin-width bornée, petitesse, et résolution efficace de problèmes du premier ordre sont des conditions équivalentes. De plus, la twin-width peut être rapidement approximée dans ces structures. Nous généralisons ces résultats aux *tournois* (un ensemble de sommets avec pour chaque paire un choix de direction, ou du "gagnant").

Les *permutations* peuvent être vues comme un cas particulier de graphes ordonnés. Si les graphes ordonnés en général se comportent bien vis à vis de la twin-width, les permutations sont tout particulièrement intéressantes : dans leurs travaux fondateurs, Guillemot et Marx montrent que pour les permutations, avoir twin-width bornée est équivalent à éviter un motif. Après avoir reformulé quelques résultats classiques de combinatoire des permutations sous l'angle de la twin-width, nous présenterons un résultat de décomposition : les

permutations de twin-width bornée se factorisent en un nombre borné de permutations dites *séparables*, qui sont les permutations de twin-width 0.

Nous terminons cette étude avec des structures pour lesquelles notre compréhension de la twin-width est bien moins complète : les graphes éparses et les groupes. Deux définitions équivalentes de la twin-width des groupes sont présentées, l'une passant par les *graphes de Cayley*, et l'autre se ramenant aux permutations par les actions de groupes. Ces deux définitions permettent de montrer que la twin-width des groupes est préservée par un grand nombre d'opérations et constructions classiques. En revanche, montrer ne serait ce qu'il existe des groupes de twin-width infinie est difficile. Nous le prouvons grâce à un théorème d'Osajda, permettant de plonger une suite de graphes de degré borné dans un groupe. Les groupes, comme les graphes de degré borné, se trouvent ainsi dans une situation étonnante : on montre qu'il en existe de twin-width infinie, sans savoir les expliciter. L'existence d'un groupe de twin-width infinie nous permet de construire une petite classe avec twin-width non bornée.

Summary

A graph consists of a set of vertices connected by edges. Graphs are versatile structures, commonly used to represent networks of transportation, communication, or persons. This versatility has a cost: numerous natural algorithmic problems on graphs are hard to solve, such as finding a largest set of pairwise non-adjacent vertices. Rather than trying to solve such problems in full generality, one may constrain the input graphs to simplify the problem, for instance by considering only planar graphs, i.e. the ones drawn in the plane with no crossing edges.

This work studies such a constraint: *twin-width*, introduced in 2020 by Bonnet, Kim, Thomassé, and Watrigant, and inspired by a *width of permutations* defined by Guillemot and Marx. Twin-width is defined through *contraction sequences*, during which one identifies pairs of vertices until the graph is reduced to a single vertex, while measuring some notion of errors.

Many well-known graph classes have bounded twin-width: for instance planar graphs, or more generally graphs avoiding a fixed minor, as well as graphs with bounded tree-width or clique-width. In such a class $\mathcal{C}$, contraction sequences have remarkable applications. Notably, they can be used to obtain an efficient algorithm (in the sense of *parameterized complexity*) for any problem expressed through first-order logic; graph colouring results which show that $\mathcal{C}$ is χ-*bounded*; and an upper bound on the number of graphs in $\mathcal{C}$ with a given size: $\mathcal{C}$ is called *small*. Let us however mention a limitation: efficiently finding good contraction sequences is an open problem in general, although it can be done for all the aforementioned examples of graphs with bounded twin-width.

After an in-depth introduction of these notions, known results, and of the techniques involved, this thesis considers the following question: are the previous properties (algorithm for first-order properties, colouring, smallness) exclusive to classes with bounded twin-width? In general, this is not the case. Indeed graphs with bounded degree are counterexamples for the first two properties, and we construct a small class with unbounded twin-width, using groups and Cayley graphs.

Nonetheless, one may ask the same question for structures other than graphs. Indeed twin-width easily extends to any structure consisting of binary relations. Thus, in *ordered graphs* (graphs with a total ordering of the vertices), Bonnet, Giocanti, Ossona de Mendez, Simon, Thomassé, and Toruńczyk proved that bounded twin-width, smallness, and efficient algorithms for first-order properties are all equivalent. Further, twin-width can be efficiently approximated in these structures. We extend these results to *tournaments* (a set of vertices with, for each pair, the choice of a direction or 'winner').

Permutations can be seen as a special case of ordered graphs. While ordered graphs in general are well-behaved for twin-width, permutations are particularly interesting: in their founding work, Guillemot and Marx show that for permutations, bounded twin-width is equivalent to avoiding a pattern. After revisiting some classical results on permutations with the point of view of twin-width, we present a decomposition result: permutations with bounded twin-width factorise into bounded products of *separable* permutations, which are the permutations with twin-width 0.

We conclude this work with structures where our understanding of twin-width is far more limited: sparse graphs and groups. Two equivalent definitions

of twin-width in groups are introduced, the first using *Cayley graphs*, and the second using permutations and group actions. Using these two definitions, we show that twin-width of groups is stable under a number of classical operations and constructions. However, merely showing that groups with infinite twin-width exist is difficult. We prove it using a theorem of Osjada, allowing to embed sequences of bounded degree graphs into groups. Groups and bounded degree graphs alike exhibit a peculiar situation: while it can be shown that there exist some with infinite twin-width, no explicit construction is known. The existence of a group with infinite twin-width allows us to construct a small class with unbounded twin-width.

CONTENTS

LIST OF FIGURES

HOW TO READ THIS THESIS

This work is divided in two parts and an introduction. The introduction presents the context motivating twin-width: graph algorithms, parameterized complexity, and the underlying question of what being simple or complex means for graphs.

The first part, Chapters 2 to 4, is an in-depth introduction to twin-width. It presents key properties of graphs with small twin-width related algorithms, logic, colourings, encodings, and enumeration, as well as several examples of classes of graphs with and without bounded twin-width. It is meant as a first introduction to twin-width, and assume only general knowledge of graph theory.

The second part, Chapters 5 to 7, builds upon the first half and present some more involved results about twin-width, many of which involve structures others than graphs. These three chapters are mostly, but not entirely independent of each other.

The glossary is meant as a reminder for the definitions used in this work, with pointers to the full definition and context in the main text.

Chapter 1

INTRODUCTION

A *graph* G consists of a set $V(G)$ of *vertices* or abstract points, together with a set $E(G)$ of *edges*, i.e. pairs of vertices which can be understood as connections between the vertices. Graphs are simple but extremely versatile structures: they are commonly used to represent networks of connections (e.g. railways, communication links, social connections, or bridges in Königsberg), or conflicts between objects represented by vertices (e.g. interferences between radio stations, simultaneously live variables in a register allocation problem). One can even argue that *anything* can be represented by graphs: the computer scientist may notice that any relational database can be drawn as a graph whose vertices are objects and tuples with edges representing membership; the mathematician meanwhile will notice that models of set theory are infinite directed graphs, vertices being sets and edges again representing membership.

Unfortunately, this versatility comes at the cost: one should expect objects which can represent anything to be complex. This complexity takes various aspects. Algorithmically, many natural problems on graphs cannot be solved quickly; the ones which can, e.g. finding a shortest path, are the exception rather than the rule. Mathematically, one can hardly hope to find any interesting structure or description in arbitrary graphs. Let us illustrate this situation through an algorithmic problem which is remarkably easy to state, yet hard to solve: finding independent sets.

1.1 Graph algorithms and Independent Sets

In a graph G, a subset of vertices $S \subset V(G)$ is called *independent* or *stable* if there is no edge $xy \in E(G)$ with $x, y \in S$ (see Figure 1.1). The definition is particularly natural if the edges of G are understood as conflicts between vertices: an independent set is a conflict-free subset. The *independent set* problem asks to find an independent set with as many vertices as possible in a given graph G.

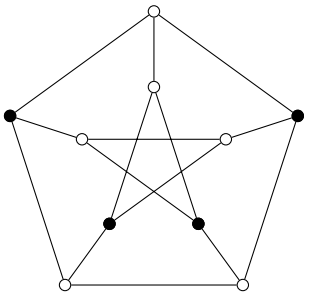


Figure 1.1: Example of independent set (filled vertices) in the Petersen graph.

Polynomial algorithms. What is meant by an *efficient* algorithm? The simplest and most common requirement is a *polynomial time* algorithm, i.e. one which runs in time bounded by some polynomial $O(n^c)$ of the input size n, which for graphs means the number of vertices $n = |V(G)|$. The *independent set* problem is NP-complete: it is in fact one of the oldest known NP-complete problems, featured in Karp's list in 1972 [66]. Thus a polynomial time algorithm for independent set would imply that P = NP, considered an unlikely answer to the famous P vs NP problem. This is a strong indication that independent set can likely *not* be solved in polynomial time.

When interested in algorithms, knowing that a problem can likely not be solved in polynomial time is hardly a satisfying answer: it is natural to wonder if this NP-hard problem could nonetheless be solved efficiently, for relaxed notions of 'solving' and 'efficiently'.

Approximation algorithms. A very natural idea is to look not for the best solution, but a good enough one: a solution close to the optimal. The *vertex cover* problem is a good example. A set $C \subset V(G)$ of vertices is called a vertex cover if any edge in $E(G)$ has an endpoint inside C. Remark that C is a vertex cover if and only if its complement $V(G) \setminus C$ is an independent set. It follows that finding a smallest vertex cover is as hard as finding a largest independent set: both problems are NP-complete. However, the following simple algorithm will quickly find a vertex cover at most twice as large as the smallest one: greedily pick a maximal matching M (a set of edges with no shared endpoints), and take all endpoints of edges in M as a vertex cover. This is called a *2-approximation* algorithm, running in polynomial time.

Despite their link, remark that finding a 2-approximation for vertex cover (one at most twice as large as the minimum) is a very different problem from finding a 2-approximation for independent set (one at least half as large as the maximum). And indeed, while the former can be done in polynomial time, the latter is hard: unless P = NP, it is impossible to approximate independent set up to any constant factor, or in fact up to a factor which is less than a polynomial of the input size [87]. Thus for independent set, finding an approximate solution is just as hard as finding the optimal one.

Parameterized complexity. An approximation algorithm accepts non optimal solutions. What about trade-offs on the complexity instead? Parameterized complexity, pioneered by Downey and Fellows [38], allows the complexity to depend on a *parameter*, for instance the size of the solution. Say we are not looking for the largest independent set, but for one of a given size k. The naive algorithm enumerating all subsets of size k runs in time roughly n^k. Even with small values of k, say 10, this quickly becomes prohibitively long when n grows. *Fixed parameterized tractable* algorithms (FPT), the key notion of parameterized complexity, require to decouple the dependencies in k and n: an FPT algorithm is one running in time $f(k) \cdot n^c$ for some constant c, and an arbitrary computable function f—often the exponential 2^k. Thus for fixed k this is a polynomial algorithm. When k grows, the multiplicative constant in this polynomial grows possibly very quickly, but the degree does not.

Vertex cover is again a good example. Suppose that we are looking for a vertex cover C of size k in G. Pick any vertex v; there are two choices: either

have v in C, or have all of its neighbours in C—as the edges incident to v would be uncovered if we do neither. Then, for each of the two choices, repeat the process. Note that at each choice, a vertex at least is added to C, thus if no solution is found after k choices, one should simply give up: C is already too large. Hence this process explores a tree of choices, with two possibilities at each point, and stopping after k choices. There are 2^k branches in this tree, and the resulting algorithm runs in time $2^k \cdot n$. The vertex cover problem is therefore FPT.

For independent set however, the results are once again negative. Again, despite the link between the two problems, an FPT algorithm for vertex cover (which should be particularly fast when there is a small vertex cover, hence a very large independent set) is very different from an FPT algorithm for independent set (which needs to be fast when looking for small independent sets). Independent set is known to be complete for the parameterized complexity class W[1]. One may think of FPT vs W[1] as the parameterized complexity analogue of P vs NP: independent set being W[1]-complete is a strong indication that it should not admit an FPT algorithm.

Restricted graphs. Independent set is thus a truly difficult problem for which none of the previous approaches yields any kind of efficient algorithm. It is unreasonable to try to solve it for all graphs, which naturally leads to the next question: are there specific graphs for which independent set can be solved quickly?

An extremely simple example is trees, i.e. connected graphs with no cycles. In trees, a simple greedy algorithm finds a largest independent set S: pick a leaf (vertex with only one neighbour), add it to S, remove its neighbour from the graph, and repeat.

A more complex picture is found in *planar graphs*, that is graphs which can be drawn on the plane without crossing edges: the independent set problem is NP-complete [50], but it can be approximated. Euler's formula implies that a planar graph on n vertices contains an independent set S with $\frac{n}{6}$ vertices, obtained by repetitively picking a vertex with no more than 5 neighbours and deleting its neighbours (compare with the algorithm in trees). In a trivial sense, this is a 6-approximation algorithm for independent set, and an FPT algorithm can be obtained for similar reasons. More interestingly, independent set in planar graphs admits polynomial time c-approximation algorithms for any constant $c > 1$ [7]. This is called a *polynomial time approximation scheme (PTAS)*.

In both trees and planar graphs there always exist independent sets of size linear in the number of vertices. Let us give an example in which this is not the case. A *cograph* is a graph constructed starting from individual vertices by two operations: the *disjoint* union of two graphs, and the *complete* union (i.e. taking the disjoint union of graphs and adding all edges between them). For example, edgeless graphs and cliques (graphs with all possible edges) are both cographs. For both operations, one can easily find a maximum independent set in the resulting graph given solutions in the two parts: suppose we have maximum independent sets S_i in G_i, $i = 1, 2$. Then $S_1 \cup S_2$ is a maximum independent set in the disjoint union of G_1 and G_2, while in the complete union, whichever of S_1 or S_2 is largest will be a maximum independent set. Given a

cograph G, it is also possible to reconstruct the sequence of operations which built G. Applying the above to this sequence of operations yields a maximum independent set in G in polynomial time.

1.2 Classes of graphs

The previous examples demonstrate how a problem which seems hopelessly difficult when considering all graphs, may admit interesting algorithms when restricting the graphs considered. Oftentimes, the question one should ask is not *is it possible to solve this efficiently?* but rather *for which graphs can this problem be solved efficiently?* In a sense, we want to classify graphs as either simple or complex.

This phrasing is slightly misleading: it does not make sense to ask if e.g. the independent set problem is hard for one specific graph G. Indeed nobody would care for an algorithm designed only for G. Algorithms are designed for *classes* of graphs, and for instance the independent set problem is simple in the classes of trees, of cographs, of planar graphs, while it is hard in the class of all graphs. The question we really mean to ask is *for which* classes *of graphs can this problem be solved efficiently?*.

Here, a class of graphs simply means a collection of graphs closed under isomorphism. Without additional restrictions, a class of graphs can be extremely complex to describe, as one may arbitrarily pick which graphs (up to isomorphism) it contains. It is unreasonable to attempt to characterise all classes of graphs in which e.g. the independent set problem is simple. Instead, one will usually restrict their attention to classes of graphs with a coherence condition: if G is in the class $\mathcal{C}$, then any graph H contained in G should also belong to $\mathcal{C}$. For graphs, *contained* can have many different meanings. The most common ones are the following.

subgraphs A subgraph in G is a graph obtained by removing any subset of edges and vertices. Naturally, if a vertex v is removed, so should all edges incident to v.

A class $\mathcal{C}$ of graphs closed under subgraphs—meaning that when $G \in \mathcal{C}$, any subgraph of G is also in $\mathcal{C}$—is called *monotone*. For example, the classes of forests (disjoint unions of trees) and of planar graphs are monotone, but that of cographs is not: it contains all cliques K_n (n vertices all pairwise connected), and any graph on n vertices is a subgraph of K_n.

induced subgraphs An induced subgraph of G is obtained by deleting vertices from G, and removing only the edges incident to the deleted vertices. That is, if X is the set of preserved vertices, then all edges of G with endpoints in X are kept. This subgraph is said to be *induced by* X, and is denoted by $G[X]$.

A class closed under induced subgraphs is called *hereditary*. Since induced subgraphs are a more restrictive notion than subgraphs, a class which is monotone is also hereditary. Forests and planar graphs are hereditary because monotone, and cographs are hereditary but not monotone.

minors A minor of G is obtained by deleting vertices and edges, and *contracting* edges. Contracting an edge uv means merging the vertices u and v

while preserving their edges: the vertex resulting from this merge will be adjacent to some x if either u or v was adjacent to x.

Any minor closed class is also monotone. Forests and planar graphs are examples of minor closed classes. The class of *subcubic* graphs, i.e. in which the degree (number of neighbours) of each vertex is at most 3, is a monotone class which is not minor closed.

We have thus three notions of stability under some containment relation for graph classes: hereditary, monotone, and minor closed, from the weakest to the strongest requirement.

Minor closed classes. Minor closed classes are well understood thanks to the colossal work of Robertson and Seymour in the *graph minors* series. A minor closed class $\mathcal{C}$ which does not contain all graphs must *avoid* some graph H as minor, i.e. H is not a minor of any $G \in \mathcal{C}$: indeed this holds for any $H \in \mathcal{C}$. Robertson and Seymour describe a structure in graphs avoiding any fixed minor: they decompose into graphs which up to small errors can be embedded in surfaces with fixed genus [84]. Thus in a very broad sense, classes which avoid a minor behave like planar graphs, and enjoy many of their properties. In particular, the independent set problem, while NP-complete, admits a PTAS and an FPT algorithm, and this generalises to a wide range of other algorithmic problems. Thus any minor closed class other than the class of all graphs is simple in a strong sense.

Monotone classes. The situation is more complex in monotone classes. Regarding independent sets, Ramsey's theorem implies that in any graph G avoiding H as subgraph, there is an independent set of size $O(n^{\frac{1}{k}})$, with n, k the number of vertices of G, H respectively. Thus for a fixed H, independent sets of at least polynomial size can be found in graphs avoiding H as subgraph. Nonetheless, a simple reduction shows that in for instance triangle-free graphs (i.e. without K_3 as subgraph), independent set is NP-hard to approximate within any constant. Avoiding one fixed H as subgraph is not in general sufficient to ensure that a class is simple.

A major advance in the understanding of monotone classes is the theory of *sparsity* of Nešetřil and Ossona de Mendez [75]. *Sparse* is a broad term referring to graphs with few edges, as opposed to *dense* graphs. It can have many different meanings. Having bounded degree and avoiding a minor are two incomparable and strong notions of sparsity. Generalising both, Nešetřil and Ossona de Mendez defined *nowhere dense* classes. Grohe, Kreutzer, and Siebertz proved that nowhere dense classes have FPT algorithm to solve any problem described using *first-order logic*, i.e. with a logical formula quantifying on vertices of the input graph. First-order logic allows to describe a broad class of problems, including independent set. Conversely, a monotone class $\mathcal{C}$ which is not nowhere dense can in a sense encode all graphs through some first-order formula [1], a property called *first-order independence*. This implies that first-order logic problems in $\mathcal{C}$ cannot be solved by an FPT algorithm [69] (assuming that $\mathsf{FPT} \neq W[1]$).

Thus a monotone class $\mathcal{C}$ either is nowhere dense, which gives it sufficient structure to solve first-order problems, or can encode all graphs, which implies

that first-order problems are hard in $\mathcal{C}$. This is precisely the kind of dichotomy which we are interested in: for a fixed class of problems (first-order definable), a structural notion (nowhere dense) characterises the classes of graphs in which these problem can be efficiently solved.

Hereditary classes. Being far more general than monotone classes, the current understanding of hereditary classes is limited. A question which has raised significant work is the following. Given a graph H, one can consider the class of H-free graphs (i.e. of graphs which do not contain H as induced subgraph), which is by construction hereditary. For which H is there a polynomial algorithm for independent set in H-free graphs? Using that independent set is NP-hard in graphs with maximum degree 3 [50], one can show that H must be restricted to very simple graphs: paths, subdivided claws (i.e. three paths sharing an endpoint), and disjoint unions thereof. It is conjectured that when H is such a graph, there is a polynomial algorithm for independent set in H-free graphs. A quasi-polynomial approximation scheme is known [31], while polynomial algorithms are only known for special cases of H [5, 71, 60, 24].

The former conjecture however only applies to classes defined by avoiding a single induced subgraph H, and many interesting hereditary classes cannot be described with a single, or even a finite list of induced subgraphs to avoid. For instance *chordal graphs* are the graphs which do not contain a cycle C_k on k vertices as induced subgraph for any $k \geqslant 4$. A maximum independent set can be found in a chordal graph by repetitively picking a vertex whose neighbourhood is a clique (which always exists in a chordal graph) and deleting its neighbours. This is a generalisation of the algorithm on trees.

There is no conjectured characterisation of the hereditary classes in which independent set has a polynomial algorithm. A bold conjecture was however proposed for problems defined in first-order logic. Recall that in a monotone class $\mathcal{C}$, there is an FPT algorithm for first-order problems if and only if $\mathcal{C}$ cannot encode all graphs through a first-order formula. It is conjectured that the same also holds for hereditary classes [48]. For monotone classes, this equivalence was established through a third, more structural condition: being nowhere dense. A generalisation of nowhere dense would most likely be necessary to extend the result to hereditary classes, and even with such a notion, finding FPT algorithms for first-order problems may prove very difficult. We will see a few examples of classes with such algorithms—in addition to nowhere dense classes—in the remainder of this introduction and this work.

1.3 Complexity measures

The previous section made a case for trying to classify *classes* of graphs as simple or complex, and not the graphs themselves: a single finite graph cannot by itself be considered complex. On the other hand, it is reasonable to define a gradual measure of the complexity of individual graphs, as a function from graphs to numbers: the smaller the number, the simpler the graph. Such functions, called *graph complexity measure*, or more colloquially *width functions*, have become central in graph theory and parameterized algorithms.

Tree-width. Undoubtably the most famous graph complexity measure is *tree-width*. The notion was proposed independently by different authors around 1975, but its development came as part of the work on graph minors of Roberston and Seymour [82]. The tree-width $\mathrm{tw}(G)$ indicates how close to a tree G is. For instance, forests have tree-width 1. With regards to tree-width, a class of graphs $\mathcal{C}$ is considered simple if it has *bounded tree-width*: there is some constant c such that $\mathrm{tw}(G) \leqslant c$ for any $G \in \mathcal{C}$.

Numerous problems, including independent set, can be solved in linear time on bounded tree-width classes. Precisely, they have FPT algorithms with the tree-width as parameter: for instance, a maximum independent set in G can be found in time $2^{\mathrm{tw}(G)} \cdot n$. Notice here that the parameter in the complexity is only the tree-width, and not the size of the desired independent set. This extends to any problem described using a *monadic second-order* formula [32], a logic far more expressive than first-order logic, allowing quantification on subsets of vertices and edges. Notable such problems include finding proper colourings (colourings of vertices with distinct colours on adjacent vertices) or Hamiltonian cycles (a cycle going through all vertices).

The class of graphs with tree-width at most k is minor closed, hence avoids some graphs as minor (e.g. K_k). There are however minor avoiding classes with unbounded tree-width: for instance planar graphs. The *grid minor theorem* of Robertson and Seymour proves that a class $\mathcal{C}$ has bounded tree-width if and only if it avoids a *planar* graph as minor [83]. Thus for a minor-closed class $\mathcal{C}$, tree-width gives a second, stronger notion of simplicity: either $\mathcal{C}$ has bounded tree-width, which makes it very simple (independent set is solved in linear time), or $\mathcal{C}$ avoids a minor but contains all planar graphs, and is relatively simple (independent set is NP-hard, but has a PTAS), or $\mathcal{C}$ is the very complex class of all graphs.

Clique-width. Tree-width is a sparse graph notion: any dense graph has large tree-width, for instance cliques have $\mathrm{tw}(K_n) = n$. Results about tree-width are thus of little use to dense graphs. Clique-width, attributed to Courcelle, Engelfriet, and Rozenberg [33], is a variant of tree-width which accommodates dense graphs. Whereas the tree-width $\mathrm{tw}(G)$ indicates whether G resembles a tree, the clique-width $\mathrm{cw}(G)$ indicates how easily G can be encoded as a tree. For instance, cographs, which are easily encoded by the tree of operations (disjoint and complete sum) used to construct them, have clique-width at most 2.

Clique-width satisfies $\mathrm{cw}(G) \leqslant O(2^{\mathrm{tw}(G)})$. Thus any class of graphs with bounded tree-width also has bounded clique-width. Conversely, a class with bounded clique-width which avoids some bipartite complete graph $K_{t,t}$ as subgraph has bounded tree-width. In that sense, clique-width is the dense analogue of tree-width.

With clique-width as parameter, there are FPT algorithms for all problems expressed in a weaker variant of monadic second-order logic allowing quantification on subsets of vertices, but not subsets of edges. This still includes independent set and colouring, but not Hamiltonian cycles.

Twin-width. This leads us to the main subject of this work: *twin-width* is a graph complexity measure introduced by Bonnet, Kim, Thomassé, and

Watrigant [19], based on a notion of Guillemot and Marx for permutations [61]. We will see that classes with bounded tree-width or clique-width have bounded twin-width: indeed $\mathrm{tww}(G) \leqslant 2\,\mathrm{cw}(G)$. Twin-width however goes far further: planar graphs, and more generally classes avoiding a minor have bounded twin-width, but not bounded clique-width.

In a class $\mathcal{C}$ with bounded twin-width, there is an FPT algorithm for first-order logic problems [19], comparable to the result known in nowhere dense classes (note that nowhere dense and bounded twin-width are incomparable conditions). This algorithm however has a limitation: it must be given a *witness* of the twin-width of the input graph, and computing witnesses of twin-width is a major open problem. For instance, independent sets of size k in graphs of twin-width t can be found in time $t^{O(k)} \cdot n$ when provided with a witness of twin-width. Remark the difference in complexity with the tree-width algorithm: both twin-width and the solution size are parameters in the complexity. One cannot remove the dependency in k, since independent set can be NP-hard in classes with bounded twin-width such as planar graphs.

1.4 Organisation of this thesis

This thesis is concerned with twin-width, its properties and characterisations, and its relationship to some of the other notions presented in this introduction. It is organised as follows.

The first half of this work presents properties and results on twin-width, alongside examples of classes of graphs with bounded twin-width. Chapter 2 presents the definition of twin-width through *contraction sequences*, and uses it to construct the previously mentioned FPT algorithm for independent set, as well as results on graph colourings, and several simple examples of graphs with or without bounded twin-width. Chapter 3 introduces a second characterisation of graphs with bounded twin-width: their adjacency matrices, when judiciously ordered, avoid some grid-like structures. This leads to a proof that classes avoiding a minor have bounded twin-width, and to results on compact representations and enumeration of graphs with small twin-width. Chapter 4 presents two major results relating twin-width and first-order logic, with an extensive introduction of the logical notions involved. The results of these first three chapters come primarily from the work of Bonnet, Kim, Thomassé, and Watrigant which defined twin-width for graphs [19].

The latter half is motivated by the following question: can the previous remarkable properties of twin-width—algorithms, enumeration, etc.—be *characterisations* of twin-width? That is, are these properties satisfied *exclusively* by classes with bounded twin-width? While this is not the case in general for graphs, we will present examples in which such equivalences hold. In Chapter 5 presents such characterisations when extending twin-width to structures other than graphs: ordered structures, tournaments, permutations. More generally, twin-width is shown to be particularly useful and well behaved for these structures. Chapter 6 focuses further on permutations in particular: after revisiting the results of Guillemot and Marx which predate the definition of twin-width for graphs [61], we prove a factorisation theorem for permutations with bounded twin-width. Chapter 7 generalises twin-width to yet another kind of structures: infinite groups. After studying basic properties of twin-width for

groups, it presents the highly non-trivial construction of a group with infinite twin-width. This disproves a conjectured characterisation of twin-width, by providing a class which has unbounded twin-width but is small, i.e. has few graphs in some precise sense.

Chapter 2

CONTRACTION SEQUENCES

This chapter defines twin-width through *contraction sequences*, presenting a number of simple examples, and some applications on contraction sequences: dynamic programming algorithms, and graph colouring results. It is largely based on the first paper on twin-width of Bonnet, Kim, Thomassé, and Watrigant [19], and the followup work of the same with the author [14, 15].

2.1 Definitions

Consider a graph $G = (V, E)$, and a partition $\mathcal{P}$ of its vertex set V. Given distinct parts $X, Y \in \mathcal{P}$, we distinguish three situations.

1. If all vertices of X are adjacent to all vertices of Y, then X, Y are said to be *complete* to each other.

2. Symmetrically, if no vertex of X is adjacent to a vertex of Y, then X, Y are said to be *anticomplete*.

   In either of these first two cases, X and Y are also said to be *homogeneous*.

3. When X and Y are not homogeneous, meaning that there is at least one edge and one non-edge them, we say that X and Y are *in error*.

These three situations are described in a quotient structure called a *trigraph*: the trigraph $Tri(G, \mathcal{P})$ has $\mathcal{P}$ for vertex set, and between two parts $X, Y \in \mathcal{P}$, there is (1) no edge if X, Y are anti-complete, (2) a *normal edge* if they are complete, (3) and an *error edge* if they are non-homogeneous. Thus, a trigraph (V, E, R) is defined by a vertex set V, and two sets E, R of *normal* and *error* edges respectively, so that (V, E) and (V, R) are two graphs, and E, R are disjoint. By convention, normal edges are depicted in black, and error edges in red. See Figure 2.1 for an example.

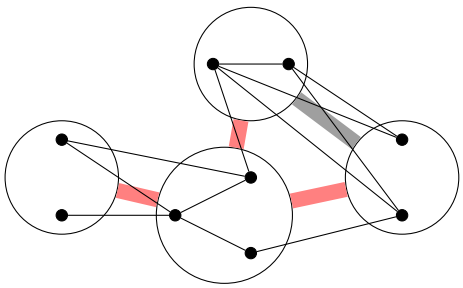


Figure 2.1: Example of a partition of a graph represented by circles, and the associated trigraph in thick edges (black for normal edges, and red for error edges).

The point of this definition is that if the trigraph $Tri(G, \mathcal{P})$ has few error edges, (the meaning of 'few' will be clarified shortly) then it is a good abstraction of G. For instance, in the extreme case where $Tri(G, \mathcal{P})$ contains no error edge, the graph G is entirely described by $Tri(G, \mathcal{P})$, and the restriction of G to each part of $\mathcal{P}$. More generally, if one wishes to describe G, it is sufficient to give:

1. the trigraph $Tri(G, \mathcal{P})$ itself, describing the large-scale structure of G,
2. for each part $X \in \mathcal{P}$, the local induced subgraph $G[X]$,
3. and for each non-homogeneous pair $X, Y \in \mathcal{P}$, the bipartite subgraph induced by G between X and Y.

When there are few error edges and the partition $\mathcal{P}$ is reasonably balanced, such a description may be significantly smaller than the naive representation of G. Nonetheless, having a partition $\mathcal{P}$ with few error edges does not by itself ensure that G is simple: the quotient trigraph, and the subgraphs induced by parts or by error edges could all be arbitrarily complex.

This leads to the following definition, whose underlying idea is to ask for a good partition at all possible scales. A *contraction sequence* for a graph G is a sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ of partitions of $V(G)$, which

1. starts with the partition into singletons $\mathcal{P}_n = \{\{x\} \mid x \in V(G)\}$,
2. finishes with the trivial partition $\mathcal{P}_1 = \{V(G)\}$, and
3. progresses from $\mathcal{P}_{i+1}$ to $\mathcal{P}_i$ by merging two parts, that is replacing some parts $X, Y \in \mathcal{P}_{i+1}$ with $(X \cup Y) \in \mathcal{P}_i$.

The indices of the partitions are chosen backwards so that $\mathcal{P}_i$ has exactly i parts.

Notice that the definition of contraction sequence does not involve the edges of G in any way: this is because we have not yet required the partitions $\mathcal{P}_i$ to 'have few error edges'. In a trigraph $H = (V, E, R)$, the *error degree* of $v \in V$, denoted by $\deg_H^{Err}(v)$, is the degree of v in (V, R), that is the number of error edges incident to v. The *width* of a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ is the maximum error degree observed throughout the sequence:

$$\max_{i \leqslant n} \max_{X \in \mathcal{P}_i} \deg^{Err}_{Tri(G,\mathcal{P}_i)}(X).$$

Thus, this condition is about the number of error edges not globally, but locally around each part. Finally, the *twin-width* of the graph G is the minimum width of a contraction sequence for G. See Figure 2.2 for a first illustration.

We invite the reader to check the following basic properties:

1. Given a graph $G = (V, E)$, denote by $G^C = (V, \binom{2}{V} \setminus E)$ its *complement*, obtained by replacing edges with non-edges and vice-versa. Then $\mathrm{tww}(G) = \mathrm{tww}(G^C)$.

2. For any induced subgraph H of G, $\mathrm{tww}(H) \leqslant \mathrm{tww}(G)$: twin-width is *monotone under taking induced subgraphs*.

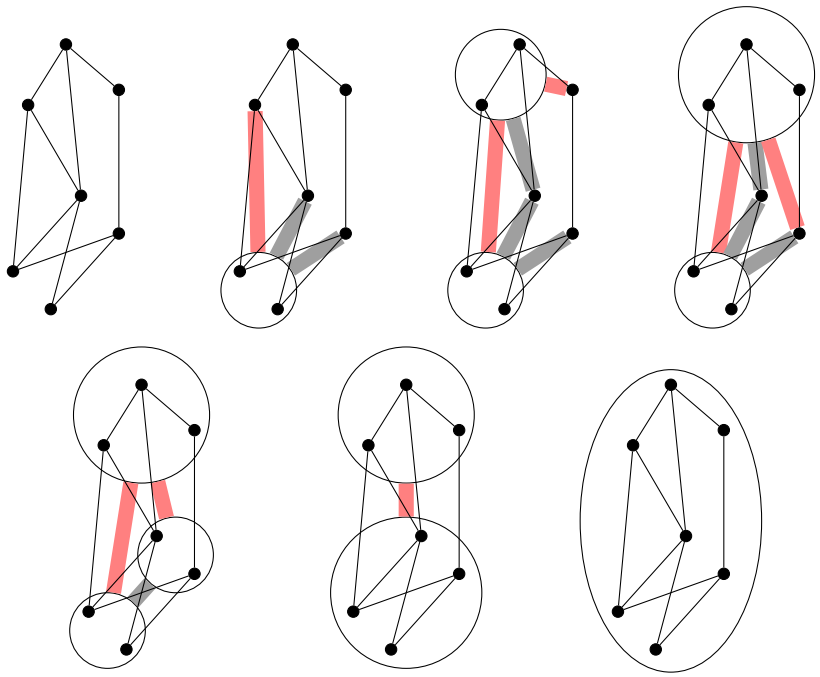


Figure 2.2: Example of a contraction sequence of width 2. The contraction steps are represented in reading order. At each step, parts are represented by circles, and the quotient trigraphs are drawn in thick edges. For simplicity, singleton parts and the edges joining them are omitted.
In this sequence of trigraphs, no vertex is incident to more than two error edges, hence the width of the contraction sequence is 2. One may check that this is optimal for this graph, because it contains C_5 as induced subgraph. Thus, its twin-width is 2.

2.2 First examples and basic constructions

The path P_n is the graph with vertex set $[n]$ (where $[n]$ denotes $\{1, \dots, n\}$), and edges from i to $(i+1)$ for all $i \in [n-1]$. Consider the sequence of partitions $\mathcal{P}_n, \dots, \mathcal{P}_1$ where

$$\mathcal{P}_i = \big\{\{1\}, \{2\}, \dots, \{i-1\}, \{i, \dots, n\}\big\},$$

see Figure 2.3. This is clearly a contraction sequence, and in each trigraph $Tri(P_n, \mathcal{P}_i)$, there is exactly one error edge between $\{i-1\}$ and $\{i, \dots, n\}$. It follows that the maximum error degree is 1, and $\mathrm{tww}(P_n) \leqslant 1$. The cycle C_n is obtained by adding the edge from 1 to n to the path P_n. With the very same sequence of partitions, the trigraph $Tri(C_n, \mathcal{P}_i)$ now has at most two error edges from $\{i, \dots, n\}$ to $\{i-1\}$ and $\{1\}$ respectively, from which it follows that $\mathrm{tww}(C_n) \leqslant 2$.

These two bounds are optimal for sufficiently long paths and cycles; this is a good opportunity to introduce the following definition, which is the etymology of 'twin-width'. Given distinct vertices x, y in a graph G, their *neighbourhood difference* $\Delta_N(x, y) = (N(x) \Delta N(y)) \setminus \{x, y\}$ is the set of other vertices adjacent

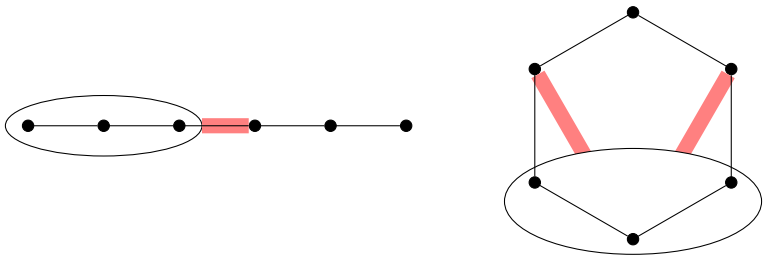


Figure 2.3: Typical partition in a contraction sequence for a path and a cycle.

to exactly one of x and y. The vertices x and y are said to be *k-near twins* if $|\Delta_N(x, y)| \leqslant k$. Vertices which are 0-near twins are simply called *twins*.[1]

Lemma 2.1. *If G is a graph with $\mathrm{tww}(G) = k$, then G contains k-near twins.*

Proof. Given a contraction sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$ of width k for G, it suffices to consider the very first contraction, i.e. $\mathcal{P}_{n-1}$. There is exactly one part $\{x, y\}$ in $\mathcal{P}_{n-1}$ with two vertices, the others being singletons. Further, if $z \in \Delta_N(x, y)$, then $\{z\}$ is in error with $\{x, y\}$. It follows that

$$|\Delta_N(x, y)| = \deg^{Err}_{Tri(G,\mathcal{P}_{n-1})}(\{x, y\}) \leqslant k,$$

i.e. x and y are k-near twins. □

The reader may now check that paths on at least 4 vertices do not contain twins, and cycles on at least 5 vertices do not contain 1-near twins, from which it follows that the former bounds on twin-width are tight.

2.2.1 Contractions in trigraphs, trees and grids. So far, we have defined contractions sequences in terms of the sequence of partitions. This point of view is useful in many contexts, for instance in the algorithms presented at the end of this chapter, and the grid theorem of Chapter 3. However, when working with concrete examples of graphs, it is often more natural to directly consider the quotient trigraphs.

Given a trigraph $G = (V, E, R)$ and two arbitrary vertices $x, y \in V$, the *contraction* of x, y is the operation which replaces x, y with a new vertex z, with the following edges: for any other vertex $v \in V \setminus \{x, y\}$

1. if v is connected to both x and y by normal edges, then v is connected to z by a normal edge,
2. symmetrically, if v is adjacent to neither x nor y, then v is not adjacent to z, and
3. in any other case, v is connected to z by an error edge.

A simple case analysis shows that contractions correspond to merging parts in the following sense.

[1] In some contexts, one distinguishes *true* and *false* twins depending on whether or not the twins are adjacent. This distinction is not relevant to twin-width.

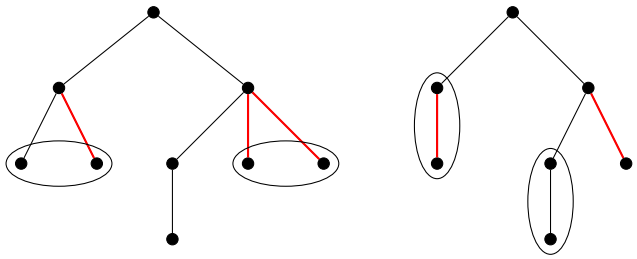


Figure 2.4: Typical trigraphs in the contraction sequences of width 2 for trees. The circled pairs of vertices should be contracted for the next step; for the trees in this example, several choices are possible.

Lemma 2.2. *Consider a graph G, a partition $\mathcal{P}$ of $V(G)$, two parts $X, Y \in \mathcal{P}$, and $\mathcal{P}'$ obtained from $\mathcal{P}$ by merging X and Y. Then $Tri(G, \mathcal{P}')$ is obtained from $Tri(G, \mathcal{P})$ by contracting the vertices X and Y.*

A characterisation of contraction sequences through trigraphs follows.

Lemma 2.3. *Let G be a graph and $G_n, \dots, G_1$ a sequence of trigraphs. Then there is a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ for G such that $G_i = Tri(G, \mathcal{P}_i)$ if and only if*

1. *$G_n = (V(G), E(G), \varnothing)$, i.e. has the same vertices and edges as G and no error edge,*
2. *G_1 is the trigraph with only one vertex, and*
3. *G_i is obtained from G_{i+1} by contracting two vertices.*

We somewhat abusively call 'contraction sequence' both the sequence of partitions $\mathcal{P}_n, \dots, \mathcal{P}_1$ and the corresponding sequence of trigraphs $G_n, \dots, G_1$. It should be clear from the context and notations whether we are referring to partitions or trigraphs.

Let us now resume our list of graphs of small twin-width, using this new characterisation to simplify the descriptions of the contraction sequences.

Fact 2.4 [19]. *Trees have twin-width at most 2.*

Proof. Consider a tree T. We will construct a contraction sequence $G_n, \dots, G_1$ subject to the following conditions:

1. for each trigraph $G_i = (V_i, E_i, R_i)$, the graph $(V_i, E_i \cup R_i)$ is a tree,
2. each error edges of G_i is incident to a leaf of this tree, and
3. the maximum error degree in G_i is at most 2.

Initially, G_n has the same edges as T and no error edge, and clearly satisfies the conditions. Then, given G_{i+1}, we construct G_i by the following rules (see Figure 2.4).

- If G_{i+1} contains two leaves with the same parent, contract them.
- Otherwise, there exists a leaf x with no siblings (e.g. take x at maximal distance from the root); contract x with its parent.

These rules apply until the remaining trigraph is reduced to one vertex, and thus yield a contraction sequence. Clearly, the rules ensure that G_i remains a tree. Furthermore, the vertex z in G_i resulting from the contraction is a leaf, and any new error edge is incident to z, hence condition (2) is preserved. Let us now verify that the error degree cannot exceed 2.

- Assume the first rule is applied to leaves x, y with parent t, contracted into z. The error degrees of vertices outside x, y, t are unaffected, hence we only need to consider z and t. Since z is a leaf, its error degree is at most 1. As for t, remark that zt is an error edge only if xt or yt was an error edge, and the contraction of x, y cannot create error edges from t to a vertex other than z. It follows that the error degree of t cannot increase in this operation.

- The second rule is only applied when for each node of G_{i+1}, at most one child is a leaf. Thus after the contraction, in G_i, no node has more than 2 children leaves. It follows from condition (2) that the error degree cannot exceed 2. □

For a lower bound matching Fact 2.4, a slightly tedious case disjunction shows that the subdivided claw (i.e. the tree whose root has 3 children, each of which has a single child leaf) has twin-width exactly 2.

The $(n \times m)$-grid[2] is the graph on vertex set $[n] \times [m]$ in which two vertices are adjacent if and only if they are at distance 1 in the plane.

Fact 2.5 [19]. *Grids have twin-width at most 4.*

Proof. The contraction sequence for the $(n \times m)$-grid is as follows: the m vertices of the first column are contracted with those of the second column, that is, $(1, 1)$ with $(2, 1)$, then $(1, 2)$ with $(2, 2)$, until $(1, n)$ with $(2, n)$. At this point, the trigraph obtained is the $(n \times (m - 1))$ grid, except that all edges incident to the leftmost column are error edges. We then repeat this process until only a path remains, which is easily contracted. The reader may verify that at all steps in this process, the maximum error degree is 4, which is reached only by the vertex resulting from the previous contraction, see Figure 2.5. □

An elegant argument of Ahn, Chakraborti, Hendrey, and Oum [2] shows that the bound of 4 is reached for grids of size at least 7×7.

This contraction process generalises to grids in any fixed dimension d (i.e. the graph on vertex set $[n]^d$ where vertices are adjacent when they are at distance 1 in $\mathbb{R}^d$). One chooses an axis, and contracts the first hyperplane with the second, then with the third, until a grid of dimension $(d-1)$ remains. This contraction sequence has width linear in d, and a near-twins argument shows that this is optimal up to the multiplicative constant.

Fact 2.6 [19, Theorem 4.3]. *Grids of dimension d have twin-width $\theta(d)$.*

[2]Not to be confused with the grids in matrices used throughout Chapter 3, which are unrelated, except for their visual representation.

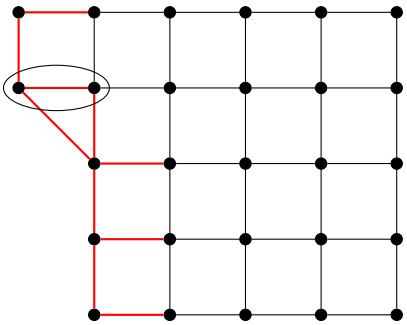


Figure 2.5: Typical trigraph in the contraction sequence and width 4 for grids. The circled pair of vertices should be contracted for the next step.

2.2.2 Contraction tree and cographs. As our next example, we will characterise the graphs of twin-width 0: they are exactly the cographs. To this end, let us describe how contraction sequences can be represented as trees.

Consider any contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ for a graph G. We incrementally define binary trees $T_1, \dots, T_n$ such that the leaves of T_i are exactly the elements of $\mathcal{P}_i$.

- The tree T_1 consists only of its root, corresponding to $V(G)$.
- Assume that T_i has been constructed, and assume that X, Y are the parts of $\mathcal{P}_{i+1}$ which are merged to obtain $\mathcal{P}_i$. Thus there is a leaf of T_i corresponding to $X \cup Y$, and T_{i+1} is obtained from T_i by adding X and Y as children of $X \cup Y$.

The last tree $T = T_n$ is called the *contraction tree* of the sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$. The nodes of T are all the subsets of $V(G)$ which appear in $\mathcal{P}_n, \dots, \mathcal{P}_1$. Its leaves are the singletons, in bijection with $V(G)$, while each internal node of T is the disjoint union of its children. The contraction tree partially records the order of contractions, but does not fully describe the contraction sequence: if X, Y are nodes of T, neither being a descendant of the other, then T does not indicate which of X or Y was created first in the sequence.

There is another, more abstract description of this contraction tree, which is an opportunity to introduce useful vocabulary on partitions. Consider again a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$. Then for any $i \geqslant j$, the partitions $\mathcal{P}_i, \mathcal{P}_j$ satisfy

$$\forall X \in \mathcal{P}_i, Y \in \mathcal{P}_j, \quad \text{either } X \subseteq Y \text{ or } X \cap Y = \varnothing, \tag{2.1}$$

i.e. $\mathcal{P}_i$ is obtained by splitting parts of $\mathcal{P}_j$. It is said that $\mathcal{P}_i$ is a *refinement* of $\mathcal{P}_j$, or that $\mathcal{P}_j$ is a *coarsening* of $\mathcal{P}_i$. To verify (2.1), remark that $\mathcal{P}_{i+1}$ obviously refines $\mathcal{P}_i$, and that the refinement relation on partitions is transitive—it is a partial ordering.

Now denote by $\mathfrak{P} = \bigcup_{i=1}^{n} \mathcal{P}_i$ the collection of all parts appearing throughout the contraction sequence. It easily follows from (2.1) that $\mathfrak{P}$ satisfies

$$\forall X, Y \in \mathfrak{P}, \quad \text{either } X \subseteq Y,\ Y \subseteq X, \text{ or } X \cap Y = \varnothing, \tag{2.2}$$

i.e. subsets in $\mathfrak{P}$ never intersect in a non-trivial way. This property of the family $\mathfrak{P}$ is known as being *laminar*; it guarantees that there exists a unique

tree[3] T whose nodes are elements of $\mathfrak{P}$, and such that a node X is a descendant of Y if and only if $X \subseteq Y$. This is again the contraction tree.

Let us now apply this construction to graphs of twin-width 0: assume that $\mathcal{P}_n, \dots, \mathcal{P}_1$ is a contraction sequence of G where the trigraphs $Tri(G, \mathcal{P}_i)$ do not contain any error edges, and consider T the associated contraction tree. Let X be an internal node of T with children Y_1, Y_2, meaning that $X = Y_1 \uplus Y_2$. These correspond to some step i in the contraction sequence: $\mathcal{P}_i$ is obtained from $\mathcal{P}_{i+1}$ by replacing $Y_1, Y_2 \in \mathcal{P}_{i+1}$ with X. Since $Tri(G, \mathcal{P}_{i+1})$ has no error edge, Y_1 and Y_2 must be either complete or anti-complete to each other. By labelling the node X with 0, resp. 1, when Y_1 and Y_2 are anti-complete, resp. complete to each other, we find that G satisfies the following:

Property 2.7. There is a tree T whose set of leaves is $V(G)$, and whose internal nodes are labelled with either 0 or 1, such that any two vertices $x, y \in V(G)$ are adjacent if and only if their least common ancestor in T is labelled with 1.

Property 2.7 characterises *cographs*, which we presented in Chapter 1 as the graphs constructed by a sequence of disjoint and complete unions. Indeed, the tree T describes how to construct G with disjoint and complete unions: calling G_t the subgraph of G induced by the descendants of t, one can check that if t is a node in T labelled with 0 (resp. 1) and with children t_1, t_2, then G_t is the disjoint (resp. complete) union of G_{t_1}, G_{t_2}. This tree T describing the construction of G is called *cotree*. See Figure 2.6 for an example.

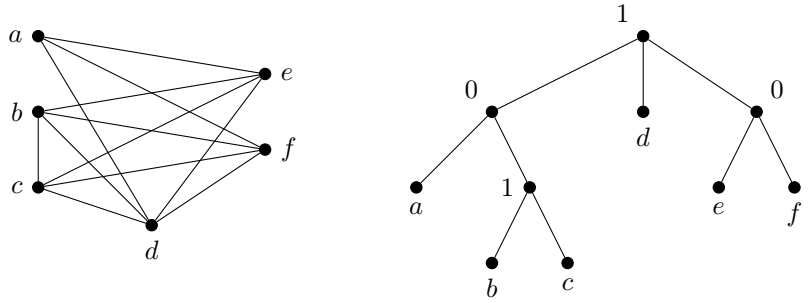


Figure 2.6: Example of a cograph, and a cotree describing it.

Conversely, given a cograph G with cotree T, one can construct a contraction sequence of width 0 as follows. Without loss of generality, there are no nodes in T with exactly one child—such nodes can be deleted. Assuming $|V(G) \geqslant 2|$, T must contain sibling leaves x, y, which are twins in the cograph G: indeed, for any $z \notin \{x, y\}$, the pairs x, z and y, z have the same least common ancestor, hence $xz \in E(G)$ if and only if $yz \in E(G)$. Now contract x and y: since they are twins, the resulting trigraph has no error edge, and has exactly the same vertices and (normal) edges as $G - x$ (or $G - y$). The latter is also a cograph, whose cotree is $T - x$ (resp. $T - y$), thus this process can be repeated until reaching the one-vertex graph. This is a contraction sequence in which trigraphs have no error edges, hence $\mathrm{tww}(G) = 0$. Thus,

Fact 2.8 [19]. *A graph G is a cograph if and only if* $\mathrm{tww}(G) = 0$.

[3] Actually a forest and not a tree in the general case. Here T is indeed a tree since the full set $V(G)$ is in $\mathfrak{P}$, and is the root of T.

2.2.3 Clique-width. In light of Fact 2.8, twin-width can be seen as a natural generalisation of cographs, in much the same way that tree-width is a generalisation of trees. This is interesting as there is another natural generalisation of cographs as a complexity measure: clique-width. As announced in the introduction, we will now see that classes with bounded clique-width also have bounded twin-width. This result was proved in [19], but the precise bound proved below was first published in [9].

Let us first define clique-width. It generalises the construction of cographs through disjoint and complete unions, by allowing more complex operations. These operations are defined on graphs G with some additional information: a *labelling* $\lambda_G : V(G) \to [k]$, for some fixed k. The operations are the following:

disjoint union $G \oplus H$, which makes a disjoint union of the vertices of G and H while preserving the existing edges and labellings.

adding edges between labels $i, j \in [k]$, which preserves the vertices and their labels, but adds the edge xy for any vertices x, y with labels i and j respectively.

relabelling by $f : [k] \to [k]$, which keeps the graph G unchanged, but replaces the labelling λ_G with $f \circ \lambda_G$.

The clique-width $\mathrm{cw}(G)$ is the minimum number k of labels needed to construct G with these operations, starting from graphs with a single vertex.

Fact 2.9 (Baril, Couceiro, Lagerkvist [9]). *For any graph G,*

$$\mathrm{tw}(G) \leqslant 2\,\mathrm{cw}(G) - 1.$$

Proof. Fix the number k of labels. Given G labelled with $\lambda_G : V(G) \to [k]$, the labelling defines a canonical partition of the vertices

$$\mathcal{P}_G = \left\{ \lambda_G^{-1}(i) \mid i \in [k] \right\}.$$

For the sake of induction, we will prove the following: for any labelled graph (G, λ_G) constructed by disjoint union, relabelling, and addition of edges using at most k labels, there is a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_\ell$ of width at most $2k - 1$ which finishes not with the trivial partition $\{V(G)\}$, but rather with the canonical partition $\mathcal{P}_\ell = \mathcal{P}_G$. When G is equipped with the trivial labelling mapping all vertices to 1 (which can always be obtained by a single relabelling operation), this proves the result.

disjoint union Assume that $(G, \lambda_G), (H, \lambda_H)$ have contraction sequences as above. In the disjoint union $G \oplus H$, one can independently apply the contraction sequence of G and that of H without any additional errors. The partition obtained at this point is $\mathcal{P}_G \uplus \mathcal{P}_H$. To obtain the desired partition $\mathcal{P}_{G \oplus H}$, it only remains to merge $\lambda_G^{-1}(i)$ with $\lambda_H^{-1}(i)$ for each label $i \in [k]$. Since there remains no more than $2k$ parts at this point, this cannot increase the error degree beyond $2k - 1$.

adding edges Fix a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_\ell$ as above for (G, λ_G). Remark that any part $P \in \mathcal{P}_i$ in the sequence is contained in some $\lambda_G^{-1}(i)$, i.e. all vertices of P have the same label. We now add edges from vertices

labelled i to the ones labelled j. Given parts $X, Y \in \mathcal{P}_i$, there are two cases: if X is labelled i and Y labelled j (or vice versa), then the new edges ensure that X, Y are homogeneous; otherwise, no edge is added between them. Either way, the new edges cannot create an error between X and Y. Thus the same contraction sequence also still satisfies the requirement after adding the edges.

relabelling Consider G labelled with $\lambda : V(G) \to [k]$, and a relabelling map $f : [k] \to [k]$. Call $\mu = f \circ \lambda$ the new labels, and $\mathcal{P}_\lambda, \mathcal{P}_\mu$ the canonical partitions corresponding to λ, μ. Notice that vertices with the same labels in λ also have the same labels in μ. It follows that $\mathcal{P}_\lambda$ is a refinement of $\mathcal{P}_\mu$. Suppose now that we have a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_\ell$ with width $2k-1$, such that $\mathcal{P}_\ell = \mathcal{P}_\lambda$. Since $\mathcal{P}_\lambda$ refines $\mathcal{P}_\mu$, this sequence can be extended with a few merges to reach $\mathcal{P}_\mu$. Since $\mathcal{P}_\lambda$ already has no more than k parts, these additional steps cannot increase the error degree beyond $k-1$. We thus obtain the desired contraction sequence for (G, μ). □

As announced in the introduction, we will see in section 3.3.1 that planar graphs have bounded twin-width. They are well-known to have unbounded clique-width, thus providing an example which separates bounded twin-width from bounded clique-width. Interestingly, the definition of twin-width can be restricted to characterise clique-width: defining *component twin-width* by measuring the maximal size of a connected component in the graphs of error edges of a contraction sequence, one finds that clique-width and component twin-width are within a factor 2 of each other [18, 9].

2.2.4 Some graphs with large twin-width. So far, we have seen several examples of families of graphs whose twin-width is bounded by small constants. Let us now present classes of graphs which on the contrary have unbounded twin-width.

The $(n \times n)$ *rook graph* is the graph with vertex set $[n] \times [n]$, in which (x_1, y_1) and (x_2, y_2) are adjacent whenever either $x_1 = x_2$ or $y_1 = y_2$. It represents the possible moves of a rook on a chessboard, hence the name.

Fact 2.10 [19]. *The $(n \times n)$ rook graph has twin-width at least $2n-4$.*

Proof. Using Lemma 2.1, it suffice to show that the $(n \times n)$ rook graph has no $(2n-5)$-near twins. Consider distinct vertices $v_1 = (x_1, y_1)$ and $v_2 = (x_2, y_2)$ in $[n] \times [n]$. Without loss of generality, assume that they are on distinct columns, i.e. $x_1 \neq x_2$. Then for any $y \in [n] \setminus \{y_1, y_2\}$, we have that

$$(x_1, y) \in N(v_1) \setminus N(v_2) \qquad \text{and} \qquad (x_2, y) \in N(v_2) \setminus N(v_1)$$

This gives $2(n-2)$ vertices in the symmetric difference of neighbourhoods, hence v_1, v_2 are not $(2n-5)$-near twins. □

This near-twins argument amounts to only considering the first step in the contraction sequence. In the next example, we will instead need to consider a well chosen step in the middle of the sequence to obtain the desired lower bound on twin-width.

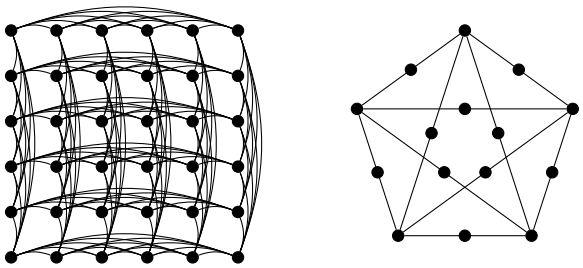


Figure 2.7: Rook graph, and 1-subdivided clique.

The r-subdivision of a graph G, denoted here by $G^{(r)}$, is the graph obtained by replacing each edge xy of G by a path on $(r+1)$ edges joining x to y. In the subdivision of G, the vertices which originate from $V(G)$ are called *central vertices*, while the added vertices (which have degree 2) are called *subdivision vertices*.

Fact 2.11 [14, section 6]. *For $k \geqslant 1$, if $K_n^{(k)}$ has twin-width t, then*

$$n - 1 \leqslant (t+2)^{k+1}.$$

Proof. Let $G = K_n^{(k)}$, and assume that $\mathrm{tww}(G) = t$. In a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ of width t for G, consider the first contraction involving a central vertex, that is, let i be maximal such that there is a part $X \in \mathcal{P}_i$ which is not a singleton and contains a central vertex x. For $r \in [k+1]$, denote by C_r the set of vertices at distance exactly r from x in G. There are exactly $n-1$ vertices in C_r, which are subdivision vertices when $r \leqslant k$, and central vertices when $r = k+1$.

Claim 2.12. Let $P \in \mathcal{P}_i$ be a part which is not a singleton. Then there are at most $t+1$ parts in $\mathcal{P}_i$ which are connected in $Tri(G, \mathcal{P}_i)$ to P by either a normal or an error edge.

Proof. By assumption, at most t parts of $\mathcal{P}_i$ are connected to P by an error edge, thus we only need to show that at most 1 part is connected to P by a normal edge. Suppose for a contradiction that P_1, P_2 are connected to P by normal edges. Consider $x, z \in P$ (since it is not a singleton), $y \in P_1$ and $w \in P_2$. Then $xyzw$ is a cycle of length 4 in G, which cannot exist since G is a subdivided graph. ■

Remark now that all vertices in C_1 are either in X, or in one of the $t+1$ parts adjacent to X (by a normal or error edge). Thus the vertices of C_1 are split among at most $t+2$ parts of $\mathcal{P}_i$, one of which, say X_1, must contain at least $\frac{n-1}{t+2}$ of the vertices of C_1. These vertices in $C_1 \cap X_1$ have at least $\frac{n-1}{t+2}$ neighbours in C_2, which again are split between at most $t+2$ parts, one of which contains at least $\frac{n-1}{(t+2)^2}$ of them. This process can be repeated to find a part X_r with at least $\frac{n-1}{(t+2)^r}$ vertices of C_r, until either $r = k+1$ is reached, or $\frac{n-1}{(t+2)^r} \leqslant 1$. Recall now that $\mathcal{P}_i$ was chosen to be the first step in which a central vertex, namely x, is involved in a contraction. It follows that no two vertices of C_{k+1} are in the same part of $\mathcal{P}_i$, hence necessarily $\frac{n-1}{(t+2)^{k+1}} \leqslant 1$. □

Fact 2.11 implies that for any constant r, the class of r-subdivided graphs has unbounded twin-width. Further, one can replace the constant r by a slow growing function: for any function $f : \mathbb{N} \to \mathbb{N}$, consider the class $\{K_n^{(f(n))}\}_{n\in\mathbb{N}}$. It follows from Fact 2.11 that this class has unbounded twin-width whenever $f(n) = o(\log n)$. We will revisit the twin-width of subdivided graphs much later in section 6.7.

2.3 Colouring

It is time to present some applications of twin-width: we will use contraction sequences of small width to obtain good colourings of graphs.

A (proper) *k-colouring* of a graph G is a map $\lambda : V(G) \to [k]$ such that the colours $\lambda(x), \lambda(y)$ are different whenever x and y are adjacent. When such a colouring exists, the graph G is called k-colourable. The *chromatic number* $\chi(G)$ is the smallest k such that G is k-colourable.

We wish to show that graphs with small twin-width can be coloured with few colours. Unfortunately, the clique K_n has twin-width 0, and requires n colours; thus one can certainly not colour graphs of twin-width t with $f(t)$ colours for any function t. Fortunately, there is a very well-known notion of 'good colourings' for graphs which may contain large cliques. The *clique number* $\omega(G)$ denotes the maximum size of a clique contained in G. Remark that $\omega(G) \leqslant \chi(G)$. A hereditary class $\mathcal{C}$ of graphs is *χ-bounded* if there exists a function f such that any $G \in \mathcal{C}$ satisfies $\chi(G) \leqslant f(\omega(G))$. In essence, $\mathcal{C}$ being χ-bounded means that the only reason for which a graph $G \in \mathcal{C}$ may require a large number of colours is that it contains a large clique. Remark that the class of all graphs is not χ-bounded: there are numerous constructions of graphs with no triangle (hence certainly no larger cliques) but arbitrarily large chromatic number [88, 74, 37], and we will present yet another in section 2.3.1.

We will show in this section that graphs of twin-width t are χ-bounded. As an introduction, we first consider the case of triangle-free graphs, that is graphs satisfying $\omega(G) = 2$.

Lemma 2.13 [15, Theorem 20]. *Triangle-free graphs of twin-width t are $(t+2)$-colourable.*

Proof. Let $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a contraction sequence for G of width t. We use that G has no triangle in the following.

Claim 2.14. A part $P \in \mathcal{P}_i$ either is edgeless (i.e. there are no edges between vertices of P), or is not incident to any normal edge in $Tri(G, \mathcal{P}_i)$.

Proof. If uv is an edge inside P and PP' is a normal edge in $Tri(G, \mathcal{P}_i)$, then uvw is a triangle for any $w \in P'$. ■

Consider now the contraction sequence in reverse order, starting with $\mathcal{P}_1$. We will construct a proper colouring $\lambda_i : \mathcal{P}_i \to [t + 2]$, meaning that adjacent parts (by either a normal or a red edge) are given distinct colours. For $\mathcal{P}_1 = \{V(G)\}$ this is trivial, we assign an arbitrary colour to $V(G)$.

Suppose now that $\mathcal{P}_i$ is obtained by merging $X, Y \in \mathcal{P}_{i+1}$, and that we have already coloured $\mathcal{P}_i$ as λ_i. Then we colour $\mathcal{P}_{i+1}$ as follows (see Figure 2.8):

- Any $Z \neq X, Y$ keeps its colour $\lambda_{i+1}(Z) = \lambda_i(Z)$.

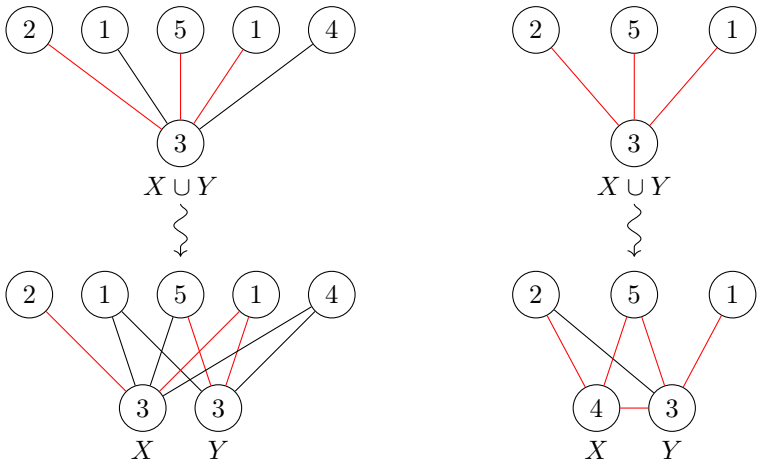


Figure 2.8: Update rules for colourings of triangle-free graphs of twin-width t ($t = 3$). The neighourhood of X, Y is represented. Left case: the normal edges incident to $X \cup Y$ ensures that X, Y are non-adjacent: they can keep the same colour. Right case: without normal edges incident to $X \cup Y$, the parts X, Y and their neighbours add up to at most $t+2$ parts, thus $t+2$ colours is enough to update X, Y.

- If $X \cup Y$ is incident to a normal edge, then it is edgeless by the claim, and in particular X, Y are non-adjacent in $Tri(G, \mathcal{P}_{i+1})$. Thus we can keep the same colour for both parts: $\lambda_{i+1}(X) = \lambda_{i+1}(Y) = \lambda_i(X \cup Y)$.
- Otherwise, $X \cup Y$ has at most t neighbours in $\mathcal{P}_i$. Discarding the colours of these neighbours, there remains two available colours among $t + 2$, which we give to X and Y respectively. □

2.3.1 A lower bound: twincut graphs. Before generalising Lemma 2.13 to a proof of χ-boundedness, let us present an almost matching lower bound: there are triangle-free graphs with twin-width t and chromatic number $t+1$ for arbitrary $t \in \mathbb{N}$. This construction, called *twincut graphs*, was introduced in a work with Bonnet, Bourneuf, Duron, Thomassé, and Trotignon [11] to answer an unrelated question of Chudnovsky, Penev, Scott, and Trotignon [30]: they are a class of triangle-free graphs which all contain either twins or a 2-cut, yet have unbounded chromatic number. The construction is heavily inspired by the well-known *Zykov graphs* [88].

The graphs G_k are constructed inductively as follows. First consider a tree T_k of depth $k-1$, in which nodes at depth $i < k-1$ have exactly $|G_{i+1}|$ children. Each node of T_k is a vertex of G_k. The edges of T_k however are *not* in G_k: they only serve to describe its structure. For each node $x \in T_k$ at depth $i < k-1$, create a copy of G_{i+1} in G_k using the children of x. Thus at depth 2, the children of the root form a copy of G_2; there are $|G_2|$ copies of G_3 at depth 3, $|G_2||G_3|$ copies of G_4 at depth 4, etc. Finally, for each branch b of T_k (i.e. a path from the root to a leaf), we add a new vertex v_b to G_k, and connect v_b to all nodes of b. We call v_b a *branch vertex*. The first graphs G_k

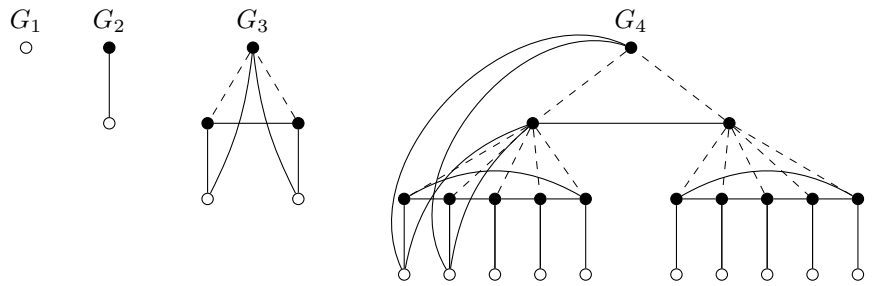


Figure 2.9: Construction of the first few twincut graphs. Dashed edges represent the underlying tree T_k; they are not part of the graph. Hollow vertices at the bottom are branch vertices. In G_4, only edges from the first two branch vertices are represented for readability.

are represented in Figure 2.9: G_1 is a single vertex, G_2 a single edge, and G_3 is the cycle C_5.

We will now show that these twincut graphs are triangle-free, and satisfy $\chi(G_k) = k$ and $\mathrm{tww}(G_k) \leqslant k-1$, and thus proving $\chi(G_k) \geqslant \mathrm{tww}(G_k)+1$, only one off the upper bound of Lemma 2.13.

Lemma 2.15. *The twincut graphs G_k are triangle-free.*

Proof. The neighbourhood of a branch vertex v_b is an independent set in G_k, hence a branch vertex cannot be part of a triangle. We can thus discard all branch vertices. What remains are disjoint copies of $G_1, \dots, G_{k-1}$, which are triangle-free by induction. □

Lemma 2.16 [11]. *The twincut graphs G_k have chromatic number $\chi(G_k) = k$.*

Proof. The proof is by induction on k. To show that G_k is k-colourable, remark that G_k consists of disjoint copies of $G_1, \dots, G_{k-1}$, connected by the branch vertices which are pairwise non-adjacent. Then we can inductively colour the copies of $G_1, \dots, G_{k-1}$ with colours $\{1, \dots, k-1\}$, and colour all branch vertices with k.

For the lower bound, consider λ be a proper colouring of G_k. We will find a branch $b = v_1, \dots, v_{k-1}$ of T_k in which the colours $\lambda(v_i)$ are all distinct. Starting from the root r, suppose by induction that we have found a path $r = v_1, \dots, v_{i-1}$ without repeated colours. The children of v_{i-1} form a copy of G_i, which by induction cannot be coloured with less than i colours. Thus there exists a child v_i of v_{i-1} whose colour $\lambda(v_i)$ is distinct from the $i-1$ already used colours $\lambda(v_1), \dots, \lambda(v_{i-1})$, by which we extend b. Having constructed b, consider its branch vertex v_b: it is adjacent to all of $v_1, \dots, v_{k-1}$, and thus needs to use a kth different colour. □

It only remains to bound the twin-width of twincut graphs. We would like to thank Bourneuf and Thomassé for this unpublished proof.

Lemma 2.17 (Bourneuf, Thomassé). *The twincut graphs G_k have twin-width* $\mathrm{tww}(G_k) \leqslant k-1$.

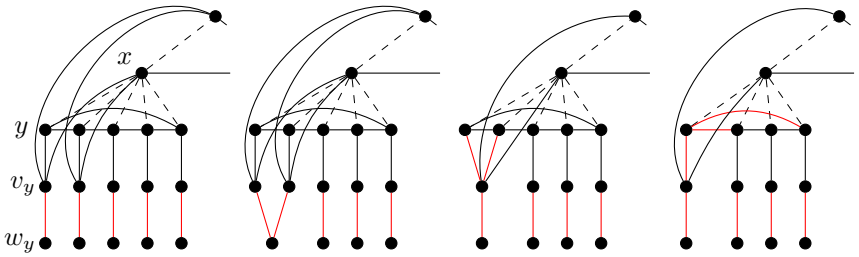


Figure 2.10: The three steps to contract y with its neighbour in G'_4.

Proof. Observe that the property is satisfied by the first few graphs: G_1, G_2 have twin-width 0, while $G_3 = C_5$ has twin-width 2.

Let us now prove the result for $k \geqslant 4$ by induction. We work with trigraphs, and for the sake of the induction, we will start with a slightly more complex graph than G_k: let G'_k be obtained by adding to G_k for each branch vertex v_b, a new vertex w_b connected only to v_b by an error edge.

Consider now a vertex $x \in T_k$ at depth $k-2$, and denote by X its children, which are leaves. Then $G'_k[X]$ is a copy of G_{k-1}, which by induction has twin-width at most $k-2$. To each child $y \in X$ is associated (1) a branch vertex v_y, corresponding to the branch from the root to the leaf y, and (2) a pending vertex w_y connected to v_y by an error edge. Fix a contraction sequence of width $k-2$ for G_{k-1}. We will replicate it on X, while also contracting the corresponding branch and pending vertices. Say that at some point in this contraction sequence, $y_1, y_2 \in X$ are to be contracted.

1. First we contract w_{y_1}, w_{y_2}, and call w_y the resulting vertex. Then w_y has error degree 2, other vertices keep the same error degree.

2. Next we contract v_{y_1}, v_{y_2} into v_y. Then v_y has exactly three error edges to y_1, y_2, w_y, as the other neighbours are shared by v_{y_1} and v_{y_2} (they are x and its ancestors). Furthermore, $y_i v_y$ is the only error edge from y_i to a vertex outside X.

3. Finally we contract y_1 and y_2 into y, reducing the error degree of v_y to 2. Once again, the only error edge from y to a vertex outside X is the one to v_y.

See Figure 2.10 for an illustration.

Thus, throughout these contractions, (1) the error degree of $y \in X$ is only one more than the error degree observed in the contraction sequence for G_{k-1}, hence at most $k-1$, and (2) all other vertices have error degree at most $3 \leqslant k-1$.

Once we have finished contracting the copy of G_{k-1} as described above, all of X is contracted into a single vertex y, with a branch vertex v_y, a pending vertex w_y, and the error edges yv_y and $w_y v_y$. This vertex y has no other neighbour, and we finally contract y and w_y. This process can be replicated for every choice of x at depth $k-2$, and the resulting trigraph is exactly G'_{k-1}, which by induction admits a contraction sequence of width $k-2$. □

2.3.2 χ-boundedness. Let us now come back to Lemma 2.13, which we wish to generalise to a χ-boundedness proof. We will reformulate the argument into a form which allows an induction on the clique number.

To prove Lemma 2.13, two recolouring rules were applied at each step of the contraction sequence: (1) if a part has no incident normal edge, then it has few neighbours, hence there always remains enough colours to recolour it, and (2) otherwise X is edgeless, hence when splitting X, both parts can keep the colour of X. Remark now that if we fix a vertex x, and consider the sequence of parts containing x, these two rules do not alternate: once $X \ni x$ becomes edgeless, any $Y \subset X$ will also be edgeless; we can in fact immediately assign the colour of X to x (and to all other vertices in X). The idea of the following reformulation is to only consider the moment when X becomes edgeless.

Theorem 2.18 [15, Theorem 21]. *Any graph G with twin-width t, clique number ω, and chromatic number χ satisfies*

$$\chi \leqslant (t+2)^{\omega-1}.$$

Proof. The proof is by induction on ω. The base case $\omega = 1$ corresponds to edgeless graphs and is trivial.

Fix a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ for G of width t, and let us consider the contraction tree T. Recall that its nodes are elements of the laminar family

$$\mathfrak{P} = \{X \in \mathcal{P}_i \mid i \in [n]\},$$

and that X is an ancestor of Y if and only if $X \supset Y$. Let $\omega = \omega(G)$ be the maximum size of a clique in G. We are now interested in the set of parts whose clique number strictly decreases compared to G:

$$\Omega = \{X \in \mathfrak{P} \mid \omega(G[X]) < \omega\}.$$

In the triangle-free case ($\omega = 2$), these were the edgeless parts. Finally, define

$$\mathcal{P} = \{X \in \mathfrak{P} \mid X \text{ is inclusion-wise maximal in } \Omega\}.$$

Claim 2.19. The family $\mathcal{P}$ is a partition of $V(G)$.

Proof. Two parts $X, Y \in \mathcal{P}$ cannot be contained in each other by maximality, and thus must be disjoint since $\mathfrak{P}$ is a laminar family. Furthermore, for any vertex $x \in V(G)$, the singleton $\{x\}$ is in Ω, hence either $\{x\}$ or some larger part containing it is in $\mathcal{P}$, and thus $\mathcal{P}$ covers $V(G)$. ■

We will now prove that $Tri(G, \mathcal{P})$ is $(t+1)$-*degenerate*: one can find an ordering $<$ of $\mathcal{P}$ for which each part $P \in \mathcal{P}$ has at most $t+1$ neighbours preceding P in $<$. This implies that $Tri(G, \mathcal{P})$ is $(t+2)$-colourable.

Given $X \in \mathcal{P}$, let $p(X)$ be its parent in the contraction tree T, and call *index* $i(X)$ of X the step of the contraction sequence at which $p(X)$ is created by merging X with some other part. Thus $p(X) \in \mathcal{P}_{i(X)}$, and $X \in \mathcal{P}_{i(X)+1}$.

Claim 2.20. For any $X \in \mathcal{P}$, there are at most $t+1$ parts $Y_1, \dots, Y_{t+1}$ with some edge between X and Y_ℓ in G and satisfying $i(Y_\ell) \leqslant i(X)$.

Proof. Fix $i = i(X)$, so that $p(X) \in \mathcal{P}_i$. By maximality of X, we know that $p(X)$ must contain an ω-clique. Then in $\mathcal{P}_i$, no normal edge can be incident to $p(X)$: this would create an $(\omega+1)$-clique in G. Since $\mathcal{P}_i$ has error degree at most t, this leaves at most t parts $Z_1, \dots, Z_t$ in $\mathcal{P}_i$ incident to $p(X)$ (by error edges).

Now consider some $Y_\ell \in \mathcal{P}$ and fix $j = i(Y_\ell) + 1$, so that $Y_\ell \in \mathcal{P}_j$. Suppose in a first time that $i(Y_\ell) < i$. This gives $i \geqslant j$, hence $\mathcal{P}_i$ refines $\mathcal{P}_j$. Since there exists an edge between Y_ℓ and X, it follows that Y_ℓ must contain some Z_m. Two different $Y_\ell, Y_{\ell'}$ are disjoint, hence cannot contain the same Z_m, leaving only t choices for Y_ℓ.

It only remains to consider the case $i(Y_\ell) = i(X)$. In that case $p(X) = p(Y_\ell)$, and X, Y_ℓ are the two parts merged to create $p(X)$; there can only be exactly be one part Y_ℓ satisfying this condition.

Thus at most $t+1$ parts adjacent to X have index at most $i(X)$: t with strictly smaller indices, and one with the same. ■

We now order the parts of $\mathcal{P}$ according to their indices, breaking the ties arbitrarily. By the claim, each part X is adjacent to at most $t+1$ parts before X in this order. Thus there is a $t+2$-colouring λ of $\mathcal{P}$: assuming that $\lambda(Y)$ is fixed for all $Y < X$, we choose $\lambda(X)$ to be any colour which is not used by any $Y < X$ adjacent to X.

Finally, using that for $X \in \mathcal{P}$ the graph $G[X]$ has no ω-clique, we can by induction hypothesis colour each $G[X]$ with $(t+2)^{\omega-2}$ colours. Call this colouring

$$\mu_X : V(G[X]) \to [(t+2)^{\omega-2}].$$

Then we colour G as follows: a vertex x contained in $X \in \mathcal{P}$ is given the colour

$$\nu(x) = (t+2)^{\omega-2}\lambda(X) + \mu_X(x).$$

Adjacent vertices in the same X are given distinct colours because μ_X is a proper colouring of $G[X]$, and adjacent vertices in different parts of $\mathcal{P}$ are given distinct colours thanks to λ. This is a $(t+2)^{\omega-1}$-colouring of G. □

We have thus proved that graphs with twin-width t and clique number ω are $(t+2)^{\omega-1}$-colourable. If the clique number ω is considered constant, this is a polynomial function of twin-width. On the other hand, if we fix a class with bounded twin-width, this proof only bounds the chromatic number by an exponential function of the clique number. Pilipczuk and Sokołowski proved that this bound can be improved to a quasi-polynomial function [79], and Bourneuf and Thomassé improved their technique to obtain a polynomial bound [23]. Thus any class $\mathcal{C}$ with bounded twin-width is *polynomially χ-bounded*: graphs $G \in \mathcal{C}$ satisfy $\chi(G) \leqslant \omega(G)^c$ for some constant c.

We will not present these results in further details, but the techniques developed in their proof will play a crucial role in Chapter 6.

2.4 Algorithmic application

Let us conclude this chapter with a second application of twin-width, an algorithm: we want to solve the *independent set* problem from the introduction on graphs of small twin-width. Later, Chapter 4 will discuss a far reaching generalisation of this algorithm to the *first-order model checking* problem.

2.4.1 Independent set. Recall that a set S of vertices is called *independent* or *stable* if there is no edge $xy \in E(G)$ with $x, y \in S$. The algorithmic problem of finding a largest independent set is notoriously simple to express, and hard to solve: it is NP-complete, and under standard complexity hypotheses, it admits no good approximation algorithm, and is not FPT with regards to the solution size. We will show that it becomes FPT when given a contraction sequence of bounded width.

Theorem 2.21 [19], [15, Theorem 9]. *There is an algorithm which, given a graph G, a contraction sequence for G of width t, and $k \in \mathbb{N}$, decides whether G contains an independent set of size k in time $f(k,t) \cdot n^{O(1)}$ for some computable function f.*

We first give a high level view of the proof of Theorem 2.21. It is a *dynamic programming* algorithm: using the contraction sequence, it defines a number of algorithmic subproblems (carefully chosen variants of the independent set problem), each of which is easily solved when given the solutions to a few simpler subproblems. Then, one can solve all of them, starting from the simplest and working our way up to increasingly more difficult ones while remembering all already found solutions. The last subproblem reached is the independent set problem itself. If the number of subproblems to solve is polynomial, and each of them takes polynomial time, this yields a polynomial algorithm.

What are these subproblems? In essence they are local variants of independent set, where the vertices which can be used are restricted. For instance, given some part P from the contraction sequence, one may ask for an independent set inside P, rather than in the entire graph. Now if P was obtained by merging P_1, P_2, we would like to quickly solve this problem on P, given the solutions for P_1, P_2. If P_1, P_2 are not connected by any edge, this is easy: the union of the best solutions for P_1 and P_2 will be the best for P. Similarly, if there are all edges between them, no independent set may intersect both P_1, P_2, so the best solution for P will be either the one from P_1, or the one from P_2. But if P_1, P_2 are in error there is no reasonable way to combine their respective solutions into an optimal one for P. The 'independent set restricted to a part' subproblem is not sufficiently general for this scheme to work.

The subproblem we actually consider is: given k parts $P_1, \dots, P_k$ from one of the partitions in the contraction sequence, is there an independent set of size k in $P_1 \cup \cdots \cup P_k$? We never need to consider more than k parts, since we are only looking for an independent set of size k. Solving this subproblem for all k-tuples of parts is not possible: the resulting complexity would be n^k, which is not FPT, and is the same as the trivial algorithm! The crucial insight is that it is sufficient to solve this for the tuples $P_1, \dots, P_k$ which are *connected* in the graph of error edges. Because the error degree is bounded, there are only linearly many such tuples for a constant k.

Let us now make these ideas precise.

Proof of Theorem 2.21. The *trace* of an independent set S on a partition $\mathcal{P}$ of $V(G)$ is the map

$$\begin{aligned} \mathrm{tr}_{\mathcal{P}}(S) : \mathcal{P} &\to \mathbb{N} \\ X &\mapsto |X \cap S| \end{aligned}$$

More generally, we call any map $f : \mathcal{P} \to \mathbb{N}$ a *potential trace* over $\mathcal{P}$, and say that f is *realisable* if there is an independent set S whose trace is f. Remark that when $\mathcal{P} = \{V(G)\}$ is the trivial partition, the trace $V(G) \mapsto k$ is realisable if and only if G contains an independent set of size k. Therefore testing whether a given potential trace is realisable generalises the independent set problem. Testing if a potential trace is realisable is the subproblem solved by the dynamic algorithm: given a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$, we compute which potential traces are realisable, first over $\mathcal{P}_n$, then $\mathcal{P}_{n-1}$, until $\mathcal{P}_1$, which answers the independent set problem.

Consider two consecutive partitions $\mathcal{P}_{i+1}, \mathcal{P}_i$ in the contraction sequence, where $\mathcal{P}_i$ is obtained by merging $X, Y \in \mathcal{P}_{i+1}$. Let $f : \mathcal{P}_i \to \mathbb{N}$ be a potential trace. We say that $g : \mathcal{P}_{i+1} \to \mathbb{N}$ *lifts to* f if $f(X \cup Y) = g(X) + g(Y)$, and $f(Z) = g(Z)$ for any $Z \in \mathcal{P}_{i+1} \setminus \{X, Y\}$. There are exactly $f(X \cup Y) + 1$ traces over $\mathcal{P}_{i+1}$ which lift to f, and f is realisable if and only if there exists a realisable trace which lifts to f. Thus an instance of the subproblem in $\mathcal{P}_i$ (testing if f is realisable) can be reduced to a few subproblems at the previous step $\mathcal{P}_{i+1}$ (testing if some g lifting to f is realisable), which is exactly what we need for a dynamic programming algorithm.

There are however n^k potential traces, hence we cannot afford to consider all of them. We will now show that only a small subset of them are required.

Define the *sum* of a potential trace f as $|f| = \sum_{X \in \mathcal{P}} f(X)$, and its *support* as $\mathrm{supp}(f) = \{X \in \mathcal{P} \mid f(X) > 0\}$. If $A \subset \mathcal{P}$, then the restriction $f_{|A}$ is the potential trace which coincides with f inside A, and is null outside A. Finally, we call *error graph* of the partition $\mathcal{P}$ the graph $Err(G, \mathcal{P})$ with vertices $\mathcal{P}$, and an edge XY whenever $X, Y \in \mathcal{P}$ are in error.

Claim 2.22. Let f be a potential trace over $\mathcal{P}$, and H the subgraph of $Err(G, \mathcal{P})$ induced by $\mathrm{supp}(f)$. Then f is realisable if and only if

1. no normal edge exists between vertices of $\mathrm{supp}(f)$ in $Tri(G, \mathcal{P})$, and
2. for any connected component C of H, the restriction $f_{|C}$ is realisable.

Proof. Firstly, the two conditions are necessary for f to be realisable: if XY is a normal edge in $Tri(G, \mathcal{P})$, then all vertices of X are adjacent to all vertices of Y, hence no independent set can intersect both X and Y; and if f is realisable, then so are all of its restrictions.

Let us now assume that the two conditions are satisfied. Let $C_1, \dots, C_k$ be the connected components of H, and S_i an independent set whose trace is $f_{|C_i}$. We claim that $S = \biguplus_{i \in [k]} S_i$ is an independent set, whose trace clearly is f. Consider $a, b \in S$. If a, b are both in the same S_i, then they are certainly non adjacent. If $a \in S_i$, $b \in S_j$ with $i \neq j$, call $A \in C_i$ (resp. $B \in C_j$) the part containing a (resp. b). By the first condition, there is no normal edge between A and B. Furthermore, since A and B are in distinct components of the error graph H, they are in particular non-adjacent. Thus there is neither a normal nor an error edge between A and B in $Tri(G, \mathcal{P})$, hence a and b are non-adjacent in G. This proves that S is independent. ■

Say that a potential trace f over $\mathcal{P}$ is *connected* if $\mathrm{supp}(f)$ is connected in $Err(G, \mathcal{P})$. Claim 2.22 implies that if we already know which connected potential traces over $\mathcal{P}$ are realisable, then we can test whether any given

potential trace over $\mathcal{P}$ is realisable in polynomial time. Using this remark, we can complete the dynamic programming algorithm.

Given a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ for G of width t, and the target size k for the independent set, we proceed as follows.

1. For each $\mathcal{P}_i$, enumerate the set $\mathcal{T}_i$ of potential traces over $\mathcal{P}_i$ which are connected and have sum at most k. Call $\mathcal{R}_i \subset \mathcal{T}_i$ the subset of realisable traces, which we will compute.

2. If $f \in \mathcal{T}_n$ is a connected trace over the partition into singletons $\mathcal{P}_n$, then the support of f must be a singleton $\{x\}$, since $\mathcal{P}_n$ induces no error edge. Thus testing if f is realisable, and computing $\mathcal{R}_n$ is trivial.

3. For i from $n-1$ to 1, we compute $\mathcal{R}_i$ as follows, assuming that $\mathcal{R}_{i+1}$ is already known: Given $f \in \mathcal{T}_i$, enumerate the at most $(k+1)$ traces $g_1, \dots, g_{k+1}$ over $\mathcal{P}_{i+1}$ which lift to f. Claim 2.22 and the knowledge of $\mathcal{R}_{i+1}$ allow to test whether each g_ℓ is realisable, and f is realisable if and only one of the g_ℓ is.

4. Finally, check whether the potential trace $V(G) \mapsto k$ in $\mathcal{T}_1$ is realisable.

The complexity of this algorithm is $(\sum_{i=1}^{n} |\mathcal{T}_i|) \cdot n^{O(1)}$. Thus, to conclude the proof, we only need to bound the size of the $\mathcal{T}_i$. First remark that at most k^k distinct potential traces can have the same support. Thus it suffice to bound the number of possible supports, i.e. of connected subgraphs of $Err(G, \mathcal{P}_i)$ on at most k vertices.

Claim 2.23. In a graph G on n vertices with maximum degree t, the number of subsets of at most k vertices inducing a connected subgraph is at most $n \cdot t^{2k-2}$.

Proof. If X induces a connected subgraph and $|X| \leqslant k$, then there is a walk of length $2k-2$ in G whose vertex set is exactly X: it can be obtained by walking along a spanning tree of $G[X]$. Thus the number of such subsets X is bounded by the number of walks of length $2k-2$, which is at most $n \cdot t^{2k-2}$. ■

It follows from the claim and previous remarks that $|\mathcal{T}_i| = (kt)^{O(k)} \cdot n$, and the algorithm runs in $(kt)^{O(k)} \cdot n^{O(1)}$. □

2.4.2 Computing twin-width. The algorithm of Theorem 2.21 makes one crucial assumption: the input graph is given together with a contraction sequence, witnessing its twin-width. Thus we cannot say that independent set is FPT parameterized by twin-width and the solution size: this would require an algorithm doing the same work as Theorem 2.21, but without having access to a contraction sequence in the input. This is a limitation of most currently known algorithms using twin-width ([55] is a notable exception).

It is therefore crucial to be able to compute good contraction sequences. This is a major open problem, arguably the most important one related to twin-width. This section is a short introduction to this problem—later chapters will discuss some special cases in more depth.

First the bad news: finding an optimal contraction sequence is NP-hard.

Theorem 2.24 (Bergé, Bonnet, Déprés [10]). *Given a graph G, it is* NP*-complete to test whether* $\mathrm{tww}(G) \leqslant 4$.

This rules out any hope of efficiently and *exactly* computing twin-width in the general case. But we need not find the exact value of twin-width: if given G with twin-width t, we can find a contraction sequence of width say $2t$, then algorithms such as Theorem 2.21 can still be applied to this contraction sequence, with complexity FPT in t. Thus approximation algorithm are what we really are looking for, and Theorem 2.24 does not preclude them.

We will be very generous regarding the meaning of approximation, for it would be unwise to be picky about it when no algorithm is currently known: the goal is an approximation up to any function of the optimum.

Question 2.25. Is there an algorithm which given a graph G with twin-width t, finds a contraction sequence of width $f(t)$ in time $g(t) \cdot n^{O(1)}$, for some computable functions f, g?

We call FPT *approximation* an algorithm which satisfies the constraints of Question 2.25. Remark that these constraints are precisely chosen to ensure that an FPT approximation of twin-width, combined with Theorem 2.21, gives an FPT algorithm for independent set parameterized by twin-width and the solution size.

As already stated, Question 2.25 is a major open problem. Chapter 5 will discuss some significant specific structures for which approximation algorithms for twin-width are known: ordered structures, permutations, and tournaments. Section 3.4.3 also gives some insight on the difficulties underlying Question 2.25: when considering only cubic graphs, we not only do not know how to approximate twin-width, but cannot even answer a much simpler question: explicitly construct cubic graphs with large twin-width.

Bibliographic notice

Twin-width as presented here, and the surrounding notions and terminology, were defined in 2020 by Bonnet, Kim, Thomassé, and Watrigant [19]. The underlying ideas however date back to 2014: they were developed by Guillemot and Marx for permutations under the simple name of 'width', with the goal of efficiently finding patterns in permutations [61].

The examples presented throughout section 2.2 come from the first work on twin-width of Bonnet et al. [19], and its followup with the author [14].

The dynamic programming scheme over contraction sequences used in Theorem 2.21 is one of the key ideas of Guillemot and Marx used for pattern recognition [61], and the generalisation to first-order model checking which will be explained in Chapter 4 is present in the initial work on twin-width [19]. This specific presentation of the independent set algorithm is based the third paper in the twin-width series with Bonnet et al. [15].

The colouring and χ-boundedness result in section 2.3 are also from [15], while the twincut graphs used as lower bound were introduced by Bonnet, Bourneuf, Duron, Thomassé, Trotignon, and the author in [11].

Chapter 3

GRIDS IN MATRICES

The previous chapter introduced twin-width through contraction sequences. We will now present a second, equally important characterisation, the *grid theorem for twin-width*: a graph G has bounded twin-width if and only if for some choice of ordering of $V(G)$, the adjacency matrix of G contains no large grid-like structure. This grid theorem relies on a major result of extremal combinatorics of Marcus and Tardos [73].

After introducing the Marcus–Tardos theorem, we will prove the grid theorem for twin-width, and demonstrate through examples how it can provide upper bounds on twin-width where explicitly constructing contraction sequences may be tedious.

Finally, we will use a variant of this grid theorem to construct *balanced* contraction sequences. Through compact encodings of graphs with small twin-width, these balanced contraction sequence lead to a crucial result: an upper bound on the number of graphs with bounded twin-width, which is exceptionally useful to prove that certain classes of graphs have unbounded twin-width, for instance graphs of bounded degree.

Preliminaries: matrices. Let us first fix a couple of notations and conventions regarding matrices which are used throughout this chapter.

The most important is that we see the ordering of rows and columns as an intrinsic part of a matrix, which plays an important role in the definitions of this chapter. In particular, when manipulating the adjacency matrix of a graph G, we will need to specify the ordering $<$ of $V(G)$ used for the rows and columns: this choice of ordering may significantly alter the properties of the matrix. We denote by $A(G, <)$ this adjacency matrix.

The intersection of a row and column of the matrix is an *entry*, which has a *value*. An entry with value v is called a v-entry. We will almost exclusively manipulate 0–1 matrices, i.e. the values are either 0 or 1. For such matrices, the *rank* is understood over the binary field $\mathbb{F}_2$.

3.1 Grids and sparsity

A *division* of a matrix M is a pair $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ of partitions of the rows and columns of M *into intervals*: each part $R \in \mathcal{R}$ is an interval in the ordered set of rows of M, and similarly with columns. If $|\mathcal{R}| = k$ and $|\mathcal{C}| = \ell$, we say that $\mathcal{D}$ is a $(k \times \ell)$-*division*, or simply a k-division when $k = \ell$. Parts $R \in \mathcal{R}$ or $C \in \mathcal{C}$ are called *blocks* of rows and columns respectively, and the submatrix $M_{|R \times C}$ induced by the intersection of R and C is called a *cell* of $\mathcal{D}$.

When $\mathcal{D}$ is a $(k \times l)$ division of a 0–1 matrix M, it is natural to define a *quotient* $M/\mathcal{D}$: it is a $(k \times l)$ 0–1 matrix whose rows and columns are $\mathcal{R}$ and $\mathcal{C}$ respectively, which inherit the order of the rows and columns of M. In $M/\mathcal{D}$, there is a '1' at the intersection of row $R \in \mathcal{R}$ and column $C \in \mathcal{C}$ if and only if the cell $M_{|R \times C}$ contains a '1'.

Finally, a *k-grid* in a 0–1 matrix M is a k-division $\mathcal{D}$ of M such that the quotient $M/\mathcal{D}$ consists only of '1's, i.e. every cell of $\mathcal{D}$ contains a '1'. See Figure 3.1 for an example. Having a k-grid in M for some large k means that M contains many '1' which in a sense are well distributed. We think of such a matrix as complex, and will be interested in matrices without grids of a given size-k.

0	1	0	0	0	0	1	1	0
0	0	1	1	0	0	1	0	1
0	0	0	0	1	0	0	0	0
1	0	0	0	1	1	0	1	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1
0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	0
0	1	0	1	0	1	0	0	0

Figure 3.1: Example of a 4-grid in a matrix: the lines indicate the division, every cell of which contains a '1'.

Before introducing the main theorem of this section, let us mention two easy and useful lemmas on grids and divisions.

Lemma 3.1. *If a submatrix of M contains a k-grid, then so does M.*

Proof. Let N be a submatrix of M and $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ a k-grid of N. The rows and columns of M missing in N can be added to existing parts of $\mathcal{R}$ and $\mathcal{C}$ respectively, while respecting the condition that $\mathcal{R}$ and $\mathcal{C}$ are partitions into intervals. The result is a k-grid in M. □

Lemma 3.2. *Let M be a 0–1 matrix, $\mathcal{D}_1$ a division of M, and $\mathcal{D}_2$ a division of $M/\mathcal{D}_1$. Then there is a third division $\mathcal{D}_3$ of M such that*

$$M/\mathcal{D}_3 = (M/\mathcal{D}_1)/\mathcal{D}_2.$$

In particular, if $M/\mathcal{D}_1$ has a k-grid, then so does M.

Proof. Let $\mathcal{D}_i = (\mathcal{R}_i, \mathcal{C}_i)$ be the partitions defining $\mathcal{D}_i$ for $i = 1, 2, 3$. A part $R_2 \in \mathcal{R}_2$ is a set of rows of $M/\mathcal{D}_1$, hence an element $R_1 \in R_2$ is itself a block in $\mathcal{R}_1$. Thus R_2 is a set of sets of rows of M. The division $\mathcal{D}_3$ is defined by flattening:

$$\mathcal{R}_3 = \left\{ \bigcup_{R_1 \in R_2} R_1 \mid R_2 \in \mathcal{R}_2 \right\}, \tag{3.1}$$

and similarly for columns. It is easy to verify that $\mathcal{D}_3$ satisfies the desired property. □

Let us now focus on matrices which *do not* admit any k-grid, which are called *k-grid free*. Intuitively, such matrices—if they are much larger than k–cannot contain too many '1'. This intuition is formalised by the following result, known as the Marcus–Tardos theorem, or formerly the Füredi–Hajnal conjecture: for fixed k, the maximum number of '1's in a k-grid free matrix M

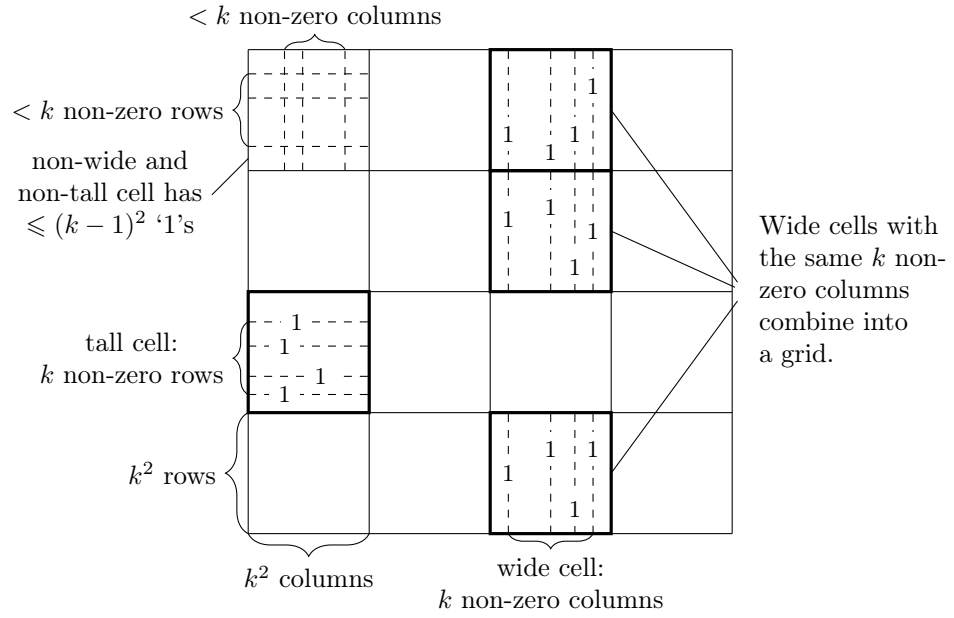


Figure 3.2: Types of cells in the proof of the Marcus Tardos theorem.

is linear in the size of M. This cornerstone result implies a famous conjecture of Stanley and Wilf—which we shall discuss later in section 3.4.3 and Chapter 6—and a major part of the theory of twin-width is built upon it. Its original proof, reproduced below, is beautifully simple.

Given a 0–1 matrix M, let $|M|$ denote the number of 1-entries in M. We are interested in the maximum number of '1's in a k-grid free matrix of a given size, i.e. the map

$$f(n,k) = \max\{|M| : M \text{ is a } k\text{-grid free } n \times n \text{ matrix}\}$$

Theorem 3.3 (Marcus–Tardos [73]). *For any $k \in \mathbb{N}$, there is a constant c_k such that*

$$f(n,k) \leqslant c_k \cdot n.$$

Proof. Let $k \in \mathbb{N}$ be fixed arbitrarily. We will prove a linear recurrence relation for $f(n,k)$. Without loss of generality, let us assume that n is a power of k^2, and consider M an $n \times n$ matrix without a k-grid.

Let $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ be the regular division of M into n/k^2 parts: each cell of $\mathcal{D}$ is a $k^2 \times k^2$ matrix. By Lemma 3.2, the quotient matrix $M/\mathcal{D}$ is k-grid free. Thus, $|M/\mathcal{D}| \leqslant f(n/k^2, k)$, meaning that at most $f(n/k^2, k)$ cells of $\mathcal{D}$ contain a '1'. Let us now split them in several types. A cell A of $\mathcal{D}$ is said to be *wide* if inside A, at least k distinct columns contain a '1'. Similarly, A is *tall* if at least k rows of A contain a '1'. See Figure 3.2 for an illustration. We will count the '1's contained in tall or wide cells separately from the rest.

Claim 3.4. A block $R \in \mathcal{R}$ of columns contains less than $k\binom{k^2}{k}$ wide cells.

Proof. Suppose for a contradiction that $R \in \mathcal{R}$ contains $k\binom{k^2}{k}$ wide cells. For each of these cells, pick k columns each containing a '1', among the k^2 columns of R. By pigeonhole principle, the same subset $R' \subset R$ of k columns will be chosen for at least k of these cells, say $A_1, \dots, A_k$. Now consider the submatrix formed by the columns R', and the rows intersecting one of $A_1, \dots, A_k$. This submatrix has a k-grid, with the following division: each column of R' is in its own block, and each cell A_i defines a block of rows. Thus there is a submatrix of M with a k-grid, a contradiction. ■

Claim 3.4 implies that there are at most $\frac{n}{k^2} \cdot k\binom{k^2}{k}$ wide cells in total, each of size $k^2 \times k^2$. Thus the total number of '1's in M contained in wide cells is at most

$$k^4 \cdot \frac{n}{k^2} \cdot k\binom{k^2}{k} = n \cdot k^3 \binom{k^2}{k}. \tag{3.2}$$

Naturally, the same bound applies to the '1's contained in tall cells. Now consider finally the cells which are neither tall nor wide. In such a cell, at most $(k-1)$ rows and as many columns are non-zero, hence the number of '1's is at most $(k-1)^2$. Further, recall from the beginning of the proof that there are at most $f(n/k^2, k)$ non-zero cells. It follows that the number of '1's in M contained in cells which are neither tall nor wide is at most

$$f\left(\frac{n}{k^2}, k\right) \cdot (k-1)^2. \tag{3.3}$$

Combining (3.2) and (3.3), we obtain

$$f(n,k) \leqslant f\left(\frac{n}{k^2}, k\right) \cdot (k-1)^2 + n \cdot 2k^3 \binom{k^2}{k}. \tag{3.4}$$

This last equation is of the form

$$f(n,k) \leqslant b_k \cdot f\left(\frac{n}{a_k}, k\right) + d_k \cdot n, \tag{3.5}$$

where a_k, b_k, d_k depend only of k, and $a_k > b_k$. Under such a recurrence equation, it is well known that $f(n,k) = O(n)$, for k fixed. □

The constant c_k given by Theorem 3.3 is called the *Marcus–Tardos constant*. The bound given by the previous proof, which we did not explicit, is $c_k = 2^{O(k \log k)}$. A refinement of this argument due to Fox shows that it can be improved to $c_k = 2^{O(k)}$:

Theorem 3.5 [44, Theorem 13]. *For any $k, n \in \mathbb{N}$, $f(n,k) \leqslant n \cdot 3k \cdot 2^{8k}$.*

For algorithmic purposes, let us finally remark that the proof of Theorem 3.3 is effective, and yields an FPT algorithm parameterized by k to find a k-grid in a sufficiently dense matrix.

3.2 A grid theorem for twin-width

Let us now introduce the main result of this chapter: graphs whose adjacency matrix contains no large grid have small twin-width. The idea originates from

the work of Guillemot and Marx [61], and was generalised in [19, 17]. We will see a number of variants of this result, starting with the simplest which applies to graphs of bounded degree.

Theorem 3.6. *Let G be a graph and $<$ an ordering of $V(G)$ such that the adjacency matrix $A(G,<)$ is k-grid free. Then* $\mathrm{tww}(G) \leqslant \max(2c_k, \Delta(G))$, *where $\Delta(G)$ is the maximum degree in G.*

Furthermore, there is a polynomial algorithm which given G and $<$ finds contraction sequence of width $\max(2c_k, \Delta(G))$.

Proof. Let $t = \max(2c_k, \Delta(G))$ be the bound on twin-width we are aiming for. We greedily construct a contraction sequence for G subject to the following restrictions:

- The sequence only consists of partitions of $V(G)$ into intervals of $<$, i.e. one only merges parts which are consecutive for $<$.
- All parts have degree at most t, counting both normal and error edges.

That is, we start with the partition $\mathcal{P}_n$ into singletons, and if $\mathcal{P}_i$ has been constructed, we choose $\mathcal{P}_{i-1}$ to be any partition obtained by merging two consecutive parts of $\mathcal{P}_i$ such that all parts in $Tri(G, \mathcal{P}_{i-1})$ have degree at most t (hence a fortiori *error* degree at most t). Notice that initially $\mathcal{P}_n$ satisfies the second condition because we chose $t \geqslant \Delta(G)$. This process can be implemented in polynomial time, and if it succeeds we obtain a contraction sequence of width at most t as desired.

Let us thus assume that the former process fails after reaching a partition $\mathcal{P}_i$. Denote by $P_1 < \cdots < P_r$ the parts of $\mathcal{P}_i$. Furthermore, call $M = A(G,<)$ the k-grid free adjacency matrix, and consider the division $\mathcal{D} = (\mathcal{P}_i, \mathcal{P}_i)$ of M. We will prove that the quotient $M/\mathcal{D}$ contains a k-grid, a contradiction.

Claim 3.7. In $M/\mathcal{D}$, for any $i \in [r-1]$, there are at least $2c_k + 1$ entries '1's contained in the two consecutive columns P_i and P_{i+1}.

Proof. By assumption, merging P_i and P_{i+1} is disallowed. Notice that merging P_i and P_{i+1} will never increase the degree of a third part—this is why we use total degree and not error degree.

Thus the reason merging P_i and P_{i+1} is disallowed must be that $P_i \cup P_{i+1}$ is adjacent to more than $t \geqslant 2c_k$ other parts of $\mathcal{P}_i$. Each of these parts gives a non-zero cell in the block P_i or P_{i+1} of $\mathcal{D}$. ■

By pairing consecutive columns in $M/\mathcal{D}$, it follows from this claim that

$$|M/\mathcal{D}| \geqslant \lfloor r/2 \rfloor \cdot (2c_k + 1). \tag{3.6}$$

Remark here that $r > 2c_k + 1$, for otherwise it would be impossible to have degree more than $2c_k$. Therefore,

$$\frac{r/2}{\lfloor r/2 \rfloor} \leqslant \frac{r}{r-1} < \frac{2c_k + 1}{2c_k}, \tag{3.7}$$

allowing to rewrite (3.6) as

$$|M/\mathcal{D}| \geqslant \lfloor r/2 \rfloor \cdot (2c_k + 1) > \frac{r}{2} \cdot 2c_k = rc_k. \tag{3.8}$$

Thus $M/\mathcal{D}$ is an $(r \times r)$-matrix containing more than rc_k '1's, hence it contains a k-grid by Theorem 3.3. This contradicts the hypothesis that M has no k-grid. □

Theorem 3.6 is helpful to obtain upper bounds on the twin-width of graphs: instead of directly constructing a contraction sequence, one describes a well chosen ordering of the vertices, for which the adjacency matrix has no large grid. This good ordering may be much simpler to describe than the contraction sequence. The following result is a beautiful example of this technique.

Recall from the Chapter 1 that a *minor* of G is a graph obtained by deleting vertices or edges, and by contracting edges. We claimed that for any fixed H, graphs G which *avoid* H as a minor have bounded twin-width. We will now prove it under simplifying assumptions: that G has bounded degree, and contains a Hamiltonian path (a path through all vertices without repetition). Remark that it is sufficient to prove the result when H is a biclique $K_{t,t}$, since any graph H on t vertices is a minor of $K_{t,t}$: if G avoids $K_{t,t}$ as minor, it also avoids H.

Theorem 3.8 [19, section 6.3]. *Let G be a graph with a Hamiltonian path and without $K_{t,t}$ as minor. Then*

$$\mathrm{tww}(G) \leqslant \max(2c_{2t}, \Delta(G)).$$

Proof. We wish to use Theorem 3.6, and thus need to find an ordering $<$ of $V(G)$ such that the adjacency matrix $A(G,<)$ is $2t$-grid free. Choose $<$ to be the order of vertices along some Hamiltonian path P of G. This ensures a crucial property: if $X \subset V(G)$ is an interval of $<$, then $G[X]$ is a connected subgraph, as witnessed by the subpath of P

Now suppose for a contradiction that $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ is a $2t$-grid in $A(G,<)$. Enumerate the blocks as $\mathcal{R} = \{R_1 < \cdots < R_{2t}\}$ and $\mathcal{C} = \{C_1 < \cdots < C_{2t}\}$. Remark now that either $R_t < C_{t+1}$, or $C_t < R_{t+1}$. Without loss of generality, let us assume the former case—otherwise the roles of $\mathcal{R}$ and $\mathcal{C}$ merely need to be swapped. We now contract each of the parts $R_1, \ldots, R_t, C_{t+1}, \ldots, C_{2t}$ (which are all disjoint) to a single vertex: this is possible thanks to the Hamiltonian path. Finally, we delete any other vertex, and any edge between parts R_i and R_j or between C_i and C_j. Since $\mathcal{D}$ is a grid, each cell $R_i \times C_j$ contains a '1', hence the parts R_i and C_j are adjacent in G. It follows that the minor of G obtained by the previous operations is $K_{t,t}$, a contradiction. □

Both the Hamiltonicity and the degree requirement in Theorem 3.8 can be removed. This was first proven in [19], using a specific depth-first search tree as replacement for the Hamiltonian path. This proof is however fairly technical, we will present a much simpler argument from [18] in section 3.3.1.

3.3 The rank perspective

Theorem 3.6 is only applicable to sparse graphs: dense graphs have large grids in their adjacency matrix regardless of the choice of ordering, as proved by the Marcus–Tardos theorem. Nonetheless, dense graphs may have small twin-width, cliques being an extreme example. To accommodate these dense graphs, this section presents a modified notion of grid, and the corresponding variant of Theorem 3.6 which characterises twin-width.

In grids, a cell is considered 'bad' if it contains a '1'. Cliques show that for twin-width, there is nothing bad in a cell consisting *only* of '1's. Thus we will change the notion of grid by requiring every cell to be complex in a stronger sense than just containing a '1'. It may seem natural to require every cell to be non-constant—this seems to match well with the errors contraction sequences—but this turns out insufficient. We will require an even stronger condition: a cell is 'bad' if its rank is large. Since we are working with 0–1 matrices, rank in this section is understood over the 2-element field $\mathbb{F}_2$—other reasonable notions would also work.

Precisely, in a matrix M, a d-division $\mathcal{D}$ is called a *rank-k d-division* if every cell of $\mathcal{D}$ is a submatrix of rank at least k. A rank-k k-division is abbreviated as *rank-k division*, and the *grid rank* of a matrix M is the largest k such that M admits a rank-k division. Remark that any rank-k division is a fortiori a grid: a cell of rank k in a 0–1 matrix must certainly contain a '1'.

Theorem 3.9 [17]. *For any graph G,*

1. *if* $\mathrm{tww}(G) = k$*, then there exists an ordering $<$ of $V(G)$ for which $A(G,<)$ has grid rank at most $2k+3$, and*
2. *for any ordering $<$ of $V(G)$, if $A(G,<)$ has grid rank at most k, then* $\mathrm{tww}(G) \leqslant f(k)$ *for some function $f(k) = 2^{2^{k^{O(1)}}}$, and a contraction sequence witnessing this can be computed in polynomial time.*

The first half of this theorem is the purpose of grid rank—the claim fails when replacing grid-rank with grids—while the second half is a natural extension of Theorem 3.6.

Instead of directly proving Theorem 3.9, we will introduce rank in the notion of contraction sequences too, then show that this *rank twin-width* is equivalent to the usual twin-width, and prove Theorem 3.9 with twin-width replaced by rank twin-width.

3.3.1 Error rank. In a matrix M, if R and C are subsets of rows and columns respectively, then $\mathrm{rk}_M(R;C)$ denotes the rank of the submatrix induced by $R \times C$. Similarly, if G is a graph and A, B are subsets of vertices, then $\mathrm{rk}_G(A;B)$ is the rank of the adjacency matrix of G restricted to rows in A and columns in B. We drop the matrix or graph in this notation when there is no ambiguity.

Consider a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ for a graph G. We wish to weaken the notion of width, introducing rank in it. A naive notion would be the following: two parts $A, B \in \mathcal{P}_i$ are in k-error if $\mathrm{rk}(A;B) > k$, and one asks for a contraction sequence in which each part is in k-error with at most k other parts. This fails horribly: for G an arbitrary graph on vertices $\{v_1, \dots, v_n\}$, consider the contraction sequence which regroups vertices one by one in a single large part, i.e. $\mathcal{P}_i$ consists of a part $\{i, \dots, n\}$, and singletons $\{1\}, \dots, \{i-1\}$. Since all but one part of $\mathcal{P}_i$ are singletons, the rank between two distinct parts never exceeds 1: there are no 2-errors in this contraction sequence. Remark however that the rank between $\{i, \dots, n\}$ and the *union* of the remaining parts is large, motivating the next definition.

Given a graph $G = (V, E)$, a partition $\mathcal{P}$ of V, and a part P, call *error rank* $\mathrm{erk}_{G,\mathcal{P}}(P)$ the smallest k such that there is a subset $\mathcal{Q} \subset \mathcal{P}$ of at most k

parts satisfying

$$\mathrm{rk}(P; V \setminus (\cup\mathcal{Q} \cup P)) \leqslant k,$$

where $\cup\mathcal{Q}$ denotes the union of parts in $\mathcal{Q}$. That is, ignoring k 'bad' parts $\mathcal{Q}$ chosen depending on P, the rank of P against all the rest of the graph (rather than individual other parts) is bounded by k. The *error rank* of the contraction sequence $\mathcal{P}_n, \ldots, \mathcal{P}_i$ naturally is the smallest k such that $\mathrm{erk}_{G,\mathcal{P}_i}(P) \leqslant k$ for any step i and part $P \in \mathcal{P}_i$.

It is useful to notice that merging parts $P_1, P_2 \in \mathcal{P}$ can never increase the error ranks of the remaining parts.

Lemma 3.10. *Let $\mathcal{P}'$ be a refinement of $\mathcal{P}$, and assume that P is a part in both $\mathcal{P}$ and $\mathcal{P}'$. Then*

$$\mathrm{erk}_{G,\mathcal{P}}(P) \leqslant \mathrm{erk}_{G,\mathcal{P}'}(P).$$

Proof. Let $k = \mathrm{erk}_{G,\mathcal{P}'}(P)$: there is a family $\mathcal{Q}' \subset \mathcal{P}'$ of k-parts such that $\mathrm{rk}(P; V(G) \setminus (\cup\mathcal{Q}' \cup P)) \leqslant k$. Since $\mathcal{P}'$ refines $\mathcal{P}$, each $Q' \in \mathcal{Q}'$ is contained in some part $Q \in \mathcal{P}$. Denote by $\mathcal{Q} \subset \mathcal{P}$ the family of at most k such parts Q, and remark that $\cup\mathcal{Q}' \subseteq \cup\mathcal{Q}$. It follows that $\mathrm{rk}(P; V(G) \setminus (\cup\mathcal{Q} \cup P)) \leqslant k$, hence $\mathrm{erk}_{G,\mathcal{P}}(P) \leqslant k$. □

For contraction sequences, error rank is less constraining than width.

Lemma 3.11. *A contraction sequence of width $k \geqslant 1$ has error rank at most k.*

Proof. If $\mathcal{P}_n, \ldots, \mathcal{P}_1$ has width k, then for any part $P \in \mathcal{P}_i$, ignoring the k parts in error with P, the adjacency matrix between P and the rest of the graph only contains columns full of '0' or full of '1'. This has rank 1. □

Conversely, we will show that any contraction sequence of error rank k can be converted into a new contraction sequence whose width is bounded by a function of k. The ideas underlying this result can be found in the proofs of [19, Theorem 5.4], [14, Lemma 3.8], and [17, Theorem 2].

Theorem 3.12. *Given a contraction sequence for G of error rank k, one can compute in polynomial time a contraction sequence of width $2^{O(k^4)}$.*

The idea behind Theorem 3.12 is that a partition $\mathcal{P}$ of bounded error rank can be refined into a partition of bounded error degree.

We will need a number of lemmas. The first shows that in a partition $\mathcal{P}$ of bounded error rank, when considering $P \in \mathcal{P}$, the choice of 'bad' parts to ignore can be fixed: they are the parts with high rank towards P. Crucially this choice is symmetrical: if Q is bad for P, then P is also bad for Q.

Lemma 3.13. *Consider $\mathcal{P}$ a partition of the vertices of $G = (V, E)$ with error rank at most k, and a part $P \in \mathcal{P}$. Define $\mathcal{Q}$ as the set of parts Q other than P satisfying $\mathrm{rk}(P, Q) > k$. Then $|\mathcal{Q}| \leqslant k$, and*

$$\mathrm{rk}(P; V \setminus (\cup\mathcal{Q} \cup P)) \leqslant k(k+1).$$

Proof. By assumption, there is a set of k parts $\mathcal{R} = \{R_1, \ldots, R_k\}$ such that $\mathrm{rk}(P; V \setminus (\cup\mathcal{R} \cup P)) \leqslant k$. Any part $Q \in \mathcal{Q}$ satisfies $\mathrm{rk}(P; Q) > k$, which immediately implies that Q is in $\mathcal{R}$: we must discard Q to find error rank less than k. Thus $\mathcal{Q} \subseteq \mathcal{R}$, proving that $|\mathcal{Q}| \leqslant k$.

Now consider the matrix with rows P and columns $V \setminus (\cup\mathcal{Q} \cup P)$. The columns are the ones of $V \setminus (\cup\mathcal{R} \cup P)$, plus the ones from R_i for each i satisfying $\mathrm{rk}(P; R_i) \leqslant k$. Thus this matrix consists of at most $k+1$ matrices each of rank at most k put side by side, hence its rank is at most $k(k+1)$ as desired. □

Using this lemma, we will next show how to refine a partition of error rank k into one of error degree $2^{O(k^2)}$. It is important that this refinement be sufficiently close to $\mathcal{P}$, in the following sense: recall that $\mathcal{P}'$ is called a refinement of $\mathcal{P}$ if each part of $\mathcal{P}$ is equal to some union of parts of $\mathcal{P}'$. We say that $\mathcal{P}'$ is furthermore a *k-refinement* of $\mathcal{P}$ (and $\mathcal{P}$ a *k-coarsening* of $\mathcal{P}'$) if each part of $\mathcal{P}$ is equal to the union of at most k parts of $\mathcal{P}'$. For example, in a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$, the partition $\mathcal{P}_i$ is a 2-refinement of $\mathcal{P}_{i+1}$.

Lemma 3.14. *Let $\mathcal{P}$ be a partition of the vertices of $G = (V, E)$ with error rank at most k. Then there exists a $2^{k(k+1)}$-refinement $R(\mathcal{P})$ of $\mathcal{P}$, computable in polynomial time, with error degree less than $(k+1) \cdot 2^{k(k+1)}$.*

Furthermore, if $\mathcal{P}'$ refines $\mathcal{P}$, both with error rank at most k, then $R(\mathcal{P}')$ refines $R(\mathcal{P})$.

Proof. Given a part $P \in \mathcal{P}$, denote by $\mathcal{Q}^P$ the set of parts Q^P other than P satisfying $\mathrm{rk}(P; Q^P) > k$. Let M_P denote the adjacency matrix of G, restricted to columns in P and rows in $V \setminus (\cup\mathcal{Q}_P \cup P)$. By Lemma 3.13, $\mathrm{rk}(M_P) \leqslant k(k+1)$. Since M_P is a 0–1 matrix with rank at most $k(k+1)$, it contains at most $2^{k(k+1)}$ distinct columns. Then, we repartition $P = \biguplus_{i \leqslant 2^{k(k+1)}} P'_i$ so that for each i, all columns of M_P in P'_i are equal. Finally, $R(\mathcal{P})$ is the partition

$$R(\mathcal{P}) = \left\{ P'_i \mid P \in \mathcal{P}, i \in [2^{k(k+1)}] \right\}.$$

Computing $R(\mathcal{P})$ in polynomial time is easy. Let us show that it has bounded error degree.

Consider two parts $A'_i, B'_j \in R(\mathcal{P})$ (i.e. $A, B \in \mathcal{P}$, $i, j \in [2^{k(k+1)}]$). Assuming that $A \neq B$ and $\mathrm{rk}(A; B) \leqslant k$, we claim that A'_i and B'_j are homogeneous. Indeed, consider the submatrix with columns A and rows B. This is both a submatrix of M_A and of the transpose of M_B. Thus by construction, all columns of A'_i are equal, and all rows of B'_j are equal, and the submatrix consisting of columns in A'_i and rows in B'_j is constant. Therefore A'_i and B'_j are in error only if $A = B$ or $\mathrm{rk}(A; B) > k$. For fixed A and i this leaves $k+1$ choices of B and $(k+1)2^{k(k+1)} - 1$ choices of B'_j, hence the bound on the error degree.

Let us finally prove that this construction is compatible with refinement. Let $\mathcal{P}'$, also of error rank k, be a refinement of $\mathcal{P}$. Given a part $P \in \mathcal{P}'$, let $\overline{P}$ denote the part of $\mathcal{P}$ containing P. If $P, Q \in \mathcal{P}'$ are such that $\mathrm{rk}(P, Q) > k$, then certainly $\mathrm{rk}(\overline{P}, \overline{Q}) > k$. It follows that the rows considered in the matrix $M_{\overline{P}}$ (defined relative to the partition $\mathcal{P}$) are a subset of the rows considered in M_P (defined relative to $\mathcal{P}'$). Thus, if columns $x, y \in P$ are equal in M_P, they a fortiori are equal in $M_{\overline{P}}$. This shows that $R(\mathcal{P}')$ refines $R(\mathcal{P})$. □

To prove Theorem 3.12, we will apply Lemma 3.14 to each partition of a contraction sequence of bounded error rank. This does not quite give a contraction sequence, but the next lemmas allow to fill the gaps.

Lemma 3.15. *In a graph G, let $\mathcal{P}$ be a partition with error degree k, and $\mathcal{P}'$ a t-refinement of $\mathcal{P}$. Then $\mathcal{P}'$ has error degree less than $(k+1)t$.*

Proof. Consider a part $P' \in \mathcal{P}'$, contained within $P \in \mathcal{P}$. Let $Q_1, \dots, Q_k$ be the parts in error with P in $\mathcal{P}$. Then any part in error with P' in $\mathcal{P}'$ is contained in one of $P, Q_1, \dots, Q_k$, each of which contains at most t parts of $\mathcal{P}'$. This leaves $(k+1)t$ choices for a part in error with P', minus P' itself. □

Given any two partitions $\mathcal{P}_m, \mathcal{P}_1$, we say that $\mathcal{P}_m, \dots, \mathcal{P}_1$ is a *partial contraction sequence* from $\mathcal{P}_m$ to $\mathcal{P}_1$ if $\mathcal{P}_i$ is obtained by merging two parts of $\mathcal{P}_{i+1}$. Naturally, the width of this partial contraction sequence is the maximum error degree of $\mathcal{P}_m, \dots, \mathcal{P}_1$. Remark that whenever $\mathcal{P}_m$ is a refinement of $\mathcal{P}_1$, there exists a partial contraction sequence from $\mathcal{P}_m$ to $\mathcal{P}_1$.

Lemma 3.16. *Let $\mathcal{P}_m, \dots, \mathcal{P}_1$ be a partial contraction sequence. If $\mathcal{P}_m$ is a t-refinement of $\mathcal{P}_1$ and $\mathcal{P}_1$ has error degree at most k, then this contraction sequence has width less than $(k+1)t$.*

Proof. Immediate by Lemma 3.15. □

Combining these lemmas proves the theorem.

Proof of Theorem 3.12. Consider $\mathcal{P}_n, \dots, \mathcal{P}_1$ a contraction sequence of error rank k for G. Define

$$t = 2^{k(k+1)} \qquad \text{and} \qquad k' = (k+1)2^{k(k+1)}.$$

By Lemma 3.14, there is a t-refinement $R(\mathcal{P}_i)$ of $\mathcal{P}_i$ with error degree at most k', and furthermore $R(\mathcal{P}_{i+1})$ refines $R(\mathcal{P}_i)$. Remark that $R(\mathcal{P}_n) = \mathcal{P}_n$ must the partition into singletons, and it is easy to check from its construction that $R(\mathcal{P}_1) = \mathcal{P}_1$ is the trivial partition.

To apply Lemma 3.16, we additionally need $R(\mathcal{P}_{i+1})$ to be a bounded refinement of $R(\mathcal{P}_i)$. To this end, remark first that $R(\mathcal{P}_{i+1})$ t-refines $\mathcal{P}_{i+1}$, which itself 2-refines $\mathcal{P}_i$. It follows that $R(\mathcal{P}_{i+1})$ $2t$-refines $\mathcal{P}_i$. Then note that whenever $\mathcal{A}$ refines $\mathcal{B}$ which itself refines $\mathcal{C}$, if $\mathcal{A}$ is a s-refinement of $\mathcal{C}$, then $\mathcal{A}$ also s-refines $\mathcal{B}$ (and $\mathcal{B}$ s-refines $\mathcal{C}$). Applied to $R(\mathcal{P}_{i+1}), R(\mathcal{P}_i), \mathcal{P}_i$, this gives that $R(\mathcal{P}_{i+1})$ $2t$-refines $R(\mathcal{P}_i)$.

Thus we have a sequence of partitions of error degree k', starting with singletons, ending with the trivial partition, and progressing by $2t$-refinements. By Lemma 3.16, this can be completed into a contraction sequence of width less than $2t(k'+1) = 2^{O(k^4)}$ by inserting any partial contraction sequence from $R(\mathcal{P}_{i+1})$ to $R(\mathcal{P}_i)$ for each i. Finally, since the partitions $R(\mathcal{P}_i)$ can be computed in polynomial time, so can the resulting contraction sequence. This concludes the proof of Theorem 3.12. □

The next sections will show that error rank is in a sense equivalent to grid rank, which together with Theorem 3.12 proves Theorem 3.9. Beforehand, let us present a simple application of error rank: as announced in the introduction and section 3.2, we will show that planar graphs, and more generally K_t-minor free graphs have bounded twin-width. This was first proved in [19, Theorem 17], but the proof presented here closely follows the technique of Bonnet, Kim, Reinald, and Thomassé [18, section 4].

Lemma 3.17. *Any non-trivial planar graph contains two vertices u, v in the same face, both of degree less than* 18*.*

Proof. Consider a planar graph $G = (V, E)$, and denote by X the set of vertices of degree at least 18. Remark that

$$2|E| = \sum_{v \in V} \deg(v) \geqslant \sum_{v \in X} \deg(v) \geqslant 18|X|. \tag{3.9}$$

Let $E(X)$ denote the subset of edges with both endpoints in X.

Now suppose for a contradiction that each face f of G is incident to at most one vertex outside of X. Then all but two of the edges incident to f are in $E(X)$. Since f is incident to at least 3 edges, at least a third of them are in $E(X)$. Summing over all faces, it follows that

$$|E(X)| \geqslant \frac{|E|}{3}. \tag{3.10}$$

Consider now the planar subgraph consisting of vertices X and edges $E(X)$. By (3.9) and (3.10) it satisfies

$$|E(X)| \geqslant 3|X|. \tag{3.11}$$

This contradicts the well known fact, derived from Euler's formula, that any planar graph satisfies $|E(X)| \leqslant 3|X| - 6$. □

Corollary 3.18 [19, 18]. *Planar graphs have bounded twin-width.*

Proof. Given a planar graph G, we construct a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ satisfying the following: any part $P \in \mathcal{P}_i$ either is a singleton, or is adjacent (by a normal or error edge) to at most 36 other parts of $\mathcal{P}_i$. This contraction sequence is simply obtained by iteratively contracting the two vertices given by Lemma 3.17. When a contraction creates the part P, its degree is at most 36, and contractions of other parts can never increase the degree of P. Thus the only parts which can have degree more than 36 are the ones which have yet to be involved in a contraction, i.e. singletons. Note that since the contracted vertices are in the same face, their contraction preserves planarity, allowing to apply Lemma 3.17 beyond the first step.

We claim that the error rank of this contraction sequence is at most 36. Indeed singletons always have error rank 1, and a part P with degree at most 36 also has error rank at most 36: we choose to ignore the parts adjacent to P, and what remains is a null matrix. Thus we obtain a bound on $\mathrm{tww}(G)$ by Theorem 3.12. □

The bound on the twin-width of planar graphs given by Corollary 3.18 is far from tight. Several bounds on their twin-width using a variety of techniques have been proved [19, 14, 18, 20, 65, 64], the last of which, due to Hliněný and Jedelský, shows that planar graphs have twin-width 8. This is almost matched by the construction of planar graphs with twin-width exactly 7 by Král and Lamaison [68].

In order to generalise this argument to K_t-minor free graphs, we will admit the following lemma.

Lemma 3.19 (Norine, Seymour, Thomas, Wollan [76]). *For any $t \in \mathbb{N}$, there exists $d \in \mathbb{N}$ such that any non-trivial K_t-minor free graph G contains two vertices x, y of degree at most d which are either adjacent or twins.*

Theorem 3.20 [19, 18]. *There is a function f such that any K_t-minor free graph G satisfies* $\mathrm{tww}(G) \leqslant f(t)$.

Proof. The vertices x, y to contract given by Lemma 3.19 either are twins, in which case contracting them is equivalent to deleting either one, or are adjacent. In either case, contracting them yields a minor of the original graph, hence the K_t-minor free hypothesis is preserved.

The rest of the proof is exactly the same as Corollary 3.18. □

3.3.2 From twin-width to grid rank: compatible orders. We now resume the proof of Theorem 3.9, and first prove that graphs with small twin-width have adjacency matrices with low grid rank. Given a contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ for a graph G, an ordering $<$ of $V(G)$ is said to be *compatible* with the contraction sequence if any part $X \in \mathcal{P}_i$ at any step of the contraction sequence is an interval of $(V(G), <)$. Equivalently, the ordering $<$ is compatible with the contraction sequence if $<$ is the left-to-right ordering of some plane embedding of the contraction tree. With this second definition, it is clear that any contraction sequence admits a compatible ordering.

Lemma 3.21. *Let $G = (V, E)$ be a graph and $<$ an ordering of V compatible with a contraction sequence of error rank k. Then $A(G, <)$ has grid rank at most $2k + 3$.*

Proof. Let $\mathcal{P}_n, \dots, \mathcal{P}_1$ be the given contraction sequence for G and $<$ the compatible ordering, and call $M = A(G, <)$. For a contradiction, assume that M contains a rank-$(2k + 4)$ division $\mathcal{D} = (\mathcal{R}, \mathcal{C})$.

Choose the earliest step in the sequence (i.e. the largest $i \in [n]$) such that some part of $\mathcal{P}_i$ contains as subset a part of either $\mathcal{R}$ or $\mathcal{C}$: say $R \in \mathcal{R}$ is a subset of $P \in \mathcal{P}_i$ (the situation would be symmetrical if $C \in \mathcal{C}$ was a subset of P). By maximality of i, P is the part of $\mathcal{P}_i$ obtained by merging two parts of $\mathcal{P}_{i+1}$, and no $Q \in \mathcal{P}_i \setminus \{P\}$ may contain a part of $\mathcal{R}$ or $\mathcal{C}$, while P itself may not contain more than one part of $\mathcal{C}$. Further, since all parts of $\mathcal{P}_i$ and $\mathcal{C}$ are intervals of $<$, any $Q \in \mathcal{P}_i \setminus \{P\}$ intersects at most two parts of $\mathcal{C}$, while P intersects at most three. See Figure 3.3 for an illustration.

Since $\mathcal{P}_i$ has error rank at most k, there is a set $\mathcal{Q} \subset \mathcal{P}_i$ of k parts such that $\mathrm{rk}(P; V \setminus (\cup\mathcal{Q} \cup P)) \leqslant k$. The k parts of $\mathcal{Q}$ plus P altogether intersect at most $2\ell + 3$ parts of $\mathcal{C}$. Since $\mathcal{D}$ is a division into $(2k + 4)$ parts, there remains some $C \in \mathcal{C}$ disjoint from $\cup\mathcal{Q} \cup P$, and since $\mathcal{D}$ is a rank-$(2k + 4)$ division, the cell $R \times C$ satisfies $\mathrm{rk}(R; C) \geqslant 2k + 4$. Since R is contained in P and C is disjoint from $\cup\mathcal{Q} \cup P$, it follows a fortiori that $\mathrm{rk}(P; V \setminus (\cup\mathcal{Q} \cup P)) \geqslant 2k + 4$, a contradiction. □

3.3.3 Grid rank theorem. We finally extend Theorem 3.6 to grid rank.

Theorem 3.22 (Theorem 3.9 restated). *Let G be a graph and $<$ an ordering of its vertices such that $A(G, <)$ has grid rank less than k. Then G admits a contraction sequence of error rank $2(k-1)c_{k^2} = 2^{O(k^2)}$ compatible with $<$, which can be computed in polynomial time given G, $<$, and k.*

The idea remains the same: the contraction sequence is created greedily, and if the process fail, then the Marcus–Tardos theorem yields a contradiction.

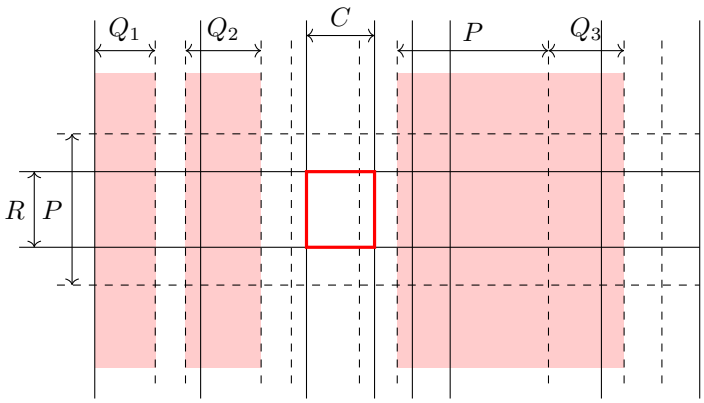


Figure 3.3: Illustration of the proof of Lemma 3.21. Two divisions of the same matrix are considered: the rank-$(2k+4)$ division $\mathcal{D}$ is drawn in solid lines, and $\mathcal{P}_i$ from the contraction sequence is drawn in dashed lines.
Parts of $\mathcal{P}_i$ non-homogeneous with P $(P, Q_1, \dots, Q_\ell)$ are filled in red. There are few of them, and each Q_i intersects at most two parts of $\mathcal{C}$, hence there remains $C \in \mathcal{C}$ disjoint from them. One reaches a contradiction by considering the rank of the cell $R \times C$ (highlighted with red border).

Let us establish a few notations before starting the proof itself. The grid rank k which we try to reach for a contradiction is fixed in all that follows. Consider a matrix M and a division $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ of M, with blocks of rows $\mathcal{R} = \{R_1 < \cdots < R_n\}$, and blocks of columns $\mathcal{C} = \{C_1 < \cdots < C_m\}$. We first say that a cell of $\mathcal{D}$ is *red* if its rank is at least k. Given a block of rows $R \in \mathcal{R}$, denote by $red(R) = \bigcup_{C \in \mathcal{C},\ \mathrm{rk}(R,C) \geqslant k} C$ the set of columns which intersect R in red cells.

Next, consider $R \in \mathcal{R}$ a block of rows and $C_i \in \mathcal{C}$ a block of columns. Define $C_{<i} = \bigcup_{j<i} C_j$ the set of columns located left of C_i, and $C_{\leqslant i} = C_{<i} \cup C_i$. We say that the cell $R \times C_i$ is *blue with regards to* R if

$$\mathrm{rk}(R; C_{<i} \setminus red(R)) < \mathrm{rk}(R; C_{\leqslant i} \setminus red(R)).$$

That is: restricting the matrix to rows in R, the cell $R \times C_i$ is *blue with regards to* R if one of its columns is linearly independent from all the columns before C_i which are not in a red cell. Being *blue with regards to a block of columns* C is defined symmetrically, and finally $R \times C$ is simply called *blue* if it is blue with regards to either R or C. The point is that several blue cells combine into a matrix of high rank.

Lemma 3.23. *Let M be a matrix and $\mathcal{D}$ a division of M containing k blue cells $Z_1, \dots, Z_k$, which belong to pairwise distinct blocks of rows and of columns. Then* $\mathrm{rk}(M) \geqslant k$.

Proof. Let $\mathcal{R} = \{R_1, \dots, R_s\}$ and $\mathcal{C} = \{C_1, \dots, C_t\}$ be the blocks of $\mathcal{D}$. We can delete any block which does not contain any Z_i (this does not increase the rank), and permute the order of blocks so that $Z_i = R_i \times C_i$ (this does not change the rank of M). Denote by M_i the submatrix with rows $R_1 \cup \cdots \cup R_i$ and columns $C_1 \cup \cdots \cup C_i$. If Z_i is blue with regards to R_i, then some column

of C_i is linearly independent from the columns in $C_1 \cup \cdots \cup C_{i-1}$, which implies $\mathrm{rk}(M_i) > \mathrm{rk}(M_{i-1})$. The reasoning is symmetrical when Z_i is blue with regards to C_i, and we find that $\mathrm{rk}(M_i) \geqslant i$ by induction. □

The important idea is that red and blue cells are the correct notions of 'bad' cells: we will next show that (1) if there are many red or blue cells in a matrix, its grid rank is large, and (2) if there are few of them in each block, then the error rank of the partition is small.

Lemma 3.24. *Let M be a matrix and $\mathcal{D}$ a m-division of M such that*

1. *either $\mathcal{D}$ contains $c_k \cdot m$ red zones,*
2. *or $\mathcal{D}$ contains $c_{k^2} \cdot m$ blue zones.*

Then M has grid rank at least k.

Proof. Consider the division $\mathcal{D} = (\mathcal{R}, \mathcal{C})$. In the first case, let $red(M, \mathcal{D})$ be the 0–1 matrix with rows $\mathcal{R}$ and columns $\mathcal{C}$, in which an entry is 1 if and only if the corresponding cell is red. By Theorem 3.3, $red(M, \mathcal{D})$ contains a k-grid, which lifts to a k-division $\mathcal{D}' = (\mathcal{R}', \mathcal{C}')$ of M satisfying the following: each cell of $\mathcal{D}'$ contains a red cell of $\mathcal{D}$. It follows that $\mathcal{D}'$ is a rank-k division.

For the second case, we similarly consider $blue(M, \mathcal{D})$ the 0–1 matrix indicating blue cells. There is a k^2-grid in $blue(M, \mathcal{D})$, lifting to a k^2-division $\mathcal{D}' = (\mathcal{R}', \mathcal{C}')$ of M such that each cell of $\mathcal{D}'$ contains a blue cell of $\mathcal{D}$. Now define the k-division $\mathcal{D}''$ by regrouping the blocks of $\mathcal{D}'$ by groups of k. Consider any cell N of $\mathcal{D}''$. Inside N, $\mathcal{D}'$ induces a k-division in which every cell contains a blue cell of $\mathcal{D}$. It follows from Lemma 3.23 that $\mathrm{rk}(N) \geqslant k$. Thus $\mathcal{D}''$ is a rank-k division. □

Lemma 3.25. *Consider a graph G with adjacency matrix $M = A(G, <)$. Let $\mathcal{P}$ be a partition of $V(G)$ into intervals of $<$, which is also seen as a division $(\mathcal{P}, \mathcal{P})$ of M. Consider a part $P \in \mathcal{P}$, seen as a block of rows. If P contains at most r_1 red cells of $\mathcal{P}$, and at most r_2 cells which are blue with regards to P, then* $\mathrm{erk}_{G,\mathcal{P}}(P) \leqslant \max(r_1, r_2(k-1))$.

Proof. Remove from the block of rows P the at most r_1 red cells, and consider the submatrix M' of M consisting of rows in P, and columns outside $red(P)$. Each cell inside M' has rank less than k (for otherwise it would be red). Furthermore, any column of M' which is not in a blue cell with regards to P is a linear combination of columns to its left. Therefore, the columns of M' are linearly generated by the at most r_2 blue cells, each of which has rank less than k. It follows that

$$\mathrm{erk}_{G,\mathcal{P}}(P) \leqslant \mathrm{rk}(M') \leqslant r_2(k-1).$$ □

We finally have all the tools to prove the main theorem of this section.

Proof of Theorem 3.22. Consider a graph G with vertices ordered by $<$ so that the adjacency matrix $M = A(G, <)$ has grid rank less than k. We greedily construct a contraction sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$ compatible with $<$ and subject to the following restriction: in $\mathcal{P}_i$, we may only merge parts P, P' if each of them contains at most $2c_k$ red cells and $2c_{k^2}$ blue cells.

Suppose in a first time that this process succeeds. Remark that testing if a cell is red or blue can be done in polynomial time, hence the contraction sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ can be computed in polynomial time.

Claim 3.26. The sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ has error rank at most $4(k-1)c_{k^2}$.

Proof. Call P, P' the two parts of $\mathcal{P}_i$ merged to obtain $\mathcal{P}_{i-1}$. By Lemma 3.10, the error rank of a part $Q \in \mathcal{P}_i \setminus \{P, P'\}$ does not increase when merging P and P', hence we only need to check the error rank of the new part $P \cup P'$.

By assumption, in $\mathcal{P}_i$ the part P has at most $2c_k$ red cells and at most $2c_{k^2}$ blue cells, hence by Lemma 3.25,

$$\mathrm{erk}_{G,\mathcal{P}_i}(P) \leqslant 2(k-1)c_{k^2}, \tag{3.12}$$

and similarly for P'. Furthermore, error rank is subadditive, hence

$$\mathrm{erk}_{G,\mathcal{P}_{i-1}}(P \cup P') \leqslant \mathrm{erk}_{G,\mathcal{P}_i}(P) + \mathrm{erk}_{G,\mathcal{P}_i}(P') \leqslant 4(k-1)c_{k^2}. \tag{3.13}$$

This proves the claim. ■

Suppose now that the construction fails: we reach a partition $\mathcal{P}_i$ such that among any two consecutive parts $P, P' \in \mathcal{P}_i$, one has more than $2c_k$ red cells, or more than $2c_{k^2}$ blue cells. Let $m = |\mathcal{P}_i|$ be the number of parts. By pairing the parts of $\mathcal{P}_i$, we find that the division $(\mathcal{P}_i, \mathcal{P}_i)$ contains at least $c_k \cdot m$ red cells, or at least $c_{k^2} \cdot m$ blue cells. It follows by Lemma 3.24 that M has grid rank at least k, a contradiction. □

Theorem 3.9 immediately follows from Theorems 3.12 and 3.22.

3.4 Versatile twin-width

In Theorems 3.6 and 3.9, we proved that when the adjacency matrix of G has no large grid (respectively no high rank division), then a greedy algorithm finds contraction sequences: one fixes a few easily checked constraints, and there will always be a choice of contraction satisfying these constraints. In this section, we show that by somewhat relaxing these constraints, one can actually find *linearly many* choices of contraction. This leads to notions of balanced contraction sequences, with several applications.

In a graph G, we say that a partition $\mathcal{P}$ of $V(G)$ admits an *ε-versatile contraction sequence of width t* if

1. either $\mathcal{P} = \{V(G)\}$ is the trivial partition,
2. or $\mathcal{P}$ has maximum error degree at most t, and for any subset $\mathcal{F} \subset \mathcal{P}$ of at most $\varepsilon|\mathcal{P}|$ *forbidden* parts, there is a pair $X, Y \in \mathcal{P} \setminus \mathcal{F}$ such that the partition obtained by merging X, Y recursively has an ε-versatile contraction sequence of width t.[1]

[1] The original definition of versatile twin-width in [14] asks for $\varepsilon|\mathcal{P}|$ different pairs which can be contracted. This is a weaker requirement than our definition, as it allows reusing a single part in all these pairs, which turns out problematic in some applications. Fortunately the issue is only in the definition: their proof in fact gives many *disjoint* pairs as noted in [15].

Finally we say that G has ε-versatile twin-width t if the partition into singletons on G has an ε-versatile contraction sequence of width t. We will prove that such versatile contraction sequences always exist, at the price of relaxing the bound on the twin-width.

Theorem 3.27 [14]. *For any $k \in \mathbb{N}$, there are $\varepsilon > 0$ and $t \in \mathbb{N}$ such that any graph with twin-width at most k also has ε-versatile twin-width t.*

The proof of Theorem 3.27 is a variant of Theorem 3.9. Once again, we first prove the result using *versatile error rank*, defined in the obvious way: width is replaced with error rank in the definition of versatile twin-width. There, we obtain a stronger result: the versatility parameter can be arbitrarily close to $\frac{1}{2}$.

Theorem 3.28. *For any $k \in \mathbb{N}$ and $\varepsilon > 0$, there exists $t' \in \mathbb{N}$ such that any graph G whose adjacency matrix has no rank-k division admits a $(\frac{1}{2} - \varepsilon)$-versatile contraction sequence of error rank t'.*

Proof. We simply replicate the proof of Theorem 3.9 with adjusted parameters. Let $<$ be an ordering of G such that $M = A(G, <)$ has grid rank less than k. Fix $t' = \frac{4}{\varepsilon}(k-1)c_{k^2}$.

Consider a partition $\mathcal{P} = \{P_1 < \cdots < P_m\}$ of $V(G)$ into m intervals of $<$, and with error rank at most t'. If $m \leqslant t'$, then all contractions will trivially preserve error rank at most t', thus let us assume $m > t'$. Choose $\mathcal{F} \subset \mathcal{P}$ a subset of at most $(\frac{1}{2} - \varepsilon)m$ forbidden parts.

Consider all disjoint the pairs (P_{2i-1}, P_{2i}) for $i \leqslant \lfloor \frac{m}{2} \rfloor$. Say that (P_{2i-1}, P_{2i}) is *good* if contracting it preserves error rank at most t', and *bad otherwise*. Thus we are looking for a good pair disjoint from $\mathcal{F}$. Assume for a contradiction that any pair disjoint from $\mathcal{F}$ is bad. Then among the $\lfloor \frac{m}{2} \rfloor$ previous disjoint pairs, the number of bad pairs is at least

$$\left\lfloor \frac{m}{2} \right\rfloor - \left(\frac{1}{2} - \varepsilon\right) m \geqslant \varepsilon m - \frac{1}{2} \geqslant \frac{\varepsilon m}{2}. \tag{3.14}$$

(The last step being a gross approximation using $m \geqslant t' \gg \varepsilon^{-1}$.)

If (P_{2i-1}, P_{2i}) is a bad pair, then either P_{2i-1} or P_{2i} has error rank more than $\frac{t'}{2} = \frac{2}{\varepsilon}(k-1)c_{k^2}$ in $\mathcal{P}$ (cf. proof of Theorem 3.9). It follows by Lemma 3.25 that P_{2i-1} or P_{2i} contains more than $\frac{2}{\varepsilon}c_k$ red cells, or more than $\frac{2}{\varepsilon}c_{k^2}$ blue cells. Summed over the $\frac{\varepsilon m}{2}$ bad pairs, this yield at least $c_k m$ red cells or $c_{k^2} m$ blue cells, and thus a rank-k division by Lemma 3.24, a contradiction. □

It remains to transform this versatile error rank into versatile twin-width: this is the equivalent of Theorem 3.12 in the versatile setting.

Lemma 3.29. *For any $k \in \mathbb{N}$, $\varepsilon > 0$, there are $k' \in \mathbb{N}$, $\varepsilon' > 0$ such that any graph with ε-versatile error rank k also has ε'-versatile twin-width k'.*

Proof. Consider a graph G with ε-versatile error rank k, and fix the parameters

$$\begin{aligned} t &= 2^{k(k+1)} \\ k' &= (k+1)2^{2k^2+2k+1} + 2^{k^2+k+1} = 2^{O(k^2)} \\ \varepsilon' &= \frac{\varepsilon}{2t} \end{aligned}$$

We will maintain the following objects and invariants:

1. a partition $\mathcal{Q}$ with ε-versatile error rank k,
2. the t-refinement $R(\mathcal{Q})$ with error degree $(k+1)t$ given by Lemma 3.14,
3. and a partition $\mathcal{P}$ which we want to prove has ε'-versatile twin-width k', such that $\mathcal{P}$ $2t$-refines $\mathcal{Q}$, and $\mathcal{P}$ refines $R(\mathcal{Q})$.

Notice that under these conditions, $\mathcal{P}$ also $2t$-refines $R(\mathcal{Q})$. It follows by Lemma 3.15 that the error degree of $\mathcal{P}$ is at most

$$((k+1)t+1)2t = (k+1)2^{2k^2+2k+1} + 2^{k^2+k+1} = k'. \tag{3.15}$$

Thus we need not worry about the error degree of $\mathcal{P}$ as long as the former invariants are maintained.

Initially, $\mathcal{Q} = R(\mathcal{Q}) = \mathcal{P}$ is the partition into singletons, which satisfies the constraints. Denote $m = |\mathcal{Q}|$, and $m' = |\mathcal{P}| \leqslant 2tm$. Say that a part $P \in R(\mathcal{Q})$ is *tight* if it also belongs to $\mathcal{P}$. If $P \in R(\mathcal{Q})$ is not tight, then it contains at least two parts P_1, P_2 of $\mathcal{P}$, and contracting P_1 with P_2 preserves the invariants. Thus if $R(\mathcal{Q})$ contains at least $\varepsilon' m'$ non-tight parts, we get as many disjoint pairs of parts of $\mathcal{P}$ which can be contracted. This gives the ε'-versatile choice of contraction for $\mathcal{P}$ which we are looking for.

Suppose now that the number of non-tight parts in $R(\mathcal{Q})$ is at most

$$\varepsilon' m' \leqslant \frac{\varepsilon}{2t} 2tm = \varepsilon m. \tag{3.16}$$

In this case, we will perform well-chosen contractions in $\mathcal{Q}$ to create new non-tight parts. Notice that there is no versatility requirement here: we are constructing a versatile contraction sequence for $\mathcal{P}$, and $\mathcal{Q}$ is not part of this contraction sequence, it is merely an object used to guide the construction.

By extension, we call $Q \in \mathcal{Q}$ tight if all parts of $R(\mathcal{Q})$ contained in Q are tight. Thus there are no more than εm non-tight parts in $\mathcal{Q}$. Since $\mathcal{Q}$ has ε-versatile error rank t, there is a contraction of two tight parts $Q_1, Q_2 \in \mathcal{Q}$ preserving this versatile error rank. We claim that this contraction preserves our invariants. Call $\mathcal{Q}'$ the partition obtained by contracting Q_1, Q_2.

- The contraction of Q_1, Q_2 preserves ε-versatile error rank t for $\mathcal{Q}$ by assumption.
- Each of Q_1, Q_2 contains at most t parts of $R(\mathcal{Q})$, all of which are tight. Thus Q_1, Q_2 each contain at most t parts of $\mathcal{P}$, and $Q_1 \cup Q_2$ contains at most $2t$. The remaining parts of $\mathcal{Q}$ each contain at most $2t$ parts of $\mathcal{P}$ by the invariant. Therefore $\mathcal{P}$ $2t$-refines $\mathcal{Q}'$.
- By the second half of Lemma 3.14, $R(\mathcal{Q}')$ is a coarsening of $R(\mathcal{Q})$. Since $\mathcal{P}$ refines $R(\mathcal{Q})$, it also refines $R(\mathcal{Q}')$.

We can thus alternate between contractions in $\mathcal{P}$ and $\mathcal{Q}$ while preserving the invariants, and leaving at least $\varepsilon'|\mathcal{P}|$ disjoint choices whenever a contraction in $\mathcal{P}$ happens. This process continues until $\mathcal{Q}$ becomes the trivial partition, at which point it is trivial to complete the versatile contraction sequence for $\mathcal{P}$. □

Theorem 3.27 follows directly from Theorem 3.28 and Lemma 3.29. The next sections will present some applications of this result, which rely on versatility to keep some object balanced.

3.4.1 Balanced partitions and integrality gap. Using versatile twin-width and avoiding contractions with the largest parts, one can stop any part from becoming much larger than the average. This yields balanced partitions with bounded error degree. We will show a weighted version of this result, with an application to *dominating sets*.

Call a (vertex) weight function on a graph G any function $w : V(G) \to \mathbb{R}^+$. Weights extend to subsets by sum: for $X \subseteq V(G)$, $w(X) = \sum_{x \in X} w(x)$.

Lemma 3.30 [15, section 5]. *For any k, there exists constants c, k' such that given a graph $G = (V, E)$ with twin-width k, a weight function $w : V \to \mathbb{R}^+$ with total weight $W = w(V)$, and $0 < \eta \leqslant W$, one can find a partition $\mathcal{P}$ of V such that*

1. *$\mathcal{P}$ has error degree at most k',*
2. *$\mathcal{P}$ consists of at most $\frac{cW}{\eta}$ parts, and*
3. *each part $P \in \mathcal{P}$ either is a singleton, or has weight $w(P) < \eta$.*

Proof. Let ε, k' be the versatile twin-width parameters given by Theorem 3.27, and fix $c = \frac{2}{\varepsilon}$. We follow an ε-versatile contraction sequence of width k' for G, allowing only contractions using parts of weight less than $\frac{\eta}{2}$. Let $\mathcal{P}$ be the partition reached when this process blocks. Clearly its error degree is at most k', and if $P \in \mathcal{P}$ has weight more than η, it must be that P was never involved in a contraction, i.e. P is a singleton. Let us prove that $|\mathcal{P}| \leqslant \frac{cW}{\eta}$.

Since the parts we forbid to use in contractions are those of weight at least $\frac{\eta}{2}$, there can be no more than $\frac{2W}{\eta}$ forbidden parts. Thus, for these to block all contractions of the ε-versatile contraction sequence, it must be that $\frac{2W}{\eta} \geqslant \varepsilon|\mathcal{P}|$, hence $|\mathcal{P}| \leqslant \frac{2W}{\eta\varepsilon} = \frac{cW}{\eta}$. □

In a graph G, a set $S \subseteq V(G)$ of vertices is a *dominating set* if any vertex either is in S or has a neighbour in S. That is, writing $N[v] = \{v\} \cup N(v)$ for the closed neighbourhood, for any vertex v, S intersects $N[v]$. Finding the minimum size of a dominating set in G, denoted $\gamma(G)$ is a well-known NP-hard problem. Like independent set, it is also hard to approximate and admits no FPT algorithm for the class of all graphs, but has an FPT algorithm when given contraction sequences of bounded width [15, section 4]. This is a variant of Theorem 2.21, and a special case of the first-order model checking algorithm which will be presented in Chapter 4.

For now, we are interested in approximations of dominating sets. One can define the *fractional relaxation* of the problem: a *fractional dominating set* is a weight function $w : V(G) \to [0, 1]$ satisfying $w(N[v]) \geqslant 1$ for any vertex v. Notice that if w is restricted to only take integral values $\{0, 1\}$, then we recover the usual notion of dominating set, with S being the set of vertices of weight 1. The minimum total weight of a fractional dominating set in G is denoted $\gamma^*(G)$. Notice that $\gamma^*(G) \leqslant \gamma(G)$. Unlike $\gamma(G)$ which is hard to compute, $\gamma^*(G)$ can be computed in polynomial time: this is because it can be expressed as a linear optimisation problem, a general class of problems solved by polynomial time algorithm using the *ellipsoid* or *interior point* methods.

Unfortunately, computing $\gamma^*(G)$ does not in general help to compute or even approximate $\gamma(G)$, as the ratio $\gamma(G)/\gamma^*(G)$—known as *integrality gap*—can be arbitrarily large. When considering graphs of bounded twin-width on the other hand, this integrality gap is bounded.

Theorem 3.31 [15, Theorem 13]. *For any k there is a constant ρ such that any graph with twin-width k and fractional domination number γ^* admits a dominating set of size at most $\rho\gamma^*$.*

Proof. Let c, k' be the balanced partition parameters given by Lemma 3.30, and fix $\eta = \frac{1}{k'+1}$ and $\rho = \frac{c}{\eta}$. Fix G of twin-width t with a fractional dominating set $w : V(G) \to [0,1]$ of total cost γ^*. We use this fractional dominating set w as our weight function, and apply Lemma 3.30 to obtain a partition $\mathcal{P}$ of vertices into at most $\frac{c\gamma^*}{\eta} = \rho\gamma^*$ parts of weight less than η each, except possibly for singletons.

Now construct $S \subset V(G)$ by picking one arbitrary vertex in each part of $\mathcal{P}$. Thus $|S| \leqslant \rho\gamma^*$ as desired. We claim that S is a dominating set. Consider any vertex $v \in V(G)$ contained in a part $P \in \mathcal{P}$, and denote by $Q_1, \dots, Q_\ell \in \mathcal{P}$ the parts other than P containing neighbours of v. Then,

- If P is a singleton, i.e. $P = \{v\}$, then v is in S, hence is dominated.
- Similarly, if $Q_i = \{u\}$ is a singleton, then u is in S and by hypothesis is a neighbour of v, which is thus dominated.
- We can thus assume that none of $P, Q_1, \dots, Q_\ell$ are a singleton, hence each of these parts has weight less than $\frac{1}{k'+1}$. On the other hand $N[v]$ has weight at least 1 since w is a fractional dominating set, and is contained in $P \cup Q_1 \cup \dots Q_\ell$. It follows that $\ell > k'$. We now use that $\mathcal{P}$ has error degree at most k': the $\ell > k'$ parts $Q_1, \dots, Q_\ell$ are adjacent to P because they contain neighbours of v, and one of them, say Q_i, must be connected to P by a normal edge in $Tri(G, \mathcal{P})$. Then the vertex $u \in Q_i \cap S$ is a neighbour of v (in fact of all of P), which is thus dominated. □

Since there are methods to efficiently compute fractional dominating sets, Theorem 3.31 can be turned into a polynomial time ρ-approximation algorithm for dominating sets in graphs of twin-width k.

3.4.2 Compact representations. This section introduces a notion of *parallelized* contractions, by allowing simultaneous contraction of several disjoint pairs of parts. Using versatile twin-width, one can construct parallelized contraction sequences of logarithmic length, which leads to some compact encodings of graphs of bounded twin-width.

A *parallel contraction sequence* for a graph G is a sequence of partitions $\mathcal{P}_m, \dots, \mathcal{P}_1$ of $V(G)$ such that $\mathcal{P}_m$ is the partition into singletons, $\mathcal{P}_1 = \{V(G)\}$ is the trivial partition, and $\mathcal{P}_{i+1}$ is a 2-refinement of $\mathcal{P}_i$ (i.e. each part of $\mathcal{P}_i$ either was already in $\mathcal{P}_{i+1}$, or is obtained by merging two parts of $\mathcal{P}_{i+1}$). The width of the sequence is the maximum error degree of the partitions. It follows from Lemma 3.16 that a parallel contraction sequence of width k can be turned into a normal contraction sequence of width $2k+1$.

The point of parallelisation is that the number m of steps can be much less than n: using versatile twin-width, it can be decreased to $O(\log n)$.

Lemma 3.32. *For any $k \in \mathbb{N}$, there are constants $\delta > 0$ and $k' \in \mathbb{N}$ such that any graph of twin-width k admits a parallel contraction sequence $\mathcal{P}_m, \dots, \mathcal{P}_1$ of width k' satisfying $|\mathcal{P}_i| \leqslant (1-\delta)|\mathcal{P}_{i+1}|$. In particular, the length m is at most $\frac{\log n}{\log(1-\delta)}$.*

Proof. Let ε, k' be the versatile twin-width parameters given by Theorem 3.27, and fix $\delta = \frac{\varepsilon}{1+\varepsilon}$.

We only consider contractions which preserve the existence of an ε-versatile contraction sequence of width k', which we simply call *good contractions*. Suppose that $\mathcal{P}_{i+1}$ of size r has already been constructed. Perform any good contraction of parts $X_1, Y_1 \in \mathcal{P}_{i+1}$, then X_2, Y_2 disjoint from X_1, Y_1, and so on until we have contracted disjoint pairs $(X_1, Y_1), \dots, (X_k, Y_k)$, and no remaining good contraction is disjoint from them. The partition reached at this point is $\mathcal{P}_i$, and we only need to prove that $k \geqslant \delta r$.

In $\mathcal{P}_i$, no good contraction remains when forbidding use of the k parts $(X_1 \cup Y_1), \dots, (X_k \cup Y_k)$. Since $|\mathcal{P}_i| = r - k$, this means that $k \geqslant \varepsilon(r-k)$, which implies $k \geqslant \frac{\varepsilon}{1+\varepsilon} r = \delta r$. □

We will now describe an encoding of graphs using parallel contractions sequences. The general idea, given the contraction sequence $\mathcal{P}_m, \dots, \mathcal{P}_1$, is to reconstruct $Tri(\mathcal{P}_{i+1})$ from $Tri(\mathcal{P}_i)$ by adding constant information to each part of $\mathcal{P}_i$.

For the purpose of encoding a trigraph T, it is convenient to assume that the error edges incident to any vertex v have some fixed arbitrary numbering $1, \dots, \ell$ (ℓ being the error degree). This numbering need not be symmetric: an error edge uv might be the ith edge from u's point of view, and the jth for v. An encoding of T is expected to contain this numbering.

Now consider a graph G and partitions $\mathcal{P}, \mathcal{P}'$ of its vertices with error degree at most k and such that $\mathcal{P}'$ 2-refines $\mathcal{P}$. Each part $P \in \mathcal{P}$ either is the union of two parts $P_1, P_2 \in \mathcal{P}'$, or is itself in $\mathcal{P}'$, in which case we say that $P_1 = P$ and P_2 is undefined. Given an encoding of $Tri(G, \mathcal{P})$, we construct an encoding of $Tri(G, \mathcal{P}')$ by adding the following information for each part $P \in \mathcal{P}$:

- whether P is obtained by merging two parts of $\mathcal{P}'$, i.e. whether P_2 exists,
- if P_2 exists, the nature of the edge P_1P_2 in $Tri(G, \mathcal{P}')$ (none, normal, or error, and its number relative to P_1 and P_2 in the latter case),
- and, if the ith error edge from P connects it to Q, then the nature of the edge P_aQ_b for each combination of $a, b \in \{1, 2\}$ (when the parts exist).

Given parts $P_i, Q_j \in \mathcal{P}'$, it is simple to determine the type of edge between them (none, normal, or error, and the numbering for the latter) from the type of edge between the respective parents $P, Q \in \mathcal{P}$, and the former information. For each part P, the first item takes 1 bit, the second $\log k + 1$, and the last $4k(\log k + 1)$ (up to $4k$ edges requiring $\log k + 1$ bits each). Thus the information added for each part $P \in \mathcal{P}$ is a function of k only.

Applying this method iteratively to all partitions of a parallel contraction sequence obtained by Lemma 3.32 gives an encoding of graphs of bounded twin-width. We will present a few different flavours of this, with the detailed encoding optimized for various goals.

Fact 3.33. *In a RAM model, graphs of twin-width k for fixed k can be encoded in $O_k(n)$ words allowing edge queries in time $O_k(\log n)$.*

Proof. Given a graph G with twin-width k, consider a parallel contraction sequence $\mathcal{P}_m, \dots, \mathcal{P}_1$ given by Lemma 3.32. We represent it as a binary tree,

where each level corresponds to a partition $\mathcal{P}_i$, and each part is the union of its children. The condition $|\mathcal{P}_i| \leqslant (1-\delta)|\mathcal{P}_{i+1}|$ from Lemma 3.32 ($\delta > 0$ function of k) then ensures that this tree is of linear size.

We now simply add the previously described information to each node of the tree to obtain a linear encoding of G. Given two vertices x, y, one can test if xy is an edge by only looking up the branches of the encoding leading to the leaves $\{x\}$ and $\{y\}$, with a constant time processing at each level. Since the encoding has depth $O_k(\log n)$, this gives the desired complexity. □

To test whether xy is an edge in this encoding, only the informations on the branches leading to x and y respectively is needed. In this sense, the encoding is local. Let us formalize this idea. An $f(n)$-bits *adjacency labelling schemes* for a class of graphs $\mathcal{C}$ consists of a labelling $\lambda : V(G) \to \{0,1\}^{f(n)}$ for each graph $G \in \mathcal{C}$ on n vertices, along with a decoding function shared by all graphs of $\mathcal{C}$ which given $\lambda(x), \lambda(y)$ indicates whether or not x, y are adjacent.

For example, graphs with maximum degree d admit a simple $(d+1)\log n$ adjacency labelling scheme: the label of v lists the number of v and of each of its neighbours, using $\log n$ bits each. Trees are known to have a $(\log n + O(1))$-bits adjacency labelling scheme [6], while planar graphs have a $(1+o(1))\log n$-bits labelling scheme [40].

In the case of twin-width, by reusing the proof of Fact 3.33 and labelling each vertex with the information found on the corresponding branch, we obtain the following.

Theorem 3.34 [14, Theorem 2.8]. *Graphs of twin-width k admit an $O_k(\log n)$-bits adjacency labelling scheme.*

In a different direction, one may aim to optimize the size of the encoding at the cost of decoding time. This leads to bounds on the number of graphs with small twin-width, which will be presented in the next section. Fact 3.33 uses $O_k(n)$ words in a RAM model, but in this model each word implicitly uses $\log n$ bits. This overhead is due to the pointer from each node of the encoding tree to its parent, but this can be removed if the decoding time is not a concern.

Fact 3.35. *Graphs of twin-width k can be encoded using $O_k(n)$ bits.*

Proof. Consider a parallel contraction sequence given by Lemma 3.32 and represent it as a tree T of linear size, with a constant size label attached to each node of T. We now enumerate the nodes of T in breath first order: level by level, then left to right. The encoding of the graph is the concatenation of the labels of all nodes along this order, which takes linear size.

The only difficulty in decoding is to find the parent or children of a given node of T. Suppose we know a node t of T is the ith node of its level ℓ. The $i-1$ nodes preceding t on level ℓ are placed just before t in the encoding. By reading the label of each of them to check whether it has one or two children, one can deduce the position of the children of t at level $\ell+1$. Applied iteratively from the root, this reconstructs the parent–children relationship from the encoding. The remainder of the decoding process is the same as in Fact 3.33. □

While the previous three encodings have the advantage of being relatively simple applications of versatile twin-width, they are far from optimal. Im-

proving upon Facts 3.33 and 3.35, the following arguably is the current most interesting encoding of graphs with bounded twin-width:

Theorem 3.36 (Pilipczuk, Sokołowski, Zych-Pawlewicz [80]). *Graphs with twin-width k and n vertices can be encoded using $O_k(n)$ bits while allowing edge queries in time $O_k(\log \log n)$.*

3.4.3 Small classes. In the previous section, we proved that graphs of bounded twin-width can be encoded using space linear in the number of vertices. This immediately yields an upper bound on the number of such graphs. We will see that this seemingly simple result is a very powerful tool to prove graphs have unbounded twin-width. Let us first give some terminology and context.

There are two common ways to count the number of graphs in a class $\mathcal{C}$:

1. *counting up to isomorphism*, i.e. counting the number of isomorphism classes of graphs in $\mathcal{C}$ on n vertices, and
2. *counting up to equality*, where one counts the number of graphs in $\mathcal{C}$ with vertex set $\{1, \dots, n\}$. This is also known as counting labelled graphs: one thinks of the vertices as being labelled from 1 to n, and two graphs are considered equal if both the labels of vertices and the edges between them match.

Counting up to equality is usually easier, and thus more common in enumerative combinatorics. While the two ways of counting give different result, the ratio is at most $n!$. Precisely, if the class $\mathcal{C}$ contains s graphs on n vertices when counted up to isomorphism, and t when counted up to equality, then we have $s \leqslant t \leqslant sn!$. Indeed, for any graph G on n vertices considered up to isomorphism, there are at most $n!$ distinct ways to label $V(G)$ from 1 to n, but potentially fewer if G has automorphisms.

A class $\mathcal{C}$ is called *small* if $\mathcal{C}$ contains at most $c^n \cdot n!$ graphs on n vertices counted up to equality, for some constant c. It is called *tiny* it contains at most c^n graphs on n vertices counted up to isomorphism, for some constant c. The former remark shows that any tiny class is also small.

For example, the number of trees with vertices $\{1, \dots, n\}$ is n^{n-2}, as first proved by Cayley [28], hence the class of trees is small. When counting up to isomorphisms, the number of trees is easily bounded by Catalan numbers, themselves bounded by 4^n, hence trees are also a tiny class. As a significant generalisation, Norine, Seymour, Thomas, and Wollan proved that any class of graphs avoiding a minor (e.g. planar graphs) is small [76].

The Stanley–Wilf conjecture, proved by Marcus and Tardos as corollary of Theorem 3.3 (cf. [67]), states that any class $\mathcal{C}$ of permutations avoiding a pattern has at most c^n permutations on n elements for some constant c, i.e. is tiny for the obvious generalisation of the notion to permutations. Although we have defined neither patterns nor twin-width for permutations, it is crucial to mention that for permutations, avoiding a pattern is equivalent to having bounded twin-width, as noticed by Guillemot and Marx [61]. We will present this result, and more generally twin-width of permutations in details in Chapter 6.

Notice that all the former examples of small or tiny classes have bounded twin-width. Generalising all these results, we obtain the following as immediate corollary of Fact 3.35.

Theorem 3.37 [14]. *Any class of graphs with bounded twin-width is tiny (and a fortiori small).*

Proof. If $\mathcal{C}$ is a class with bounded twin-width, graphs in $\mathcal{C}$ with n vertices can be encoded using cn bits for some constant c, hence there are at most $(2^c)^n$ of them up to isomorphism. □

Theorem 3.37 proves remarkably useful to show the existence of graphs with unbounded twin-width: if one can show that a class $\mathcal{C}$ is not small, then $\mathcal{C}$ must contain graphs with arbitrarily large twin-width. The prime example of this technique is graphs with bounded degree. Call a graph subcubic if all its vertices have degree at most 3.

Lemma 3.38. *The number of subcubic graphs with vertex set $\{1, \dots, n\}$ is at least*

$$2^{\frac{3}{2}n \log n - O(n)}.$$

Proof. For n even, we construct a family of bipartite subcubic graphs G with vertex set $\{1, \dots, n\}$ as follows: fix $A = \{1, \dots, \frac{n}{2}\}$ and $B = \{\frac{n}{2} + 1, \dots, n\}$, and pick three arbitrary perfect matchings E_1, E_2, E_3 between A and B, whose union in the edge set of G.

There are $(n/2)!$ choices for each E_i independently. Several of these choices can however lead to the same graph G. For a fixed G with no more than $\frac{3n}{2}$ edges, there are at most $7^{3n/2} \leqslant 19^n$ ways to write $E(G) = E_1 \cup E_2 \cup E_3$. Thus the number of distinct graphs constructed by this method is at least

$$\frac{(\frac{n}{2}!)^{\frac{3}{2}}}{19^n} = 2^{\frac{3}{2}n \log n - O(n)}. \tag{3.17}$$

□

A small class contains no more than $n! c^n = 2^{n \log n + O(n)}$ graphs on the vertex set $\{1, \dots, n\}$. This is asymptotically much smaller than the lower bound given by Lemma 3.38. Thus the class of cubic graphs is not small, and we obtain from Theorem 3.37:

Corollary 3.39 [14]. *There are subcubic graphs with arbitrarily large twin-width.*

We can in fact be more precise: Lemma 3.38 shows that for any fixed t the number of subcubic graphs is asymptotically much larger than the number of graphs with twin-width at most t. Thus, if G is drawn uniformly at random among subcubic graphs on n vertices, then the probability that $\operatorname{tww}(G) \leqslant t$ tends to 0 as n increases. Therefore, if one fixes a sequence of sizes x_n which tends to infinity, and randomly draws G_n subcubic of size x_n, then the graphs $\{G_n \mid n \in \mathbb{N}\}$ have unbounded twin-width with probability 1. Hendrey, Norin, Steiner, and Turcotte significantly improved this result by showing in [63] that almost all subcubic graphs on n vertices have twin-width $n^{1/4+o(1)}$.

Unfortunately, these enumerative or probabilistic constructions currently are the only known proofs of Corollary 3.39: although almost all graphs with bounded degree have large twin-width, we are unable to explicitly describe a sequence of subcubic graphs with unbounded twin-width—a problem humorously called *finding hay in a haystack*. This partly explains the difficulty behind finding efficient approximations of twin-width (Question 2.25): although not a formal argument by any stretch, it is reasonable to expect any approximation

algorithm for twin-width to result from some classification, or at least deep understanding of graphs with large twin-width. However, when restricted to cubic graphs, we are unable to exhibit a single example, let alone classify them.

Chapter 7 will further focus on bounded degree graphs: while leaving the former question open, it will show that twin-width satisfies some interesting geometric properties on bounded degree graphs, which do not hold in general, and lead to the study of twin-width of groups. It will also prove that the converse of Theorem 3.37 is false, constructing a tiny class with unbounded twin-width, answering a question asked alongside the original proof of Theorem 3.37 in [14].

Bibliographic notice

The Marcus–Tardos theorem was proved in 2004 [73], and the bounds on its constant were significantly improved by Fox ten years later [44]. The theorem was conjectured by Füredi and Hajnal in 1992 [46], and Klazar observed in 2000 [67] that it implied a famous conjecture formulated independently by Stanley and Wilf in the 1980s.

Applying the Marcus–Tardos theorem to contraction sequences and the resulting greedy algorithm of Theorem 3.6 are again key ideas of Guillemot and Marx [61]. A first characterisation of twin-width with a variant of grids was presented in the initial work on twin-width of Bonnet, Kim, Thomassé and Watrigant [19]. It involves *mixed minors*, a notion in-between grids and rank divisions, which we will use in Chapter 6. Its proof is somewhat tedious, as complications arise from errors occurring at the frontier of two cells. This is why we elect to present the similar result on grid rank of Bonnet, Giocanti, Ossona de Mendez, Simon, Thomassé, and Toruńczyk [17], which is the source of most of section 3.3.

The idea of error rank of contraction sequence as the natural equivalent of grid rank of matrices has appeared implicitly several times, but has not so far be formalised to our best knowledge. The idea notably appears in the proofs of Theorem 4.16 in [19, 49]. It also draws inspiration from the *oriented twin-width* of Bonnet, Kim, Reinald, and Thomassé [18].

Versatile twin-width and its applications (small classes, graph encodings, integrality gap) are results from the second and third papers on twin-width with Bonnet et al. [14, 15]. The presentation of Theorem 3.37 as a corollary of encoding results is original; its first proof in [14] was a variant of the result of Norine, Seymour, Thomas, and Wollan on classes avoiding a minor [76].

Chapter 4

FIRST-ORDER LOGIC

This chapter presents, without proofs, two major results relating twin-width and first-order logic from [19]: a *model checking* algorithm, generalising the independent set algorithm of Theorem 2.21, and a stability result under *transductions*, i.e. transformations described using first-order logic.

This short chapter is primarily meant for readers with little to no familiarity with logic. To the seasoned logician interested in the proofs of these results and some extensions, we recommend the work of Gajarský, Pilipczuk, Przybyszewski, and Toruńczyk [49].

We will start with an informal introduction of first-order logic on graphs, which presents and motivates the main notions, algorithmic problems, and questions considered. This will be followed by the precise definitions of these notions and some important known results regarding them, and finally the two announced results.

4.1 An informal introduction

We are interested in first-order formulæ over graphs: they are constructed by quantifying on vertices, testing whether vertices are adjacent, and combining these tests through the usual boolean operators $\vee$ (or), $\wedge$ (and), $\neg$ (not). For example, the formula

$$\phi \;=\; \exists x_1, \dots, \exists x_k, \forall y, \bigvee_{i=1}^{k} y = x_i \vee E(y, x_i) \tag{4.1}$$

expresses the existence of a dominating set of size k: there is a subset of k vertices $D = \{x_1, \dots, x_k\}$ such that any vertex y is either in D or adjacent to it (adjacency being denoted by $E(y, x_i)$ in the formula). Given a graph G, the fact that G *satisfies* the formula ϕ (in this case, meaning that G has a dominating set of size k) is denoted by $G \models \phi$. The point of this section is to introduce some of the most common problems and questions considered in this context of logic on graphs.

Algorithmic problems. There are two algorithmic problems usually associated with any logic. The first is satisfiability:

Problem 4.1 (Satisfiability)**.** Given a first-order formula ϕ, is there a graph G such that $G \models \phi$?

This is none other than Hilbert's famous *Entscheidungsproblem*: is a given formula ϕ always true? Of course, Hilbert problem was meant for general mathematical logic, whereas we only consider finite graphs. This is no less difficult: satisfiability of first-order logic on graphs is well known to be undecidable. To make the problem easier, one may restrict it by requiring the graph G to come

from a fixed class $\mathcal{C}$: this is called the satisfiability problem in $\mathcal{C}$. Unfortunately, satisfiability is still undecidable even when restricted to relatively simple classes of graphs. Indeed, one can show the following by reduction from the halting problem.

Theorem 4.2 (Folklore). *Let $\mathcal{C}$ be any class of graphs containing at least the planar graphs. Then the satisfiability problem in $\mathcal{C}$ is undecidable.*

In fact, far less than all planar graphs is needed: grids with small paths pending from each vertex are sufficient. The latter have twin-width at most 5, and thus proving interesting results on the satisfiability problem using twin-width seems hopeless.

Instead, we will focus the second most classical algorithmic question for any logic: model checking.

Problem 4.3 (Model Checking)**.** Given a graph G and a first-order formula ϕ, test whether $G \models \phi$.

Note the difference: for model checking, the graph G is given as input, for satisfiability, one asks if it exists. Since finite graphs are considered, model checking is decidable by a brute force enumeration. This brute force algorithm in fact shows that first-order model checking is in the complexity class PSPACE (problems solved using polynomial space, and up to exponential sat), while an easy reduction from quantified SAT shows that it is also PSPACE-hard. But should the reader write this reduction for themselves, they may notice an amusing fact: when fixing G to be the graph with two vertices and no edge, the model checking problem remains PSPACE-hard! The complexity of this problem comes entirely from the formula, while the graph is largely irrelevant.

For this reason, model checking is almost exclusively considered from the point of view of parameterised complexity: we will ask whether model checking is fixed parameter tractable (FPT), with the formula ϕ as parameter, that is whether there is an algorithm running in time $f(\phi) \cdot n^{O(1)}$, for some (computable) function f from FO formulæ to $\mathbb{N}$. As will be explained later in this chapter, it can reasonably be assumed that such an algorithm does not exist in general. However, FPT algorithms exist when the input graphs are restricted to specific classes: historically, the first example of such a result comes from graphs of bounded degree:

Theorem 4.4 (Seese [85]). *There is an algorithm which tests whether $G \models \phi$ in time $f(\phi, \Delta(G)) \cdot n$ for some function f.*

Expressiveness. The second kind of questions we consider bears on the expressiveness of first-order logic: can a given property $\mathcal{P}$ be defined using a first-order formula? To continue our introductory example, for any fixed k, the existence of a k-dominating set is expressible in first-order logic: there is a formula ϕ_k such that a graph G has a k-dominating set if and only if $G \models \phi_k$.

On the other hand, first-order logic cannot express that a graph is connected. This is a folklore result proved using so called *Ehrenfreucht–Fraïssé* games: one can show that for any formula ϕ, there exists a length ℓ such that

$$C_\ell \models \phi \quad \iff \quad (C_\ell \uplus C_\ell) \models \phi,$$

where $C_\ell \uplus C_\ell$ denotes the union of two disjoint copies of the cycle C_ℓ. Since C_ℓ is connected and $C_\ell \uplus C_\ell$ is not, it follows that there does not exist a formula ϕ which is satisfied exactly by connected graphs.

The question of expressiveness is not limited to boolean properties: one can for instance ask if a graph G can be encoded in another graph H through a first-order formula. For example, given a graph G, the *square* G^2 is the graph with the same vertices as G, and such that x, y are adjacent in G^2 if and only if their distance in G is at most 2. Being at distance at most 2 can be expressed in first-order logic as follows:

$$\phi(x,y) = E(x,y) \ \vee \ (\exists z, \ E(x,z) \wedge E(z,y)) \,, \tag{4.2}$$

that is: x, y are at distance at most 2 if they are adjacent, or have a common neighbour z. Thus the edge set of the square G^2 is characterised by a formula ϕ, applied to G:

$$xy \in E(G^2) \quad \iff \quad G \models \phi(x,y).$$

This kind of construction is called a *first-order interpretation*.

In this context, we are most interested in the following question: given a class $\mathcal{C}$ of graphs, is there a first-order interpretation which allows to construct all graphs starting from $\mathcal{C}$? When that is the case, $\mathcal{C}$ is called *independent*.

For example, consider the 1-subdivision $G^{(1)}$ of the graph G: each edge is replaced by a path of length 2. Then it is easy to verify that G is an induced subgraph of the square of $G^{(1)}$. Thus, if $\mathcal{C} = \left\{G^{(1)} \mid G \text{ is a graph}\right\}$ denotes the class of all subdivided graphs, the 'square' FO interpretation applied to $\mathcal{C}$ yields all graphs as induced subgraphs, which implies that $\mathcal{C}$ is independent.

By contrast, if $\mathcal{D}$ is the class of subcubic graphs (graphs with maximum degree 3) one may show that the classes obtained by interpretation from $\mathcal{D}$ are very restricted, and in particular, the class of all graphs cannot be reached. Thus the class of subcubic graphs is *dependent*, which is also called *NIP* (standing for 'non independence property').

These two aspects of first-order logic on graphs are related by the following conjecture of Gajarský, Hliněný, Obdržálek, Lokshtanov, and Ramanujan.

Conjecture 4.5 [48, Conjecture 8.2]. *A hereditary class $\mathcal{C}$ has an* FPT *first-order model checking algorithm if and only if it is NIP.*

4.2 Definitions

Let us now formalise the ideas presented in the introduction of this chapter.

4.2.1 Relational structures. Before even discussing any logical notion, we will generalise the objects to which they are applied: A graph (V, E) is a set V of vertices equipped with *one* relation E, which is *binary*, *symmetric*, and *irreflexive*. From the logic point of view, there is no reason to keep any of these restrictions. Thus, we define a *relational structure* $(V, E_1, \dots, E_k)$ as a set of vertices equipped with several relations $E_1, \dots, E_k$, each with a fixed but arbitrary arity, and with no symmetry or similar requirement.

It is useful to specify the number of relations which a structure should have, and their arities. This is the purpose of a *relational signature* Σ: a

set $\{R_1, \ldots, R_k\}$ of *relation symbols*, each with an associated arity $\mathrm{ar}(R_i) \in \mathbb{N}$. Relational structure can now be properly defined: a *relational structure* S *over the signature* Σ, or simply Σ-*structure*, consists of:

- a set $V(S)$ called the *domain*, *universe*, or by analogy with graphs, *vertex set*[1] of S, and
- for each symbol $R \in \Sigma$ of arity $r = \mathrm{ar}(R)$, a *valuation* as a relation $R(S) \subseteq V(S)^r$.

For example:

- Graphs are structures over the signature $\{E\}$, with $\mathrm{ar}(E) = 2$.
- In Chapter 5, we will work with *ordered graphs*: they are structures over the signature $\{<, E\}$, both of arity 2, where $<$ is interpreted as a linear order on the vertices and E as a set of edges.
- In Chapter 6, we will represent permutations as the superposition of two linear orders $<_1, <_2$ on the same set, i.e. as structures over $\{<_1, <_2\}$, with $\mathrm{ar}(<_i) = 2$.
- Figure 4.1 gives an example of a relational structure with a relation of arity 3.

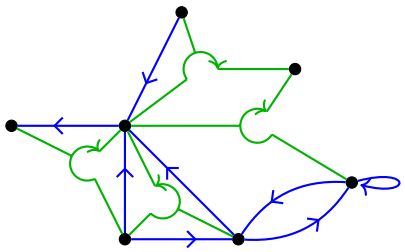


Figure 4.1: A relational structure S over the signature $\{E, F\}$ with $\mathrm{ar}(E) = 2$, $\mathrm{ar}(F) = 3$. The relation E is represented by blue edges, which are directed since $E(S)$ is not required to be symmetric. Similarly, F is represented by green hyperedges, which are ordered 3-tuples.

Given a Σ-structure S and a subset $X \subset V(S)$ of vertices, the *induced substructure* $S[X]$ is the Σ-structure with vertices X such that $R(S[X])$ consists of all tuples of $R(S)$ contained within X for each symbol $R \in \Sigma$.

We will often consider *classes* of relational structures. A class $\mathcal{C}$ of Σ-structures is a collection of Σ-structures closed under isomorphism. The class is *hereditary* if it is closed under taking induced substructures. When considering a class $\mathcal{C}$, it is implicit that all structures of $\mathcal{C}$ have the same signature, which we also call the signature of the class.

[1] The notations $V(S)$ and $R(S)$ for the domain and the valuation of relations, blatantly taken from graph theory, are not standard in model theory.

4.2.2 First-order formulæ. Given a signature Σ, one can define the *first-order formulæ over* Σ, i.e. the formulæ which can meaningfully be applied to Σ-structures. For each relation $R \in \Sigma$ of arity r, there is a *relation predicate* $R(x_1, \ldots, x_k)$, denoting that the k-tuple of vertices represented by the variables $x_1, \ldots, x_k$ belongs to the valuation of R. Additionally, the *equality predicate* $x = y$ checks that two variables denote equal vertices. These relation and equality predicate are also called *atomic formulæ*. They can then be combined with the usual boolean operators $\vee, \wedge, \neg$, and the quantifiers $\exists$ and $\forall$, which quantify on the vertices of the structure.

A first-order formula ϕ may have free variables, i.e. variables which appear without any binding quantifier, and which are usually written as explicit parameters, e.g. $\phi(x, y)$. A formula without free variables is said to be *closed*, and is also called a *sentence*. Using examples from the introduction of this chapter, the formula (4.1), expressing the existence of a dominating set, is closed: all appearing variables have a corresponding quantifier. On the other hand, formula (4.2), expressing that two vertices are at distance at most 2, has two free variables standing for the vertices whose distance is to be tested.

If $\phi(x_1, \ldots, x_k)$ is a formula over the signature Σ, S is a Σ-structure, and $v_1, \ldots, v_k$ are vertices of S, then $S \models \phi(v_1, \ldots, v_k)$ denotes that S satisfies the formula ϕ, when the free variables $x_1, \ldots, x_k$ are interpreted as the respective vertices $v_1, \ldots, v_k$.

4.2.3 Model-Checking. We are interested in the following algorithmic problem.

Problem 4.3 (Model Checking)**.** Given a graph G and a first-order formula ϕ, test whether $G \models \phi$.

Specifically, we are interested in the complexity of this problem parameterised by the size of the formula ϕ. This is a hard problem.

Theorem 4.6 (Downey, Fellows, and Taylor, [39]). *First-order model checking on graphs is* $\mathsf{AW}[*]$*-complete.*

We refer the reader to [39] for the definition of the complexity class $\mathsf{AW}[*]$; we will not use the definition, and only manipulate the class $\mathsf{AW}[*]$ using Theorem 4.6 (which for our purpose could be taken as a definition). What matters is that the complexity class $\mathsf{AW}[*]$ is located fairly high in the parameterised complexity hierarchy; it contains the hierarchy of classes $\mathsf{W}[k]$, and in particular is harder that the independent set problem. It is thus reasonable to assume that $\mathsf{AW}[*]$-complete problems do not admit FPT algorithms: the converse would be a significant collapse in the parameterised complexity hierarchy.

Assumption 4.7 ($\mathsf{FPT} \neq \mathsf{AW}[*]$). *First-order model checking on graphs does not admit an* FPT *algorithm.*

Assuming thus that there is no FPT algorithm for first-order model checking in general, it becomes meaningful to enquire about FPT algorithms for model checking restricted to specific classes.

This line of research started by Seese who proved in 1996 that graphs with bounded degree admit such algorithms [85] (see Theorem 4.4). Five years later, similar result were obtained by Frick and Grohe for planar graphs [45], and more

generally for graphs avoiding a minor by Flum and Grohe [43]. These results were unified under the theory of *sparsity* of Nešetřil and Ossona de Mendez [75], as they were generalised to classes of bounded expansion by Dvořák, Král, and Thomas [41], and finally to nowhere dense classes by Grohe, Kreutzer, and Siebertz [54]. Conversely, in any monotone (i.e. closed under subgraphs) class $\mathcal{C}$ which is *not* nowhere dense, Kreutzer and Dawar remarked that first-order model checking is $\mathsf{AW}[*]$-complete [69], showing that the last result is in a sense optimal: for monotone classes, nowhere density characterises efficient model checking.

Outside of this sparsity setting, surprisingly little was known until recently: the main result was that classes of bounded clique-width have FPT model checking not just for first-order, but also for the much more expressive *monadic second-order logic* [34]. This is disappointing: one would expect the weaker first-order logic to be efficiently checkable on more general classes of graphs. Outside the realm of graphs, an interesting previously known result for dense relational structures is an FPT algorithm for model checking in posets of bounded width [47].

The algorithm using twin-width of [19] which we will present at the end of this chapter was the first major advance in first-order model checking for dense graphs beyond clique-width. Note here that both graphs with bounded clique-width [19] and posets of bounded width [19, 8] have bounded twin-width.

For now, let us continue our introduction of logical notions with interpretations, which are most useful when trying to prove hardness results for model checking.

4.2.4 Interpretations, Transductions. Given relational signatures Σ, Γ, a *first-order interpretation* Φ from Σ to Γ consists of

1. a *domain formula* $\phi_V(x)$ with one free variable, and
2. for each $R \in \Gamma$, a formula $\phi_R(x_1, \ldots, x_r)$ with $r = \mathrm{ar}(R)$ free variables.

Given a Σ-structure S, its image $\Phi(S)$ is a Γ-structure, where

1. the domain consists of vertices of S satisfying the domain formula, i.e. $V(\Phi(S)) = \{x \in V(S) \mid S \models \phi_V(x)\}$, and
2. each relation $R \in \Gamma$ of arity $r = \mathrm{ar}(R)$ is interpreted as the set of tuples satisfying the corresponding formula, i.e.

$$R(\Phi(S)) = \{(x_1, \ldots, x_r) \in V(\Phi(S))^r \mid S \models \phi_R(x_1, \ldots, x_r)\}.$$

A fundamental property of interpretations is *backwards translation*: any first-order property of the image $\Phi(S)$ can be described on S itself.

Lemma 4.8 (Backwards translation). *Given Φ an interpretation from Σ to Γ and ψ a formula over Γ, there exists a formula $\psi \circ \Phi$ over Σ such that for any Σ-structure S,*

$$S \models \psi \circ \Phi \quad \iff \quad \Phi(S) \models \psi.$$

Sketch of proof. The formula $\psi \circ \Phi$ is obtained from ψ by (1) replacing each occurrence of a symbol $R \in \Gamma$ by the corresponding formula ϕ_R from Φ, and (2) restricting quantifiers to vertices satisfying the domain formula $\phi_V(x)$. □

This allows to compose interpretations.

Lemma 4.9. *If Φ and Θ are interpretations from Σ to Γ and from Γ to Δ respectively, then the function $\Theta \circ \Phi$ is an interpretation from Σ to Δ.*

Sketch of proof. Apply backwards translation through Φ to the domain and relation formulæ which define Θ. □

Transductions generalise interpretations by introducing a *non-deterministic colouring* operation. Given a set $\mathcal{U}$ of unary relation symbols (i.e. symbols of arity 1), $\mathcal{U}$-colouring is the operation which extends a Σ-structure S into a $(\Sigma \cup \mathcal{U})$-structure by choosing *non deterministically* an arbitrary valuation for the symbols in $\mathcal{U}$. Thus, the result of $\mathcal{U}$-colouring S is the set of all $(\Sigma \cup \mathcal{U})$ structures S' such that (1) S and S' have the same vertices, (2) symbols $R \in \Sigma$ have the same valuation in S and S', and (3) each unary symbol $U \in \mathcal{U}$ is evaluated as an arbitrary subset of $V(S)$.

Finally, a *first-order transduction* Φ from Σ to Γ is the composition of a $\mathcal{U}$-colouring and a first-order interpretation Φ' from $(\Sigma \cup \mathcal{U})$ to Γ. The result $\Phi(S)$ of this transduction on a Σ-structure S is the *set* of all $\Phi'(S')$ for S' a $\mathcal{U}$-coulourings of S.

Transductions are also closed under composition, which means that transductions may equivalently be defined as the composition of any sequence of colourings and interpretations.

Lemma 4.10. *If Φ and Θ are transduction from Σ to Γ and from Γ to Δ respectively, then $\Theta \circ \Phi$ is an transduction from Σ to Δ.*

Proof. The composition of two colourings clearly is a colouring, and interpretations are closed under composition by Lemma 4.9. Finally, an interpretation Φ followed by an $\mathcal{U}$-colouring can always be rewritten as first the $\mathcal{U}$-colouring, and then applying Φ while ignoring the new unary relations. This allows to simplify any sequence of colourings and interpretations into the canonical form of first a colouring, then an interpretation. □

Let us give some examples.

- There is a transduction Ψ which given a graph G, yields the set $\Psi(G)$ of all induced subgraphs of G: the colouring extends G with a single unary predicate U which amounts to picking an arbitrary subset $U(G) \subset V(G)$. Then, the interpretation deletes vertices outside $U(G)$ (with the domain formula $\psi_V(x) = U(x)$), while keeping edges unchanged (with the edge formula $\psi_E(x, y) = E(x, y)$).

- In the introduction, we mentioned that there is an interpretation Φ, namely *squaring*, such that any G is an induced subgraph of $\Phi(G^{(1)})$, where $G^{(1)}$ is the 1-subdivision of G. Define $\Phi' = \Psi \circ \Phi$: using Ψ to retrieve an arbitrary induced subgraph in the result of Φ, we find that $G \in \Phi'(G^{(1)})$ for any graph G.

- Define $\Phi'' = \Psi \circ \Phi \circ \Psi$: we can now pick an induced subgraph before and after the squaring step. Notice that if H is a (non-induced) subgraph of G, then $H^{(1)}$ is an induced subgraph of $G^{(1)}$. It follows that for any subgraph H of G, $H \in \Phi''(G^{(1)})$. In particular, for $G = K_n$ the clique on n vertices, $\Phi''(K_n^{(1)})$ contains all graphs on at most n vertices.

Notice from these examples that when constructing a transduction Φ, we might care about finding a specific graph H inside $\Phi(G)$, but are not bothered by $\Phi(G)$ containing other, unwanted graphs. This can be thought of as pretending that the non-deterministic colouring always makes the choice which is most convenient for us.

If in a different context, these spurious results in $\Phi(G)$ had to be removed, one could add to the transduction a *check* operation which aborts it if a given formula ϕ_c is not satisfied as a result of unwanted non-deterministic choices.

Let us finally mention that it is common to add a *copying* operation to transductions: given a structure S it creates a new copy of S, and adds as a new relation a matching joining each vertex $x \in V(S)$ to its new copy; plus a unary predicate indicating whether a vertex comes from S or from its copy. This allows a transduction Φ to produce an output $\Phi(S)$ larger (although only linearly so) than its input. It can be shown that whether or not copying is allowed is irrelevant for the key notion of independence presented in the next section, which is why we choose to ignore it. All results on transductions presented in this work hold both with and without copying.

4.2.5 Independence. If $\mathcal{C}, \mathcal{D}$ are classes of Σ and Γ-structures respectively, we say that $\mathcal{C}$ *interprets*, respectively *transduces* $\mathcal{D}$ if there is an interpretation, resp. transduction Φ such that $\mathcal{D} \subseteq \Phi(\mathcal{C})$ (where in the case of transductions, $\Phi(\mathcal{C}) = \bigcup_{S \in \mathcal{C}} \Phi(S)$). One can think of this as an encoding of structures of $\mathcal{D}$ with structures of $\mathcal{C}$, where the *decoding* function $\mathcal{C} \to \mathcal{D}$ can be expressed in first-order logic. On the other hand, no assumption is made on the encoding $\mathcal{D} \to \mathcal{C}$.

The case where $\mathcal{D}$ is the class $\mathcal{G}$ of all graphs is of utmost importance: a class $\mathcal{C}$ of structures is called *monadically independent* if $\mathcal{C}$ transduces $\mathcal{G}$. Intuitively, this means that $\mathcal{C}$ is as complex (up to a first-order decoding) as the class of all graphs. This is as complex as it gets, as $\mathcal{G}$ interprets all classes of relational structures.

Lemma 4.11. *For any relational signature Σ, the class $\mathcal{G}$ of all graphs interprets the class of all Σ-structures.*

Proof. Let $\Sigma = \{R_1, \dots, R_k\}$. Given a Σ-structure S, construct the graph G with vertices $V(S) \uplus R_1(S) \uplus \dots \uplus R_k(S)$, where the vertex $(x_1, \dots, x_r) \in R_i(S)$ is adjacent to $x_1, \dots, x_r$.

Now the following transduction (independent of S) reconstructs S from G: first guess with a colouring for each vertex of G whether it comes from $V(S)$, $R_1(S)$, etc. The domain formula discards all vertices outside $V(S)$. Now a tuple $(x_1, \dots, x_r)$ is in $R_i(S)$ if and only if G contains a vertex from $R_i(S)$ adjacent to $x_1, \dots, x_r$. This is a first-order formula which recontructs $R_i(S)$ from G.

Finally, one can remove the colouring step of this transduction by adding to each vertex of G a small gadget which encodes the set it comes from. □

A class $\mathcal{C}$ which is not monadically independent is called *monadically dependent*, or *monadically NIP* (for Non Independence Property). While an independent class is complex (as complex as the class of all graphs), a NIP class is in a sense simple.

The reader will certainly have noticed that we defined *monadic* independence, hinting at a non-monadic variant. Monadicity here refers to the nondeterministic colouring step. Roughly, a class $\mathcal{C}$ is non-monadically independent if it interprets (as opposed to tranduces) the class of all graphs, but for a stronger notion of interpretation which can manipulate tuples of vertices of the input structure. Fortunately, it turns out that the distinction is irrelevant thanks to the following recent and remarkable result:

Theorem 4.12 (Braunfeld and Laskowski [26]). *A hereditary class $\mathcal{C}$ of structures is NIP if and only if it is monadically NIP.*

Since we will only care about hereditary classes, we can safely ignore the difference between NIP and monadically NIP: we will always use the monadic definition (i.e. with transductions), while simply calling it NIP.

With this vocabulary matter out of the way, let us present some examples.

- The class $\left\{G^{(1)} \mid G \text{ graph}\right\}$ of subdivided graphs is independent: it transduces all graphs through the squaring interpretation, as argued in the previous section.
- Any class $\mathcal{C}$ of graphs with bounded cliquewidth is NIP. Indeed one can show that transductions preserve bounded cliquewidth: for any transduction Φ, the image $\Phi(\mathcal{C})$ also has bounded cliquewidth, hence does not contain all graphs (this is true more generally for *monadic second-order* transductions).
- Adler and Adler proved that a class $\mathcal{C}$ of graphs closed under subgraphs is NIP if and only if it is nowhere dense [1] by adapting an older result on infinite graphs of Podewski and Ziegler [81]. This was later generalised to arbitrary relational structures [25].

Remark how these examples of NIP classes coincide with the FPT algorithms for model checking presented in section 4.2.4. This motivates the conjecture which closed the introduction of this chapter.

Conjecture 4.5 [48, Conjecture 8.2]. *A hereditary class $\mathcal{C}$ has an* FPT *first-order model checking algorithm if and only if it is NIP.*

The simpler implication in Conjecture 4.5 is that independent classes do not admit FPT model checking. Even this is deceivingly difficult: the following false proof is enlightening. Consider an independent class of structures $\mathcal{C}$: there is a transduction Φ such that $\Phi(\mathcal{C})$ is the class $\mathcal{G}$ of all graphs. Suppose that there is an FPT algorithm for model checking in $\mathcal{C}$; we use it to solve model checking in $\mathcal{G}$, contradicting Assumption 4.7. Given a graph G and a formula ψ, and want to test if $G \models \psi$. By assumption, there is a preimage $S \in \mathcal{C}$ such that $G \in \Phi(S)$, and by backward translation, we have $G \models \psi$ if and only if $S \models \psi \circ \Phi$; the latter can be tested using the algorithm for $\mathcal{C}$.

There are two errors here:

- The first and minor issue is that backwards translation only works with interpretations, not transductions: in $\Phi(S)$ there might be some graphs which satisfy ψ and others which do not, in which case there can be no reasonable definition of $\Phi \circ \psi$. What we would really prove is that there is no FPT algorithm for model checking of *colourings* of $\mathcal{C}$, which would still be interesting.

- The major issue is that we have no control over the encoding of G into the structure $S \in \mathcal{C}$. A priori, S may be arbitrarily larger than G, in which case testing $S \models \psi \circ \Phi$ would becomes prohibitively costly. Worse, there is no guarantee that the function constructing S from G is computable, let alone efficiently so. Thus the supposed model checking algorithm in $\mathcal{G}$ we just constructed might not even be computable.

Let us fix the proof by adjusting the definition: given classes of structures $\mathcal{C}, \mathcal{D}$, we say that $\mathcal{C}$ *efficiently interprets* $\mathcal{D}$ if there is an interpretation Φ and a polynomial algorithm which given $T \in \mathcal{D}$ finds $S \in \mathcal{C}$ satisfying $T \in \Phi(S)$. With this stronger hypothesis, the previous argument becomes correct and yields the following.

Lemma 4.13. *If $\mathcal{C}$ efficiently interprets $\mathcal{D}$, then there is an* FPT *reduction from model checking in $\mathcal{D}$ to model checking in $\mathcal{C}$.*

And together with Theorem 4.6,

Corollary 4.14. *If $\mathcal{C}$ efficiently interprets the class of all graphs, then first-order model checking in $\mathcal{C}$ is* AW[$*$]*-complete.*

4.3 Twin-width and first-order logic

Let us finally explain how twin-width relates to the logical notions introduced in this chapter. First, we will generalise twin-width to relational structures beyond graphs; this will be essential in Chapters 5 and 6.

A relational signature Σ is called *binary* if all relations in Σ have arity 2. By extension, a structure is binary if its signature is binary. Binary structures turn out to be the most natural setting in which to define twin-width. Consider a binary structure S with signature Σ, a partition $\mathcal{P}$ of $V(S)$, two parts $X, Y \in \mathcal{P}$, and a symbol $R \in \Sigma$. The parts X, Y are *homogeneous* for the binary relation R if either all or none of the pairs $(x, y) \in X \times Y$ are in $R(S)$, and symmetrically all or none of $(y, x) \in Y \times X$ are in $R(S)$. Otherwise, X and Y are in error for R. For instance:

- When the relation is symmetric, we retrieve the notion of errors from Chapter 2.
- When the relation is a linear order $<$, X, Y are homogeneous if and only if $X < Y$ or $Y < X$ (one part is entirely before the other).
- When R is the set of arcs of tournament, meaning that for any vertices $x \neq y$, exactly one of the arcs (x, y) and (y, x) belongs to R, then X, Y are homogeneous if and only if all arcs are oriented from X to Y, or all from Y to X.

Finally, two parts $X, Y \in \mathcal{P}$ are *in error* if they are in error for any of the relations $R \in \Sigma$. The remainder of the definition is then as in graphs: the error degree of X is the number of parts in error with X, the width of the contraction sequence is the maximum error degree, and the twin-width is the minimum width of a contraction sequence.

We can finally state the two main results relating twin-width and first-order logic. The first, announced in section 2.4, is a model checking algorithm.

Theorem 4.15 [19]. *There is an algorithm which given a binary structure S, a contraction sequence of width k for S, and a first-order formula ϕ over the signature of S, tests whether $S \models \phi$ in time $f(k, \phi) \cdot |V(S)|$.*

The second is a stability result under transductions.

Theorem 4.16 [19]. *For any transduction $\Phi : \Sigma \to \Gamma$ with Σ, Γ binary signatures, there is a function f such that any Σ-structure S and $T \in \Phi(S)$ satisfy* $\mathrm{tww}(T) \leqslant f(\mathrm{tww}(S))$.

This can be restated as: for classes of binary structures, if $\mathcal{C}$ has bounded twin-width, then so does any transduction of $\mathcal{C}$.

Since the class of all graphs does not have bounded twin-width, this implies the following.

Corollary 4.17 [19]. *Any class of binary structures with bounded twin-width is NIP.*

We will not prove Theorems 4.15 and 4.16. There are two published proofs of them: the original proof in [19], and a reformulation by Gajarský, Pilipczuk, Przybyszewski, and Toruńczyk [49], which replaces some combinatorial objects with the more abstract notion of *logical types*, leading to several extensions of the results.

Theorem 4.15 uses the same fundamental idea as the independent set algorithm of Theorem 2.21: it is a dynamic programming algorithm which for each part P of each partition in the contraction sequence solves a local model checking problem within a bounded radius around P (radius being meant with regards to the error graph distance).

The proof of Theorem 4.16 uses Theorem 3.12: one shows that any contraction sequence of bounded width for S also has bounded error rank for the transduction $T \in \Phi(S)$. Precisely, for any part P in the contraction sequence, one shows that after deleting parts within a bounded radius from P (where distance is again in the error graph, and the radius depends only on the transduction), the rank of P against the remainder of the graph is bounded.

Chapter 5

NOWHERE SPARSE STRUCTURES

In Chapter 3, we explained how a 'good' ordering of the vertices of a graph, namely one with low grid rank, can be used as witness of twin-width: given such an ordering, a greedy algorithm can compute a contraction sequence of small width (see Theorem 3.9). Finding such a good order is in general an open problem, as hard as approximating twin-width (Question 2.25).

This chapter focuses on specific relational structures S in which one can find a *candidate* ordering $<$ satisfying the converse of Theorem 3.9: if the adjacency matrix $A(S, <)$ has large grid-rank, then S has large twin-width. Then, Theorem 3.9 gives an FPT approximation of twin-width. We present two examples: *ordered structures* where the candidate order $<$ is part of S itself, and *tournaments*, an interesting generalisation of linear orders. Both have a sufficiently rigid structure to ensure that any high-rank division in $A(S, <)$ forbids any low-width contraction sequence.

In addition to this FPT approximation, we will show that several of the remarkable properties of twin-width presented in the previous chapters become *characterisations*: having bounded twin-width is equivalent to being small (cf. Theorem 3.37), having FPT first-order model checking (cf. Theorem 4.15), and to being NIP (cf. Corollary 4.17). Remark that these equivalence do not hold for graphs: the class of cubic graphs has unbounded twin-width (Corollary 3.39), but is well-known to be NIP and have an FPT model checking algorithm [85]; separating small classes from bounded twin-width will be the main result of Chapter 7.

These results on ordered structures were proved by Bonnet, Giocanti, Ossona de Mendez, Simon, Thomassé, and Toruńczyk in [17], a significant part of which was already presented as Theorem 3.9 in Chapter 3. They were generalised to tournaments by Thomassé and the author in [51].

5.1 Definitions and preliminaries

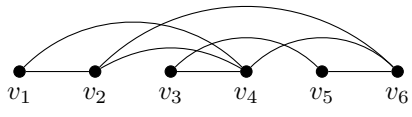
This section introduces the different kinds of relational structures considered in this chapter.

Adjacency matrices We will use adjacency matrices of binary relational structures. If $R \subseteq V^2$ is a binary relation and $<$ is an ordering of V, then the adjacency matrix $A(R, V)$ has V for set of columns and rows ordered by $<$, and contains a '1' at the intersection of row x and column y if and only if $(x, y) \in R$. Thus for a graph $G = (V, E)$, we have $A(G, <) = A(E, <)$.

Theorem 3.9 generalises to binary relational structures as follows.

Theorem 5.1 [17]. *Let S be a binary Σ-structure and $<$ an ordering of $V(S)$. There is a function f depending only of $|\Sigma|$ such that if $A(R(S), <)$ has grid rank at most k for each $R \in \Sigma$, then $\mathrm{tww}(R) \leqslant f(k)$. Furthermore, a contraction sequence witnessing this can be computed in polynomial time.*

Ordered structures An *ordered structure* S is a relational structure whose signature contains a distinguished binary relation symbol $<$, whose valuation in S is a total order $<_S$ over $V(S)$. For example an *ordered graph* $(V, <, E)$ consists of two binary relations, where (V, E) is a graph and $<$ is a total ordering of V (see Figure 5.1).



	v_1	v_2	v_3	v_4	v_5	v_6
v_1		1		1		
v_2	1			1		1
v_3				1	1	
v_4	1	1	1			1
v_5			1			1
v_6		1		1	1	

Figure 5.1: Representation of an ordered graph $(V, <, E)$, and its adjacency matrix $A(E, <)$. Vertices are ordered left-to-right according to $<$.

Ordered structures are interesting for twin-width because it is possible to add a total order to any structure without changing its twin-width. Given a structure $S = (V, R_1, \ldots, R_k)$ and an ordering $<$ of its vertices V, define the structure $(S, <) = (V, <, R_1, \ldots, R_k)$, i.e. S with the relation $<$ added.

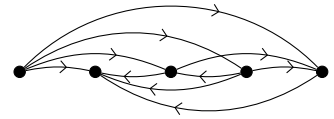
Lemma 5.2. *For any binary structure S, there is an ordering $<$ of $V(S)$ such that* $\mathrm{tww}(S) = \mathrm{tww}(S, <)$.

Proof. Choose an optimal contraction sequence for S, and take $<$ to be an ordering compatible with this contraction sequence. Then there are no errors relative to the relation $<$, hence adding it to the structure does not change the twin-width. □

Permutations and bi-orders Another interesting example of ordered structure is *bi-orders* $(V, <_1, <_2)$, consisting of two different linear orders on the same vertex set. Bi-orders are a natural encoding of permutations as relational structures: a permutation $\sigma : [n] \to [n]$ is encoded as $([n], <, <_\sigma)$ where $<$ is the usual order, and $i <_\sigma j$ if and only if $\sigma(i) < \sigma(j)$, see Figure 5.2.

When a bi-order $(V, <_1, <_2)$ is seen as an ordered structure, either of $<_1$ or $<_2$ can be used as the reference order. We will study permutations and bi-orders in more depth in Chapter 6.

Tournaments A *tournament* $T = (V, A)$ is a structure consisting of a total, irreflexive, and antisymmetric binary relation A: for any two distinct vertices $x, y \in V$, exactly one of (x, y) or (y, x) belongs to A. A pair $(x, y) \in A$ is called an *arc* directed from x to y. Borrowing conventions from graph theory, the arc is simply denoted xy, or $x \to y$ to emphasise the direction. See Figure 5.3 for an example. If the relation A is transitive, then it is a linear order on V, this is called a *transitive tournament*; in this sense, tournaments generalise total orders.



i	1	2	3	4	5
$\sigma(i)$	1	5	3	2	4

Figure 5.2: The permutation $\sigma = 15324$ represented as a biorder $([5], <, <_\sigma)$. The vertices $1, \dots, 5$ are ordered left-to-right by the usual order $<$, while the permuted order $<_\sigma$ is drawn explicitly with an arc $x \to y$ whenever $x <_\sigma y$. These different representations of $<$ and $<_\sigma$ are purely a matter of readability: the two orders play similar roles in the relational structure.

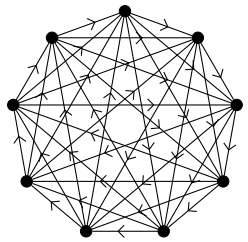


Figure 5.3: Example of tournament: the *rotating tournament*, in which vertices are placed on a circle and arcs are directed clockwise.

Given an ordered graph $G = (V, <, E)$, there is a natural construction of a tournament called the *backedge tournament* T: start with the transitive tournament $(V, <)$, and for each edge $xy \in E$, reverse the direction of the arc between x and y in T. It is simple to check that if $X, Y \subset V$ are homogeneous in G, then they also are in T, which implies that $\mathrm{tww}(T) \leqslant \mathrm{tww}(G)$. This is a very special case of Theorem 4.16: the transformation from G to T is a first-order interpretation, where the arcs of T are defined by the formula

$$xy \in A \quad \iff \quad (x < y) \oplus (xy \in E), \tag{5.1}$$

where $\oplus$ denotes exclusive or.

The converse is false: there are ordered graphs with arbitrarily large twin-width whose backedge tournaments are transitive, hence have twin-width 0. Indeed consider a bi-order $B = (V, <_1, <_2)$ with arbitrarily large twin-width, and construct the *permutation graph* $G = (V, <_1, E)$ where $xy \in E$ if and only if $<_1, <_2$ disagree over xy, i.e. $x <_1 y$ and $y <_2 x$ or vice versa. One can show that parts $X, Y \subset V$ are homogeneous in B if and only if they are homogeneous in G, and thus $\mathrm{tww}(B) = \mathrm{tww}(G)$. However, the backedge tournament of G is $(V, <_2)$, which is transitive, hence has twin-width 0.

5.2 Approximating twin-width in ordered structures

Theorem 3.9 states that for a graph $G = (V, E)$ and an ordering $<$, if $A(G, <)$ has small grid rank, then G has small twin-width. The converse is false: a

stupid choice of $<$ may cause high grid rank even if G is a path. However, we will show that in this case, the *ordered graph* $(V, <, E)$ has large twin-width. This is a generalisation of Lemma 3.21: the latter showed that if $A(G, <)$ has large grid rank then any contraction sequence compatible with $<$ has large width; we need to generalise this to contraction sequences with bounded error degree with regards to $<$.

Given a subset $X \subset V(G)$, denote by $\overline{X}$ its closure, meaning the smallest interval of $<$ containing X. Recall that two parts $X, Y \subset V(G)$ are in error with regards to $<$ if and only if $\overline{X}$ intersects $\overline{Y}$.

Lemma 5.3 [17]. *Let $G = (V, <, R)$ be an ordered binary structure of twin-width k. Then $A(R, <)$ has grid rank at most $2(k+1)^2 + 1$.*

Proof. Fix $\mathcal{P}_n, \dots, \mathcal{P}_1$ a contraction sequence of width k for the ordered structure G. Denote by $M = A(R, <)$ its adjacency matrix, and consider for a contradiction a rank-$(2(k+1)^2+2)$ division $\mathcal{D} = (\mathcal{R}, \mathcal{C})$. Note that parts in $\mathcal{R}$ or $\mathcal{C}$ must be intervals of $<$, while the parts of $\mathcal{P}_i$ need not.

Choose the earliest step in the sequence (i.e. the largest $i \in [n]$) such that there is a part $P \in \mathcal{P}_i$ whose closure $\overline{P}$ contains some part of either $\mathcal{R}$ or $\mathcal{C}$, say $R \subseteq \overline{P}$ with $R \in \mathcal{R}$.

Call $P_0 = P$, and let $P_1, \dots, P_\ell$ be the remaining parts of $\mathcal{P}$ which intersect $\overline{P}$. Since $\mathcal{P}$ has error degree at most k with regards to $<$, we have $\ell \leqslant k$. Furthermore, let $Q_1, \dots, Q_t$ be the parts in error with one of the P_i for the relation R. We have $t \leqslant k \cdot (\ell + 1) \leqslant k(k+1)$.

Now by choice of i, for any $A \in \mathcal{P} \setminus \{P\}$, the closure $\overline{A}$ intersects at most two parts of $\mathcal{C}$, while $\overline{P}$ itself intersects at most three parts of $\mathcal{C}$. Since $\mathcal{C}$ consists of at least $2(k+1)^2+2$ parts, one of them, say $C \in \mathcal{C}$ is disjoint from all P_i and Q_j, hence C is homogeneous to each of $P_0, \dots, P_\ell$. Using that $R \subset \overline{P} \subset \bigcup_{i=0}^{\ell} P_i$, it follows that

$$\operatorname{rk}(R; C) \leqslant \operatorname{rk}\left(\bigcup_{i=0}^{\ell} P_i;\ C\right) \leqslant \sum_{i=0}^{\ell} \operatorname{rk}(P_i; C) \leqslant k, \tag{5.2}$$

contradicting the hypothesis that $\mathcal{D}$ is a rank-$(2(k+1)^2+2)$ division. □

Combining Lemma 5.3 and Theorem 5.1 yields two important results.

Corollary 5.4 [17]. *For any ordered binary signature Σ, there is a function f and a polynomial time algorithm which given a Σ-structure S, computes a contraction sequence of width at most $f(\operatorname{tww}(S))$.*

Corollary 5.5 [17]. *Let $\mathcal{C}$ be a class of ordered binary Σ-structures, and $\mathcal{A}$ the class of all adjacency matrices $A(R(S), <_S)$ obtained from structures $S \in \mathcal{C}$ and relations $R \in \Sigma$. Then $\mathcal{C}$ has bounded twin-width if and only if $\mathcal{A}$ has bounded grid rank.*

An interesting consequence of Corollary 5.5 is that ordered structures can be superposed while preserving bounded twin-width.

Lemma 5.6. *Consider two ordered binary structures $S = (V, <, R_1, \dots, R_k)$ and $T = (V, <, Q_1, \dots, Q_\ell)$ with the same vertices and same ordering, and construct the structure $S + T = (V, <, R_1, \dots, R_k, Q_1, \dots, Q_\ell)$ by combining the relations of S and T. Then $\operatorname{tww}(S+T)$ is bounded by a function of $\operatorname{tww}(S)$ and $\operatorname{tww}(T)$.*

Proof. By Corollary 5.5, the grid rank of all the adjacency matrices $A(R_i, <)$ and $A(Q_j, <)$ is bounded by functions of $\mathrm{tww}(S)$ and $\mathrm{tww}(T)$ respectively. By Corollary 5.5, the twin-width of $S + T$ is bounded by a function of the former grid ranks. □

This is particularly useful for permutations, where this superposition corresponds to composition:

Corollary 5.7. *Let $\mathcal{C}_1, \mathcal{C}_2$ be classes of permutations represented as bi-orders, and $\mathcal{C} = \mathcal{C}_1 \circ \mathcal{C}_2$ the class of all compositions $\sigma \circ \tau$, $\sigma \in \mathcal{C}_1$, $\tau \in \mathcal{C}_2$ when it is well-defined. If $\mathcal{C}_1, \mathcal{C}_2$ have bounded twin-width, then so does $\mathcal{C}$.*

Proof. Given permutations $\sigma \in \mathcal{C}_1$, $\tau \in \mathcal{C}_2$, consider the encodings as bi-orders $\sigma = ([n], <, <_\sigma)$ and $\tau^{-1} = ([n], <, <_{\tau^{-1}})$. Using Lemma 5.6 to superpose them along the standard ordering $<$, we obtain that the bi-order $([n], <_{\tau^{-1}}, <_\sigma)$ has twin-width bounded by a function of $\mathrm{tww}(\sigma), \mathrm{tww}(\tau)$. It is simple to check that $([n], <_{\tau^{-1}}, <_\sigma)$ is isomorphic to the encoding of the composition $\sigma \circ \tau$. □

5.3 Approximating twin-width in tournaments

We will now extend the results of the previous section to tournaments. The fundamental idea remains the same: for any tournament T, there is an ordering $<$ such that $\mathrm{tww}(T)$ is functionally equivalent to the grid rank of $A(T, <)$. For ordered structures, the ordering $<$ was already given; for tournaments, we need to construct it.

5.3.1 Binary search trees. The following construction is inspired by the classical algorithmic data structure called binary search trees, used to store elements of a linear order: this version instead contains vertices of a tournament. Interestingly, this exact construction was used in a different context by Ailon, Charikar, and Newman [3]: when randomized, it yields an approximation algorithm called *kwiksort* for the *feedback arc set* problem in tournaments. Our analysis of the construction is however entirely different from theirs, and does not require randomisation.

Consider a tournament T. Given a vertex $x \in V$, its *out-neighbourhood* is $N_T^+(x) = \{y \in V(T) \mid x \to y\}$, and symmetrically its *in-neighbourhood* is $N_T^-(x) = \{y \in V(T) \mid y \to x\}$. Remark that $\{x\}, N_T^+(x), N_T^-(x)$ is a partition of $V(T)$. A *binary search tree* (BST) in T is a rooted ordered binary tree S (meaning that each node has a left and right child, either of which may be missing), whose nodes are the vertices of T, and such that for any $x \in S$

- the left child of x (if any) and its descendants are in $N_T^-(x)$, and
- the right child of x (if any) and its descendants are in $N_T^+(x)$,

see Figure 5.4. In the classical notion of BST over a linear order, the whole left subtree of x is smaller than the whole right subtree. There is no such restriction for tournaments: edges between the left and right child of x or their respective descendants can have arbitrary orientations.

The *ordering* associated with S, denoted $<_S$, is the left-to-right order, i.e. the one which places a node x after its left child and its descendants, but before its right child and its descendants. Such an order is called a *BST order*. A

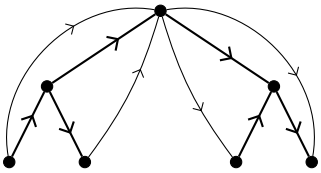


Figure 5.4: A binary search tree in a tournament. The direction of omitted edges is not constrained.

tournament T in general admits many different BST orders, and any of them will be appropriate for our purpose. Constructing one in polynomial time is easy.

Note that if x is an ancestor of y, then there is an edge oriented from x to y if and only if $x <_S y$. Thus we have

Remark 5.8. In a tournament T, any branch B of a BST S forms a transitive subtournament $T[B]$, whose order coincides with $<_S$. That is, for $x, y \in B$, we have $x \to y$ if and only if $x <_S y$.

In the remainder of this section, we will prove the following.

Theorem 5.9 [51]. *There is a function f such that for any tournament T and BST order $<_S$, the grid rank of $A(T, <_S)$ is at most $f(\mathrm{tww}(T))$.*

Similarly to ordered structures, Theorem 5.9 combined with Theorem 5.1 yields an approximation algorithm and a characterisation of bounded twin-width for tournaments.

Corollary 5.10 [51]. *There is a function f and a polynomial algorithm which given a tournament T, finds a contraction sequence of width at most $f(\mathrm{tww}(T))$.*

Corollary 5.11 [51]. *Let $\mathcal{C}$ be a class of tournaments, and $\mathcal{A}$ the class of all adjacency matrices $A(T, <_S)$ for all $T \in \mathcal{C}$ and BST order $<_S$ in T. Then $\mathcal{C}$ has bounded twin-width if and only if $\mathcal{A}$ has bounded grid rank.*

5.3.2 Chain orders. Our goal is now to show that if $<_S$ is a BST order for T and $A(T, <_S)$ has large grid rank, then T has large twin-width. To this end, we will extract a small subtournament $T' \subset T$ such that the restriction $(T', <_S)$ can be obtained by first-order transduction from T, while ensuring that $A(T', <_S)$ still has large grid rank, thereby reducing the problem to Corollary 5.5.

The first-order transduction uses a variant of lexicographic orderings defined below. The transduction actually constructs a quasi-ordering $\preceq$ (i.e. there can be equivalent vertices: $x \preceq y$ and $y \preceq x$); and we latter extract a subset of vertices on which $\preceq$ is a total ordering.

In a tournament T, say that a subset $C \subset V(T)$ is a *chain* if it induces a transitive subtournament $T[C]$. The *chain quasi-order* $\preceq^+_C$ is defined as follows. First enumerate the vertices of C as $c_1, \dots, c_k$ so that $c_i \to c_j$ whenever $i < j$. Define $A_i = \bigcap_{j \leqslant i} N^+(c_j)$, and $B_i = A_{i-1} \cap N^-(c_i)$. Then each of $B_1, \dots, B_k$ and A_k is an equivalence class of $\preceq^+_C$, and the classes are ordered as

$$B_1 \prec^+_C c_1 \prec^+_C B_2 \prec^+_C c_2 \prec^+_C \dots B_k \prec^+_C c_k \prec^+_C A_k,$$

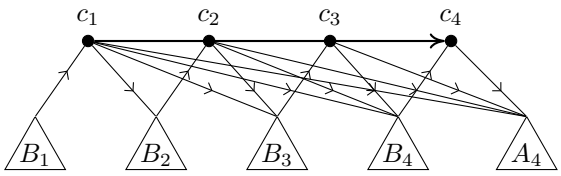


Figure 5.5: Example of construction of the quasi-order $\preceq_C^+$. The quasi-order is from left to right, and the triangles are equivalence classes. The direction of omitted edges (from B_i to $B_j \cup \{c_j\}$ for $i < j$) is not constrained. For $\preceq_C^-$, the direction of all edges would be reversed.

see Figure 5.5. This can be seen as the left-to-right order of a partial BST consisting only of a single branch $c_1, \dots, c_k$, with c_1 as root and c_k as leaf.

The dual quasi-order $\preceq_C^-$ is defined in the same way, but reversing the direction of all edges. Thus, we now enumerate C so that edges are from c_i to c_j when $i > j$, while $A_i = \bigcap_{j \leqslant i} N^-(c_i)$ and $B_i = A_{i-1} \cap N^+(c_i)$. The rest of the definition is the same.

Lemma 5.12. *There is a transduction Φ such that for any tournament T and chain C in T, we have $(T, \preceq_C^+) \in \Phi(T)$.*

Proof. The transduction Φ first uses a colouring step to guess the chain C. Depending on this non-deterministic choice, it may yield *any* chain quasi-ordering. The remainder of Φ is a deterministic interpretation.

Having guessed C, the ordering $\preceq_C^+$ restricted to C is simply given by the orientation of edges. Enumerate C as $\{c_1 \preceq_C^+ \dots \preceq_C^+ c_k\}$. Given $x \notin C$, let $i(x) \in \{0, \dots, k\}$ be maximal such that there are edges oriented $c_j \to x$ for all $1 \leqslant j \leqslant i(x)$. Then

- for any $x \notin C$, we have $c_1 \dots c_{i(x)} \preceq_C^+ x \preceq_C^+ c_{i(x)+1}, \dots, c_k$, and
- for $x, y \notin C$, we have $x \preceq_C^+ y$ if and only if $i(x) \leqslant i(y)$.

This characterisation of $\preceq_C^+$ can be expressed by a first-order formula. □

5.3.3 Extraction. We now come to the main technical result of this section: from a BST order with very large grid rank, we extract a chain order with large grid rank.

Relative to a quasi-ordering $\preceq$, we say that two subsets X, Y are *non-overlapping* if either $X \prec Y$ (all points in X are strictly smaller than all those in Y) or $Y \prec X$.

Lemma 5.13 [51]. *Let T be a tournament and S be a BST with associated order $<_S$. There is a function $f(k) = 2^{O(k)}$ independent of T and S satisfying the following. For any family $\mathcal{P}$ of at least $f(k)$ disjoint intervals of $<_S$, there is a chain C in T, an orientation $o \in \{+, -\}$ and a subfamily $\mathcal{P}' \subset \mathcal{P}$ such that $|\mathcal{P}'| \geqslant k$ and such that the parts of $\mathcal{P}'$ are non-overlapping for $\preceq_C^o$.*

Furthermore, C, o, and $\mathcal{P}'$ can be computed in linear time.

Proof. Let T be a tournament, S a BST of T and $<_S$ the corresponding order. Consider a family $\mathcal{P}$ of at least $f(k)$ disjoint intervals of $<_S$, where $f(k) = 2^{O(k)}$ will be determined later.

We choose a branch $B = b_0, \dots, b_p$ of S by the following process. First b_0 is the root of S. For each (yet to be determined) b_i, let S_i be the subtree of S rooted in b_i, and define the weight w_i to be the number of classes of $\mathcal{P}$ intersected by S_i. Then b_{i+1} is chosen to be the child of b_i which maximizes w_{i+1}. This choice ensures that

$$2w_{i+1} + 1 \geqslant w_i. \tag{5.3}$$

For each $i < p$, let d_i be the child of b_i other than b_{i+1} (sometimes d_i does not exist), and let D_i be the subtree of S rooted at d_i (D_i is empty if d_i does not exist). Furthermore, let L, R be the sets of vertices which are before, resp. after the leaf b_p in the order $<_S$. For any $0 \leqslant i \leqslant j \leqslant p$, let

$$L_{i,j} = \bigcup_{\substack{i \leqslant \ell < j \\ b_\ell \in L}} \{b_\ell\} \cup D_\ell, \qquad \text{and} \qquad R_{i,j} = \bigcup_{\substack{i \leqslant \ell < j \\ b_\ell \in R}} \{b_\ell\} \cup D_\ell.$$

Roughly speaking, $L_{i,j}$, resp. $R_{i,j}$ consists of subtrees branching out of B on the left, resp. right, between b_i and b_j.

Claim 5.14. For any i, j, the subtree S_i is partitioned into $L_{i,j} <_S S_j <_S R_{i,j}$.

Proof. Clearly $L_{i,j}, S_j, R_{i,j}$ partition S_i. Furthermore, if $\ell < j$ and $b_\ell \in L$, then $b_\ell <_S S_j$, and in turn $D_\ell <_S b_\ell$. This proves $L_{i,j} <_S S_j$, and symmetrically $S_j <_S R_{i,j}$. ■

Claim 5.15. For $0 \leqslant i < j \leqslant p$, if $w_i \geqslant w_j + 3$, then there is a part $P \in \mathcal{P}$ such that $P \subset L_{i,j}$ or $P \subset R_{i,j}$.

Proof. There are at least three parts of $\mathcal{P}$ which intersect S_i but not S_j. Since these three parts and S_i are all intervals of $<_S$, one of these parts, say P, is contained in S_i. Thus P is a subset of S_i but does not intersect S_j, which using Claim 5.14 implies that $P \subset L_{i,j}$ or $P \subset R_{i,j}$. ■

Construct a sequence $i_0 < \cdots < i_{2k}$ of indices in $\{0, \dots, p\}$ inductively by taking $i_0 = 0$, and choosing $i_{\ell+1}$ minimal such that $w_{i_{\ell+1}} \leqslant w_{i_\ell} - 3$. Using (5.3) and the minimality of $i_{\ell+1}$, we obtain for all ℓ that

$$2w_{i_{\ell+1}} + 1 \geqslant w_{i_{\ell+1}-1} > w_{i_l} - 3, \tag{5.4}$$

$$\text{hence} \quad 2w_{i_{\ell+1}} + 3 \geqslant w_{i_l}. \tag{5.5}$$

We now define f by $f(0) = 1$ and $f(k+1) = 4f(k) + 9$. By assumption, $w_0 = |\mathcal{P}| \geqslant f(k)$, and it follows from (5.5) that the construction of i_ℓ can be carried out up to at least i_{2k}.

Define $L'_\ell = L_{i_{\ell-1}, i_\ell}$, and similarly $R'_\ell = R_{i_{\ell-1}, i_\ell}$, see Figure 5.6. By Claim 5.15, for any $\ell \in [2k]$, either L'_ℓ or R'_ℓ contains a part of $\mathcal{P}$. Thus, either there are at least k distinct L'_ℓ containing a part of $\mathcal{P}$, or there are at least k distinct R'_ℓ containing a part of $\mathcal{P}$. Without loss of generality, assume that we are in the former case. We will now forget about vertices which are not in L.

Define $C = L \cap B$. By Remark 5.8, this is a chain, whose order coincides with $<_S$. Furthermore, at any node x of C, the branch B does descend on the right side, since $x <_S b_p$. Thus, the order in C also coincides with the

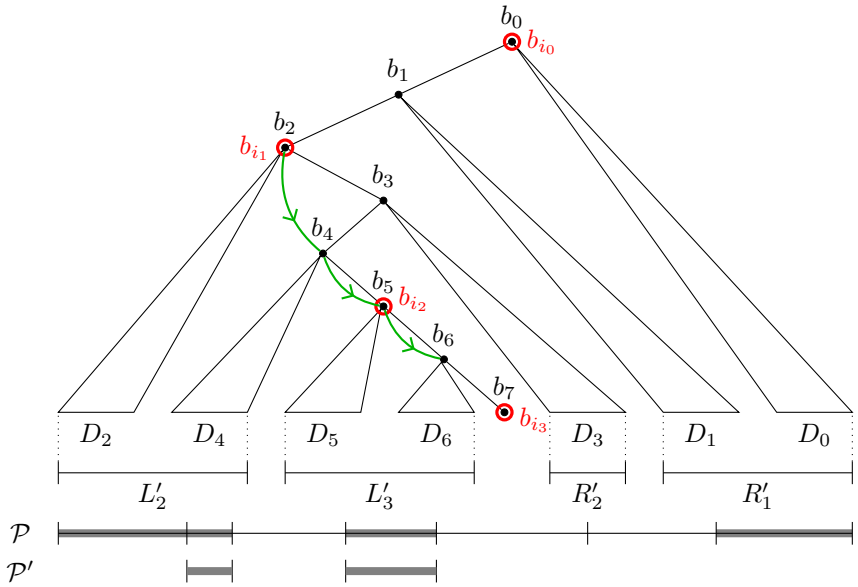


Figure 5.6: Sketch of the proof of Lemma 5.13. In the upper half, the BST T with the extracted branch B; circled in red, the extracted subsequence b_{i_ℓ}; in green arrows, the chain $C = B \cap L = \{b_2, b_4, b_5, b_6\}$. Below the tree, from top to bottom: the partition in L'_ℓ and R'_ℓ; the initial family (here partition) $\mathcal{P}$, with the parts contained in some L'_ℓ or R'_ℓ highlighted; the final family $\mathcal{P}'$, obtained by selecting a part of $\mathcal{P}$ inside each possible L'_ℓ.

ancestor–descendent order of S. (Remark here that if we were in R instead of L, the order of C would be the inverse of the ancestor–descendant order.) Now, if C is enumerated as $c_0 <_S \cdots <_S c_t$, and C_i is the subtree branching out on the left of c_i, defined similarly to D_i, then the chain quasi-order $\preceq^+_C$ restricted to L is exactly

$$C_0 \prec^+_C c_0 \prec^+_C C_1 \prec^+_C c_1 \prec^+_C \dots \prec^+_C c_t$$

where each subtree C_i is an equivalence class. (In R, we would instead use $\preceq^-_C$.) From this description, we obtain that any $L_{i,j}$ is an interval of $\preceq^+_C$ restricted to L.

For each L'_ℓ, select a part of $\mathcal{P}$ included in L'_ℓ if any, and define $\mathcal{P}'$ as the collection of selected parts. Thus $\mathcal{P}' \subset \mathcal{P}$, and we know from the choice of the family $\{L'_\ell\}_{\ell \in [2k]}$ that $|\mathcal{P}'| \geqslant k$. Furthermore, if $X \neq Y$ are parts of $\mathcal{P}'$, there are $s \neq t$ such that $X \subseteq L'_s$ and $Y \subseteq L'_t$. Since each L'_ℓ is an interval of $(L, \preceq^+_C)$, this implies that X and Y are non-overlapping for $\preceq^+_C$. Thus $\mathcal{P}'$ satisfies all desired properties.

Finally, given the BST S and the family $\mathcal{P}$, it is routine to compute the weights w_i of all nodes in S by a bottom-up procedure; this only requires to compute the left-most and right-most parts of $\mathcal{P}$ intersecting each subtree. From this, it is simple to choose in linear time the branch B, the indices i_ℓ, the better side L or R, and finally to compute C and $\mathcal{P}'$. □

5.3.4 Characterisation of twin-width in tournaments. We are almost ready to prove Theorem 5.9. The last lemma needed is used to extract one element from each class of a chain quasi-order while preserving high grid-rank.

Lemma 5.16. *Given a rank-k division $(\mathcal{R},\mathcal{C})$ in a matrix M, one can extract a full-rank $k\times k$ submatrix M' which contains exactly one row (resp. column) from each part $R\in\mathcal{R}$ (resp. $C\in\mathcal{C}$).*

Proof. Let $\mathcal{R}=\{R_1,\dots,R_k\}$ and $\mathcal{C}=\{C_1,\dots,C_k\}$ be the blocks of the division. Suppose we have already extracted one row (resp. column) from each of $R_1,\dots,R_{k-1}$ (resp. $C_1,\dots,C_{k-1}$) to form a submatrix M_{k-1} of rank $k-1$. Add all rows of R_k and all columns of C_k to M_{k-1}. This matrix has rank at least k because it contains $R_k\times C_k$. Thus we can remove all but one row of R_k while preserving rank k, then similarly remove all but one column of C_k, yielding the desired matrix M'. □

Lemma 5.17. *Given a rank-(k^3) division $(\mathcal{R},\mathcal{C})$ in a matrix M, one can extract a submatrix M' with grid rank at least k which contains exactly one row (resp. column) from each part $R\in\mathcal{R}$ (resp. $C\in\mathcal{C}$).*

Proof. Let $\mathcal{R}=\{R_0<\cdots<R_{k^3-1}\}$ and $\mathcal{C}=\{C_0<\cdots<C_{k^3-1}\}$ be the blocks of the division. We will use two levels of coarser partitions, regrouping blocks k by k: define the partition $\mathcal{R}'$ with parts $R'_i=R_{ki}\cup\cdots\cup R_{k(i+1)-1}$ for $i<k^2$, and $\mathcal{R}''$ with parts $R''_i=R'_{ki}\cup\cdots\cup R'_{k(i+1)-1}$ for $i<k$, and similarly $\mathcal{C}'$ and $\mathcal{C}''$. Thus the partitions $\mathcal{R}',\mathcal{C}'$ have k^2 parts each, while $\mathcal{R}'',\mathcal{C}''$ have k parts each.

Now for each $i,j\in[k]$, we consider the submatrix $R'_{ki+j}\times C'_{kj+i}$. It has a rank-$k$ division given by $(\mathcal{R},\mathcal{C})$, hence by Lemma 5.16, we can extract from it a rank-k submatrix $M_{i,j}$ using only one row or column from the relevant blocks of $\mathcal{R}$ and $\mathcal{C}$. We discard the rows of R'_{ki+j} and C'_{kj+i} which are not used in $M_{i,j}$: they will not be used elsewhere. Applying the above for each $i,j\in[k]$ yields a submatrix M' of M containing exactly one row (resp. column) from each part of $\mathcal{R}$ (resp. $\mathcal{C}$), as required. Consider the restriction of $(\mathcal{R}'',\mathcal{C}'')$ on M': the zone $R''_i\times C''_j$ contains the submatrix $M_{i,j}$, hence has rank at least k. This shows that $(\mathcal{R}'',\mathcal{C}'')$ induces a rank-k division in M'. □

We are now ready to characterise twin-width of tournaments.

Proof of Theorem 5.9. Consider a tournament T with twin-width k, and a BST order $<_S$ such that $A(T,<_S)$ has a rank-t division $\mathcal{D}=(\mathcal{R},\mathcal{C})$. We want to bound t by a function of k.

By dropping half of the parts in $\mathcal{R}$, and half of those in $\mathcal{C}$, we can assume that they do not intersect: Indeed, for $t_1=\lfloor t/2\rfloor$, if $\mathcal{R}=\{R_1<_S\cdots<_S R_t\}$ and $\mathcal{C}=\{C_1<_S\cdots<_S C_t\}$, then either $R_{t_1}<_S C_{t-t_1+1}$ in which case $R_1,\dots,R_{t_1}$ are disjoint from $C_{t-t_1+1},\dots,C_t$, or $C_{t_1}<_S R_{t-t_1+1}$ and vice versa. Either way, we find t_1 parts of $\mathcal{R}$ disjoint from t_1 parts of $\mathcal{C}$.

Choose t_2 maximal satisfying $t_1\geqslant f(t_2)$, where f is the bound given by Lemma 5.13. We then apply Lemma 5.13 to $\mathcal{R}$ and to $\mathcal{C}$ independently. This yields subfamilies $\mathcal{R}'\subset\mathcal{R}$ and $\mathcal{C}'\subset\mathcal{C}$ each of cardinality t_2, as well as chain quasi-orders $\preceq_1$ and $\preceq_2$ such that parts in $\mathcal{R}'$ (resp. $\mathcal{C}'$) are non-overlapping for $\preceq_1$ (resp. $\preceq_2$).

Call $X=\bigcup_{R\in\mathcal{R}'}R$ and $Y=\bigcup_{C\in\mathcal{C}'}C$ the sets of rows and columns remaining in $\mathcal{R}'$ and $\mathcal{C}'$. Consider the adjacency matrix of T restricted to rows in X

and columns in Y ordered by $\preceq_1$ and $\preceq_2$ respectively, with ties in the quasi orderings broken arbitrarily. In this matrix, $(\mathcal{R}', \mathcal{C}')$ is a rank t_2 division. By Lemma 5.17, we can extract subsets $X' \subset X$ and $Y' \subset Y$ such that the submatrix on $X' \times Y'$ contains a rank-t_3 division for $t_3 = \left\lfloor \sqrt[3]{t_2} \right\rfloor$, and X' (resp. Y') intersects each part of $\mathcal{R}'$ (resp. $\mathcal{C}'$) at most once. Recall that Lemma 5.13 ensures that the parts of $\mathcal{R}'$ and $\mathcal{C}'$ are non-overlapping for $\preceq_1$ and $\preceq_2$ respectively. The above implies that $\preceq_1$ restricted to X and $\preceq_2$ restricted to Y are total orderings.

Finally, let T' be the subtournament induced by $X \cup Y$, and define the ordering $<$ of its vertices to coincide with $\preceq_1$ inside X and with $\preceq_2$ inside Y, while $X < Y$. By construction, $A(T', <)$ has grid rank at least t_3. Furthermore, it follows from Lemma 5.12 that there is a fixed transduction Φ such that $(T', <) \in \Phi(T)$. Thus, by Theorem 4.16, there is some function g such that $\mathrm{tww}(T', <) \leqslant g(\mathrm{tww}(T))$. Call $k' = \mathrm{tww}(T', <)$. Finally, applying Lemma 5.3 to the ordered structure $(T', <)$ yields that its grid rank is $t_3 \leqslant 2(k'+1)^2 + 1$.

Combining the inequalities on t, t_1, t_2, t_3, k', and k throughout the proof, we find that the grid rank t is bounded by some function of the twin-width k. □

5.3.5 Structures over tournaments. Theorem 5.9 only considers tournaments, but it can be extended to *binary structures over tournaments*, that is structures of the form $S = (V, A, R_1, \ldots, R_k)$ where (V, A) is a tournament and R_i are arbitrary binary relations. In a sense, the tournament (V, A) in S plays a role comparable to the ordering in an ordered structure: its presence constrains twin-width sufficiently to approximate it.

Given a structure $S = (V, A, R_1, \ldots, R_k)$, we call BST order in S any BST order of the tournament (V, A).

Theorem 5.18 [51]. *Let $\mathcal{C}$ be a class of binary structures over tournaments, and $\mathcal{A}$ the class of all adjacency matrices $A(R(S), <)$ for structures $S \in \mathcal{C}$, relations R in its signature, and BST orderings $<$. Then $\mathcal{C}$ has bounded twin-width if and only if $\mathcal{A}$ has bounded grid rank.*

Sketch of proof. If $\mathcal{A}$ has bounded grid rank, then Theorem 5.1 applied to BST orders of structures in $\mathcal{C}$ implies that $\mathcal{C}$ has bounded twin-width.

The converse is a simple variant of the proof of Theorem 5.9: consider $S \in \mathcal{C}$ with a BST order $<$, and a relation R in its signature such that $A(R(S), <)$ has a very high rank division. Lemmas 5.13 and 5.17 allow to extract a smaller division whose parts rows and columns are ordered by some chain orders, and from this, Lemma 5.12 yields through a first-order transduction an ordered structure with high grid rank. □

5.4 Extracting canonical obstructions

In the previous sections, we proved that the lack of high-rank divisions in appropriate orderings characterises bounded twin-width for ordered structures and tournaments: high-rank divisions are the only obstructions to twin-width. Our goal is now to clean up these obstructions to obtain some canonical form thereof. Using these *canonical obstructions*, we will be able to prove the converses of Theorems 3.37 and 4.15 and Corollary 4.17:

Theorem 5.19 [17, 51]. *Let $\mathcal{C}$ be a hereditary class of ordered structures or tournaments. Then the following are equivalent:*

1. *$\mathcal{C}$ has bounded twin-width,*
2. *$\mathcal{C}$ is a small class,*
3. *$\mathcal{C}$ has an FPT first-order model checking algorithm, and*
4. *$\mathcal{C}$ is NIP.*

Where equivalence with (3) *is under the $\mathsf{FPT} \neq \mathsf{AW}[*]$ assumption.*

We know from the previous chapters that bounded twin-width implies all other conditions in Theorem 5.19 (using the approximation algorithms from Corollaries 5.4 and 5.10 for the FPT algorithm). The part we need to prove, which is specific to ordered structures and tournaments, is that classes with unbounded twin-width cannot satisfy the other conditions. This proceeds in two steps: find the canonical form of obstructions found as induced substructure in the structures with large twin-width, and show that these obstructions do not satisfy conditions (2)–(4).

The extraction of canonical obstructions is a tedious process, involving repeated use of Ramsey's theorem and similar results. We will thus omit this part of the proof, and instead focus on the obstructions themselves. A few different forms of obstructions appear for both ordered structures and tournaments, but they all share a commonality: they are encodings of arbitrary permutations. That is, a class has unbounded twin-width if and only if it contains encodings of *all* permutations, for one of the encodings which we will describe. Since our goal is to show that these encodings of permutations do not satisfy (2)–(4), it is natural to first prove it for the permutations themselves.

5.4.1 Permutations as obstructions. Recall that we encode permutations as biorders: the superposition of two total orderings of the same vertex set. Call $\mathcal{B}$ the class of all biorders, which we want to show to be complex in the sense of enumeration and first-order logic. The former is very simple:

Fact 5.20. *The class of all biorders $\mathcal{B}$ is not small (and a fortiori not tiny).*

Proof. Up to isomorphism, there are exactly $n!$ biorders on n vertices. When counting labelled biorders, the labelling encode a second permutation, hence they are $(n!)^2$ biorders on the vertex set $[n]$ counted up to equality. □

We now focus on the latter: the class $\mathcal{B}$ is independent (in the sense of first-order logic). That is, one can encode all graphs in $\mathcal{B}$, so that the decoding is a first-order interpretation. We will additionally show that this is an *efficient interpretation* (the encoding function is polynomially computable), which implies that first-order model checking in $\mathcal{B}$ is hard.

Lemma 5.21 (Folklore). *The class $\mathcal{B}$ efficiently interprets all graphs.*

Proof. Consider a graph $G = (V, E)$ and let us describe its encoding as a biorder $B_G = (X, <_1, <_2)$. The vertex set X consists of the following:

- the vertices V of G,
- for each edge $e = uv$ in E, three vertices u_e, v_e, s_e, where u_e, v_e can be seen as half-edges, while s_e is a separator, and
- two additional separators s_1, s_2.

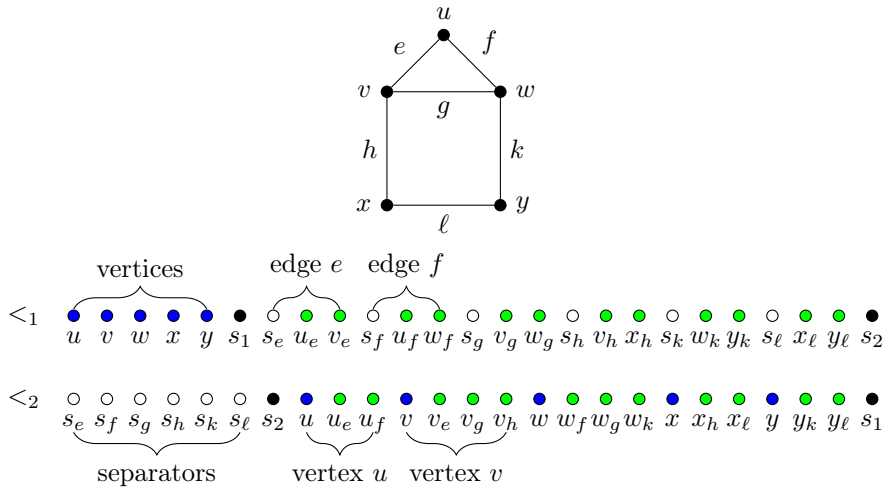


Figure 5.7: Encoding of a graph G as a biorder $(X, <_1, <_2)$. Each of the two orderings $<_1, <_2$ of the encoding is depicted by listing vertices left-to-right. Vertices are color-coded for readability only: vertices from G in blue, half-edges in green, edge separators in white, special separators in black.

The orderings are as follows (see Figure 5.7):

1. In $<_1$, V comes first, followed by the separator s_1. Then, for each edge $e \in E$, the vertices s_e, u_e, v_e in that order (up to swapping u and v), and finally s_2.

2. In $<_2$, all the vertices s_e, $e \in E$ come first, followed by the separator s_2. Then come V and all the half-edges u_e, v_e, with the only constraint that each half-edge u_e is located after u, but before the next vertex of V. Finally s_1 is the last vertex for $<_2$.

Clearly this encoding can be computed in polynomial time given G. The decoding proceeds as follows, given B_G:

1. Firstly s_1, s_2 are immediately identified as the last vertices of $<_2, <_1$ respectively.

2. Using s_1, s_2, one can identify the type of all remaining vertices of B_G: the vertices from G are the ones before s_1 in $<_1$, the separators s_e are before s_2 in $<_2$, and everything else is a half-edge.

3. In $<_1$, the two half edges u_e, v_e of each edge $e = uv$ are next to each other, with separators on either side. This allows to reconstruct the pairing of half-edges.

4. Finally, in $<_2$ the half-edges incident to u are the ones located after u but before the next vertex of G. This allows to reconstruct the vertex–half-edge incidence.

5. From the vertex–half-edge incidence, and the pairing of half-edges, it is simple to reconstruct the graph.

Each of the above steps is easily expressed in first-order logic, thus resulting in a first-order interpretation which reconstructs G from B_G. □

Thus, using we obtain,

Corollary 5.22. *The class $\mathcal{B}$ of all biorders is independent.*

And using Corollary 4.14,

Corollary 5.23. *First-order model checking in the class $\mathcal{B}$ is $\mathsf{AW}[*]$-complete.*

5.4.2 Obstructions in ordered graphs. We now come back to the question of extracting obstructions: in an ordered graph $(V, <, E)$ whose adjacency matrix $M = A(E, <)$ contains a high-rank division, we want to extract some encoding of permutations.

As an example, let us demonstrate the process in the case of graphs with bounded degree. There, we obtain the most common encoding of permutations as matrices: the *permutations matrix* of $\sigma : [n] \times [n]$ is the matrix with $[n]$ as rows and columns (in the natural order), and a '1' at the intersection of column i and row j when $j = \sigma(i)$. Remark that a matrix is a permutation matrix if and only if every row and column contains a single '1'.

Lemma 5.24. *Let $\mathcal{M}$ be a class of matrices with at most d '1's per row or column, and containing arbitrarily high rank divisions. Then $\mathcal{M}$ contains as submatrices all permutation matrices.*

Proof. Assume that each row and column of M contains at most d '1's, and furthermore that M contains a rank-k division $(\mathcal{R}, \mathcal{C})$. We in fact only need this division to be a grid (each cell contains a '1'). In order to transform it into a permutation matrix, we will first extract a submatrix satisfying the following:

(5.6) in each column c, all the '1's belong to the same block of rows $R \in \mathcal{R}$,

and symmetrically for rows and blocks of columns.

Claim 5.25. There is a submatrix M' of M on which $(\mathcal{R}, \mathcal{C})$ induces a $\frac{k^{1/4}}{(d+1)^{3/2}}$-grid satisfying (5.6).

Proof. Fix $k' = \frac{\sqrt{k}}{d+1}$. Firstly, we will not need most of the blocks of columns from $\mathcal{C}$: we delete all but k' of them. Then, a block $R \in \mathcal{R}$ of rows contains k' cells. In each of these, we pick a '1' ensuring that the cell is non-zero, and we discard the rows of R which do not contain one of the k' picked cells. Thus, we can assume that each block $R \in \mathcal{R}$ contains at most k' rows, hence at most $d \cdot k'$ entries '1'.

Say that two blocks $R, R' \in \mathcal{R}$ are *in conflict* if there is a column c containing both a '1' in R, and one in R', violating (5.6). One can represent conflicts as a graph on vertex set $\mathcal{R}$, whose degree is at most $d^2 \cdot k'$ (each '1' in R can result in at most d conflicts). Thus we have a graph on $|\mathcal{R}| = k$ vertices with degree at most $d^2 \cdot k'$; a greedy process finds an independent set of size $\frac{k}{d^2 \cdot k' + 1}$. If we delete all blocks outside this independent set, we obtain a submatrix satisfying (5.6) for columns and blocks of rows. The number of remaining blocks

of rows is

$$\frac{k}{d^2 \cdot k' + 1} = \frac{k}{\frac{d^2}{d+1}\sqrt{k} + 1} \tag{5.7}$$

$$\geqslant \frac{k}{(d+1)\sqrt{k}} = k' \tag{5.8}$$

Symmetrically, we apply the same process to ensure that (5.6) is also satisfied by rows and blocks of columns. After this second iteration, the number of blocks in the division is

$$\frac{\sqrt{k'}}{d+1} = \frac{k^{1/4}}{(d+1)^{3/2}}. \tag{5.9}$$

■

Thus, from M we obtain a submatrix M' with a k'-grid satisfying (5.6), where k' grows to infinity together with the initial grid size k. We can now extract from M' the permutation matrix of any permutation σ on k' elements. Number the blocks in M' as $R_1, \ldots, R_{k'}$ for the rows and $C_1, \ldots, C_{k'}$ for the columns. For each $i \in [k']$, pick a '1'-entry in $R_{\sigma(i)} \times C_i$, hence k' '1'-entries in total, and delete every row or column which doesn't go through one of these entries. If we rename the remaining rows and columns as $\{1, \ldots, k'\}$, we obtain a '1' at the intersection of column i and row $\sigma(i)$ as desired. Furthermore, note that we extracted exactly one row (resp. column) from each block. Thus (5.6) ensures that in the extracted submatrix, there is no '1' entry besides the previously described ones. That is, the extracted submatrix is the permutation matrix of σ.

We have thus proved that from a matrix M with bounded degree and a k-grid, we can extract the matrix of any permutation on k' elements, with k' an unbounded function of k. If rather than a single matrix M, we start from a class of matrices with bounded degree and arbitrarily large grids, we then obtain all permutation matrices. □

Of course, the former proof heavily uses the assumption that M comes from a bounded degree graph. In general, we cannot hope to always find this kind of permutations matrices: if for instance M were the matrix of the complement of a cubic graph, it would simply have far too few '0's to find any non-trivial permutation matrix. It would however contain the entry-wise complements of permutation matrices, which is just as good for our purpose.

Generalising this, [17] describes a family of six possible encodings of permutations as matrices: the first two are the permutation matrix as described above, and its entry-wise complement. The remaining four are obtained from the permutation matrix by propagating each '1' in one of the four directions, up, down, left, or right. Formally, for a permutation σ on $[n]$, and $s \in \{=, \neq, \leqslant_R, \geqslant_R, \leqslant_C, \geqslant_C\}$, the $n \times n$ matrix $M_s(\sigma)$ is defined to have a '1' at the intersection of column i and row j if and only if the following holds:

$j = \sigma(i)$ when s is '$=$'	$j \neq \sigma(i)$ when s is '$\neq$'
$j \leqslant \sigma(i)$ when s is '$\leqslant_R$'	$j \geqslant \sigma(i)$ when s is '$\geqslant_R$'
$i \leqslant \sigma^{-1}(j)$ when s is '$\leqslant_C$'	$i \geqslant \sigma^{-1}(j)$ when s is '$\geqslant_C$'

Here $M_{=}(\sigma)$ is the permutation matrix, $M_{\neq}(\sigma)$ its complement, while the matrices $M_{\leqslant_R}(\sigma)$, $M_{\geqslant_R}(\sigma)$, $M_{\leqslant_R}(\sigma)$, and $M_{\geqslant_R}(\sigma)$ are obtained by propagating the '1's upwards, downwards, left, and right respectively.

Theorem 5.26 [17]. *If $\mathcal{M}$ is a submatrix-closed class of matrices containing arbitrarily high rank divisions, then for some $s \in \{=, \neq, \leqslant_R, \geqslant_R, \leqslant_C, \geqslant_C\}$ and for all permutations σ, $M_s(\sigma) \in \mathcal{M}$.*

Using Theorem 5.26 and the hardness results on permutations from section 5.4.1, one can prove Theorem 5.19 for ordered graphs.

Theorem 5.27 [17]. *For a hereditary class $\mathcal{C}$ of ordered structures, the following are equivalent:*

1. *$\mathcal{C}$ has bounded twin-width,*
2. *$\mathcal{C}$ is a small class,*
3. *$\mathcal{C}$ has an FPT first-order model checking algorithm, and*
4. *$\mathcal{C}$ is NIP.*

Where equivalence with (3) *is under the $\mathsf{FPT} \neq \mathsf{AW}[*]$ assumption.*

Sketch of proof. We already know that bounded twin-width implies all other three conditions, and must conversely prove that if $\mathcal{C}$ has unbounded twin-width, then it does not satisfy any of them.

Assume thus that $\mathcal{C}$ has unbounded twin-width. By Corollary 5.5, adjacency matrices of ordered graphs in $\mathcal{C}$ have arbitrarily high grid rank. Then, Theorem 5.26 yields some encoding of all permutations: $\sigma \in \mathfrak{S}_n$ is encoded by two sets X, Y of n vertices, such that the adjacency matrix of X and Y is $M_s(\sigma)$ for some $s \in \{=, \neq, \leqslant_R, \geqslant_R, \leqslant_C, \geqslant_C\}$.

Thus $\mathcal{C}$ contains encodings of all $n!$ permutations in $\mathfrak{S}_n$ using only $2n$ vertices ($\mathcal{C}$ being hereditary is used here). This is more than $2^{O(n)}$ non-isomorphic graphs on n vertices, hence $\mathcal{C}$ is not tiny. Since ordered graphs have no non-trivial automorphism, counting graphs with vertices labelled $1, \dots, n$ adds exactly $n!$ choices, hence $\mathcal{C}$ is not small either.

Furthermore, from these encodings of all permutations, one can obtain the class of all biorders through a first-order transduction, which by Lemma 5.21 shows that $\mathcal{C}$ is independent, i.e. (4) does not hold. The key of this transduction is to find the correspondence between column i and row $\sigma(i)$ in $M_s(\sigma)$. When s is $=$ (resp. $\neq$), it is given by the '1's (resp. '0's) in the matrix. For $\leqslant_R$ ($\geqslant_R$), $\sigma(i)$ is the largest (smallest) row intersecting column i at a '1', while similarly with columns for $\leqslant_C$ and $\geqslant_C$. Each of these conditions is expressed by a first-order formula. With the correspondence i–$\sigma(i)$ in hand, one can construct the biorder σ by combining the ordering of columns i with the ordering of the corresponding rows $\sigma(i)$. This defines six first-order transductions Φ_s for $s \in \{=, \neq, \leqslant_R, \geqslant_R, \leqslant_C, \geqslant_C\}$ from matrices to permutations, one for each encoding. If $\mathcal{C}$ has unbounded twin-width, then it contains some encoding of all permutations, hence one of these transductions Φ_s yields all biorders.

Proving that (3) does not hold, i.e. that first-order model checking in $\mathcal{C}$ is $\mathsf{AW}[*]$-complete is more complex: we have a transduction of all graphs from $\mathcal{C}$, but we cannot conclude with Corollary 4.14 since it is not an efficient interpretation. The problem is that the encoding of σ as an ordered graph is only partially defined: the edges between X and Y form the matrix $M_s(\sigma)$, but at no point did we specify the edges inside X, nor inside Y. Thus we cannot

describe an encoding algorithm which given σ finds some ordered graph in $\mathcal{C}$ encoding it, as we do not know which combination of edges inside X and Y yields a graph in $\mathcal{C}$.

The solution is yet another application of Ramsey-like results to extract a simple structure from these unspecified edges. This results in further case disjunctions, and no less than 25 classes of ordered graphs which are obstructions to twin-width, one of which must be contained in $\mathcal{C}$. In each of these 25 classes, there is an efficient interpretation of all graphs, proving that model-checking is $\mathsf{AW}[*]$-hard. We refer the reader to [17] for the details. □

5.4.3 Obstructions in tournaments. We will conclude this section by similarly presenting the obstructions to twin-width in tournaments. The grand lines are the same as in ordered graphs: the obstructions correspond to a few possible encodings of all permutations, and any class of tournaments with unbounded twin-width contains one of these classes of obstructions. The proof involves Theorem 5.26 and other Ramsey-like results, which we will omit to focus on the obstructions themselves.

Thanks to interesting symmetries in tournaments, there are in fact fewer classes of obstructions—that is fewer encodings of permutations—than for matrices or ordered graphs: only three, named $\mathcal{F}_=$, $\mathcal{F}_\leqslant$, and $\mathcal{F}_\geqslant$, where $\mathcal{F}_s$ is the class of encodings $F_s(\sigma)$ for all permutations σ as described below.

Consider $\sigma \in \mathfrak{S}_n$ a permutation on n elements, and s one of the relations $=, \leqslant, \geqslant$. Then the tournament $F_s(\sigma)$ consists of $2n$ vertices, called $X = \{x_1, \dots, x_n\}$ and $Y = \{x_1, \dots, y_n\}$. Each of X, Y induces a transitive tournament with the natural ordering, i.e. there are edges $x_i \to x_j$ and $y_i \to y_j$ whenever $i < j$. The edges between X and Y encode σ as specified by the relation s: there is an edge $y_j \to x_i$ if and only if $i \, s \, \sigma^{-1}(j)$ holds.

Thus in $F_=(\sigma)$ the edges oriented from Y to X form a matching which encodes σ. In $F_\leqslant(\sigma)$ and $F_\geqslant(\sigma)$, these edges form a half-graph which orders X and Y by inclusion of neighbourhoods, so that the order on X is the natural one, and the order on Y encodes σ. Precisely, in $F_\geqslant(\sigma)$, for any $i, j \in [n]$, we have

$$N^-(x_i) \cap Y \subseteq N^-(x_j) \cap Y \iff i \leqslant j \tag{5.10}$$

$$\text{and} \quad N^-(y_i) \cap X \subseteq N^-(y_j) \cap X \iff \sigma^{-1}(i) \leqslant \sigma^{-1}(j), \tag{5.11}$$

while in $F_\leqslant(\sigma)$, the direction of inclusions is reversed. See Figure 5.8 for an example of these encodings.

Lemma 5.28 [51]. *For each $s \in \{=, \leqslant, \geqslant\}$, the class $\mathcal{F}_s$ efficiently interprets the class $\mathcal{B}$ of all permutations represented as bi-orders.*

Precisely, there is an interpretation Φ_s, and for any permutation $\sigma \in \mathfrak{S}_n$, there exists $\sigma' \in \mathfrak{S}_{n+1}$ computable in polynomial time such that $\sigma = \Phi_s(F_s(\sigma'))$.

Proof. We will first show that $F_s(\sigma)$ transduces σ, and then how to remove the coloring step of the transduction by extending σ to a slightly larger σ'.

Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$ be as in the definition of $F_s(\sigma)$. The transduction uses coloring to guess the set X. It then defines two total orderings on Y, which together describe σ. The first ordering is given by the direction of edges inside Y. The second depends on s:

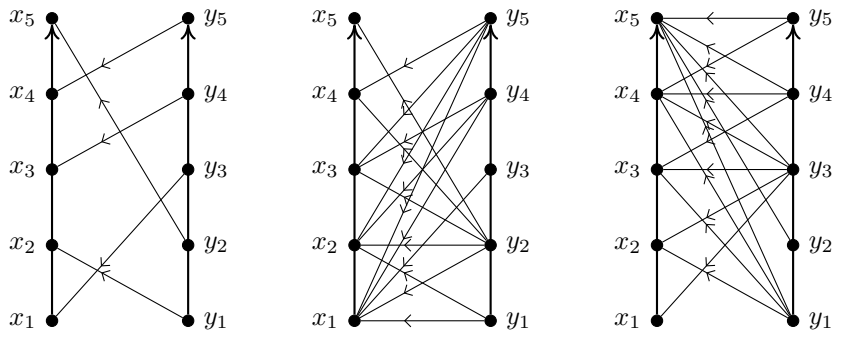


Figure 5.8: The three encodings of permutations (here $\sigma = 31452$) as tournaments. For readability, edges oriented left to right are omitted.

- If s is $=$, edges oriented from Y to X are a perfect matching. The direction of edges in X, interpreted through this matching, defines the second order on Y.
- If s is $\geqslant$ or $\leqslant$, the second order is inclusion, respectively inverse inclusion, of in-neighbourhoods intersected with X, see (5.11).

If we already know which vertices belong to X, it is simple to define these two orderings with a first-order formula. Finally, the transduction deletes vertices of X, leaving only Y and the two orderings which encode σ.

Let us now show how to define the partition X, Y in first-order logic, without the non-deterministic colouring, at the cost of extending σ with one fixed value. Here, we assume $n \geqslant 2$.

- If s is $=$, define $\sigma' \in \mathfrak{S}_{n+1}$ by $\sigma'(n+1) = n+1$ and $\sigma'(i) = \sigma(i)$ otherwise. Then, in $F_=(\sigma')$, the unique vertex with out-degree 1 is y_{n+1}. Its out-neighbour is x_{n+1}, and
$$X = N^-(x_{n+1}) \cup \{x_{n+1}\} \setminus \{y_{n+1}\}.$$
- If s is $\leqslant$, define $\sigma'(1) = n+1$ and $\sigma'(i+1) = \sigma(i)$. Then y_{n+1} is the unique vertex with out-degree 1, and its out-neighbour is x_1, and
$$X = N^+(x_1) \cup \{x_1\}.$$
- If s is $\geqslant$, we once again define $\sigma'(1) = n+1$ and $\sigma'(i+1) = \sigma(i)$. Then x_1 has in-degree 1, and its in-neighbour is y_{n+1}. The only other vertex which may have in-degree 1 is y_1, and this happens if and only if $\sigma'(2) = 1$. When this is the case, the direction of the edge $x_1 \to y_1$ still allows to distinguish x_1 in FO logic. Then, having defined x_1, we obtain y_{n+1} as its in-neighbour, and we have
$$X = N^+(y_{n+1}).$$

In all three cases, $F_s(\sigma')$ contains two extra vertices compared to $F_s(\sigma)$. These extra vertices can be uniquely identified in first-order logic, and can then be used to define X. Combined with the previous argument, this gives an interpretation of σ from $F_s(\sigma')$. □

We can now prove that the classes $\mathcal{F}_s$ are complex.

Theorem 5.29 [51]. *For each $s \in \{=, \leqslant, \geqslant\}$, the class $\mathcal{F}_s$*

1. *is not small,*
2. *has an $\mathsf{AW}[*]$-complete first-order model checking problem, and*
3. *is first-order independent.*

Proof. By Lemma 5.28, all $n!$ permutations on n elements are encoded by tournaments in $\mathcal{F}_s$ on $2n + 2$ vertices, hence there are at least $n!$ non-isomorphic such tournaments. This is more than $2^{O(n)}$ tournaments on n vertices, hence the class $\mathcal{F}_s$ is not tiny. Using the arguments of Lemma 5.28, we can also show that these encodings $F_s(\sigma')$ do not admit any non-trivial automorphism. For tournaments on n vertices, adding labels $1, \ldots, n$ then gives $n!$ additional choices, hence the class has more than $n!2^{O(n)}$ labelled tournaments on n vertices, i.e. is not small.

Finally Lemma 5.28 proves that $\mathcal{F}_s$ efficiently interprets the class of all permutations, and thus of all graphs by Lemma 5.21. Thus $\mathcal{F}_s$ is first-order independent, and has $\mathsf{AW}[*]$-complete first-order model checking by Corollary 4.14. □

Chapter 6

PERMUTATIONS

Permutations are at the origin of twin-width: it is to find patterns in permutations that Guillemot and Marx defined their *width* [61]. Using our knowledge of twin-width developed thus far, we will in this chapter revisit the results of Guillemot and Marx, and a few other classical results on permutations such as the Stanley–Wilf conjecture proved by Marcus and Tardos [73].

We will then prove a decomposition result of Bonnet, Bourneuf, Thomassé, and the author [12]: permutations of bounded twin-width can be written as a bounded composition of permutations with twin-width 0, which are well-known as the *separable* permutations. This result is based on a complex decomposition of structures of bounded twin-width, introduced by Pilipczuk and Sokołowski in [79] and refined by Bourneuf and Thomassé in [23] to prove polynomial χ-boundedness of graphs with bounded twin-width. Finally, we will present some consequences on graphs of this decomposition of permutations: for any k, graphs of twin-width k can be encoded (in the sense of first-order interpretations) by graphs with twin-width bounded by some universal constant.

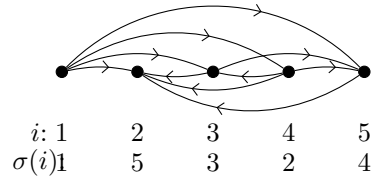
6.1 Encodings of permutations

Before proving any result on permutations, we need to explain how they are encoded. We already introduced in Chapter 5 our preferred encoding of permutations as biorders. However, it is not the only conceivable encoding. In this section, we present a couple of alternatives, and show that they are equivalent to biorders for twin-width, but not for clique-width. These encodings and their comparison are partly inspired by the work of Albert, Bouvel, and Féray [4], which approaches the question from the point of view of first-order logic.

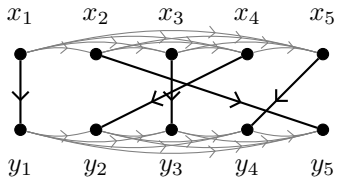
Biorders Recall from Chapter 5 that the permutation $\sigma \in \mathfrak{S}_n$ is encoded as $B_\sigma = ([n], <, <_\sigma)$ where $x <_\sigma y$ if and only if $\sigma(x) < \sigma(y)$, see Figure 6.1a. Notice that any biorder $(X, <_1, <_2)$ is isomorphic to B_σ for some permutation σ: it suffice to identify X with $\{1, \dots, |X|\}$ following the ordering $<_1$. Furthermore, B_σ and B_τ are isomorphic if and only if $\sigma = \tau$: this encoding is a bijection between permutations and biorders considered up to isomorphism.

Composition and inverses of permutations have a natural expression in term of biorders. The inverse is simply obtained by swapping the roles of the two orderings: if B_σ is isomorphic to $(X, <_1, <_2)$, then $B_{\sigma^{-1}}$ is isomorphic to $(X, <_2, <_1)$. Composition is obtained by identifying two of the orderings: if B_σ, B_τ are isomorphic to $(X, <_1, <_2)$ and $(X, <_2, <_3)$ respectively, then $B_{\tau\circ\sigma}$ is isomorphic to $(X, <_1, <_3)$.

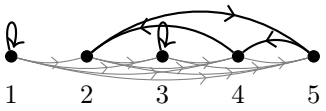
Ordered matching Instead of superposing the two orderings as a biorder, one can separate the domain and image of the permutation, linking them by a matching representing the permutation. Precisely, for $\sigma \in \mathfrak{S}_n$, construct two



(a) The biorder $B_\sigma = ([5], <, <_\sigma)$ (left-to-right ordering $<$ is omitted).



(b) The ordered matching $Mt_\sigma = (X \uplus Y, <, E)$ (order $<$ in gray and edges E in black).



(c) The ordered bijection $Bij_\sigma = ([n], <, E)$ (order $<$ in gray and edges E in black).

Figure 6.1: Three encodings of the permutation $\sigma = 15324$.

copies $X = \{x_1 < \cdots < x_n\}$ and $Y = \{y_1 < \cdots < y_n\}$ of the linear order on n points, and add the edges $E = \{x_i y_{\sigma(i)} \mid i \in [n]\}$. The *ordered matching* is the structure $Mt_\sigma = (X \uplus Y, <, E)$, see Figure 6.1b.

If we consider X as columns and Y as rows, each ordered by $<$, then the adjacency matrix of the relation E is isomorphic to the *permutation matrix* M_σ: the rows and columns are $[n]$, and there is a '1' at position (x, y) if and only if $y = \sigma(x)$.

Ordered bijection Given the ordered matching Mt_σ, if we merge the vertices according to the edges of the matching E, we obtain the biorder B_σ. Instead, one could merge the vertices according to the order: identify x_i with y_i. We call the resulting structure the ordered bijection $Bij_\sigma = ([n], <, E)$ with directed edges $E = \{x\sigma(x) \mid x \in [n]\}$, see Figure 6.1c.

Remark that $A(E, <)$ is once again the permutation matrix M_σ. However, here the columns and rows of the matrix correspond to the same vertices, whereas they were distinct vertex sets in the case of ordered matchings.

Let us now prove that these encodings are equivalent for twin-width.

Theorem 6.1. *For any permutation σ, $\mathrm{tww}(B_\sigma)$, $\mathrm{tww}(Mt_\sigma)$, and $\mathrm{tww}(Bij_\sigma)$ are bounded by functions of each other.*

Proof. Let us start with the two simpler inequalities.

Claim 6.2. $\mathrm{tww}(Mt_\sigma) \leqslant \mathrm{tww}(B_\sigma) + 2$.

Proof. Consider a contraction sequence for $B_\sigma = ([n], <, <_\sigma)$ of width k. We adapt it to $Mt_\sigma = (X \uplus Y, <, E)$ as follows: when merging $i, j \in [n]$ in B_σ we merge x_i with x_j, then $y_{\sigma(i)}$ with $y_{\sigma(j)}$. In the resulting contraction sequence, the error degree with regards to E never exceeds 2. The errors in X (resp. Y) with regards to $<$ correspond to errors in B_σ with regards to $<$ (resp. $<_\sigma$), hence the error degree with regards to $<$ is bounded by 2. ■

Claim 6.3. $\mathrm{tww}(Mt_\sigma) \leqslant 2 \cdot \mathrm{tww}(Bij_\sigma)$.

Proof. Consider a contraction sequence for $Bij_\sigma = ([n], <, E)$ of width k. This time, we adapt it to $Mt_\sigma = (X \uplus Y, <, E)$ as follows: when merging $i, j \in [n]$ in Bij_σ, we merge x_i with x_j, then y_i with y_j. This time, the resulting contraction has error degree at most k with regards to $<$, and at most k with regards to E. ■

We now assume that the ordered matching Mt_σ has small twin-width, and must bound the twin-width of the other two structures. For B_σ, this can be done using a well-chosen refinement of the contraction sequence, using techniques developed for error rank in Chapter 3.

Claim 6.4. $\mathrm{tww}(B_\sigma) \leqslant 4(\mathrm{tww}(Mt_\sigma) + 1)^3$.

Proof. Consider the ordered matching $Mt_\sigma = (X \uplus Y, <, E)$, and a contraction sequence $\mathcal{P}_{2n}, \dots, \mathcal{P}_1$ of width k. Define the projection of a part $P \in \mathcal{P}_i$ on X and Y as

$$P^X = \{i \mid x_i \in P\} \quad \text{and} \quad P^Y = \left\{i \mid y_{\sigma(i)} \in P\right\}$$

(note the permutation σ for P^Y). Call $\mathcal{P}_i^X = \left\{P^X \mid P \in \mathcal{P}\right\}$ the projection of $\mathcal{P}$ on X, and similarly $\mathcal{P}_i^Y$.

If we see $\mathcal{P}_i^X$ as a partition of the vertex set of $B_\sigma = ([n], <, <_\sigma)$, then the error degree of $\mathcal{P}_i^X$ with regards to $<$ is at most k: any error for $<$ in B_σ directly translates into an error in Mt_σ Note however that $\mathcal{P}_i^X$ may have large error degree with regards to $<_\sigma$. Similarly, $\mathcal{P}_i^Y$ has error degree at most k with regards to $<_\sigma$.

For $P_1, P_2 \in \mathcal{P}_i$, if P_1^X intersects P_2^Y, then there is an edge of E between P_1 and P_2 in Mt_σ, which implies that they are in error or equal, unless they are both singletons. It follows that any part of $\mathcal{P}_i^X$ intersects at most $k+1$ parts of $\mathcal{P}_i^Y$, and vice-versa. Define now the common refinement of $\mathcal{P}_i^X$ and $\mathcal{P}_i^Y$

$$\mathcal{Q}_i = \left\{P \cap P' \mid P \in \mathcal{P}_i^X, P' \in \mathcal{P}_i^Y\right\}.$$

It follows from the previous remarks that $\mathcal{Q}_i$ is a $(k+1)$-refinement of $\mathcal{P}_i^X$ and $\mathcal{P}_i^Y$.

Since $\mathcal{Q}_i$ $(k+1)$-refines $\mathcal{P}_i^X$ which has error degree k for $<$, by Lemma 3.15, $\mathcal{Q}_i$ has error degree at most $(k+1)^2$ for $<$. For the same reason using $\mathcal{P}_i^Y$, $\mathcal{Q}_i$ has error degree at most $(k+1)^2$ for $<_\sigma$, hence error degree at most $2(k+1)^2$ in total.

Finally, note that $\mathcal{Q}_{2n}$ must be the partition into singletons, $\mathcal{Q}_1$ the trivial partition, and that $\mathcal{Q}_{i+1}$ refines $\mathcal{Q}_i$. Also, $\mathcal{Q}_{i+1}$ $2(k+1)$-refines $\mathcal{P}_i$, hence it a fortiori $2(k+1)$-refines $\mathcal{Q}_i$. Under these conditions, Lemma 3.16 allows to complete $\mathcal{Q}_{2n}, \dots, \mathcal{Q}_1$ into a contraction sequence of width at most $4(k+1)^3$ ■

Finally, we need to bound $\mathrm{tww}(Bij_\sigma)$. Here, we rely on the characterisation of twin-width with grid rank.

Claim 6.5. $\mathrm{tww}(Bij_\sigma) \leqslant f(\mathrm{tww}(Mt_\sigma))$ for some function f.

Proof. Consider a contraction sequence for $Mt_\sigma = (X \uplus Y, <, E)$. Remark that if $\mathcal{P}$ is a partition of $X \uplus Y$ and $P \in \mathcal{P}$ intersects both X and Y, then P is in error with *every* part of $\mathcal{P}$ due to $<$. Let us slightly tweak Mt_σ to make it an ordered structure: we extend $<$ by $X < Y$ into a total ordering of $X \uplus Y$. The previous remark implies that this does not increase the twin-width.

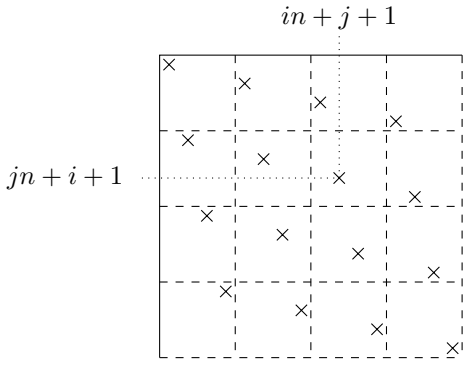


Figure 6.2: Matrix of a permutation on $[n^2]$ with an n-grid.

We now apply Theorem 3.9 and find that the adjacency matrix $A(E, <)$ has bounded grid rank. Notice that $A(E, <)$ contains the permutation matrix M_σ as submatrix.

Consider now the ordered bijection $Bij_\sigma = ([n], <, E')$. Recall that its adjacency matrix $A(E', <)$ is exactly the permutation matrix M_σ. Theorem 3.9 thus gives a bound on $\mathrm{tww}(Bij_\sigma)$. ■

This completes the proof of equivalence of the three encodings of permutations. □

Theorem 6.1 tells us that for a class of permutations, having bounded twin-width is meaningful independently of the choice of encoding. Furthermore, we have implicitly found a characterisation of twin-width through grids in the permutation matrices. Let us restate it:

Theorem 6.6 [61]. *A class $\mathcal{C}$ of permutations has bounded twin-width if and only if the class of permutation matrices $\{M_\sigma \mid \sigma \in \mathcal{C}\}$ does not admit arbitrarily large grids.*

Proof. Using the ordered bijection encoding, we find that σ has bounded twin-width if and only if M_σ has bounded grid rank. For permutation matrices, having bounded grid rank is equivalent to avoiding grids (in the sparse sense of section 3.1) of some size. Indeed, a rank-k division is always also a k-grid, and a k^2-grid in a permutation matrix yields a rank-k division by regrouping blocks of the grid k by k. □

One can construct permutations of large twin-width using Theorem 6.6: it suffice to ensure that they contain a large grids. Given $n \in \mathbb{N}$, consider the permutation $\sigma \in \mathfrak{S}_{n^2}$ defined by $\sigma(in+j+1) = jn+i+1$ for any $i, j < n$. Then M_σ contains an n-grid (see Figure 6.2), hence the twin-width σ is unbounded as n increases.

To conclude this section, let us show that the equivalence between biorder and ordered matchings goes beyond twin-width: they are equivalent up to first-order transductions. By contrast, ordered bijections are more expressive for first-order logic: no fixed transduction can reconstruct Bij_σ from B_σ for arbitrary σ. Intuitively, the reason is that Bij_σ gives access the composition $\sigma \circ \sigma$:

given $i \in [n]$, it suffice to follow the edges of E twice to find $\sigma(\sigma(n))$ in Bij_σ, which is not possible in B_σ or Mt_σ.

Fact 6.7. *There are first-order transductions which for any permutation σ, compute B_σ when given Mt_σ, and vice versa.*

Proof. Reconstructing B_σ from Mt_σ is simple: identify i in B_σ with $x_i \in Mt_\sigma$. Now given $i, j \in V(B_\sigma)$, the ordering $i < j$ coincides with the ordering in Mt_σ. Further, following the edges of the matching in Mt_σ leads to $y_{\sigma(i)}, y_{\sigma(j)}$, and we have $i <_\sigma j$ in B_σ if and only if $y_{\sigma(i)} < y_{\sigma(j)}$ in Mt_σ. All of the above is easily expressed using first-order formulæ.

To conversely reconstruct Mt_σ from B_σ, we need the extended notion of transductions with *copying*: given B_σ, this operation makes two copies of B_σ and adds a matching joining each vertex x in B_σ to its new copy. The result is almost exactly Mt_σ: all that is left to do is to forget the ordering $<_\sigma$ in the original copy, which becomes X, and forget $<$ in the new copy, which becomes Y. □

Fact 6.7 and Theorem 4.16 give a new proof of the equivalence of biorders and ordered matchings as encodings of permutations for twin-width, although with worse bounds than Theorem 6.1. This is not exclusive to twin-width: clique-width can be generalised to relational structures, and is stable under first-order (and even monadic second-order) transductions. Thus the clique-width of B_σ and of Mt_σ are bounded by functions of each other.

The same is not true of Bij_σ:

Fact 6.8. *There are permutations σ such that B_σ has bounded clique-width and Bij_σ has arbitrarily large clique-width.*

Sketch of proof. For any $n \in \mathbb{N}$, let $\sigma \in \mathfrak{S}_n$ be the permutation which adds $\sqrt{n}$ modulo n.

In B_σ, denote $X = \{1, \dots, n - \sqrt{n}\}$ and $Y = \{n - \sqrt{n} + 1, \dots, n\}$. Both inside X and inside Y, the orders $<$ and $<_\sigma$ coincide. Furthermore, $X < Y$ and $X >_\sigma Y$. This implies that the clique-width of B_σ is 2.

In Bij_σ, replace the ordering $<$ by a Hamiltonian path joining each vertex to its successor: this is a first-order interpretation. In the resulting graph, vertex i has edges to $i - \sqrt{n}$, $i - 1$, $i + 1$, and $i + \sqrt{n}$ modulo n: this is the $\sqrt{n} \times \sqrt{n}$ grid, with some additional edges connected each border to the opposite one. Regardless of these edges, the clique-width of this graph is known to be in the order of $\sqrt{n}$. Since this graph was obtained by first-order interpretation from Bij_σ, it follows that Bij_σ has unbounded clique-width. □

Since B_σ and Bij_σ are not equivalent for clique-width, they cannot be equivalent up to first-order transductions:

Corollary 6.9. *There is no first-order transduction which for any permutations σ, given B_σ, constructs Bij_σ.*

6.2 Patterns, substitutions, and separability

We will now focus on the twin-width of a number of constructions and classes of permutations. Some of the following results involve the precise value of twin-width and not just its asymptotic behaviour, thus we need to fix the encoding:

by abuse of notation, we identify a permutation σ with its biorder B_σ, and will thus write e.g. $\mathrm{tww}(\sigma)$ for $\mathrm{tww}(B_\sigma)$.

6.2.1 Patterns. Given a biorder $B = (X, <_1, <_2)$, one can for any subset $Y \subset X$ of vertices consider the induced substructure $B[Y] = (Y, <_1, <_2)$. If B and $B[Y]$ represent permutations σ and τ respectively, we say that τ is a *pattern* σ. An alternative characterisation is through permutation matrices: τ is a pattern of σ if and only if M_τ is a submatrix of M_σ.

Notice that this notion, while very natural when expressed with biorders, is less convenient to define in term of the bijection $\sigma : [n] \to [n]$: given any subset $Y \subset [n]$, one can consider the restriction of σ to Y. Its image $\sigma(Y)$ is in general not equal to Y. To reconstruct a permutation from this restriction, one would need to 'compact' Y and independently $\sigma(Y)$ into an interval $[k]$, preserving only the relative order of elements.

Since a pattern is a substructure, we immediately obtain:

Lemma 6.10. *If τ is a pattern of σ, then* $\mathrm{tww}(\tau) \leqslant \mathrm{tww}(\sigma)$.

Given a permutation τ, one can consider the class $\mathcal{F}(\tau)$ of permutations *avoiding* τ: $\sigma \in \mathcal{F}(\tau)$ if and only if τ is *not* a pattern of σ. This is very similar to defining a class of graphs by forbidding an induced subgraph, such as the cographs which can be characterised as the graphs without the path P_4 as induced subgraph.

In general, classes of graphs avoiding a given induced subgraph may be very complex: while $\mathcal{F}(H)$ might be simple for specific choices of H (e.g. the cographs $\mathcal{F}(P_4)$), one could hardly expect the class $\mathcal{F}(H)$ to satisfy many interesting properties for arbitrary H.[1]

By contrast, when considering permutations, the class $\mathcal{F}(\tau)$ is relatively simple for any pattern τ. For example, the motivation of the Marcus–Tardos theorem (Theorem 3.3) was to show that $\mathcal{F}(\tau)$ is small for any permutation τ, (i.e. has at most c^n permutations on n elements for some c function of τ), a result known as the Stanley–Wilf conjecture. Guillemot and Marx main result in [61] is that for any fixed τ, one can recognise permutations in $\mathcal{F}(\tau)$ in linear time. Neither of these results hold when replacing permutations with graphs.

With the knowledge of twin-width, these results on pattern-avoiding classes are explained by the following key remark of Guillemot and Marx:

Theorem 6.11 [61]. *A class $\mathcal{C}$ of permutations has bounded twin-width if and only if $\mathcal{C} \subset \mathcal{F}(\tau)$ for some τ.*

This is based on the existence of universal permutations:

Lemma 6.12. *Let σ be a permutation whose matrix M_σ contains a k-grid. Then any $\tau \in \mathfrak{S}_k$ is a pattern of σ.*

Proof. Say that $\sigma \in \mathfrak{S}_n$. A k-grid in M_σ is a pair $(\mathcal{R}, \mathcal{C})$ of partitions of $[n]$ into k intervals each, $\mathcal{R} = \{R_1 < \cdots < R_k\}$ and $\mathcal{C} = \{C_1 < \cdots < C_k\}$ such that for all $i, j \in [k]$, there is $x \in C_i$ such that $\sigma(x) \in R_j$.

Consider now some $\tau \in \mathfrak{S}_k$. For each $i \in [k]$, choose $x_i \in C_i$ such that $\sigma(x_i) \in R_{\tau(j)}$. Then $\sigma(x_i) < \sigma(x_j)$ if and only if $\tau(i) < \tau(j)$. Thus, restricting σ to $\{x_1, \ldots, x_k\}$ yields τ as a pattern. □

[1] The famous Erdős–Hajnal Conjecture is a notable exception.

From Lemma 6.12 and Theorem 6.6, we obtain the equivalence between bounded twin-width and avoiding patterns.

Proof of Theorem 6.11. Assume that $\mathcal{C}$ has bounded twin-width. Thanks to Lemma 6.10, we can also assume $\mathcal{C}$ to be closed under patterns. Since $\mathcal{C}$ has bounded twin-width, it cannot contain all permutations (see the example of Figure 6.2). Thus $\mathcal{C}$ avoids τ, for any $\tau \notin \mathcal{C}$.

Conversely, if $\mathcal{C}$ avoids τ, by Lemma 6.12, the matrix M_σ has no k-grid for any $\sigma \in \mathcal{C}$, where k is the size of τ. By Theorem 6.6, this implies that $\mathcal{C}$ has bounded twin-width. □

With Theorem 6.11, we can restate the Stanley–Wilf conjecture as: any class of permutations with bounded twin-width is small. This is Theorem 3.37. The Guillemot–Marx pattern recognition algorithm can similarly be obtained from the results on twin-width we have presented.

Theorem 6.13 [61]. *Given permutations σ, τ, there is an* FPT *algorithm to test if τ is a pattern of σ, with $|\tau|$ as parameter.*

Proof. Given σ, which is an ordered structure, one first computes an approximation of twin-width using Corollary 5.4. We know that $\mathcal{F}(\tau)$ has bounded twin-width, hence if this approximation of $\mathrm{tww}(\sigma)$ is too large, τ must be a pattern of σ. Thus we can assume that we have computed a contraction sequence for σ of width bounded by a function of τ.

This contraction sequence can be used to test if τ is a pattern of σ with FPT complexity: this is a variant of the independent set algorithm (Theorem 2.21), and a special case of the first-order model checking algorithm (Theorem 4.15). Indeed for any fixed relational structure S, there is a first-order formula ϕ_S which expresses the presence of S as induced substructure.

We can thus test whether τ is a pattern of σ with FPT complexity, parameterised by the width of the contraction sequence and the size of ϕ_τ, both of which are functions of τ only. □

6.2.2 Separable permutations. We will now focus on a particularly important class of permutations: separable permutations. They very closely resemble cographs (cf. section 2.2.2), as both cographs and separable permutations can be characterised either as constructed through a pair of operations, or by forbidding some simple induced substructure, and both coincide with twin-width 0.

Consider two biorders $B = (X, <_1, <_2)$ and $B' = (X', <'_1, <'_2)$. Their *direct sum* $B \oplus B'$ is the biorder obtained from the disjoint union of B and B' by placing X before X' for both orderings. Their *skew sum* $B \otimes B'$ is obtained similarly, except that X is placed before X' in the first ordering, but after X' in the second, A permutation is called *separable* if it can be constructed by direct and skew sums starting from the one-element permutation $1 \mapsto 1$. We denote by $\mathcal{S}$ the class of separable permutations. See Figure 6.3 for an example.

A folklore result is that separable permutations can be characterised by excluding just two patterns.

Fact 6.14 [22, Lemma 5]. *The permutation σ is separable if and only if neither* 3142 *nor* 2413 *is a pattern of σ.*

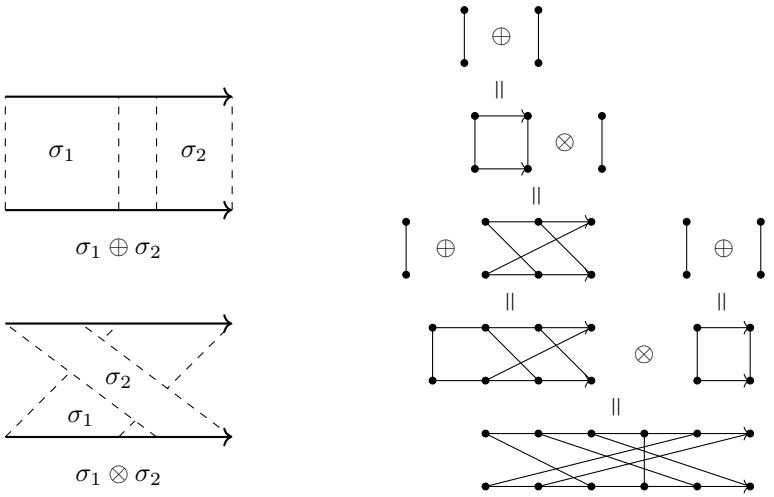


Figure 6.3: Left: visualisation of direct and skew sum of permutations σ_1, σ_2. Right: steps in the construction of the separable permutation 356412 by direct and skew sums. Permutations are drawn as ordered matchings.

Furthermore, Guillemot and Marx noticed that separable permutations can be characterised using twin-width.

Fact 6.15 [61, Proposition 3.3]. *A permutation σ is separable if and only if* $\mathrm{tww}(\sigma) = 0$.

Proof. Firstly, one may notice that neither 3142 nor 2413 contains twins, hence by Lemma 2.1, their twin-width cannot be 0. Thus, any non-separable permutation has twin-width at least 1 since it contains the pattern 3142 or 2413.

To show that conversely separable permutations have twin-width 0, it suffice to show that direct and skew sums preserve twin-width. Consider arbitrary permutations σ, τ, and the biorder $B = ([n], <_1, <_2)$ representing their direct sum. The domain is partitioned as $[n] = X \uplus Y$ with $X <_1 Y$ and $X <_2 Y$ so that $B[X]$ represents σ while $B[Y]$ represents τ. Now given a contraction sequence for σ, we can simply replicate it in B inside X without creating errors with Y. Then, one can similarly replicate a contraction sequence for τ inside Y, and finally contract X with Y. This implies that

$$\mathrm{tww}(\sigma \oplus \tau) = \max(\mathrm{tww}(\sigma), \mathrm{tww}(\tau)). \tag{6.1}$$

The same proof also works for the skew sum. □

6.2.3 Substitution. Let us generalise the direct and skew sums. Given two biorders $B = (X, <_1, <_2)$ and $B' = (X', <'_1, <'_2)$, and a vertex $x \in B$, the *substitution of x by B' in B* is the biorder obtained from B as follows: x is deleted and replaced with $V(B')$, and for in both orderings $<_i$, $i = 1, 2$, the newly added vertices of $V(B')$ form an interval at the old position of x, while keeping the ordering given by $<'_i$ between vertices of $V(B')$.

Given permutations σ_1, σ_2, the substitution of 1 by σ_1 and of 2 by σ_2 in 12 (the identity permutation on two elements) yields the direct sum $\sigma_1 \oplus \sigma_2$.

Similarly, substituting i by σ_i in 21 yields the skew sum $\sigma_1 \ominus \sigma_2$. This allows to reformulate the definition of separability: a permutation is separable if and only if it can be obtained by a sequence of substitutions, starting from the permutations 12 and 21.

More generally, given any class $\mathcal{C}$ of permutation, the *closure* of $\mathcal{C}$ under substitution, denoted $\mathcal{C}^*$, is the smallest class containing $\mathcal{C}$ closed under substitution (i.e. the result of any substitution of permutations in $\mathcal{C}^*$ is itself in $\mathcal{C}^*$). Thus, the separable permutations are $\mathcal{S} = \{12, 21\}^*$.

In the proof of Fact 6.15, we showed that direct and skew sums preserve twin-width. A direct generalisation of this argument gives:

Lemma 6.16 [61, Proposition 3.4]. *If σ is obtained by substitution from permutations σ_1, σ_2, then* $\mathrm{tww}(\sigma) = \max(\mathrm{tww}(\sigma_1), \mathrm{tww}(\sigma_2))$.

In particular, for any class $\mathcal{C}$ of permutations, $\mathcal{C}^*$ has bounded twin-width if and only if $\mathcal{C}$ does, and with the same bound.

6.2.4 Substitution trees. The construction of a permutation $\sigma \in \mathcal{C}^*$ by a sequence of substitutions starting from $\mathcal{C}$ can be described as a tree. These *substitution trees* will play a significant role latter in this chapter, which is why we take some time to introduce some terminology.

We work with rooted trees. Vertices of a trees are called *nodes*. The *ancestors* of a node x are the nodes in the unique path from x to the root r, and the *parent* of x is the first node on this path. We will also speak of *descendants* (inverse of ancestors), *children* (inverse of parent), and later in this chapter *grandchildren* (children of children), *siblings* (nodes with same parent), and *cousins* (non-sibling nodes with the same grandparent). The set of leaves of a tree T is denoted by $L(T)$, while the internal nodes are $I(T) = V(T) \setminus L(T)$. For any node $t \in T$, we denote by $T(t)$ the subtree rooted at t (i.e. consisting of the descendants of t), and by $L(t) \subseteq L(T)$ the set of leaves of $T(t)$.

An *ordered tree* $(T, <)$ is a rooted tree T equipped with a linear order $<$ on $L(T)$, such that for each node $t \in T$, the leaves $L(t)$ form an interval of $<$. In that case we also say that $<$ is *compatible* with T. One can think of $<$ as a *left-to-right* order. If $x, y \in T$ are not in an ancestor or descendant of each other, then $L(x), L(y)$ are disjoint interval of $<$, hence either $L(x) < L(y)$, or $L(y) < L(x)$. We then say that $x < y$ or $y < x$ respectively. Thus $<$ induces a total order on the children of any given internal node t. This process can be reversed. Given an ordering $<_t$ of the children of t for each internal node t, we define $<$ compatible with T as follows: for any leaves x, y, we have $x < y$ iff $x' <_t y'$, where t is the last (that is, closest) common ancestor of x, y, and x' and y' are the children of t which are ancestors of x and y respectively.

We can now redefine substitutions: the idea is to find a tree compatible with both orderings of a biorder. Consider T a tree and $B = (L(T), <, \prec)$ a biorder on its leaves such that both $<$ and $\prec$ are compatible with T. Then for every internal node $t \in T$, $<$ (resp. $\prec$) induces an ordering $<_t$ (resp. $\prec_t$) on the children of t, thus yielding a family of biorders $\{B_t = (<_t, \prec_t)\}_{t \in I(T)}$. Then B can be constructed by a sequence of substitutions starting from $\{B_t\}_{t \in I(T)}$. We say that B is obtained by *substitution of* $\{B_t\}_{t \in I(T)}$ *along* T. One may now check that a permutation σ is in the substitution closure $\mathcal{C}^*$ if and only if σ can be obtained by substitution of permutations in $\mathcal{C}$ along some tree.

6.3 Composition and decomposition

Patterns and substitutions are, as we saw in Lemmas 6.10 and 6.16, particularly well behaved with regards to twin-width. We will now add to our library a third and most natural operation for permutations: composition, in the sense of functions. Its relation to twin-width and patterns is more complex, but no less interesting.

Let us start with simple lemmas relating composition, patterns, and substitution.

Lemma 6.17. *If τ is a pattern of $\sigma_1 \circ \sigma_2$, then there exists τ_i pattern of σ_i for $i = 1, 2$ such that $\tau = \tau_1 \circ \tau_2$.*

Proof. Represent σ_1, σ_2 as biorders $(X, <_2, <_3)$ and $(X, <_1, <_2)$ on the same vertex set. Then $\sigma_1 \circ \sigma_2$ is represented by $(X, <_1, <_3)$. Now let $Y \subset X$ induce the pattern τ in $\sigma_1 \circ \sigma_2$. Choosing τ_i to be the pattern of σ_i induced by Y yields the result. □

Lemma 6.18. *If $\mathcal{C}, \mathcal{D}$ are substitution-closed classes of permutations, then so is the class of their compositions $\mathcal{C} \circ \mathcal{D}$.*

Proof. Consider a substitution tree T with for each internal node t a permutation $\sigma_t \circ \tau_t$ on its children, with $\sigma_t \in \mathcal{C}$ and $\tau_t \in \mathcal{D}$. Then we can also consider T as substitution tree with permutation σ_t on the children of t: this yields a permutation σ on $L(T)$, and $\sigma \in \mathcal{C}$ since $\mathcal{C}$ is substitution closed. We similarly obtain $\tau \in \mathcal{D}$ when equipping the children of t with τ_t. Then, it is simple to verify that the permutation corresponding to the original substitution tree (with $\sigma_t \circ \tau_t$) is $\sigma \circ \tau \in \mathcal{C} \circ \mathcal{D}$. □

In Chapter 5, we saw that composition of permutations preserves bounded twin-width (but not the exact bound), because it can be expressed as glueing two structures along a total ordering.

Lemma 6.19 (Corollary 5.7 restated). *There is a function f such that for any permutations $\sigma_1, \sigma_2 \in \mathfrak{S}_n$, $\operatorname{tww}(\sigma_1 \circ \sigma_2) \leqslant f(\operatorname{tww}(\sigma_1), \operatorname{tww}(\sigma_2))$.*

Lemma 6.19 is a remarkably simple statement, yet to our best knowledge it has no direct proof. Through Theorem 6.11, it can also be restated in terms of patterns. Under this form, an alternative proof by enumeration can be deduced from Marcus and Tardos proof of the Stanley–Wilf conjecture [73].

Lemma 6.20 (Corollary 5.7 re-restated). *For any patterns τ_1, τ_2, there is some τ such for any given $\sigma_1 \in \mathcal{F}(\tau_1)$, $\sigma_2 \in \mathcal{F}(\tau_2)$, we have $\sigma_1 \circ \sigma_2 \in \mathcal{F}(\tau)$.*

Proof. Define $\mathcal{C} = \mathcal{F}(\tau_1) \circ \mathcal{F}(\tau_2)$ the class of all compositions $\sigma_1 \circ \sigma_2$ of permutations $\sigma_i \in \mathcal{F}(\tau_i)$ (with σ_1, σ_2 of the same size). By the Stanley–Wilf conjecture (or Theorem 3.37), the class $\mathcal{F}(\tau_i)$ is small: there is a constant c_i such that there are at most c_i^n permutations of size n in $\mathcal{F}(\tau_i)$. It follows that there are at most $(c_1 c_2)^n \ll n!$ permutations of size n in $\mathcal{C}$, hence $\mathcal{C}$ does not contain all permutations. By Lemma 6.17, $\mathcal{C}$ is closed under patterns, hence choosing $\tau \notin \mathcal{C}$ gives $\mathcal{C} \subseteq \mathcal{F}(\tau)$ as desired. □

Suppose now that we start from the very simple class $\mathcal{S}$ of separable permutations. By Lemma 6.19, for any fixed k, the class $\mathcal{S}^k$ of compositions of k separable permutations has bounded twin-width. The main result of this

chapter, whose proof spans the next three sections, is that the converse also holds.

Theorem 6.21 [12]. *For any twin-width bound $t \in \mathbb{N}$, there is a length $k \in \mathbb{N}$ such that any permutation σ with $\mathrm{tww}(\sigma) \leqslant t$ factorises as a composition $\sigma = \sigma_1 \circ \cdots \circ \sigma_k$ with σ_i separable.*

Equivalently, a class $\mathcal{C}$ of permutations has bounded twin-width (or avoids some pattern) if and only if it is contained in $\mathcal{S}^k$ for some $k \in \mathbb{N}$.

The general strategy of the proof of Theorem 6.21 is based on colouring results for graphs of bounded twin-width by Pilipczuk and Sokołowski [79], and Bourneuf and Thomassé [23]. It proceeds by induction on a parameter functionally equivalent to twin-width (a variant of grid rank): a given permutation σ is decomposed into permutations which are strictly simpler for this parameter. The decomposition relies on more abstract operations which we show can themselves be expressed as compositions of separable permutations.

We will conclude this section with a first simple example of a high level operation, *shuffles*, which decomposes into separable permutations. A permutation $\sigma \in \mathfrak{S}_n$ is a *k-shuffle* if there is some partition $\biguplus_{i=1}^{k} X_i$ of its domain $[n]$ such that the restriction (in the sense of patterns) of σ to any X_i is the identity permutation. More generally, given a class $\mathcal{C}$ of permutations, σ is a *k-shuffle of $\mathcal{C}$* if there is a partition $\biguplus_{i=1}^{k} X_i = [n]$ such that the restriction of σ to any X_i is in $\mathcal{C}$.

Let us say that a class $\mathcal{C}$ of permutations is *closed under symmetry* if it is closed by conjugating with the decreasing permutation: if $\sigma \in \mathcal{C}$, then the permutation $i \mapsto n+1-\sigma(n+1-i)$ should also be in $\mathcal{C}$. In terms of biorders, this corresponds to replacing each of the two linear orders by its dual, i.e. its reverse order. Remark that if $\mathcal{C}, \mathcal{D}$ are closed under symmetry, then so is $\mathcal{C} \circ \mathcal{D}$. For example, the class $\mathcal{S}$ of separable permutations is closed under symmetry.

Lemma 6.22. *For a class $\mathcal{C}$ of permutations closed under substitution and symmetry, any 2^k-shuffle of $\mathcal{C}$ is in $\mathcal{S}^k \circ \mathcal{C} \circ \mathcal{S}^k$. In particular, any 2^k-shuffle is in $\mathcal{S}^{2k}$.*

Proof. Let $\sigma = (X, <_1, <_2)$ be a 2^k-shuffle of $\mathcal{C}$. Then there is a partition of its domain $X = Y \uplus Z$ such that the restrictions of σ to both Y and Z are 2^{k-1}-shuffles of $\mathcal{C}$.

Define an intermediate linear order $\prec_1$ by

- $Y \prec_1 Z$,
- inside Y, $\prec_1$ coincides with $<_1$, and
- inside Z, $\prec_1$ coincides with the dual of $<_1$.

We claim that $(X, <_1, \prec_1)$, is separable. Indeed, let x be the minimum of X for $<_1$. If $x \in Y$, then x is the minimum of $\prec_1$, while if $x \in Z$ it is the maximum of $\prec_1$. Either way, we can separate X into $\{x\}$ and $X \setminus \{x\}$, and proceed by induction on the latter. This scheme is a decomposition of $(X, <_1, \prec_1)$ as a separable permutation, with the specificity that the separating tree is a caterpillar. We define $\prec_2$ similarly with regards to $<_2$, so that $(X, \prec_2, <_2)$ is separable.

Consider now the permutation $(X, \prec_1, \prec_2)$. Since we have $Y \prec_i Z$ for both $i = 1, 2$, it suffices to consider this permutation restricted to either Y or Z. By construction, the restriction to Y is exactly $(Y, <_1, <_2)$, and the

restriction to Z is $(Z, <_1, <_2)$ up to symmetry. These two permutations are 2^{k-1}-shuffles of $\mathcal{C}$, hence by induction are in $\mathcal{S}^{k-1} \circ \mathcal{C} \circ \mathcal{S}^{k-1}$. Since this class is closed under symmetry and substitution, we obtain that $(X, \prec_1, \prec_2)$ is also in $\mathcal{S}^{k-1} \circ \mathcal{C} \circ \mathcal{S}^{k-1}$, and conclude by composing with $(X, <_1, \prec_1)$ and $(X, \prec_2, <_2)$.

Finally, the remark that 2^k-shuffles are in $\mathcal{S}^{2k}$ is obtained by applying the result with $\mathcal{C}$ the class of identity permutations. □

6.4 Delayed substitutions

We now begin the proof of Theorem 6.21, with arguably the most important step of the decomposition: *delayed substitutions*, which as the name suggests generalise the substitution trees of section 6.2.4.

The technique of delayed substitution was introduced by Bourneuf and Thomassé to prove that classes of graphs with bounded twin-width are polynomially χ-bounded [23, Section 2]. We will see that for permutations, these delayed substitutions can be expressed as a bounded product of substitutions.

6.4.1 Definition. A *delayed structured tree* $(T, <, \{\prec_t\}_{t \in T})$ consists of an ordered tree $(T, <)$, equipped with, for each node $t \in T$, a linear order $\prec_t$ on the *grandchildren* of t. This is analogous to the trees describing substitutions, except that $\prec_t$ is defined on the grandchildren instead of the children, hence 'delayed'. We add the technical requirement that each leaf is a single child (with no siblings), so that whenever $x \neq y$ are leaves, their closest ancestor is at distance at least 2.

The *realisation* of this delayed structured tree is the structure $(L(T), <, \prec)$, where for two leaves x, y, we have $x \prec y$ if and only if $x' \prec_t y'$, where t is the closest ancestor of x, y, and x', y' are the grandchildren of t which are ancestors of x, y respectively. In general, this binary relation $\prec$ is not a linear ordering, as it might not be transitive. We call the delayed structured tree *well-formed* if $\prec$ is a linear order, so that the realisation is a permutation on $L(T)$. We will only consider well-formed delayed structured trees. Remark that construction only uses the ordering $\prec_t$ between cousins: the ordering between siblings is irrelevant.

We say that the permutation $(<, \prec)$ is obtained by *delayed substitution* from the permutations $\{(<_t, \prec_t)\}_{t \in T}$, understood as permutations on the grandchildren of t. If for each $t \in T$ the permutation $(<_t, \prec_t)$ on its grandchildren is in a given class $\mathcal{C}$, we also say that the delayed structured tree T is *labelled* with permutations in $\mathcal{C}$, and that the permutation $(<, \prec)$ is obtained by *delayed substitution from* $\mathcal{C}$.

6.4.2 Distinguishability. The linear order $\prec$ is defined on the leaves of T. We extend it to a partial order on all nodes of T where $x \prec y$ iff $L(x) \prec L(y)$, i.e. all descendants of x are strictly before all descendants of y for $\prec$.

Remark 6.23. Let x, y be nodes in T, and t their least common ancestor. If t is at distance at least 2 of both x and y, then x, y are comparable under $\prec$, meaning either $x \prec y$ or $y \prec x$. In particular, if x, y are at the same level in T but are not siblings, then they are comparable by $\prec$.

Being incomparable by $\prec$ is not an equivalence relation, even when restricted to siblings: one may have three siblings x, y, z such that $x \prec z$,

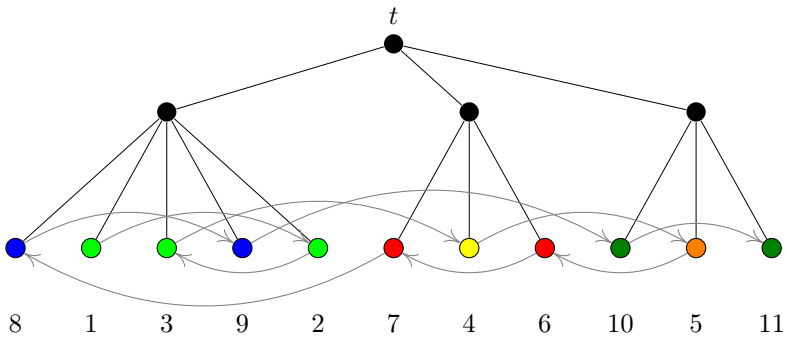


Figure 6.4: Indistinguishability for the grandchildren of a node t. The values and arrows represent the linear order $\prec_t$, and colours represent equivalence classes of indistinguishability.

but $L(x), L(y)$, resp. $L(y), L(z)$ are interleaving for $\prec$. We can however define an equivalence relation by considering how other nodes can separate siblings.

Consider x, y two children of a node t, and $v \in L(T) \setminus L(t)$. By Remark 6.23, v is comparable with both x and y under $\prec$. We say that x and y are *distinguished* by v if $x \prec v \prec y$ or $y \prec v \prec x$. One can check that if $v \in L(T) \setminus L(t)$ distinguishes x, y, it must be a descendant of a cousin of x and y. Finally, we call x, y *indistinguishable*, denoted $x \sim y$, if x, y are siblings with parent t, and no $v \in L(T) \setminus L(t)$ distinguishes x and y. See Figure 6.4 for an example.

Lemma 6.24. *Indistinguishability is an equivalence relation.*

Proof. Clearly $\sim$ is reflexive and symmetric. Let x, y, z be siblings with parent t, and suppose that v distinguishes x, z, say $x \prec v \prec z$. Since $v \notin L(t)$, y and v are comparable. Thus it either holds that $v \prec y$, in which case v distinguishes x, y, or $y \prec v$, and v distinguishes y, z. By contraposition, $\sim$ is transitive. □

Lemma 6.25. *Let x, y be grandchildren of t such that $x \not\sim y$. Then $x \prec y$ if and only if $x \prec_t y$.*

Proof. Between cousins with grandparent t, the orderings $\prec$ and $\prec_t$ coincide by definition of the realisation. Thus we only need to consider the case where x, y are siblings distinguished by some v, say $x \prec v \prec y$. Let v' be the cousin of x, y which is an ancestor of v. Being cousins, v' is comparable with both x and y under $\prec$. It follows that $x \prec v' \prec y$. Using once again that $\prec$ and $\prec_t$ coincide for cousins, this implies $x \prec_t v' \prec_t y$, hence $\prec$ and $\prec_t$ coincide for the pair x, y as desired. □

Indistinguishability is compatible with $\prec$ as follows.

Lemma 6.26. *Let A be an equivalence class of $\sim$. Then among $L(T)$, the subset $L(A) := \bigcup_{x \in A} L(x)$ is an interval of $\prec$.*

Proof. Let A be an equivalence class of $\sim$. All elements of A are siblings. Consider t their parent. Let $x, z \in L(A)$ and $y \in L(T) \setminus L(A)$ be leaves, and suppose for a contradiction that $x \prec y \prec z$. Let x', z' be the ancestors in A of x, z respectively. We have two cases to consider.

1. If y is not a descendant of t, then x' and z' are comparable by $\prec$ with y, and it must be that $x' \prec y \prec z'$. Thus x', z' are distinguished by y, a contradiction.

2. Otherwise, y is a descendant of some child y' of t, which is distinguishable from x' and from z'. Thus there exist $v_1, v_2 \notin L(t)$ such that $x' \prec v_1 \prec y'$ and $y' \prec v_2 \prec z'$. But then we also have $x' \prec v_1 \prec z'$, a contradiction. □

6.4.3 Factoring delayed substitutions. We will now prove that any delayed substitution can be decomposed into products of substitutions.

Lemma 6.27. *Let $\mathcal{C}$ be a class of permutations closed under substitution, patterns, and inverse. Then any permutation obtained by delayed substitution from $\mathcal{C}$ is in $\mathcal{C}^3$.*

Proof. Let $(X, <, \prec)$ be the realisation of a delayed tree $(T, <, \{\prec_t\}_{t \in T})$ with leaves $L(T) = X$, labelled in $\mathcal{C}$. The linear order $<$ is compatible with T, while $\prec$ is the realisation of the delayed substitution.

Let us define an intermediate linear order $<'$ on X. For each internal node t of T and for each child x of t, choose an arbitrary descendant $f_t(x) \in X$ of x. Then, on the children of t, define $x <'_t y$ if and only if $f_t(x) \prec f_t(y)$. These local orders extend to a linear order $<'$ on X, which by construction is compatible with T.

Claim 6.28. The permutation $(<, <'_t)$ on the children of t is in $\mathcal{C}$.

Proof. Let A be the set of children of t. For $x \in A$, call $g(x)$ the sole child of x which is an ancestor of $f_t(x)$. Thus $g(x)$ is a grandchild of t. For $x \neq y \in A$, we have $g(x) < g(y)$ if and only if $x < y$, by compatibility of $<$ with T. Furthermore, the closest ancestor of $f_t(x)$ and $f_t(y)$ is t, hence $f_t(x) \prec f_t(y)$ if and only if $g(x) \prec_t g(y)$ by definition of $\prec$. Thus the permutation $(<, <'_t)$ on A is isomorphic to the permutation $(<, \prec_t)$ on $g(A)$, which is a pattern of the permutation $(<, \prec_t)$ on the grandchildren of t. The latter is in $\mathcal{C}$ by hypothesis. ■

Since $\mathcal{C}$ is closed under substitution, it follows that $(X, <, <')$ is in $\mathcal{C}$.

We now quotient the tree T by indistinguishability. Since $\sim$ can only identify siblings, the quotient $T' = T/\sim$ naturally has a tree structure. Furthermore, since the leaves of T are required to be single children, no two leaves are identified, hence the set of leaves of T' is exactly X.

Claim 6.29. The tree T' is compatible with both $<'$ and $\prec$.

Proof. Lemma 6.26 precisely proves that T' is compatible with $\prec$: each node of T', which is an equivalence class of $\sim$, corresponds to an interval of $(X, \prec)$.

Since $<'$ is compatible with T, to show that it is also compatible with the quotient $T' = T/\sim$, it is sufficient to consider only the children of a given $t \in T$. That is, it is enough to prove that among the children of t, any equivalence class A for indistinguishability is an interval for $<'$. Suppose for a contradiction that there are children $x <' y <' z$ of t with $x, z \in A$ and $y \notin A$. Since $x \not\sim y$, it must be that $x \prec y$ or $y \prec x$, but the latter is impossible as it would imply $y <' x$. Thus $x \prec y$, and similarly $y \prec z$, contradicting Lemma 6.26. ■

Thus the permutation $(X, <', \prec)$ is obtained by substitution along the tree T'. It only remains to show that this structured tree is labelled with permutations in $\mathcal{C}^2$.

Fix a node $\bar{s} \in T'$, which in T is an equivalence class of $\sim$. Let $t \in T$ be the parent of the nodes of $\bar{s}$ (which is well defined as only siblings can be equivalent for $\sim$). The children in T' of $\bar{s}$ are equivalence classes of grandchildren of t in T. Denote by $\bar{x}_1, \dots, \bar{x}_k$ these children, where the representatives $R = \{x_1, \dots, x_k\}$ are grandchildren of t. Recall that $L(x_i) \subseteq X$ is the set of leaves descendant of x_i. The subsets $L(x_1), \dots, L(x_k)$ are non-interleaved for $<$ and $<'$ because these linear orders are compatible with T. Furthermore, they are non-interleaved for $\prec$ because the x_i are pairwise distinguished. Thus R is equipped with the three linear orders $<, <', \prec$. In the structured tree T' which realises $(X, <', \prec)$, $\bar{s}$ is labelled with $(R, <', \prec)$, hence we only need the following to conclude.

Claim 6.30. The permutation $(R, <', \prec)$ is in $\mathcal{C}^2$.

Proof. We decompose the permutation $(R, <', \prec)$ into $(R, <', <)$ and $(R, <, \prec)$. This may at first seem counterproductive, as we go back to the initial permutation $(<, \prec)$, but the point is that we now only consider a small subset R, rather than all X.

We already know that $(X, <, <')$ is in $\mathcal{C}$, hence so is $(R, <', <)$, which is a pattern of its inverse. Furthermore, the x_i are pairwise distinguished grandchildren of t, hence the linear orders $\prec$ and $\prec_t$ coincide on R by Lemma 6.25. Thus $(R, <, \prec)$ is a pattern of $(<, \prec_t)$, which is in $\mathcal{C}$. ■

As $(X, <, \prec)$ factorises into $(X, <, <') \in \mathcal{C}$ and $(X, <', \prec) \in \mathcal{C}^2$, it is in $\mathcal{C}^3$. □

6.4.4 Constructing delayed structured trees. Finally, we show how to express any permutation as a delayed substitution.

Lemma 6.31. *For any biorder $\sigma = (X, <, \prec)$, there is a delayed structured tree $(T, <, \{\prec_t\}_{t \in T})$ whose realisation is σ satisfying the following:*

1. *For any node $t \in T$, the linear ordering $\prec_t$ of its grandchildren is obtained by a choice of representatives. That is, if A is the set of grandchildren of t, then there is a mapping to leaves $f : A \to X$ with $f(a)$ descendant of a such that for $a, b \in A$, $a \prec_t b$ if and only if $f(a) \prec f(b)$.*

2. *If x, y are consecutive siblings along $<$, then x and y are distinguishable for $\prec$, except possibly if x and y do not have any other sibling.*

In the previous statement, Condition 1 is a technical requirement ensuring that the permutations labelling T are patterns of σ, so that hypotheses on σ can also be applied to the former. Condition 2 will be crucial in the proof of Theorem 6.21. Informally, it ensures that the subpermutations $\{(<, \prec_t)\}_{t \in T}$ labelling T are strictly simpler than the global permutation $(X, <, \prec)$.

Proof. We construct the tree T inductively starting from the root, choosing for each internal node t the interval $L(t) \subseteq X$ of $<$ which at the end of the construction will be the set of leaves descendant of t. We maintain the condition

that whenever t' is a child of t and $x \in X \setminus L(t)$, then x does not *split* $L(t')$, i.e. either $x \prec L(t')$ or $x \succ L(t')$.

Initially, there is only the root r with $L(r) = X$. If t and $L(t)$ have already been constructed, then we create the children of t by the following rules.

- If $L(t) = \{x\}$ is a singleton, we add a leaf x as the sole child of t, and the construction stops there for this branch.

- If $L(t)$ has size at least 2 and is an interval of $(X, \prec)$ (in addition to being one for $<$), then we add two children u, v to t and split $L(t)$ arbitrarily into two intervals $L(u), L(v)$ for $<$. Remark that this case occurs at least for the root r.

  If $x \notin L(t)$, then x does not split $L(t)$, hence a fortiori it splits neither $L(u)$ nor $L(v)$, as required. On the other hand, this means that u and v will be indistinguishable in T, which is why condition 2 is waived for nodes with only two children.

- Otherwise, enumerate $L(t)$ as $x_1 < \cdots < x_k$. Say that a subset $A \subseteq L(t)$ is a *local module* of $L(t)$ if no $y \notin L(t)$ splits A (for $\prec$). We greedily partition $L(t)$ into local modules $A_1, \ldots, A_l$: A_1 is $\{x_1, \ldots, x_{i_1}\}$ with i_1 chosen maximal such that A_1 is a local module, then A_2 is $\{x_{i_1+1}, \ldots, x_{i_2}\}$ with again i_2 maximal, etc.

  By construction, no element $y \notin L(t)$ can split any A_i. Further, A_i, A_{i+1} are distinguished (in fact, the last element of A_i is distinguished from the first element of A_{i+1}), as otherwise A_i could have been extended. Also, the number l of local modules in this partition is at least 2, as otherwise $L(t)$ would be an interval of $(X, \prec)$ and we would fall in the previous case.

With this tree constructed, consider a node $t \in T$ and u, v two grandchildren of t which are not siblings. Then the condition maintained during the construction ensures that no $x \in L(u)$ splits $L(v)$, and symmetrically no $y \in L(v)$ splits $L(u)$. It follows that either $L(u) \prec L(v)$ or $L(v) \prec L(u)$. In short, the order $\prec$ is well defined between cousins in T.

Now, define the linear order $\prec_t$ on the grandchildren of t through an arbitrary choice of representatives, as required in condition 1 of the lemma. Then for cousins u, v, we have $u \prec_t v$ if and only if $u \prec v$. In turn, this implies that the realisation of $(T, <, \{\prec_t\}_{t \in T})$ is the permutation $(X, <, \prec)$ from which we started. Condition 1 is satisfied by definition of $\prec_t$, and condition 2 was checked during the construction of T. □

6.5 Partitions and mixity

In this section, we introduce tools to further decompose a single level of a delayed structured tree. Once again, the idea is based on results of Bourneuf and Thomassé [23, Section 4], but working with permutations gives stronger conclusions and simpler proofs.

6.5.1 Preliminaries.

Colouring and degeneracy Recall that a k-colouring of a graph G is a map $c : V(G) \to [k]$ which assigns distinct colours to adjacent vertices. Equivalently, it is a partition into k colour classes $c^{-1}(1), \dots, c^{-1}(k)$, each of which is an independent set.

A graph G is *k-degenerate* if G and all its subgraphs have a vertex of degree at most k. A well-known characterisation is that G is k-degenerate if and only if there is an acyclic orientation of the edges of G such that vertices have out-degree at most k.

Lemma 6.32 (Folklore). *Any k-degenerate graph G can be $(k+1)$-coloured.*

A closely related parameter is the *maximum edge density* of G, that is the maximum over all subgraphs $H \subset G$ of $\frac{|E(H)|}{|V(H)|}$. It is easy to see that k-degenerate graphs have maximum edge density at most k, and conversely graphs with maximum edge density k are $2k$-degenerate. If G admits an orientation (possibly with cycles) in which vertices have out-degree at most k, then its maximum edge density is at most k.

Circle graphs Let $\mathcal{I}$ be a family of intervals of some linear order $<$. Two intervals A, B *overlap* if they intersect, but neither contains the other. The *overlap graph* of $\mathcal{I}$ is the graph whose vertex set is $\mathcal{I}$, and where intervals are adjacent exactly when they overlap. Overlap graphs are also known as *circle graphs*, because they can be described by the intersections of a set of chords of a circle.

We will show that overlap graphs without large cliques or bicliques can be coloured with few colours: they are in fact degenerate. The complete graph on k vertices is denoted by K_t, and the bipartite complete graph with t vertices on each side by $K_{t,t}$.

Lemma 6.33. *Any circle graph containing neither K_t nor $K_{t,t}$ as subgraph has maximum edge density at most $2(t-1)^2$.*

Proof. Let $\mathcal{I}$ be a family of intervals whose overlap graph has no K_t or $K_{t,t}$ subgraph. We define an orientation of the edges with out-degree at most $2(t-1)^2$.

Consider two overlapping intervals $A, B \in \mathcal{I}$. Relative to the edge AB, say that $C \in \mathcal{I}$ is a *private ancestor of A* if $A \subseteq C$ and $B \not\subseteq C$, and symmetrically with B. By convention, A and B are also private ancestors of themselves.

Claim 6.34. Every private ancestor of A overlaps every private ancestor of B.

Proof. Let C, D be private ancestors of A, B respectively, with possibly $C = A$ or $D = B$. Pick $x \in A \cap B$ (which is non-empty as they overlap), and $y \in A \setminus D$ (non-empty since D is private to B), and symmetrically $z \in B \setminus C$. Then, because $A \subseteq C$ and $B \subseteq D$, we also have $x \in C \cap D$, $y \in C \setminus D$, and $z \in D \setminus C$, hence C and D overlap. ■

Thus, there is a biclique between A and its private ancestors on the one hand, and B and its private ancestors on the other. Since the overlap graph does not contain $K_{t,t}$ as subgraph, it follows that either A or B has less than t private ancestors. We orient the edge AB towards A if it has less than t private ancestors, and towards B otherwise.

Let us bound the out-degree of this orientation. Fix $I \in \mathcal{I}$ an interval with endpoints $x < y$, and consider $N^+(I)$ the set of $A \in \mathcal{I}$ with an edge oriented from I to A. Any $A \in N^+(I)$ contains either x or y, but not both. Let X be the subset of intervals in $N^+(I)$ containing x, and consider the inclusion poset $(X, \subseteq)$.

Claim 6.35. The poset $(X, \subseteq)$ does not contain a chain of length t.

Proof. If $A_1 \subsetneq \cdots \subsetneq A_s$ are in X, then A_1 overlaps I, and y witnesses that each A_i is a private ancestor of A_1 w.r.t. the edge $A_1 I$. Since this edge is oriented from I to A_1, we know that A_1 has less than t private ancestors, hence $s < t$. ■

Claim 6.36. The poset $(X, \subseteq)$ does not contain an antichain of size t.

Proof. If $A, B \in X$ are incomparable for inclusion, then they overlap because both contain x. Thus an antichain in the poset $(X, \subseteq)$ is a clique in the overlap graph. ■

By Dilworth's theorem, it follows from the two claims that $|X| \leqslant (t-1)^2$. The same reasoning applies to intervals in $N^+(I)$ containing y, and we obtain $|N^+(I)| \leqslant 2(t-1)^2$. □

While Lemma 6.33 proves degeneracy, we only use it to construct colourings. For this purpose, we could instead use a much more general result of Davies and McCarty: circle graphs without K_t as subgraph (but possibly with $K_{t,t}$) can be coloured with $O(t^2)$ colours [36], later improved to $O(t \log t)$ in [35]. Their proof is however far more complex.

Splitting and mixing Let $(X, \prec)$ be a linear order, and $\mathcal{P}$ a partition of X. For any subsets $X_1, X_2 \in \mathcal{P}$, we distinguish three cases:

1. X_1, X_2 do not interleave, i.e. $X_1 \prec X_2$ or $X_2 \prec X_1$.
2. Restricted to $X_1 \cup X_2$, X_1 is an interval which is said to *split* X_2 in two, written $X_1 \sqsubset X_2$, or vice versa.
3. Neither X_1 nor X_2 is an interval in $X_1 \cup X_2$, and we say that X_1, X_2 are *mixed*.

Assume now that no pair of subsets in $\mathcal{P}$ is mixed—we then say that $\mathcal{P}$ is *non-mixed*. For any $X_1 \in \mathcal{P}$, let $\overline{X}_1$ denote the interval closure of X_1, i.e. the interval of $(X, \prec)$ between the minimum and maximum of X_1. Then we have $X_1 \prec X_2$ if and only if $\overline{X}_1 \prec \overline{X}_2$, and $X_1 \sqsubset X_2$ if and only if $\overline{X}_1 \subset \overline{X}_2$. In particular, either $\overline{X}_1, \overline{X}_2$ are disjoint, or one contains the other. Thus $(X, \mathcal{P})$ induces a laminar family, meaning that subsets in $\mathcal{P}$ are the nodes of a rooted forest F, where X_1 is a descendant of X_2 if and only if $X_1 \sqsubset X_2$. Furthermore, the ordering $\prec$ is defined for any pair of subsets $X_1, X_2 \in \mathcal{P}$ which are not in an ancestor–descendant relationship. Thus $\prec$ gives to F the structure of an ordered forest: each connected component of F is an ordered tree, and $\prec$ orders the components.

6.5.2 Non-mixed partitions. We now consider a biorder $(X, <, \prec)$, and a partition $\mathcal{P}$ of X into intervals for $<$. In later proofs, such a structure will arise from delayed substitutions as follows: Given a node t in a delayed structured tree, X is the set of grandchildren of t, the partition $\mathcal{P}$ is the one given by the children of t, $<$ is the linear order compatible with the tree, while $\prec$ is the linear order on the grandchildren of t given as part of the delayed structured tree. Our goal is to decompose this biorder into simpler permutations.

Lemma 6.37. *Let $\mathcal{C}$ be a substitution-closed class of permutations. Consider a biorder $(X, <, \prec)$ and a partition $\mathcal{P}$ of X into intervals of $<$. Suppose that $\mathcal{P}$ is non-mixed w.r.t. $\prec$, and that*

1. *for each part $P \in \mathcal{P}$, the induced permutation $(P, <, \prec)$ is in $\mathcal{C}$, and*
2. *there exists a* transversal *R of $\mathcal{P}$ (i.e. a choice of a single element in each part of $\mathcal{P}$) such that the permutation $(R, <, \prec)$ is in $\mathcal{C}$.*

Then $(X, <, \prec)$ is in $\mathcal{S}^2 \circ \mathcal{C}$, where $\mathcal{S}$ denotes the class of separable permutations.

Proof. Let us define an intermediate linear order $<'$ on X as follows:

- each part $P \in \mathcal{P}$ is an interval of $(X, <')$,
- inside each part P, the linear order $<'$ coincides with $\prec$, and
- between parts $P_1, P_2 \in \mathcal{P}$, the order is given by the representatives in R, i.e. if x_i is the sole element in $P_i \cap R$, then $P_1 <' P_2$ if and only if $x_1 \prec x_2$.

Claim 6.38. The permutation $(X, <, <')$ is in $\mathcal{C}$.

Proof. The permutation inside a given part $P \in \mathcal{P}$ is $(P, <, \prec)$, which is known to be in $\mathcal{C}$. Furthermore, the permutation on the set $\mathcal{P}$ is isomorphic to $(R, <, \prec)$, which is also in $\mathcal{C}$. Therefore $(X, <, <')$ can be expressed as a substitution from permutations in $\mathcal{C}$, with a structured tree of depth 2. ■

We now focus on $(X, <', \prec)$, and will prove that this permutation is obtained by substitution from 2-shuffles. Such permutations are in $\mathcal{S}^2$ by Lemma 6.22 since $\mathcal{S}$ is closed under substitution.

Recall the structure on $(X, \prec)$ and $\mathcal{P}$ described in section 6.5.1: Since the partition $\mathcal{P}$ is non-mixed w.r.t. $\prec$, the *splitting* partial order $\sqsubset$ gives $\mathcal{P}$ the structure of a rooted forest F, which furthermore is compatible with the linear order $\prec$. We construct a tree T from F by

- adding a new root, whose children are the roots of the connected components of F, and
- for each node $P \in \mathcal{P}$, adding each element $x \in P$ as a new child of P.

The leaves of T are exactly the elements of X, and we will describe $(X, <', \prec)$ as a substitution along T. See Figure 6.5 for an illustration.

Claim 6.39. The linear orders $<'$ and $\prec$ are compatible with T.

Proof. Consider a node t of T, other than the leaves and the root. Thus t can be identified with a part $P \in \mathcal{P}$. Then it is simple to check that $L(P)$, the set of leaves descending from P in T, is exactly $\bigcup_{P' \sqsubseteq P} P'$, i.e. the union of parts which are descendants of P in F. We claim that this is an interval for both $<'$ and $\prec$.

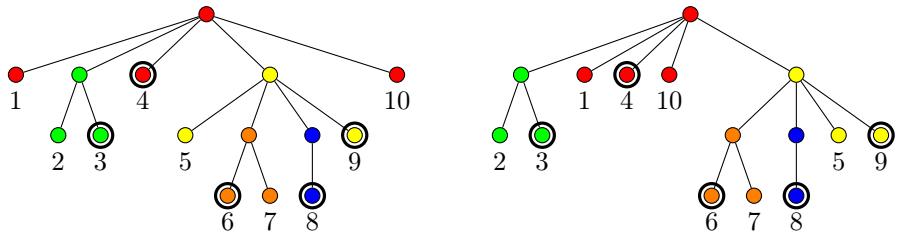


Figure 6.5: The tree T, ordered by $\prec$ (left) and $<'$ (right) respectively. Leaves are elements of X, and their numbering is according to $\prec$. Internal nodes correspond to parts in $\mathcal{P}$, with each colour representing a part of $\mathcal{P}$. The transversal R is indicated by circled leaves, and decides how $<'$ orders the parts.

Suppose otherwise towards a contradiction. Then there are $u, w \in L(P)$ and $v \notin L(P)$ such that either $u \prec v \prec w$ or $u <' v <' w$. Let U, V, W be the parts of $\mathcal{P}$ containing u, v, w respectively. We have $U, W \sqsubseteq P$, and $V \not\sqsubseteq P$. Recall that $\overline{P}$ denotes the interval between the minimum and maximum of P for $\prec$. Then we have $U, W \subseteq \overline{P}$, while V is disjoint from $\overline{P}$. Under these conditions and since $\overline{P}$ is an interval of $\prec$, it cannot be that $u \prec v \prec w$.

For $<'$, we also need to consider the representative $v' \in V \cap R$. For the same reasons as above, we have either $v' \prec U \cup W$, or $U \cup W \prec v'$, which implies $V <' U \cup W$ or $U \cup W <' V$ respectively. Either way, $u <' v <' w$ is impossible. ■

To conclude the proof, we only need to show that for any internal node P of T, the permutation $(X, <', \prec)$ restricted to children of P is a 2-shuffle. We partition the children of P in two categories: leaves (which are in X), and internal nodes (which are in $\mathcal{P}$). We claim that $(X, <', \prec)$ restricted to either of these categories is the identity.

1. The leaves which are children of P are exactly the elements of P, and inside P the orderings $<'$ and $\prec$ coincide by construction.

2. The parts of $\mathcal{P}$ which are children of P in T are also the children of P in F. We know that they are totally ordered under $\prec$, i.e. if $A, B \in \mathcal{P}$ are children of P, then either $A \prec B$ or $B \prec A$. By construction of $<'$, if $A \prec B$ (resp. $B \prec A$) then $A <' B$ (resp. $B <' A$). It follows from these two remarks that $<'$ and $\prec$ coincide on the children of P in F. □

6.5.3 Separating mixed parts. The previous lemma shows how to decompose a permutation which is non-mixed w.r.t. a given partition. We will now generalise it to permutations with few pairs of mixed parts. Let $(X, <)$ be a linear order and $\mathcal{P}$ a partition of X. The associated *mixed graph* is the graph with vertex set $\mathcal{P}$ where two parts $A, B \in \mathcal{P}$ are adjacent when they are mixed.

Lemma 6.40. *Let G be the mixed graph of a partition, and assume that it contains neither K_t nor $K_{t,t}$ as subgraph. Then G is $(4t^2 - 1)$-degenerate.*

Proof. Let $G = (\mathcal{P}, E)$ be the mixed graph of a partition $\mathcal{P}$ of a linear order $(X, <)$. We partition the edges of G into $E = E_1 \uplus E_2$ as follows:

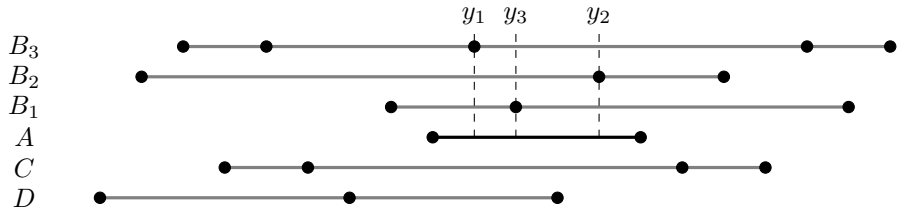


Figure 6.6: Illustration of the proof of Lemma 6.40. Each row represents a part of $P \in \mathcal{P}$: the dots are elements of P, and line is the interval $\overline{P}$. The left-to-right order is $\prec$. The B_is are out-neighbours of A in the graph $(\mathcal{P}, E_1)$, and $y_i \in B_i \cap \overline{A}$ witnesses that A and B_i are mixed. For any $i \neq j$, y_i, y_j together with the endpoints of $\overline{B_i}, \overline{B_j}$ ensure that B_i and B_j are mixed.
The parts C, D are *not* out-neighbours of A in $(\mathcal{P}, E_1)$: C satisfies $\overline{A} \subset \overline{C}$ but is not mixed with A, and $\overline{A}$ and $\overline{D}$ overlap, hence the edge AD is in E_2, not E_1.

1. An edge $AB \in E$ is in E_1 if the interval closures satisfy $\overline{A} \subset \overline{B}$ or $\overline{B} \subset \overline{A}$.

2. The remaining edges are in E_2. Thus, a pair $(A, B) \in \mathcal{P}$ is an edge in E_2 if and only if $\overline{A}$ and $\overline{B}$ overlap.

The graph $(\mathcal{P}, E_2)$ is precisely the overlap graph of the interval closures of parts of $\mathcal{P}$. Thus by Lemma 6.33, its maximum edge density is at most $2(t-1)^2$. We will show that $(\mathcal{P}, E_1)$ is $(t-1)$-degenerate, hence has maximum edge density at most $t-1$. Then, their union $(\mathcal{P}, E)$ has maximum edge density at most $t - 1 + 2(t-1)^2 \leqslant 2t^2 - 1$, hence it is $(4t^2-1)$-degenerate as desired.

To this end, we define an acyclic orientation of $(\mathcal{P}, E_1)$ for which the out-degree is at most $t-1$. If AB is an edge in E_1 and $\overline{A} \subset \overline{B}$, we orient it from A to B. We claim that for any part $A \in \mathcal{P}$, the out-degree of A in $(\mathcal{P}, E_1)$ is at most $t-1$. Indeed, let $B_1, \dots, B_k$ be the out-neighbours of A, i.e. each B_i is mixed with A and $\overline{A} \subset \overline{B}_i$. We will show that $A, B_1, \dots, B_k$ is a clique in $(\mathcal{P}, E)$ which implies the claim as this graph has no clique of size t. To this end, we need to show that any two $B_i \neq B_j$ are mixed. Without loss of generality, assume that $\overline{B}_i \not\subset \overline{B}_j$, hence there is some $x \in B_i$ outside of $\overline{B}_j$, say $x < B_j$. On the other hand, since B_i is mixed with A, there must be $y_i \in B_i \cap \overline{A}$, and thus $y_i \in B_i \cap \overline{B}_j$. Thus we have two points in B_i, one inside the interval closure of B_j and one outside. This implies that B_i and B_j are mixed. See Figure 6.6 for an illustration. □

Combining this with Lemma 6.37, we obtain the following.

Lemma 6.41. *Let $\mathcal{C}$ be a class of permutations closed under substitution, symmetry, and taking patterns. Let $(X, <, \prec)$ be a biorder, and $\mathcal{P}$ a partition of X into intervals of $<$, satisfying the following:*

1. *The mixed graph of $\mathcal{P}$ for the linear order $(X, \prec)$ does not contain K_t or $K_{t,t}$ subgraphs.*

2. *For each part $P \in \mathcal{P}$, the permutation $(P, <, \prec)$ is in $\mathcal{C}$.*

3. *There exists a transversal R of $\mathcal{P}$ (i.e. a choice of a single element in each part of $\mathcal{P}$) such that the permutation $(R, <, \prec)$ is in $\mathcal{C}$.*

Then the permutation $(X, <, \prec)$ *is in* $\mathcal{S}^{k+2} \circ \mathcal{C} \circ \mathcal{S}^k$*, with* $k = \lceil 2 \log t \rceil + 2$.

Proof. By Lemma 6.40, the mixed graph of $\mathcal{P}$ is $(4t^2 - 1)$-degenerate, hence $4t^2$-colourable by Lemma 6.32. Fix a proper $4t^2$-colouring, and for any colour c call $\mathcal{P}_c$ be the set of parts of colour c and $X_c = \bigcup_{P \in \mathcal{P}_c} P$ the points contained therein. Since the colouring is proper, no two parts of $\mathcal{P}_c$ are mixed. Thus the restricted biorder $(X_c, <, \prec)$ with the partition $\mathcal{P}_c$ satisfies the conditions of Lemma 6.37, and the permutation $(X_c, <, \prec)$ is in $\mathcal{S}^2 \circ \mathcal{C}$. Hence, $(X, <, \prec)$ is a $4t^2$-shuffle of permutations in $\mathcal{S}^2 \circ \mathcal{C}$. We conclude by applying Lemma 6.22 with $k = \lceil \log(4t^2) \rceil = \lceil 2 \log t \rceil + 2$. □

6.6 Reducing the size of mixed divisions

We now reach the core of the proof of Theorem 6.21: using the tools developed in sections 6.4 and 6.5, we decompose a given permutation σ into a number of strictly simpler permutations, in the sense of some parameter equivalent to twin-width.

This parameter is the size of *mixed divisions*, a variant of the rank divisions of Chapter 3. The idea of mixed divisions in fact predates rank divisions: the former were used in [19] in the original form of Theorem 3.9, while the latter were only introduced in [17]. The idea of decomposing a structure into several ones which smaller mixed divisions is due to Pilipczuk and Sokołowski [79].

A 0–1 matrix M is called *horizontal* if each row r of M is constant, i.e. all entries in r are equal. Equivalently, M is horizontal if and only if all its columns are equal. Symmetrically, M is called *vertical* if each of its columns is constant, i.e. all its rows are equal. Note that a matrix which is both horizontal and vertical is constant. Finally, M is *mixed* if it is neither horizontal nor vertical. Remark that if X_1, X_2 are disjoint subsets of a linear order $(X, <)$, then X_1, X_2 are mixed in the sense of section 6.5 if and only if the adjacency matrix of $<$ restricted to rows in X_1 and columns in X_2 is mixed.

A *k-mixed division* in a matrix M is a k-division in which every cell is mixed. For $k \geqslant 2$, a rank-k division is also a k-mixed division. Using the very same arguments as Lemma 3.21, one can show the following.

Lemma 6.42 [19, Theorem 5.4]. *Consider a binary structure* $S = (V, R)$ *and an ordering* $<$ *of* V *compatible with some contraction sequence of width* k*. Then* $A(R, <)$ *has no* $(2k + 4)$*-mixed division.*

Given a permutation $\sigma = (X, <, \prec)$, consider the adjacency matrix $A(\prec, <)$: this is the adjacency matrix of the relation $\prec$, with rows and columns ordered by $<$. Assume that $A(\prec, <)$ has no k-mixed division, or in short is *k-mixed free*. We will first show how to decompose k-mixed free permutations by induction on k. The base case is the following simple remark.

Lemma 6.43. *Any 2-mixed free permutation is separable.*

Proof. A permutation which is not separable contains either 3142 or 2413 as pattern, and both have 2-mixed divisions in their adjacency matrices. See Figure 6.7 for the adjacency matrix of 3142; that of 2413 is its transpose. □

Let $\mathcal{A}_k$ denote the class of k-mixed free permutations, and recall that $\mathcal{S}$ is the class of separable permutations.

Figure 6.7: Adjacency matrix of the permutation $\sigma = 3142$, which contains a 2-mixed division represented by dashed lines. The matrix has a 1 at the intersection of the ith row and jth column if and only if $\sigma(i) < \sigma(j)$.

Theorem 6.44 [12]. *If $\mathcal{A}_{k-1} \subseteq \mathcal{S}^r$ for some $r \in \mathbb{N}$, then $\mathcal{A}_k \subseteq \mathcal{S}^s$ for*

$$s = 3r + 12\lceil \log k \rceil + 28.$$

Proof. Assume that $\mathcal{A}_{k-1} \subseteq \mathcal{S}^r$, and consider $\sigma = (X, <, \prec) \in \mathcal{A}_k$ a k-mixed free permutation. Denote by **first** and **last** the minimum and maximum of X w.r.t. $<$. For the linear order $\prec$, **first** and **last** split X into three intervals as

$$X_1 \prec \mathbf{first} \prec X_2 \prec \mathbf{last} \prec X_3,$$

up to swapping **first** and **last**. We consider each X_i independently, before recombining X_1, X_2, X_3, and $\{\mathbf{first}, \mathbf{last}\}$. Let $X' = X_i$ be one of these intervals, and $\sigma' = (X', <, \prec)$ the induced permutation.

Let $(T, <, \{\prec_t\}_{t \in T})$ be a delayed structured tree for σ' given by Lemma 6.31. Consider any internal node $t \in T$, let A be its set of children, and consider a transversal $S \subseteq X'$ of A, i.e. all elements of S are descendants of t, and each child $v \in A$ has exactly one descendant in S.

Claim 6.45. The transversal S admits a partition into two parts $S = S_L \uplus S_R$ such that $(<, \prec)$ restricted to each of S_L, S_R is $(k-1)$-mixed free.

Proof. Enumerate the children of t as $A = \{v_1 < \cdots < v_\ell\}$, and call x_i the descendant of v_i in S. We can assume $\ell > 2$ as the claim is otherwise trivial. Then condition 2 of Lemma 6.31 gives that for all $i \in \{1, \ldots, \ell - 1\}$, v_i and v_{i+1} are distinguishable. Thus, we can choose some $y_i \in X' \setminus L(t)$ such that $v_i \prec y_i \prec v_{i+1}$, or $v_{i+1} \prec y_i \prec v_i$.

If $y_i < L(t)$ (resp. $y_i > L(t)$) we say that v_i is *split to the left* (resp. *to the right*). Thus every v_i except v_ℓ is split either to the left or to the right. We partition S accordingly, so that S_L (resp. S_R) contains x_i only if v_i is split to the left (resp. to the right). We place x_ℓ in either S_L or S_R arbitrarily. We will prove that $(S_L, <, \prec)$ is $(k-1)$-mixed free, the case of S_R being symmetrical.

Suppose for a contradiction that $(S_L, <, \prec)$ has a $(k-1)$-mixed division. It is given by two partitions of $(S_L, <)$ into $(k-1)$ intervals, say

$$\mathcal{R} = \{R_1 < \cdots < R_{k-1}\} \qquad \text{and} \qquad \mathcal{C} = \{C_1 < \cdots < C_{k-1}\}$$

such that for every $i, j \in [k-1]$, R_i and C_j are mixed in the linear order $\prec$. Note in particular that all R_i and C_j have size at least 2. Starting from $\mathcal{R}$ and $\mathcal{C}$, we build two new partitions $\mathcal{R}' = \{R'_0 < \cdots < R'_{k-1}\}$ and $\mathcal{C}' = \{C'_0 < \cdots < C'_{k-1}\}$, which will form a k-mixed division in σ.

- The parts R'_0 and C'_0 are equal, they consist of **first** and all splitting elements y_i satisfying $y_i < L(t)$, $i \in [\ell - 1]$.

- For each $i \geqslant 1$, consider x_j the minimum of R_i w.r.t. $<$. Since R_i has size at least 2, we have $j < \ell$, and either $x_{j+1} \in R_i$, or x_{j+1} is not in S_L. Either way, we define $R'_i = R_i \cup \{x_{j+1}\}$. Note here that x_j is split to the left, hence $y_j \in C'_0$.

- The same operation is applied to $\mathcal{C}'$.

After this modification, all elements of R'_0 are smaller than $L(t)$ for $<$, hence $\mathcal{R}'$ is still a partition of a subset of X into intervals of $<$, and similarly with $\mathcal{C}'$. The parts which were originally in $\mathcal{R}, \mathcal{C}$ have only increased, hence for $i, j \geqslant 1$, R'_i and C'_j are mixed. Further, R'_0 and C'_0 are equal and are not singletons. Since we are considering the adjacency matrix of a linear order, this implies that the corresponding cell $R'_0 \times C'_0$ is mixed. We claim that R'_0 is mixed with C'_i for any $i > 0$, and symmetrically for C'_0, R'_i, which implies that the new $\mathcal{R}', \mathcal{C}'$ form a k-mixed division in (a submatrix of) M_σ, a contradiction.

Indeed, let x_j be the smallest element of R_i, so that $y_j \in C'_0$ distinguishes x_j and x_{j+1}, i.e. either $x_j \prec y_j \prec x_{j+1}$ or $x_{j+1} \prec y_j \prec x_j$. By construction of X', **first** is not interleaved with x_j, x_{j+1} (or any elements of X'), i.e. either **first** $\prec x_j, x_{j+1}$ or $x_j, x_{j+1} \prec$ **first**. The above implies that $\{\textbf{first}, y_j\}$ and $\{x_j, x_{j+1}\}$ are mixed. The former is contained in C'_0, while the latter is contained in R'_i, proving the claim.

The same reasoning also holds for S_R, using **last** instead of **first** and adding parts R'_k and C'_k instead of R'_0 and C'_0. ■

It follows from Claim 6.45 that for any transversal S of the children of a node t, the permutation $(S, <, \prec)$ is a 2-shuffle of permutations in $\mathcal{A}_{k-1}$, hence is in $\mathcal{S}^{r+2}$ by Lemma 6.22.

Fix now a node $t \in T$, and call A the set of grandchildren of t. Consider the permutation $(A, <, \prec_t)$, and the partition $\mathcal{P}$ of A induced by the children of t. Recall that $\mathcal{S}^{r+2}$ is closed under substitution, inverse and taking patterns. Let us check that the three conditions of Lemma 6.41 are satisfied.

1. If the mixed graph of $\mathcal{P}$ for the linear order $(A, \prec_t)$ contains either K_k or $K_{k,k}$ as a subgraph, then the corresponding parts form a k-mixed division in the adjacency matrix of $(A, <, \prec_t)$. We know that the linear order $\prec_t$ is defined through a choice of representatives, meaning that $(A, <, \prec_t)$ is a pattern of $(X, <, \prec)$. Then $(X, <, \prec)$ would also have a k-mixed division, a contradiction.

2. Fix a part $P \in \mathcal{P}$. The elements of P are the children of some child t' of t, thus the restricted permutation $(P, <, \prec_t)$ is isomorphic to $(S, <, \prec)$ for some transversal S of P. By Claim 6.45 applied on t', this permutation is in $\mathcal{S}^{r+2}$.

3. Similarly, applying Claim 6.45 this time to t shows that for any transversal S of $\mathcal{P}$, the permutation $(S, <, \prec_t)$ is in $\mathcal{S}^{r+2}$.

It follows from Lemma 6.41 that the permutation $(A, <, \prec_t)$ is in $\mathcal{S}^{r'}$ with

$$\begin{aligned} r' &= (r+2) + 2\left(\lceil 2\log k\rceil + 2\right) + 2 \\ &\leqslant r + 4\lceil \log k\rceil + 8. \end{aligned}$$

Next, we apply Lemma 6.27 to the delayed structured tree T and conclude that $(X', <, \prec)$ is in $\mathcal{S}^{3r'}$. Recall that X' was one of three intervals of $(X, \prec)$ defined by **first** and **last**. Combining the permutations on these three intervals and on $\{\mathbf{first}, \mathbf{last}\}$, we finally find that $\sigma = (X, <, \prec)$ is a 4-shuffle of permutations in $\mathcal{S}^{3r'}$. By Lemma 6.22, this implies that σ is in $\mathcal{S}^s$ for

$$\begin{aligned} s &= 3r' + 4 \\ &\leqslant 3r + 12\lceil \log k\rceil + 28. \end{aligned}$$

□

Recursively applying Theorem 6.44 gives the following bound on the length of factorisations of k-mixed free permutations.

Corollary 6.46 [12]. *Any k-mixed free permutation factorises into a product of at most $4 \cdot 3^k$ separable permutations.*

Proof. Lemma 6.43 and Theorem 6.44 give that k-mixed free permutations are product of at most $f(k)$ separable permutations for any function f satisfying

$$\begin{aligned} f(2) &\geqslant 1 \\ \text{and } f(k) &\geqslant 3f(k-1) + 12\lceil \log k\rceil + 28. \end{aligned}$$

This is satisfied for

$$f(k) = 4 \cdot 3^k - 6\lceil \log k\rceil - 23.$$

Indeed, we have

$$\begin{aligned} f(2) &= 4 \cdot 3^2 - 6\lceil \log 2\rceil - 23 = 7. \\ \text{and } f(k) &= 4 \cdot 3^k - 6\lceil \log k\rceil - 23 \\ &\geqslant 4 \cdot 3^k - 18(1 + \lceil \log(k-1)\rceil) + 12\lceil \log k\rceil - 23 \\ &= 3\big(4 \cdot 3^{k-1} - 6\lceil \log(k-1)\rceil - 23\big) - 18 + 12\lceil \log k\rceil + 2 \cdot 23 \\ &= 3f(k-1) + 12\lceil \log k\rceil + 28. \end{aligned}$$

□

Thus we are able to decompose permutations which have no large mixed division. It only remains to generalise this to permutations with bounded twin-width. We will use Lemma 6.42 to show that permutations with bounded twin-width can be factorized as two permutations with no large mixed divisions.

It is worth noting that a permutation with bounded twin-width may contain arbitrarily large mixed divisions. This is the case of 2-shuffles: if $<$ is the usual order on $[n]$, and $\prec$ orders all odd integers before the even ones, then the adjacency matrix of $A(\prec, <)$ contains an $(n/2)$-mixed division. Thus the permutation $\sigma = (<, \prec)$ has arbitrary mixed minors; it does however avoid the pattern 321, hence has small twin-width. In this specific example, the inverse permutation $\sigma^{-1} = (\prec, <)$ *is* 4-mixed free, but one could combine σ and σ^{-1} to obtain an example where the inverse also has arbitrary mixed minors.

In general, we need to introduce a third linear order to obtain a k-mixed free structure.

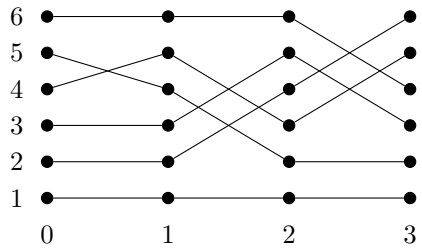


Figure 6.8: A factorisation of 163524 into three separable permutations represented as a path system. The ordering of vertices is left-to-right, then bottom-to-top.

Lemma 6.47. *Any permutation σ with* $\mathrm{tww}(\sigma) = k$ *factorises as*

$$\sigma = \sigma_2 \circ \sigma_1^{-1}$$

where σ_1, σ_2 are $(2k+4)$-mixed free.

Proof. Represent σ as a biorder $(X, <_1, <_2)$. Consider a contraction sequence of width k for σ, and choose $<_3$ to be any linear ordering of X compatible with this contraction sequence. Then by Lemma 6.42, both $A(<_1, <_3)$ and $A(<_2, <_3)$ are $(2k+4)$-mixed free. Defining permutations $\sigma_1 = (X, <_3, <_1)$ and $\sigma_2 = (X, <_3, <_2)$, we obtain the result. □

The main result of this chapter immediately follows from Corollary 6.46 and Lemma 6.47.

Theorem 6.21 [12]. *For any twin-width bound $t \in \mathbb{N}$, there is a length $k \in \mathbb{N}$ such that any permutation σ with* $\mathrm{tww}(\sigma) \leqslant t$ *factorises as a composition $\sigma = \sigma_1 \circ \cdots \circ \sigma_k$ with σ_i separable.*

6.7 Encoding graphs and permutations

In the final section of this chapter, we will show how Theorem 6.21 translates to results on graphs.

Theorem 6.21 shows that *any* permutation σ can be constructed by starting from a fixed based class (separable permutations) and using an operation (composition) whose complexity (the number of permutations to compose) is allowed to depend on $\mathrm{tww}(\sigma)$ only. To adapt this to graph, we need to replace composition by a graph construction. To this end, we first define an encoding of compositions as ordered graphs.

6.7.1 Path system representations. Let $\sigma_1, \ldots, \sigma_m \in \mathfrak{S}_n$ be permutations. The *path system representation* of the product $\sigma_m \circ \cdots \circ \sigma_1$ consists of the ordered matchings $Mt_{\sigma_1}, \ldots, Mt_{\sigma_m}$ joined together by identifying one side of Mt_{σ_i} with the other of $Mt_{\sigma_{i+1}}$. Precisely, it is the ordered graph $(V, E, <)$ with vertex set $V = [0, m] \times [n]$ ordered lexicographically by $<$, and with an edge between $(i-1, x)$ and $(i, \sigma_i(x))$ for every $i \in [m]$, $x \in [n]$. See Figure 6.8 for an example..

We will prove that if all the permutations σ_i have bounded twin-width, then the path system representation has bounded twin-width, independently of the length m. The proof use permutation matrices; recall from Theorem 6.6 that permutations with small twin-width have no large grid in their matrices.

Lemma 6.48 [14, section 6],[12, Lemma 7.1]. *Let $\sigma_1, \dots, \sigma_m$ be permutations whose matrices M_{σ_i} have no r-grid, and consider the path system representation $(V, E, <)$ of the product $\sigma_m \circ \cdots \circ \sigma_1$. Then the adjacency matrix $A(E, <)$ has no $3r + 2$ grid.*

Proof. Define the ith column of vertices $X_i = \{(i, j) \mid j \in [n]\}$ in the path system representation for $i \in \{0, \dots, m\}$. Thus σ_i is encoded by the edges between X_{i-1} and X_i. The adjacency matrix $A(E, <)$ consists of a double diagonal of blocks corresponding to the adjacency matrices of X_{i-1} against X_i for each i. The latter is exactly the permutation matrix M_{σ_i} or its transpose, see Figure 6.9.

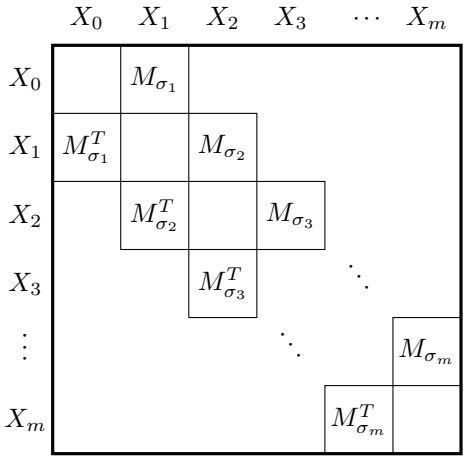


Figure 6.9: Adjacency matrix of the path system representation of $\sigma_m \circ \cdots \circ \sigma_1$.

Consider now an l-grid in this matrix induced by a division $\mathcal{R}, \mathcal{C}$.

Claim 6.49. There exists $i \in [m]$ such that any part of $\mathcal{R}$ intersects one of X_{i-1}, X_i, X_{i+1}.

Proof. Let C_1 be the first part of $\mathcal{C}$, and consider i minimal such that C_1 is contained in $X_0 \cup \cdots \cup X_i$. Then there is no edge between C_1 and X_j for $j > i+1$, hence every $R \in \mathcal{R}$ must intersect $X_0 \cup \cdots \cup X_{i+1}$. Symmetrically, if C_l is the last part of $\mathcal{C}$ and j is maximal such that $C_l \subseteq X_j \cup \cdots \cup X_m$, we find that any $R \in \mathcal{R}$ must intersect $X_{j-1} \cup \cdots \cup X_m$. Thus any $R \in \mathcal{R}$ must intersect $X_{j-1} \cup X_j \cup \cdots \cup X_i \cup X_{i+1}$. But necessarily $i \leqslant j$, hence any part $R \in \mathcal{R}$ must intersect $X_{i-1} \cup X_i \cup X_{i+1}$. ■

Claim 6.50. There exists $j \in [m]$ such that at least $\frac{l-4}{3}$ parts of $\mathcal{R}$ are in X_j.

Proof. By Claim 6.49, any part of $\mathcal{R}$ intersects one of X_{i-1}, X_i, X_{i+1}. Excluding four parts at the borders of the latter, at least $l-4$ parts of $\mathcal{R}$ are contained in one of X_{i-1}, X_i, X_{i+1}. By the pigeonhole principle, at least $\frac{l-4}{3}$ parts are contained in the same of these three subsets. ■

Naturally, Claims 6.49 and 6.50 also hold for columns. Thus we obtain two sets X_i, X_j which contain at least $\frac{l-4}{3}$ parts of $\mathcal{R}$ and $\mathcal{C}$ respectively. This gives a $\left(\frac{l-4}{3}\right)$-grid in one of the blocks of the adjacency matrix, which are r-grid free by hypothesis. It follows that $l \leqslant 3r+1$, i.e. the adjacency matrix is $(3r+2)$-grid free. □

6.7.2 Subdivisions of sparse graphs. Recall that for any graph G, the *r-subdivision* $G^{(r)}$ is obtained by replacing each edge of G by a path of $(r+1)$ edges. We saw in Fact 2.11 that if G is the clique K_n and r is asymptotically smaller than $\log n$, then these graphs have unbounded twin-width. Furthermore, for any constant r, there is a first-order interpretation Φ_r such that $\Phi_r(G^{(r)}) = G$ (cf. section 4.1). It follows by Theorem 4.16 that if a class $\mathcal{C}$ of graphs has unbounded twin-width, then so do the r-subdivisions of graphs in $\mathcal{C}$.

Let us thus exclude these two problematic cases: consider a graph G with twin-width k and no $K_{t,t}$ as subgraph. Then not only does $G^{(r)}$ have bounded twin-width, but we will show that for r an appropriate function of k, t, tww$(G^{(r)}$ is bounded by a *universal* constant c, independent of k and t.

This combination of bounded twin-width and no $K_{t,t}$ subgraph is an interesting hypothesis: it was observed in [14] that it is a very natural notion of *sparse graphs with bounded twin-width*, with several equivalent characterisation. The relevant one here is the absence of grids in the adjacency matrix:

Lemma 6.51 [14, Theorem 2.12]. *Let G be a graph with twin-width k and no $K_{t,t}$ subgraph. Then there is some linear ordering $<$ of the vertices of G for which the adjacency matrix of G has no $t(2k+4)$-grid.*

Proof. Choose $<$ to be an ordering compatible with some contraction sequence of width k. Then by Lemma 6.42, $A(G,<)$ has no $(2k+4)$-mixed division. Now suppose for a contradiction that $A(G,<)$ contains a $t(2k+4)$-grid given by the division $(\mathcal{R},\mathcal{C})$. Regroup the parts of $\mathcal{R}$, resp. $\mathcal{C}$ by groups of t consecutive parts. This yields a $(2k+4)$-division, which by hypothesis contains a non-mixed cell M. Without loss of generality, assume that the cell M is horizontal. Further, M was created from t parts of $\mathcal{R}$ and t parts of $\mathcal{C}$, hence it contains a t-grid. In particular, there are at least t non-zero rows in M. But since M is horizontal, a non-zero row is in fact filled with '1's. These t rows of '1's give $K_{t,t}$ as subgraph. □

Theorem 6.52 [12, Theorem 7.2]. *There is a universal constant c and a function f such that for any graph G with twin-width k and no $K_{t,t}$-subgraph, the subdivision $G^{f(k,t)}$ has twin-width at most c.*

Proof. Let $G = (V, E)$ be $K_{t,t}$-subgraph-free with twin-width k, and, applying Lemma 6.51, consider a linear ordering $<_V$ on V for which the adjacency matrix is r-grid free for $r = t(2k+4)$. Fix an arbitrary orientation of the edges of G, and denote by $\vec{E}$ the set of oriented edges: for each $uv \in E$, exactly one of (u,v) or (v,u) is in $\vec{E}$. If $\vec{e} = (u,v) \in \vec{E}$, we denote by $s(\vec{e}) = u$ its starting point and by $t(\vec{e}) = v$ its endpoint.

We define two lexicographic orders on $\vec{E}$: $<_s$ orders first by starting points (ordered by $<_V$), and then by endpoints, while $<_t$ orders first by endpoints,

then by starting points. We consider the permutation $\sigma = (\vec{E}, <_s, <_t)$, and its matrix M_σ: the columns are $\vec{E}$ ordered by $<_s$, while the rows are $\vec{E}$ ordered by $<_t$, and for each $\vec{e} \in \vec{E}$, there is a 1 at the intersection of the row and the column corresponding to $\vec{e}$.

Claim 6.53. The matrix M_σ is $3r$-grid free.

Proof. Suppose $(\mathcal{R}, \mathcal{C})$ is a $3r$-grid: parts of $\mathcal{R}$ (resp. $\mathcal{C}$) are intervals of $<_s$ (resp. $<_t$), and for each $C \in \mathcal{C}, R \in \mathcal{R}$, there is some $\vec{e} \in C \cap R$.

Consider $\mathcal{P}_s$ the partition of $\vec{E}$ which groups edges with the same starting point. Remark that inside each $P \in \mathcal{P}_s$, the orders $<_s$ and $<_t$ coincide, hence the matrix M_σ restricted to the columns in P has no 2-grid. It follows that there cannot be two distinct parts $C, C' \in \mathcal{C}$ such that $C, C' \subseteq P$ as $(\mathcal{R}, \{C, C'\})$ would yield a 2-grid using only columns of P. Therefore, any given $P \in \mathcal{P}_s$ intersects at most three parts of $\mathcal{C}$. Naturally, the same applies to $\mathcal{R}$ and the partition by endpoints $\mathcal{P}_t$.

We then pick every third part of $\mathcal{C}$ and of $\mathcal{R}$, yielding subsets $\mathcal{C}' \subset \mathcal{C}$ and $\mathcal{R}' \subset \mathcal{R}$ such that each part $P \in \mathcal{P}_s$ (resp. $\mathcal{P}_t$) intersects at most one part of $\mathcal{C}'$ (resp. $\mathcal{R}'$). The families of intervals $\mathcal{C}', \mathcal{R}'$ have size r, and induce an r-grid in the matrix M_σ. By projecting on starting points and endpoints respectively, we obtain $\mathcal{C}'' := \{s(C) \ : \ C \in \mathcal{C}'\}$ and $\mathcal{R}'' := \{t(R) \ : \ R \in \mathcal{R}'\}$, two families of r disjoint intervals of $(V, <_V)$. For any $s(C) \in \mathcal{C}''$, $t(R) \in \mathcal{R}''$, there is an edge $\vec{e} \in C \cap R$, hence $s(\vec{e}) \in s(C)$ is adjacent to $t(\vec{e}) \in t(R)$. This proves that $(\mathcal{C}'', \mathcal{R}'')$ is an r-grid in the adjacency matrix of G, a contradiction. ■

By Claim 6.53 and Theorem 6.6 we obtain a bound on $\mathrm{tww}(\sigma)$ as function of k, t. Applying Theorem 6.21, this yields a factorisation $\sigma = \sigma_m \circ \cdots \circ \sigma_1$, with m again function of k and t. Consider the path system representation $(X, E, <_X)$ of this factorisation, where U is partitioned into the layers $X_0, \ldots, X_m$ as in the previous section. Then $(X_0, <_X)$ and $(X_m, <_X)$ are in naturally in bijection with $(\vec{E}, <_s)$ and $(\vec{E}, <_t)$ respectively, and for each $\vec{e} \in \vec{E}$, there is a path joining the copy of $\vec{e}$ in X_0 to its copy in X_m.

We now add the vertices V of G to this structure, and for each edge $\vec{e} \in \vec{E}$, we connect the copy of $\vec{e}$ in X_0 to $s(\vec{e})$, and the copy of $\vec{e}$ in X_m to $t(\vec{e})$. Thus, for each edge $\vec{e} \in \vec{E}$, a path on m vertices joins $s(\vec{e})$ to $t(\vec{e})$, hence the resulting graph is the $(m+1)$-subdivision of G. Furthermore, for each $v \in V$, the neighbourhood of v inside X_0, resp. X_m, is an interval of $<_X$.

We define an ordering $<$ on the whole structure: it coincides with $<_V$ inside V, with $<_X$ inside X, and has $V < X$. The adjacency matrix of this graph then consists of

1. the adjacency matrix of the path system representation, which is 11-grid free using Lemma 6.48, because separable permutations have no 3-grid, and

2. the adjacency matrix of V against X_0 and X_m, which consists of two increasing sequences (one for X_0, one for X_m), and can be seen to be 3-grid free.

Using arguments similar to the ones from the proof of Lemma 6.48, this can be shown to imply that the adjacency matrix of the $(m+1)$-subdivision of G is 15-grid free. □

6.7.3 Transducing bounded twin-width classes. Recall that for any constant r, there is a first-order interpretation Φ_r such that $\Phi_r(G^{(r)}) = G$ (cf. section 4.1). One can think of the subdivision $G^{(r)}$ as some encoding of G, with the decoding function being the interpretation Φ_r. With this viewpoint, Theorem 6.52 can be restated as such: there is a fixed class $\mathcal{C}$ of graphs with bounded twin-width (the relevant subdivisions), such that any class $\mathcal{D}$ of graphs with twin-width at most k and no $K_{t,t}$ subgraph is interpreted by $\mathcal{C}$ (the interpretation being Φ_r with r function of k, t). We will use Theorem 6.21 to prove other results of the same shape: a fixed class $\mathcal{C}$ transduces any class $\mathcal{D}$ satisfying some conditions.

Let us start with permutations.

Lemma 6.54. *There is a fixed class $\mathcal{C}$ of structures with bounded twin-width such that any bounded twin-width class of permutations is interpreted by $\mathcal{C}$.*

Proof. The class $\mathcal{C}$ consists of all ordered graphs $(V, E, <)$ whose adjacency matrix $A(E, <)$ has no 11 grid. Since separable permutations have no 3-grids in their matrices, by Lemma 6.48, the class $\mathcal{C}$ contains the path system representation of any composition of separable permutations.

Now consider a class $\mathcal{D}$ of permutations with bounded twin-width, and r the bound on the length of factorisations of permutations $\sigma \in \mathcal{D}$ into separable permutations from Theorem 6.21. Given $\sigma \in \mathcal{D}$, consider some factorisation into separable permutations $\sigma = \sigma_r \circ \cdots \circ \sigma_1$, and $(V, E, <) \in \mathcal{C}$ its path system representation. The following first-order interpretation reconstructs the biorder σ from $(V, E, <)$. As in section 6.7.1, denote by X_0 the set of starting points of paths in $(V, E, <)$. Vertices in X_0 can be identified by a first-order formula: a vertex x is in X_0 if and only if x has no neighbour y with $y < x$. We now define a second ordering $\prec$ on X_0: given $x, y \in X_0$, follow the paths in (V, E) starting with x and y to find the respective endpoints x', y'. Then $x \prec y$ if and only if $x' < y'$. The biorder $(X_0, <, \prec)$ is σ. Since the paths joining x, y to x', y' consist of exactly r edges, this can be expressed by a first-order formula depending only on r. This describes a first-order interpretation depending only on r (hence only on $\mathcal{D}$) which given $(V, E, <) \in \mathcal{C}$ reconstructs $\sigma \in \mathcal{D}$. □

To further extend Lemma 6.54, we will admit the following result of Bonnet, Nešetřil, Ossona de Mendez, Siebertz, and Thomassé. It is an encoding of arbitrary binary structures in permutations, which preserves bounded twin-width, and is decoded by first-order transductions.

Theorem 6.55 [21]. *For any class $\mathcal{D}$ of binary structures with bounded twin-width, there is a class of permutations $\mathcal{C}$ with bounded twin-width such that $\mathcal{C}$ transduces $\mathcal{D}$.*

Note that the class $\mathcal{C}$ of encodings in Theorem 6.55 depends on $\mathcal{D}$. Thanks to Lemma 6.54, we strengthen the result by fixing $\mathcal{C}$.

Theorem 6.56 [12, Theorem 7.3]. *There is a fixed class $\mathcal{C}$ of permutations with bounded twin-width such that a class $\mathcal{D}$ of binary structures has bounded twin-width if and only if it is transduced by $\mathcal{C}$.*

Proof. The 'if' part of the claim is Theorem 4.16: the (yet to be defined) class $\mathcal{C}$ has bounded twin-width, hence so does any transduction thereof.

For the other direction, given the class $\mathcal{D}$, we first apply Theorem 6.55 to obtain a class $\mathcal{C}_3$ of permutations with bounded twin-width, and Φ_3 an FO

transduction such that $\mathcal{D} \subseteq \Phi_3(\mathcal{C}_3)$. Next we apply Lemma 6.54 to obtain a class $\mathcal{C}_2$ of ordered graphs with bounded twin-width *independent of the choice of* $\mathcal{D}$, and Φ_2 an FO transduction such that $\mathcal{C}_3 \subseteq \Phi_2(\mathcal{C}_2)$. Finally, we apply Theorem 6.55 a second time to $\mathcal{C}_2$, yielding a class $\mathcal{C}$ of permutations with bounded twin-width and Φ_1 an FO transduction such that $\mathcal{C}_2 \subseteq \Phi_1(\mathcal{C})$. Again, $\mathcal{C}$ is independent of the choice of $\mathcal{D}$.

We conclude by composing the transductions as $\mathcal{C} \xrightarrow{\Phi_1} \mathcal{C}_2 \xrightarrow{\Phi_2} \mathcal{C}_3 \xrightarrow{\Phi_3} \mathcal{D}$. □

Chapter 7

LIMITATIONS OF TWIN-WIDTH: SPARSE GRAPHS AND GROUPS

In the previous chapters, we have seen twin-width as particularly relevant to dense structures such as ordered graphs, permutations, and tournaments. We will take the opposite point of view and focus exclusively on graphs with bounded degree.

Corollary 3.39 proved by a counting argument that graphs with bounded degree can have unbounded twin-width—in fact most of them do. We also explained that *explicitly* constructing graphs with bounded degree and unbounded twin-width remains an open question. While we do not answer this question, this chapter aims at giving a better understanding of twin-width in graphs with bounded degree. We will prove a number of stability results for twin-width on graphs of bounded degree, which lead us to extend twin-width to *infinite groups* through Cayley graphs. We will show that many natural group constructions preserve finiteness of twin-width, and yield numerous examples of groups with finite twin-width. Groups however present the same difficulty as bounded degree graphs: we are unable to explicitly construct groups with infinite twin-width. Nonetheless, the counting argument for bounded degree graphs can be pushed through a powerful embedding result of Osajda to obtain the following.

Theorem 7.1. *There is a finitely generated group with infinite twin-width.*

Theorem 7.1 gives a counter-example to a conjecture asked in [14]: from a group with infinite twin-width, one can create a monotone class of graphs which is small but has infinite twin-width. Thus while classes of bounded twin-width are small (Theorem 3.37), the converse is false.

All results presented in this chapter come from work of Bonnet, Tessera, Thomassé, and the author [16].

7.1 Preliminaries: graphs and groups

Let us first define the objects and notions used throughout this chapter.

7.1.1 Cayley graphs. *Cayley graphs* are the classical way to construct graphs from groups. Given a group Γ (usually infinite) and a finite generating set $S \subset \Gamma$, the graph $\mathrm{Cay}(\Gamma, S)$ has Γ as vertex set, and an edge xy whenever $x \in \Gamma$ and $y = xs$ for some $s \in S \cup S^{-1}$: the edges connect elements of Γ which differ only by a generator. Since S is required to be a generating subset, the graph $\mathrm{Cay}(\Gamma, S)$ is connected, and since S is finite, the degree is bounded by $2|S|$.

For example:

- For cyclic groups, $\mathrm{Cay}(\mathbb{Z}/n\mathbb{Z}, \{1\})$ is the cycle of length n.

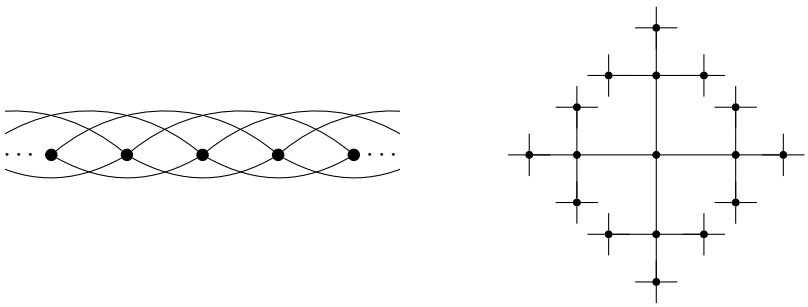


Figure 7.1: On the left, $\mathrm{Cay}(\mathbb{Z}, \{2, 3\})$; on the right, the infinite 4-regular tree $\mathrm{Cay}(\mathbb{F}_2, \{a, b\})$.

- $\mathrm{Cay}(\mathbb{Z}, \{1\})$ is the bi-infinite path, while Figure 7.1 represents another Cayley graph of $\mathbb{Z}$ for a different generating set.
- More generally, if B is the standard basis of $\mathbb{Z}^n$, then $\mathrm{Cay}(\mathbb{Z}^n, B)$ is the n-dimensional infinite grid.
- Consider the free group $\mathbb{F}_2$, that is the group generated by two elements a, b, in which the only equalities which hold are the ones derived from group axioms. Then $\mathrm{Cay}(\mathbb{F}_2, \{a, b\})$ is the infinite 4-regular tree, see Figure 7.1.

  Indeed, one may check that a cycle in a Cayley graph corresponds to a non-trivial equality between products of generators, i.e. an equality which does not follow only from group axioms. Thus $\mathrm{Cay}(\mathbb{F}_2, \{a, b\})$ has no cycle, i.e. is a tree.

7.1.2 Powers of graphs. Cayley graphs are commonly used to extend graph theoretic notions to groups. For example, Kuske and Lohrey consider in [70] *groups with finite tree-width*, meaning groups whose Cayley graphs have finite tree-width, and show the equivalence with group theoretic properies: being context-free and being virtually free.

Naturally, one should be careful that a single group can have several distinct Cayley graphs. In the case of tree-width, this is not an issue: one can prove that if some Cayley graph of a group Γ has finite tree-width, then *all* Cayley graphs of Γ also have finite tree-width. Such a property, satisfied by either all or none of the Cayley graphs of any given group, is called a *group invariant*. By contrast, one might call a group Γ *planar* if some Cayley graph of Γ is planar, but this does certainly not imply that all Cayley graphs of Γ are planar: planarity is not a group invariant. The following notion is helpful to prove that some property is a group invariant.

The *rth power* of a graph G, denoted by G^r, is the graph with the same vertices as G, and such that $x \neq y$ are adjacent in G^r if and only if they are at distance at most r in G. We already used the *square* G^2 as an example of interpretation in section 4.1.

Lemma 7.2. *For any group Γ and finite generating subsets S_1, S_2, there exists $r \in \mathbb{N}$ such that $\mathrm{Cay}(\Gamma, S_1)$ is a subgraph of $(\mathrm{Cay}(\Gamma, S_2))^r$.*

Proof. Since S_2 generates Γ, any $s \in S_1$ is equal to some finite product of elements of $S_2 \cup S_2^{-1}$. Furthermore, there is a maximum length r to these products since S_1 is finite. Now consider $x \in \Gamma$ and $s \in S_1$ which decomposes as $s = t_1 \cdot \cdots \cdot t_\ell$ with $t_i \in S_2 \cup S_2^{-1}$ and $\ell \leqslant r$. Then in $\mathrm{Cay}(\Gamma, S_2)$, $x, xt_1, xt_1t_2, \ldots, xt_1 \ldots t_\ell$ is a path of length ℓ from x to xs. Thus for any edge xy of $\mathrm{Cay}(\Gamma, S_1)$, there is a path of length at most r connecting x to y in $\mathrm{Cay}(\Gamma, S_2)$. □

Thus, Cayley graphs of the same group are subgraphs of powers of each other.

Now, continuing with the example of tree-width: one can show that if G is a graph with bounded degree and finite tree-width, then G^r also has finite tree-width for any $\mathbb{N}$. Together with Lemma 7.2, this implies that finite tree-width is a group invariant. We will show a similar result for twin-width.

7.1.3 Coarse geometry. Powers of graphs are enough to capture the resemblance between Cayley graphs of the same group, but it is common to use the more general and flexible notion of *quasi-isometry*. The notion is defined for arbitrary metric spaces, but we will mostly use it in graphs, where the points of the space are vertices and the distance is the shortest path metric.

Consider two metric spaces X, Y, with the associated metrics d_X and d_Y. A map $f : X \to Y$ is a *λ-quasi-isometric embedding* if for all $x, x' \in X$,

$$\lambda^{-1} d_X(x, x') - \lambda \ \leqslant\ d_Y\left(f(x), f(x')\right) \ \leqslant\ \lambda d_X(x, x') + \lambda.$$

That is, f preserves distances up to affine lower and upper bound. The map f is a *λ-quasi-isometry* if in addition it is λ-cobounded, meaning that every $y \in Y$ is at distance at most λ of $f(X)$. For example,

- The identity map on $V(G)$ is an r-quasi-isometry from G to G^r, and also from G^r to G: indeed the distances differ by a factor of at most r, while coboundedness is trivial as the map is bijective.

- Consider $\mathbb{Z}^n$ and $\mathbb{R}^n$ as metric spaces with the maximum norm. The inclusion $\mathbb{Z}^n \to \mathbb{R}^n$ is a $\frac{1}{2}$-quasi-isometry: it exactly preserves distances, and any point of the euclidean space is at distance at most $\frac{1}{2}$ of an integer point. The retraction $r : \mathbb{R}^n \to \mathbb{Z}^n$ which rounds coordinates to the closest integer is a 1-quasi-isometry: the distance $d(x, r(x))$ is at most $\frac{1}{2}$ for any point x, hence distances between pairs of points are distorted by at most 1.

- Given any graph G, there are two common ways to create a metric space: one can see $V(G)$ as a discrete space with the shortest path metric (which is our standard point of view in this chapter), or consider the *realisation* of G where each edge is replaced by a copy of the segment $[0, 1]$ connecting the vertices. The inclusion of $V(G)$ into the realisation of G is a $\frac{1}{2}$-quasi-isometry, while the retraction onto $V(G)$ mapping to the closest vertex is a 1-quasi-isometry.

The spaces X and Y are said to be quasi-isometric if there is a λ-quasi-isometry $X \to Y$ for some λ. It is simple to check that quasi-isometries are closed under composition. Further, given a quasi-isometry $f : X \to Y$, one can

find a *coarse inverse* $g : Y \to X$, that is a second quasi-isometry such that $f \circ g$ and $g \circ f$ are at bounded distance of the identity maps. It follows from these remarks that being quasi-isometric is an equivalence relation on metric spaces.

Coarse geometry is concerned with metric spaces considered up to quasi-isometry (or even broader notions of approximate distances). Lemma 7.2 implies that all Cayley graphs of a given group Γ are quasi-isometric through the identity map on Γ. Thus coarse geometry allows to consider the group Γ as a metric space up to quasi-isometry, without needing to specify a Cayley graph.

To prove stability under quasi-isometries, we will use the following decomposition into a few simpler operations, the main one being a power graph. Given a graph G and a partition $\mathcal{P}$ of $V(G)$, the *quotient graph* $G/\mathcal{P}$ has vertex set $\mathcal{P}$, and its edges are all XY such that there is an edge between the parts X and Y in G.

Lemma 7.3. *If $f : V(G) \to V(H)$ is a quasi-isometric embedding of graphs of bounded degree, then there is a partition $\mathcal{P}$ of $V(G)$ into parts of bounded size and a radius $r \in \mathbb{N}$ such that $G/\mathcal{P}$ is a subgraph of H^r.*

Proof. Let $\mathcal{P} = \{f^{-1}(x) \mid x \in V(H)\}$ be the partition into preimages. Since f is quasi-isometric, parts $P \in \mathcal{P}$ have bounded diameter, hence bounded size since G has bounded degree.

If x, y are adjacent in G, then $f(x), f(y)$ are at bounded distance in H, say at most r, hence$f(x), f(y)$ are either equal or adjacent in H^r. This implies that $G/\mathcal{P}$ is a subgraph of H. □

7.2 Twin-width of groups

We will now relate twin-width to the constructions introduced in the previous section, leading to a notion of *finite twin-width* for groups.

7.2.1 Sparse twin-width. Let us begin by defining a simplified variant of twin-width, tailored for graphs of bounded degree.

Recall that the maximum degree in a graph H is denoted by $\Delta(H)$. The *sparse twin-width* $\mathrm{stww}(G)$ is the minimum over all contraction sequences $\mathcal{P}_n, \dots, \mathcal{P}_1$ of $\max_i \Delta(G/\mathcal{P}_i)$. The difference with twin-width is that the distinction between normal and error edges is forgotten.

For graphs with maximum degree Δ, twin-width and sparse twin-width differ by at most Δ.

Lemma 7.4. *Any graph G satisfies*

$$\max(\mathrm{tww}(G), \Delta(G)) \leqslant \mathrm{stww}(G) \leqslant \mathrm{tww}(G) + \Delta(G).$$

Proof. The left inequality is immediate. The right inequality is obtained by remarking that for any partition $\mathcal{P}$ of $V(G)$, any vertex in $Tri(G, \mathcal{P})$ is incident to no more than $\Delta(G)$ normal edges, and thus the total degree of a part $X \in \mathcal{P}$, which is equal to the degree of X in $G/\mathcal{P}$, is bounded by $\Delta(G)+\deg^{Err}(X)$. □

Notice that sparse twin-width is monotone under taking *non-induced* subgraphs, quite unlike twin-width.

Lemma 7.5. *If H is a subgraph of G, then* $\mathrm{stww}(H) \leqslant \mathrm{stww}(G)$.

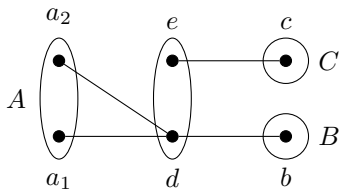


Figure 7.2: A graph G with a partition $\mathcal{P}$. In both $G^2/\mathcal{P}$ and $(G/\mathcal{P})^2$, parts A and B are adjacent because of the path a_1, d, b. In $(G/\mathcal{P})^2$ only, A and C are adjacent thanks to the path a_1, d, e, c, which can 'jump' from d to e since they are in the same part of $\mathcal{P}$.

Proof. If H is a subgraph of G and $\mathcal{P}$ is any partition of $V(G)$, then $H/\mathcal{P}$ (where $\mathcal{P}$ is identified with its restriction to $V(H)$) is a subgraph of $G/\mathcal{P}$, and thus $\Delta(H/\mathcal{P}) \leqslant \Delta(G/\mathcal{P})$. The result easily follows by applying this inequality to a contraction sequence for G. □

7.2.2 Power graphs. In section 4.1, squares of graphs were presented as a special case of first-order interpretation. More generally, for any r there is a first-order formula $\phi(x, y)$ expressing that x, y are at distance at most r, and thus a first-order interpretation $G \mapsto G^r$. Thus, Theorem 4.16 implies that $\mathrm{tww}(G^r) \leqslant f_r(\mathrm{tww}(G))$ for some function f_r depending only on r.

Using sparse twin-width, we can now give a simple proof of this special case in the context of bounded degree graphs.

Lemma 7.6. *For any $r \in \mathbb{N}$, and any graph G,*

$$\mathrm{stww}(G^r) \leqslant \mathrm{stww}(G)^r.$$

Proof. We will use the following easily verified inequality (which is not tight, but is certainly convenient to remember):

$$\Delta(G^r) \leqslant (\Delta(G))^r. \tag{7.1}$$

Consider $\mathcal{P}$ a partition of $V(G)$. Then

$$G^r/\mathcal{P} \quad \text{is a subgraph of} \quad (G/\mathcal{P})^r. \tag{7.2}$$

Indeed, for $X, Y \in \mathcal{P}$, there is an edge XY in $G^r/\mathcal{P}$ if and only if there are $x \in X$, $y \in Y$ and a path of length at most k in G from x to y. When that is the case, the parts containing the successive vertices of this path yield a path from X to Y of length at most k in $G/\mathcal{P}$, see Figure 7.2.

Now if $\mathcal{P}_n, \ldots, \mathcal{P}_1$ is a contraction sequence for G whose quotient graphs satisfy $\Delta(G/\mathcal{P}_i) \leqslant k$, then $(G/\mathcal{P}_i)^r$ has degree bounded by k^r by (7.1), hence so does $G^r/\mathcal{P}_i$. Thus $\mathcal{P}_n, \ldots, \mathcal{P}_1$ now seen as contraction sequence for G^r shows that $\mathrm{stww}(G^r) \leqslant k^r$. □

Using Lemmas 7.2 and 7.6, we are almost ready prove that finite twin-width is a group invariant. There remains only one key issue: Cayley graphs are usually infinite, and we have not yet defined twin-width for infinite graphs.

7.2.3 Infinite graphs. Our goal is thus to extend twin-width to infinite graphs.

Let us sketch one possible definition using infinite contraction sequences. A contraction sequence for an infinite graph G is a family $(\mathcal{P}_i)_{i\in I}$ of partitions of $V(G)$, indexed by an arbitrary linear order $(I, <)$. The sequence should progress by coarsening the partitions, i.e. for $i < j$, $\mathcal{P}_i$ refines $\mathcal{P}_j$. Further, we should ensure that $\mathcal{P}_i$ converges to the singletons and trivial partitions on the respective ends of I, and that it does not skip steps along the way. This can be specified with a somewhat tedious list of axioms, or simply by requiring that for any finite $X \subset V(G)$, the restriction of $(\mathcal{P}_i)_{i\in I}$ to X be a contraction sequence in the usual sense, after removing steps where the partition does not change.

With such a definition, we would next prove using classical compactness arguments that twin-width is continuous in the following sense:

$$\operatorname{tww}(G) = \sup\{\operatorname{tww}(H) \mid H \text{ finite induced subgraph of } G\}. \tag{7.3}$$

Characterisation (7.3) turns out to be how we will manipulate twin-width of infinite graphs. Rather than go through these length, we choose to simply take (7.3) as the definition of twin-width in infinite graphs. We similarly extend sparse twin-width to infinite graphs by considering the supremum over finite subgraphs.

With this new definition, let us show how to generalise some useful lemmas on twin-width to infinite graphs. This is not meant to be exhaustive: most of the generalisations to infinite graphs are immediate. Rather, the goal is to illustrate the difficulties which can appear in these generalisations, and how to overcome them.

Subgraphs If H is an (infinite) subgraph of G, then any finite subgraph of H is also a finite subgraph of G. It follows that $\operatorname{stww}(H) \leqslant \operatorname{stww}(G)$, generalising Lemma 7.5.

Power graphs Let G be an infinite graph with $\operatorname{stww}(G) = k$. For $r \in \mathbb{N}$, consider H a finite subgraph of G^r with vertex set X. It is not true that H is a subgraph of $(G[X])^r$: vertices x, y in X might be adjacent in G^r only thanks to a path which leaves X, in which case x, y are not adjacent in $(G[X])^r$. However one can consider Y the set of vertices at distance at most r of X in G, which is finite since G has bounded maximum degree. Now it is simple to check that H is a subgraph of $(G[Y])^r$, and $\operatorname{stww}(G[Y]) \leqslant k$, hence $\operatorname{stww}(H) \leqslant k^r$ by Lemma 7.6. This implies $\operatorname{stww}(G^r) \leqslant k^r$, generalising Lemma 7.6.

Compatible orders Of all the results presented in this work, the existence of compatible orders for contraction sequences (Lemma 5.2) stands out for being particularly simple for finite graphs, but requiring some care in the infinite case. Indeed this is where we have to pay for the choice of defining twin-width through finite subgraphs by having to construct some infinite structure: a compatible order.

Lemma 7.7. *For any (possibly infinite) binary structure S there exists a total ordering $<$ of $V(S)$ such that*

$$\operatorname{tww}(S) = \operatorname{tww}(S, <).$$

To prove Lemma 7.7, we will use a construction of *limit* ordering.

Let S be some structure (in a very broad sense: in our case, a group, a graph, a relational structure), and $(S_i)_{i\in I}$ a family of substructures of S. The family $(S_i)_{i\in I}$ is *directed* if for all $i, j \in I$, there exists $k \in I$ with $S_i \cup S_j \subseteq S_k$. The structure S is said to be the direct limit of $(S_i)_{i\in I}$ if $(S_i)_{i\in I}$ is directed and $S = \bigcup_{i\in I} S_i$. For instance, an increasing sequence is a very special case of directed families; a graph is always the direct limit of all its finite induced subgraphs; and similarly a group is the direct limit of its finitely generated subgroups.

The following is a folklore lemma, whose proof we omit. It can be proven using compactness, ultrafilters, or simply Kőnig's lemma in the countable case.

Lemma 7.8. *Let S be the direct limit of $(S_i)_{i\in I}$, and for each $i \in I$ let $<_i$ be a total ordering of S_i. Then there exists a limit ordering $<$ on S such that for any finite $X \subset S$, there is some $i \in I$ with $X \subseteq S_i$ and such that $<$ and $<_i$ coincide inside X.*

Proof of Lemma 7.7. Let S be an infinite structure with twin-width k. For any finite induced substructure $T \subset S$, we have $\mathrm{tww}(T) \leqslant k$, hence there is by Lemma 5.2 an ordering $<_T$ such that the *ordered* structure $(T, <_T)$ has twin-width at most k.

Now S is the direct limit of its finite substructures. Then, by Lemma 7.8, we obtain a total ordering $<$ of S as limit of the $<_T$. We claim that $\mathrm{tww}(S, <) \leqslant k$. Indeed, for any finite $X \subset V(S)$, there is some substructure T of S containing X such that $<$ and $<_T$ coincide on X. Thus, $(S, <)$ restricted to X is a substructure of $(T, <_T)$, which has twin-width at most k. □

7.2.4 Group invariant. We are finally ready to extend twin-width to groups. Consider a group Γ, and two finite generating subsets S_1, S_2. By Lemma 7.2, $\mathrm{Cay}(\Gamma, S_2)$ is a subgraph of some power of $\mathrm{Cay}(\Gamma, S_1)$, hence if the latter has finite sparse twin-width, then so does the former by Lemmas 7.5 and 7.6. Hence finite sparse twin-width (or equivalently finite twin-width, since the graphs have bounded degree) is a group invariant, and we say that a finitely generated group has finite twin-width if its Cayley graphs do. For example

1. Finite groups trivially have finite twin-width.
2. The free group $\mathbb{F}_2$ admits the 4-regular tree as Cayley graph, hence has finite twin-width by Fact 2.4.
3. The group $\mathbb{Z}^n$ admits the n-dimensional grid as Cayley graph, hence has finite twin-width by Fact 2.6.
4. The *cartesian product* of graphs G_1, G_2 is the graph $G_1 \times G_2$ with vertex set $V(G_1) \times V(G_2)$, and an edge between $(x_1, x_2), (y_1, y_2)$ whenever x_1 and y_1 are adjacent and $x_2 = y_2$, or vice versa. It is simple to show that

$$\mathrm{stww}(G_1 \times G_2) < 2(\mathrm{stww}(G_1) + 1)(\mathrm{stww}(G_2) + 1). \tag{7.4}$$

If Γ_1, Γ_2 are groups with Cayley graphs $G_i = \mathrm{Cay}(\Gamma_i, S_i)$, then

$$G_1 \times G_2 = \mathrm{Cay}(\Gamma_1 \times \Gamma_2, S_1 \cup S_2), \tag{7.5}$$

i.e. the cartesian product of Cayley graphs, is a Cayley graph of the cartesian product of groups. It follows that $G_1 \times G_2$ has finite twin-width if and only if G_1 and G_2 do.

5. Finitely generated abelian groups are well-known to be the groups constructed by cartesian products when starting from cyclic groups and $\mathbb{Z}$. These starting groups have finite twin-width by (1) and (3), hence all finitely generated abelian groups have finite twin-width.

Let us point out that while having finite or infinite twin-width is a group invariant, we cannot define the 'twin-width of a group' as some meaningful number: the different Cayley graphs of the same group have different twin-width, and if Γ is an infinite group with finite twin-width, one can create Cayley graphs with arbitrarily large twin-width for Γ by picking sufficiently complex generating sets.

7.2.5 Quasi-isometric invariant. The previous proof that twin-width is a group invariant extends to quasi-isometries.

Lemma 7.9. *If G, H are graphs with bounded degree, G has finite sparse twin-width and H quasi-isometrically embeds into G, then H also has finite sparse twin-width.*

Proof. By Lemma 7.3, $H/\mathcal{P}$ is a subgraph of G^r for some partition $\mathcal{P}$ into parts of bounded size and radius $r \in \mathbb{N}$. Lemmas 7.5 and 7.6 immediately give that $H/\mathcal{P}$ has finite sparse twin-width. Finally, a simple variant of Lemma 3.16 shows that since $H/\mathcal{P}$ has finite sparse twin-width and parts in $\mathcal{P}$ have bounded size, H also has finite sparse twin-width. □

Finite twin-width being closed under quasi-isometry, it is natural to compare it to other classical quasi-isometric invariants of groups. Hyperbolicity in the sense of Gromov [57] is one such notion. By a remarkable result of Buyalo, Dranishnikov, and Schroeder, hyperbolic groups—and more generally, hyperbolic graphs of bounded degree—quasi-isometrically embed into finite cartesian products of trees of bounded degree [27]. Since trees of bounded degree have finite twin-width, and cartesian products preserve it, we obtain the following.

Fact 7.10. *Gromov-hyperbolic groups have finite twin-width.*

7.3 Permutation characterisation

The definition of twin-width of groups through Cayley graphs is natural, and interesting since it shows that twin-width is a quasi-isometric invariant. It is however not the most convenient when considering simple algebraic group constructions such as products, quotients, etc. This section gives a second characterisation of twin-width of groups through permutations which is more helpful for this purpose. It also allows to generalise twin-width beyond finitely generated groups, and leads to a natural strengthening, called *uniform twin-width*.

7.3.1 Twin-width of group actions. Let Γ be a group acting on a (possibly infinite) set X by $\phi : \Gamma \to \mathfrak{S}_X$, where $\mathfrak{S}_X$ denotes the group of permutations on X. If we fix a reference ordering $<$ of X, then each permutation $\phi(g)$, $g \in \Gamma$ can be encoded by the biorder $(X, <, <_{\phi(g)})$ where,

$$x <_{\phi(g)} y \quad \iff \quad \phi(g)(x) < \phi(g)(y) \tag{7.6}$$

(cf. section 6.1). We say that the action of Γ through ϕ has finite twin-width with regards to $<$ if the twin-width of the biorder $(X, <, <_{\phi(g)})$ is finite for every $g \in \Gamma$—but possibly arbitrarily large. This action is said to have finite twin-width if it has finite twin-width with regards to some ordering of X.

Lemma 7.11. *For a finitely generated group Γ, the following are equivalent.*

1. *Some (equivalently, every) Cayley graph of Γ has finite twin-width.*
2. *The action of Γ on itself by right product has finite twin-width.*

Proof. Suppose that the action of Γ on itself by right product has finite twin-width, witnessed by some total ordering $<$ of Γ: for any $g \in \Gamma$, if $<_g$ is the permuted order defined by $x <_g y$ if and only if $xg < yg$, then the biorder $(\Gamma, <, <_g)$ has finite twin-width.

Define the edge set $E_g = \{(x, xg) \mid x \in \Gamma\}$. Theorem 6.1 implies that since the biorder $(\Gamma, <, <_g)$ encoding of $x \mapsto xg$ has finite twin-width, so does the ordered bijection $(\Gamma, <, E_g)$. Consider now an arbitrary finite generating subset $S = \{s_1, \dots, s_k\}$, and assume without loss of generality that $S = S^{-1}$. Each $(\Gamma, <, E_{s_i})$ has finite twin-width, and since they are ordered graphs, we obtain by Lemma 5.6 that their superposition

$$(\Gamma, <, E_{s_1}, \dots, E_{s_k})$$

also has finite twin-width. It is then simple to verify that replacing all the relations $E_{s_1}, \dots, E_{s_k}$ by their union cannot increase the twin-width. Thus

$$(\Gamma, <, E_{s_1} \cup \cdots \cup E_{s_k})$$

has finite twin-width. The latter is exactly the Cayley graph $\mathrm{Cay}(\Gamma, S)$ with the ordering $<$ added. Finally, since twin-width is a group invariant, we obtain that all Cayley graphs of Γ have finite twin-width.

Conversely, assume that $\mathrm{Cay}(\Gamma, S)$ has finite twin-width for some finite generating set $S = \{s_1, \dots, s_k\}$. Once again, we write

$$\mathrm{Cay}(\Gamma, S) = (\Gamma, E_{s_1} \cup \cdots \cup E_{s_k}).$$

Since $\mathrm{Cay}(\Gamma, S)$ has bounded degree, it also has finite sparse twin-width. With sparse twin-width, it is simple to check that splitting the edges into several relations as

$$(\Gamma, E_{s_1}, \dots, E_{s_k})$$

preserves finite twin-width. Now using Lemma 7.7, there exists a total ordering $<$ of Γ such that the ordered structure

$$(\Gamma, <, E_{s_1}, \dots, E_{s_k})$$

has finite twin-width. A fortiori, this implies that each $(\Gamma, <, E_s)$ has finite twin-width for any $s \in S$. Then by Theorem 6.1, the biorder $(\Gamma, <, <_s)$ also

has finite twin-width. It only remains to extend this to elements $g \in \Gamma$ outside the generating set S. Since S is a generating subset, any permutation $x \mapsto xg$, $g \in \Gamma$ can be written as composition of permutations $x \mapsto xs$ with $s \in S$. By Corollary 5.7, composition of permutation preserves finite twin-width, which implies the result. □

Thus we can take as alternative definition that Γ has finite twin-width if its self action by right product has finite twin-width. Notice that this new definition does not refer to a generating set, and thus generalises finite twin-width to non-finitely generated groups.

7.3.2 Uniform twin-width. In the former characterisation, each biorder $(X, <, <_{\phi(g)})$ is required to have finite twin-width, but possibly arbitrarily large. A natural strengthening is to ask for a uniform bound. Consider an action of Γ on X given by $\phi : \Gamma \to \mathfrak{S}_X$. Given a total order $<$ on X, the *uniform twin-width* of ϕ with regards to $<$ is

$$\mathrm{utww}_{(X,<)}(\phi) = \sup_{g \in \Gamma} \mathrm{tww}\,(X, <, <_{\phi(g)}).$$

Then, the uniform twin-width of ϕ, denoted by $\mathrm{utww}(\phi)$, is the minimum of $\mathrm{utww}_{(X,<)}(\phi)$ over all choices of the ordering $<$ of X. The uniform twin-width of a group Γ, denoted $\mathrm{utww}(\Gamma)$, is defined as the uniform twin-width of its self action by right product. Unlike twin-width, the uniform twin-width of a group or group action is a well defined number, or infinity.

With these new definitions using permutations, we obtain a new important example of groups with finite twin-width: A *right-invariant* ordering of a group Γ is a total ordering $<$ such that for all $x, y, z \in \Gamma$, $x < y$ implies $xz < yz$. If Γ admits such an ordering, it is called *(right-)orderable*. Groups with finite twin-width can be seen as a vast generalisation of orderable groups. Indeed, if $<$ is a right-invariant order on Γ, then the action of any $x \in \Gamma$ by right product is monotone, hence the corresponding permutation has twin-width 0. Thus,

Fact 7.12. *Any right-orderable group has uniform twin-width 0.*

Let us refine the examples of section 7.2. Finitely generated free groups are well-known to be orderable (see e.g. [86] for a simple proof), thus they have uniform twin-width 0. Further, it is easy to see that finite cyclic groups with the natural ordering have uniform twin-width 0.

Now suppose that Γ_1, Γ_2 are two groups with orderings $<_1, <_2$, and consider the Cartesian product $\Gamma_1 \times \Gamma_2$ with the lexicographic ordering. Then the permutation corresponding to the right multiplication by $(g_1, g_2) \in \Gamma_1 \times \Gamma_2$ can be obtained by substitution (in the sense of section 6.2.4) from the permutations $x \mapsto xg_i$ in Γ_i for $i = 1, 2$. By Lemma 6.16, it follows that Cartesian product preserves uniform twin-width:

$$\mathrm{utww}(\Gamma_1 \times \Gamma_2) = \max(\mathrm{utww}(\Gamma_1), \mathrm{utww}(\Gamma_2)). \tag{7.7}$$

Thus, since cyclic groups and $\mathbb{Z}$ have uniform twin-width 0, and Cartesian product preserves uniform twin-width, we obtain

Fact 7.13. *Any finitely generated abelian group has uniform twin-width 0.*

The next section will present generalisations of this stability result and many other examples of groups with small uniform twin-width.

7.4 Groups with finite twin-width

The goal of this section is to show that twin-width of groups, and especially uniform twin-width, is preserved by many common group constructions, leading to several examples of groups with finite twin-width. In this section, unlike the rest of this chapter, we use G, H etc. to denote groups and not graphs.

Let us start with the trivial remark that twin-width is monotone under taking subgroups. It is used to prove lower bounds in results of this section.

Lemma 7.14. *If G is a group, $H \leqslant G$ is a subgroup, and G has finite twin-width, then so does H, and* $\mathrm{utww}(H) \leqslant \mathrm{utww}(G)$.

7.4.1 Limits. Defining twin-width of groups through permutations allows to extend it to infinitely generated groups. For uniform twin-width, we will show that it is in fact enough to consider finitely generated groups: any group is the direct limit of finitely generated groups, and these limits preserve uniform twin-width.

Lemma 7.15. *If G is the direct limit of $(G_i)_{i\in I}$, then*

$$\mathrm{utww}(G) = \sup_{i\in I} \mathrm{utww}(G_i).$$

Proof. For each G_i, let $<_i$ be an ordering witnessing that $\mathrm{utww}(G_i) \leqslant k$. By Lemma 7.8, there is an ordering $<$ of G such that for any finite $X \subset G$, the orders $<$ and $<_i$ coincide on X for some i.

Consider the biorder $B_g = (G, <, <_g)$ representing the action of some $g \in G$ on $(G, <)$ by right product. Given a finite substructure of B_g with vertex set X, define $Y = X \cup (X \cdot g) \cup \{g\}$. There is some G_i containing Y such that $<$ and $<_i$ coincide on Y. Since $<_i$ witnesses that $\mathrm{utww}(G_i) \leqslant k$, the biorder $B'_g = (G_i, <_i, <_{i,g})$ has twin-width at most k. Furthermore, using that $<$ and $<_i$ coincide on Y, one may check that the induced substructures $B_g[X]$ and $B'_g[X]$ are equal. Thus $\mathrm{tww}(B_g[X]) \leqslant k$, which implies the result. □

Corollary 7.16. *Let G be a group and let $\mathcal{H}$ be the collection of finitely generated subgroups of G. Then*

$$\mathrm{utww}(G) = \sup_{H\in\mathcal{H}} \mathrm{utww}(H).$$

Proof. The group G is the direct limit of $\mathcal{H}$. □

In section 7.3, we already proved that finitely generated abelian groups have uniform twin-width 0. Applying Corollary 7.16, we obtain

Fact 7.17. *Abelian groups have uniform twin-width 0.*

7.4.2 Products and quotients. In sections 7.2 and 7.3, we remarked that (uniform) twin-width is stable under Cartesian product. We will now give a significant generalisation of this result.

Let G be a group and $H \leqslant G$. We write G/H for the right quotient, i.e. the set of right cosets $\{Hg \mid g \in G\}$. Then, G acts on G/H by right product, and we abusively talk about the (uniform) twin-width of G/H when meaning

the (uniform) twin-width of this action. When H is a normal subgroup, the (uniform) twin-width of G/H as group indeed coincides with that of G acting on G/H, justifying this convention.

Lemma 7.18. *For any group G and $H \leqslant G$, the following holds.*

1. $\mathrm{utww}(G) \leqslant \max(\mathrm{utww}(H), \mathrm{utww}(G/H))$.
2. *If H has finite* uniform *twin-width and G/H has finite twin-width, then G has finite twin-width.*
3. *If H has finite twin-width and G/H is finite, then G has finite twin-width.*

Proof. For $x \in G$, let $\bar{x} = Hx$ be its equivalence class in G/H. We choose a transversal $T \subset G$ of G/H, meaning that each class of G/H can be uniquely written as $\bar{t}$ for some $t \in T$. Consider orderings $<_H$, $<_{G/H}$ of H and G/H witnessing their twin-width or uniform twin-width. We define an ordering $<$ of G as follows.

- The order between cosets is defined by
$$\bar{x} <_{G/H} \bar{y} \quad \Longrightarrow \quad x < y.$$
- For $t \in T$, the order inside the coset $\bar{t}$ is defined by
$$\forall x, y \in \bar{t}, \quad x < y \quad \Longleftrightarrow \quad xt^{-1} <_H yt^{-1}.$$

For $a \in G$, let $B_a^G = (G, <, <_a)$ be the biorder encoding the action of a on G ordered by $<$. Similarly, $B_a^{G/H}$ encodes the action of a on G/H ordered by $<_{G/H}$, and for $b \in H$, B_b^H encodes the action of b acting on H ordered by $<_H$. Fix some $a \in G$. The crucial remark is the following.

Claim 7.19. Suppose that $\mathrm{tww}(B_a^{G/H}) \leqslant k$, and that for any $t_1, t_2 \in T$, for $r = t_1 a t_2^{-1} \in H$, we have $\mathrm{tww}(B_r^H) \leqslant k$. Then $\mathrm{tww}(B_a^G) \leqslant k$.

Let us first show that the claim implies the three points of the lemma:

1. is immediate.
2. Given a uniform bound on $\mathrm{tww}(B_r^H)$ for all $r \in H$, as well as a bound on $\mathrm{tww}(B_a^{G/H})$, the claim gives a bound on $\mathrm{tww}(B_a^G)$. Remark that since the bound on $\mathrm{tww}(B_r^H)$ is used for all appropriate $r = t_1 a t_2^{-1} \in H$, the uniform bound is a priori required.
3. If G/H is finite however, there are only finitely many $t_1, t_2 \in T$ satisfying $t_1 a t_2^{-1} \in H$ (for a fixed), hence the uniform bound in the previous argument is no longer required.

Proof of the Claim. We want to bound the twin-width of $B_a^G = (G, <, <_a)$. By construction of $<$, any coset $\bar{t}_1$ is an interval of $<$. Furthermore, the image $\bar{t}_1 \cdot a$ is also a coset $\bar{t}_2$, hence $\bar{t}_1$ is also an interval of $<_a$. This allows to construct B_a^G as a substitution of permutations along a tree with two levels: the top level is the permutation $(G/H, <, <_a)$ on the cosets, and the bottom level consists of permutations $(\bar{t}_1, <, <_a)$ for each coset $\bar{t}_1$. By Lemma 6.16, we

only need to bound the twin-width of each of the aforementioned substitutions independently.

The quotient permutation $(G/H, <, <_a)$ is exactly $B_a^{G/H}$, which has twin-width at most k by assumption. Now consider a coset $\bar{t}_1 = Ht_1$, which is mapped to some $\bar{t}_2 = Ht_2$ by right multiplication by a. In this case $t_1 a t_2^{-1} \in H$, and we call $r = t_1 a t_2^{-1}$. Now for any $x, y \in \bar{t}_1$, we have by definition of $<$

$$x < y \quad \iff \quad x t_1^{-1} <_H y t_1^{-1},$$

and since $xa, ya \in \bar{t}_2$,

$$\begin{aligned} x <_a y &\iff xa < ya \\ &\iff x a t_2^{-1} <_H y a t_2^{-1} \\ &\iff (x t_1^{-1}) r <_H (y t_1^{-1}) r. \end{aligned}$$

It follows that the map $x \mapsto x t_1^{-1}$ is an isomorphism from $(\bar{t}_1, <, <_a)$ to B_r^H, giving the desired bound on $\mathrm{tww}(\bar{t}_1, <, <_a)$. ■

This completes the proof of Lemma 7.18. □

Lemma 7.18 is a fairly technical statement, but has natural corollaries.

Corollary 7.20. *If G is an extension of Q by H, i.e. $H \lhd G$ and $Q \cong G/H$*

$$\mathrm{utww}(G) \leqslant \max(\mathrm{utww}(H), \mathrm{utww}(Q)).$$

Corollary 7.21. *If $G \rtimes H$ is any semi-direct product, then*

$$\mathrm{utww}(G \rtimes H) = \max(\mathrm{utww}(G), \mathrm{utww}(H)).$$

Corollary 7.22. *If $H \leqslant G$ is a subgroup of finite index $[G : H] = k$, then G has finite twin-width if and only if H does, and*

$$\mathrm{utww}(H) \leqslant \mathrm{utww}(G) \leqslant \max(\mathrm{utww}(H), k).$$

With these results, we can give new examples of groups with finite twin-width. The well-known class of *solvable groups*, originating from Galois theory, is defined inductively as follows.

- The trivial group is solvable.
- If $H \lhd G$, H is solvable and G/H is abelian, then G is solvable.

Using Fact 7.17 and Corollary 7.20, we immediately obtain

Fact 7.23. *Solvable groups have uniform twin-width 0.*

A particularly important subclass of solvable groups is *nilpotent groups* (which we will not define). They are part of the following extremely famous result of Gromov. If Γ is a group finitely generated by S, its *growth function* is

$$g_{(\Gamma,S)}(r) = |B(1_\Gamma, r)|,$$

where $B(1_\Gamma, r)$ is the ball of radius r around the identity element in $\mathrm{Cay}(\Gamma, S)$. This function depends on the choice of S, but it is simple to check that whether or not it is e.g. a polynomial or exponential function, is independent of S. Thus one can talk about groups with polynomial or with exponential growth. Gromov proved that any group G with polynomial growth is *virtually nilpotent* [56]: there is some nilpotent subgroup $H \leqslant G$ with finite index. From Fact 7.23 and Corollary 7.22, we thus obtain

Fact 7.24. *Groups of polynomial growth have finite uniform twin-width.*

7.4.3 Infinite products and wreath products. Given $(G_i)_{i\in I}$ a family of groups, their *direct product* is denoted by $\prod_{i\in I} G_i$. Their *direct sum* $\bigoplus_{i\in I} G_i$ is the subgroup of the direct product consisting of finitely supported tuples, i.e. the $(x_i)_{i\in I}$ with $x_i \in G_i$ such that all but finitely many x_i are the identity element. When I is finite, the two notions coincide.

Lemma 7.18 of course implies that finite products preserve twin-width and uniform twin-width. We will show that this generalises to infinite direct sum, and for uniform twin-width only, to infinite direct products. To prove this, we first need a corresponding result on permutations: stability under substitutions with infinite depth.

Consider $(I, <_I)$ a well-ordered set of indices, and for each $i \in I$ a biorder $B_i = (X_i, <_i, \prec_i)$. On the product $X = \prod_{i\in I} X_i$, we define the orderings $<, \prec$ lexicographically: given $x = (x_i)_{i\in I}$ and $y = (y_i)_{i\in I}$ in X, we have $x < y$ if and only if $x_i <_i y_i$ where i is the minimal index on which x and y differ. The second ordering $\prec$ is defined similarly. The resulting biorder $(X, <, \prec)$ is called *lexicographic product* of $(B_i)_{i\in I}$, denoted $\prod_{i\in(I,<_I)} B_i$. Note that the only reason $<_I$ needs to be well-founded is for the lexicographic orderings to be well-defined.

Lemma 7.25. *Let $(B_i)_{i\in I}$ be a family of biorder indexed by a well-ordered set $(I, <_I)$ of indices, and $B = \prod_{i\in(I,<_I)} B_i$ its lexicographic product. Then* $\mathrm{tww}(B) = \sup_{i\in I} \mathrm{tww}(B_i)$.

Proof. Suppose that $\mathrm{tww}(B_i) \leqslant k$ for any i, and let us show that $\mathrm{tww}(B) \leqslant k$. By definition of twin-width for infinite structures, it suffice to show that for any finite subset of vertices $X \subset V(B)$, the substructure $B[X]$ has twin-width at most k.

Fix such a finite X. For any two $x \neq y \in X$, consider the minimal index in I on which they differ, and let J be the collection of all these indices, which is finite. Consider then the finite product over J

$$B' = \prod_{i\in(J,<_I)} B_i.$$

It is simple to check that $B[X]$ is isomorphic to an induced substructure of B', this isomorphism being the restriction of the natural projection $B \to B'$.

We have thus reduced the problem to a finite product, which is a special case of substitution in the sense of Lemma 6.16. □

Lemma 7.26. *Given $(G_i)_{i\in I}$ a family of groups, consider $S = \bigoplus_{i\in I} G_i$ their direct sum, and $P = \prod_{i\in I} G_i$ their direct product. Then S has finite twin-width if and only if all G_i do, and*

$$\mathrm{utww}(S) = \mathrm{utww}(P) = \sup_{i\in I} \mathrm{utww}(G_i).$$

Proof. Choose some well founded ordering on I, a total ordering $<_i$ of each G_i witnessing its twin-width, and order S and P lexicographically accordingly.

Let us first consider the claim on uniform twin-width: assume that $<_i$ witnesses that $\mathrm{utww}(G_i) \leqslant k$. Given a tuple $g = (g_i)_{i\in I} \in P$, the permutation

$x \mapsto xg$ on P is the lexicographic product indexed by $(I, <_I)$ of the permutations $x \mapsto xg_i$ on G_i. By Lemma 7.25, this lexicographic product preserves twin-width. It follows that $\mathrm{utww}(P) \leqslant k$, and a fortiori $\mathrm{utww}(S) \leqslant k$.

For non-uniform twin-width, assume that all G_i have finite twin-width witnessed by $<_i$, and consider $g = (g_i)_{i \in I} \in S$. Now all but finitely many of the permutations $x \mapsto xg_i$ in G_i are the identity with twin-width 0, and the remaining ones each have finite twin-width. Thus we obtain some uniform bound k_g (depending on g) on the twin-width of the permutation $x \mapsto xg_i$, and we conclude using Lemma 7.25 once again. □

A particularly interesting construction using infinite sums and products is the following. Given groups G, H, the (complete) *wreath product* is

$$G \wr H = G^H \rtimes H$$

where G^H stands for the direct product $\prod_{h \in H} G^H$, and H acts by permuting indices by right product, i.e. for $h, h' \in H$ and $x \in G^H$, the action of h is

$$(x \cdot h)(h') = x(h'h).$$

One may also consider the *restricted wreath product*, with a direct sum replacing the direct product, and the same action of H.

$$G \wr_r H = \left(\bigoplus_{h \in H} G \right) \rtimes H.$$

It is a subgroup of the complete wreath product. A simple and well-known example of wreath product is the *lamplighter group* $(\mathbb{Z}/2\mathbb{Z}) \wr_r \mathbb{Z}$, so called because it can be thought of as acting on an infinite line of lamps turned on or off (represented by $(\mathbb{Z}/2\mathbb{Z})^{\mathbb{Z}}$), along which a lamplighter moves (the position being represented by $\mathbb{Z}$) and toggles the lamps.

From Lemma 7.26 and Corollary 7.21, we obtain

Lemma 7.27. $\mathrm{utww}(G \wr H) = \max(\mathrm{utww}(G), \mathrm{utww}(H))$.

Thus for instance, the lamplighter group has uniform twin-width 0.

7.4.4 Group actions. Section 7.3 defined twin-width of arbitrary group actions, but so far we have only used it in for the self-action by product. We will now show that an action of a group G with finite twin-width may be used to show that G itself has finite twin-width.

Lemma 7.28. *Let $(X, <)$ be a well-ordered set, and G a group acting faithfully on X on the right. If this action has finite twin-width with regards to $<$, then G has finite twin-width. Furthermore if this action has uniform twin-width k with regards to $<$, then* $\mathrm{utww}(G) \leqslant k$.

Proof. Consider the set of functions X^X ordered lexicographically: for any $f \neq g \in X^X$, define $m_{f,g} = \min \{x \in X \mid f(x) \neq g(x)\}$ and

$$f \prec g \quad \iff \quad f(m_{f,g}) < g(m_{f,g}).$$

Since G acts faithfully on X, it can be seen as a subset of the permutation group of X, hence as a subset of X^X.

Now given $g \in G$, we consider two permutations:

1. the biorder $B_g^X = (X, <, <_g)$ corresponding to the action of g on X, and
2. the biorder $B_g^{X^X} = (X^X, \prec, \prec_g)$ corresponding to the action of g on X^X by post-composition; which is equivalent to seeing elements of X^X as tuples indexed by X, with g acting coordinate-wise.

Then $B_g^{X^X}$ is the lexicographic product of B_g^X with itself, indexed by $(X, <)$. It follows from Lemma 7.25 that

$$\mathrm{tww}(B_g^X) = \mathrm{tww}(B_g^{X^X}). \tag{7.8}$$

This implies the lemma, as the biorder encoding the action of g on G by post-composition, which is used to define the twin-width of G, is a substructure of $B_g^{X^X}$. □

For example, consider the infinite rooted binary tree T and its group $\mathrm{Aut}(T)$ of automorphisms, i.e. bijections on nodes of T which preserve the root and the parent relation. The group $\mathrm{Aut}(T)$ is uncountable, but it has some interesting finitely generated subgroups, notably the Grigorchuk group [53] whose growth is neither polynomial nor exponential.

Fact 7.29. *The group* $\mathrm{Aut}(T)$ *has uniform twin-width 0.*

Proof. Let $<$ be the breadth-first search order on T, which orders first by increasing depth, then left-to-right at each level. This is a well-order, hence by Lemma 7.28 it suffices to prove that the action of $\mathrm{Aut}(T)$ on T with regards to $<$ has uniform twin-width 0.

It is simple to show that for any automorphism $\sigma \in \mathrm{Aut}(T)$, the permutation induced by σ on each level ℓ of T is separable: it can be constructed by substitution from the permutation 12 and 21 along the tree T itself, restricted to nodes at level ℓ or less (cf. section 6.2). It then suffice to put the permutations on each level one after the other, which still is a separable permutation, hence has twin-width 0 by Fact 6.15. □

7.4.5 Uniform and non-uniform twin-width. Any group with finite uniform twin-width also has finite twin-width. We conjecture that the converse does not hold, and propose a candidate counter-example in this section.

Let $\mathfrak{S}_\mathbb{Z}$ be the group of permutations on $\mathbb{Z}$, and let $\mathfrak{S}_\mathbb{Z}^f$ be the subgroup of finitely supported permutations. Furthermore, call $T = \{x \mapsto x + c \mid c \in \mathbb{Z}\}$ the subgroup of translations. While $\mathfrak{S}_\mathbb{Z}^f$ is not finitely generated, $\mathfrak{S}_\mathbb{Z}^f \rtimes T$ is. The latter is sometimes called *lampshuffler group* [52], by analogy with the lamplighter group.

Now consider the well-order $0, 1, -1, 2, -2, \ldots$ on $\mathbb{Z}$. Swapping of 0 with 1 and translating by 1 are two permutations of $\mathbb{Z}$ which generate the lampshuffler and whose action on $\mathbb{Z}$ (with the previous order) clearly has finite twin-width. Thus, by Corollary 5.7, the action of the lampshuffler on $\mathbb{Z}$ has finite twin-width, and by Lemma 7.28, so does the lampshuffler group itself.

We however conjecture that the lampshuffler has infinite uniform twin-width, and thus separates uniform and non-uniform twin-width. Indeed the lampshuffler group contains every finite group: if it had uniform twin-width k, then *every* finite group would have uniform twin-width at most k, which would be surprising.

This can be pushed further: a group is called *elementary amenable* if it can be constructed from finite groups and abelian groups by taking subgroups, quotients, extensions, and direct limits.

Fact 7.30. *The following are equivalent for any $k \in \mathbb{N}$.*

1. *For all finite group G,* $\operatorname{utww}(G) \leqslant k$.
2. *The lampshuffler group satisfies* $\operatorname{utww}(\mathfrak{S}_{\mathbb{Z}}^{f} \rtimes T) \leqslant k$.
3. *For all elementary amenable group G,* $\operatorname{utww}(G) \leqslant k$.

Proof. The lampshuffler $\mathfrak{S}_{\mathbb{Z}}^{f} \rtimes T$ is elementary amenable and contains all finite groups, hence (3) implies (2) which implies (1). Suppose now that all finite groups have uniform twin-width at most k, and recall that abelian groups have uniform twin-width 0. It is known that all elementary amenable groups can be obtained from finite groups and abelian groups using only extensions and direct limits [29]. It follows by Corollary 7.20 and Lemma 7.15 that all elementary amenable groups have twin-width at most k. □

7.5 Constructing groups with infinite twin-width

The previous sections presented numerous examples of groups with finite twin-width, and stability results for twin-width of groups. This leaves an obvious question: are there groups with *infinite* twin-width? The goal of this section is to construct one.

Theorem 7.1. *There is a finitely generated group with infinite twin-width.*

The situation surrounding Theorem 7.1 closely resembles that of graphs with bounded degree (cf. Corollary 3.39): the construction uses probabilistic arguments, resulting in a group which can hardly be described in an useful way, and unfortunately no explicit construction of groups with infinite twin-width is known.

Nonetheless, the mere existence of a group with infinite twin-width gives a negative answer to the following question which we asked in [14, Conjecture 2.6]. Theorem 3.37 states any class with bounded twin-width is small (has at most $O(1)^n$ graphs on n vertices). Does the converse hold, i.e. does any small hereditary[1] class have bounded twin-width? Indeed, along with asking this question, [14] noticed that any group with infinite twin-width would provide a counter-example: a hereditary class which is small (and even tiny) but has unbounded twin-width.

Lemma 7.31 [14, Lemma 8.3]. *For any Cayley graph G, the class of finite induced subgraphs of G is tiny.*

Proof. Say that $G = \operatorname{Cay}(\Gamma, S)$ for some group Γ and generating set S, and consider H a finite induced subgraph of G. Take a spanning forest F in H (i.e. a spanning tree in each component of H), orient its edges arbitrarily, and label each oriented edge uv of F by the generator $s \in S \cup S^{-1}$ satisfying $u \cdot s = v$. Then F with these labels is enough to reconstruct H. Consider two vertices u, v in H. If they are in distinct components of F, they are not adjacent in H.

[1] If the class is not required to be hereditary, any sequence of graphs with increasing size and twin-width is a trivial counter-example.

Otherwise, we can find the value $u^{-1}v \in \Gamma$ from the path connecting them in F, and there is an edge uv in H if and only if $u^{-1}v \in S \cup S^{-1}$.

Thus for fixed Γ and S, the number of non-isomorphic induced subgraphs on n vertices in $\mathrm{Cay}(\Gamma, S)$ is bounded by the number of trees on n vertices with edge labels in $S \cup S^{-1}$. It is well-known that there are at most 4^n trees on n vertices up to isomorphism, and the edge labels add at most $(2|S|)^n$ choices. □

Corollary 7.32. *There is a tiny hereditary class of graphs with unbounded twin-width.*

Proof. If G is a Cayley graph of some group with infinite twin-width given by Theorem 7.1, then the class of finite induced subgraphs of G is hereditary, and tiny by Lemma 7.31, but has unbounded twin-width. □

Let us mention that Bonnet, Duron, Sylvester, Zamaraev, and Zhukovskii later obtained an entirely different proof of Corollary 7.32 [13], based on a counting argument from Hatami and Hatami [62].

To prove Theorem 7.1, we embeds an appropriate sequence of graphs with bounded degree and unbounded twin-width into a Cayley graph. The graphs are obtained by counting arguments (cf. Theorem 3.37), and the embedding uses the following remarkable result of Osajda.

In a graph G, the *diameter* $\mathrm{diam}(G)$ is the maximum distance between any two vertices, and the *girth* $\mathrm{girth}(G)$ is the minimum length of a cycle.

Theorem 7.33 (Osajda [78, Theorem 3.2]). *Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of connected graphs satisfying, for some constants $A > 0$ and $\Delta \in \mathbb{N}$,*

1. $\Delta(G_n) \leqslant \Delta$,
2. $\mathrm{girth}(G_n) \geqslant \mathrm{girth}(G_{n-1}) + 6$, *and*
3. $\mathrm{diam}(G_n)/\mathrm{girth}(G_n) \leqslant A$.

Then there exists a group Γ finitely generated by S such that every G_n isometrically embeds in $\mathrm{Cay}(\Gamma, S)$.

Osajda goal in proving Theorem 7.33 was to construct groups containing 'complex' graphs, namely expanders. We are thus very much using it for its intended purpose, only complex for us means large twin-width. Its proof proceeds in two steps:

1. First the graphs G_n are equipped with an edge labelling satisfying the *small cancellation condition*: if G_n, G_m contain two paths of length ℓ with identical labels, then $\ell < \frac{1}{6}\min(\mathrm{girth}(G_n), \mathrm{girth}(G_m))$. These labellings are obtained by probabilistic arguments, and constitute the core of Osajda's proof in [78]. See also [42] for a simplified variant of the proof.

2. The group Γ is then obtained using *graphical small cancellation theory*: given the labelled graphs $(G_n)_{n\in\mathbb{N}}$, one can always construct a group Γ by taking the set S of labels as generators, and whenever a cycle in some G_n has edges labelled $s_1, \dots, s_k \in S$, imposing that $s_1 \cdot \dots \cdot s_k = 1$. This is called a *(graphical) group presentation*. The small cancellation condition then ensures that each G_n embeds isometrically into $\mathrm{Cay}(\Gamma, S)$, which is not true in general. This construction, attributed to Gromov [58], is

based on the well established techniques of small cancellation theory (see [72, Chapter V]). See [77, 59] for details.

We now focus on constructing graphs with unbounded twin-width satisfying the hypotheses of Theorem 7.33. Precisely, we construct a sequence of graphs $(G_n)_{n\in\mathbb{N}}$ satisfying the following[2]:

1. Each G_n has maximum degree at most 6,
2. $\operatorname{diam}(G_n) \leqslant 3\log|V(G_n)|$,
3. $\operatorname{girth}(G_n) > \frac{1}{4}\log|V(G_n)|$, and
4. $\operatorname{tww}(G_n)$ is unbounded.

For n even, let $\mathcal{C}_1(n)$ be the class of edge-labeled graphs on vertex set $[n]$ which are formed by the union of three perfect matchings with labels 1, 2, and 3. In $\mathcal{C}_1(n)$, we allow parallel edges (multiple edges with the same endpoints) as long as they have distinct labels. This class is not small, cf. Lemma 3.38.

Lemma 7.34. *For any constant c and any sufficiently large even n,*

$$|\mathcal{C}_1(n)| > n! \cdot c^n.$$

Fix $n \in \mathbb{N}$ even and define $g = \log(n)/4$. Call $\mathcal{C}_2(n)$ the class of graphs on vertex set $[n]$ with degree at most 6, diameter at most $3\log n$ and girth at least g. Let us show that at least half of $\mathcal{C}_1(n)$ can be obtained from graphs of $\mathcal{C}_2(n)$ by editing (adding or removing) at most $n^{7/8}$ edges and adding labels $1, 2, 3$ to edges.

Lemma 7.35. *If G is a graph chosen uniformly at random in $\mathcal{C}_1(n)$, then the expected number of cycles in G of length at most g is at most $2 \cdot 6^g$.*

Proof. Let C be a *potential* cycle of length $\ell \leqslant g$ properly edge-colored $1, 2, 3$, that is, an arbitrary non-repeating sequence $v_1, \dots, v_\ell$ of vertices in $[n]$ where each pair v_iv_{i+1} (modulo ℓ) is seen as a *potential edge* and given a label in $1, 2, 3$, so that consecutive edges have distinct labels. Let us bound the probability that C appears as a cycle with the assigned colors in G.

When choosing a random perfect matching M on n vertices (n even), the probability that a given pair of vertices belongs to M is $\frac{n}{2}/\binom{n}{2} = \frac{1}{n-1}$. If these probabilities were independent, C would appear in G with probability exactly $(n-1)^{-\ell}$. But the fact that each color class of G forms a matching introduces a small dependency: Conditioned by the fact that t edges of a random perfect matching M are already known, the probability that a given pair of not yet matched vertices appears in M is $1/(n-2t-1)$. Now if these already known edges are part of C, then $t \leqslant \ell \leqslant g \ll n$, hence this probability is upper bounded by $2/n$. Thus the probability that C appears in G is at most $(2/n)^\ell$.

Finally, the number of properly edge colored potential cycles of length ℓ is at most $(3n)^\ell$, hence the expected total number of cycles of length at most g in G is

$$\sum_{\ell=2}^{g} (3n)^\ell \cdot (2/n)^\ell \leqslant 2 \cdot 6^g. \qquad \square$$

[2]Logarithms are in base 2.

Lemma 7.36. *For any even n, for at least half of the graphs $G = (V, E)$ in $\mathcal{C}_1(n)$, there exists $G' = (V, E')$ in $\mathcal{C}_2(n)$ such that the symmetric difference of edge sets is $|E \triangle E'| = O(n^{7/8})$.*

Proof. By Lemma 7.35 and Markov's inequality, at least half of the graphs G_1 in $\mathcal{C}_1(n)$ have at most $4 \cdot 6^g$ cycles of length at most g. Let G_1 be one such graph, and choose $F \subset E(G_1)$ of size at most $4 \cdot 6^g$ intersecting every cycle of length at most g in G_1 (simply choose an edge in each such cycle). Let G_2 be the subgraph of G_1 obtained by deleting the edges of F. By construction, G_2 has girth more than g. We will now add edges to G_2 to ensure its diameter is at most $3 \log n$ without creating new short cycles.

Let Z be the set of endpoints of edges in F. Thus $|Z| \leqslant 8 \cdot 6^g$. Let N be the union of balls of radius $g/2$ centered on vertices of Z. Since G_2 has maximum degree 3, for any fixed vertex x, the ball of radius d around x in G_2 contains at most $3 \cdot 2^d$ vertices. It follows that

$$|N| \leqslant 8 \cdot 6^g \cdot 3 \cdot 2^{\frac{g}{2}} \tag{7.9}$$

$$= 24 \cdot (6\sqrt{2})^g \tag{7.10}$$

$$= 24 \cdot n^{\log(6\sqrt{2})/4}. \qquad \text{(recall } g = \log n/4\text{)} \tag{7.11}$$

Observe that $\log(6\sqrt{2})/4 \leqslant 7/8$, hence

$$|N| \leqslant 24 \cdot n^{7/8}. \tag{7.12}$$

Choose X an inclusion-wise maximal subset of $V(G_2)$ such that vertices in X pairwise have distance at least $\log n$ in G_2. Define $X_1 = X \cap N$, and $X_2 = X \setminus N$.

Let $v \in X_2$, and let B_v be the closed ball of radius $g/2$ in G_2. Since v is at distance more than $g/2$ from Z, no edge of G_1 was removed inside this ball, hence all vertices of B_v have degree 3. Furthermore, since G_2 has girth more than g, G_2 restricted to B_v is a tree, except possibly for edges between vertices at distance $g/2$ from v which we may ignore. This implies

$$|B_v| \geqslant 2^{g/2} = n^{1/8}. \tag{7.13}$$

Finally, any distinct $v, v' \in X_2$ are at distance at least $\log n$, which is more than g. Thus the balls B_v and $B_{v'}$ are disjoint. It follows from (7.13) that

$$|X_2| \leqslant \frac{|V(G_2)|}{n^{1/8}} = n^{7/8}. \tag{7.14}$$

Using that $X = X_1 \cup X_2$, $X_1 \subset N$, and (7.12) and (7.14), we obtain

$$|X| = O(n^{7/8}). \tag{7.15}$$

Construct G_3 by adding to G_2 a balanced ternary tree T with vertex set X. Thus, vertices in X are pairwise at distance at most $\log n$ in G_3. Furthermore, by maximality of X, all vertices of G_2 are at distance at most $\log n$ of X. It follows that the diameter of G_3 is at most $3 \log n$. Finally, G_3 still has girth at least g, because T only connects vertices of X, which are far apart in G_2. Clearly the degree of vertices in G_3 does not exceed 6. Thus $G_3 \in \mathcal{C}_2(n)$.

To summarize, we transform G_1 into G_3 by deleting $|F| \leqslant 4 \cdot 6^g$ edges, and then adding $|X|$ edges, which gives $O(n^{7/8})$ edge editions in total. □

Lemma 7.37. *The class $\mathcal{C}_2$ is not small, that is for any constant c and any sufficiently large even n,*

$$|\mathcal{C}_2(n)| \geqslant n! \cdot c^n.$$

Proof. Fix c, consider n even sufficiently large, and suppose for a contradiction that $|\mathcal{C}_2(n)| \leqslant n! \cdot c^n$. For any graph G on n vertices, the number of graphs which can be obtained from G by editing up to $n^{7/8}$ edges is at most

$$\binom{n^2}{n^{7/8}} \leqslant n^{2n^{7/8}} \leqslant 2^n, \text{ asymptotically.} \tag{7.16}$$

Hence the total number of cubic graphs with edges labeled 1,2,3 obtained by $n^{7/8}$ editions from any graph in $\mathcal{C}_2(n)$ is asymptotically at most

$$2^n \cdot 3^{3n/2} \cdot n! \cdot c^n \leqslant n! \cdot c_2^n \tag{7.17}$$

for some constant c_2. By Lemma 7.36, the cardinality of $\mathcal{C}_1(n)$ is at most double the previous value, a contradiction of Lemma 7.34. □

Theorem 7.1. *There is a finitely generated group with infinite twin-width.*

Proof. By Lemma 7.37 and Theorem 3.37, for any $k \in \mathbb{N}$ and for any sufficiently large n_k, $\mathcal{C}_2(n_k)$ contains graphs of twin-width more than k. Thus, one can construct a sequence $(G_k)_{k\in\mathbb{N}}$ of graphs such that $\mathrm{tww}(G_k) > k$ and $G_k \in \mathcal{C}_2(n_k)$, for any sequence $(n_k)_{k\in\mathbb{N}}$ of even integers growing sufficiently fast. Since $\mathrm{girth}(G_k) \geqslant \log(n_k)/4$, an appropriate choice of n_k ensures that $\mathrm{girth}(G_k) \geqslant \mathrm{girth}(G_{k-1}) + 6$. Then $(G_k)_{k\in\mathbb{N}}$ satisfies the hypotheses of Theorem 7.33, hence there exists a group Γ finitely generated by S such that every G_k isometrically embeds in $\mathrm{Cay}(\Gamma, S)$. This group has infinite twin-width. □

BIBLIOGRAPHY

[1] Hans Adler and Isolde Adler. "Interpreting nowhere dense graph classes as a classical notion of model theory". In: *European Journal of Combinatorics* 36 (2014), pp. 322–330. DOI: https://doi.org/10.1016/j.ejc.2013.06.048.

[2] Jungho Ahn, Debsoumya Chakraborti, Kevin Hendrey and Sang-il Oum. *Twin-width of subdivisions of multigraphs*. 2023. DOI: 10.48550/arXiv.2306.05334. arXiv: 2306.05334.

[3] Nir Ailon, Moses Charikar and Alantha Newman. "Aggregating Inconsistent Information: Ranking and Clustering". In: *J. ACM* 55.5 (Nov. 2008). DOI: 10.1145/1411509.1411513.

[4] Michael Albert, Mathilde Bouvel and Valentin Féray. "Two first-order logics of permutations". In: *Journal of Combinatorial Theory, Series A* 171 (2020), p. 105158. DOI: https://doi.org/10.1016/j.jcta.2019.105158.

[5] V.E. Alekseev. "Polynomial algorithm for finding the largest independent sets in graphs without forks". In: *Discrete Applied Mathematics* 135.1 (2004). Russian Translations II, pp. 3–16. DOI: https://doi.org/10.1016/S0166-218X(02)00290-1.

[6] Stephen Alstrup, Søren Dahlgaard and Mathias Bæk Tejs Knudsen. "Optimal Induced Universal Graphs and Adjacency Labeling for Trees". In: *J. ACM* 64.4 (2017), 27:1–27:22. DOI: 10.1145/3088513.

[7] Brenda S. Baker. "Approximation algorithms for NP-complete problems on planar graphs". In: *J. ACM* 41.1 (Jan. 1994), pp. 153–180. DOI: 10.1145/174644.174650.

[8] Jakub Balabán and Petr Hlinený. "Twin-Width Is Linear in the Poset Width". In: *16th International Symposium on Parameterized and Exact Computation, IPEC 2021*. Vol. 214. LIPIcs. Schloss Dagstuhl, 2021, 6:1–6:13. DOI: 10.4230/LIPIcs.IPEC.2021.6.

[9] Ambroise Baril, Miguel Couceiro and Victor Lagerkvist. "Linear Bounds between Component Twin-Width and Clique-Width with Algorithmic Applications to Counting Graph Colorings". working paper or preprint. Dec. 2023. URL: https://hal.science/hal-04142719.

[10] Pierre Bergé, Édouard Bonnet and Hugues Déprés. "Deciding Twin-Width at Most 4 Is NP-Complete". In: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Vol. 229. LIPIcs. Schloss Dagstuhl, 2022, 18:1–18:20. DOI: 10.4230/LIPIcs.ICALP.2022.18.

[11] Édouard Bonnet, Romain Bourneuf, Julien Duron, Colin Geniet, Stéphan Thomassé and Nicolas Trotignon. *A tamed family of triangle-free graphs with unbounded chromatic number*. 2023. arXiv: 2304.04296.

[12] Édouard Bonnet, Romain Bourneuf, Colin Geniet and Stéphan Thomassé. "Factoring Pattern-Free Permutations into Separable ones". In: *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2024, pp. 752–779. DOI: 10.1137/1.9781611977912.30. arXiv: 2308.02981.

[13] Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev and Maksim Zhukovskii. *Small But Unwieldy: A Lower Bound on Adjacency Labels for Small Classes*. 2023. arXiv: 2307.11225.

[14] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé and Rémi Watrigant. "Twin-width II: small classes". In: *Combinatorial Theory* 2.2 (2022). DOI: 10.5070/C62257876.

[15] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé and Rémi Watrigant. "Twin-width III: Max Independent Set, Min Dominating Set, and Coloring". In: *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Vol. 198. LIPIcs. Schloss Dagstuhl, 2021, 35:1–35:20. DOI: 10.4230/LIPIcs.ICALP.2021.35. arXiv: 2007.14161.

[16] Édouard Bonnet, Colin Geniet, Romain Tessera and Stéphan Thomassé. *Twin-width VII: groups*. 2022. DOI: 10.48550/ARXIV.2204.12330. arXiv: 2204.12330.

[17] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé and Szymon Toruńczyk. "Twin-Width IV: Ordered Graphs and Matrices". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. Association for Computing Machinery, 2022, pp. 924–937. DOI: 10.1145/3519935.3520037. arXiv: 2102.03117.

[18] Édouard Bonnet, Eun Jung Kim, Amadeus Reinald and Stéphan Thomassé. "Twin-width VI: the lens of contraction sequences". In: *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2022, pp. 1036–1056. DOI: 10.1137/1.9781611977073.45.

[19] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé and Rémi Watrigant. "Twin-width I: Tractable FO Model Checking". In: *J. ACM* 69.1 (2022), 3:1–3:46. DOI: 10.1145/3486655. arXiv: 2004.14789.

[20] Édouard Bonnet, O-joung Kwon and David R. Wood. *Reduced bandwidth: a qualitative strengthening of twin-width in minor-closed classes (and beyond)*. 2022. DOI: 10.48550/arXiv.2202.11858. arXiv: 2202.11858.

[21] Édouard Bonnet, Jaroslav Nešetřil, Patrice Ossona de Mendez, Sebastian Siebertz and Stéphan Thomassé. *Twin-width and permutations*. 2021. arXiv: 2102.06880. URL: https://arxiv.org/abs/2102.06880.

[22] Prosenjit Bose, Jonathan F. Buss and Anna Lubiw. "Pattern Matching for Permutations". In: *Inf. Process. Lett.* 65.5 (1998), pp. 277–283. DOI: 10.1016/S0020-0190(97)00209-3.

[23] Romain Bourneuf and Stéphan Thomassé. *Bounded twin-width graphs are polynomially χ-bounded*. 2023. arXiv: 2303.11231.

[24] Andreas Brandstädt and Raffaele Mosca. "Maximum weight independent set for ℓclaw-free graphs in polynomial time". In: *Discrete Applied Mathematics* 237 (2018), pp. 57–64. DOI: https://doi.org/10.1016/j.dam.2017.11.029.

[25] Samuel Braunfeld, Anuj Dawar, Ioannis Eleftheriadis and Aris Papadopoulos. "Monadic NIP in monotone classes of relational structures". In: *European Conference on Combinatorics, Graph Theory and Applications*. 12. 2023. DOI: 10.5817/CZ.MUNI.EUROCOMB23-029.

[26] Samuel Braunfeld and Michael C Laskowski. *Existential characterizations of monadic NIP*. 2022. DOI: 10.48550/ARXIV.2209.05120. arXiv: 2209.05120.

[27] Sergei Buyalo, Alexander Dranishnikov and Viktor Schroeder. "Embedding of hyperbolic groups into products of binary trees". In: *Inventiones mathematicae* 169.1 (2007), pp. 153–192.

[28] Arthur Cayley. "A theorem on trees". In: *Quart. J. Math.* 23 (1889), pp. 376–378.

[29] Ching Chou. "Elementary amenable groups". In: *Illinois Journal of Mathematics* 24.3 (1980), pp. 396–407.

[30] Maria Chudnovsky, Irena Penev, Alex Scott and Nicolas Trotignon. "Substitution and χ-boundedness". In: *Journal of Combinatorial Theory, Series B* 103.5 (2013), pp. 567–586. DOI: https://doi.org/10.1016/j.jctb.2013.02.004.

[31] Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk and Stéphan Thomassé. "Quasi-polynomial time approximation schemes for the Maximum Weight Independent Set Problem in H-free graphs". In: *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2260–2278. DOI: 10.1137/1.9781611975994.139. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611975994.139.

[32] Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic: a language-theoretic approach*. Vol. 138. Cambridge University Press, 2012.

[33] Bruno Courcelle, Joost Engelfriet and Grzegorz Rozenberg. "Handle-rewriting hypergraph grammars". In: *Journal of Computer and System Sciences* 46.2 (1993), pp. 218–270. DOI: https://doi.org/10.1016/0022-0000(93)90004-G.

[34] Bruno Courcelle, Johann A. Makowsky and Udi Rotics. "Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width". In: *Theory Comput. Syst.* 33.2 (2000), pp. 125–150. DOI: 10.1007/s002249910009.

[35] James Davies. "Improved bounds for colouring circle graphs". In: *Proceedings of the American Mathematical Society* 150.12 (2022), pp. 5121–5135.

[36] James Davies and Rose McCarty. "Circle graphs are quadratically χ-bounded". In: *Bulletin of the London Mathematical Society* 53.3 (2021), pp. 673–679. DOI: https://doi.org/10.1112/blms.12447. eprint: https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/blms.12447.

[37] Blanche Descartes. "Solution to advanced problem no. 4526". In: *Amer. Math. Monthly* 61.352 (1954), p. 216.

[38] RG Downey and MR Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer, 1999. DOI: 10.1007/978-1-4612-0515-9.

[39] Rodney G Downey, Michael R Fellows and Udayan Taylor. "The parameterized complexity of relational database queries and an improved characterization of W[1]". In: *DMTCS* 96 (1996), pp. 194–213.

[40] Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek and Pat Morin. "Adjacency Labelling for Planar Graphs (and Beyond)". In: *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. IEEE, 2020, pp. 577–588. DOI: 10.1109/FOCS46700.2020.00060.

[41] Zdeněk Dvořák, Daniel Král and Robin Thomas. "Testing first-order properties for subclasses of sparse graphs". In: *J. ACM* 60.5 (2013), 36:1–36:24. DOI: 10.1145/2499483.

[42] Louis Esperet and Ugo Giocanti. "Optimization in graphical small cancellation theory". In: *Discrete Mathematics* 347.4 (2024), p. 113842. DOI: https://doi.org/10.1016/j.disc.2023.113842.

[43] Jörg Flum and Martin Grohe. "Fixed-Parameter Tractability, Definability, and Model-Checking". In: *SIAM J. Comput.* 31.1 (2001), pp. 113–145. DOI: 10.1137/S0097539799360768.

[44] Jacob Fox. *Stanley–Wilf limits are typically exponential*. 2013. arXiv: 1310.8378.

[45] Markus Frick and Martin Grohe. "Deciding first-order properties of locally tree-decomposable structures". In: *J. ACM* 48.6 (2001), pp. 1184–1206. DOI: 10.1145/504794.504798.

[46] Zoltán Füredi and Péter Hajnal. "Davenport-Schinzel theory of matrices". In: *Discrete Mathematics* 103.3 (1992), pp. 233–251. DOI: https://doi.org/10.1016/0012-365X(92)90316-8.

[47] Jakub Gajarský, Petr Hlinený, Daniel Lokshtanov, Jan Obdrzálek, Sebastian Ordyniak, M. S. Ramanujan and Saket Saurabh. "FO Model Checking on Posets of Bounded Width". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*. 2015, pp. 963–974. DOI: 10.1109/FOCS.2015.63.

[48] Jakub Gajarský, Petr Hliněný, Jan Obdržálek, Daniel Lokshtanov and M. S. Ramanujan. "A New Perspective on FO Model Checking of Dense Graph Classes". In: *ACM Trans. Comput. Logic* 21.4 (2020). DOI: 10.1145/3383206.

[49] Jakub Gajarský, Michał Pilipczuk, Wojciech Przybyszewski and Szymon Toruńczyk. "Twin-Width and Types". In: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Vol. 229. LIPIcs. Schloss Dagstuhl, 2022, 123:1–123:21. DOI: 10.4230/LIPIcs.ICALP.2022.123.

[50] M. R. Garey, D. S. Johnson and L. Stockmeyer. "Some simplified NP-complete problems". In: *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*. STOC '74. Association for Computing Machinery, 1974, pp. 47–63. DOI: 10.1145/800119.803884.

[51] Colin Geniet and Stéphan Thomassé. "First Order Logic and Twin-Width in Tournaments". In: *31st Annual European Symposium on Algorithms (ESA 2023)*. Vol. 274. LIPIcs. Schloss Dagstuhl, 2023, 53:1–53:14. DOI: 10.4230/LIPIcs.ESA.2023.53. arXiv: 2207.07683.

[52] Maxime Gheysens and Nicolas Monod. *Between free and direct products of groups*. 2022. DOI: 10.48550/arXiv.2201.03625. arXiv: 2201.03625.

[53] Rostislav Ivanovich Grigorchuk. "Burnside problem on periodic groups". In: *Funktsional'nyi Analiz i ego Prilozheniya* 14.1 (1980), pp. 53–54.

[54] Martin Grohe, Stephan Kreutzer and Sebastian Siebertz. "Deciding First-Order Properties of Nowhere Dense Graphs". In: *J. ACM* 64.3 (2017), 17:1–17:32. DOI: 10.1145/3051095.

[55] Martin Grohe and Daniel Neuen. *Isomorphism for Tournaments of Small Twin Width*. 2023. DOI: 10.48550/arXiv.2312.02048. arXiv: 2312.02048.

[56] Mikhael Gromov. "Groups of polynomial growth and expanding maps (with an appendix by Jacques Tits)". en. In: *Publications Mathématiques de l'IHÉS* 53 (1981), pp. 53–78. URL: http://www.numdam.org/item/PMIHES_1981__53__53_0/.

[57] Mikhael Gromov. "Hyperbolic groups". In: *Essays in group theory*. Springer, 1987, pp. 75–263.

[58] Mikhail Gromov. "Random walk in random groups". In: *Geometric & Functional Analysis GAFA* 13.1 (2003), pp. 73–146. DOI: 10.1007/s000390300002.

[59] Dominik Gruber. "Groups with graphical $\mathcal{C}(6)$ and $\mathcal{C}(7)$ small cancellation presentations". In: *Transactions of the American Mathematical Society* 367.3 (2015), pp. 2051–2078. DOI: 10.1090/S0002-9947-2014-06198-9. arXiv: 1210.0178.

[60] Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk and Michał Pilipczuk. "Polynomial-time Algorithm for Maximum Weight Independent Set on P_6-free Graphs". In: *ACM Trans. Algorithms* 18.1 (Jan. 2022). DOI: 10.1145/3414473.

[61] Sylvain Guillemot and Dániel Marx. "Finding small patterns in permutations in linear time". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*. 2014, pp. 82–101. DOI: 10.1137/1.9781611973402.7.

[62] Hamed Hatami and Pooya Hatami. "The Implicit Graph Conjecture is False". In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. 2022, pp. 1134–1137. DOI: 10.1109/FOCS54457.2022.00109.

[63] Kevin Hendrey, Sergey Norin, Raphael Steiner and Jérémie Turcotte. *Twin-width of sparse random graphs*. 2023. DOI: 10.48550/arXiv.2312.03688. arXiv: 2312.03688.

[64] Petr Hliněný and Jan Jedelský. "Twin-Width of Planar Graphs Is at Most 8, and at Most 6 When Bipartite Planar". In: *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*. Vol. 261. LIPIcs. Schloss Dagstuhl, 2023, 75:1–75:18. DOI: 10.4230/LIPIcs.ICALP.2023.75.

[65] Hugo Jacob and Marcin Pilipczuk. "Bounding Twin-Width for Bounded-Treewidth Graphs, Planar Graphs, and Bipartite Graphs". In: *Graph-Theoretic Concepts in Computer Science*. Springer International Publishing, 2022, pp. 287–299. DOI: 10.1007/978-3-031-15914-5_21. arXiv: 2201.09749.

[66] Richard M. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations*. Springer US, 1972, pp. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.

[67] Martin Klazar. "The Füredi-Hajnal Conjecture Implies the Stanley-Wilf Conjecture". In: *Formal Power Series and Algebraic Combinatorics*. Springer Berlin Heidelberg, 2000, pp. 250–255. DOI: 10.1007/978-3-662-04166-6_22.

[68] Daniel Král' and Ander Lamaison. *Planar graph with twin-width seven*. 2023. DOI: https://doi.org/10.1016/j.ejc.2023.103749. arXiv: 2209.11537.

[69] Stephan Kreutzer and Anuj Dawar. "Parameterized Complexity of First-Order Logic". In: *Electronic Colloquium on Computational Complexity (ECCC)* 16 (2009), p. 131. URL: http://eccc.hpi-web.de/report/2009/131.

[70] Dietrich Kuske and Markus Lohrey. "Logical aspects of Cayley-graphs: the group case". In: *Annals of Pure and Applied Logic* 131.1-3 (2005), pp. 263–286.

[71] Daniel Lokshtanov, Martin Vatshelle and Yngve Villanger. "Independent set in P_5-free graphs in polynomial time". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '14. Society for Industrial and Applied Mathematics, 2014, pp. 570–581. ISBN: 9781611973389.

[72] Roger C Lyndon and Paul E Schupp. *Combinatorial group theory*. Vol. 188. Springer, 1977.

[73] Adam Marcus and Gábor Tardos. "Excluded permutation matrices and the Stanley-Wilf conjecture". In: *J. Comb. Theory, Ser. A* 107.1 (2004), pp. 153–160. DOI: 10.1016/j.jcta.2004.04.002.

[74] Jan Mycielski. "Sur le coloriage des graphs". in French. In: *Colloquium Mathematicae*. Vol. 3. 2. 1955, pp. 161–162. DOI: 10.4064/cm-3-2-161-162.

[75] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*. Vol. 28. Algorithms and combinatorics. Springer, 2012. DOI: 10.1007/978-3-642-27875-4.

[76] Serguei Norine, Paul D. Seymour, Robin Thomas and Paul Wollan. "Proper minor-closed families are small". In: *J. Comb. Theory, Ser. B* 96.5 (2006), pp. 754–757. DOI: 10.1016/j.jctb.2006.01.006.

[77] Yann Ollivier. "On a small cancellation theorem of Gromov". In: *Bulletin of the Belgian Mathematical Society–Simon Stevin* 13.1 (2006), pp. 75–89. DOI: 10.36045/bbms/1148059334. arXiv: math/0310022.

[78] Damian Osajda. "Small cancellation labellings of some infinite graphs and applications". In: *Acta Mathematica* 225.1 (2020), pp. 159–191. DOI: 10.4310/ACTA.2020.v225.n1.a3.

[79] Michał Pilipczuk and Marek Sokołowski. "Graphs of bounded twin-width are quasi-polynomially χ-bounded". In: *Journal of Combinatorial Theory, Series B* 161 (2023), pp. 382–406. DOI: https://doi.org/10.1016/j.jctb.2023.02.006.

[80] Michał Pilipczuk, Marek Sokołowski and Anna Zych-Pawlewicz. "Compact Representation for Matrices of Bounded Twin-Width". In: *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*. Vol. 219. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 52:1–52:14. DOI: 10.4230/LIPIcs.STACS.2022.52.

[81] Klaus-Peter Podewski and Martin Ziegler. "Stable graphs". In: *Fundamenta Mathematicea* 100.2 (1978), pp. 101–107. DOI: 10.4064/fm-100-2-101-107.

[82] Neil Robertson and P.D Seymour. "Graph minors. III. Planar tree-width". In: *Journal of Combinatorial Theory, Series B* 36.1 (1984), pp. 49–64. DOI: https://doi.org/10.1016/0095-8956(84)90013-3.

[83] Neil Robertson and P.D Seymour. "Graph minors. V. Excluding a planar graph". In: *Journal of Combinatorial Theory, Series B* 41.1 (1986), pp. 92–114. DOI: https://doi.org/10.1016/0095-8956(86)90030-4.

[84] Neil Robertson and P.D Seymour. "Graph minors. XVI. Excluding a non-planar graph". In: *Journal of Combinatorial Theory, Series B* 89.1 (2003), pp. 43–76. DOI: https://doi.org/10.1016/S0095-8956(03)00042-X.

[85] Detlef Seese. "Linear time computable problems and first-order descriptions". In: *Mathematical Structures in Computer Science* 6.6 (1996), pp. 505–526. DOI: 10.1017/S0960129500070079.

[86] Zoran Šunić. "Explicit left orders on free groups extending the lexicographic order on free monoids". In: *Comptes Rendus Mathematique* 351.13 (2013), pp. 507–511. DOI: https://doi.org/10.1016/j.crma.2013.07.001.

[87] Luca Trevisan. *Inapproximability of Combinatorial Optimization Problems*. 2004. arXiv: cs/0409043.

[88] Alexander A. Zykov. "On some properties of linear complexes". in Russian. In: *Mat. Sb. (N.S.)* 24.66 (2 1949), pp. 163–188.

GLOSSARY

Symbols

$[n]$ the interval of integers $\{1, \ldots, n\}$,
$\Delta(G)$ maximum degree of vertices in G
$G[X]$ *see* induced subgraph
K_n *see* clique
$K_{s,t}$ *see* biclique
$\chi(G)$ *see* chromatic number
c_k *see* Marcus–Tardos constant
$\omega(G)$ *see* clique number

B

biclique ($K_{s,t}$) The graph with two sets X, Y of r and s vertices respectively, with no edge inside X nor Y, and all edges between them. p. 4
binary Said of a relational signature or structure whose relations are all of arity 2. p. 66
biorder Relational structure $(V, <_1, <_2)$ consisting of two total orderings of the same vertex set V. Represents a permutation. p. 70, 89
block (matrix) *see* division

C

Cayley graph For a group Γ and a finite generating subset $S \subset \Gamma$, the graph $\mathrm{Cay}(\Gamma, S)$ with vertices Γ, and edges xy whenever $y = xs$ for some $x \in \Gamma$ and $s \in S$. p. 121
cell (matrix) *see* division
χ-bounded A hereditary class $\mathcal{C}$ of graphs satisfying $\chi(G) \leqslant f(\omega(G))$ for some function f and all $G \in \mathcal{C}$. p. 22
chromatic number ($\chi(G)$) Minimum number k of colours needed to properly colour G, i.e. such that there is a map $\lambda : V(G) \to [k]$ satisfying $\lambda(x) \neq \lambda(y)$ whenever $xy \in E(G)$. p. 22
clique (K_n) The graph with n vertices and all edges between them. p. 4
clique number ($\omega(G)$) Maximum size of a clique contained in G. p. 22
coarsening *see* refinement
cograph Graph constructed by disjoint and complete unions starting from single vertices. Equivalently, graph without P_4 as induced subgraph. p. 18
compatible (order)
- **with a contraction sequence** $\mathcal{P}_n, \ldots, \mathcal{P}_1$: a linear ordering $<$ such that all parts of all $\mathcal{P}_i$ are intervals on $<$. p. 44
- **with a rooted tree** T: a linear ordering $<$ of the leaves of T obtained as left-to-right ordering for some plane embedding of T. An ordering $<$

is compatible with T if and only if for any node $t \in T$, the set of leaves descendant of t is an interval of $<$. p. 44, 97

contraction sequence Sequence of partitions of the vertices of a graph, starting with the partition into singleton, merging two parts at each step, and ending with a single part. Also, the sequence of quotient trigraphs defined by these partitions. p. 12, 14, *see also* trigraph

ε-versatile Process for creating a contraction sequence (usually with a fixed bound on the width) in which, when considering a partition $\mathcal{P}$, one can arbitrarily forbid use of $\varepsilon|\mathcal{P}|$ of the parts, and still find a contraction to continue the sequence from $\mathcal{P}$. p. 47

error rank Maximum error rank over all parts at all steps of the contraction sequence. p. 39

width Maximum error degree over all vertices in all quotient trigraphs the contraction sequence. p. 12

D

degenerate G is *k-degenerate* if in G and all of its subgraphs, there is a vertex of degree at most k. Equivalently, there is an ordering $<$ of $V(G)$ such that each $x \in V(G)$ has at most k neighbours placed before x in $<$. p. 26, 105

division Pair of partitions $(\mathcal{R}, \mathcal{C})$ of the rows and columns respectively of a matrix into intervals. p. 33

block (of rows, resp. columns) One of the interval $R \in \mathcal{R}$ (resp. $C \in \mathcal{C}$) of the partitions defining the division. p. 33

cell The submatrix defined by the intersection of blocks $R \in \mathcal{R}, C \in \mathcal{C}$. p. 33

d-division Division with d blocks of rows and d blocks of columns. p. 33

rank-k d-division d-division of a matrix in which every cell has rank at least k. p. 39

rank-k division Short for rank-k k-division. p. 39

E

error Two sets of vertices X, Y between which there is both an edge and a non-edge. When X, Y are not in error (there are either all possible edges between them, or none of them), they are *homogeneous*. p. 11

for binary relations Subsets of vertices $X, Y \subset V$ are *in error* w.r.t. a binary relation $R \subset V^2$ if there are pairs $(x_1, y_1), (x_2, y_2) \in X \times Y$ such that $(x_1, y_1) \in R$ and $(x_2, y_2) \notin R$, or vice versa up to swapping X and Y. Otherwise, they are *homogeneous*: either all or none of the pairs $(x, y) \in X \times Y$ are in R, and similarly either or all of the pairs in $Y \times X$ are in R. p. 66

error degree For a vertex x in a trigraph, the number of vertices connected to x by an error edge. For a part X in a partition $\mathcal{P}$ of the vertices of a graph G, the error degree of X in the quotient trigraph $Tri(G, \mathcal{P})$, i.e. the number of parts of $\mathcal{P}$ in error with X. p. 12, *see also* trigraph

error graph For a graph G and a partition $\mathcal{P}$ of $V(G)$, the graph with vertices $\mathcal{P}$ in which parts are adjacent when they are in error. p. 29

error rank ($\mathrm{erk}_{\mathcal{P}}(P)$) For a part $P \in \mathcal{P}$ of a partition of the vertices of a graph $G = (V, E)$, the smallest k satisfying: there is a subset $\mathcal{Q} \subset \mathcal{P}$ of k such that $\mathrm{rk}(P; V \setminus (\cup\mathcal{Q} \cup P)) \leqslant k$. p. 39

F

first-order logic

- **formula** (over graphs) Logical formula constructed from quantification on vertices, boolean combinators, and adjacency and equality tests on vertices. p. 57
- **closed formula** Formula without free variables. Syn. *sentence*. p. 61
- **independent** A class $\mathcal{C}$ of relational structures which admits a first-order transduction Φ such that $\Phi(\mathcal{C})$ contains all graphs. A class which is *not* independent is called *dependent* or *NIP* (for Non Independence Property). p. 59, 64
- **interpretation** A map Φ of relational structures specified using first-order logic: for each relational symbol E of the image, a formula ϕ_E defines $E(\Phi(S))$ by

$$(x_1, \ldots, x_k) \in E(\Phi(S)) \quad \Longleftrightarrow \quad S \models \phi_E(x_1, \ldots, x_k).$$

  An additional formula ϕ_V may be used to restrict the domain as $x \in V(\Phi(S))$ if and only if $S \models \phi_V(x)$. p. 59, 62
- **model checking problem** Algorithmic problem of testing if a given graph G (or relational structure) satisfies a given formula ϕ. p. 58
- **transduction** Generalisation of interpretations with a *non-deterministic colouring* step: The input structure S is augmented by a fixed number of unary predicates $U_1, \ldots, U_r$, with $U_i(S)$ chosen arbitrarily and non-deterministically, before applying an interpretation. p. 63

Fixed parameter tractable (FPT) Algorithm with running time $f(k) \cdot n^c$, where f is any computable function, c is constant, n is the input size, and k is a *parameter* depending on the algorithmic problem. Usually, the parameter k is either the size of the desired solution, or some complexity measure (e.g. treewidth). p. 2

G

grid Division of a 0–1 matrix in which every cell contains a '1'. p. 33

- **k-grid** Grid consisting of k blocks of rows and k blocks of columns. p. 33

grid rank Maximum k for which the matrix admits a rank-k division. p. 39

H

hereditary A class of graphs (or relational structures) closed under induced subgraphs. p. 4

homogeneous *see* error

I

independent set Set of pairwise non-adjacent vertices. Syn. *stable* set. p. 1

problem Algorithmic problem of finding an independent set in a graph, either of maximum size (maximum independent set problem, MIS) or of a prescribed size k (k-independent set problem). p. 1

induced subgraph ($G[X]$) In G, the graph obtained by keeping only the subset of vertices $X \subset V(G)$, and all edges of G with endpoints in X. p. 4

L

laminar family Family $\mathcal{F}$ of non-empty sets, any two of which are either disjoint or contained in one another. Implicitly describes a forest with nodes $\mathcal{F}$, where X is an ancestor of Y if and only if $Y \subset X$. p. 17

M

Marcus–Tardos constant (c_k) Smallest integer such that any $n \times n$ matrix with at least $c_k \cdot n$ entries '1's contains a k-grid. The Marcus–Tardos theorem proves its existence [73], Fox proved $c_k \leqslant 3k2^{8k}$ [44]. p. 36

O

ordered graph Ordered structure $(V, <, E)$ where (V, E) is a graph and $<$ is a total ordering of V. p. 70

ordered structure Relational structure S containing a relation $<_S$ which is a total ordering of the vertices. p. 70

P

pattern (permutation) $\tau \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if there are increasing maps $f, g : [k] \to [n]$ satisfying $f \circ \tau = \sigma \circ g$. When represented as biorders, a pattern corresponds to an induced substructure. When represented with permutations matrices, a pattern corresponds to a submatrix. p. 94

permutation matrix (M_σ) For $\sigma \in \mathfrak{S}_n$, the $n \times n$ matrix with a '1' at the intersection of column i and row $\sigma(i)$ for any $i \in [n]$. p. 82, 90

power graph (G^k) The graph with the same vertices as G, and an edge xy whenever x and y are at distance at most k in G. p. 122

Q

quasi-isometric embedding A map $f : X \to Y$ between metric spaces which preserves distances up to some affine upper and lower bounds. p. 123

quasi-isometry A quasi-isometric embedding $f : X \to Y$ which furthermore is *cobounded*: all points of Y are at bounded distance of $f(X)$. p. 123

quotient graph ($G/\mathcal{P}$) For a graph G and a partition $\mathcal{P}$ of $V(G)$, the graph with vertices $\mathcal{P}$ and an edge XY whenever there is an edge between parts X and Y in G. p. 124

R

rank, grid *see* grid

rank-k division *see* division

refinement $\mathcal{P}'$ is a refinement of $\mathcal{P}$, and $\mathcal{P}$ a *coarsening* of $\mathcal{P}'$, if they are partitions of the same set, and each part $X \in \mathcal{P}'$ is a subset of some $Y \in \mathcal{P}$. Equivalently, each part of $\mathcal{P}$ the union of some parts of $\mathcal{P}'$. p. 17

- **k-refinement** $\mathcal{P}'$ is a k-refinement of $\mathcal{P}$, and $\mathcal{P}$ a k-coarsening of $\mathcal{P}'$, if each part $P \in \mathcal{P}$ is equal to the union of at most k parts of $\mathcal{P}'$. p. 41

relational

- **signature** Set $\{R_1, \ldots, R_k\}$ of *relation symbols*, each of which is given an *arity* $\mathrm{ar}(R_i) \in \mathbb{N}$. p. 59
- **structure** For a signature Σ, a Σ-structure S consists of a set $V(S)$ of vertices, also called *domain* or *universe*, and for each symbol $R \in \Sigma$ of arity $r = \mathrm{ar}(R)$, a *valuation* of the symbol as a relation $R(S) \subseteq V(S)^r$. p. 59

S

separable permutation Permutation constructed by direct and skew sums. Equivalently, permutation avoiding the patterns 2413 and 3142. p. 95

small A class of graphs (or relational structures) containing at most $c^n \cdot n!$ distinct graphs with vertices $\{1, \ldots, n\}$, for some constant c and any n. p. 54, , *see also* tiny

sparse twin-width (stww(G)) For a graph G the minimum over all contraction sequences $\mathcal{P}_n, \ldots, \mathcal{P}_1$ of $\max_i \Delta(G/\mathcal{P}_i)$. p. 124

stable set *see* independent set

subgraph In G, a graph obtained by removing any subset of edges and of vertices (together with their incident edges). p. 4, *see also* induced subgraph

T

tiny A class of graphs (or relational structures) containing at most c^n graphs on n vertices up to isomorphism, for some constant c and any n. Implies *small*. p. 54

tournament Directed graph $T = (V, A)$ where for any vertices $x \neq y$, exactly one of the directed edges $x \to y$ or $y \to x$ is in A. p. 70

trigraph Structure consisting of a set V of vertices, with between two distinct vertices either no edge, a normal edge, or an *error* (or *red*) edge. p. 11

- **quotient** ($Tri(G, \mathcal{P})$) For a graph G and a partition $\mathcal{P}$ of $V(G)$, the trigraph $Tri(G, \mathcal{P})$ with vertices $\mathcal{P}$, in which parts are connected by an edge when they are fully connected, no edge when they are fully disconnected, and an error edge otherwise, i.e. when they are non-homogeneous. p. 11

twin-width (tww(G)) Minimum width of a contraction sequence. p. 12

twins Two vertices with equal neighourhoods (excluding themselves). p. 13

- **k-near twins** Two vertices with neighbourhoods differing on at most k vertices (excluding themselves). p. 13

INDEX