Mémoire d'habilitation à diriger des recherches

Studying Graphs: Structure via Rank-Width, and Listing of Minimal Dominating Sets

MAMADOU MOUSTAPHA KANTÉ

Soutenue le 14/12/2015 en présence du jury suivant:
BRUNO COURCELLE (Université de Bordeaux)
MICHEL HABIB (Université Paris 7)
STEFAN KREUTZER (TU Berlin)
CHRISTIAN LAFOREST (Université Blaise Pascal)
LHOUARI NOURINE (Université Blaise Pascal, responsable tutélaire)
SANG-IL OUM (KAIST)
CHRISTOPHE PAUL (DR CNRS)

Rapporteurs: FEDOR FOMIN (Bergen University), MICHEL HABIB, STEFAN KREUTZER, SANG-IL OUM

Acknowledgments. I first gracefully thank Michel Habib, Fedor Fomin, Stephan Kreutzer, Lhouari Nourine and Sang-il Oum for accepting to review this work, and Bruno Courcelle, Christian Laforest and Christophe Paul for their presence in the defense jury, and all of them for remarks made to help improve the manuscript. This work would not be possible without the relatively peace environment I can find at home and the writing of this document would take much long time without the very insistence of my wife, and also mother and father, and so I am indebted to all my family for their support. I also thank all my co-authors and the people who participated to my scientific career by their help or advice. I finally thank Beatrice Bourdieu who managed to prepare all the administrative parts of the defence, and Hervé Kerivin and Benny Hill for their help in improving the english.

Contents

Chapter 1

Introduction

This document presents my main research activities during the last six years and concern problems in combinatorics and can be classified into two principal themes. The first one is a continuation of a part of my works during my PhD, namely an extension of the notion of *rank-width* to 2-structures, and the second part deals with the *enumeration of minimal dominating sets in graphs*. There is a priori no *fil conducteur* between these two parts except that one can *easily* enumerate minimal dominating sets in graphs of bounded rank-width. However, my own work on the enumeration of minimal dominating sets will take advantage of the acquired knowledge on graph decompositions.

1.1 Rank-Width

Graphs[1] are probably one of the simplest objects, not to say the simplest one, in combinatorics. But, they are probably the fundamental object in (theoretical) computer science and are used to model so many situations in several areas such as biology, economics, social sciences and so on. So, it is important to understand the structure of graphs for twofold: first being able to classify them (no matter what we mean) as any mathematical object, and second solve the algorithmic questions arising in the use of graphs as models.

The main and common tool in graph theory is the notion of *graph decomposition*. Decomposing a graph G means transforming G into a *target* graph H that is supposed to be simpler and such that (1) one can reconstruct G from H, (2) the algorithmic as well structural properties of H can be transferred to G. This simple paradigm, known since Descartes as *divide and conquer*, has proven its usefulness in settling several important problems in graph theory. We can cite the proof of the Strong Perfect Graph Theorem [25], the proof of the Graph Minor Theorem [144], or the seminal work by Martin Grohe on the definability of PTIME in minor-closed classes of graphs.

The simplest graphs are trees and it is well-known since the sixties that any problem expressible in *monadic second-order logic* (MSOL for short), among them several NP-complete problems, can be solved in polynomial time on trees [26, 153]. Moreover, in the structural point of view trees admit several characterisations: (1) connected graphs without cycles as induced subgraphs, (2) connected graphs without a triangle as a minor, (3) critically connected graphs, (4) trees are well-quasi-ordered by the minor relation, etc. [53]. Hence, trees are naturally good candidates as target graphs in graph decompositions and in fact are the most considered in structural and algorithmic graph theory: *tree-decomposition* [141], *modular decomposition* [68], *split decomposition* [44], *rank-decomposition* [130] to cite the most studied and fruitful ones[2]. Even though split decomposition [13, 18, 138] and modular decomposition [136] have proven their importance in structural and algorithmic graph theory, the most important ones are indisputably tree-decomposition and rank-decomposition, and their variants.

[1] In most cases we deal with simple graphs, *i.e.*, without parallel edges and loops.

[2] Some ad-doc decompositions have been proposed to solve open problems, for instance the given decomposition of perfect graphs in [25].

Tree-decomposition played a central role in the proof of the Graph Minor Theorem and is associated with the now established graph parameter *tree-width* [142] which measures how far a graph is from being a tree. Indeed since the seminal paper by Courcelle [32] stating that every $MSOL_2$ property can be tested in linear time in graph classes of bounded tree-width[3] (extending then the result of Thatcher and Wright [153]), tree-width is popular in algorithmics and is now a mature graph parameter particularly in Fixed Parameter Theory and has found many applications in several areas of theoretical computer science, *e.g.*, Fixed Parameter Theory, Exact Exponential algorithms, CSP, to cite a few. However, only sparse graphs can have bounded tree-width.

On the other hand some dense graph classes have a tree-like structure, *e.g.*, co-graphs or distance-hereditary graphs, which were used in the past to provide polynomial time algorithms for some difficult problems. Parallel to the works in combinatorics, there were some attempts to extend the Formal Language Theory to finite structures, particularly finite graphs. Several graph operations extending word concatenation were proposed and grammars, called *graph grammars*, that allow to generate graphs can be defined [36, 119]. One of the multiple interests of this research line is the possibility to give uniform algorithms for a class of problems or use induction to prove non trivial properties. Moreover, graph grammars appear to be a fundamental tool with respect to the research line which attempts to characterise or classify graph classes that can be expressed in MSOL [36]. Among the many arisen graph grammars two were particularly adopted by the graph theory community: the *Hypergraph Replacement grammar* (HR grammar for short) and the *Vertex Replacement grammar* (VR grammar for short). The HR grammar is indeed related to the tree-width notion since a graph class has bounded tree-width if it is *HR equational*[4] [32, 36]. Moreover, the seminal result by Courcelle [32] was proved using the HR definability of graphs of bounded tree-width. Further, the series of papers in the Graph Minor Theory were used to solve some questions about the MSOL definability of some graph classes (see the book [36] for examples and/or references). As for the VR grammar Courcelle et al. derived the complexity measure *clique-width* [40]. Clique-width extends tree-width because every graph class of bounded tree-width has bounded clique-width [40, 28] while the converse is not true, for instance co-graphs have clique-width 2, but unbounded tree-width. Indeed, several dense graph classes which do not have bounded tree-width have bounded clique-width : distance-hereditary graphs [79], graphs with few P_4s [40, 120], (see for instance [48, 90, 117] for other examples). Moreover, every $MSOL_1$ property can be checked in linear time in graph classes of bounded clique-width, provided the clique-width expressions are given as inputs[5] [30]. However, contrary to tree-width which is defined in the basis of a combinatorial decomposition, clique-width suffers from the main drawback of parameters arising from graph grammars: the difficulty to construct a parse tree. Indeed, for fixed k, there is no known polynomial time algorithm that, given a graph G, computes its clique-width expression witnessing its clique-width, except for $k \leq 3$ [27]. Maybe the recent characterisation of clique-width in terms of nested set partitions [38] can shed some light on the computation of clique-width, but by now I do not see any new insight this characterisation has added. Recall that approximating clique-width is difficult [61].

Rank-decomposition and the associated complexity measure *rank-width* was defined by Oum and Seymour [135] in order to approximate the clique-width of undirected graphs. Indeed, they proved that any clique-width k-expression can be transformed into a rank-decomposition of rank-width k and which in return can be transformed into a clique-width $(2^{k+1}-1)$-expression. Moreover, they provided a polynomial-time algorithm for checking whether a graph has rank-width at most k. But, clique-width is defined in a more general setting, namely for directed edge-coloured graphs, which is not the case of rank-width. But, this seems not restrictive since we can embed any directed edge-coloured graph G into an undirected bipartite graph $B(G)$ in such a way that the clique-width of G is approximated by the rank-width of $B(G)$, and then we can use rank-width to also approximate the clique-width of directed edge-coloured graphs [36]. Nevertheless, this encoding is not satisfactory for several reasons because, usually, we are mostly interested in the following points when studying a graph class $\mathcal{C}$:

1. a polynomial-time algorithm to decide whether a given graph is in $\mathcal{C}$.

[3] A graph parameter wd is a monotone increasing function from graphs to positive integers. We say that a graph class $\mathcal{C}$ has bounded wd if there is a constant c such that for each graph G in $\mathcal{C}$, $wd(G) \leq c$.

[4] The notion of equational set is the analog of context-free grammar in the setting of graph grammars [36].

[5] With every term from a given grammar is associated a tree which reflects the structure of the associated generated object from the algebra associated with the grammar. This tree is usually called the *parse tree* or *expression*. We refer to [36] for more information.

2. The existence of a quasi-order $\preceq$ to characterise $\mathcal{C}$ by a finite list of excluded configurations. Such a characterisation can allow certificates for recognition algorithms. It is moreover desirable, for fixed H, to be able to check in polynomial time whether $H \preceq G$ for a given graph $G \in \mathcal{C}$.
3. If a nice quasi-order $\preceq$ is known, does it well-quasi-order $\mathcal{C}$? all graphs?
4. Which problems are solvable in polynomial time in $\mathcal{C}$?

With respect to those four items, tree-width and the minor pre-order behave well. Indeed, From [9] we can decide in time $f(k)\cdot n$ whether a graph has tree-width $\leq k$, and from the Graph Minor series we know that points (2) and (3) admit positive answers, and from [32] we know that $MSOL_2$ properties can be checked in linear time. But, as for clique-width we only know that it is preserved by the induced subgraph ordering which is not a well-quasi-ordering even in graphs of bounded clique-width: cycles have clique-width at most 4 but are not well-quasi-ordered by the induced subgraph ordering. Nonetheless, rank-width, unlike clique-width, behaves better and is related to the *vertex-minor* quasi-order.

Definition 1.1 (Vertex-Minor [13, 130]). *Given a graph G and a vertex x the* local complementation *at x, denoted by $G*x$, consists in replacing the subgraph induced on the neighbours of x by its complement, and a graph H is a vertex-minor of a graph G if H is isomorphic to a graph obtained from G by applying a sequence of local complementations and deletions of vertices. Given an edge xy of G, the* pivot-complementation *at xy is the graph $G \wedge xy := G*x*y*x = G*y*x*y$, and H is a* pivot-minor *of G if H is isomorphic to a graph obtained from G by applying a sequence of pivot-complementations and deletions of vertices.*

Notice that a pivot-minor is always a vertex-minor. The following are analogues of similar results known for tree-width and graph minor relation.

1. rank-width is preserved under local complementations and hence bounded rank-width is preserved under vertex-minor [130],
2. for fixed k, we can check whether a given n-vertex graph G has rank-width at most k in time $f(k)\cdot n^3$ [86],
3. from [30, 135] we can check in cubic time every $MSOL_1$ property in graphs of bounded rank-width (a direct proof can be found in [39] where a graph grammar that characterises rank-width is proposed),
4. graph classes of bounded rank-width are well-quasi-ordered under pivot-minor [131, 134],
5. rank-width is intimately related to the *branch-width* of binary matroids [130, 131], and even though it is still open whether graphs are well-quasi-ordered under vertex-minor, the announced proof by Geelen et al. of the "Matroid Minors's conjecture" and also Rota's conjecture [73] suggests that the answer is YES[6],
6. even though checking in polynomial time whether a fixed graph H is a vertex-minor of a given graph G is open, we have a positive answer in graphs of bounded rank-width [41].

Now, the encoding of directed edge-coloured graphs into undirected bipartite graphs does not allow to translate any of those positive properties to the directed edge-coloured graphs except an approximation of their clique-width. For instance, a vertex-minor of $B(G)$ is not necessarily the encoding of some directed edge-coloured graphs. Instead, we can ask whether we can define an analogue of rank-width for directed edge-coloured graphs in such a way that analogues of (1)-(6) exist. During my PhD I proposed an extension of rank-width to directed graphs and some partial answers to (1)-(3) were proposed. In Chapter 3 I will present a more robust definition which extends to directed edge-coloured graphs and for which Michael Rao and I were able to prove analogues of (1)-(6) [91, 103, 104].

Well-quasi-ordering theorems are interesting in the sense that they prove the existence of finite lists of obstructions for closed sets. But, for algorithmic purposes they are useless and it is common to search explicitly for the list of excluded configurations. In the case of tree-width and minor closed classes of graphs several works have been done to identify the obstructions, we can cite the following notable ones:

[6] Rota's conjecture says that finitely representable matroids are characterised by a finite list of excluded matroid minors, while the Matroid Minor's conjecture states that finitely representable matroids are well-quasi-ordered by the matroid minor, which generalizes Graph Minor Theorem [73].

1. Graph classes of unbounded path-width contain trees as minors [141].
2. Graph classes of unbounded tree-width contain planar graphs as minors [142].
3. The sizes of minor obstructions for path-width as well as for tree-width have known upper bounds [113].
4. One can compute the minor obstructions of an HR-equational minor-closed set of graphs, provided the HR equations are given [31, 113].

It is worth noticing that it is not always possible to effectively compute the set of obstructions [60, 35]. As for rank-width we are far from proving analogues, except that we know an upper bound on the size of obstructions for rank-width at most k [130]. Nevertheless, precise conjectures were asked.

Conjecture 1 ([33])**.** *Every graph class of unbounded linear rank-width*[7] *contains all trees as vertex-minors.*

Conjecture 2 ([130])**.** *Every graph class of unbounded rank-width contains all bipartite circle graphs as vertex-minors.*

Conjecture 2 is natural once the link between binary matroids and rank-width is established [130], and also knowing its matroid analogue is true [72] and the well-established link between circle graphs and planar graphs [50]. Currently, Conjecture 2 is known to be true only for bipartite graphs through [130, 72], line graphs and circle graphs [133]. However, nothing is known about Conjecture 1 and I spent more than three years trying to solve it, without any success except in the case of distance-hereditary graphs. Observe that Conjecture 1 would imply the following matroid analogue conjecture which is also open.

Conjecture 3. *Every matroid class of unbounded path-width contains cycle matroids of all outer-planar graphs.*

I was interested in Conjecture 1 for the following reasons. Blumensath and Courcelle used the fact that graph classes of unbounded path-width contain all trees as minors to propose a hierarchy of incidence graphs based on $MSOL_2$ transductions [8]. I have been interested in obtaining a similar hierarchy with $MSOL_1$ transductions and solving Conjecture 1 is a first step. Unfortunately all the attempts to solve Conjecture 1 failed. I will present in Chapter 4 the partial results obtained in this direction. In collaboration with I. Adler [1] we were able to show that path-width and linear rank-width coincide on trees which is quite surprising because the two parameters are structurally different. We were also able to characterise the linear clique-width of trees with respect to their path-width, and this is more or less the only nontrivial graph class with such a characterisation of their clique-width. This implies in particular from [58] that we can characterise in a recursive way the linear rank-width (or linear clique-width) of trees. Since distance-hereditary graphs are totally decomposable with respect to the split decomposition and then are tree-like we investigated their linear rank-width in collaboration with I. Adler and O-J. Kwon [2] and showed a recursive characterisation of their linear rank-width, similar to the one of trees. As a fist consequence we can compute the linear rank-width of distance-hereditary graphs in polynomial time and this contrasts with the computation of their path-width that is proved NP-complete in [111]. This characterisation were next used to compute the distance-hereditary obstructions for linear rank-width k, for any k, and also to prove Conjecture 1 in the case of distance-hereditary graphs. Another nice consequence is the computation of the path-width of 2-connected matroids of branch-width 2. All these results were published as an extended abstract in [2], and two journal versions are submitted [3, 4].

1.2 Enumeration Algorithms

Sometimes we are also interested in listing all the solutions of a question such as in databases where an SQL request is typically asking for the collection of entries satisfying some conditions. Enumeration problems[8] are at least as old as mathematics or algorithmic since people always have met the need to list objects, *e.g.*, prime numbers, groups of size n, permutations of size n, partitions, partially ordered sets, etc. (see for

[7] Linear rank-width is the linearised variant of rank-width as path-width is the linearised version of tree-width.

[8] An enumeration problem consists in listing, without duplications, exactly the elements of a set.

instance the book [145] where several enumeration problems are discussed or the book [62] where counting and sampling problems are mostly studied).

Even though enumeration problems had appeared at the beginning of computer science much efforts have been concentrated on optimisation problems where a lot of research has been done on algorithms and complexity. Indeed, it was widely believed that enumeration problems are more difficult than the counting, research or optimisation problems as once we can enumerate we can count the number of solutions, give a solution or compute an optimal solution. But, nowadays the mass of information has created the need to analyse the data in order to provide some statistics or extrapolates the distribution of the data or even help in making decisions, and for that purposes efficient enumeration algorithms are needed (see [106] for references). Moreover, in the Exact Exponential Algorithms community many state-of-the-art algorithms, coupled with a combinatorial bound on the number of solutions, are based on enumerating the set (or subset) of solutions [64, 47], and many computational problems are intimately related to enumeration ones such as the computation of a polynomial [150] (see also the webpage [121] or the book [145] for several references on graph polynomials).

In algorithmics one has to deal with limited ressources such as space and time. In standard complexity theory (an optimisation, a counting or a research problem) the solution is smaller than the input and an efficient algorithm is an algorithm which runs in time polynomial in the size of the input. But, for enumeration problems the set of solutions is usually exponential in the size of the input: the number of binary words of size n is 2^n, the number of labelled trees of size n is n^{n-2}, the number of independent sets in an n-vertex graph is $O(3^{n/3})$, etc. So, with respect to the standard complexity criteria enumeration algorithms are not efficient even though the algorithm takes exactly $O(m)$ time where m is the number of solutions. So, it makes sense to search for another way to measure the efficiency of enumeration algorithms, and the natural way is to take into account the sizes of the input and of the output. This has the advantage of allowing polynomial (total) time enumeration algorithms. Enumeration algorithms with running time bounded by a polynomial $p(n,m)$ with n the size of the input and m the size of the output are called *output-polynomial algorithms* (the class of such enumeration problems is called **TotalP** for short). Another way to measure the efficiency of an enumeration algorithm is to measure the *delay*, *i.e.*, the maximal time between two consecutive solutions, and with respect to this criteria two classes are widely considered: the enumeration problems with delay polynomial on the size of the input (**DelayP**) and those with delay polynomial on the size of the input and the number of already output solutions (**IncP**). As usual, given a complexity class[9] $\mathcal{C}$ we can be interested in knowing which problems are included in $\mathcal{C}$, and also compare it with other complexity classes. For instance, it is easy to check that **DelayP** $\subseteq$ **IncP** $\subseteq$ **TotalP**, and it is even proved that **IncP** $\subset$ **TotalP** [150], and it is known that several enumeration problems, under the assumption $P \neq NP$, are not included in **TotalP** (see for instance [114, 150] to cite a few).

Nonetheless, it is known that several enumeration problems are in **TotalP**: words of size n [81], maximal independent sets (or minimal vertex-covers) in graphs [155], maximal cliques or bicliques [74], feedback vertex sets [146], bases (or circuits) of a matroid [11], maximal matchings [158], minimal separators [147], sequence mining [49], etc. (see also the book [145] for more examples). On the other hand, the complexity statuses of many enumeration problems in (hyper)graphs are still open and I have been mostly interested in the *Hypergraph Dualisation problem* (TRANS-ENUM) which consists in enumerating all the (inclusion-wise) minimal transversals of a hypergraph. A hypergraph is a collection $\mathcal{E}$ of subsets of a finite set V and a transversal of $\mathcal{E}$ is a subset of V that intersects every $E \in \mathcal{E}$. It is a fifty-year open problem to decide whether TRANS-ENUM belongs to **TotalP**. TRANS-ENUM is a well-studied problem due to the numerous applications in several areas such as databases, data mining, artificial intelligence, etc. and also because several other enumeration problems in (hyper)graphs are special cases of it. For instance, most of the tractable cases cited above are special cases of TRANS-ENUM and many enumeration problems related to data mining, database theory or artificial intelligence are either equivalent or can be reduced to TRANS-ENUM [57, 128].

Resolving the complexity status of TRANS-ENUM is a long-standing open question and until now only few significant tractable cases have been identified and the best algorithm is the quasi-polynomial time algorithm by Fredman and Khachiyan [65] which runs in time $N^{o(\log N)}$ where N is the sum of the sizes of

[9] **TotalP** is called by abuse a complexity class, but it is not yet proved to be one as we still fail in finding what a reduction should be and also a complete problem, whereas for instance #P is well-defined *w.r.t.* criteria.

the input and of the output. It is worth noticing that several other algorithms with the same running time have been given in the literature [56, 107, 125], and it is known from [80] that it belongs to $DSPACE[\log^2 n]$. Such a lack of success may be explained by a lack in structural hypergraph theory whereas structural graph theory have been investigated for solving several graph enumeration problems. Therefore, knowing that TRANS-ENUM is equivalent to an enumeration problem in graphs would be valuable and would bring new strategies for tackling this fifty-year open problem.

A *dominating set* in a graph is a subset D of its vertex set such that each vertex is either in D or has a neighbour in D. The MINIMUM DOMINATING SET problem is a classic and well-studied optimisation problem due to its numerous applications in networks, graph theory [83] and with respect to Fixed Parameter Theory it constitutes a central problem with its variants [54, 64]. DOM-ENUM consists in enumerating all the (inclusion-wise) minimal dominating sets, and it is easy to see that it is a special case of TRANS-ENUM since D is a minimal dominating set of a graph if and only if D is a minimal hitting set of its *closed neighbourhood hypergraph*[10]. But, contrary to other well-known problems in graphs (independent sets, cliques, bi-cliques, spanning trees, etc.) it was not known whether DOM-ENUM is in **TotalP**. This is rather surprising, especially since every maximal independent set is a minimal dominating set and one would expect that the algorithm in [155] could be adapted to prove that DOM-ENUM $\in$ **TotalP**. But, surprisingly we proved in [97] that DOM-ENUM and TRANS-ENUM are equivalent in the sense that DOM-ENUM $\in$ **DelayP** if and only if TRANS-ENUM $\in$ **DelayP**, and proved at the same time that DOM-ENUM is equivalent to the enumeration of several other minimal dominating like sets (total dominating sets, connected dominating sets in chordal graphs, etc.) (reduction given in Chapter 5). We expected to benefit from the tools in structural graph theory to resolve TRANS-ENUM and during the last 4 years with my co-authors we considered special cases that I will present in the following lines. We did not consider complexity questions except which well-studied graph classes $\mathcal{C}$ are tractable with respect to DOM-ENUM and we aimed to better understand the current tools for enumeration.

An *independent system* is a hereditary collection of subsets of a finite set V. For instance, the independent sets of a matroid form an independent system. If an independent system $\mathcal{I}$ is given by an oracle $\mathcal{O}_{\mathcal{I}}$, then after linearly ordering V the call EnumIndSet$(V,\emptyset,\mathcal{O}_{\mathcal{I}})$ enumerates the elements of $\mathcal{I}$ with delay $O(n+f(n))$ where n is the size of V and f is the time needed by $\mathcal{O}_{\mathcal{I}}$ to return YES or NO (EnumIndSet is depicted in Figure 1.1). But, since independent systems are characterised by their maximal sets one would like instead to enumerate only the maximal ones. For instance, if $\mathcal{T}$ is the set of transversals of a hypergraph $\mathcal{E} \subseteq 2^V$, then $\mathcal{I} := \{V \setminus T \mid T \in \mathcal{T}\}$ forms an independent system, and the question whether TRANS-ENUM belongs to **TotalP** is the same as asking whether the enumeration of maximal elements in $\mathcal{I}$ belongs to **TotalP**. The algorithm EnumIndSet cannot be adapted to list the maximal sets of $\mathcal{I}$ for the reason that we could generate the same maximal independent set several times. Indeed, it is proved in [114] that, unless $P = NP$, the enumeration of the maximal sets of an independent system does not belong to **TotalP**. On the other hand, if we can answer the following question SUBSETIND in polynomial time, then the call EnumMaxIndSet$(V,\emptyset,\emptyset,\mathcal{O}^s_{\mathcal{I}})$ enumerates the maximal independent sets with delay $O(n+g(n))$ where $g(n)$ is the time needed by $\mathcal{O}^s_{\mathcal{I}}$ to answer SUBSETIND (EnumMaxIndSet is depicted in Figure 1.2).

SUBSETIND (or EXTENSION PROBLEM or FLASHLIGHT)
Input. $X, Y \subseteq V$
Output. Does there exist a maximal independent set that contains X and does not intersect Y?

From [114] we know that SUBSETIND is NP-complete in general, and it remains NP-complete even when restricted to transversals in hypergraphs (see the PhD thesis [122] for a proof). As we proved in [100, 99] SUBSETIND with respect to DOM-ENUM is NP-complete even in restricted graph classes as split graphs or line graphs. But, SUBSETIND can be solved in polynomial time if we restrict the instances to some special cases. For instance, we proved in [97] that it can be solved in linear time in split graphs if (X, Y) forms a partition of the clique part and this was the key of the linear delay algorithm for DOM-ENUM in split graphs. We pushed further this idea in chordal graphs and reduce DOM-ENUM in chordal graphs to solving special instances of SUBSETIND. We refer to Chapter 6 for the details.

[10]The *closed neighbourhood hypergraph* of a graph $G = (V,E)$ is the collection $\{N_G[x] \mid x \in V\}$ where $N_G[x]$ is the set containing x and all the neighbours of x.

As recalled above in Section 1.1 dynamic programming is probably the favourite tool in solving (graph) optimisation problems and it works usually as follows: if a graph $G = f(G_1,\ldots,G_p)$ for some *composition function* f, then solve the optimisation problem (or variants depending on f) in the G_i's and then combine them to compute an optimal solution in G. Among the decompositions, those yielding linear orderings are the desirable ones because they allow to construct incremental algorithms and usually yield simpler algorithms as can be attested by the number of polynomial time (or even linear time) algorithms based on specific linear orderings in interval graphs, strongly chordal graphs, chordal graphs, distance-hereditary graphs, etc. The dynamic programming approach naturally arises in solving enumeration problems [145] and one can find in the literature two meta-algorithms, one based on a linear ordering of the ground set (*Lawler's algorithm* [114]), and another one for TRANS-ENUM based on a linear ordering of the sets of a given hypergraph (*Berge's algorithm* [7]).

Given a linear ordering $v_1,\ldots,v_n$ of the ground set of an independent system $\mathcal{I}$, Lawler's algorithm consists in computing, for each $j\in[n]$, the set $\mathcal{I}_j$ of independent sets that are maximal within $\{v_1,\ldots,v_j\}$. One easily checks that any set in $\mathcal{I}_j$ can be extended to an independent set in $\mathcal{I}_{j+1}$ and then any solution at step j can be extended to a maximal independent set of $\mathcal{I}$. But, the whole problem lies on how to compute in polynomial time all the maximal independent sets of $I\cup\{v_{j+1}\}$ within $\{v_1,\ldots,v_{j+1}\}$ for each $I\in\mathcal{I}_j$ [114]. This strategy has been used in [114, 56, 126] to show that several enumeration problems, *e.g.*, maximal packings, maximal k-partite subgraphs, facets of a convex hull, TRANS-ENUM restricted to several hypergraph classes, etc. belongs to **TotalP** (even in **DelayP** for some of them).

Berge's algorithm instead consists in taking a linear ordering $E_1,\ldots,E_m$ of a hypergraph $\mathcal{E}$ and for each $i\in[m]$ computes the minimal transversals of $\{E_1,\ldots,E_i\}$. Still one can compute in polynomial time the minimal transversals of $\{E_1,\ldots,E_i\}$ from the minimal transversals of $\{E_1,\ldots,E_{i-1}\}$, but it is not always the case that any minimal transversal of $\{E_1,\ldots,E_i\}$ can be extended into a minimal transversal of $\mathcal{E}$. Indeed, it is proved in [151] that there exist hypergraphs for which Berge's algorithm does not provide an output-polynomial time algorithm for any ordering. Nevertheless, Berge's algorithm admits a depth-first search on the solution space [107, 126] and one can try to identify the indices $i_1 < i_1 < \cdots < i_\ell = m$ such that a minimal transversal of $\{E_1,\ldots,E_{i_j}\}$ can be always extended to a minimal transversal of $\mathcal{E}$ and skip the other costly steps. If the computation of the minimal transversals of $\{E_1,\ldots,E_i\}$ from the minimal transversals of $\{E_1,\ldots,E_{i-1}\}$ is easy, we have to spend much time and use much space to compute the minimal transversals of $\{E_1,\ldots,E_{i_j}\}$ from those of $\{E_1,\ldots,E_{i_{j-1}}\}$. In some cases this turns out to be harder than an NP-complete problem. We used this idea in [100] to give a polynomial delay algorithm for the enumeration of minimal edge-dominating sets in graphs and, besides the result, the most interesting part was that we faced an NP-complete problem in constructing the solutions of $\{E_1,\ldots,E_{i_j}\}$ from the ones of $\{E_1,\ldots,E_{i_{j-1}}\}$ and we were able to overcome this NP-completeness by proposing a new way to obtain the difficult solutions from the easy ones. To our knowledge such a traversal of the set of solutions was not investigated in the past. The main ideas of the paper [100] are presented in Chapter 9.

In several areas of computer science and mathematics *parsimonious reduction* is a nice tool to prove that a problem is either difficult or easy. In the case of enumeration problems if one wants to enumerate a list of objects $\mathcal{O}$, one constructs a bijection $b:\mathcal{O}\to\mathcal{T}$ such that there is an efficient algorithm for listing the objects in $\mathcal{T}$ (see for instance [62, 145]). For instance our algorithm for DOM-ENUM in split graphs [97] is based on a bijection between the minimal dominating sets in split graphs and the members of an independent system which can be enumerated with linear delay. In [6] the authors introduce the notion of *(linear) maximum induced matching width* which, when bounded, immediately implies that several domination like problems admit polynomial time algorithms and thus extends the results in [23] on graphs of bounded rank-width to some graph classes with unbounded rank-width. For instance, graphs of bounded linear rank-width, interval graphs, permutation graphs, circular-arc graphs, k-trapezoid graphs, Dilworth-k graphs, complements of bounded degeneracy graphs, all of them have bounded linear maximum induced matching width [6], and graphs of bounded rank-width, directed path graphs have bounded maximum induced matching width [29]. In [75] we showed that if an n-vertex graph G has linear maximum induced matching width at most a constant c then one can construct, in time $O(n^c)$, a DAG whose maximal paths correspond to the minimal dominating sets of G. Since such paths can be counted in linear time and listed with linear delay we deduce that DOM-ENUM when restricted to several interesting graph classes belongs to **DelayP**. We derived also a polynomial delay and polynomial space algorithm for listing the minimal transversals of

Algorithm $\mathsf{EnumIndSet}(V, I, \mathcal{O}_\mathcal{I})$
  V:ground set linearly ordered, $\mathcal{O}_\mathcal{I}$:oracle for $\mathcal{I}$
1. **for** each $x \in V$ greater than $\max(I)$ **do**
2. **if** $\mathcal{O}_\mathcal{I}(I \cup \{x\})$ is YES, **then**
3. **output** $I \cup \{x\}$ and **call** $\mathsf{EnumIndSet}(V, I \cup \{x\}, \mathcal{O}_\mathcal{I})$
4. **end for**

Figure 1.1: Algorithm for enumerating the elements of an independent system $\mathcal{I}$.

Algorithm $\mathsf{EnumMaxIndSet}(V, X, Y, \mathcal{O}^s_\mathcal{I})$
  V:ground set linearly ordered, $\mathcal{O}_\mathcal{I}$:oracle for SUBSETIND, $X \cap Y = \emptyset$
1. **if** $X \cup Y = V$, **then output** X and **stop**
2. Let x be the smallest element in $V \setminus (X \cup Y)$
3. **if** $\mathcal{O}_\mathcal{I}(X \cup \{x\}, Y)$ is YES, **then call** $\mathsf{EnumMaxIndSet}(V, X \cup \{x\}, Y, \mathcal{O}^s_\mathcal{I})$
4. **if** $\mathcal{O}_\mathcal{I}(X, Y \cup \{x\})$ is YES, **then call** $\mathsf{EnumMaxIndSet}(V, X, Y \cup \{x\}, \mathcal{O}^s_\mathcal{I})$

Figure 1.2: Algorithm for enumerating the maximal elements of an independent system $\mathcal{I}$.

interval and circular-arc hypergraphs improving the only known incremental polynomial time algorithms [139]. This paper generalises a former one [98] where we proved a similar result for interval and permutation graphs. A generalisation of this result to more general graph classes is presented in Chapter 7.

The *reverse search* technique popularised since the paper [5] by Avis and Fukuda is one of the most powerful techniques in proving that an enumeration problem is in **TotalP** and indeed several output-polynomial time algorithms are special cases or adaptations of the reverse search technique [7, 74, 89, 114, 126, 145]. The idea consists in constructing a *transition graph* the vertices of which corresponds to the solutions and there is an arc from S to S' if $S' = f(S)$ for some defined function f. If the function f can be computed in polynomial time, the algorithm typically traverses the transition graph in a depth-first or breadth-first search manner. In [78] the authors introduced a function f based on a flipping method and proposed an enumeration algorithm for DOM-ENUM which under some hypothesis gives an incremental polynomial time algorithm. We adapted this flipping method in [77, 75] to show that DOM-ENUM restricted to chordal bipartite graphs and to unit-square graphs belongs to **IncP** (see Chapter 8).

Nota Bene

This manuscript should be seen as a summary of my principal research activities. The proofs are omitted, and can be found in the cited papers (see Chapter 11 for links), but I will sometimes recall the intermediate lemmas to give the flavour/tools of the proofs.

During the last four years I co-advised with C. Laforest the PhD thesis of B. Momège on *connectivity problems in graphs with conflicts* which are graphs with a set of pairs of edges called *conflicts*. The goal is to find subgraphs without conflicts, *e.g.*, induced paths, trees, etc. Several polynomial time problems on graphs become NP-complete in graphs with conflicts. In the papers [94, 95, 101] we considered algorithmic issues, but I preferred not to include them because I am now less interested in the subject and the questions are not totally related to the other parts of this manuscript.

Chapter 2

Preliminaries

We compile in this chapter the common notations and definitions. The power set of a set V is denoted by 2^V and the set $\{x\}$ is often written x for convenience. The size of a subset $\mathcal{C}$ of 2^V, denoted by $\|\mathcal{C}\|$, is defined as $\sum_{C\in\mathcal{C}}|C|$. For $\mathcal{C}\subseteq 2^V$ we denote by $\min(\mathcal{C})$ the set of (inclusion-wise) minimal sets in $\mathcal{C}$. We denote by $\mathbf{N}$ the set of positive integers, including 0, and for an integer n we let $[n]$ denote the set $\{1,2,\ldots,n\}$. We denote by $+$ and $\cdot$ respectively the addition and multiplication operations of any field, and by 0 and 1 the identity elements of $+$ and $\cdot$ respectively. For a prime number p and a positive integer k we denote by $\mathbb{F}_{p^k}$ the finite field of characteristic p and order p^k. For a field $\mathbb{F}$, we let $\mathbb{F}^*$ be $\mathbb{F}\setminus\{0\}$. We recall that finite fields are commutative and we refer to [115] for our field terminology.

If $f:A\to B$ is a function, we let $f|_X$, the restriction of f to $X\subseteq A$, be the function $f|_X:X\to B$ where for every $a\in X$, $f|_X(a):=f(a)$. A function $f:2^V\to\mathbf{N}$ is *symmetric* if $f(X)=f(V\setminus X)$ for every $X\subseteq V$; it is *submodular* if $f(X)+f(Y)\geq f(X\cup Y)+f(X\cap Y)$ for any $X,Y\subseteq V$.

If R is an equivalence relation on V, we write $[x]_R$ for the equivalence class of x, and V/R denotes the set of equivalence classes of R.

2.1 Graphs and Matrices

(Hyper)Graphs. We refer to [53] for our graph terminology. The vertex set of a graph G is denoted by V_G and its edge/arc set by E_G. Arcs in directed graphs are denoted as pairs of vertices, and we write xy (equivalently yx) to denote an edge between the vertices x and y in an undirected graph. The subgraph of G induced by $X\subseteq V_G$ is denoted by $G[X]$ and $G\setminus X$ denotes the induced subgraph $G[V_G\setminus X]$. For $F\subseteq E_G$, we let G-F be the subgraph $(V_G,E_G\setminus F)$.

For a vertex x of an undirected graph G, we denote by $N_G(x)$ the set of neighbours of x, let $N_G[x]:=N_G(x)\cup\{x\}$ (the *closed neighbourhood* of x), and denote by $N_G^r[x]$ be the set of vertices at distance r from x. For $X\subseteq V_G$, we let $N_G[X]:=\bigcup_{x\in X}N_G[x]$ and $N_G(X):=N_G[X]\setminus X$. As usual in all our notations we will omit the sub or sup-scripts G whenever it is clear from the context.

Let C be a (possibly infinite) set that we call the *colours*. A *C-edge-coloured graph* G is a tuple (V_G,E_G,ℓ_G) where (V_G,E_G) is a directed graph and $\ell_G:E_G\to 2^C\setminus\{\emptyset\}$ is a mapping. Its associated *underlying graph* $u(G)$ is the directed graph (V_G,E_G). Two C-edge-coloured graphs G and H are isomorphic if there is an isomorphism h between $u(G)$ and $u(H)$ such that for every $(x,y)\in E_G$, $\ell_G((x,y))=\ell_H((h(x),h(y)))$. We call h an *isomorphism* between G and H. We let $\mathscr{G}(C)$ be the class of C-edge-coloured graphs for a fixed colour set C (and we write $\mathscr{G}$ for the class of undirected graphs). Even though we authorise infinite colour sets in the definition, most of the results are valid only when the colour set is finite. It is worth noticing that an edge-uncoloured graph is an edge-coloured graph with all edges of the same colour. Notice also that C-edge-coloured graphs are equivalent to the notion of *2-structures* studied in the works of Ehrenfeucht, Harju and Rozenberg (see the book [55] for more information).

Because we deal at the same time with graphs and trees associated with them the vertices of trees are called *nodes*, and we denote by L_T the set of leaves of a tree T. A *cubic tree* is a tree where each node has degree 1 or 3. A *rooted tree* is a tree with a distinguished node called the *root*. In a rooted tree T, we denote by $\preceq_T$ the ascendant-descendant relation, *i.e.*, $u \preceq_T v$ if v is on the unique path from the root to u. Two nodes u and v of a rooted tree T are *comparable* if $u \preceq_T v$ or $v \preceq_T u$, and *incomparable* otherwise. The subtree of a rooted tree T rooted at u is the tree $T[\{v \in V_T \mid v \preceq_T u\}]$ rooted at u.

A *hypergraph* $\mathcal{H}$ is a pair $(V_\mathcal{H}, \mathcal{E}_\mathcal{H})$ with $\mathcal{E}_\mathcal{H} \subseteq 2^V \setminus \{\emptyset\}$[1]. The members of $V_\mathcal{H}$ are called vertices and those of $\mathcal{E}_\mathcal{H}$ *hyperedges*. It is worth noticing that a graph is a hypergraph with all hyperedges of size 2. The size of a hypergraph $\mathcal{H}$, denoted by $\|\mathcal{H}\|$, is defined as $|V_\mathcal{H}| + \|\mathcal{E}_\mathcal{H}\|$. It is convenient sometimes to consider a subset $\mathcal{E}$ of 2^V as a hypergraph, and the reader should keep in mind, in that case, that we refer to the hypergraph $(\bigcup_{E\in\mathcal{E}} E, \mathcal{E})$. A hypergraph $\mathcal{H}$ is said *simple* if $\mathcal{E}_\mathcal{H} = \min(\mathcal{E}_\mathcal{H})$ and $V_\mathcal{H} = \bigcup_{E\in\mathcal{E}_\mathcal{H}} E$. For a hypergraph $\mathcal{H}$ we let $\min(\mathcal{H})$ be the simple hypergraph $\min(\mathcal{E}_\mathcal{H})$. The set of hypergraphs is denoted by $\mathscr{H}$.

Graph Parameters. A *parameter* on $\mathscr{G}(C)$ is a function $wd : \mathscr{G}(C) \to \mathbf{N}$ that is invariant under isomorphism. Two parameters on $\mathscr{G}(C)$, say wd and wd', are *equivalent* if there exist two mutually increasing integer functions f and g such that for every edge-coloured graph $G \in \mathscr{G}(C)$, $f(wd'(G)) \leq wd(G) \leq g(wd'(G))$.

A *layout* of a finite set V is a pair $(T, \mathcal{L})$ of a tree T and a bijective function $\mathcal{L} : V \to \mathrm{L}_T$. For each edge e of T, the connected components of T-e induce a bipartition $(X_e, V \setminus X_e)$ of L_T, and thus a bipartition $(X^e, V\backslash X^e) = (\mathcal{L}^{-1}(X_e), \mathcal{L}^{-1}(V \setminus X_e))$ of V (we will omit the subscript or superscript e when it is clear from the context). A *linear layout* of a finite set V is a layout $(T, \mathcal{L})$ of V such that T is a caterpillar.

Let $f : 2^V \to \mathbf{N}$ be a symmetric function and $(T, \mathcal{L})$ a (linear) layout of V. The *(linear) f–width of each edge e of T* is defined as $f(X^e)$ and the *(linear) f–width of $(T, \mathcal{L})$* is the maximum f–width over all edges of T. The *(linear) f–width of V* is the minimum (linear) f–width over all (linear) cubic layouts of V.

Matrices. For sets R and C, an *(R, C)-matrix* is a matrix where the rows are indexed by elements in R and columns indexed by elements in C. If the entries are over a field $\mathbb{F}$, we call it an (R, C)-matrix over $\mathbb{F}$. For an (R, C)-matrix M, if $X \subseteq R$ and $Y \subseteq C$, we let $M[X, Y]$ be the submatrix of M where the rows and the columns are indexed by X and Y respectively. Along this manuscript matrices are denoted by capital letters, which will allow us to write m_{xy} for $M[x, y]$ when it is possible. The matrix rank-function is denoted by rk. We will write $M[X]$ instead of $M[X, X]$ and such submatrices are called *principal submatrices*. The transpose of a matrix M is denoted by M^t, and the inverse of M, if it exists, *i.e.*, if M is *non-singular*, is denoted by M^{-1}. A (V_1, V_1)-matrix M is said *isomorphic* to a (V_2, V_2)-matrix N if there exists a bijection $h : V_1 \to V_2$ such that $m_{xy} = n_{h(x)h(y)}$. We refer to [116] for our linear algebra terminology.

2.2 Enumeration

Let $\mathcal{D}$ be a family of subsets of the vertex set of a given hypergraph $\mathcal{H}$ on n vertices and m hyperedges. An *enumeration algorithm* for $\mathcal{D}$ lists the elements of $\mathcal{D}$ without repetitions. An *enumeration problem* for $\mathcal{D}$ asks for an enumeration algorithm for $\mathcal{D}$. The running time of an enumeration algorithm $\mathcal{A}$ is said to be *output polynomial* if there is a polynomial $p(x, y)$ such that all the elements of $\mathcal{D}$ are listed in time bounded by $p((n+m), |\mathcal{D}|)$. Assume now that $D_1, \ldots, D_\ell$ are the elements of $\mathcal{D}$ enumerated in the order in which they are generated by $\mathcal{A}$. Let us denote by $T(\mathcal{A}, i)$ the time $\mathcal{A}$ requires until it outputs D_i, also $T(\mathcal{A}, \ell+1)$ is the time required by $\mathcal{A}$ until it stops. Let $delay(\mathcal{A}, 1) = T(\mathcal{A}, 1)$ and $delay(\mathcal{A}, i) = T(\mathcal{A}, i) - T(\mathcal{A}, i-1)$. The *delay* of $\mathcal{A}$ is $\max\{delay(\mathcal{A}, i)\}$. Algorithm $\mathcal{A}$ runs in *incremental polynomial* time if there is a polynomial $p(x, i)$ such that $delay(\mathcal{A}, i) \leq p(n+m, i)$. Furthermore $\mathcal{A}$ is a *polynomial delay* algorithm if there is a polynomial $p(x)$ such that the delay of $\mathcal{A}$ is at most $p(n+m)$. Finally $\mathcal{A}$ is a *linear delay* algorithm if $delay(\mathcal{A}, 1)$ is bounded by a polynomial in $n+m$ and $delay(\mathcal{A}, i)$ is bounded by a linear function in $n+m$.

We denote by **TotalP**, **IncP**, **DelayP** and **DelayL** respectively the set of enumeration problems which admit an output polynomial enumeration algorithm, an incremental polynomial time enumeration algorithm, an enumeration algorithm with a polynomial delay and an enumeration algorithm with linear delay. Let $\mathbf{ep} \in \{\mathbf{TotalP}, \mathbf{IncP}, \mathbf{DelayP}, \mathbf{DelayL}\}$. For two enumeration problems P and P' we write $P \leq_{\mathbf{ep}} P'$ if an enumeration algorithm $\mathcal{A}' \in \mathbf{ep}$ for P' implies an enumeration algorithm $\mathcal{A} \in \mathbf{ep}$ for P; and we will say that

[1] We remove the $\emptyset$ as a member of $\mathcal{E}_\mathcal{H}$ because of technical reasons that will appear in the enumeration part.

P and P' are **ep**-*equivalent* if $P \leq_{\mathbf{ep}} P'$ and $P' \leq_{\mathbf{ep}} P$. See for instance [122] for some robust notions of reductions $f_{\mathbf{ep}}$ between enumeration problems such that whenever $P' = f_{\mathbf{ep}}(P)$, then $P \leq_{\mathbf{ep}} P'$.

Let us now define the two main enumeration problems we dealt with. Given a hypergraph $\mathcal{H}$, a *transversal* of $\mathcal{H}$ is a subset T of $V_{\mathcal{H}}$ such that $T \cap E \neq \emptyset$ for all $E \in \mathcal{E}_{\mathcal{H}}$. The set of (inclusion-wise) minimal transversals $\mathcal{H}$ is denoted by $tr(\mathcal{H})$. It is folklore to check that $tr(\mathcal{H}) = tr(\min(\mathcal{H}))$. The enumeration problem TRANS-ENUM is the following.

TRANS-ENUM **Input.** A simple hypergraph $\mathcal{H}$ **Output.** $tr(\mathcal{H})$

Given an undirected graph G, a *dominating set* of G is a subset D of V_G such that every vertex is either in D or has a neighbour in D. The set of (inclusion-wise) minimal dominating sets in a graph G is denoted by $\mathcal{D}(G)$. The enumeration problem DOM-ENUM is the following.

DOM-ENUM **Input.** An undirected graph G **Output.** $\mathcal{D}(G)$

Given an undirected graph G, the *closed neighbourhood hypergraph* of G, denoted by $\mathcal{N}(G)$, is the hypergraph with vertex set V_G and with hyperedges the collection $\{N_G[x] \mid x \in V_G\}$. The following is folklore.

***Lemma* 2.1** (Folklore [22])**.** *For every undirected graph G, $D \subseteq V_G$ is a dominating set of G if and only if D is a transversal of $\mathcal{N}(G)$. Hence, $\mathcal{D}(G) = tr(\mathcal{N}(G))$.*

From Lemma 2.1 we can deduce that DOM-ENUM $\leq_{\mathbf{DelayP}}$ TRANS-ENUM. We know from [12] that there exist hypergraphs which are not closed neighbourhood hypergraphs of any graph. One can however wonder whether with every hypergraph $\mathcal{H}$ one can associate a graph $G(\mathcal{H})$ such that $tr(\mathcal{H}) = \mathcal{D}(G(\mathcal{H}))$. The following answers in the negative, but we will see in Chapter 5 that TRANS-ENUM $\leq_{\mathbf{DelayP}}$ DOM-ENUM indeed.

***Proposition* 2.2** ([97])**.** *For every function $f : \mathscr{H} \to \mathscr{G}$, there exists $\mathcal{H} \in \mathscr{H}$ such that $tr(\mathcal{H}) \neq \mathcal{D}(f(\mathcal{H}))$.*

Given a subset T of the vertex set of a hypergraph $\mathcal{H}$ and a vertex $x \in T$, we denote by $P_{\mathcal{H}}(T, x)$ the set $\{E \in E_{\mathcal{H}} \mid E \cap T = \{x\}\}$, called the set of *private neighbours* of x. When $\mathcal{H} = \mathcal{N}(G)$ for some graph G, then $P_{\mathcal{H}}(T, x)$ is in one-to-one correspondence with $\{y \in V_G \mid N_G[y] \cap T = \{x\}\}$, and we will prefer this latter whenever we deal with graphs. $T \subseteq V_{\mathcal{H}}$ is an *irredundant set* if $P_{\mathcal{H}}(T, x) \neq \emptyset$ for all $x \in T$ [22]. The following is easy to check.

***Lemma* 2.3** (folklore)**.** *$T \subseteq V_{\mathcal{H}}$ is a minimal transversal of $\mathcal{H}$ if and only if T is a transversal and is irredundant.*

Given an undirected graph G, $D \subseteq V_G$ is a *total dominating set* of G if each vertex in G has a neighbour in D. The set of (inclusion-wise) minimal total dominating sets of G is denoted by $\mathcal{TD}(G)$, and the associated enumeration problem TDOM-ENUM is the following.

TDOM-ENUM **Input.** An undirected graph G **Output.** $\mathcal{TD}(G)$

Given an undirected graph G, the *open neighbourhood hypergraph* of G, denoted by $\mathcal{N}^o(G)$, is the hypergraph with vertex set V_G and with hyperedges the collection $\{N_G(x) \mid x \in V_G\}$.

***Lemma* 2.4** ([154])**.** *For every undirected graph G, $D \subseteq V_G$ is a total dominating set of G if and only if D is a transversal of $\mathcal{N}^o(G)$. Hence, $\mathcal{TD}(G) = tr(\mathcal{N}^o(G))$.*

From Lemma 2.4 we can deduce that TDOM-ENUM $\leq_{\mathbf{DelayP}}$ TRANS-ENUM. We will prove in Chapter 5 that TRANS-ENUM $\leq_{\mathbf{DelayP}}$ TDOM-ENUM.

For an enumeration problem ENUM_{φ} in the set of all (hyper)graphs, we write $\text{ENUM}_{\varphi}(\mathscr{C})$ for ENUM_{φ} restricted to (hyper)graphs in $\mathscr{C}$.

Part I

About Rank-Width

Chapter 3

Rank-Width of Edge-Coloured Graphs

We recall in this section the results concerning the rank-width of edge-coloured graphs and that is a compilation of results published in [91, 103, 104] and mostly in collaboration with M. Rao.

3.1 Rank-Width of Edge-Coloured Graphs

Let $\mathbb{F}$ be a field and $\sigma:\mathbb{F}\to\mathbb{F}$ a bijection. We call σ a *sesqui-morphism* if σ is an involution and the mapping $\tilde{\sigma}:=[x\mapsto\frac{\sigma(x)}{\sigma(1)}]$ is an automorphism. Notice that if σ is a sesqui-morphism, then $\tilde{\sigma}$ is an involution, $\sigma(0)=0$ and $\sigma(a+b)=\sigma(a)+\sigma(b)$.

***Property* 3.1** ([91, 104])**.** *Let $\sigma:\mathbb{F}\to\mathbb{F}$ be a sesqui-morphism. Then, for all $a,b,a_i\in\mathbb{F}$, $c\in\mathbb{F}^*$ and all $n\in\mathbf{N}$,*

$$\sigma(-a)=-\sigma(a), \tag{3.1}$$

$$\sigma(a_1\cdot a_2\cdots a_n)=\frac{\sigma(a_1)\cdot\sigma(a_2)\cdots\sigma(a_n)}{\sigma(1)^{n-1}}, \tag{3.2}$$

$$\sigma(a^n)=\frac{\sigma(a)^n}{\sigma(1)^{n-1}}, \tag{3.3}$$

$$\sigma(a^{-n})=\frac{\sigma(1)^{n+1}}{\sigma(a)^n}, \tag{3.4}$$

$$\sigma\left(\frac{a}{c}\right)=\frac{\sigma(1)\cdot\sigma(a)}{\sigma(c)}, \tag{3.5}$$

$$\sigma\left(\frac{a\cdot b}{c}\right)=\frac{\sigma(a)\cdot\sigma(b)}{\sigma(c)}. \tag{3.6}$$

The identity automorphism and the mapping $[x\mapsto -x]$ are examples of sesqui-morphisms known respectively as *symmetric* and *skew-symmetric* sesqui-morphisms. One can without difficulties prove that they are the only sesqui-morphisms in prime fields [91].

Let $\sigma:\mathbb{F}\to\mathbb{F}$ be a sesqui-morphism. A (V,V)-matrix M over $\mathbb{F}$ is said σ*-symmetric* if $m_{xy}=\sigma(m_{yx})$ for all $x,y\in V$. Let us now explain how to encode edge-coloured graphs as σ-symmetric matrices.

An $\mathbb{F}^*$*-graph* G is an $\mathbb{F}^*$-edge-coloured graph such that $\ell_G(x,y)\in\mathbb{F}^*$ for every arc $(x,y)\in E_G$. Every $\mathbb{F}^*$-graph can be trivially represented by a (V_G,V_G)-matrix M_G over $\mathbb{F}$ such that

$$M_G[x,y]:=\begin{cases}\ell_G(x,y) & \text{if } (x,y)\in E_G,\\ 0 & \text{otherwise.}\end{cases}$$

Let C be a fixed finite colour set and let us take an injection from $2^C\setminus\{\emptyset\}$ to $\mathbb{F}^*$ for some large enough field which is not algebraically closed, enabling the representation of any C-edge-coloured graph as an $\mathbb{F}^*$-graph. We proved in [104] that we can always turn the injection, in a canonical way, into a sesqui-morphism

$\sigma : \mathbb{F}^2 \to \mathbb{F}^2$, *i.e.*, every C-edge-coloured graph can be represented by a σ-symmetric matrix over some field $\mathbb{G}$. Even though this representation is not unique because it not only depends on the injection but also on the chosen field, as we will see the different rank-width parameters are equivalent.

From now on we can fix a finite field $\mathbb{F}$ and a sesqui-morphism $\sigma : \mathbb{F} \to \mathbb{F}$ to avoid to overload the text. Notice that even if the results are stated with respect to finite fields many are still valid on infinite fields. A σ-symmetric graph is an $\mathbb{F}^*$-graph G such that M_G is σ-symmetric.

Definition 3.2 ($\mathbb{F}$-rank-width)**.** *The* $\mathbb{F}$-cut-rank *function of a σ-symmetric graph G is the function* $\mathrm{cutrk}_G^{\mathbb{F}} : 2^{V_G} \to \mathbf{N}$ *where* $\mathrm{cutrk}_G^{\mathbb{F}}(X) = \mathrm{rk}(M_G[X, V_G \setminus X])$ *for all $X \subseteq V_G$. It is not hard to check that* $\mathrm{cutrk}_G^{\mathbb{F}}$ *is symmetric and submodular.*

The $\mathbb{F}$-rank-width *of G, denoted by* $\mathrm{rwd}^{\mathbb{F}}(G)$*, is the* $\mathrm{cutrk}_G^{\mathbb{F}}$*-width of V_G.*

Since any undirected graph is a σ_1-symmetric graph with σ_1 the symmetric function over $\mathbb{F}_2$, this definition generalises the notion of rank-width introduced and studied by Oum [130, 131, 135] and the objective was to extend results concerning the rank-width of undirected graphs to the $\mathbb{F}$-rank-width of σ-symmetric graphs. We first proved in [104] that clique-width and $\mathbb{F}$-rank-width are two equivalent measures. Since computing the rank-width of undirected graphs is NP-complete [87], the computation of the $\mathbb{F}$-rank-width is also NP-complete for all extensions $\mathbb{F}$ of $\mathbb{F}_2$. Observe however that the computation of the $\mathbb{F}$-rank-width is NP-complete, for any fixed field $\mathbb{F}$, by reducing the computation of the branch-width of graphs to it as follows.

1. Since tree-width cannot be approximated within a constant factor and branch-width and tree-width are linearly related [143], one can deduce that the computation of the branch-width of a graph cannot be approximated within a constant factor.
2. The branch-width of a graph and the branch-width of its cycle matroid are linearly related [123].
3. Since a cycle matroid is representable over any field and such a representation can be found in polynomial time, one can deduce then that the branch-width of representable matroids given with their representations, over any fixed field, cannot be also approximated within a constant factor.
4. As we can associate (in polynomial time) with every representable matroid M over a fixed field $\mathbb{F}$, given with its representation, a σ-symmetric bipartite graph such that its $\mathbb{F}$-rank-width is linearly related to the branch-width of M, we can deduce that $\mathbb{F}$-rank-width cannot be approximated within a constant factor.

But, the computation of the $\mathbb{F}$-rank-width is FPT with parameter the $\mathbb{F}$-rank-width (the proof is based on a coding of the $\mathbb{F}$-cut-rank function with the connectivity function of a partitioned matroid and then use the FPT algorithm in [86] for partitioned matroids).

Theorem 3.3 ([104])**.** *Let k be a fixed integer. There is an algorithm that takes as input a σ-symmetric graph G and in time $O(|V_G|^3)$ either outputs a layout of* $\mathrm{cutrk}_G^{\mathbb{F}}$*-width at most k or confirms that the $\mathbb{F}$-rank-width of G is at least $k+1$.*

The main advantage of rank-width over clique-width is that rank-width is preserved with respect to *vertex-minor* or *pivot-minor* relations. One would like to know whether such operations exist for σ-symmetric graphs. In [104] we proved that there are some fields, *e.g.*, $\mathbb{F}_3$, where a vertex-minor notion does not exist (at least as we defined it in [104]). The notion of *principal pivot transform* was introduced by Tucker [156] and generalises the notion of pivot-minor of graphs. We now explain how we adapted it to define a pivot-minor operation for σ-symmetric graphs.

Let M be a matrix of the form $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ where $A := M[X]$ is non-singular. The *Schur complement of A in M*, denoted by M/A, is $D - C \cdot A^{-1} \cdot B$, and the *principal pivot transform of M at X*, denoted by $M * X$, is the matrix

$$\begin{pmatrix} A^{-1} & A^{-1} \cdot B \\ -C \cdot A^{-1} & M/A \end{pmatrix}.$$

It was proved in [134, 156] that if an undirected graph H is a pivot-minor of an undirected graph G, then there exists $X \subseteq V_G$ such that M_H is a sub-matrix of $M_G * X$.

For a finite set V and for $X \subseteq V$, P_X and I_X are non-singular diagonal (V, V)-matrices where

$$P_X[x,x] := \begin{cases} \sigma(-1) & \text{if } x \in X, \\ 1 & \text{otherwise,} \end{cases} \qquad \text{and} \qquad I_X[x,x] := \begin{cases} -1 & \text{if } x \in X, \\ 1 & \text{otherwise.} \end{cases}$$

A pair (p,q) of non-zero scalars in $\mathbb{F}$ is said σ-compatible if $p^{-1} = \sigma(q)\cdot\sigma(1)^{-1}$ (equivalently $q^{-1} = \sigma(p)\cdot\sigma(1)^{-1}$). That means that (q,p) is also σ-compatible. It is worth noticing that if (p,q) is σ-compatible, then (p^{-1}, q^{-1}) is also σ-compatible. A pair (P,Q) of non-singular diagonal (V,V)-matrices is said σ-compatible if (p_{xx}, q_{xx}) is σ-compatible for all $x \in V$. For instance the pair (P_X, P_X^{-1}) is σ-compatible.

Definition 3.4 (Pivot-Minor). *A σ-symmetric graph H is* pivot-equivalent *to a σ-symmetric graph G if H is equal $I_Z \cdot P \cdot P_X \cdot (M * X) \cdot Q^{-1} \cdot I_{Z'}$ for some $X, Z, Z' \subseteq V_G$ and (P,Q) a pair of σ-compatible diagonal (V_G, V_G)-matrices, after possibly turning some diagonal entries into 0. And H is a* pivot-minor *of G if H is isomorphic to an induced subgraph of G' pivot-equivalent to G.*

***Proposition* 3.5** ([91, 104]). *If H is a pivot-minor of G, then* $\mathrm{rwd}^{\mathbb{F}}(H) \leq \mathrm{rwd}^{\mathbb{F}}(G)$.

Since the class of σ-symmetric graphs of $\mathbb{F}$-rank-width at most k is closed under pivot-minor one can wonder whether this set is characterised by a finite list of σ-symmetric graphs to exclude as pivot-minors. We answered positively in [104], which generalises the main results in [71, 130]. The proof techniques are the same in all three papers, and are based on the notion of titanic sets (particularly [86, Lemma 3.3]), how the connectivity function behaves after taking an elementary pivot-minor and the nice notion of (m,g)-connectedness introduced in [71] which says roughly that if $\mathrm{cutrk}_G^{\mathbb{F}}(X) = \ell < m$, then the size of X or $V_G \setminus X$ should be bounded by $g(\ell)$ with $g : \mathbf{N} \to \mathbf{N}$ and $m \in \mathbf{N}$.

Theorem 3.6 ([104]). *For each positive integer $k \geq 1$, there is a set $\mathscr{C}_k^{(\mathbb{F},\sigma)}$ of σ-symmetric graphs, each having at most $(6^{k+1}-1)/5$ vertices, such that a σ-symmetric graph G has $\mathbb{F}$-rank-width at most k if and only if no σ-symmetric graph in $\mathscr{C}_k^{(\mathbb{F},\sigma)}$ is isomorphic to a pivot-minor of G.*

With B. Courcelle we characterised algebraically in [39] the rank-width of undirected graphs which can be used to bypass the translation into a clique-width expression for solving $MSOL_1$-definable properties on graphs of bounded rank-width. We did similarly for σ-symmetric graphs in [104]. An $\mathbb{F}^k$-coloured σ-symmetric graph is a σ-symmetric graph G equipped with a colouring function $\gamma_G : V_G \to \mathbb{F}^k$.

Definition 3.7 (Bilinear Products). *Let k, ℓ and m be positive integers and let M, N and P be $k \times \ell$, $k \times m$ and $\ell \times m$ matrices, respectively, over $\mathbb{F}$. For an $\mathbb{F}^k$-coloured σ-symmetric graph G and an $\mathbb{F}^\ell$-coloured σ-symmetric graph H, we let $G \otimes_{M,N,P} H$ be the $\mathbb{F}^m$-coloured σ-symmetric graph $K := (V_G \cup V_H, E_G \cup E_H \cup E', \ell_K, \gamma_K)$ where:*

$$E' := \{xy \mid x \in V_G,\ y \in V_H \text{ and } \gamma_G(x)\cdot M \cdot \sigma(\gamma_H(y))^T \neq 0\},$$

$$\gamma_K(x) := \begin{cases} \gamma_G(x)\cdot N & \text{if } x \in V_G, \\ \gamma_H(x)\cdot P & \text{if } x \in V_H, \end{cases}$$

and for every $(x,y) \in E_K$

$$\ell_K((x,y)) := \begin{cases} \ell_G((x,y)) & \text{if } (x,y) \in E_G, \\ \ell_H((x,y)) & \text{if } (x,y) \in E_H, \\ \gamma_G(x)\cdot M \cdot \sigma(\gamma_H(y))^T & \text{if } x \in V_G,\ y \in V_H, \\ \sigma\left(\gamma_G(y)\cdot M \cdot \sigma(\gamma_H(x))^T\right) & \text{if } y \in V_G,\ x \in V_H. \end{cases}$$

Definition 3.8 (Constants). *For each $u \in \mathbb{F}^k$, we let $\mathbf{u}$ be a constant denoting an $\mathbb{F}^k$-coloured σ-symmetric graph with exactly one vertex and no edge; this unique vertex is coloured by u.*

We denote by $\mathcal{C}_n^{\mathbb{F}}$ the set $\{\mathbf{u} \mid u \in \mathbb{F}^1 \cup \cdots \cup \mathbb{F}^n\}$. We let $\mathcal{R}_n^{(\mathbb{F},\sigma)}$ be the set of bilinear products $\otimes_{M,N,P}$ where M, N and P are respectively $k\times\ell$, $k\times m$ and $\ell\times m$ matrices for $k,\ell,m \leq n$. $T(\mathcal{R}_n^{(\mathbb{F},\sigma)}, \mathcal{C}_n^{\mathbb{F}})$ is the set of finite well-formed terms built with symbols in $\mathcal{R}_n^{(\mathbb{F},\sigma)} \cup \mathcal{C}_n^{\mathbb{F}}$ and each term t in $T(\mathcal{R}_n^{(\mathbb{F},\sigma)}, \mathcal{C}_n^{\mathbb{F}})$ defines, up to isomorphism, a σ-symmetric graph $val(t)$.

Theorem 3.9 ([104])**.** *A σ-symmetric graph G has $\mathbb{F}$-rank-width at most k if and only if it is isomorphic to $val(t)$ for some term t in $T(\mathcal{R}_k^{(\mathbb{F},\sigma)}, \mathcal{C}_k^{\mathbb{F}})$.*

3.2 Well-Quasi-Ordering under Pivot-Minor

It was proved in [131] that undirected graphs of bounded rank-width are well-quasi-ordered by the pivot-minor relation and this result was extended later to (skew) symmetric matrices of bounded $\mathbb{F}$-rank-width [134]. We proved similarly (and in the same way as in [134]) that σ-symmetric graphs of bounded $\mathbb{F}$-rank-width are well-quasi-ordered by the pivot-minor relation. For that purposes we adapted the notion of *lagrangian chain groups* introduced in [134] that generalises the notion of *Tutte chain groups* [157] which are equivalent to representable matroids and studied for instance in [15, 16, 13]. The idea consists in embedding n-vertex σ-symmetric graphs into isotropic subspaces of dimension n of a vector space equipped with a kind of bilinear form, and then show that these isotropic subspaces can be well-quasi-ordered by an appropriate minor relation. Let us be more precise now.

We let $\mathbb{K}_\sigma$ be the 2-dimensional vector space $\mathbb{F}^2$ over $\mathbb{F}$ equipped with the application $\mathbf{b}_\sigma : \mathbb{K}_\sigma \times \mathbb{K}_\sigma \to \mathbb{F}$ where $\mathbf{b}_\sigma(\binom{a}{b}, \binom{c}{d}) = \sigma(1)\cdot a\cdot\sigma(d) - b\cdot\sigma(c)$. The application $\mathbf{b}_\sigma$ is not bilinear, however it is linear with respect to its left operand, which is enough for our purposes. Notice that if σ is skew-symmetric (or symmetric), then $\mathbf{b}_\sigma$ is what is called b^+ (or b^-) in [134]. The following properties are easy to obtain from the definition of $\mathbf{b}_\sigma$.

***Property* 3.10.** *Let $u, v, w \in \mathbb{K}_\sigma$ and $k \in \mathbb{F}$. Then,*

$$\begin{aligned}
\mathbf{b}_\sigma(u+v, w) &= \mathbf{b}_\sigma(u,w) + \mathbf{b}_\sigma(v,w),\\
\mathbf{b}_\sigma(u, v+w) &= \mathbf{b}_\sigma(u,v) + \mathbf{b}_\sigma(u,w),\\
\mathbf{b}_\sigma(k\cdot u, v) &= k\cdot \mathbf{b}_\sigma(u,v),\\
\mathbf{b}_\sigma(u, k\cdot v) &= \tilde{\sigma}(k)\cdot \mathbf{b}_\sigma(u,v).\\
\sigma(\mathbf{b}_\sigma(u,v)) &= \frac{-1}{\sigma(1)^2}\cdot \mathbf{b}_\sigma(v,u).
\end{aligned}$$

***Property* 3.11.** *Let $u \in \mathbb{K}_\sigma$.*

(i) If $\mathbf{b}_\sigma(u,v) = 0$ for all $v \in \mathbb{K}_\sigma$, then $u = 0$.

(ii) If $\mathbf{b}_\sigma(v,u) = 0$ for all $v \in \mathbb{K}_\sigma$, then $u = 0$.

Let V be a finite set. A *$\mathbb{K}_\sigma$-chain on V* is a function $f : V \to \mathbb{K}_\sigma$. We let $\mathbb{K}_\sigma^V$ be the set of $\mathbb{K}_\sigma$-chains on V. It is well-known that $\mathbb{K}_\sigma^V$ is a vector space over $\mathbb{F}$ by letting $(f+g)(x) := f(x) + g(x)$ and $(k\cdot f)(x) := k\cdot f(x)$ for all $x \in V$ and $k \in \mathbb{F}$, and by setting the $\mathbb{K}_\sigma$-chain $[x \mapsto 0]$ as the zero vector. It is worth noticing that $\dim(\mathbb{K}_\sigma^V) = 2\cdot|V|$. We let $\langle,\rangle : \mathbb{K}_\sigma^V \times \mathbb{K}_\sigma^V \to \mathbb{F}$ be such that for all $f, g \in \mathbb{K}_\sigma^V$,

$$\langle f, g\rangle := \sum_{x\in V} \mathbf{b}_\sigma(f(x), g(x)).$$

A vector u of $\mathbb{K}_\sigma^V$ is said *isotropic* if $\mathbf{b}_\sigma(u,u) = 0$. A subspace L of $\mathbb{K}_\sigma^V$ is called *totally isotropic* if $\mathbf{b}_\sigma(u,v) = 0$ for all $u, v \in L$. For a subspace L of $\mathbb{K}_\sigma^V$, we let $L^\perp := \{v \in \mathbb{K}_\sigma^V \mid \mathbf{b}_\sigma(u,v) = 0 \text{ for all } u \in L\}$. It is worth noticing that if L is totally isotropic, then $L \subseteq L^\perp$. The following theorem is a well-known theorem in the case where $\mathbf{b}_\sigma$ is a non-degenerate bilinear form.

Theorem 3.12 ([91])**.** $\dim(L) + \dim(L^\perp) = \dim(\mathbb{K}_\sigma)$ *for any subspace L of $\mathbb{K}_\sigma^V$.*

It is straightforward to verify that $\langle,\rangle$ satisfies Properties 3.10 and 3.11. Subspaces of $\mathbb{K}_\sigma^V$ are called $\mathbb{K}_\sigma$-*chain groups on V*. A $\mathbb{K}_\sigma$-chain group L on V is said *lagrangian* if it is totally isotropic and $\dim(L)=|V|$.

A *simple isomorphism* from a $\mathbb{K}_\sigma$-chain group L on V to a $\mathbb{K}_\sigma$-chain group L' on V' is a bijection $\mu: V \to V'$ such that $L = \{f \circ \mu \mid f \in L'\}$ where $(f\circ\mu)(x) = f(\mu(x))$ for all $x \in V$. In this case we say that L and L' are *simply isomorphic*.

Let us introduce *minors* for $\mathbb{K}_\sigma$-chain groups on V. For $L \subseteq \mathbb{K}_\sigma^V$, $\alpha \in \mathbb{K}_\sigma^*$ and $X \subseteq V$, we let $L \|_\alpha X$ be the $\mathbb{K}_\sigma$-chain group

$$L \|_\alpha X := \{f|_{(V\setminus X)} \mid f \in L \text{ and } \mathbf{b}_\sigma(f(x),\alpha) = 0 \text{ for all } x \in X\}$$

on $V \setminus X$. A pair $\{\alpha,\beta\} \subseteq \mathbb{K}_\sigma^*$ is said *minor-compatible* if $\mathbf{b}_\sigma(\alpha,\alpha) = \mathbf{b}_\sigma(\beta,\beta) = 0$ and $\{\alpha,\beta\}$ forms a basis for $\mathbb{K}_\sigma$. For a minor-compatible pair $\{\alpha,\beta\}$, a $\mathbb{K}_\sigma$-chain group on $V \setminus (X \cup Y)$ of the form $L \|_\alpha X \|_\beta Y$ is called an *$\alpha\beta$-minor* of L.

One easily verifies that $L \|_\alpha X \|_\alpha Y = L \|_\alpha (X \cup Y)$, and $L \|_\alpha X \|_\beta Y = L \|_\beta Y \|_\alpha X$. As a consequence an $\alpha\beta$-minor of an $\alpha\beta$-minor of a lagrangian $\mathbb{K}_\sigma$-chain group L is an $\alpha\beta$-minor of L [91].

We now define the connectivity function for lagrangian $\mathbb{K}_\sigma$-chain groups. Let L be a lagrangian $\mathbb{K}_\sigma$-chain group on V. For every $X \subseteq V$, we let

$$\lambda_L(X) := |X| - \dim(L^{|X})$$

where $Sp(f) := \{x \in V \mid f(x) \neq 0\}$ and $L^{|X} := \{f|_X \mid f \in L \text{ and } Sp(f) \subseteq X\}$. Since $L^{|X}$ is totally isotropic, $\dim(L^{|X}) \leq |X|$, and hence $\lambda_L(X) \geq 0$. It is not hard to prove that λ_L is symmetric and submodular [91, 134]. The *branch-width* of L is defined as the λ_L-width of V.

By adapting the proof ideas in [134] we were able to prove the following which extends a similar one in [134].

Theorem 3.13 ([91]). *Let k a positive integer, and let $\{\alpha,\beta\}$ be minor-compatible. Let $L_1, L_2, \ldots$ be an infinite sequence of lagrangian $\mathbb{K}_{\sigma_i}$-chain groups having branch-width at most k. Then, there exist $i<j$ such that L_i is simply isomorphic to an $\alpha\beta$-minor of L_j.*

It remains now to explain how embed σ-symmetric graphs into lagrangian $\mathbb{K}_\sigma$-chain groups. Two $\mathbb{K}_\sigma$-chains f and g on V are *supplementary* if, for all $x \in V$,

(i) $\mathbf{b}_\sigma(f(x), f(x)) = \mathbf{b}_\sigma(g(x), g(x)) = 0$,

(ii) $\mathbf{b}_\sigma(f(x), g(x)) = \sigma(1)$ and

(iii) $\mathbf{b}_\sigma(g(x), f(x)) = -\sigma(1)^2$.

Supplementary $\mathbb{K}_\sigma$-chains on V do exist.

***Property* 3.14** ([91]). *For any $c \in \mathbb{F}^*$, we have*

$$\begin{cases} \mathbf{b}_\sigma\left(\binom{c}{0}, \binom{0}{\sigma(c^{-1})}\right) & = \sigma(1) \\ \mathbf{b}_\sigma\left(\binom{0}{\sigma(c^{-1})}, \binom{c}{0}\right) & = -\sigma(1)^2 \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{b}_\sigma\left(\binom{0}{c}, \binom{-\sigma(1)\cdot\sigma(c)^{-1}}{0}\right) & = \sigma(1) \\ \mathbf{b}_\sigma\left(\binom{-\sigma(1)\cdot\sigma(c)^{-1}}{0}, \binom{0}{c}\right) & = -\sigma(1)^2 \end{cases}$$

We associated with every σ-symmetric graph a lagrangian $\mathbb{K}_\sigma$-chain group.

***Proposition* 3.15** ([91]). *Let G be a σ-symmetric graph, and let f and g be supplementary $\mathbb{K}_\sigma$-chains on V_G. For every $x \in V_G$, we let f_x be the $\mathbb{K}_\sigma$-chain on V_G such that, for all $y \in V_G$,*

$$f_x(y) := \begin{cases} m_{xx}\cdot f(x) + g(x) & \text{if } y = x, \\ m_{xy}\cdot f(y) & \text{otherwise.} \end{cases}$$

Then, the span of $\{f_x \mid x \in V_G\}$ is a lagrangian $\mathbb{K}_\sigma$-chain group on V_G. (We will denote it by (M_G, f, g).)

We call (M_G, f, g) a *matrix representation* of any lagrangian $\mathbb{K}_\sigma$-chain group L simply isomorphic to (M_G, f, g). A $\mathbb{K}_\sigma$-chain f on V is called an *eulerian chain* of a lagrangian $\mathbb{K}_\sigma$-chain group L on V if:

(i) for all $x \in V$, $f(x) \neq 0$ and $\mathbf{b}_\sigma(f(x), f(x)) = 0$, and

(ii) there is no non-zero $\mathbb{K}_\sigma$-chain h in L such that $\mathbf{b}_\sigma(h(x), f(x)) = 0$ for all $x \in V$.

***Proposition* 3.16** ([91, 134])**.** *Every lagrangian $\mathbb{K}_\sigma$-chain group on V has an eulerian chain.*

Every lagrangian $\mathbb{K}_\sigma$-chain group admits a matrix representation.

***Proposition* 3.17** ([91])**.** *Let L be a lagrangian $\mathbb{K}_\sigma$-chain group on V. Let f and g be supplementary with f being an eulerian chain of L. For every $x \in V$, there exists a unique $\mathbb{K}_\sigma$-chain $f_x \in L$ such that*

(i) $\mathbf{b}_\sigma(f(y), f_x(y)) = 0$ *for all* $y \in V \setminus x$,

(ii) $\mathbf{b}_\sigma(f(x), f_x(x)) = \sigma(1)$.

Moreover, $\{f_x \mid x \in V\}$ is a basis for L. If we let M be the (V,V)-matrix such that $m_{xy} := \mathbf{b}_\sigma(f_x(y), g(y)) \cdot \sigma(1)^{-1}$, then M is σ-symmetric and (M, f, g) is a matrix representation of L.

We finally related the branch-width of lagrangian $\mathbb{K}_\sigma$-chain groups with the $\mathbb{F}$-rank-width of their matrix representations, and also pivot-minors of σ-symmetric graphs with the $\alpha\beta$-minors of associated lagrangian $\mathbb{K}_\sigma$-chain groups.

***Proposition* 3.18** ([91, 134])**.** *Let (M_G, f, g) be a matrix representation of a lagrangian $\mathbb{K}_\sigma$-chain group L on V_G. Then the branch-width of L is equal to the $\mathbb{F}$-rank-width of G.*

***Proposition* 3.19** ([91])**.** *Let $\{\alpha, \beta\}$ be minor-compatible. Let L and L' be lagrangian $\mathbb{K}_\sigma$-chain groups on V and V' respectively. Let (M, f, g) and (M', f', g') be special matrix representations of L and L' respectively with $f(x) := \pm\alpha$, $g(x) := \beta$ for all $x \in V$, and $f'(x) := \pm\alpha$, $g'(x) := \beta$ for all $x \in V'$. If $L' = L \|_\beta X \|_\alpha Y$, then $M' = \big((M/M[A])[V']\big) \cdot I_Z$ with $A \subseteq X$ and $Z := \{x \in V' \mid f'(x) = -f(x)\}$.*

By combining all the previous propositions, we were able to prove the following.

Theorem 3.20 ([91])**.** *Let k be a positive integer. For every infinite sequence $G_1, G_2, \ldots$ of σ_i-symmetric graphs of $\mathbb{F}$-rank-width at most k, there exist $i < j$ such that G_i is isomorphic to a pivot-minor of G_j.*

3.3 $\mathbb{F}$-split Decompositions

It was proved in [129] that the rank-width of a graph is equal to the maximum rank-width over all its prime induced subgraphs with respect to split decomposition [45, 46]. It is also known that distance-hereditary graphs are exactly graphs of rank-width at most 1 [130]. We defined in [103] the analogous of split decomposition for σ-symmetric graphs and characterised also σ-symmetric graphs of $\mathbb{F}$-rank-width 1 [103]. For better readability, the results were presented in [103] only for directed graphs, but we will give them here in terms of σ-symmetric graphs.

Two bipartitions $\{X_1, X_2\}$ and $\{Y_1, Y_2\}$ of a set V *overlap* if $X_i \cap Y_j \neq \emptyset$ for every $i, j \in \{1,2\}$.

Definition 3.21 (Bi-Partitive Family)**.** *Let V be a finite set and let $\mathcal{F}$ be a family of bipartitions of V. Then $\mathcal{F}$ is* bi-partitive *if:*

- $\{\emptyset, V\} \notin \mathcal{F}$,
- *for all* $v \in V$, $\{\{v\}, V \setminus \{v\}\} \in \mathcal{F}$ *and*
- *for all* $\{X_1, X_2\} \in \mathcal{F}$ *and* $\{Y_1, Y_2\} \in \mathcal{F}$ *such that* $\{X_1, X_2\}$ *and* $\{Y_1, Y_2\}$ *overlap, then* $\{X_i \cap Y_j, V \setminus (X_i \cap Y_j)\} \in \mathcal{F}$, *for every* $i, j \in \{1,2\}$.

A member $\{X_1, X_2\}$ of a bi-partitive family $\mathcal{F}$ is trivial *if $|X_1| \leq 1$ or $|X_2| \leq 1$, and is* strong *if there is no $\{Y_1, Y_2\} \in \mathcal{F}$ such that $\{X_1, X_2\}$ and $\{Y_1, Y_2\}$ overlap.*

Bi-partitive families were studied in [46] and are similar to partitive families [24, 124]. It was for instance proved in [46] that splits in strongly connected graphs form a bi-partitive family. Examples of partitive families are modules in graphs [24].

***Proposition* 3.22** (Folklore [103]). *Let $f : 2^V \to \mathbb{N}$ be a symmetric and sub-modular function and let $m = \min_{\emptyset \subsetneq X \subsetneq V} f(X)$. Then the family of minimums $\mathcal{F} := \{\{X, V \setminus X\} \mid f(X) = m\}$ is bi-partitive.*

It is proved in [46] that for every bi-partitive family $\mathcal{F}$ on V there is, up to isomorphism, a unique layout $(T, \mathcal{L})$ of V such that for every edge e of T, the bipartitition $\{X^e, V \setminus X^e\}$ is a strong bipartititon in $\mathcal{F}$. We call this layout the *canonical decomposition* of $\mathcal{F}$.

Definition 3.23 ($\mathbb{F}$-split). *Let G be a σ-symmetric graph. A bipartition $\{X, Y\}$ of V_G is an* $\mathbb{F}$-split *if $X \neq \emptyset$, $Y \neq \emptyset$ and* $\mathrm{cutrk}_G^{\mathbb{F}}(X) \leq 1$.

A σ-symmetric graph G is said *prime* if every $\mathbb{F}$-split in G is trivial. A σ-symmetric graph is *degenerate* if every bipartititon is an $\mathbb{F}$-split, and it is *linear* if it admits an ordering $x_1, \ldots, x_n$ such that $\{\{x_i, \ldots, x_j\}, V_G \setminus \{x_i, \ldots, x_j\}\}$ is an $\mathbb{F}$-split for all $1 \leq i \leq j < n$. This notion of $\mathbb{F}$-split generalises the notion of split in undirected graphs [46]. Moreover, every $\mathbb{F}$-split $\{X, Y\}$ in G is a split in $u(G)$. By Proposition 3.22 the set of $\mathbb{F}$-splits in a σ-symmetric graph G forms a bi-partitive family, and we proved in [103] that the tree structure can be constructed in polynomial time.

Theorem 3.24 ([103]). *The canonical decomposition of the $\mathbb{F}$-splits of every n-vertex σ-symmetric graph with m edges, called $\mathbb{F}$-split decomposition, can be constructed in time $O(nm)$.*

Let us now relate the $\mathbb{F}$-rank-width of a σ-symmetric graph with the $\mathbb{F}$-rank-width of its induced prime graphs. Let $(T, \mathcal{L})$ be the $\mathbb{F}$-split decomposition of a σ-symmetric graph G and let u be an internal node of T. Recall that $\{X^{uv} \mid v \in N_T(u)\}$ is a partition of V_G, and for each $v \in N_T(u)$, the bipartition $\{X^{uv}, V_G \setminus X^{uv}\}$ is a strong $\mathbb{F}$-split.

For every node $v \in N_T(u)$, we choose a vertex x_v^u in X^{uv} that is adjacent to a vertex in $V_G \setminus X^{uv}$, which always exists because $u(G)$ is connected. We let $\mathrm{b}(u)$ be the σ-symmetric graph $G[\{x_v^u \mid v \in N_T(u)\}]$. Notice that $\mathrm{b}(u)$ is not unique and depends on the choice of the representatives. We proved the following, which is a generalisation of a similar result in [129].

Theorem 3.25 ([103]). *Let G be a connected σ-symmetric graph with at least 3 vertices and let $(T, \mathcal{L})$ be the $\mathbb{F}$-split decomposition of G. Then,* $\mathrm{rwd}^{\mathbb{F}}(G) = \max\{\mathrm{rwd}^{\mathbb{F}}(b(u)) \mid u \in V_T \setminus L_T\}$.

We also characterised the σ-symmetric graphs of $\mathbb{F}$-rank-width 1 as special orientations of distance-hereditary graphs. A vertex x is a *pendant* vertex of another vertex y if y is the unique neighbour of x. Two vertices x and y are *dtwins* if there is some constant $c \in \mathbb{F}^*$ such that $M_{G\setminus\{x,y\}}[y,z] = c \cdot M_{G\setminus\{x,y\}}[x,z]$ for all $z \in V_G \setminus \{x, y\}$. A σ-symmetric graph G is *completely decomposable* if for every node u of its $\mathbb{F}$-split decomposition the σ-symmetric graph $\mathrm{b}(u)$ is either degenerate or linear.

Theorem 3.26 ([103]). *Let G be a connected σ-symmetric graph with at least 2 vertices. Then the following are equivalent.*

1. *G is completely decomposable by the $\mathbb{F}$-split decomposition.*
2. *G can be obtained from a single vertex by creating dtwins or adding pendant vertices.*
3. *G has $\mathbb{F}$-rank-width 1.*
4. *For every $W \subseteq V$ with $|W| \geq 4$, $G[W]$ has a non-trivial $\mathbb{F}$-split.*
5. *$u(G)$ is distance-hereditary and for every $W \subseteq V$ with $|W| \leq 5$,* $\mathrm{rwd}^{\mathbb{F}}(G[W]) \leq 1$.

3.4 Concluding Remarks

In the series of papers [91, 103, 104] we extended the notion of rank-width of undirected graphs to the C-edge-coloured graphs and extended all the results proved by Oum in the series of papers [86, 130, 131]. We defined the notion of $\mathbb{F}$-rank-width and pivot-minor, showed that graphs of bounded $\mathbb{F}$-rank-width are characterised by a finite list of graphs to exclude as pivot-minors (with an upper bound on the sizes of obstructions), and showed that indeed graphs of bounded $\mathbb{F}$-rank-width are well-quasi-ordered by the pivot-minor relation. We also extended the notion of split decomposition to C-edge-coloured graphs and defined algebraic graph operations that generalised the ones defined in [39] and that characterised exactly $\mathbb{F}$-rank-width.

In [134] links between skew-symmetric matrices and representable matroids are recalled and Oum proved in particular that an $\mathbb{F}$-representable matroid N is a minor of an $\mathbb{F}$-representable matroid M whenever the associated skew-symmetric matrix of N is a pivot-minor of the associated skew-symmetric matrix of M. It is announced in [73] that $\mathbb{F}$-representable matroids are well-quasi-ordered by the matroid minor relation, generalising Graph Minor Theorem. A more general theorem would be a positive answer to the following conjecture.

Conjecture 3.27. *Let $\mathbb{F}$ be a finite field. For every infinite sequence $M_1, M_2, \ldots$ of σ_i-symmetric matrices, there exist $i < j$ such that M_i is isomorphic to a pivot-minor of M_j.*

Because the theory of σ-symmetric matrices enables to define a robust rank-width notion for directed graphs, called in [91, 103, 104] $\mathbb{F}_4$*-rank-width* or $GF(4)$*-rank-width*, a positive answer to Conjecture 3.27 would open a strong theory for directed graphs similar to the Graph Minor theory. Notice that the announced proof in [73] already yields a positive answer to Conjecture 3.27 in the case of σ-symmetric bipartite graphs (see [93, Section 5] for the links between skew-symmetric bipartite graphs and representable matroids). In the case of oriented graphs, the notion of $\mathbb{F}_3$*-rank-width* is also defined in [104] and behaves similarly as for $\mathbb{F}_4$-rank-width, except we always work on the field $\mathbb{F}_3$ which maintains the obtained graphs oriented.

We did not discuss about the computational complexity of checking whether a σ-symmetric graph H is a pivot-minor of another σ-symmetric graph G (PIVOT-MINOR TESTING). Indeed, it is NP-complete since the GRAPH MINOR TESTING problem is already known to be NP-complete and can be reduced to the PIVOT-MINOR TESTING problem. Notice that checking whether a fixed $\mathbb{F}$-representable matroid N is a matroid minor of an $\mathbb{F}$-representable matroid M is also announced to be polynomial [73] (the case of graph minor is already settled in the seminal Graph Minor XIII [140]). Even in the case of undirected graphs the case of pivot-minor is open.

In his works on circle graphs Bouchet developed the notion of $\triangle$-matroids (see for instance [19, 21, 20, 14]) and proved in particular that the non-singular minors of a (skew-) symmetric matrix forms a $\triangle$-matroid, and he called such $\triangle$-matroids as *representable*. In [91] we observed that this is also the case for σ-symmetric matrices. This leaves open the question of finding the good notion of representability for $\triangle$-matroids. We notice that our notion of lagrangian chain groups generalise the notion of chain groups developed in [157] for representable matroids and in [21, 134] for representable $\triangle$-matroids. I am interested in investigating more the notion of $\triangle$-matroids, and in particular filling the gap with matroids: a notion of *connectivity function* that would be the same, in the case of representable $\triangle$-matroids, as the $\mathbb{F}$-rank-width of σ-symmetric matrices.

We also point out that we also defined in [104] a second notion of rank-width for C-edge-coloured graphs, called $\mathbb{F}$*-bi-rank-width*, that unfortunately does not behave well on the structural point of view. Indeed, we do not have an operation similar to the pivot-minor operation for this notion. Nonetheless, its specialisation to directed graphs, known as *bi-rank-width* were popularised by the works by Ganian et al. who showed that many problems are FPT when parametrised by bi-rank-width [70, 69]. Let us just observe that all they did with bi-rank-width is still valid with $\mathbb{F}_4$-rank-width which we think has the more potential of giving small constants, and has more desirable structural properties.

We finally observe that the proof techniques used in [91, 103, 104], the series of papers by Geelen et al. concerning the branch-width of finitely representable matroids, and the series of papers by Oum [130, 131] are more or less the same. This probably reveals that if we can solve the conjecture on undirected graphs (via isotropic systems), except if it relies heavily on the field $\mathbb{F}_2$, one would, probably, be able with some more technical efforts to adapt the proof in the case of σ-symmetric graphs. And this postulate motivates mostly why I studied only the linear rank-width of undirected graphs.

Chapter 4

Linear Rank-Width

In this chapter we deal only with undirected graphs and it constitutes a summary of the papers [1, 3, 4]. An extended abstract of [3, 4] appears in the proceedings of WG'14. Failing in proving that trees are obstructions for the linear rank-width of graphs, we investigated the computation of linear rank-width in graphs of bounded rank-width with the hope that it can help in understanding graphs of bounded linear rank-width. We were able to give a quasi-linear time algorithm for trees and a polynomial time algorithm for distance-hereditary graphs. A polynomial time algorithm for graphs of bounded rank-width is still open. The two algorithms are based on a characterisation of linear rank-width on trees and distance-hereditary graphs. Before starting, let us recall the following alternative definition of linear f-widths.

***Lemma* 4.1** (Folklore)**.** *Let $f : 2^V \to \mathbf{N}$ be a symmetric function and let $k \geq \max\limits_{1\leq i\leq n}\{f(v_i)\}$. Then V has linear f-width k if and only if V admits a linear ordering $v_1,\ldots,v_n$ such that $k = \max\{f(\{v_1,\ldots,v_i\}) \mid 1 \leq i \leq n-1\}$.*

The *linear rank-width* of a graph G, denoted by $\mathrm{lrwd}(G)$, is the linear $\mathbb{F}_2$-rank-width of G. We refer to [141] for the definition of *path-width* and to [61] for the definition of *linear clique-width*. We denote respectively by $\mathrm{pwd}(G)$ and $\mathrm{lcwd}(G)$ the path-width and linear clique-width of a graph G.

4.1 Computing the Linear Rank-Width of Distance-Hereditary Graphs

It is known from [132] that the rank-width of any graph is at most its tree-width plus one. One can without difficulties prove a similar bound between path-width and linear rank-width.

***Proposition* 4.2** ([1])**.** $\mathrm{lrwd}(G) \leq \mathrm{pwd}(G)$ *for every graph G.*

Surprisingly, in collaboration with I. Adler we were able to prove that its converse is true in forests and the proof was constructive because we showed the following using the characterisation of path-width by the cops and invisible robber game [51, 110].

***Proposition* 4.3** ([1])**.** *Any linear layout of* $\mathrm{cutrk}^{\mathbb{F}_2}$*-width k of an n-vertex tree can be transformed in time $O(n^2 \cdot \log(n)^2)$ into a path-decomposition of width at most k.*

A consequence of these two propositions is that we can compute in linear time the linear rank-width of any forest since the path-width of any forest can be computed in linear time [58] and were the first non trivial graph class with an algorithm computing its linear rank-width.

The equality $\mathrm{lrwd}(F) = \mathrm{pwd}(F)$ for any forest F implies the following characterisation of the linear rank-width of trees.

***Proposition* 4.4** ([1, 58])**.** *Let $k \geq 1$. A tree T has linear rank-width at most k if and only if for any node u of T, at most two connected components of $T \setminus u$ have linear rank-width k, and every other connected component of $T \setminus u$ has linear rank-width at most $k-1$.*

With I. Adler and O-J. Kwon we were able to obtain a similar characterisation for distance-hereditary graphs by using the $\mathbb{F}_2$-split decomposition of distance-hereditary graphs [3]. Let us give some definitions before presenting the characterisation and its consequences. Let us first recall that a graph is distance-hereditary if and only if for each node u of its $\mathbb{F}_2$-split decomposition, the graph $\mathsf{b}(u)$ is either a clique or a star [13].

Let G be a graph and let $(T,\mathcal{L})$ be the $\mathbb{F}_2$-split decomposition of G. We associate with it the graph D_G, called *canonical decomposition*, such that

$$V_{D_G} := \bigcup_{u\in V_T\setminus L_T} V_{\mathsf{b}(u)},$$
$$E_{D_G} := \bigcup_{u\in V_T\setminus L_T} E_{\mathsf{b}(u)} \cup \{x_v^u x_u^v \mid uv\in E_T\}.$$

One easily verifies that the set of edges of D_G not in a graph $\mathsf{b}(u)$ forms a matching, are exactly the edges of T and are called *marked edges*. Vertices adjacent to marked edges are called *marked vertices* and others *unmarked vertices*. For each $u\in N_T\setminus L_T$ we call $\mathsf{b}(u)$ a *bag*. A vertex v of D_G *represents* an unmarked vertex x (or is a *representative* of x) if either $v=x$ or there is a path of even length from v to x in D_G starting with a marked edge such that marked edges and unmarked edges appear alternately in the path. Two unmarked vertices x and y are *linked* in D_G if there is a path from x to y in D_G such that unmarked edges and marked edges appear alternately in the path.

A *local complementation* at an unmarked vertex x in D_G, denoted by $D_G * x$, is the operation which replaces each bag B containing a representative w of x with $B*w$. Let x and y be linked unmarked vertices in D_G, and let P be the alternating path in D_G linking x and y. The *pivoting on xy of D_G*, denoted by $D_G\wedge xy$, is the graph obtained as follows: for each bag B containing an unmarked edge of P, if $v,w\in V_B$ represent respectively x and y in D_G, then we replace B with $B\wedge vw$.

***Lemma* 4.5** ([13, 3])**.** *$D_G * x$ is the canonical decomposition of $G*x$. Similarly, $D_G\wedge xy$ is the canonical decomposition of $G\wedge xy$.*

From Lemma 4.5 G and its vertex-minors share the same $\mathbb{F}_2$-split decomposition $(T,\mathcal{L})$. If G' is locally equivalent (or pivot equivalent) to G, we will by abuse say similarly that $D_{G'}$ is locally equivalent (or pivot equivalent) to D_G. In order to avoid confusions, for each node u of $N_T\setminus L_T$ and each canonical decomposition $D_{G'}$ locally equivalent to D_G, we will write $\mathsf{b}_{D_{G'}}(u)$ to denote the graph associated with the node u in the graph G'.

From now on let us fix a distance-hereditary graph G and let $(T,\mathcal{L})$ and D be respectively its $\mathbb{F}_2$-split decomposition and its canonical decomposition. For an unmarked vertex y in D and a bag B of D containing a marked vertex that represents y, let C be the component of $D\setminus V_B$ containing y, and let w be the marked vertex of B adjacent to a vertex of C, and let v be this neighbour. We define the *limb* $\mathcal{L} := \mathcal{L}_D[B,y]$ with respect to B and y as follows:

1. if B is a clique, then $\mathcal{L} := C*v\setminus v$,
2. if B is a star and w is a leaf, then $\mathcal{L} := C\setminus v$,
3. if B is a star and w is the center, then $\mathcal{L} := C\wedge vy\setminus v$.

Since v becomes an unmarked vertex in C, the limb is well-defined. But, while C is a canonical decomposition, $\mathcal{L}$ may not be a canonical decomposition at all, because deleting v may create a bag of size 2. It is explained in [3] how to obtain a canonical decomposition from $\mathcal{L}$, and let $\mathcal{LC}_D[B,y]$ be the canonical decomposition obtained from $\mathcal{L}_D[B,y]$ and we call it the *canonical limb* and let $\mathcal{LG}_D[B,y]$ be the graph which has $\mathcal{LC}_D[B,y]$ as canonical decomposition. The following tells us that the choice of the vertex y is not important with respect to linear rank-width. And also taking D or a canonical decomposition locally equivalent to D' does not matter.

***Proposition* 4.6** ([3])**.** *Let u be a node of T.*

1. *If an unmarked vertex y of D is represented by a marked vertex of $\mathsf{b}_D(u)$, then $\mathcal{LC}_D[\mathsf{b}_D(u),y]$ is connected.*

2. *If two unmarked vertices x and y are both represented by a same marked vertex in $b_D(u)$, then $\mathcal{LC}_D[b_D(u),x]$ is locally equivalent to $\mathcal{LC}_D[b_D(u),y]$.*
3. *If D' is locally equivalent to D, and x and y in the same connected component of $D \setminus V_{b_D(u)}$ are represented in D and D' respectively by a same marked vertex in $b_D(u)$, then $\mathcal{LC}_D[b_D(u),x]$ is locally equivalent to $\mathcal{LC}_{D'}[b_{D'}(u),y]$.*

For a bag B of D and a component C of $D \setminus V_B$, we define $f_D(B,C)$ as the linear rank-width of $\mathcal{LG}_D[B,y]$ for some unmarked vertex $y \in V_C$. By Proposition 4.6, $f_D(B,C)$ does not depend on the choice of y neither on the choice of the canonical decomposition. We proved the following recursive characterisation which extends Proposition 4.4.

Theorem 4.7 ([3])**.** *Let $k \geq 1$ and let D be the canonical decomposition of a connected distance-hereditary graph G. Then $\mathrm{lrwd}(G) \leq k$ if and only if for each bag B of D, D has at most two components C of $D \setminus V_B$ such that $f_D(B,C) = k$, and for every other component C' of $D \setminus V_B$, $f_D(B,C') \leq k-1$.*

We can now explain the ideas of the algorithm which follows, in a non straightforward way, the same lines as the one for trees given in [58]. Let us first explain the one for trees. We first root the tree F into a node, and from bottom-up we compute for each internal node u the linear rank-width of the tree $F(u)$ rooted at u. Let $k := \max\{\mathrm{lrwd}(F(v)) \mid v \text{ child of } u\}$. If there is a node v, called *k-critical node*, that is a descendant of u and such that v has two children v_1 and v_2 such that $\mathrm{lrwd}(F(v)) = \mathrm{lrwd}(F(v_1)) = \mathrm{lrwd}(F(v_2))$, then by Proposition 4.4 in order to decide the linear rank-width of $F(u)$ we need to know the linear rank-width of $F(u) \setminus V_{F(v)}$. We can recursively call the algorithm on $F(u) \setminus V_{F(v)}$, but this would not give a linear time algorithm, and similar situations can happen in $F(u) \setminus V_{F(v)}$. The idea introduced in [58] to cope with this difficulty was to keep in u the linear rank-width of the subtrees that may cause a recursive call to the algorithm because of the presence of ℓ-critical nodes for $\ell \leq k$. For instance, in $F' := F(u) \setminus V_{F(v)}$ we may have a k'-critical node w with $k' := \max\{\mathrm{lrwd}(F'(v)) \mid v \text{ child of } u \text{ in } F'\}$, then we may need the linear rank-width of $F' \setminus V_{F'(w)}$ to answer, and so on.

In the case of a distance-hereditary graph G, we still start by rooting the $\mathbb{F}_2$-split decomposition and by Theorem 4.7 for each node u of $V_T \setminus L_T$ with parent v, we still need to compute $f_D(\mathsf{b}_D(v),C)$ where C is the component of $D \setminus \mathsf{b}_D(v)$ containing $\mathsf{b}_D(u)$. Now, if w is a k-critical node in $T(u)$, as in the case of trees we need to compute $f_D(\mathsf{b}_D(w),C'])$ where C' is the component of $D \setminus \mathsf{b}_D(w)$ containing the bag $\mathsf{b}_D(w')$ with w' the parent of w. However, contrary to the case of trees, the canonical limb $\mathcal{LC}_D[\mathsf{b}_D(w),y]$, for some unmarked vertex y in C', is not necessarily an induced subgraph of D_G. We overcame this difficulty by showing that the order in which we can recursively compute canonical limbs is not important, which enabled us to store information similar to the cases of trees. We needed however to compute limbs, which explained the time complexity in the following.

Theorem 4.8 ([3])**.** *The linear rank-width of every distance-hereditary graph with n vertices can be computed in time $\mathcal{O}(n^2 \cdot \log(n))$. Moreover, a linear layout of the graph witnessing the linear rank-width can be computed with the same time complexity.*

By using the characterisation by Dharmatilake [52] of matroids of branch-width 2 and the links between branch-width of binary matroids and rank-width of undirected graphs in [130] we were able to deduce from Theorem 4.8 the following.

***Corollary* 4.9** ([3])**.** *The path-width of every n-element matroid of branch-width at most 2 can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$, provided that the matroid is given with its binary representation. Moreover, a linear layout of the matroid witnessing the path-width can be computed with the same time complexity.*

4.2 Obstructions for Linear Rank-Width

When we started working on linear rank-width we were mostly interested in identifying the obstructions for linear rank-width : can we obtain a bound on the sizes of obstructions? are trees obstructions for linear rank-width? We do not have an answer for any of these questions, except in distance-hereditary graphs[1].

Let $\preceq$ be a quasi-order on graphs. We say that H is a *minimal $\preceq$ obstruction* for a class of graphs $\mathcal{C}$ if $H \notin \mathcal{C}$ but every $H' \prec H$ belongs to $\mathcal{C}$. For instance, we may be interested in vertex-minor obstructions or pivot-minor obstructions. For example we gave in [103, 104] the set of pivot-minor obstructions for directed graphs of $\mathbb{F}_4$-rank-width 1 and also for oriented graphs of $\mathbb{F}_3$-rank-width 1. Let us explain how we used Theorem 4.7 to give the set of distance-hereditary vertex-minor obstructions for linear rank-width at most k.

First, the fact that $\mathrm{lrwd}(T) = \mathrm{pwd}(T)$ for every forest can help in proving that no tree is a minimal vertex-minor obstruction for linear rank-width k. Let $\mathcal{H}_1 := \{R_3\}$ (see Figure 4.1). For $k \geq 2$, let $\mathcal{H}_k$ be the set of (pairwise non isomorphic) trees obtained by taking a new vertex r and three trees in $\mathcal{H}_{k-1}$, and by linking this new vertex to one vertex in each of these three trees. Notice that all the trees in $\mathcal{H}_k$ have the same size. Moreover, it is known that $\mathcal{H}_k$ is exactly the set of minimal acyclic minor obstructions for path-width at most k [58].

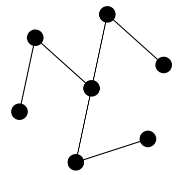

Figure 4.1: The subdivided 3-star R_3

***Lemma* 4.10** ([1]). *Let $k \geq 1$ be an integer. Every tree of linear rank-width $k+1$ contains a tree in $\mathcal{H}_k$ as a vertex-minor.*

As a corollary one can prove the following.

***Corollary* 4.11.** *Let $k \geq 1$ be an integer. No tree is a minimal vertex-minor obstruction for linear rank-width at most k.*

Proof. Let T be an acyclic vertex-minor obstruction for linear rank-width at most k. By Lemma 4.10 and Proposition 4.4 such T should be necessarily in $\mathcal{H}_k$, *i.e.*, there is a node x in V_T such that $T \setminus x$ contains exactly three trees, each in $\mathcal{H}_{k-1}$. Let $T' := T * x \setminus x$, and let D and D' be respectively the canonical decompositions of T and T'. By the characterisation of the canonical decompositions of trees in [17], one can easily verify that D' is obtained from D as follows: let B be the bag containing x, replace B by $B' := B * x \setminus x$. Now, then any component T_1 of $D' \setminus B'$ is in $\mathcal{H}_{k-1}$, *i.e.*, $f_{D'}(B, T_1) = k$. By Theorem 4.7 we can then conclude that $\mathrm{lrwd}(T') = k+1$, contradicting the assumption that T is a minimal vertex-minor obstruction. □

If, nevertheless, one is interested in the set of trees of linear rank-width $k+1$ such that any proper acyclic vertex-minor has linear rank-width k, then we proved in [1] that this set is exactly $\mathcal{H}_k$. By adapting the construction of $\mathcal{H}_k$, we managed in [4] to construct the set of distance-hereditary vertex-minor obstructions for linear rank-width at most k and the construction is based on Theorem 4.7. The construction in [4] generalises the construction given in [88], and as a consequence of both papers we can deduce that the number of distance-hereditary vertex-minor obstructions for linear rank-width at most k is $2^{\theta(3^k)}$.

As for finding big trees in distance-hereditary graphs of large linear rank-width, we used again Theorem 4.7 to relate the path-width of $\mathbb{F}_2$-split decompositions with the linear rank-width of associated distance-hereditary graphs, and obtained the following.

Theorem 4.12 ([4]). *Let $p \geq 1$ be a positive integer and let T be a tree. Let G be a graph such that every prime induced subgraph of G has linear rank-width at most p. If G has linear rank-width at least $40(p+2)|V(T)|$, then G contains a vertex-minor isomorphic to T.*

[1] We claimed in [93] a bound on the sizes of obstructions for linear $\mathbb{F}$-rank-width of σ-symmetric graphs, but we later found a flaw that we still fail to fix.

By Theorem 4.12 in order to solve Conjecture 1 it is enough to consider only prime graphs, and since prime graphs in distance-hereditary graphs have linear rank-width 1, we can deduce that the conjecture is true in distance-hereditary graphs.

4.3 Concluding Remarks

We proved in [1] that the linear rank-width of forests can be computed in linear time and in [3] we gave a recursive characterisation of the linear rank-width of distance-hereditary graphs which were used to propose a polynomial time algorithm for computing their linear rank-width. Other consequences of this characterisation is the construction of minimal distance-hereditary vertex-minor obstructions for linear rank-width and a proof of Conjecture 1 whenever the prime graphs in the $\mathbb{F}_2$-split decomposition have bounded linear rank-width. We ask whether we can extend these results on distance-hereditary graphs to all graphs of bounded rank-width. We can probably use the notion of limbs defined for distance-hereditary graphs in order to compute the linear rank-width of graphs whose prime induced subgraphs have bounded size, however we need new techniques to be able to compute the linear rank-width of graphs of bounded rank-width.

It was proved in [61] that $\mathrm{lcwd}(G) \leq \mathrm{pwd}(G)+2$ for any graph G and we also proved in [1] that the equality holds on forests with a path of length at least 3. As a consequence, we could characterise the linear clique-width of forests in terms of their path-width, which in turn implies a linear time algorithm for computing the linear clique-width of forests.

***Proposition* 4.13** ([1]). *Let T be a forest with at least one edge. If T contains a path of length 3, then* $\mathrm{lcwd}(T) = \mathrm{pwd}(T)+2$. *If T contains no path of length 3, and all its connected components, but one, are singletons, then* $\mathrm{lcwd}(T) = \mathrm{pwd}(T)+1$. *If T contains no path of length 3 and at least two of the connected components of T are not singletons, then* $\mathrm{lcwd}(T) = \mathrm{pwd}(T)+2$.

Forests are essentially the only non trivial graph class for which we know a polynomial time algorithm for computing their linear clique-width. We do not believe that the methods used in [1, 3] could be adapted to compute the linear clique-width of graphs of bounded rank-width, but we wonder whether we can adapt them to compute the linear clique-width of distance-hereditary graphs.

Part II

On the Enumeration of Minimal Dominating Sets

Chapter 5

Equivalence of Dom-Enum and Trans-Enum

The material presented in this chapter is a part of the article [97], where we showed that DOM-ENUM(co-bipartite graphs), TDOM-ENUM(split graphs), DOM-ENUM, TDOM-ENUM and TRANS-ENUM are all **DelayP**-equivalent. We also discussed about the enumeration of minimal connected dominating sets. Since for any hypergraph $\mathcal{H}$ we have that $tr(\mathcal{H}) = tr(\min(\mathcal{H}))$, we can assume that all hypergraphs in this chapter are simple.

5.1 DOM-ENUM and TRANS-ENUM

Let us first define usual graphs associated with hypergraphs.

Definition 5.1. *Given a hypergraph $\mathcal{H}$, the* (bipartite) incidence graph $\mathcal{I}(\mathcal{H})$ *of $\mathcal{H}$ is the bipartite graph $(V_{\mathcal{H}}, \{y_E \mid E \in \mathcal{E}_{\mathcal{H}}\}, \{x y_E \mid x \in E\})$, and the* incidence split graph $\mathcal{I}'(\mathcal{H})$ *of $\mathcal{H}$ is the split graph obtained from $\mathcal{I}(\mathcal{H})$ by turning $V_{\mathcal{H}}$ into a clique. The* co-bipartite incidence graph $\mathcal{B}(\mathcal{H})$ *of $\mathcal{H}$ is the co-bipartite graph obtained from $\mathcal{I}'(\mathcal{H})$ by turning the independent set $\{y_E \mid E \in \mathcal{E}_{\mathcal{H}}\}$ into a clique, and adding a new vertex v that is universal to $V_{\mathcal{H}}$. See Figure 5.1 for illustrations.*

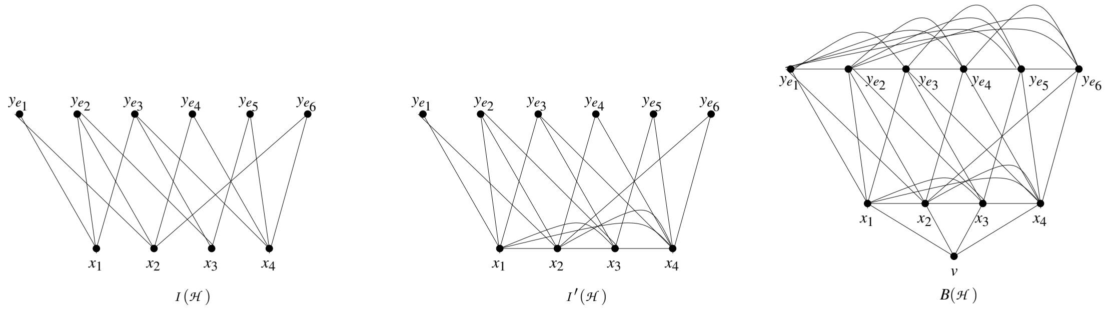

Figure 5.1: Illustrations of graphs from a hypergraph $\mathcal{H}$.

It is not hard to prove that any transversal of $\mathcal{H}$ is a dominating set of $\mathcal{B}(\mathcal{H})$. In the other direction, we proved the following.

***Lemma* 5.2** ([97])**.** *Let $\mathcal{H}$ be a hypergraph and D be a minimal dominating set of $\mathcal{B}(\mathcal{H})$. Then either D is a minimal transversal of $\mathcal{H}$ or $D \subseteq V_{\mathcal{H}} \times \{y_E \mid E \in \mathcal{E}_{\mathcal{H}}\}$.*

Since $\mathcal{B}(\mathcal{H})$ can be constructed in time $O(||\mathcal{H}||)$ and the number of minimal dominating sets of $\mathcal{B}(\mathcal{H})$ that are not minimal transversals of $\mathcal{H}$ are bounded by $|V_{\mathcal{H}}| \times |\mathcal{E}_{\mathcal{H}}|$ we can conclude the following.

Theorem 5.3 ([97])**.** DOM-ENUM *(co-bipartite graphs),* DOM-ENUM *and* TRANS-ENUM *are all* **DelayP***-equivalent.*

From Lemma 2.4 we know that TDOM-ENUM $\leq_{\mathbf{DelayP}}$ TRANS-ENUM, we proved the following.

***Lemma* 5.4** ([97])**.** TRANS-ENUM $\leq_{\mathbf{DelayP}}$ TDOM-ENUM*(split graphs).*

Corollary* 5.5** ([97])**.** DOM-ENUM*(co-bipartite graphs),* TDOM-ENUM*(split graphs),* DOM-ENUM*,* TDOM-ENUM *and* TRANS-ENUM *are all* **DelayP-equivalent.*

In [122] it is proved that the enumeration of (inclusion-wise) minimal d-dominating sets[1] is also **DelayP**-equivalent to TRANS-ENUM. It would be interesting to classify the domination problems for which the enumeration of (inclusion-wise) minimal sets is equivalent to DOM-ENUM or are less harder than DOM-ENUM. We finish this section with the following natural question we asked in [97].

DOM-GRAPH
Input. A hypergraph $\mathcal{H}$ and a positive integer k
Output. A graph G and a set $F \subseteq 2^{V_G}$ with $|F| \leq k$ such that $tr(\mathcal{H}) = \mathcal{D}(G) \cup F$

DOM-GRAPH is NP-complete since the problem whether a hypergraph is a closed neighbourhood hypergraph is NP-complete [12]. For $k = |V_{\mathcal{H}}| \times |\mathcal{E}_{\mathcal{H}}|$ the problem is trivially polynomial since $\mathcal{B}(\mathcal{H})$ satisfies the desired property. What about its complexity for $1 \leq k < |V_{\mathcal{H}}| \times |E_{\mathcal{H}}|$?

5.2 Connected Dominating Sets

In the literature the enumeration of connected subsets satisfying some given property are less studied and this is probably due to the fact that problems related to connected sets are usually harder and we still do not yet have a good understanding of simpler enumeration problems. The MINIMUM CONNECTED DOMINATING SET problem is a well-studied variant of the MINIMUM DOMINATING SET problem due to its numerous applications in networks, and also to its links with the STEINER TREE problem. So, we wanted to know its complexity status with respect to TRANS-ENUM and we were only able to prove that it is harder than TRANS-ENUM.

A *connected dominating set* in a graph G is a dominating set D of G such that $G[D]$ is connected. The set of (inclusion-wise) minimal connected dominating sets of a graph G is denoted by $\mathcal{CD}(G)$ and CDOM-ENUM consists in enumerating $\mathcal{CD}(G)$ for a given graph G. It is worth noticing that minimal connected dominating sets and minimal dominating sets that are connected are two different sets, but in split graphs the two sets coincide.

***Proposition* 5.6** ([97])**.** $\mathcal{CD}(\mathcal{I}'(\mathcal{H})) = tr(\mathcal{H})$ *for every hypergraph* $\mathcal{H}$*. Hence,* TRANS-ENUM $\leq_{\mathbf{DelayP}}$ CDOM-ENUM*(split graphs).*

Because $\mathcal{H} = tr(tr(\mathcal{H}))$ for every simple hypergraph $\mathcal{H}$, one can deduce from Proposition 5.6 that every hypergraph is the set of minimal connected dominating sets of a (split) graph, which is not the case with minimal dominating sets [12].

One would know whether CDOM-ENUM $\leq_{\mathbf{DelayP}}$ TRANS-ENUM. However, the following gives some evidence that it should not be the case. A subset $S \subseteq V_G$ of a connected graph G is called a *separator* of G if $G \setminus S$ is not connected; S is minimal if it does not contain any other separator. For two vertices a and b, an ab-*separator* is a subset $S \subseteq V_G \setminus \{a, b\}$ which disconnects a from b; it is said to be minimal if no proper subset of S disconnects a from b. Every minimal separator is an ab-separator for some pair of vertices a, b. The minimal separators are exactly the minimal ab-separators which do not contain any other cd-separator. For this reason they are often called the *inclusion minimal separators*. A graph may have an exponential

[1] A set D is a d-*dominating set* if each vertex has at least d neighbours in D.

number of minimal separators, but one can enumerate them with polynomial delay [147]. We define $\mathcal{S}(G)$ as the hypergraph $(V_G, \{S \subset V_G \mid S \text{ is a minimal separator of } G\})$.

***Proposition* 5.7** ([97]). *$\mathcal{CD}(G) = tr(\mathcal{S}(G))$ for every non-complete graph G.*

As a corollary CDOM-ENUM($\mathcal{C}$) $\leq_{\mathbf{DelayP}}$ DOM-ENUM for any graph class $\mathcal{C}$ which has a polynomial number of minimal separators, *e.g.*, circular-arc graphs, circle graphs, chordal bipartite graphs, trapezoid graphs, etc. If $\mathcal{C}$ moreover contains the split graphs, then CDOM-ENUM($\mathcal{C}$) is **DelayP**-equivalent to DOM-ENUM (or equivalently TRANS-ENUM), *e.g.*, chordal graphs, weakly chordal graphs.

5.3 Conclusion

We proved that DOM-ENUM, TDOM-ENUM and TRANS-ENUM are all **DelayP**-equivalent, and CDOM-ENUM is harder than all of them. But, we did not look at a characterisation of domination like problems which acts similarly. Can we have a trichotomy theorem (**TotalP**, **TotalP**-equivalent to TRANS-ENUM or **TotalP**-harder than TRANS-ENUM) for say the minimal (σ, ρ)-dominating sets (see Chapter 7 for the definition)?

One easily checks that TRANS-ENUM can be also polynomially reduced to CDOM-ENUM(co-bipartite graphs) since there are only a polynomial number of minimal connected dominating sets in $\mathcal{B}(\mathcal{H})$ that intersect the set $\{y_E \mid E \in \mathcal{E}_{\mathcal{H}}\}$. Can we characterise the graph classes $\mathcal{C}$ for which TRANS-ENUM and DOM-ENUM ($\mathcal{C}$) are equivalent? Given a graph class $\mathcal{C}$, for which hypergraph classes $\mathscr{D}$ do we have TRANS-ENUM($\mathscr{D}$) $\leq_{\mathbf{TotalP}}$ DOM-ENUM($\mathcal{C}$)? Similarly for (connected) minimal (σ, ρ)-dominating sets.

Chapter 6

Independent Systems

We recall that an independent system is a set system $(V, \mathcal{E} \subseteq 2^V)$ that is hereditary under inclusion. We have seen in Section 2.2 that the sets of an independent system can be enumerated with polynomial delay given a polynomial oracle, and that in general the enumeration of maximal sets is intractable. In [97] we defined an independent system in split graphs which are in one-to-one correspondence with the minimal dominating sets in split graphs, and derived from this a linear delay algorithm for DOM-ENUM(split graphs) (Section 6.1). This algorithm does not extend to chordal graphs and the tree structure of chordal graphs, namely *clique trees*, does not help in a straightforward way. Instead we associated with chordal graphs several independent systems and proved that the enumeration of minimal dominating sets in chordal graphs reduces to the enumeration of maximal sets of these independent systems. We finally showed that those maximal sets can be enumerated with polynomial delay by showing that some instances of SUBSETIND can be answered in polynomial time with the help of clique trees [99] (Section 6.2).

6.1 Split Graphs

We cannot use SUBSETIND to solve DOM-ENUM(split graphs) because it is NP-complete and this can be easily proved by reducing the general question into the special case of split graphs [99]. Let us denote by $G := (C, S, E)$ a split graph G with C the clique that we assume maximal and S is the independent part. One easily checks that the minimal dominating sets in a split graph G cannot be characterised by their intersection with S because several minimal dominating sets may have the same intersection with S. Fortunately, the next lemma says that we can characterise the minimal dominating sets of a split graph G with their intersection with C.

***Lemma* 6.1** ([97]). *Let D be a minimal dominating set of a split graph G. Then $D \cap S = S \setminus N_G(D \cap C)$.*

Let G be a split graph and let $\mathcal{DI}(G) := \{A \subseteq C \mid A \text{ is an irredundant set}\}$. The next lemma combined with Lemma 6.1 proves that $\mathcal{DI}(G)$ is in bijection with $\mathcal{D}(G)$. They also show that minimal connected dominating sets and minimal dominating sets that are connected coincide in split graphs. Hence, the enumeration of the latter set is also harder than TRANS-ENUM.

***Lemma* 6.2** ([97]). *Let $A \subseteq C$ be an irredundant set of a split graph G. Then $A \cup (S \setminus N_G(A))$ is a minimal dominating set of G.*

The next lemma shows that $\mathcal{DI}(G)$ is an independent system.

***Lemma* 6.3** ([97]). *Let D be a minimal dominating set of a split graph G. Then for every irredundant set $A \subseteq D \cap C$, the set $A \cup (S \setminus N_G(A))$ is a minimal dominating set.*

Since we can check in linear time whether a set $A \subseteq C$ is an irredundant set, Algorithm 1.1 enumerates the set $\mathcal{D}(G)$ in linear time for every split graph G.

Theorem 6.4 ([97])**.** DOM-ENUM(*split graphs*) *belongs to* **DelayL**.

6.2 Chordal Graphs

Chordal graphs admit nice tree structures, namely *clique trees*, which were used in the past to solve efficiently several NP-complete problems, *e.g.*, MAXIMUM INDEPENDENT SET, FEEDBACK VERTEX SET, etc. by dynamic programming. These algorithms work usually as follows: take a clique separator and subdivide the instance into sub-instances and because we can control the intersections of solutions between the smaller instances, we can efficiently combine the solutions. However, this is not the case for minimal dominating sets because they not only can intersect the separator in an arbitrary number of vertices, but we cannot even control how they do, and this can maybe explain why MINIMUM DOMINATING SET in NP-complete in chordal graphs (even in split graphs whose clique trees have diameter 2). One can expect to obtain a lemma similar to Lemma 6.1, however even in such restricted cases SUBSETIND is already NP-complete in chordal graphs.

Theorem 6.5 ([99])**.** SUBSETIND *is NP-complete in chordal graphs even if a path $\mathcal{P}$, from the root, of the clique tree satisfies that any child C of a clique in $\mathcal{P}$ satisfies either $V(C)\cap(S\cup X)=\emptyset$ or $V(C)\subseteq(S\cup X)$.*

In [99] we proposed to decompose chordal graphs along antichains of cliques and proved the following.

Theorem 6.6 ([99])**.** DOM-ENUM(*chordal graphs*) *belongs to* **DelayP**.

We explain how the algorithm works. Let us from now on consider a chordal graph $G=(V,E)$ and a clique tree $\mathcal{T}$ that is rooted at a node C_r, and recall that it is provided with the ancestor-descendant relation $\preceq_{\mathcal{T}}$. Let us denote by $\mathcal{C}$ its set of (inclusion-wise) maximal cliques. For each $C\in\mathcal{C}$, let us denote by $f(C)$ the set of vertices in C that are not in any maximal clique C' ancestor of C. Notice that $\{f(C)\mid C\in\mathcal{T}\}$ is a partition of V, and for $x\in V$ let $C(x)$ be the unique $C\in\mathcal{C}$ such that $x\in f(C)$. For $C\in\mathcal{C}$, the subtree rooted at C is denoted by $\mathcal{T}(C)$, and let $V(C):=\bigcup_{C'\in\mathcal{T}(C)} f(C')$.

Let us extend the ascendant-descendant relation of $\mathcal{T}$ into a linear ordering of V such that a vertex x is smaller than a vertex y whenever $C(x)\preceq_{\mathcal{T}} C(y)$. For $S\subseteq V$ let

- ⋆ $\mathcal{C}(S):=\{C(x)\mid x\in S\}$,
- ⋆ $Up(S):=\{x\in V\mid \exists C\in\mathcal{C}(S),\ C\prec_{\mathcal{T}} C(x)\}$,
- ⋆
$$\mathcal{L}(S):=\begin{cases}\max\limits_{\preceq_{\mathcal{T}}}\{C\in\mathcal{C}\mid C \text{ has no descendant in } \mathcal{C}(S)\} & \text{if } S\neq\emptyset,\\ \{C_r\} & \text{otherwise.}\end{cases}$$
- ⋆ $A(S):=\{x\in S\mid C(x)\in\max\limits_{\preceq_{\mathcal{T}}}\{\mathcal{C}(S)\}\}$, called *top-set* of S,
- ⋆ $tail(S)$ denotes the largest vertex in S.

A subset $A\subseteq V$ is an *antichain* if $\mathcal{C}(A)$ is a maximal set of pairwise incomparable maximal cliques. A *prefix* of a vertex set S is a subset $S'\subseteq S$ such that no vertex in $S\setminus S'$ is smaller than $tail(S')$. A *partial antichain* is a prefix of an antichain. We allow the $\emptyset$ to be a partial antichain.

Let $K_1,K_2\subseteq C_r$ be given disjoint sets and without confusion we denote $K_1\cup K_2$ by K. A (K_1,K_2)-*extension* of a partial antichain A is a vertex set D such that $(A\cup K)\subseteq D$ and $D\setminus(A\cup K)\subseteq \bigcup_{C\in\mathcal{L}(A\cup K)} V(C)$. Observe that if D is a (K_1,K_2)-extension of A, then A is a prefix of $A(D)$. A (K_1,K_2)-extension D is *feasible* if it is a dominating set and $P(D,x)\neq\emptyset$ for all $x\in D\setminus K_2$. A partial antichain A is (K_1,K_2)-*extendable* if it has a feasible (K_1,K_2)-extension.

Let us briefly explain the ideas of the algorithm and why we introduced (K_1,K_2)-extensions. We first observe that for any minimal dominating set D of G, its top-set $A(D)$ is an $(\emptyset,\emptyset)$-extendable antichain. Moreover, $D\setminus A(D)$ is composed of vertices below $A(D)$, *i.e.*, any vertex in $D\setminus A(D)$ is included in $V(C)\setminus C$ for some $C \in \mathcal{C}(D)$. Using this, we can partition the minimal dominating sets according to their top-sets. Since these top-sets are $(\emptyset,\emptyset)$-extendable, it is enough to enumerate all $(\emptyset,\emptyset)$-extendable antichains, and for each $(\emptyset,\emptyset)$-extendable antichain A, enumerate all minimal dominating sets whose top-set is A. By definition, for each $(\emptyset,\emptyset)$-antichain A there is at least one minimal dominating set whose top-set is A. Therefore, each output $(\emptyset,\emptyset)$-antichain will give rise to a solution.

Now for a minimal dominating set D and a clique $C \in \mathcal{C}(A(D))$, each vertex x in $D \cap (V(C) \cup C)$ cannot have a private neighbor in another $G[V(C') \cup C']$ for some other $C' \in \mathcal{C}(A(D))$. Therefore, we can treat each $G[V(C) \cup C]$ independently. However, for each $C \in \mathcal{C}(A(D))$ the set $D \cap (V(C) \cup C)$ is not necessarily a minimal dominating set of $G[V(C) \cup C]$ since $D \cap C$ may be equal to a singleton $\{x\}$ with x having a private neighbor in $Up(A(D))$. In such cases we are looking in $G[V(C) \cup C]$ a dominating set D' of $G[V(C) \cup C]$ containing x where x does not necessarily have a private neighbor, but all the other vertices in D' do, *i.e.*, D' is a feasible $(\emptyset, \{x\})$-extension in $G[V(C) \cup C]$ with clique tree $\mathcal{T}(C)$. This situation is what exactly motivated the notion of (K_1, K_2)-extensions.

Assume now we are given a (K_1, K_2)-extendable antichain A. Contrary to $(\emptyset,\emptyset)$-antichains we can have a vertex x in K that belongs to several cliques in A. So we cannot independently make recursive calls in $G[V(C) \cup C]$ for each $C \in \mathcal{C}(A)$. But, for each feasible (K_1, K_2)-extension of A and each $C \in \mathcal{C}(A)$ the set $D \cap (V(C) \cup C)$ is a feasible (K_C^1, K_C^2)-extension of $G[V(C) \cup C]$ for some disjoint K_C^1 and K_C^2 in $(A \cup K) \cap C$. Now the whole task was to define for each $C \in \mathcal{C}(A)$ the sets K_C^1 and K_C^2 in $(A \cup K) \cap C$ in such a way that by combining all these feasible (K_C^1, K_C^2)-extensions we obtain a feasible (K_1, K_2)-extension of A, and also any feasible (K_1, K_2)-extension can be obtained in that way. Actually, the way of setting K_C^1 and K_C^2 was the key, and will be described next. By the way we have proved the following in [99] which shows that one can enumerate with polynomial delay all the (K_1, K_2)-extendable antichains with Algorithm 1.2.

***Lemma* 6.7** ([99])**.** *The set of (K_1, K_2)-extendable partial antichains form an independent system and one can moreover check in polynomial time whether a given partial antichain is (K_1, K_2)-extendable.*

Now, it remains to show how to enumerate with polynomial delay the feasible (K_1, K_2)-extensions. Given a (K_1, K_2)-extendable antichain A the enumeration consists in enumerating feasible (K_C^1, K_C^2)-extensions of $G[V(C) \cup C]$ for each $C \in \mathcal{C}(A)$, but for appropriate pairs (K_C^1, K_C^2).

A vertex set $D \supseteq A \cup K$ is called a *partial* (K_1, K_2)*-extension* of A if there is a feasible (K_1, K_2)-extension D' of A such that $D \setminus (A \cup K)$ is a prefix of $D' \setminus (A \cup K)$, and all the vertices in $V(C(x))$ for $x \in A$ is dominated by D if x is smaller than $tail(D \setminus (A \cup K))$. Our strategy was to enumerate all partial (K_1, K_2)-extensions of A, similar to the antichain enumeration. For a partial (K_1, K_2)-extension D of A, let $C(D)$ be the smallest clique C in $\mathcal{C}(A)$ such that a vertex in $V(C)$ is not dominated by D. To enumerate all partial (K_1, K_2)-extensions of A and find all (K_1, K_2)-extensions of A, we start from $D = A \cup K$ and repeatedly add a $(K_{C(D)}^1, K_{C(D)}^2)$-extension of $G[V(C(D)) \cup C(D)]$ to D for appropriate $(K_{C(D)}^1, K_{C(D)}^2)$, while keeping the extendability. We characterized the possible $(K_{C(D)}^1, K_{C(D)}^2)$ in [99] and showed that the partial (K_1, K_2)-extensions form an independent system and the maximal ones, which are exactly the feasible (K_1, K_2)-extensions, can be enumerated with polynomial delay with the help of Algorithm 1.2.

6.3 Future Work

We used the enumeration of (maximal) sets in independent systems to derive polynomial delay algorithms for DOM-ENUM in split and chordal graphs and these algorithms use deeply the structure of split and chordal graphs. To what extent did we strongly use clique trees? Can we define similar techniques for other graph classes, *e.g.*, chordal bipartite graphs which admit a tree-decomposition with complete bipartite graphs as bags [112] and weakly chordal graphs which admit a nice linear ordering [84, 85][1]?

[1] We erroneously claimed that co-bipartite graphs are weakly chordal in [99, 77] because of a misunderstanding of the definition of weakly chordal graphs, and hence the tractability of weakly chordal graphs is still open.

Chapter 7

Parsimonious Reductions

It is common in order to count or enumerate the elements of a set $\mathcal{O}$, to identify a set $\mathcal{T}$ such that we can easily count or enumerate the members of $\mathcal{T}$ and produce a bijection $b : \mathcal{O} \to \mathcal{T}$. We proved for instance in [98] that DOM-ENUM(interval $\cup$ permutation) belongs to **DelayL** by exhibiting a bijection with the set of maximal paths in a DAG. We extended this result to a more general graph class which contains permutation graphs, circular-arc graphs, complements of k-degenerate graphs, etc. [75] (see Section 7.1). We characterised in [97] the set of edges that can be added into a graph without changing the set of minimal dominating sets, and we demonstrated its usefulness with the case of P_6-free chordal graphs (see Section 7.2).

7.1 Graphs with Polynomially Linear Bounded Neighbourhood

Given a graph G and a subset A of its vertex set, let us denote by $N^A := \{N_G(X) \cap \overline{A} \mid X \subseteq A\}$. It is proved in [23] that if G admits a layout (T, L) such that for each edge e the size of $\max\{N^{X^e}, N^{\overline{X^e}}\}$ is bounded by some polynomial $p(|V_G|)$, then a large set of NP-complete problems can be solved on G in time $poly(|V_G|, p(|V_G|))$, including INDEPENDENT SET, DOMINATING SET, TOTAL DOMINATING SET, etc. Examples of such graph classes are graphs of bounded rank-width, interval graphs, permutation graphs, complements of k-degenerate graphs, etc. In particular in graphs of bounded rank-width this set of problems admits a singly exponential FPT polynomial time algorithm, which sounds much better than the bounds one can obtain from the meta-theorem [30]. Since the solutions of every $MSOL_1$ property can be listed with linear delay on graphs of bounded rank-width by the meta-theorem in [34], yielding of course non-elementary constants, one can wonder whether the non-elementary constants can be reduced to a single exponential in the case of problems studied in [23]. We consider in this section this question in the case where the given layout is a linear layout. What is presented is an extension of results published in [75] where we considered graphs of bounded *linear maximum induced matching width* introduced by Vatshelle in his PhD thesis [159]. Let us first give some definitions before stating the result, mostly proved in [75]. Because we will use the results given here in Chapter 8, we use graphs with vertex set bi-coloured.

Let G be a graph and let $\mathbf{Red}, \mathbf{Blue} \subseteq V_G$ such that $\mathbf{Red} \cup \mathbf{Blue} = V_G$. We refer to the vertices (or subsets) of **Red** as the *red* vertices (or sets), the vertices (or subsets) of **Blue** as the *blue* vertices (or sets), and we say that G together with given sets **Red** and **Blue** is a *coloured graph*. For simplicity, whenever we say that G is a coloured graph, it is assumed that the sets **Red** and **Blue** are given. Notice that **Red** and **Blue** are not necessarily disjoint. In particular, it can happen that $\mathbf{Red} = \mathbf{Blue} = V_G$; a non-coloured graph G is a coloured graph with $\mathbf{Red} = \mathbf{Blue} = V_G$.

Definition 7.1. *Let $d \in \mathbf{N}$. Let G be a coloured graph and let $A \subseteq V_G$. Two subsets X and Y of A of same colour are* d-neighbour equivalent *w.r.t. A, denoted by $X \equiv_A^d Y$, if* $\min(d, |X \cap N_G(x)|) = \min(d, |Y \cap N_G(x)|)$ *for all $x \in \overline{A}$ of different colour from X and Y. It is folklore to prove that $\equiv_A^d$ is an equivalence relation, and we let $nec_R(\equiv_A^d)$ and $nec_B(\equiv_A^d)$ be respectively the number of red and blue equivalence classes. We let*

$$\mathrm{neigh}_G^d(A) := \max\{nec_R(\equiv_A^d), nec_B(\equiv_A^1), nec_R(\equiv_{\overline{A}}^d), nec_B(\equiv_{\overline{A}}^1)\}.$$

An n-vertex graph is said to be of *polynomially (linear) bounded neighbourhood* if, for each $d \in \mathbf{N}$, there are a polynomial $p(x,y)$ and a (linear) layout of V_G of neigh_G^d-width at most $p(n,d)$. Several graph classes have been proved to be of polynomially linear bounded neighbourhood, *e.g.*, interval graphs, permutation graphs, Dilworth-k graphs, circular permutation graphs, bipartite tolerance graphs, bipartite unit-disk graphs, etc. [6, 29].

The (σ,ρ)-*dominating set* notion was introduced by Telle and Proskurowski [152] as a generalization of dominating sets. Indeed, many NP-hard domination type problems such as the problems INDEPENDENT SET, d-DOMINATING SET, INDEPENDENT DOMINATING SET and TOTAL DOMINATING SET are special cases of the (σ,ρ)-Dominating Set Problem. More examples are given in [23, Table 1].

Definition 7.2. *Let σ and ρ be finite or co-finite subsets of $\mathbf{N}$ and let G be a coloured graph. We say that a set $D \subseteq V_G$ (σ,ρ)-*dominates *$U \subseteq V(G)$ if it (σ,ρ)-dominates every $u \in U$,* i.e., *for each $u \in U$, $|N_G(u) \cap D| \in \sigma$ if $u \in D$, otherwise $|N_G(u) \cap D| \in \rho$.*

*A set of vertices $D \subseteq$ **Red** is a* **Red** (σ,ρ)-dominating set *if D (σ,ρ)-dominates **Blue**.*

Let $d(\mathbf{N}) = 0$. For every finite set $\mu \subseteq \mathbf{N}$, let $d(\mu) = 1 + \max\{a \mid a \in \mu\}$, and for every co-finite set $\mu \subseteq \mathbf{N}$, let $d(\mu) = 1 + \max\{a \mid a \in \mathbf{N} \setminus \mu\}$. For finite or co-finite subsets σ and ρ of $\mathbf{N}$, we let $d(\sigma,\rho) = \max(d(\sigma), d(\rho))$.

Let G be a coloured graph. If **Red** $=$ **Blue** $= V_G$, then a **Red** (σ,ρ)-dominating set is the classical notion of (σ,ρ)-*dominating set*. As pointed out in [23] given a subset D of **Red**, we can check if D is a **Red** (σ,ρ)-dominating set by computing $|D \cap N_G(x)|$ up to $d(\sigma,\rho)$ for each vertex x in **Blue**. It was also proved in [23] that a (inclusion-wise) minimum (or maximum) (σ,ρ)-dominating set can be computed in time $poly(n,c)$ in a graph G given with a layout of neigh_G^d-width c.

A **Red** (σ,ρ)-dominating set D of a graph G is said 1-*minimal* if for each vertex x in D, $D \setminus x$ is not a **Red** (σ,ρ)-dominating set. Clearly, every (inclusion-wise) minimal **Red** (σ,ρ)-dominating set is 1-minimal, but the converse is not true for arbitrary σ and ρ. In the case of total domination and classical domination however the two notions coincide. We will prove next how to enumerate with linear delay all 1-minimal (σ,ρ)-dominating sets in graphs with polynomially linear bounded neighbourhood, and show in the concluding remarks how to adapt it for 1-maximal ones. The main result in [75] is the following which generalises the results in [98].

Theorem 7.3 ([75])**.** *Let (σ,ρ) be a pair of finite or co-finite subsets of $\mathbf{N}$ and let $d := d(\sigma,\rho)$. For an n-vertex coloured graph G given with a linear layout of V_G of neigh_G^d-width at most c, one can count in time bounded by $poly(n,c)$, and enumerate with linear delay, all 1-minimal **Red** (σ,ρ)-dominating sets of G.*

***Corollary* 7.4** ([75])**.** *Let (σ,ρ) be a pair of finite or co-finite subsets of $\mathbf{N}$. Then, for every colored graph G in one of the following graph classes, we can count in polynomial time, and enumerate with linear delay all 1-minimal **Red** (σ,ρ)-dominating sets of G: interval graphs, permutation graphs, circular-arc graphs, circular permutation graphs, trapezoid graphs, convex graphs, bipartite unit-disks, bipartite tolerance graphs, and for fixed k, k-polygon graphs, Dilworth-k graphs, complements of k-degenerate graphs.*

The following corollary improves some known results in the enumeration of minimal transversals of interval and circular-arc hypergraphs where only an incremental polynomial time algorithm was known (see e.g. [139]).

***Corollary* 7.5** ([75])**.** *For every hypergraph $\mathcal{H}$ being an interval hypergraph or a circular-arc hypergraph one can count in polynomial time, and enumerate with linear delay, all minimal transversals of $\mathcal{H}$.*

In order to show the main ideas of the proof of Theorem 7.3, let us fix a pair (σ,ρ) of finite or co-finite subsets of $\mathbf{N}$ and let $d := d(\sigma,\rho)$. Let also G be a fixed coloured graph given with a linear layout $x_1,\ldots,x_n$ of neigh_G^d-width at most c. Furthermore, for all $i \in \{1,2,\ldots,n\}$, we let $A_i := \{x_1, x_2, \ldots x_i\}$ and $\overline{A}_i := \{x_{i+1}, x_{i+2}, \ldots x_n\}$. First, we need some certificate that a considered set is 1-minimal. Let D be a **Red** (σ,ρ)-dominating set of G. For a vertex $u \in D$, the vertex $v \in$ **Blue** is its *certifying vertex* (or a *certificate*) if v is not (σ,ρ)-dominated by $D \setminus \{u\}$.

***Lemma* 7.6** ([75]).
1. *A set $D \subseteq$ **Red** is a 1-minimal **Red** (σ,ρ)-dominating set of G if and only if each vertex $u \in D$ has a certificate.*
2. *Let D be a **Red** (σ,ρ)-dominating set of G. If v is a certificate for $u \in D$, then $v = u$ or v is a certificate for all vertices of $N_G(v) \cap D$.*

We define $\sigma^* = \sigma \setminus \rho$. Let also $\sigma^- = \{i \in \sigma \mid i-1 \notin \sigma\}$, $\rho^- = \{i \in \rho \mid i-1 \notin \rho\}$.

***Lemma* 7.7** ([75]). *The sets σ^*, σ^- and ρ^- are finite or co-finite. Also, $d(\sigma^*) \leq d(\sigma,\rho)$ and $d(\sigma^-,\rho^-) \leq d(\sigma,\rho)+1$.*

***Lemma* 7.8** ([75]). *Let D be a **Red** (σ,ρ)-dominating set of G and let $u \in D$. The vertex u is a certificate for itself if and only if $u \in$ **Blue** and $|N_G(u) \cap D| \in \sigma^*$. A vertex $v \in N_G(u) \cap$ **Blue** is a certificate for u if and only if D (σ^-,ρ^-)-dominates v.*

We borrow a last idea from [23]. For every $A \subset V_G$ and every subset X of A, we denote by $rep_A^d(X)$ the lexicographically smallest set $R \subseteq A$ such that $|R|$ is minimised and $R \equiv_A^d X$. Notice that it can happen that $R = \emptyset$. The following is proved in [23] in uncoloured graphs, but it holds for the coloured graphs.

***Lemma* 7.9** ([23]). *For every $i \in \{1,\ldots,n\}$, one can compute a list LR_i^d containing all representatives* w.r.t. $\equiv_{A_i}^d$ *in time $O((nec_R(\equiv_{A_i}^d) + nec_B(\equiv_{A_i}^d)) \cdot \log(nec_R(\equiv_{A_i}^d) + nec_B(\equiv_{A_i}^d)) \cdot n^2)$. One can also compute a data structure that given a set $X \subseteq A_i$ in time $O(\log(nec_R(\equiv_{A_i}^d) + nec_B(\equiv_{A_i}^d)) \cdot |X| \cdot n)$ allows us to find a pointer to $rep_{A_i}^d(X)$ in LR_i^d. Similar statements hold for the list $LR_{\bar{i}}^d$ containing all representatives* w.r.t. $\equiv_{\overline{A_i}}^d$.

We are now ready to define the DAG the maximal paths of which correspond to the 1-minimal **Red** (σ,ρ)-dominating sets of G.

Let $1 \leq j < n$ and let $(R_j, R'_j, C_j, C'_j) \in LR_j^d \times LR_{\bar{j}}^d \times LR_j^1 \times LR_{\bar{j}}^1$ and $(R_{j+1}, R'_{j+1}, C_{j+1}, C'_{j+1}) \in LR_{j+1}^d \times LR_{\overline{j+1}}^d \times LR_{j+1}^1 \times LR_{\overline{j+1}}^1$.

There is an *ε-arc-1* from (R_j, R'_j, C_j, C'_j) to $(R_{j+1}, R'_{j+1}, C_{j+1}, C'_{j+1})$ if

(1.1) $R_j \equiv_{A_{j+1}}^d R_{j+1}$ and $R'_j \equiv_{\overline{A_j}}^d R'_{j+1}$, and

(1.2) if ($x_{j+1} \notin$ **Blue** or ($x_{j+1} \in$ **Blue** and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \in \rho$ and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \rho^-$)) then ($C_{j+1} = rep_{A_{j+1}}^1(C_j)$) and $C'_j = rep_{\overline{A_j}}^1(C'_{j+1})$, otherwise we should have ($|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \in \rho^-$) and

- (1.2.a) if $N(x_{j+1}) \cap (\overline{A}_{j+1} \cap \textbf{Red}) \neq \emptyset$, then $C_{j+1} = rep_{A_{j+1}}^1(C_j \cup \{x_{j+1}\})$, else $C_{j+1} = rep_{A_{j+1}}^1(C_j)$,
- (1.2.b) if $N(x_{j+1}) \cap (A_j \cap \textbf{Red}) \neq \emptyset$, then $C'_j = rep_{\overline{A_j}}^1(C'_{j+1} \cup \{x_{j+1}\})$, else $C'_j = rep_{\overline{A_j}}^1(C'_{j+1})$.

There is an *ε-arc-2* from (R_j, R'_j, C_j, C'_j) to $(R_{j+1}, R'_{j+1}, C_{j+1}, C'_{j+1})$ if

(2.1) $R_{j+1} \equiv_{A_{j+1}}^d (R_j \cup \{x_{j+1}\})$, $R'_j \equiv_{\overline{A_j}}^d (R'_{j+1} \cup \{x_{j+1}\})$, $x_{j+1} \in$ **Red**, ($|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \in \sigma$ if $x_{j+1} \in$ **Blue**), and

(2.2) if ($x_{j+1} \notin$ **Blue** or ($x_{j+1} \in$ **Blue** and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \sigma^-$)), then ($C_{j+1} = rep_{A_{j+1}}^1(C_j)$ and $C'_j = rep_{\overline{A_j}}^1(C'_{j+1})$), otherwise we should have ($|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \in \sigma^-$) and

- (2.2.a) if $N(x_{j+1}) \cap (\overline{A}_{j+1} \cap \textbf{Red}) \neq \emptyset$, then $C_{j+1} = rep_{A_{j+1}}^1(C_j \cup \{x_{j+1}\})$, else $C_{j+1} = rep_{A_{j+1}}^1(C_j)$, and
- (2.2.b) if $N(x_{j+1}) \cap (A_j \cap \textbf{Red}) \neq \emptyset$, then $C'_j = rep_{\overline{A_j}}^1(C'_{j+1} \cup \{x_{j+1}\})$, else $C'_j = rep_{\overline{A_j}}^1(C'_{j+1})$, and

(2.3) either ($N(x_{j+1}) \cap (C_j \cup C'_{j+1}) \neq \emptyset$) or (($x_{j+1} \in$ **Blue** and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \in \sigma^*$).

If D is a minimal **Red** (σ,ρ)-dominating set containing the vertex x_i, then there is a tuple (R, R', C, C') such that $D \cap A_i \equiv_{A_i}^d R$, $D \cap \overline{A}_i \equiv_{\overline{A}_i}^d R'$, and C and C' are certificates for vertices in $D \cap \overline{A}_i$ and $D \cap A_i$ respectively. The nodes of the DAG will code this information, and the arcs will code the way to construct iteratively D (an ε-arc-1 tells us that we will not add the next vertex, but if it is blue it should be dominated, and the ε-arc-2 tells that we will add the next vertex and dominate it also if it is a blue vertex).

The nodes of $DAG(G)$**.** $(R, R', C, C', i) \in LR_i^d \times LR_{\overline{i}}^d \times (LR_i^1 \cap 2^{A_i \cap \mathbf{Blue}}) \times (LR_{\overline{i}}^1 \cap 2^{\overline{A}_i \cap \mathbf{Blue}}) \times [n]$ is a node of $DAG(G)$ whenever $x_i \in \mathbf{Red}$. We call i the *index* of (R, R', C, C', i). Finally $s = (\emptyset, \emptyset, \emptyset, \emptyset, 0)$ is the *source node* and $t = (\emptyset, \emptyset, \emptyset, \emptyset, n+1)$ is the *terminal node* of $DAG(G)$.

The arcs of $DAG(G)$**.** There is an arc from the node $(R_0, R_0', C_0, C_0', j)$ to the node $(R_p, R_p', C_p, C_p', j+p)$ with $1 \leq j < j+p \leq n$ if there exist tuples $(R_1, R_1', C_1, C_1'), \ldots, (R_{p-1}, R_{p-1}', C_{p-1}, C_{p-1}')$ such that (1) for each $1 \leq i \leq p-1$ $(R_i, R_i', C_i, C_i') \in LR_{j+i}^d \times LR_{\overline{j+i}}^d \times (LR_i^1 \cap 2^{A_{j+i} \cap \mathbf{Blue}}) \times (LR_{\overline{j+i}}^1 \cap 2^{\overline{A}_{j+i} \cap \mathbf{Blue}})$ and there is an ε-arc-1 from $(R_{i-1}, R_{i-1}', C_{i-1}, C_{i-1}')$ to (R_i, R_i', C_i, C_i'), and (2) there is an ε-arc-2 from $(R_{p-1}, R_{p-1}', C_{p-1}, C_{p-1}')$ to (R_p, R_p', C_p, C_p').

There is an arc from the source node to a node (R, R', C, C', j) if $(S := \{x \in (A_j \cap \mathbf{Blue}) \setminus \{x_j\} \mid N(x) \cap (\overline{A}_j \cap \mathbf{Red}) \neq \emptyset \text{ and } |N(x) \cap (\{x_j\} \cup R')| \in \rho^-\})$

(S1) $\{x_j\} \equiv_{A_j}^d R$ and $(\{x_j\} \cup R')$ (σ, ρ)-dominates $A_j \cap \mathbf{Blue}$,

(S2) if ($x_j \in \mathbf{Blue}$ and $|N(x_j) \cap R'| \in \sigma^-$) then $C \equiv_{A_j}^1 (S \cup \{x_j\})$, otherwise $C \equiv_{A_j}^1 S$, and

(S3) either $(N(x_j) \cap (C' \cup C) \neq \emptyset)$ or ($x_j \in \mathbf{Blue}$ and $|N(x_j) \cap R'| \in \sigma^*$).

There is an arc from a node (R, R', C, C', j) to the terminal node if

(T1) $|N(x) \cap R| \in \rho$ for each $x \in \overline{A}_{j+1} \cap \mathbf{Blue}$, and

(T2) $C' \equiv_{\overline{A_j}}^1 \{x \in \overline{A}_j \cap \mathbf{Blue} \mid N(x) \cap (A_j \cap \mathbf{Red}) \neq \emptyset \text{ and } |N(x) \cap R| \in \rho^-\}$.

An arc from the source node to a node (R, R', C, C', j) says roughly that all blue vertices in A_j are dominated and x_j is the first vertex of the dominating set, while an arc from (R, R', C, C', j) to the terminal node tells us that all blue vertices in $\overline{A}_j$ are dominated and x_j is the last vertex of the dominating set.

One easily checks that $DAG(G)$ is a DAG since the arcs are always from a node at index j to a node at higher index, and can be also constructed in time $poly(n, c)$. If $P = (s, v_1, v_2, \ldots, v_p, t)$ is a path in $DAG(G)$, then the *trace of* P, denoted by $\mathbf{trace}(P)$, is defined as $\{x_{j_1}, x_{j_2}, \ldots, x_{j_p}\}$ where for all $i \in \{1, 2, \ldots, p\}$, j_i is the index of the node v_i. We proved the following which says that it is sufficient to count and enumerate the maximal paths of $DAG(G)$ to count and enumerate the 1-minimal **Red** (σ, ρ)-dominating sets of G. And since in a DAG we can count the maximal paths and enumerate them with linear delay (see for instance [98]), this concludes the proof of Theorem 7.3.

Lemma* 7.10** ([75])**.** *Let $\mathcal{P}$ be the set of paths in $DAG(G)$ from the source node to the terminal node. The mapping which associates with every $P \in \mathcal{P}$* $\mathbf{trace}(P)$ *is a one-to-one correspondence with the set of* 1*-minimal* ***Red (σ, ρ)*-dominating sets.*

7.2 Completion

It is well-known that $tr(\mathcal{H}) = tr(\min(\mathcal{H}))$ for any hypergraph $\mathcal{H}$. We were curious to know given a graph G such that $\mathcal{N}(G) \neq \min(\mathcal{N}(G))$, how can we modify it into a graph G' on the same vertex set such that $\mathcal{N}(G') = \min(\mathcal{N}(G))$. For that purposes we characterised in [97] the set of edges that can be added without modifying the set of minimal dominating sets, and define the notion of *maximal extension* of a graph. It turns out that this notion can be used to obtain new output-polynomial algorithms, that we demonstrated with the case of P_6-free chordal graphs.

Definition 7.11 ([97])**.** *For a graph G we denote by $IR(G)$ the set of vertices (called* irredundant *vertices) that are minimal with respect to the neighbourhood inclusion. In case of equality between minimal vertices, exactly one is considered as irredundant. All the other vertices are called* redundant *and the set of redundant vertices is denoted by $RN(G)$. Notice that if a vertex x is redundant, then there exists an irredundant vertex y such that $N_G[y] \subseteq N_G[x]$. The* completion graph *of a graph G is the graph G_{co} with vertex set V_G and edge set $E_G \cup \{xy \mid x, y \in RN(G),\ x \neq y\}$,* i.e., *$G_{co}$ is obtained from G by turning $RN(G)$ into a clique.*

The completion graph of a split graph $G=(C,S,E)$ is G itself, since all vertices in S are irredundant. Graph classes are not hereditary with respect to the completion operation. For instance, the completion operation does not preserve the chordality of a graph, paths are chordal graphs but their completion graphs are not chordal. The following characterised the optimality of the addition of the edges.

***Proposition* 7.12** ([97])**.** *Let G be a graph. Then $\mathcal{D}(G)=\mathcal{D}(G_{co})$. Let G' be $(V_G, E_G \cup \{e\})$ with e a non-edge of G. Then $\mathcal{D}(G) \neq \mathcal{D}(G')$ if and only if $e \cap IR(G) \neq \emptyset$.*

We demonstrated its usefulness by showing that completion graphs of P_6-free chordal graphs are split graphs.

***Proposition* 7.13** ([97])**.** *Let G be a P_6-free chordal graph. Then G_{co} is a split graph.*

As a corollary, DOM-ENUM(P_6-free chordal graphs) belongs to **DelayL** [97], which is better than the polynomial delay obtained from Section 6.2. We also characterised completion graphs that are chordal graphs by showing that they must be split.

***Proposition* 7.14** ([97])**.** *Let G be a graph. Then G_{co} is a chordal graph if and only if G_{co} is a split graph.*

7.3 Concluding Remarks

We have seen how to reduce the enumeration of 1-minimal **Red** (σ,ρ)-dominating sets into the enumeration of paths in a DAG when the graph G is given with a linear layout of neigh_G-width at most c. As a consequence, in graphs of bounded linear rank-width, one can enumerate the 1-minimal **Red** (σ,ρ)-dominating sets with delay depending on the next output and the pre-processing is only singly exponential in the linear rank-width, contrary to [34]. Also, we can count in polynomial time, and enumerate with delay depending only on the next output the minimal (total) dominating sets in several well-known graph classes: circular permutation, trapezoid graphs, etc. pushing further the graph classes where DOM-ENUM is tractable. We wonder whether we can extend the construction to graphs given with a layout of neigh_G-width at most c. In this case, we are considering the enumeration of AND-trees in an AND-DAG [34]. We can probably construct an AND-DAG such that its AND-trees correspond to the 1-minimal **Red** (σ,ρ)-dominating sets. However, the delay will depend on the height of the layout and not on the size of the output. Probably, a cleaning as the one done in [34] is possible.

Let us now explain how to adapt the definition of certificate so that we can enumerate in the same way all 1-maximal **Red** (σ,ρ)-dominating sets, *i.e.*, those **Red** (σ,ρ)-dominating sets D such that $D\cup\{x\}$ is not a **Red** (σ,ρ)-dominating set for any $x \in V_G \setminus D$. Let D be a **Red** (σ,ρ)-dominating set. For a red vertex $u \notin D$, the blue vertex v is a certificate for u if v is not (σ,ρ)-dominated by $D\cup\{u\}$. If we set $\sigma^+ := \{i \in \sigma \mid i+1 \notin \sigma\}$ and $\rho^+ := \{i \in \rho \mid i+1 \notin \rho\}$, then by substituting 1-maximal, σ^+ and ρ^+ to respectively 1-minimal, σ^- and ρ^-, one can still prove that Lemmas 7.6, 7.7 and 7.8 are still valid. From this one can adapt without difficulties the construction of the DAG.

If every (inclusion-wise) minimal/maximal **Red** (σ,ρ)-dominating set is a 1-minimal/1-maximal **Red** (σ,ρ)-dominating set, the converse is not true. The technique used here can be probably adapted to the enumeration of the (inclusion-wise) minimal/maximal **Red** (σ,ρ)-dominating sets. The main obstacle is the definition of a certificate which would give a local characterisation of minimality/maximality.

We did not push further the notion of completion, in particular we do not know for instance which graphs have chordal completion graphs, and the only proof we are aware of Proposition 7.14 does not tell us much because it uses the finiteness of the graphs we deal with. It would be interesting to characterise hypergraph classes for which after applying the completion to their associated co-bipartite graphs DOM-ENUM becomes tractable.

Chapter 8

Flipping Method in the Graph of Solutions

The most general tool in enumeration algorithm is probably the one based on traversing the graph of solutions. The idea is to define a neighbouring relation between the solutions and then define a way to traverse it. One can for instance prove that the obtained graph is Hamiltonian, connected or identify a set of solutions and prove that any solution can be reached from that set, this latter raising the following difficulty: how to enumerate once each solution. We refer to for instance [74, 89, 118] for more information. In this chapter we present a variant of this technique developed in [78] which we adapted in [75, 77] to show that DOM-ENUM(chordal bipartite graphs ∪ unit square graphs) belongs to **IncP**.

8.1 Flipping Method

Roughly, the flipping method consists in taking an isolated vertex from a minimal dominating set and replaces it with a maximal independent set from its set of private neighbours. This operation is the neighbouring relation of the graph of solutions and the basis set will be the set of maximal independent sets. Let us now explain it formally and state the main lemma implicitly proved in [78].

Let $G := (V, E)$ be a graph and let us assume that its set of vertices is arbitrarily ordered as $x_1, \ldots, x_n$. Given a dominating set D', we say that the minimal dominating set D is *obtained from D' by greedy removal* if we initially set $D := D'$, and repeatedly apply the following rule: if D is not minimal, then take the smallest vertex x in D such that $D \setminus x$ is a dominating set and replace D with $D \setminus x$. One clearly notices that we never remove vertices in D' that already have a private neighbour.

Let D be a minimal dominating set of G such that $G[D]$ has at least one edge uv. Then the vertex $u \in D$ is dominated by the vertex $v \in D$. Therefore, $P(D, u) \neq \emptyset$. Let X be a non-empty (inclusion-wise) maximal independent set such that $X \subseteq P(D, u)$. Consider the set $D' := (D \setminus \{u\}) \cup X$. Notice that D' is a dominating set in G, since all vertices of $P(D, u)$ are dominated by X by the maximality of X and u is dominated by v, but D' is not necessarily minimal, because it can happen that X dominates all the certificates of some vertex of $D \setminus \{u\}$. We apply greedy removal of vertices to D' to obtain a minimal dominating set. Let Z be the set of vertices that are removed by this to ensure minimality. Observe that $X \cap Z = \emptyset$ and $u \notin Z$ by the definition of these sets; in fact there is no edge between a vertex of X and a vertex of Z. Finally, let $D^* := ((D \setminus \{u\}) \cup X) \setminus Z$.

It is important to notice that $|E(G[D^*])| < |E(G[D])|$. Indeed, to construct D^*, we remove the endpoint u of the edge $uv \in E(G[D])$ and, therefore, reduce the number of edges. Then we add X, but these vertices form an independent set in G and, because they are privates for u with respect to D, they are not adjacent to any vertex of $D \setminus \{u\}$. Therefore, $|E(G[D^*])| \leq |E(G[D'])| < |E(G[D])|$.

The *flipping* operation is exactly the *reverse* of how we generated D^* from D, *i.e.*, it replaces a non-empty independent set X in $G[D^*]$ such that $X \subseteq G[D^*] \cap N_G(u)$ for a vertex $u \notin D^*$ with their neighbour u in G to obtain D. In particular, we are interested in all minimal dominating sets D that can be generated from D^* in this way. Given D and D^* as defined above, we say that D^* is a *parent of D with respect to flipping u and X*. We say that D^* is a *parent* of D if there is a vertex $u \in V$ and an independent set $X \subseteq N_G(u)$ such that D^* is a parent with respect to flipping u and X. It is important to note that each minimal dominating set D such that $E(G[D]) \neq \emptyset$ has a unique parent with respect to flipping of any $u \in D \cap N_G[D \setminus \{u\}]$ and a maximal independent set $X \subseteq P(D,u)$, as Z is lexicographically selected by a greedy algorithm. Similarly, we say that D is a *child* of D^* (with respect to flipping u and X) if D^* is the parent of D (with respect to flipping u and X). The proof of the following lemma is implicit in [78].

***Lemma* 8.1** ([78])**.** *Suppose that for a graph G, all independent sets $X \subseteq N_G(u)$ for a vertex u can be enumerated in polynomial time. Suppose also that there is an enumeration algorithm $\mathcal{A}$ that, given a minimal dominating set D^* of a graph G such that $G[D^*]$ has an isolated vertex, a vertex $u \in V_G \setminus D^*$ and a non-empty independent set X of $G[D^*]$ such that $X \subseteq D^* \cap N_G(u)$, generates with polynomial delay a family of minimal dominating sets $\mathcal{D}$ with the property that $\mathcal{D}$ contains all minimal dominating sets D that are children of D^* with respect to flipping u and X. Then all minimal dominating sets of G can be enumerated in incremental polynomial time.*

In the next two sections, we will show that there is indeed an algorithm as algorithm $\mathcal{A}$ described in the statement of Lemma 8.1 when the input graph G is a chordal bipartite or unit square graph.

8.2 Chordal Bipartite Graphs

Let $G := (V_1, V_2, E)$ be a chordal bipartite graph and let us colour the vertices in V_1 in red and the vertices in V_2 in blue. The following, of independent interest, is used in the construction of $\mathcal{A}$ as required by Lemma 8.1.

Proposition* 8.2** ([77])**.** *All minimal* ***Red *dominating sets of G can be enumerated with delay $O(n \cdot \min\{m \log n, n^2\})$.*

As a corollary, we can indeed enumerate with polynomial delay all the minimal total dominating sets of G since if D is a minimal total dominating set, then $D \cap V_1$ is a minimal **Red** dominating set and $D \cap V_2$ is a minimal **Blue** dominating set.

***Corollary* 8.3** ([77])**.** *All minimal total dominating sets of G can be enumerated with delay $O(n \cdot \min\{m \log n, n^2\})$.*

Let us now explain the algorithm $\mathcal{A}$ as required by Lemma 8.1. Let $v \in D$. First of all, we should output $\mathcal{D} = \emptyset$ and stop if there is $x \in N_G(v)$ such that $N_G(x) = \{v\}$, because in this case one can see that D^* has no child. Assume from now on that this is not the case and let $u \in N_G(v) \cap (V_G \setminus D^*)$.

Let $R \subseteq (N_G(u) \setminus \{v\}) \cap D^*$ be the set of all vertices $x \in (N_G(u) \setminus \{v\}) \cap D^*$ such that $x \in P(D^*, x)$, *i.e.*, these vertices are not dominated by other vertices of D^*. Denote by B the set of all vertices $y \in N_G(R \cup \{v\}) \setminus \{u\}$ such that $N_G(y) \cap D^* \subseteq R \cup \{v\}$. Observe that by the definition of R, $B \cap D^* = \emptyset$. Notice also that R and B are subsets of distinct sets of the bipartition of G. Without loss of generality we assume that $R \subseteq V_1$ and $B \subseteq V_2$. Let $R' = N_G(B) \setminus \{v\}$. Clearly, $R \subseteq R' \subseteq V_1$. Consider the red-blue bipartite graph $F = G[R' \cup B]$, where R' and B are the sets of red and blue vertices respectively. Using Proposition 8.2 we enumerate all minimal **Red** dominating sets of F.

For each minimal red dominating set X of F, we consider the (not necessarily minimal) dominating set $D' = (D^* \setminus (R \cup \{v\})) \cup (X \cup \{u\})$ of G. We apply greedy removal of vertices to obtain a minimal dominating set D from D', and output D.

Denote by $\mathcal{D}$ the collection of generated sets. The set $\mathcal{D}$ can be of course enumerated with delay $O(n^3)$. We proved in [77] that all elements of $\mathcal{D}$ are pairwise distinct and contains all children of D^* with respect to flipping u and R, yielding the following.

Theorem 8.4 ([77])**.** DOM-ENUM(*chordal bipartite graphs*) *is in* **IncP**.

8.3 Unit Square Graphs

Let $G := (V, E)$ be a unit square graph. In order to prove the existence of the algorithm $\mathcal{A}$ as required by Lemma 8.1, we first proved the following which says that unit square graphs have *locally polynomially linear bounded neighbourhood.*

***Proposition* 8.5** ([75])**.** *For every positive integer r and every vertex u of G, there is a linear layout of $N_G^r[u]$ of* $\text{neigh}_{G'}$*-width at most n^{r^2} with $G' := G[N_G^r[u]]$. Moreover, if a realization $f : V \to \mathbb{Q}^2$ of G is given, then a linear layout of the vertices of* $\text{neigh}_{G'}$*-width at most $O(n^{r^2})$ can be constructed in polynomial time.*

What follows is valid for any graph that has locally polynomially linear bounded neighbourhood and also satisfies that we can enumerate the independent sets on the neighbourhood of a vertex in polynomial time. Indeed, we showed that we can construct $\mathcal{A}$ by a reduction to the enumeration of minimal **Red** dominating sets in an auxiliary coloured induced subgraph of $G[N_G^3[u]]$.

Let D^* be a minimal dominating set of G such that $G[D^*]$ has an isolated vertex. Let also $u \in V_G \setminus D^*$ and X is a non-empty independent set of $G[D^*]$ such that $X \subseteq D^* \cap N_G(u)$. Consider the set $D' := (D \setminus X) \cup \{u\}$. Denote by **Blue** the set of vertices that are not dominated by D'. Notice that $\mathbf{Blue} \subseteq N_G(X) \setminus N_G[u]$. Therefore, $\mathbf{Blue} \subseteq N_G^2[u]$. Let $\mathbf{Red} := N_G(\mathbf{Blue}) \setminus N_G[X]$. Clearly, $\mathbf{Red} \subseteq N_G^3[u]$. We construct the coloured graph $H := G[\mathbf{Red} \cup \mathbf{Blue}]$. Let $\mathcal{A}'$ be an algorithm that enumerates minimal **Red** dominating sets in H. Assume that if $\mathbf{Blue} = \emptyset$, then $\mathcal{A}'$ returns $\emptyset$ as the unique **Red** dominating set. We constructed $\mathcal{A}$ as follows.

Step 1. If $\mathcal{A}'$ returns an empty list of sets, then $\mathcal{A}$ returns an empty list as well.

Step 2. For each **Red** dominating set R of H, consider $D'' := D' \cup R$ and construct a minimal dominating set D from D'' by greedy removal.

We again proved in [75] that $\mathcal{A}$ generates all minimal dominating sets that are children of D^* with respect to flippping u and X, which proves the following.

Theorem 8.6 ([75])**.** DOM-ENUM(*unit square graphs*) *is in* **IncP**.

8.4 Concluding Remarks

We used the flipping method developed in [78] in order to show that DOM-ENUM(chordal bipartite graphs $\cup$ unit square graphs) is in **IncP**. The flipping method seems to be a nice tool, but we need to transform it into a polynomial space or even a polynomial delay algorithm. For that purposes we can try the *reverse search algorithm* [5] that was used in the past to get a polynomial space or a polynomial delay algorithm [5, 107, 126], that we explain in the next lines.

Assume we want to enumerate all the elements of a set $\mathcal{Z}$ that is a subset of an implicitly given set $\mathcal{X}$. Assume that we have a polynomial time computable function $P : \mathcal{X} \to \mathcal{X} \cup \{nil\}$. For each $X \in \mathcal{X}$, $P(X)$ is called the *parent* of X, and the elements Y such that $P(Y) = X$ are called the *children* of X. The parent-child relation of P is *acyclic* if any $X \in \mathcal{X}$ is not a proper ancestor of itself, that is, it always holds that $X \neq P(P(\cdots P(X))\cdots)$. We say that an acyclic parent-child relation is *irredundant* when any $X \in \mathcal{X}$ has a descendant in $\mathcal{Z}$, in the parent-child relation. The *depth* of an acyclic parent-child relation P is the size of the longest chain between nil and an element of $\mathcal{Z}$. Examples of irredundant parent-child relations can be found in the literature [5, 148, 118]. The following statements are well-known in the literature.

***Proposition* 8.7.** *All elements in $\mathcal{Z}$ can be enumerated with polynomial space if there is a polynomial depth acyclic parent-child relation $P : \mathcal{X} \to \mathcal{X} \cup \{nil\}$ such that there is a polynomial space algorithm for enumerating all the children of each $X \in \mathcal{X} \cup \{nil\}$.*

***Proposition* 8.8.** *All elements in $\mathcal{Z}$ can be enumerated with polynomial delay and polynomial space if there is a polynomial depth irredundant parent-child relation $P : \mathcal{X} \to \mathcal{X} \cup \{nil\}$ such that there is a polynomial delay polynomial space algorithm for enumerating all the children of each $X \in \mathcal{X} \cup \{nil\}$.*

Algorithm ReverseSearch(X)
1. **if** $X \in \mathcal{Z}$ **then output** X
2. **for each** child Y of X **call** ReverseSearch(Y)

Figure 8.1: Reverse Search Algorithm.

With an acyclic (resp., irredundant) parent-child relation $P : \mathcal{X} \to \mathcal{X} \cup \{nil\}$, Algorithm 8.1 enumerates all the elements in $\mathcal{Z}$, with polynomial space (resp., with polynomial delay and polynomial space).

Can we use this technique to turn the flipping method into a polynomial space, polynomial delay algorithm? at least in the considered cases?

Chapter 9

Berge's Algorithm

Berge's algorithm [7] is one of the classical algorithms for enumerating minimal transversals in hypergraphs and several versions have been used in the past to show that TRANS-ENUM is tractable in some special classes of hypergraphs. In this chapter we present the variant given in [100] which enabled us to prove that the enumeration of minimal edge dominating sets in graphs is in **DelayP**. It is worth noticing that the problem was known to be in **IncP** since we showed in [96] that closed neighbourhood hypergraphs of line graphs are 6-conformal, but an independent proof was also given by Golovach et al. in [78] by using the flipping method. We also point out that there is no known tool to prove that an enumeration problem does not admit a polynomial delay algorithm, while there exist tools for proving that it does not admit an output or incremental polynomial time algorithm. Therefore, it is interesting to identify problems where a polynomial delay algorithm is known, even though the delay may be of no practical interest.

9.1 Berge's Algorithm

Let $\mathcal{H} := (V, \mathcal{F})$ be a hypergraph and let us enumerate the hyperedges of $\mathcal{H}$ as $F_1, \ldots, F_m$. For every $1 \le i \le m$, let $\mathcal{H}_i$ be the hypergraph with hyperedges $\{F_1, \ldots, F_i\}$. Berge's algorithm consists in computing $tr(\mathcal{H}_i)$ from $tr(\mathcal{H}_{i-1})$ for each $2 \le i \le m$ and this can be clearly done in time polynomial from $tr(\mathcal{H}_{i-1})$. Berge's algorithm is not output polynomial because at some intermediate step i the set $tr(\mathcal{H}_i)$ can be much larger than the set $tr(\mathcal{H})$ and indeed it is proved that there exist hypergraphs for which Berge's algorithm is not output polynomial for any ordering of their hyperedges [151]. Moreover, Berge's algorithm is not polynomial space, but this can be remedied. Berge's algorithm follows a tree of height m and the nodes at level i corresponds to $tr(\mathcal{H}_i)$. Now, instead of performing a Breadth-First Search (BFS for short) of this tree, one can traverse the tree in a Depth-First Search (DFS for short) manner, which allows a polynomial space. For that purposes, one defines the following parent-child relation. For $i > 1$ and $T \in tr(\mathcal{H}_i)$, we define the parent $Q'(T, i)$ of T as follows

$$Q'(T,i) := \begin{cases} T & \text{if } T \in tr(\mathcal{H}_{i-1}), \\ T \setminus \{v\} & \text{if } v \text{ is such that } P_{\mathcal{H}_i}(T,v) = \{F_i\}. \end{cases}$$

We can observe that $T \notin tr(\mathcal{H}_{i-1})$ if and only if $P_{\mathcal{H}_i}(T, v) = \{F_i\}$ holds for some $v \in T$, thus the parent is well defined and is always in $tr(\mathcal{H}_{i-1})$ [107, 126]. One can moreover compute the parent of any $T \in tr(\mathcal{H}_i)$ in time polynomial in $\|\mathcal{H}\|$ and since a child is obtained by adding at most one vertex, then the children of any $T \in tr(\mathcal{H}_i)$ can be also listed in polynomial time. The tree induced by the parent-child relation spans all the members of $\bigcup_{1 \le i \le m} tr(\mathcal{H}_i)$, and since the parent-child relation Q' is acyclic the reverse-search algorithm (see Algorithm 8.1) can be used to enumerate all the minimal transversals of hypergraphs with polynomial space. However, Q' is not irredundant and then does not guarantee a polynomial delay neither an output polynomiality. Indeed, we can expect that the size of $tr(\mathcal{H}_i)$ increases as i increases, and it can be observed in practice [126]. But, $tr(\mathcal{H}_i)$ can be exponentially larger than $tr(\mathcal{H})$, thus Berge's algorithm is not output polynomial [151] in general.

One idea to avoid the lack of irredundancy is to certify the existence of minimal transversals in the descendants. Suppose that we choose some levels $1 = l_1, \ldots, l_k = m$ of Berge's algorithm, and state that for any $T \in tr(\mathcal{H}_{l_j})$, we have at least one descendant in $tr(\mathcal{H}_{l_{j+1}})$. This implies that any transversal in $tr(\mathcal{H}_{l_j})$ has a descendant in $tr(\mathcal{H})$, thus we can have an irredundant parent-child relation by looking only at these levels, and the enumeration can be polynomial delay and uses polynomial space. We adapted this idea in the case of minimal edge dominating sets.

9.2 Enumeration of Minimal Edge Dominating Sets

Let $G := (V, E)$ be from now on a fixed graph. For a vertex x, we denote by $\widetilde{N}(x)$ the set of edges incident to x. For every edge xy, we denote by $N[xy]$ the set $\widetilde{N}(x) \cup \widetilde{N}(y)$. For $E' \subseteq E$, we let $\mathcal{H}(E')$ be the hypergraph $(E, \{N[e] \mid e \in E'\})$. An edge dominating set in a graph G is a subset D of E_G such that any edge is either in D or is adjacent to some edge in D. In other words, D is an edge dominating set if for every edge e of G, we have $N[e] \cap D \neq \emptyset$. We are interested in enumerating the set of (inclusion-wise) minimal edge dominating sets in G, which by definition corresponds to $tr(\mathcal{H}(E))$. By abuse of notation we will write $f \in P_{\mathcal{H}(E')}(T, e)$ instead of $N[f] \in P_{\mathcal{H}(E')}(T, e)$.

Let $\{b_1, \ldots, b_k\}$ be a maximal matching of G, and let $b_i = x_i y_i$. For each $0 \leq i \leq k$, let $V_i := V \setminus \left(\bigcup_{i' > i} b_{i'} \right)$, and let $E_i := \{e \mid e \subseteq V_i\}$). Let $B_i := E_i \setminus E_{i-1}$ for $i > 1$. Note that any edge in E_1 is adjacent to b_1 and by definition B_i never includes an edge $b_j \neq b_i$. Without loss of generality, we here assume that we have taken a linear ordering $\leq$ on the edges of G so that: (1) for each $e \in B_i$ and each $f \in E_{i-1}$ we have $f < e$, (2) for each $e \in \widetilde{N}(x_i) \cap B_i$, each $f \in \widetilde{N}(y_i) \cap B_i$ we have $b_i < e < f$. Observe that with that ordering we have $e < f$ whenever $e \in B_i$ and $f \in B_j$ with $i < j$. We consider that Berge's algorithm on $\mathcal{H}(E)$ follows that ordering. In fact we will prove using Berge's algorithm that we can define an irredundant parent-child relation to enumerate $tr(\mathcal{H}(E_i))$ from $tr(\mathcal{H}(E_{i-1}))$.

***Lemma* 9.1** ([100])**.** *Let $1 \leq i < k$ and $T \in tr(\mathcal{H}(E_{i-1}))$. Then $T \cup \{b_i\} \in tr(\mathcal{H}(E_i))$ and is a descendant of T.*

Given $T \in tr(\mathcal{H}(E_i))$, we define the *skip-parent* $Q(T, i)$ of T as $Q'(Q'(\cdots(T, |E_i|), |E_i| - 1), \cdots, |E_{i-1}| + 1)$ which corresponds to the ancestor of T in $tr(\mathcal{H}(E_{i-1}))$. The goal now is to show that one can enumerate the skip-children with polynomial delay and use polynomial space.

Let T be in $tr(\mathcal{H}(E_{i-1}))$ and $T' \in tr(\mathcal{H}(E_i))$ a skip-child of T. First notice that every edge in $T' \setminus T$ can have a private neighbour only in B_i. Indeed every edge in E_{i-1} is already dominated by T and an edge in $T' \setminus T$ is only used to dominate an edge in B_i. Moreover, an edge $e \neq b_i$ in $\widetilde{N}(x_i) \cap (T' \setminus T)$ (resp. in $\widetilde{N}(y_i) \cap (T' \setminus T)$) can only have private neighbours in $\widetilde{N}(x_i) \cap B_i$ (resp. $\widetilde{N}(y_i) \cap B_i$). Also, if $b_i \in T' \setminus T$ then $T' \setminus T = \{b_i\}$. From this one can enumerate with polynomial delay and polynomial space all skip-children T' of T such that $T' \setminus T \subseteq B_i$.

We now consider the remaining case that an edge in $T' \setminus T$ is not adjacent to b_i. We call such a skip-child *extra*. We can see that at least one edge $f \neq b_i$ adjacent to b_i has to be included in T' to dominate b_i. Actually, since $b_i < e$ for any $e \in B_i \setminus \{b_i\}$, any extra skip-child of T is a descendant (*w.r.t.* Berge's algorithm) of some $T \cup \{f\}$ with $f \neq b_i$ incident to x_i or y_i in the original parent-child relation. So, without loss of generality, we will assume that such an edge $f \neq b_i$ is incident to x_i and is included in T. Hereafter, we suppose that $N(y_i) := \{z_1, \ldots, z_k\}$ and assume T' is an extra skip-child of T.

A vertex $z_h \in N(y_i) \cap V_i$ is *free* if it is not incident to an edge in T, and is *non-free* otherwise. A free vertex is said to be *isolated* if it is not incident to an edge in E_{i-1}. Clearly, if there is an isolated free vertex, then T has no extra skip-child. Thus, we assume that there is no isolated free vertex. Edges in $E_i \setminus B_i$ that are incident to some free vertices are called *border edges.* Observe that any border edge vz_h incident to a free vertex z_h is adjacent to an edge $vw \in T$ if $v \in V_{i-1}$. The set of border edges is denoted by $Bd(T, i)$. Note that no edge in $Bd(T, i)$ is incident to two free vertices, otherwise the edge is in E_{i-1} but not dominated by T, and then any border edge is incident to exactly one free vertex. We can see that an edge of B_i incident to y_i is not dominated by T if and only if it is incident to a free vertex, and any edge in $T' \setminus T$ that is not incident to x_i is a border edge. Then, for any border edge set $Z \subseteq Bd(T,i)$, $T \cup Z \in tr(\mathcal{H}(E_i))$ only if each free vertex has a border edge $e \in Z$ incident to it. Since any border edge is incident to exactly one free vertex, for any $Z \subseteq Bd(T,i)$ such that $T \cup Z$ is irredundant and for any edge $vz_h \in Z$ with free vertex z_h, $P_{\mathcal{H}(E_i)}(T \cup Z, e)$ is always $\{vz_h\}$. This implies that $T \cup Z$ is in $tr(\mathcal{H}(E_i))$ only if $Z \subseteq Bd(T,i)$ includes exactly one edge incident to each free vertex. We call such an edge set Z a *selection*. It is clear therefore that $T \cup Z \in tr(\mathcal{H}(E_i))$ with $Z \cap T = \emptyset$ only if Z is a selection. One can hope a characterisation of selections Z such that $T \cup Z \in tr(\mathcal{H}(E_i))$, but, checking whether there is such a selection Z is NP-complete.

Theorem 9.2 ([100])**.** *Given $T \in tr(\mathcal{H}(E_{i-1}))$, it is NP-complete to check whether there is a selection Z such that $T \cup Z \in tr(\mathcal{H}(E_i))$.*

In order to overcome this difficulty, we identified a pattern, that we called an *H-pattern*, that makes the problem difficult. A vertex set $\{z_\ell, v_\ell, z_j, v_j\}$ is an *H-pattern* if z_ℓ and z_j are free vertices, $v_\ell v_j$ is in T, and $v_\ell v_j$ has two non-border private neighbours in $E_{i-1} \setminus T$: one is adjacent to v_ℓ and the other to v_j. We also say that the edges $z_\ell v_\ell$, $z_j v_j$ and $v_\ell v_j$ induces an H-pattern.

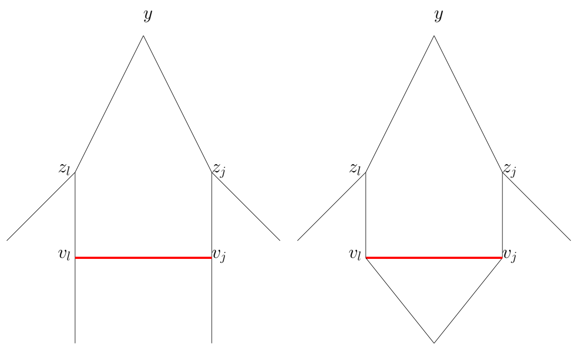

Figure 9.1: Examples of H-patterns.

H-patterns are illustrated in Figure 9.1. We will see that the NP-completeness comes from the presence of H-patterns. Indeed, for an H-pattern $\{z_\ell, v_\ell, z_j, v_j\}$, any private neighbour of $v_\ell v_j$ is adjacent to either $z_\ell v_\ell$ or to $z_j v_j$, thus we cannot add both to a selection Z since in that case $P_{\mathcal{H}(E_i)}(T \cup Z, v_\ell v_j)$ will be empty.

An edge $e \in T$ is called *redundant* if all edges in $P_{\mathcal{H}(E_{i-1})}(T,e)$ are border edges and no edge $y_i z_h$ is in $P_{\mathcal{H}(E_i)}(T,e)$. Let $X_T := \{e \in Bd(T,i) \mid \exists e' \in T \text{ and } P_{\mathcal{H}(E_i)}(T \cup \{e\}, e') \subseteq Bd(T,i)\}$. The addition of any edge $e \in X_T$ to T transforms an edge e' of T into a redundant one with respect to $T \cup \{e\}$. Let H_T be the set of border edges included in an H-pattern.

***Lemma* 9.3** ([100])**.** *$T \cup Z \in tr(\mathcal{H}(E_i))$ for any selection $Z \subseteq Bd(T,i) \setminus (X_T \cup H_T)$ if and only if T has no redundant edges.*

Lemma 9.3 demonstrate how to construct minimal transversals $T \subseteq T'$ not intersecting H_T, but some generated may not be extra skip-children of T. Such redundancies happen for example when two edges f_1 and f_2 in T' have private neighbours only in B_i, but after the removal of either one from T', the other will have a private neighbour outside B_i. Assuming in this case that $f_1 \in T$ and $f_2 \in T' \setminus T$, it holds that T' can be generated from T or from $(T \setminus \{f_1\}) \cup \{f_2\}$. And since the number of selections Z such that $T \cup Z \in tr(\mathcal{H}(E_i))$ can be arbitrarily large, we need to avoid such redundancies. To address this issue, we characterise the edges not to be added to selections Z such that $T \cup Z$ is an extra skip-child of T.

We say that a border edge vz_ℓ is *preceding* if there is an edge vz_h in T satisfying $P_{\mathcal{H}(E_{i-1})}(T, vz_h) \subseteq N[vz_\ell]$ and $y_i z_\ell < y_i z_h$, and denote the set of preceding edges by X'_T. We also say that an edge $vz_h \in T$ is *fail* if $P_{\mathcal{H}(E_{i-1})}(T, vz_h) \subseteq Bd(T,i)$, $y_i z_h$ is in $P_{\mathcal{H}(E_i)}(T, vz_h)$, and no edge $wz_\ell \in P_{\mathcal{H}(E_{i-1})}(T, vz_h)$ satisfies $y_i z_h < y_i z_\ell$.

***Lemma* 9.4** ([100])**.** *Suppose that T has neither redundant edges nor fail edges and any free vertex is incident to an edge in $Bd(T,i)$. Then, $T \cup Z$ with $T \cap Z = \emptyset$ is an extra skip-child of T including no edge of H_T if and only if Z is a selection including no edge of $X_T \cup X'_T \cup H_T$.*

With Lemma 9.4 one can enumerate with polynomial delay and polynomial space all the extra skip-children of T not intersecting H_T. Since, it is hard to enumerate, from T, all the extra skip-children of T intersecting H_T, we will define another parent-child relation which will help in listing the extra skip-children intersecting H_T.

For two sets S and S' of edges we say that S is *lexicographically smallest than* S' if $\min(S \Delta S') \in S$. Hereafter, we consider an extra skip-child $T' = T \cup Z$ of $T \in tr(\mathcal{H}(E_{i-1}))$ such that $T' \cap H_T \neq \emptyset$. Let $H^*(T') := \{v_h z_h, v_\ell z_\ell, v_h v_\ell\}$ be the lexicographically minimum H-pattern among all H-patterns of T that includes an edge of Z. Without loss of generality, we assume that $v_\ell z_\ell$ is in Z. Let uz_h be the edge in Z incident to z_h. Notice that such an edge exists because z_h is a free vertex. Then, we define the *slide-parent* of T' by $T' \cup \{v_h z_h\} \setminus \{uz_h, v_h v_\ell\}$.

***Lemma* 9.5** ([100])**.** *The slide-parent of T' is well-defined and is a member of $tr(\mathcal{H}(E_i))$.*

The slide-parent of T' has less edges than T', thus the (slide-parent)-(slide-child) relationship is acyclic, and for each $T' \in tr(\mathcal{H}(E_i))$, there is an ancestor $T'' \in tr(\mathcal{H}(E_i))$ in the (slide-parent)-(slide-child) relation such that the slide-parent of T'' has no H-pattern, which means that by following the (slide-parent)-(slide-child) relation one can enumerate all the extra skip-children of T intersecting H_T. Further, any slide-child is obtained from its slide-parent by adding two edges and removing one edge, and then the slide-children of any $T \in tr(\mathcal{H}(E_i))$ can be enumerated with polynomial delay and polynomial space with the reverse-search algorithm.

By combining all these steps, we can state the following.

Theorem 9.6 ([100])**.** *All edge minimal dominating sets in a graph G can be enumerated with polynomial delay and polynomial space.*

9.3 Concluding Remarks

We modified the traversal tree from Berge's algorithm so that we can ensure that any intermediate solution will give rise to a final solution, overcoming the main drawback of Berge's algorithm. However, with this new traversal we faced a difficult problem which consists in listing solutions intersecting H-patterns. To cope with the enumeration of these difficult problems we proposed a new parent-child relation that is totally independent from Berge's traversal route (the (slide-parent)-(slide-child) relation). How can we adapt this new idea to other enumeration problems? Interesting future works are applications of this idea to other kind of enumeration algorithms, *e.g.*, the one used by Lawler et al. for enumerating maximal subsets [114] or other algorithms for enumerating minimal transversals (see for instance [56] or the flipping method introduced in Chapter 8). It would be also interesting to apply this technique for DOM-ENUM in other graph classes, *e.g.*, some subclasses of claw-free graphs or weakly chordal graphs.

Chapter 10

Conclusion and Perspectives

10.1 (Linear) Rank-Width

We extended the notion of rank-width to edge-coloured graphs and were able to generalise all the results known in the case of undirected graphs. The main objectives in this research line are (1) to prove that σ-symmetric graphs are well-quasi-ordered by pivot-minor, and (2) to test in polynomial time whether a fixed σ-symmetric graph H is a pivot-minor of a given σ-symmetric graph G. Both objectives are intimately related and are in the near future unreachable as the only techniques to tackle them are to mimic Robertson and Seymour's proof. Indeed, following Robertson and Seymour, and also Geelen, Gerards and Whittle, in order to answer these questions we need to understand the structure of graphs that do not contain a fixed graph H as a pivot-minor. But for these we lack several interesting properties, in particular graph/matroid minors have a local counterpart, while pivot-minor does not seem to have any such local property, except that we know if H is a pivot-minor of G, then A_H can be obtained from A_G by applying a principal pivot transform once. So, understanding graphs without a fixed graph H as a pivot-minor turns out to be equivalent to understanding matrices without some fixed principal minor in their equivalence class (*w.r.t.* principal pivot transform), *i.e.*, we go back to representable $\triangle$-matroids. The study of representable $\triangle$-matroids were initiated by Bouchet and not so much is done since, and we now probably need to think in terms of linear algebra (which is confirmed by the results on the well-quasi-ordering of graphs of bounded rank-width), instead of only in terms of structure of graphs.

In parallel, we started to study the structure of graphs of bounded linear rank-width, and in particular the question whether trees are pivot-minor obstructions for bounded linear rank-width. Unfortunately, the question is still wide open as we even do not know whether graphs excluding a fixed path as a vertex/pivot-minor have bounded linear rank-width. We aim at looking again at this question by studying $\triangle$-matroids as we consider trees (or distance-hereditary graphs) as a basis case for studying graphs excluding a fixed (circle) graph as a pivot-minor.

We also looked at the sizes of obstructions for linear rank-width and in collaboration with O-J. Kwon we published in arxiv [93] a manuscript giving a doubly exponential upper bound on the sizes of pivot-minor obstructions. Unfortunately, there is a flaw in the proof that we still fail to correct. If the result turns out to be true this would give also a doubly exponential upper bound on the sizes of matroid minor obstructions for matroid path-width. We conjecture that the bound given in the manuscript is true, and we will be able in the short term to correct the proof or conclude that the proof technique in that manuscript is hopeless.

The techniques we used to study the linear rank-width of distance-hereditary graphs seem to be applicable to graphs such that each induced prime graph has bounded linear rank-width. A recursive characterisation of graphs of linear rank-width k for graphs of bounded rank-width is unlikely to be obtained, except probably for graphs of rank-width 2 (to our knowledge no such characterisation exists for the path-width of graphs of bounded tree-width). However, we conjecture that one can compute in polynomial time the linear rank-width of graphs of bounded rank-width. One way of doing it is to propose, as in [10], a $f(\ell)\cdot 2^{O(k)}\cdot n^{O(1)}$

time algorithm for checking whether a given graph of rank-width ℓ has linear rank-width at most k. Then, since a graph of rank-width ℓ has linear rank-width at most $\ell \cdot log(n)$, it suffices to call that algorithm for all $k \leq \ell \cdot \log(n)$, to obtain an $O(n^{\ell})$ time algorithm (and that is roughly the only known algorithm for the computation of the path-width of graphs of bounded tree-width [10]).

Another research line that we did not investigate is the (L)RW($\leq k$) VERTEX-DELETION problem which asks whether we can remove at most ℓ vertices to obtain a graph of (linear) rank-width at most k. We know from the two papers [30, 41] that it is FPT parametrised by $k+\ell$. However, as it is based on the meta-theorem in [30] the constant is as usual too gigantic and one would want to obtain a reasonable one, say a single exponential. We investigated in [92] the LRW(≤ 1) VERTEX-DELETION and obtained a single exponential FPT algorithm and a polynomial kernel. We aim at pursuing this question. Right now the techniques in [92] are unlikely to be extendable since big cycles were the main obstacles to deal with in [92]. Maybe the protrusion technique [109] can be adapted to our situation (at least the definitions have natural counterparts), but rank-width is radically different from tree-width as for instance we do not know whether excluding a bipartite circle graph as a vertex-minor yields a bounded rank-width (excluding a planar graph was crucial in [109]). Restricting first to circle or line graphs, as toy cases of course, is probably the right thing to do and then look at the finitely representable matroids which admit more desirable properties.

10.2 Enumeration

For the enumeration of minimal dominating sets we acquired a lot of experience in techniques and their combinations. We proved that TRANS-ENUM, TDOM-ENUM and DOM-ENUM are **DelayP**-equivalent and from [122] we know also that it is the case for the enumeration of d-dominating sets. It would be interesting to have a dichotomy or trichotomy theorem for the enumeration of (σ,ρ)-dominating sets, *i.e.*, for which pairs (σ,ρ) do we have TRANS-ENUM **DelayP**-equivalent to DOM-ENUM? for which pairs (σ,ρ) do we have the associated enumeration in **TotalP**?

We also know that DOM-ENUM (co-bipartite graphs) is **DelayP**-equivalent to DOM-ENUM and for many graph classes $\mathcal{C}$ we now know that DOM-ENUM ($\mathcal{C}$) is in **TotalP**. It would be interesting to identify other graph classes (incomparable to co-bipartite graphs) that are equivalent to DOM-ENUM. As DOM-ENUM (chordal graphs) is in **DelayP** and DOM-ENUM (graphs with girth ≥ 7) is in **IncP**, we know that the presence of short cycles makes the problem difficult. Because co-bipartite graphs are C_5-free, a first step consists maybe in clarifying the complexity status for graphs with girth 5 and 6.

The best algorithm for DOM-ENUM is the quasi-polynomial time algorithm for TRANS-ENUM. The reduction from TRANS-ENUM to DOM-ENUM would be more valuable if, failing at proving that DOM-ENUM is in **TotalP**, we can use it to give a better algorithm than the Khachiyan and Fredman's one. There are at least two ways of doing so: either study carefully the structure of minimal dominating sets in co-bipartite graphs and use it to derive a better algorithm, or find other graph classes that are better structured than co-bipartite graphs. In both cases we will gain in understanding the problem. The possibility that DOM-ENUM is not in **TotalP** (under some complexity hypothesis) should be also considered as it has been open for a while. However, notice that if such a case is to be considered, we must consider at least hypothesis like *ETH* or *SETH*. Is it reasonable to consider DOM-ENUM as a complete problem in a complexity class (to be defined)?

The links between FPT theory and enumeration are not yet (enough) investigated and from [43] we can consider that we have a strong definition of what a parameterised enumeration is. We have several (hyper)graph classes with a natural parameter (bounded degenerate graphs, bounded conformal graphs, etc.) and such that DOM-ENUM were proved to be in **IncP** or in **DelayP** but the exponent of the polynomial depends on the parameter. How can we prove that we cannot have an FPT delay? For instance, we conjecture that DOM-ENUM(k-degenerate graphs) does not admit an FPT-delay enumeration algorithm, but we need techniques to answer such questions.

During my PhD in collaboration with B. Courcelle and C. Gavoille [37] we considered the enumeration of the solutions of a First-Order formula in graphs of bounded local clique-width, and we were only able to give an intermediate solution similar to the one given by Frick [66] for graphs of bounded local tree-width. Graphs of bounded local clique-width are somehow intriguing for me as while the definition is well-suited for checking FO formulas (once Courcelle's theorem and Gaifman's theorem are known), we do not know how to decompose them for the enumeration, in particular we do not know how to bound, by a constant, the neighbourhoods where to look and hence none of Gaifman or Frick's decomposition of FO formulas can be used [66] (in [37] the subclass we considered allowed to bound the neighbourhoods to look into). The game theoretical tools used in [82] to propose an FPT polynomial time algorithm for checking FO formulas on nowhere dense graphs is different from the others in the spirit and even though I still fail to understand it deeply I had the impression (informally and with the claim that on bounded expansion an r-neighbourhood cover of bounded spread can be constructed) that it can be adapted to enumerate/count the solutions of FO formulas in at least graphs of bounded expansion (a result already proved in [108] by using one of the numerous decompositions of bounded expansion graphs). How can we adapt the machineries used in [82, 108] to enumerate solutions of an FO formula in graphs of bounded local clique-width? A restriction to a subset of FO formulas is may be a first step, *e.g.*, limiting the alternation of quantifiers? the number of first-order variables? Can we use the notion of *fraternal augmentation* [127] to show that on graphs of bounded local clique-width an r-neighbourhood cover of bounded spread can be constructed?

In the Exact Exponential Algorithms community many algorithms are based on giving an upper bound on the number of possible objects and then prove that we can enumerate the set of objects in the same time bound (input-sensitive algorithms). For instance, in order to compute the *domatic number* of a graph Fomin et al. proposed in [63] an input-sensitive algorithm for enumerating minimal dominating sets which gives an upper bound of 1.7159^n (surprisingly this bound is the best known so far and the best lower bound is $\frac{15}{6}^n$). Such bounds are usually obtained by a fine analysis of branch&reduce algorithms. When we have a tight bound on a number of objects and dispose at the same time of an output polynomial algorithm for enumerating them, then this algorithm can be used as a black box as we know an upper bound on the running time, and this is commonly used in several Exact (Exponential) Algorithms [64]. It is then natural to think about the links between the two communities, and our freshly funded grant GraphEn by ANR[1] aims at considering such links. For instance, the polynomial delay algorithm for Dom-Enum(interval graphs) given in [98] was used in [76] to give a tight bound on the number of minimal dominating sets in interval graphs, and the linear delay algorithm given in Section 6.1 for Dom-Enum(split graphs) were more or less used in [42] to also give a tight upper bound on the number of minimal dominating sets in split graphs. What do all such algorithms have in common? Can we identify techniques in the enumeration community to propose input-sensitive algorithms giving tight bounds? And vice-versa? It is worth noticing that Ruskey discusses in [145, Page 98] the common way to transform dynamic programming algorithms into enumeration algorithms which consists in analysing the recursive procedures and writes recurrences. Such recursive enumeration algorithms are likely to give tight bounds whenever the recurrences are *accurate* (a toy example is a recurrence relation for the set of minimal dominating sets in co-graphs using the co-tree, which also gives the tight bound for co-graphs). The most challenging objects are probably the minimal dominating sets as we do not have neither a tight bound nor an output polynomial algorithm, and the minimal feedback vertex sets as we know an output polynomial algorithm but a tight bound is not known and the current techniques seem to be not sufficient.

We know that once we can count exactly, then the enumeration is easy. However, counting is much more difficult than enumeration as for instance we can enumerate "easily" matchings, but the counting is #P-complete. In collaboration with T. Uno (in preparation) we proved that in almost all situations (where the enumeration is known to be output-polynomial), counting the number of minimal dominating sets is also #P-complete. However, for some counting problems we know how to (randomly) count within an ε (see for instance [62, 67, 149]), and one natural question is whether such an approximation counting can be turned into an enumeration algorithm (usually such a counting algorithm can be turned into a good sampling algorithm [62]). What about also enumerating (or counting) the set of ε-minimal sets, *i.e.*, those sets X such that a minimal one can be obtained after deleting at most $\varepsilon|X|$ vertices?

[1]The GraphEn project funded by the National Agency for Research (ANR), deals with enumeration algorithms for objects in (hyper)graphs from input and output-sensitive views, and the members are people from Orleans, Metz, Bordeaux and Clermont-Ferrand. It is leaded by D. Kratsch and I am the local coordinator in Clermont-Ferrand.

10.3 Links with Lattice Theory

For the last three years I have been also interested in convexity spaces, particularly those arising from paths in graphs. A *convexity space* on a ground set V is a subset of 2^V that is closed under intersection, and its elements are called *convex sets*. Over the last years path convexities, *i.e.*, those arising from paths in graphs [137] have been studied and people have been mostly interested in computing parameters like the *Helly number*, the *Radon number*, the *hull number*, the *rank*, etc. In collaboration with L. Nourine we gave in [102] a polynomial time algorithm for computing a minimum hull set in chordal graphs by using tools from database theory, a question open since the introduction of the notion of hull number by Farber and Jamison in 1986 [59]. We also initiated a collaboration with J.L. Szwarcfiter on the rank of the geodetic convexity [105].

We think that the links between structural graph theory and lattice theory have not been (to our knowledge) enough investigated and lattice theory can be maybe used to explain why some parameters are easy to compute. For instance, I am not aware of any paper using the fact that tree-width can be characterised in terms of a specific lattice (the lattice of completions into a chordal graph), and this deserves to be investigated. For instance, there is a closed link between lattice theory and coding theory, the latter being closely related to matroid theory. In the near future we are considering looking at graph problems using lattice theory, in particular how can we take advantage of the lattice of bicliques (or neighbourhoods) for covering and partitionning problems? I will advise in collaboration with L. Nourine a PhD thesis (an MENRT funding and starting next 10/01/2015) and the objective is to consider such questions using matrix decompositions.

Chapter 11

List of Presented Papers

All the papers are either accessible from arxiv or from my webpage (items correspond to numbers in references). Please check the CV for the complete list of publications and other information about invitations and supervisions.

(1) I. Adler, M.M. Kanté. Linear Rank-Width and Linear Clique-Width of Trees. *Theoretical Computer Science*, 589: 87-98 (2015). Available at http://www.isima.fr/ kante/articles/kanteA13.pdf.

(77) P.A. Golovach, P. Heggernes, M.M. Kanté, D. Kratsch, Y. Villanger. Enumerating Minimal Dominating Sets in Chordal Bipartite Graphs. To appear in *Discrete Applied Mathematics*, 2015. Available at http://www.isima.fr/ kante/articles/kanteGHKV15.pdf.

(97) M.M. Kanté, V. Limouzy, A. Mary and L. Nourine. On the Enumeration of Minimal Dominating Sets and Related Notions. *SIAM J. Discrete Math.*, 28(4): 1916-1929 (2014). Available at http://arxiv.org/abs/1407.2053.

(104) M.M. Kanté and M. Rao. The Rank-Width of Edge-Coloured Graphs. *Theory of Computing Systems*, 52(4):599-644 (2013). Available (with long introduction) at http://www.isima.fr/ kante/articles/kanteR11.pdf.

(91) M.M. Kanté. Well-Quasi-Ordering of Matrices under Schur Complement and Applications to Directed Graphs. *European Journal of Combinatorics*, 33(8):1820-1841 (2012). Available at http://arxiv.org/abs/1102.2134.

(75) P.A. Golovach, P. Heggernes, M.M. Kanté, D. Kratsch, S.H. Saether and Y. Villanger. Output-Polynomial Enumeration on Graphs of Bounded (Local) Linear MIM-Width. In *ISAAC*, 2015. Journal version available at http://arxiv.org/abs/1509.03753.

(100) M.M. Kanté, A. Mary, V. Limouzy, L. Nourine and T. Uno. Polynomial Delay Algorithm for Listing Minimal Edge Dominating Sets in Graphs. In *WADS*, 2015. Available at http://arxiv.org/abs/1404.3501.

(99) M.M. Kanté, A. Mary, V. Limouzy, L. Nourine and T. Uno. A Polynomial Delay Algorithm for Enumerating Minimal Dominating Sets in Chordal Graphs. In *WG*, 2015. Available at http://arxiv.org/abs/1407.2036.

(2) I. Adler, M.M. Kanté and O. Kwon. Linear Rank-Width of Distance-Hereditary Graphs. In *WG'2014*, volume 8747 of *LNCS*, pages 42-55. Springer, 2014. Journal versions available at http://arxiv.org/abs/1403.1081v3 and http://arxiv.org/abs/1508.04718.

(103) M.M. Kanté and M. Rao. Directed Rank-Width and Displit Decomposition. In *WG'2009*, volume 5911 of *LNCS*, pages 214-225. Springer, 2010. Available at http://www.isima.fr/ kante/articles/kanteR09.pdf.

Bibliography

[1] Isolde Adler and Mamadou Moustapha Kanté. Linear rank-width and linear clique-width of trees. *Theoret. Comput. Sci.*, 589:87–98, 2015. 4, 21, 24, 25

[2] Isolde Adler, Mamadou Moustapha Kanté, and O-joung Kwon. Linear rank-width of distance-hereditary graphs. In *Graph-theoretic concepts in computer science*, volume 8747 of *Lecture Notes in Comput. Sci.*, pages 42–55. Springer, Cham, 2014. Journal versions are submitted, see [3, 4]. 4

[3] Isolde Adler, Mamadou Moustapha Kanté, and O-joung Kwon. Linear rank-width of distance-hereditary graphs I. a polynomial-time algorithm. *Submitted*, 2015. Available at arXiv:1403.1081v3. 4, 21, 22, 23, 25, 51

[4] Isolde Adler, Mamadou Moustapha Kanté, and O-joung Kwon. Linear rank-width of distance-hereditary graphs II. vertex-minor obstructions. *Submitted*, 2015. Available at arXiv:1508.04718v1. 4, 21, 24, 51

[5] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3):21–46, 1996. First International Colloquium on Graphs and Optimization (GOI), 1992 (Grimentz). 8, 40

[6] Rémy Belmonte and Martin Vatshelle. Graph classes with structured neighborhoods and algorithmic applications. *Theoret. Comput. Sci.*, 511:54–65, 2013. 7, 34

[7] Claude Berge. *Hypergraphs*, volume 45 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, 1989. Combinatorics of finite sets, Translated from the French. 7, 8, 42

[8] Achim Blumensath and Bruno Courcelle. On the monadic second-order transduction hierarchy. *Log. Methods Comput. Sci.*, 6(2):2:2, 28, 2010. 4

[9] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. 3

[10] Hans L. Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms*, 21(2):358–402, 1996. 46, 47

[11] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Leonid Khachiyan. Algorithms for enumerating circuits in matroids. In *Algorithms and computation*, volume 2906 of *Lecture Notes in Comput. Sci.*, pages 485–494. Springer, Berlin, 2003. 5

[12] Endre Boros, Vladimir Gurvich, and Peter L. Hammer. Dual subimplicants of positive Boolean functions. *Optim. Methods Softw.*, 10(2):147–156, 1998. Dedicated to Professor Masao Iri on the occasion of his 65th birthday. 11, 28

[13] A. Bouchet. Representability of $\triangle$-matroids. In *Combinatorics (Eger, 1987)*, volume 52 of *Colloq. Math. Soc. János Bolyai*, pages 167–182. North-Holland, Amsterdam, 1988. 1, 3, 16, 22

[14] A. Bouchet and A. Duchamp. Representability of $\triangle$-matroids over GF(2). *Linear Algebra Appl.*, 146:67–78, 1991. 20

[15] André Bouchet. Greedy algorithm and symmetric matroids. *Math. Programming*, 38(2):147–159, 1987. 16

[16] André Bouchet. Isotropic systems. *European J. Combin.*, 8(3):231–244, 1987. 16

[17] André Bouchet. Transforming trees by successive local complementations. *J. Graph Theory*, 12(2):195–207, 1988. 24

[18] André Bouchet. Circle graph obstructions. *J. Combin. Theory Ser. B*, 60(1):107–144, 1994. 1

[19] André Bouchet. Multimatroids. I. Coverings by independent sets. *SIAM J. Discrete Math.*, 10(4):626–646, 1997. 20

[20] André Bouchet. Multimatroids. II. Orthogonality, minors and connectivity. *Electron. J. Combin.*, 5:Research Paper 8, 25 pp. (electronic), 1998. 20

[21] André Bouchet. Multimatroids. IV. Chain-group representations. *Linear Algebra Appl.*, 277(1-3):271–289, 1998. 20

[22] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. 11

[23] Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theoret. Comput. Sci.*, 511:66–76, 2013. 7, 33, 34, 35

[24] M. Chein, M. Habib, and M. C. Maurer. Partitive hypergraphs. *Discrete Math.*, 37(1):35–50, 1981. 18

[25] Maria Chudnovsky, Neil Robertson, Paul D. Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164(1):51–229, 2006. 1

[26] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007. 1

[27] Derek G. Corneil, Michel Habib, Jean-Marc Lanlignel, Bruce Reed, and Udi Rotics. Polynomial-time recognition of clique-width ≤ 3 graphs. *Discrete Appl. Math.*, 160(6):834–865, 2012. 2

[28] Derek G. Corneil and Udi Rotics. On the relationship between clique-width and treewidth. *SIAM J. Comput.*, 34(4):825–847 (electronic), 2005. 2

[29] Alexis Cornet and Mamadou Moustapha Kanté. Some properties about mim-width. Technical report, Université Blaise Pascal, 2015. Results from the Master Thesis of Alexis Cornet. 7, 34

[30] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. 2, 3, 33, 47

[31] B. Courcelle and G. Sénizergues. The obstructions of a minor-closed set of graphs defined by a context-free grammar. *Discrete Math.*, 182(1-3):29–51, 1998. Graph theory (Lake Bled, 1995). 4

[32] Bruno Courcelle. The monadic second-order logic of graphs IV: definability properties of equational graphs. *Annals of Pure and Applied Logic*, 49(3):193–255, 1990. 2, 3

[33] Bruno Courcelle. Graph transformations expressed in logic and applications to structural graph theory. Report of Banff workshop in Graph Minors (08w5079), 2008. http://www.birs.ca/events/2008/5-day-workshops/08w5079/report08w5079.pdf. 4

[34] Bruno Courcelle. Linear delay enumeration and monadic second-order logic. *Discrete Appl. Math.*, 157(12):2675–2700, 2009. 33, 37

[35] Bruno Courcelle, Rodney G. Downey, and Michael R. Fellows. A note on the computability of graph minor obstruction sets for monadic second order ideals. *J.UCS*, 3(11):1194–1198 (electronic), 1997. 4

[36] Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic*, volume 138 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2012. A language-theoretic approach, With a foreword by Maurice Nivat. 2

[37] Bruno Courcelle, Cyril Gavoille, and Mamadou Moustapha Kanté. Compact labelings for efficient first-order model-checking. *J. Comb. Optim.*, 21(1):19–46, 2011. 47, 48

[38] Bruno Courcelle, Pinar Heggernes, Daniel Meister, Charis Papadopoulos, and Udi Rotics. A characterisation of clique-width through nested partitions. *Discrete Applied Mathematics*, 187:70–81, 2015. 2

[39] Bruno Courcelle and Mamadou Moustapha Kanté. Graph operations characterizing rank-width. *Discrete Applied Mathematics*, 157(4):627–640, 2009. 3, 15, 19

[40] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000. 2

[41] Bruno Courcelle and Sang-il Oum. Vertex-minors, monadic second-order logic, and a conjecture by Seese. *J. Combin. Theory Ser. B*, 97(1):91–126, 2007. 3, 47

[42] Jean-François Couturier, Romain Letourneur, and Mathieu Liedloff. On the number of minimal dominating sets on some graph classes. *Theoret. Comput. Sci.*, 562:634–642, 2015. 48

[43] Nadia Creignou, Arne Meier, Julian-Steffen Müller, Johannes Schmidt, and Heribert Vollmer. Paradigms for parameterized enumeration. In *Mathematical foundations of computer science 2013*, volume 8087 of *Lecture Notes in Comput. Sci.*, pages 290–301. Springer, Heidelberg, 2013. 47

[44] William H. Cunningham. *A combinatorial decomposition theory*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1973. 1

[45] William H. Cunningham. Decomposition of directed graphs. *SIAM J. Algebraic Discrete Methods*, 3(2):214–228, 1982. 18

[46] William H. Cunningham and Jack Edmonds. A combinatorial decomposition theory. *Canad. J. Math.*, 32(3):734–765, 1980. 18, 19

[47] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer International Publishing, 2015. 5

[48] Konrad Kazimierz Dabrowski and Daniël Paulusma. Classifying the clique-width of H-free bipartite graphs. In *Computing and combinatorics*, volume 8591 of *Lecture Notes in Comput. Sci.*, pages 489–500. Springer, Cham, 2014. 2

[49] J. David and L. Nourine. A parameterizable enumeration algorithm for sequence mining. *Theoret. Comput. Sci.*, 468:59–68, 2013. 5

[50] Hubert de Fraysseix. Local complementation and interlacement graphs. *Discrete Math.*, 33(1):29–35, 1981. 4

[51] Nick D. Dendris, Lefteris M. Kirousis, and Dimitrios M. Thilikos. Fugitive-search games on graphs and related parameters. *Theoret. Comput. Sci.*, 172(1-2):233–254, 1997. 21

[52] Jack S. Dharmatilake. A min-max theorem using matroid separations. In *Matroid theory (Seattle, WA, 1995)*, volume 197 of *Contemp. Math.*, pages 333–342. Amer. Math. Soc., Providence, RI, 1996. 23

[53] Reinhard Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, third edition, 2005. 1, 9

[54] Rodney G. Downey and Michael R. Fellows. *Fundamentals of parameterized complexity*. Texts in Computer Science. Springer, London, 2013. 6

[55] A. Ehrenfeucht, T. Harju, and G. Rozenberg. *The theory of 2-structures.* World Scientific Publishing Co., Inc., River Edge, NJ, 1999. A framework for decomposition and transformation of graphs. 9

[56] Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Comput.*, 32(2):514–537 (electronic), 2003. 6, 7, 45

[57] Thomas Eiter, Kazuhisa Makino, and Georg Gottlob. Computational aspects of monotone dualization: a brief survey. *Discrete Appl. Math.*, 156(11):2035–2049, 2008. 5

[58] J. A. Ellis, I. H. Sudborough, and J. S. Turner. The vertex separation and search number of a graph. *Inform. and Comput.*, 113(1):50–79, 1994. 4, 21, 23, 24

[59] Martin Farber and Robert E. Jamison. Convexity in graphs and hypergraphs. *SIAM J. Algebraic Discrete Methods*, 7(3):433–444, 1986. 49

[60] Michael R. Fellows and Michael A. Langston. On search, decision, and the efficiency of polynomial-time algorithms. *J. Comput. System Sci.*, 49(3):769–779, 1994. 4

[61] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width is NP-complete. *SIAM J. Discrete Math.*, 23(2):909–939, 2009. 2, 21, 25

[62] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics.* Cambridge University Press, Cambridge, 2009. 5, 7, 48

[63] Fedor V. Fomin, Fabrizio Grandoni, Artem V. Pyatkin, and Alexey A. Stepanov. Combinatorial bounds via measure and conquer: bounding minimal dominating sets and applications. *ACM Trans. Algorithms*, 5(1):Art. 9, 17, 2009. 48

[64] Fedor V. Fomin and Dieter Kratsch. *Exact exponential algorithms.* Number 1 in Texts in theoretical computer science. An EATCS series. Springer-Verlag, 2010. 5, 6, 48

[65] Michael L. Fredman and Leonid Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *J. Algorithms*, 21(3):618–628, 1996. 5

[66] Markus Frick. Generalized model-checking over locally tree-decomposable classes. *Theory Comput. Syst.*, 37(1):157–191, 2004. Symposium on Theoretical Aspects of Computer Science (Antibes-Juan les Pins, 2002). 48

[67] Alan Frieze, Mark Jerrum, Michael Molloy, Robert Robinson, and Nicholas Wormald. Generating and counting Hamilton cycles in random regular graphs. *J. Algorithms*, 21(1):176–198, 1996. 48

[68] Tibor Gallai. Transitiv orientierbare Graphen. *Acta Mathematica Academiae Scientiarum Hungaricae*, 18:25–66, 1967. 1

[69] Robert Ganian, Petr Hliněný, Joachim Kneis, Alexander Langer, Jan Obdržálek, and Peter Rossmanith. Digraph width measures in parameterized algorithmics. *Discrete Appl. Math.*, 168:88–107, 2014. 20

[70] Robert Ganian, Petr Hliněný, and Jan Obdržálek. A unified approach to polynomial algorithms on graphs of bounded (bi-)rank-width. *European J. Combin.*, 34(3):680–701, 2013. 20

[71] J. F. Geelen, A. M. H. Gerards, N. Robertson, and G. P. Whittle. On the excluded minors for the matroids of branch-width k. *J. Combin. Theory Ser. B*, 88(2):261–265, 2003. 15

[72] Jim Geelen, Bert Gerards, and Geoff Whittle. Excluding a planar graph from $GF(q)$-representable matroids. *J. Combin. Theory Ser. B*, 97(6):971–998, 2007. 4

[73] Jim Geelen, Bert Gerards, and Geoff Whittle. Solving Rota's conjecture. *Notices Amer. Math. Soc.*, 61(7):736–743, 2014. 3, 20

[74] Alain Gély, Lhouari Nourine, and Bachir Sadi. Enumeration aspects of maximal cliques and bicliques. *Discrete Appl. Math.*, 157(7):1447–1459, 2009. 5, 8, 38

[75] Petr A. Golovach, Pinar Heggernes, Mamadou Moustapha Kanté, Dieter Kratsch, Sigve H. Sæther, and Yngve Villanger. Output-polynomial enumeration on graphs of bounded (local) linear mim-width. In *Algorithms and computation*, volume ???? of *Lecture Notes in Comput. Sci.*, pages ??–?? Springer, Heidelberg, 2015. A journal version is submitted, available at. 7, 8, 33, 34, 35, 36, 38, 40

[76] Petr A. Golovach, Pinar Heggernes, Mamadou Moustapha Kanté, Dieter Kratsch, and Yngve Villanger. Minimal dominating sets in interval graphs and trees. Submitted. 48

[77] Petr A. Golovach, Pinar Heggernes, Mamadou Moustapha Kanté, Dieter Kratsch, and Yngve Villanger. Enumerating minimal dominating sets in chordal bipartite graphs. *Discrete Appl. Math.*, ????(?):–, 2015. 8, 32, 38, 39

[78] Petr A. Golovach, Pinar Heggernes, Dieter Kratsch, and Yngve Villanger. An Incremental Polynomial Time Algorithm to Enumerate All Minimal Edge Dominating Sets. *Algorithmica*, 72(3):836–859, 2015. 8, 38, 39, 40, 42

[79] Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *Internat. J. Found. Comput. Sci.*, 11(3):423–443, 2000. Selected papers from the Workshop on Theoretical Aspects of Computer Science (WG 99), Part 1 (Ascona). 2

[80] Georg Gottlob and Enrico Malizia. Achieving new upper bounds for the hypergraph duality problem through logic. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 43:1–43:10. ACM, 2014. 6

[81] Frank Gray. Pulse code communication. United States Patent Number 2632058, March 17 1953. 5

[82] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 89–98. ACM, 2014. 48

[83] Teresa W. Haynes, Stephen T. Hedetniemi, and Peter J. Slater. *Fundamentals of domination in graphs*, volume 208 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, 1998. 6

[84] Ryan Hayward, Chính Hoàng, and Frédéric Maffray. Optimizing weakly triangulated graphs. *Graphs Combin.*, 5(4):339–349, 1989. 32

[85] Ryan B. Hayward, Chính Hoàng, and Frédéric Maffray. Erratum: Optimizing weakly triangulated graphs. *Graphs and Combinatorics*, 6(1):33–35, 1990. 32

[86] Petr Hliněný and Sang-il Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*, 38(3):1012–1032, 2008. 3, 14, 15, 19

[87] Petr Hlinený, Sang-il Oum, Detlef Seese, and Georg Gottlob. Width parameters beyond tree-width and their applications. *Comput. J.*, 51(3):326–362, 2008. 14

[88] Jisu Jeong, O-joung Kwon, and Sang-il Oum. Excluded vertex-minors for graphs of linear rank-width at most k. *European J. Combin.*, 41:242–257, 2014. 24

[89] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Inform. Process. Lett.*, 27(3):119–123, 1988. 8, 38

[90] Marcin Kamiński, Vadim V. Lozin, and Martin Milanič. Recent developments on graphs of bounded clique-width. *Discrete Appl. Math.*, 157(12):2747–2761, 2009. 2

[91] Mamadou Moustapha Kanté. Well-quasi-ordering of matrices under Schur complement and applications to directed graphs. *European J. Combin.*, 33(8):1820–1841, 2012. 3, 13, 15, 16, 17, 18, 19, 20

[92] Mamadou Moustapha Kanté, Eun Jung Kim, O-Joung Kwon, and Christophe Paul. An fpt algorithm and a polynomial kernel for linear rankwidth one vertex deletion. In *IPEC'15*, volume ???? of *LIPICS*, pages ??–?? Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. available at arXiv:1504.05905v1. 47

[93] Mamadou Moustapha Kanté and O.-joung Kwon. An upper bound on the size of obstructions for bounded linear rank-width. *CoRR*, abs/1412.6201, 2014. 20, 24, 46

[94] Mamadou Moustapha Kanté, Christian Laforest, and Benjamin Momège. An exact algorithm to check the existence of (elementary) paths and a generalisation of the cut problem in graphs with forbidden transitions. In *SOFSEM 2013: theory and practice of computer science*, volume 7741 of *Lecture Notes in Comput. Sci.*, pages 257–267. Springer, Heidelberg, 2013. 8

[95] Mamadou Moustapha Kanté, Christian Laforest, and Benjamin Momège. Trees in graphs with conflict edges or forbidden transitions. In *Theory and applications of models of computation*, volume 7876 of *Lecture Notes in Comput. Sci.*, pages 343–354. Springer, Heidelberg, 2013. 8

[96] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. On the neighbourhood Helly of some graph classes and applications to the enumeration of minimal dominating sets. In *Algorithms and computation*, volume 7676 of *Lecture Notes in Comput. Sci.*, pages 289–298. Springer, Heidelberg, 2012. 42

[97] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. On the enumeration of minimal dominating sets and related notions. *SIAM J. Discrete Math.*, 28(4):1916–1929, 2014. 6, 7, 11, 27, 28, 29, 30, 31, 33, 36, 37

[98] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, Lhouari Nourine, and Takeaki Uno. On the enumeration and counting of minimal dominating sets in interval and permutation graphs. In *Algorithms and computation*, volume 8283 of *Lecture Notes in Comput. Sci.*, pages 339–349. Springer, Heidelberg, 2013. 8, 33, 34, 36, 48

[99] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, Lhouari Nourine, and Takeaki Uno. A polynomial delay algorithm for enumerating minimal dominating sets in chordal graphs. In *Graph-theoretic concepts in computer science*, volume ???? of *Lecture Notes in Comput. Sci.*, pages ??–?? Springer, Cham, 2015. 6, 30, 31, 32

[100] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, Lhouari Nourine, and Takeaki Uno. Polynomial delay algorithm for listing minimal edge dominating sets in graphs. In *Algorithms and Data Structures Symposium*, volume 9214 of *Lecture Notes in Comput. Sci.*, pages 446–457. Springer, Cham, 2015. 6, 7, 42, 43, 44, 45

[101] Mamadou Moustapha Kanté, Fatima-Zahra Moataz, Benjamin Momège, and Nicolas Nisse. Finding paths in grids with forbidden transitions. In *Graph-theoretic concepts in computer science*, volume ???? of *Lecture Notes in Comput. Sci.*, pages ??–?? Springer, Cham, 2015. 8

[102] Mamadou Moustapha Kanté and Lhouari Nourine. Polynomial time algorithms for computing a minimum hull set in distance-hereditary and chordal graphs. In *SOFSEM 2013: theory and practice of computer science*, volume 7741 of *Lecture Notes in Comput. Sci.*, pages 268–279. Springer, Heidelberg, 2013. A journal version is submitted, available at http://www.isima.fr/ kante/articles/kanteN15.pdf. 49

[103] Mamadou Moustapha Kanté and Michaël Rao. Directed rank-width and displit decomposition. In *Graph-theoretic concepts in computer science*, volume 5911 of *Lecture Notes in Comput. Sci.*, pages 214–225. Springer, Berlin, 2010. 3, 13, 18, 19, 20, 24

[104] Mamadou Moustapha Kanté and Michael Rao. The rank-width of edge-coloured graphs. *Theory Comput. Syst.*, 52(4):599–644, 2013. 3, 13, 14, 15, 16, 19, 20, 24

[105] Mamadou Moustapha Kanté, Rudini M. Sampaio, Vinícius F. dos Santos, and Jayme L. Szwarcfiter. On the geodetic rank of a graph. Technical report, Univesité Blaise Pascal, 2015. 49

[106] Ming-Yang Kao, editor. *Encyclopedia of Algorithms*. Springer, 2015. 5

[107] Dimitris J. Kavvadias and Elias C. Stavropoulos. An efficient algorithm for the transversal hypergraph generation. *J. Graph Algorithms Appl.*, 9(2):239–264 (electronic), 2005. 6, 7, 40, 42

[108] Wojciech Kazana and Luc Segoufin. Enumeration of first-order queries on classes of structures with bounded expansion. In Richard Hull and Wenfei Fan, editors, *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 297–308. ACM, 2013. 48

[109] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *Automata, languages, and programming. Part I*, volume 7965 of *Lecture Notes in Comput. Sci.*, pages 613–624. Springer, Heidelberg, 2013. 47

[110] Lefteris M. Kirousis and Christos H. Papadimitriou. Searching and pebbling. *Theoret. Comput. Sci.*, 47(2):205–218, 1986. 21

[111] T. Kloks, H. Bodlaender, H. Müller, and D. Kratsch. Computing treewidth and minimum fill-in: all you need are the minimal separators. In *Algorithms—ESA '93 (Bad Honnef, 1993)*, volume 726 of *Lecture Notes in Comput. Sci.*, pages 260–271. Springer, Berlin, 1993. 4

[112] T. Kloks and D. Kratsch. Treewidth of chordal bipartite graphs. *J. Algorithms*, 19(2):266–281, 1995. 32

[113] Jens Lagergren. Upper bounds on the size of obstructions and intertwines. *J. Comb. Theory, Ser. B*, 73(1):7–40, 1998. 4

[114] E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM J. Comput.*, 9(3):558–565, 1980. 5, 6, 7, 8, 45

[115] Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 1997. With a foreword by P. M. Cohn. 9

[116] Seymour Lipschutz. *Schaum's Outline of Theory and Problems of Linear Algebra*. Mc-Graw Hill, 2$^{\text{nd}}$ edition, 1991. 10

[117] Vadim V. Lozin. Minimal classes of graphs of unbounded clique-width. *Ann. Comb.*, 15(4):707–722, 2011. 2

[118] Kazuhisa Makino and Takeaki Uno. New algorithms for enumerating all maximal cliques. In *Algorithm theory—SWAT 2004*, volume 3111 of *Lecture Notes in Comput. Sci.*, pages 260–272. Springer, Berlin, 2004. 38, 40

[119] J. A. Makowsky. Algorithmic uses of the Feferman-Vaught theorem. *Ann. Pure Appl. Logic*, 126(1-3):159–213, 2004. Provinces of logic determined. 2

[120] J. A. Makowsky and U. Rotics. On the clique-width of graphs with few P_4's. *Internat. J. Found. Comput. Sci.*, 10(3):329–348, 1999. 2

[121] Janos Makowsky. The graph polynomials project in haifa, 2012. Webpage:http://www.cs.technion.ac.il/ janos/RESEARCH/gp-homepage.html. 5

[122] Arnaud Mary. *Énumération des dominants minimaux d'un graphe*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, Canada, 2013. 6, 11, 28, 47

[123] Frédéric Mazoit and Stéphan Thomassé. Branchwidth of graphic matroids. In *Surveys in combinatorics 2007*, volume 346 of *London Math. Soc. Lecture Note Ser.*, pages 275–286. Cambridge Univ. Press, Cambridge, 2007. 14

[124] R. H. Möhring and F. J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. In *Algebraic and combinatorial methods in operations research*, volume 95 of *North-Holland Math. Stud.*, pages 257–355. North-Holland, Amsterdam, 1984. 18

[125] Keisuke Murakami and Takeaki Uno. Efficient algorithms for dualizing large-scale hypergraphs. *Discrete Appl. Math.*, 170:83–94, 2014. 6

[126] Keisuke Murakami and Takeaki Uno. Efficient algorithms for dualizing large-scale hypergraphs. *Discrete Appl. Math.*, 170:83–94, 2014. 7, 8, 40, 42, 43

[127] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity*, volume 28 of *Algorithms and Combinatorics*. Springer, Heidelberg, 2012. Graphs, structures, and algorithms. 48

[128] Lhouari Nourine and Jean-Marc Petit. Extending set-based dualization: Application to pattern mining. In Luc De Raedt, Christian Bessière, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, editors, *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 630–635. IOS Press, 2012. 5

[129] Sang-Il Oum. *Graphs of Bounded Rank Width*. PhD thesis, Princeton University, 2005. 18, 19

[130] Sang-il Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79–100, 2005. 1, 3, 4, 14, 15, 18, 19, 20, 23

[131] Sang-il Oum. Rank-width and well-quasi-ordering. *SIAM J. Discrete Math.*, 22(2):666–682, 2008. 3, 14, 16, 19, 20

[132] Sang-il Oum. Rank-width is less than or equal to branch-width. *J. Graph Theory*, 57(3):239–244, 2008. 21

[133] Sang-il Oum. Excluding a bipartite circle graph from line graphs. *J. Graph Theory*, 60(3):183–203, 2009. 4

[134] Sang-il Oum. Rank-width and well-quasi-ordering of skew-symmetric or symmetric matrices. *Linear Algebra Appl.*, 436(7):2008–2036, 2012. 3, 14, 16, 17, 18, 19, 20

[135] Sang-il Oum and Paul Seymour. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B*, 96(4):514–528, 2006. 2, 3, 14

[136] Christophe Paul. Aspects Algorithmiques de la Décomposition Modulaire. Habilitation á Diriger des Recherches, Université Montpellier II, 2006. 1

[137] Ignacio M. Pelayo. On convexity in graphs. Technical report, UPC, 2004. 49

[138] Michaël Rao. Solving some NP-complete problems using split decomposition. *Discrete Appl. Math.*, 156(14):2768–2780, 2008. 1

[139] Imran Rauf. *Polynomially Solvable Cases of Hypergraph Transversal and Related Problems*. PhD thesis, Saarland University, Saarland, Germany, 2011. 8, 34

[140] Neil Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995. 20

[141] Neil Robertson and Paul D. Seymour. Graph minors. I. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. 1, 4, 21

[142] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. 2, 4

[143] Neil Robertson and Paul D. Seymour. Graph minors. X. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991. 14

[144] Neil Robertson and Paul D. Seymour. Graph minors. XX. wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. 1

[145] Frank Ruskey. Combinatorial generation. Working version available at http://www.1stworks.com/ref/RuskeyCombGen.pdf. 5, 7, 8, 48

[146] Benno Schwikowski and Ewald Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Appl. Math.*, 117(1-3):253–265, 2002. 5

[147] Hong Shen and Weifa Liang. Efficient enumeration of all minimal separators in a graph. *Theoret. Comput. Sci.*, 180(1-2):169–180, 1997. 5, 29

[148] Akiyoshi Shioura, Akihisa Tamura, and Takeaki Uno. An optimal algorithm for scanning all spanning trees of undirected graphs. *SIAM J. Comput.*, 26(3):678–692, 1997. 40

[149] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Inform. and Comput.*, 82(1):93–133, 1989. 48

[150] Yann Strozecki. *Enumeration complexity and matroid decomposition*. PhD thesis, Université Paris Diderot - Paris 7, Paris, France, 2010. 5

[151] Ken Takata. A worst-case analysis of the sequential method to list the minimal hitting sets of a hypergraph. *SIAM J. Discrete Math.*, 21(4):936–946, 2007. 7, 42, 43

[152] Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.*, 10(4):529–550, 1997. 34

[153] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Math. Systems Theory*, 2:57–81, 1968. 1, 2

[154] Stéphan Thomassé and Anders Yeo. Total domination of graphs and small transversals of hypergraphs. *Combinatorica*, 27(4):473–487, 2007. 11

[155] Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM J. Comput.*, 6(3):505–517, 1977. 5, 6

[156] A. W. Tucker. A combinatorial equivalence of matrices. In *Proc. Sympos. Appl. Math., Vol. 10*, pages 129–140. American Mathematical Society, Providence, R.I., 1960. 14

[157] W. T. Tutte. *Introduction to the theory of matroids*. Modern Analytic and Computational Methods in Science and Mathematics, No. 37. American Elsevier Publishing Co., Inc., New York, 1971. 16, 20

[158] Takeaki Uno. Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs. In *Algorithms and computation (Singapore, 1997)*, volume 1350 of *Lecture Notes in Comput. Sci.*, pages 92–101. Springer, Berlin, 1997. 5

[159] Martin Vatshelle. *New width parameters of graphs*. PhD thesis, University of Bergen, Bergen, Norway, 2012. 33