Order N° : 3693

DISSERTATION

PRESENTED TO THE FACULTY OF

UNIVERSITÉ BORDEAUX I

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

BY **Mamadou Moustapha KANTÉ**

IN CANDIDACY FOR THE DEGREE OF

DOCTEUR OF PHILOSOPHY

IN : COMPUTER SCIENCE

———

Graph Structurings
Some Algorithmic Applications

———

Defended on 3$^{\text{rd}}$ december 2008

Recommended for acceptance by :

Christophe Paul	Chargé de Recherche, HDR
Luc Segoufin	Directeur de Recherche, HDR

Examiners are :

Didier Caucal	Chargé de Recherche, HDR	Examinateur
Bruno Courcelle	Professeur	Supervisor
Cyril Gavoille	Professeur	Examinateur
Christophe Paul	Chargé de Recherche, HDR	Rapporteur
Luc Segoufin	Directeur de Recherche, HDR	Rapporteur

– 2008 –

Abstract. Every property definable in *monadic second order logic* can be checked in polynomial-time on graph classes of bounded *clique-width*. Clique-width is a graph parameter defined in an algebraical way, i.e., with operations "concatenating graphs" and that generalize concatenation of words. *Rank-width*, defined in a combinatorial way, is equivalent to the clique-width of undirected graphs. We give an algebraic characterization of rank-width and we show that rank-width is linearly bounded in term of tree-width. We also propose a notion of rank-width of directed graphs and a *vertex-minor* inclusion for directed graphs. We show that directed graphs of bounded rank-width are characterized by a finite list of finite directed graphs to exclude as vertex-minor.

Many graph classes do not have bounded rank-width, e.g., planar graphs. We are interested in labeling schemes on these graph classes. A labeling scheme for a property P in a graph G consists in assigning a label, as short as possible, to each vertex of G and such that, we can verify if G satisfies P by just looking at the labels. We show that every property definable in *first order logic* admit labeling schemes with labels of logarithmic size on graph classes that have bounded *local clique-width*. Bounded degree graph classes, minor closed classes of graphs that exclude an apex graph as a minor have bounded local clique-width.

If x and y are two vertices and X is a subset of the set of vertices and Y is a subset of the set of edges, we let $Conn(x, y, X, F)$ be the graph property: x and y are connected by a path that avoids the vertices in X and the edges in F. This property is not definable by a first order formula. We show that it admits a labeling scheme with labels of logarithmic size on planar graphs. We also show that $Conn(x, y, X, \emptyset)$ admits short labeling schemes with labels of logarithmic size on graph classes that are "planar gluings" of graphs of small clique-width and with limited overlaps.

Contents

Graph Structurings: Some Algorithmic Applications

We can use graphs to model many *real-life* networks, e.g., social networks, railway networks, Internet network. And studying the properties of these networks, e.g., the connectivity or the reachability, is equivalent to studying the structural properties of the underlying graphs that represent them. Algorithms follow from such study. It is thus natural to focus our attention on graphs and their many algorithmic as well as structural questions. We can cite the *Seven Königsberg Bridges* modeled and solved as a graph problem by Euler [Eul59].

A common tool to solve algorithmic problems on graphs is the notion of graph decomposition. There are several types of graph decomposition, each of them being suitable for some problems and more than others. For instance, the decomposition of 2-connected graphs into 3-connected components [Tut66] is not *a priori* suitable for solving NP-Hard problems but can be used as a tool when dealing with 2-connected graphs, particularly for reducing problems on 2-connected graphs to problems on 3-connected graphs. However, as mentioned in [Rao06], graph decompositions can be classified into two families:

1. Canonical graph decompositions that are defined by means of certain substitutions or compositions of patterns. We can cite the *modular decomposition* [Gal67] or the *split decomposition* [Cun82] as examples of such decompositions.

2. Graph decompositions that are defined by "reducing" graphs to other graphs that we call here *target graphs* and that are mostly trees. To these decompositions are associated graph parameters that measure how difficult it is to reconstruct graphs from the target graphs. We can cite the *tree-decomposition* with its associated graph parameter called *tree-width* [RS83] and the *rank-decomposition* with its associated graph parameter called *rank-width* [Oum05b] as examples of such decompositions.

The goal with graph decompositions of the second type is to transfer the algorithmic as well as the structural properties of the target graphs to the decomposed graphs. In this case the smaller is the associated parameter the better it is, because the associated parameter measures how close are the decomposed graphs from the target graphs.

Many NP-Hard problems admit linear time algorithms on trees [CDG$^+$07]. If the target graphs of the graph decompositions are trees, we can hope to transfer the algorithmic results from trees to the graphs that admit small value of the associated graph parameter. Moreover, many graph classes that have an underlying tree-structure admit linear time algorithms for

several NP-Hard problems. This is the case for instance of chordal graphs [Dir61], series-parallel graphs [TNS82] and k-partial trees [Arn85]. However, if a *reduction to trees* allows to transfer the algorithmic results of trees, not all graphs can be *reduced to trees*, otherwise we would have P=NP. For instance, the algorithmic results for trees can be transferred to graphs that have tree-width bounded by some fixed value [Cou90, ALS91] and not all graphs have tree-width bounded by a fixed value (planar graphs and chordal graphs have unbounded tree-width). It is thus important to identify classes of graphs that can be *reduced to trees* with respect to a graph decomposition. A common way to do that is to characterize graph classes by excluded configurations by means of relations on graphs such as the *minor* relation [RS83] or the *vertex-minor* relation [Oum05b]. These characterizations can lead to recognition algorithms that are not always the most efficient ones and do not always help to build an optimal decomposition. An example is the quadratic time algorithm by Robertson and Seymour for checking if a graph has tree-width at most k [RS95]. A better one by Bodlaender [Bod96] gives an optimal tree-decomposition.

Another approach for solving in polynomial time algorithmic questions is the use of algebraic tools, which consists in the definition of operations on graphs that generalize the concatenation of words. They allow to construct graphs from basic graphs in a *simple way* and there are several advantages to have algebraic definitions of graph decompositions:

- Analogously to context-free languages of words, we can define context-free sets of graphs by systems of equations. We can therefore use tools of universal algebra and derive algorithmic results for the study of certain classes of graphs [BC06, Cou90, CER93].

- We can give uniform constructions for solving in linear time large classes of problems expressible in *monadic second order logic*. Such constructions are presented in the articles [Cou90, Cou93, CER93, CMR00, Mak04] and in the survey by Grohe [Gro07].

- For algorithmic purposes, we need sometimes graph decompositions of logarithmic height and of small width, that we call *balanced decompositions* (see for instance the articles [Bod88, BH98, CV03]). Algebraic tools allow to give uniform proofs for the existence of balanced decompositions (Chapter 6).

Let us give an example of a set of graph operations. A graph with two distinguished vertices, named respectively *s-vertex* and *t-vertex*, is called a 2-*terminal graph*. Given two 2-terminal graphs, G_1 and G_2, we denote by $G_1//G_2$ the 2-terminal graph obtained by gluing the s-vertices of G_1 and G_2 and the same for the t-vertices. And we denote by $G_1 \bullet G_2$ the 2-terminal graph obtained by gluing the t-vertex of G_1 with the s-vertex of G_2 and by considering the s-vertex of G_1 and the t-vertex of G_2 as the distinguished vertices of $G_1 \bullet G_2$. Figure 1 shows examples of these two operations.

Series-parallel graphs are generated by terms written with the operations $//$ and $\bullet$ and the basic graph $\mathbf{a}$ which represents a graph with a single edge, one end-vertex colored by s and the other by t [TNS82]. Thus, to any series-parallel graph corresponds at least one term written with symbols in $\{//, \bullet, \mathbf{a}\}$. It is worth noticing that several terms can generate the same graph. Figure 2 shows a series-parallel graph and a term that generates it.

Many polynomial time algorithms on series-parallel graphs are based on such terms that generate the input graphs. The linear time algorithms of [Cou90] are also based on algebraic
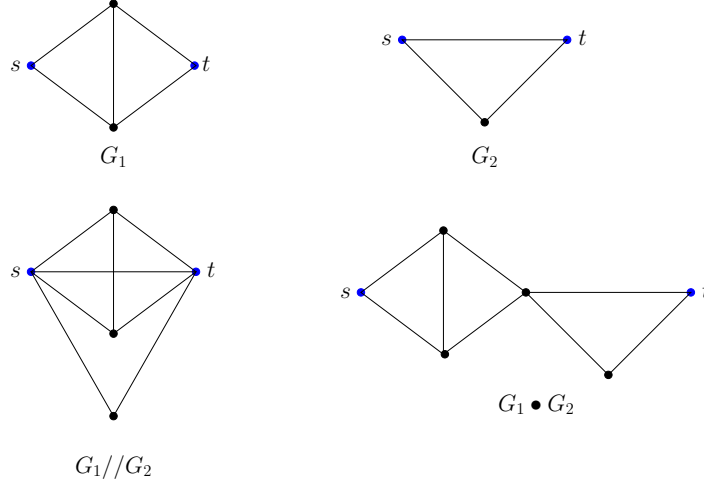
Figure 1: The two graphs in the top represent two 2-terminal graphs G_1 (left) and G_2 (right). The two graphs in the bottom are $G_1//G_2$ (left) and $G_1 \bullet G_2$ (right). The distinguished vertices are the blue ones, labeled respectively by s and t.

$$(((\mathbf{a}\bullet\mathbf{a})\bullet(((\mathbf{a}\bullet\mathbf{a})//(\mathbf{a}\bullet\mathbf{a}))//(\mathbf{a}\bullet\mathbf{a})))\bullet(\mathbf{a}\bullet\mathbf{a}))//(\mathbf{a}\bullet\mathbf{a})$$

(b)
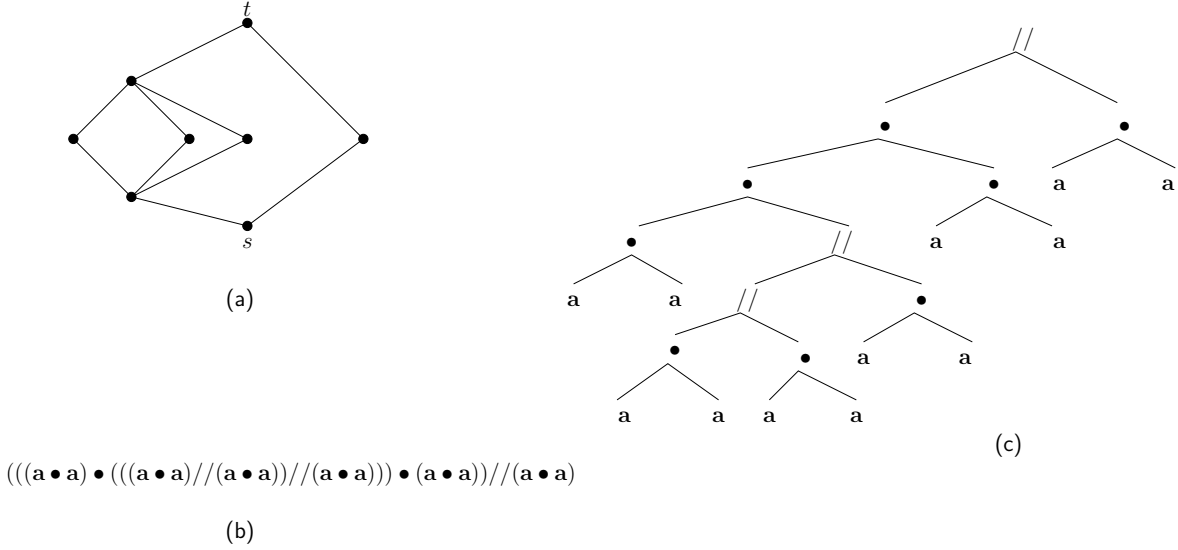
Figure 2: (a) represents a series-parallel graph, (b) a term that represents it and (c) the syntactic tree of the term.

definitions. It is then interesting to give algebraic definitions for graph decompositions that do not have one. In this perspective our work has the following motivations.

- To give algebraic definitions when they do not exist.

- To compare various notions of graph parameters associated with graph decompositions.

- To give a uniform proof for the existence of balanced decompositions.

- To give results that are *easily implementable*.

- To propose other types of graph decompositions and, particularly graph decompositions where the target graphs are not trees and, to study the algorithmic results that we can obtain.

Graph Classes of Bounded Clique-Width

Clique-width [CER93, CO00] is a graph parameter defined in an algebraic way. It measures how many colors we need to construct a graph by using as basic graphs, single colored vertices and, as graph operations: (1) the disjoint union, (2) unary operations that change colors (3) and unary operations that add edges between colored vertices. Clique-width is more general than tree-width in that every class of graphs of bounded tree-width has bounded clique-width [CO00, CR05] and the converse is false (cliques have clique-width 2 and unbounded tree-width). Clique-width is interesting because problems expressible in monadic second order logic can be solved in cubic time on graph classes of bounded clique-width by combining results from [CMR00] and [HO07, OS06]. However, deciding if the clique-width of a graph is at most k is NP-complete [FRRS06] and for fixed $k \geq 4$, there is no known polynomial time algorithm that decides whether a given graph has clique-width at most k (for $k \leq 3$, there exists a polynomial time algorithm [CHL$^+$00]).

In their investigations of the recognition of undirected graphs of clique-width at most k, Oum and Seymour introduced the notion of *rank-width* of undirected graphs [Oum05b, OS06]. Rank-width is equivalent to clique-width in the sense that the same classes of graphs have bounded clique-width and bounded rank-width. As for clique-width, deciding the rank-width of an undirected graph is NP-complete [HOSG08] but deciding whether the rank-width of an undirected graph is at most k can be done in cubic time [HO07]. Contrary to the case of clique-width, there is a notion of minor, called *vertex-minor*, related to rank-width. Moreover, graphs of rank-width at most k are characterized by a finite list of excluded vertex-minors [Oum05b].

Clique-width is defined for undirected as well as for directed graphs. However, rank-width is defined only for undirected graphs and, does not have any algebraic definition in Oum's works. Our work on rank-width has the following motivations.

- To better understand the notion of rank-width and perhaps to contribute to the proof of a *Vertex-Minor Theorem* conjectured by Oum [Oum08a, Oum09], saying that undirected graphs are well-quasi-ordered under vertex-minor ordering.

- To extend the notion of rank-width to directed graphs and, in the future to relational structures.

- To find an algebraic definition for these notions of rank-width.

We will propose a notion of rank-width for directed graphs and we will give an algebraic definition for rank-width. We will also relate the two notions of minor and of vertex-minor and as a consequence we will prove that rank-width is linearly bounded in term of tree-width.

Labeling Schemes

We can represent graphs by adjacency lists or by adjacency matrices. However, the two representations can present some drawbacks when the number of vertices is large. Typically, answering adjacency queries needs to search in an adjacency list while the use of adjacency matrices requires $O(n^2)$-memory space if n is the number of vertices even if the graph is sparse. Even if answering the adjacency takes constant-time in adjacency matrices, it cannot be used in many algorithms because of space constraints. Moreover, it is frequently useful to have a distributed representation of a network because we want each node to have a partial knowledge (only) of the network. We need then a more compact representation of graphs which allows to answer adjacency queries relatively quickly. One way consists in assigning to each vertex a label, as short as possible, such that we can answer the adjacency query about two vertices just by looking at their labels. This is commonly known under the name of *implicit representation* [Spi03]. We can find in the book [Spi03] and the PhD thesis [Lab07] classes of graphs where short implicit representations, say by labels of logarithmic size, are defined. For instance, if a graph with n vertices excludes a fixed minor, then it admits an implicit representation that uses labels of size at most $2\log(n) + O(\log(\log(n)))$ (measured in bits) [GL07].

In other algorithms, the adjacency is not the only frequently asked query. For instance, a common query in managing routing networks is the existence of a path between two nodes and if it does exist, the description of this path. We can generalize the notion of labeling from adjacency to any property P. Formally, given a property $P(x_1, \ldots, x_m, X_1, \ldots, X_q)$ that depends on vertices $x_1, \ldots, x_m$ and sets of vertices $X_1, \ldots, X_q$ and an injective function $f : \mathbb{N} \to \mathbb{N}$, an f-*labeling* for P in a graph G of a fixed class $\mathcal{C}$ consists in assigning to each vertex of G a label of size $O(f(|V_G|))$ and such that given $a_1, \ldots, a_m \in V_G$ and $Y_1, \ldots, Y_q \subseteq V_G$, we can test whether G satisfies $P(a_1, \ldots, a_m, Y_1, \ldots, Y_q)$ just by looking at the labels of $a_1, \ldots, a_m$ and the labels of the vertices in $Y_1, \ldots, Y_q$. For example, the k-*vertex connectivity* query admits a $(k^2 \cdot \log)$-labeling in the class of graphs [Kor07b]. In the articles [GKK$^+$01, GP03a] several labelings are presented for the distance query and the approximate distance query in many graph classes, e.g., planar graphs, graph classes of bounded tree-width, ...

The use of decompositions and of algebraic definitions of decompositions can also help to get uniform constructionsf of f-labelings for large classes of properties. For instance, Courcelle and Vanicat [CV03] proved that every property expressible in monadic second order logic admits an $(f(k) \cdot \log)$-labeling on classes of graphs of clique-width at most k, the function f depends on k and on the property. Their proof relies deeply on tree-automata and the fact that if a graph has clique-width k, then it can be generated by a balanced term that uses operations defining clique-width and a number of basic graphs that depends only on k. We will see how to get a similar result for properties expressible in first order logic on certain classes of graphs of unbounded clique-width.

Graph Classes of Unbounded Clique-Width

Many *real-life* classes of graphs do not have bounded clique-width (Johansson proved for instance that random graphs have unbounded clique-width [Joh98]). Moreover, it is interesting by itself to study classes of graphs of unbounded clique-width from an algorithmic point of view. However, for many of them - bounded degree graph classes, planar graphs, or classes

of graphs that exclude an *apex graph*[1] as a minor - for each positive integer r, the tree-width of the ball of radius r of each vertex is bounded by a function that depends only on r. In addition, Baker and later Eppstein et al. used this fact to give polynomial time approximation algorithms for many NP-Hard problems in respectively planar graphs [Bak94] and classes of graphs that exclude an apex graph as a minor[Epp00, DH04a, DH04b].

In this thesis we investigate f-labelings in graph classes of unbounded clique-width that are obtained by *gluing* graphs of small clique-width and particularly classes of graphs of bounded local clique-width (the clique-width of each ball of radius r is bounded by a function depending on r). We can cite graph classes of bounded degree, planar graphs or classes of graphs that exclude an apex graph as a minor as such classes of graphs. This work is inspired by the results of Frick and Grohe [FG01b, Fri04] stating that every property expressible in first order logic admits an almost linear time algorithm in graph classes of bounded local tree-width (the tree-width of each ball of radius r is bounded by a function depending on r). We adapt to our case several of their definitions.

Dynamic Graphs

In graph theory, most of the results deal with static graphs, i.e., with graphs that do not change in time. However, in *real-life* networks - the Internet network or any cellular telephone network - vertices may be lost or some new vertices can join the network. We model these networks by dynamic graphs. The algorithms in dynamic graphs must take into account these changes. And there is a particular challenge for dynamic graphs in algorithmic graph theory.

> If an algorithm uses a data structure, how can one maintain the data structure at each change without recomputing it in total? Moreover, the maintaining time must be bounded in terms of the number of changes.

Paul et al. study this question for recognition algorithms [CP06, GP07] and, Thorup et al. construct oracles for routing queries in networks [TZ05, PT07, DTCR08].

When we deal with f-labelings of properties in dynamic graphs, it is not clear how to compute the labels of the new vertices without recomputing all the labels. Indeed, in most cases the algorithm that computes the labels takes into account the whole graph (c.f. [CV03]) and not only each node locally. To our knowledge only labelings for routing and distance queries are considered in dynamic trees [KPR04, Kor07a, KP07, Kor08].

In this thesis we investigate log-labelings for the existence of paths in sub-graphs of planar graphs. We then extend the result to other classes of graphs obtained by *planar gluings* of graphs of small clique-width with limited overlaps. For that purpose we introduce two new types of graph decomposition based respectively on partitions of vertices and on partitions of edges. They can be seen as generalizations of the *strong tree-decomposition* of Seese [See85] and of tree-decomposition.

---

[1]A graph is an *apex graph* if there exists a vertex such that its deletion yields a planar graph.

Overview of the Results

The thesis is organized into two parts: Part I, composed of Chapters 2 to 7, is devoted to graph classes of bounded clique-width through the notion of rank-width and Part II, composed of Chapters 8 to 10, is devoted to labelings on certain graph classes of unbounded clique-width.

In Chapter 1 we present all the necessary definitions (sets, graphs, terms, ...), the notions of *clique-width*, *rank-width*, *vertex-minor* and the notions of *graph operation* and of *monadic second order logic*. We also present some basic results used throughout the other chapters.

Chapter 2 introduces the notion of a σ-*symmetric matrix*, which generalizes that of a symmetric matrix. We define a notion of rank-width for σ-symmetric matrices and derive from the works by Bouchet [Bou87, Bou88] a notion of vertex-minor for them. We prove in particular that σ-symmetric matrices of rank-width at most k are characterized by a finite list of σ-symmetric matrices, which generalizes the result of Oum [Oum05b] in undirected graph classes of bounded rank-width.

In Chapter 3 we derive from the results of Chapter 2 a notion of rank-width for directed graphs, called $GF(4)$-*rank-width*, and a notion of vertex-minor for directed graphs. We also define another notion of rank-width for directed graphs, called *bi-rank-width*, and based on a representation of directed graphs by an adjacency matrix over $GF(2)$. We prove that these two notions are equivalent, in the sense that the same classes of graphs have bounded bi-rank-width and bounded $GF(4)$-rank-width. We finally derive from the works by Bouchet and Fon-Der-Flaass [Bou87, FDF96] a notion of vertex-minor, named *bvertex-minor*, for bi-rank-width.

In Chapter 4 we define algebraic graph operations that characterize exactly undirected graphs of rank-width at most k as follows:

> an undirected graph G has rank-width at most k if and only if it is generated by a term in $T(R_k, C_k)$

where R_k is a finite set of binary graph operations, C_k is a finite set of constants, both depending on k. We will also give algebraic graph operations for $GF(4)$-rank-width and bi-rank-width.

In Chapter 5 we prove that the two notions of rank-width for directed graphs are equivalent to clique-width. We give a cubic time algorithm to check if a directed graph has $GF(4)$-rank-width (resp. bi-rank-width) at most k. As a consequence, we propose a cubic time approximation algorithm for directed graphs of clique-width at most k.

Chapter 6 defines a uniform framework for constructing balanced graph decompositions. Our framework unifies the results of Bodlaender et al. [Bod88, BH98] and of Courcelle et al. [CV03, CT07]. We will also apply it to rank-width and prove in particular that undirected graph classes of rank-width at most k admit rank-decompositions of logarithmic height and of width at most $2k$.

Chapter 7 relates minor inclusion and vertex-minor inclusion. As a consequence we prove that rank-width is linearly bounded in term of tree-width.

In Chapter 8 we show that each property expressible in first order logic admits a log-labeling on certain classes of graphs of bounded local clique-width. These classes contain

classes of graphs that exclude an apex graph as a minor, unit-interval graphs and classes of graphs of bounded degree.

In Chapter 9 we show that there exists a log-labeling for connectivity (the existence of paths) in sub-graphs of planar graphs. For that purpose we will combine geometrical tools, logical tools from [CV03] and bipolar orientations of 2-connected planar graphs.

In Chapter 10 we introduce two new types of graph decomposition with associated graph parameters, based respectively on partitions of vertices and partitions of edges. We extend the log-labeling of Chapter 9 to classes of graphs that have bounded width with respect to the two decompositions. These classes of graphs can be seen as *planar gluings* of graphs of small clique-width with limited overlaps.

Chapter 1

Notations and Basic Definitions

Sets. For two sets A and B, we let $A - B$ be the set $\{x \in A \mid x \notin B\}$ and we let $A \triangle B$ be the set $(A - B) \cup (B - A)$. When the context is clear we will write u to denote the set $\{u\}$. Let V be a set and $X \subseteq V$, $\overline{X}$ denotes the set $V - X$ and 2^V denotes the power-set of V. All graphs and trees in this thesis are finite. We denote by $\mathbb{N}$ and $\mathbb{R}$ the fields of natural integers and of real numbers respectively. For every positive integer k, we let $[k]$ be the set $\{1, \ldots, k\}$ ($[0]$ is then the empty set). If F is a finite field, we denote by F^k, for any positive integer k, the set of row-vectors over F of length k.

Relational Structures. A *relational signature* is a finite set $\Sigma = \{R, S, T, \ldots\}$ of relation symbols, each of which given with an *arity* $ar(R) \geq 1$. We denote by $STR[\Sigma]$ the set of all finite relational Σ-structures $\mathfrak{A} = \langle A, (R_{\mathfrak{A}})_{R \in \Sigma} \rangle$ where $R_{\mathfrak{A}} \subseteq A^{ar(R)}$. The set A is called the *domain* of $\mathfrak{A}$. More informations on relational structures can be found in the books [Wec92, Wir] or in the survey [Cou96].

C-Colored Relational Structures. Let C be a finite set of colors and Σ a relational signature. A *C-colored* relational Σ-structure $\mathfrak{A} = \langle A, (R_{\mathfrak{A}})_{R \in \Sigma} \rangle$ is a relational Σ-structure whose elements are all colored with colors in C (uncolored relational Σ-structures are considered as relational Σ-structures whose vertices have all the same color). Formally, a C-colored relational Σ-structure $\mathfrak{A}$ is represented by the relational Γ-structure $\langle A, (R_{\mathfrak{A}})_{R \in \Sigma}, (c_{a,\mathfrak{A}})_{a \in C} \rangle$ where $\Gamma = \Sigma \cup \{c_a \mid a \in C\}$ and such that for every $x \in A$, there exists a unique color a in A such that $c_{a,\mathfrak{A}}(x)$ holds. Therefore, we can denote it by $\langle A, (R_{\mathfrak{A}})_{R \in \Sigma}, lab_{\mathfrak{A}} \rangle$ where $lab_{\mathfrak{A}} : A \to C$.

Disjoint Union. The *disjoint union* $\mathfrak{A} \oplus \mathfrak{B}$ of two relational structures $\mathfrak{A} \in STR[\Sigma]$ and $\mathfrak{B} \in STR[\Gamma]$ is the relational structure $\mathfrak{C} \in STR[\Sigma \cup \Gamma]$ whose domain is $C = A \cup B$ with $A \cap B = \emptyset$ (otherwise one takes a copy of A) and for any $R \in \Sigma \cup \Gamma$, we have $R_{\mathfrak{C}} = R_{\mathfrak{A}} \cup R_{\mathfrak{B}}$ where $R_{\mathfrak{A}} = \emptyset$ if $R \in \Gamma - \Sigma$ and $R_{\mathfrak{B}} = \emptyset$ if $R \in \Sigma - \Gamma$.

Graphs. We deal with directed as well undirected graphs. A simple graph G is represented by the couple (V_G, E_G) where V_G is the set of vertices and E_G is either a set of ordered pairs of vertices if G is *directed*, or a set of unordered pairs if G is *undirected*. E_G is called the

set of *arcs* (or *edges*) if G is directed (or undirected). An arc from x to y is denoted by (x, y) (y is called the *target* and x the *source*). An edge between x and y is denoted by xy (equivalently yx). Two vertices x and y are said *adjacent* in a directed (resp. undirected) graph G if $(x, y) \in E_G$ or $(y, x) \in E_G$ (resp. $xy \in E_G$). We use the term *graph* to denote an undirected as well as a directed graph. Graphs are simple and without loops. We denote by $\mathcal{G}$ the class of all graphs.

A *tree* is an acyclic connected graph. A *forest* is a disjoint union of trees. A tree T is *rooted* if there exists a distinguished node r called the *root* of T. Then a rooted tree is directed so that all nodes are reachable from the root by a directed path. A *rooted forest* is a forest where all the trees, which are its connected components, are rooted. If T is a rooted tree and $u \in V_T$, we denote by $T \downarrow u$ the sub-tree of T rooted at u, induced by the set of all descendants of u. In order to avoid confusions in technical lemmas, the vertices of trees will be called *nodes* and the nodes of degree 1 in rooted trees are called *leaves*.

We denote by $G[X]$ the sub-graph of G induced by $X \subseteq V_G$ and we let $G\backslash X$ be the sub-graph $G[V_G - X]$. For $F \subseteq E_G$ we also denote by $G[F]$ the sub-graph of G induced by $F \subseteq E_G$ ($E_{G[F]} = F$ and $V_{G[F]}$ is the set of vertices incident to an edge in F). The context will specify when using $G[Y]$ whether Y is a set of vertices or a set of edges/arcs. For $x \in V_G$, we denote by $N_G(x)$ the set of vertices adjacent to x; a vertex in $N_G(x)$ is called a *neighbor* of x.

A simple graph G can be seen as a relational $\{E\}$-structure where E, a binary relation, is symmetric when G is undirected. In this case a graph G is the relational $\{E\}$-structure $\langle V_G, E_G \rangle$ where V_G is its set of vertices and $E_G \subseteq V_G \times V_G$. For a graph G, a vertex x and an edge/arc e, we let $inc_G(e, x)$ express that e is incident with x. A graph G can also be seen as a relational $\{inc\}$-structure and be represented by the relational $\{inc\}$-structure $\langle V_G \cup E_G, inc_G \rangle$ where V_G is its set of vertices, E_G its set of edges and $inc_G \subseteq E_G \times V_G$. From now on by graphs we mean relational $\{E\}$-structures, unless otherwise specified.

Terms. Let F be a set of functions and C a set of constants. We denote by $T(F, C)$ the set of finite well-formed terms built with $F \cup C$. They will be handled also as labeled, directed and rooted ordered trees in the usual way. The tree corresponding to a term t in $T(F, C)$ has for set of nodes the set N_t of occurrences in t of the symbols from $F \cup C$; its root is the occurrence of the first symbol in the usual prefix notation; it is directed so that every node is reachable from the root by a directed path; each node is labeled by the symbol of which it is an occurrence and edges are ordered so as to represent the order of arguments of a function symbol. For a term $t \in T(F, C)$, we denote by $Occ_L(t)$ the finite set of occurrences of constants in t and by $Synt(t)$ the syntactic tree of t.

Let F be a set of binary functions. We define the *reduced term* of $t \in T(F, C)$ as $red(t) \in T(\{*\}, \{\#\})$ where $*$ is binary and $\#$ is a constant. It is obtained by replacing every binary symbol by $*$, every constant by $\#$ and by deleting the unary symbols. Formally,

$$red(t) = \# \qquad\qquad\qquad\qquad \text{if } t \in C,$$
$$red(f(t)) = red(t) \qquad\qquad\qquad \text{if } f \in F \text{ is unary,}$$
$$red(f(t_1, t_2)) = *(red(t_1), red(t_2)) \qquad \text{if } f \in F \text{ is binary.}$$

Notice that $Synt(t)$ and $red(t)$ are rooted. We now define the notion of context which will

be mostly used when dealing with balanced terms.

Definition 1.1 (Contexts) Let F be a set of functions and C a set of constants. A *context* is a term in $T(F, C \cup \{u\})$ having a single occurrence of the variable u (a nullary symbol). We denote by $Cxt(F, C)$ the set of contexts and by Id the particular context u. Let s be a context and t a term or context, we denote by $s[t/u]$ the term or context obtained by replacing u in s by t. We define two binary operations on terms and contexts:

$$\begin{cases} s \circ s' = s[s'/u] & \text{belonging to } Cxt(F, C) \text{ for } s, s' \text{ in } Cxt(F, C), \\ s \bullet t = s[t/u] & \text{belonging to } T(F, C) \text{ for } s \text{ in } Cxt(F, C) \text{ and } t \text{ in } T(F, C). \end{cases}$$

Equivalence of Graph Parameters. A *graph parameter* wd is a function $\mathcal{G} \to \mathbb{N}$ that is invariant under isomorphism. Two graph parameters, say wd and wd', are *equivalent* if there exist two increasing integer functions f and g such that for any graph G,

$$f(wd'(G)) \leq wd(G) \leq g(wd'(G)).$$

Composition of Multivalued Functions. Let $f : A \to 2^B$ and $g : B \to 2^C$ be two multivalued functions. We denote by $g \circ f$ the mapping $A \to 2^C$ such that $g \circ f(a) = g(f(a)) = \bigcup \{\beta(b) \mid b \in f(a)\}$. We also use $\circ$ for the normal composition of unary functions. We denote by Id_A the identity function $A \to A$. For sets $A_1, \ldots, A_m, A$, a function $f : A_1 \times \cdots \times A_m \to 2^A$ is called an *m-ary multivalued function*.

We now recall the notions of *clique-width*, *m-clique-width*, *rank-width* and *monadic second order logic*.

1.1 Clique-Width and M-Clique-Width

The definition of *clique-width* is from [CO00].

Definition 1.2 (Clique-Width of Graphs) Let k be a positive integer. We recall the following operations.

(C1) For an undirected $[k]$-colored graph $G = \langle V_G, E_G, lab_G \rangle$ and for distinct $i, j \in [k]$, we denote by $\eta_{i,j}(G)$ the undirected $[k]$-colored graph $K = \langle V_G, E_K, lab_G \rangle$ where

$$E_K = E_G \cup \{xy \mid x, y \in V_G \text{ and } x \neq y \text{ and } i = lab_G(x), \ j = lab_G(y)\}.$$

(C1') For a directed $[k]$-colored graph $G = \langle V_G, E_G, lab_G \rangle$ and for distinct $i, j \in [k]$, we denote by $\alpha_{i,j}(G)$ the directed $[k]$-colored graph $K = \langle V_G, E_K, lab_G \rangle$ where

$$E_K = E_G \cup \{(x, y) \mid x, y \in V_G \text{ and } x \neq y \text{ and } i = lab_G(x), \ j = lab_G(y)\}.$$

(C2) For a $[k]$-colored graph $G = \langle V_G, E_G, lab_G \rangle$ and for distinct $i, j \in [k]$, we denote by $\rho_{i \to j}(G)$ the $[k]$-colored graph $K = \langle V_G, E_G, lab_K \rangle$ where

$$lab_K(x) = \begin{cases} j & \text{if } lab_G(x) = i, \\ lab_G(x) & \text{otherwise.} \end{cases}$$

(C3) For each $i \in [k]$, $\mathbf{i}$ denotes a $[k]$-colored graph with a single vertex colored by i and no edge. We let $C_k^c = \{\mathbf{i} \mid i \in [k]\}$.[1]

For $k \geq 1$ we let F_k^{uc} be the set $\{\oplus, \eta_{i,j}, \rho_{i \to j} \mid i, j \in [k]\}$ and F_k^{dc} be the set $\{\oplus, \alpha_{i,j}, \rho_{i \to j} \mid i, j \in [k]\}$. Every term t in $T(F_k^{uc}, C_k^c)$ (resp. $T(F_k^{dc}, C_k^c)$) defines, up to isomorphism, an undirected (resp. a directed) $[k]$-colored graph $val(t)$. The *clique-width* of a graph G, denoted by $cwd(G)$, is the minimum k such that $G = val(t)$ where $t \in T(F_k^{uc}, C_k^c)$ if G is undirected and $t \in T(F_k^{dc}, C_k^c)$ if G is directed.

Example 1.1 gives two examples of terms that generate respectively an undirected graph and a directed graph.

Example 1.1 We will use the notation $\mathbf{i}(x)$ to mean that the vertex x is in bijection with this constant $\mathbf{i}$. We let $G_0, G_1, \ldots, G_5$ and G be the undirected graphs on Figure 3 (i)-(vii). We will explain how to construct G by means of clique-width operations and we will use for that purpose the undirected graphs $G_0, \ldots, G_5$, which are sub-graphs of G. They illustrate how G is constructed. The graph G_0 is generated by the term $t_0 = (\mathbf{1}(c) \oplus \mathbf{1}(d)) \oplus (\mathbf{2}(a) \oplus \mathbf{3}(b))$. It consists of 4 isolated vertices. The graph G_1 is generated by the term $t_1 = \eta_{1,2}(t_0)$, which adds the edges between a, colored by 2 and, c and d, colored by 1. It is worth noticing that the term $\eta_{2,1}(t_0)$ also generates G_1. The term $t_2 = \eta_{1,3}(t_1)$ adds edges between b, colored by 3 and, c and d, colored by 1. Then it generates the graph G_2. The graph G_3 is generated by the term $t_3 = \eta_{4,2}(t_1 \oplus \mathbf{4}(e))$. The term t_3 adds the vertex e, colored by 4, and adds the edge between a and e. The term $t_4 = \rho_{4 \to 3}(\rho_{3 \to 2}(\rho_{2 \to 1}(t_3)))$ changes respectively the color of a into 1 with the operation $\rho_{2 \to 1}$, the color of b into 2 with the operation $\rho_{3 \to 2}$ and the color of e into 3 with the operation $\rho_{4 \to 3}$. It does not change the other colors and then generates the graph G_4. The graph G_5 is generated by the term $t_5 = \rho_{2 \to 1}(\rho_{4 \to 3}(\eta_{4,3}(\eta_{4,2}(\mathbf{4}(f) \oplus t_4))))$. It adds the edges between f and, e and b. The graph G is isomorphic to the graph generated by the term $\eta_{2,3}(t_4 \oplus \mathbf{2}(g))$. It is also isomorphic to the graph generated by the term $\eta_{4,3}(t_4 \oplus \mathbf{4}(g))$. Then the graph G has clique-width at most 4. We can prove that it has exactly clique-width 4.

We now consider the directed graphs $\overrightarrow{G_1}, \overrightarrow{G_2}$ and $\overrightarrow{G}$ on Figure 4 (i)-(iii). The directed graphs $\overrightarrow{G_1}$ and $\overrightarrow{G_2}$ are used in order to illustrate how $\overrightarrow{G}$ is constructed by means of clique-width operations. The graph $\overrightarrow{G_1}$ is generated by the term $\overrightarrow{t_1} = \rho_{4 \to 1}(\alpha_{1,2}(\alpha_{2,3}(\alpha_{4,3}((\mathbf{1}(x_3) \oplus \mathbf{2}(x_4)) \oplus (\mathbf{3}(x_5) \oplus \mathbf{4}(x_2))))))$. This term first create the vertices x_3, x_4, x_5 and x_2 and then, adds the arcs (x_5, x_2), (x_4, x_5) and (x_3, x_4), in this order. The graph $\overrightarrow{G_2}$ is generated by the term $\overrightarrow{t_2} = \rho_{4 \to 2}(\rho_{3 \to 2}(\alpha_{4,1}(\alpha_{4,3}(\alpha_{2,4}(\mathbf{4}(x_1) \oplus \overrightarrow{t_1})))))$. The graph $\overrightarrow{G}$ is isomorphic to the graph generated by the term $\alpha_{1,3}(\mathbf{3}(x_6) \oplus \overrightarrow{t_2})$. Then the graph $\overrightarrow{G}$ has clique-width at most 4 (we can prove that it has exactly clique-width 4).

Contrary to the case of tree-width, there is no known polynomial-time algorithm for the recognition of graphs of clique-width at most k for fixed $k > 3$ (for $k \leq 3$ there exists one for undirected graphs [CHL$^+$00]) which produces a term that uses $cwd(G)$ colors and that defines G. Note that the clique-width checking problem is NP-Hard when k is part of the input [FRRS06].

---

[1]We can define constants with loops by letting for each i the constant $\mathbf{i}^\ell$ be a graph with single vertex colored by i with a loop. However we will not need them.
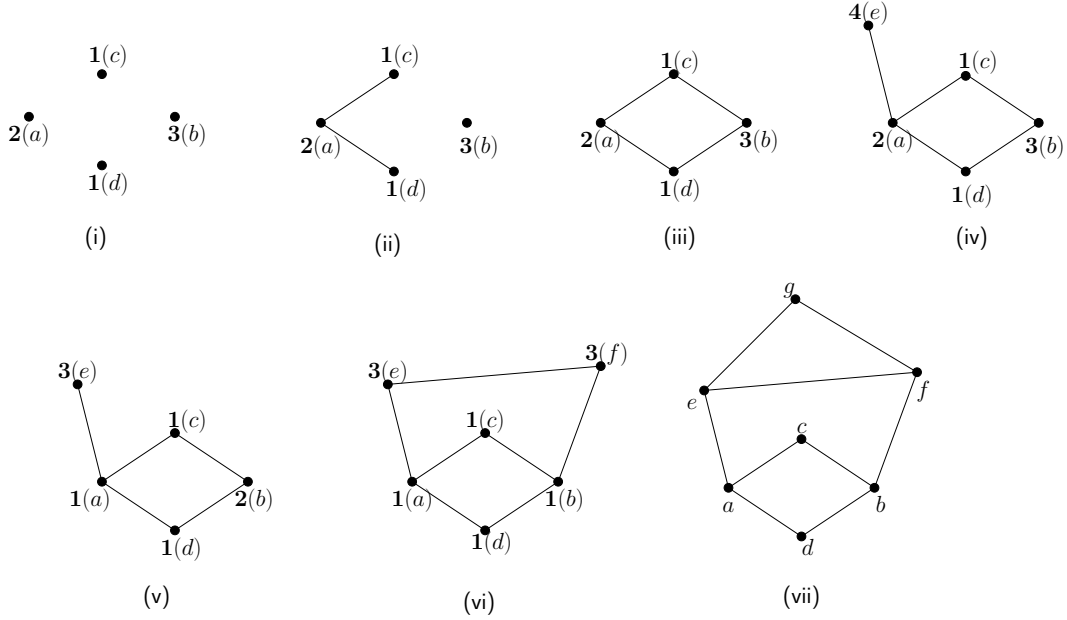
Figure 3: Undirected graphs of Example 1.1.
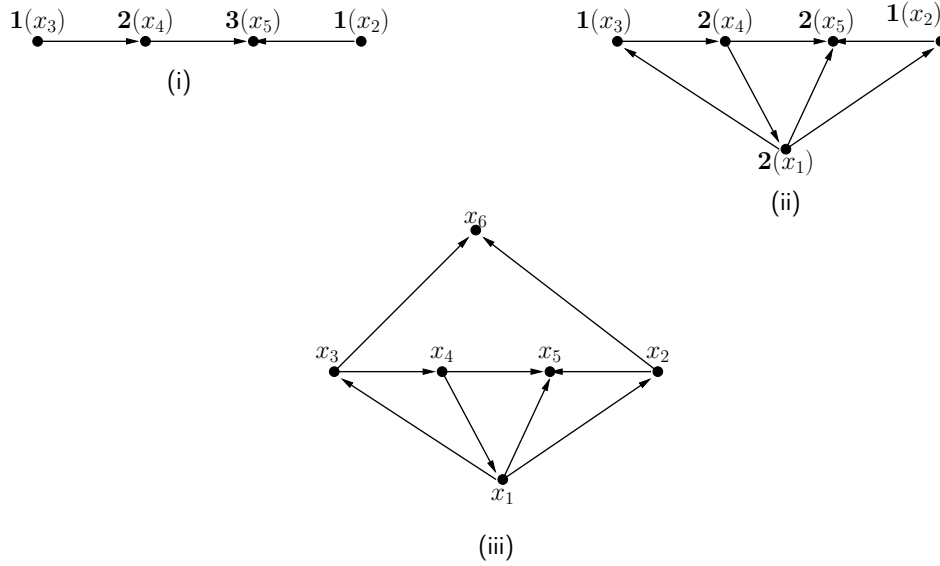
Figure 4: Directed graphs of Example 1.1.

We now define the notion of *edge-colored graphs* used in Section 4.3 for defining graph operations that handle algebraically the notion of bi-rank-width (Section 3.1). We will also define the notion of clique-width of edge-colored graphs, a notion used in Chapter 10. The notion of clique-width of edge-colored graphs is used in [FMR08] for counting the number of assignments of a propositional formula.

Definition 1.3 (Edge-Colored Graphs and Clique-Width of Edge-Colored Graph)
Let A be a finite set. An *A-edge-colored* graph is a graph whose edges/arcs are colored by colors

in A. An A-edge-colored graph G can be seen as the relational structure $\langle V_G, (E_G^a)_{a \in A} \rangle$, also denoted by G, where for every a in A and every pair of vertices (x, y), $E_G^a(x, y)$ holds if and only if there is an edge between x and y colored by a. It is undirected if for all pair of vertices (x, y) and all a in A we have $E_G^a(x, y)$ holds if and only if $E_G^a(y, x)$ holds. We define the following operations.

(AC1) For an undirected $[k]$-colored A-edge-colored graph[2] $G = \langle V_G, (E_G^a)_{a \in A}, lab_G \rangle$, for a color b in A and for distinct $i, j \in [k]$, we denote by $\eta_{i,j}^b(G)$ the undirected $[k]$-colored A-edge-colored graph $K = \langle V_G, (E_K^a)_{a \in A}, lab_G \rangle$ where:

$$E_K^a = \begin{cases} E_G^a & \text{if } a \neq b, \\ E_G^a \cup \{(x, y), (y, x) \mid x, y \in V_G \text{ and } x \neq y \text{ and } i = lab_G(x), \ j = lab_G(y)\} & \text{otherwise.} \end{cases}$$

We add b-colored edges between vertices colored by i and vertices colored by j. The colors of the vertices are not modified.

(AC1') For a directed $[k]$-colored A-edge-colored graph $G = \langle V_G, (E_G^a)_{a \in A}, lab_G \rangle$, for a color b in A and for distinct $i, j \in [k]$, we denote by $\alpha_{i,j}^b(G)$ the directed $[k]$-colored A-edge-colored graph $K = \langle V_G, (E_K^a)_{a \in A}, lab_G \rangle$ where

$$E_K^a = \begin{cases} E_G^a & \text{if } a \neq b \\ E_G^a \cup \{(x, y) \mid x, y \in V_G \text{ and } x \neq y \text{ and } i = lab_G(x), \ j = lab_G(y)\} & \text{otherwise.} \end{cases}$$

We add b-colored arcs between vertices colored by i and vertices colored by j. The colors of the vertices are not modified.

(AC2) For a $[k]$-colored A-edge-colored graph $G = \langle V_G, (E_G^a)_{a \in A}, lab_G \rangle$ and for distinct $i, j \in [k]$, we denote by $\rho_{i \to j}(G)$ the $[k]$-colored A-edge-colored graph $K = \langle V_G, (E_G^a)_{a \in A}, lab_K \rangle$ where

$$lab_K(x) = \begin{cases} j & \text{if } lab_G(x) = i, \\ lab_G(x) & \text{otherwise.} \end{cases}$$

We just recolor the vertices of G. The colors of the edges are not modified.

We let $F_{k,A}^u = \{\oplus, \eta_{i,j}^a, \rho_{i \to j} \mid i, j \in [k], a \in A\}$ and $F_{k,A}^d = \{\oplus, \alpha_{i,j}^a, \rho_{i \to j} \mid i, j \in [k], a \in A\}$. The clique-width of an A-edge-colored graph G is the minimum k such that G is isomorphic to $val(t)$ for some term t in $T(F_{k,A}^u, C_k^c)$ if G is undirected, otherwise t is in $T(F_{k,A}^d, C_k^c)$.

An A-edge-colored graph is shown on Figure 7 (iii).

Let us now define the last notion of this section, the one of *m-clique-width*, a graph parameter, defined as clique-width in terms of graph operations, and equivalent to clique-width. Courcelle and Twigg used it to prove that we can label each vertex of an undirected n-vertex graph G of clique-width k by a bit sequence of size at most $O(k^2 \cdot \log(n)^2)$ and determine for any $X \subseteq V_G$, the distance of two vertices u and v in $G \setminus X$, just by looking at the labels of u, v and of the vertices in X [CT07].

---

[2]The vertices and the edges of G are colored.

Definition 1.4 (M-Clique-Width) Let k be a positive integer. A k-*multicolored graph* is a $2^{[k]}$-colored graph. Hence a vertex may have zero, one or several colors in $[k]$. We define the following operations.

(M1) for $R \subseteq [k]^2$, for mappings $g, h : [k] \rightarrow 2^{[k]}$, called *recolorings*, and for undirected k-multicolored graphs $G = \langle V_G, E_G, lab_G \rangle$ and $H = \langle V_H, E_H, lab_H \rangle$ such that $V_G \cap V_H = \emptyset$ (otherwise we replace H by a disjoint copy) we denote by $G \otimes_{R,g,h} H$ the undirected k-multicolored graph $K = \langle V_G \cup V_H, E_K, lab_K \rangle$ where:

$$E_K = E_G \cup E_H \cup \{xy \mid x \in V_G, \ y \in V_H \text{ and } R \cap (lab_G(x) \times lab_H(y)) \neq \emptyset\},$$

$$lab_K(x) = \begin{cases} (g \circ lab_G)(x) = \{a \mid a \in g(b), b \in lab_G(x)\} & \text{if } x \in V_G, \\ (h \circ lab_H)(x) & \text{if } x \in V_H. \end{cases}$$

(M2) for $R, R' \subseteq [k]^2$, for *recolorings* $g, h : [k] \rightarrow 2^{[k]}$ and for directed k-multicolored graphs $G = \langle V_G, E_G, lab_G \rangle$ and $H = \langle V_H, E_H, lab_H \rangle$ such that $V_G \cap V_H = \emptyset$ (otherwise we replace H by a disjoint copy) we denote by $G \otimes_{R,R',g,h} H$ the directed k-multicolored graph $K = \langle V_G \cup V_H, E_K, lab_K \rangle$ where:

$$E_K = E_G \cup E_H \cup \{(x,y) \mid x \in V_G, \ y \in V_H \text{ and } R \cap (lab_G(x) \times lab_H(y)) \neq \emptyset\}$$
$$\cup \{(y,x) \mid x \in V_G, \ \ y \in V_H \text{ and } R' \cap (lab_G(x) \times lab_H(y)) \neq \emptyset\}$$

$$lab_K(x) = \begin{cases} (g \circ lab_G)(x) & \text{if } x \in V_G, \\ (h \circ lab_H)(x) & \text{if } x \in V_H. \end{cases}$$

(M3) For each $A \subseteq [k]$, $\mathbf{A}$ denotes a k-multicolored graph with a single vertex colored by A and no edge/arc. We let $C_k^{um} = \{\mathbf{A} \mid A \subseteq [k]\}$.

For $k \geq 1$ we let $F_k^{um} = \{\otimes_{R,f,g} \mid R \subseteq [k]^2, \ f, g : [k] \rightarrow [k]\}$ and $F_k^{dm} = \{\otimes_{R,R',f,g} \mid R, R' \subseteq [k]^2, \ f, g : [k] \rightarrow [k]\}$. Every term $t \in T(F_k^{um}, C_k^{um})$ (resp. $t \in T(F_k^{dm}, C_k^{um})$) defines, up to isomorphism, an undirected (resp. directed) k-multicolored graph $val(t)$. The *m-clique-width* of an undirected (resp. directed) graph G, denoted by $mcwd(G)$, is the minimum k such that $G = val(t)$, where $t \in T(F_k^{um}, C_k^{um})$ (resp. $t \in T(F_k^{dm}, C_k^{um})$).

One can verify that the graph on Figure 3 (vii) has m-clique-width 2. M-clique-width and clique-width are two equivalent graph parameters as proved in the following.

Proposition 1.1 ([CT07]) *For every graph G, $mcwd(G) \leq cwd(G) \leq 2^{mcwd(G)+1}$.*

We will show in Chapter 6 that if an undirected graph G with n vertices has m-clique-width at most k then, it is the value of a 3-balanced term, i.e., of height at most $3 \cdot \log(n)$, belonging to $T(F_{2k}^{um}, C_{2k}^{um})$.

In their investigations of recognition algorithms for the clique-width of undirected graphs, Oum and Seymour introduced the notion of *rank-width of undirected graphs*, which is a graph parameter equivalent to clique-width for undirected graphs. We now introduce this notion.

1.2   Rank-Width and Vertex-Minor

We recall the definition of *rank-width* of undirected graphs [Oum05b, OS06], based on the *branch-width of a symmetric function*, and give some results. Other results will be recalled when necessary.

Let V be a set and $f : 2^V \to \mathbb{Z}$. We say that f is *symmetric* if for any $X \subseteq V$, $f(X) = f(\overline{X})$; f is *submodular* if for any $X, Y \subseteq V, f(X \cup Y) + f(X \cap Y) \le f(X) + f(Y)$. We first define the notion of *layout of symmetric functions* [RS91].

Definition 1.5 (Sub-Cubic Trees) A *sub-cubic tree* is a tree such that the degree of each node is at most 3. By replacing in a sub-cubic tree T every induced path $x - u_1 - u_2 - \cdots - u_n - y$ by the edge $x - y$, and by deleting the intermediate vertices $u_1, \ldots, u_n$, one transforms T into a tree T' such that every node has degree 1 or 3 and $N_{T'}^{(1)} = N_T^{(1)}$ (we denote by $N_T^{(i)}$ the set of nodes of degree i). We will denote T' by $Red(T)$.

Definition 1.6 (Layout) A *layout of a symmetric function* $f : 2^V \to \mathbb{Z}$ is a pair $(T, \mathcal{L})$ of a sub-cubic tree T and a bijective function $\mathcal{L} : V \to N_T^{(1)}$. Each edge e of T induces a bipartition $(X_e, \overline{X}_e)$ of $N_T^{(1)}$, and thus a bipartition $(X^e, \overline{X}^e) = (\mathcal{L}^{-1}(X_e), \mathcal{L}^{-1}(\overline{X}_e))$ of V. (By convention if T is rooted and $e = (u, w)$ we will assume that $X^e = \mathcal{L}^{-1}(N_{T \downarrow w}^{(1)})$; we will omit the superscript e when the context is clear.)

We can now introduce the *branch-width of a symmetric function* [RS91].

Definition 1.7 (Branch-Width) Let $(T, \mathcal{L})$ be a layout of a symmetric function f. The *branch-width of an edge* e of T is $f(X^e)$. The *branch-width of a layout* $(T, \mathcal{L})$, denoted by $bwd(f, T, \mathcal{L})$, is the maximum branch-width over all edges of T. The *branch-width of* f, denoted by $bwd(f)$, is the minimum branch-width over all layouts of f.

One can verify easily that if $(T, \mathcal{L})$ is a layout of branch-width k of a symmetric function f, then $(Red(T), \mathcal{L})$ is also a layout of branch-width k of f.

One example of a graph parameter defined by using the branch-width of symmetric functions is the notion of *branch-width of a graph*, a graph parameter equivalent to tree-width and defined by Robertson and Seymour [RS91] (see the surveys [Bod98, Bod05]).

Definition 1.8 (Branch-Width of a Graph [RS91]) Let $G = \langle V_G, E_G \rangle$ be a graph. For a set X of edges, let T_X be the set of vertices incident to at least one edge in X. Let $\eta : 2^{E_G} \to \mathbb{N}$ be defined such that for each $X \subseteq E_G$, $\eta(X) = \left| \{T_X \cap T_{E_G-X}\} \right|$. The function η is symmetric. The *branch-width of* G, denoted by $bwd(G)$, is the branch-width of η.

We will use the notion of branch-width of a graph in Section 7.4. It is important to not confuse the branch-width of a graph with the branch-width of a symmetric function.

We can now define the rank-width of undirected graphs, also based on branch-width of symmetric functions. We let rk be the *rank-function* of matrices [Lip91]. 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. We simply write R-matrix when $R = C$. We call $|R| \times |C|$ the *order* of an (R, C)-matrix. For an (R, C)-matrix M, if $X \subseteq R$ and $Y \subseteq C$ we let M_X^Y be the sub-matrix of

M where the rows and the columns are indexed by X and Y respectively and we let $M \backslash X$ be the $(R - X, C - X)$-matrix M_{R-X}^{C-X}. The *adjacency matrix* of a graph $G = \langle V_G, E_G \rangle$ (directed or not) is the V_G-matrix A_G where $(A_G)_x^y = 1$ if and only if there is an edge or an arc between x and y.

Definition 1.9 ([Oum05a]) Let G be an undirected graph. For any $X \subseteq V_G$, we let $\rho_G(X) = rk\left((A_G)_X^{\overline{X}}\right)$. Then ρ_G is a symmetric function. The *rank-width of an undirected graph* G, denoted by $rwd(G)$, is the branch-width of the function ρ_G.

Example 1.2 We let G be the undirected graph on Figure 3 (vii). The induced sub-graph $G[\{e, a, c, b, f\}]$ is isomorphic to the cycle C_5. Hence, G is not a distance-hereditary graph. Therefore, the rank-width of G is at least 2. Figure 5 shows a layout of the function ρ_G of branch-width 2 because $\rho_G(\{e, f, g\}) = rk(A) = 2$ where:

$$A = \begin{array}{c|c|c|c|c|} & a & b & c & d \\ \hline e & 1 & 0 & 0 & 0 \\ \hline f & 0 & 1 & 0 & 0 \\ \hline g & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

Hence the rank-width of G is 2.
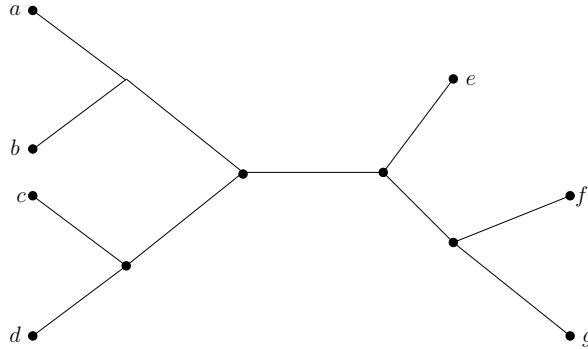
Figure 5: A layout of the function ρ_G where G is the undirected graph on Figure 3 (vii).

Oum and Seymour proved that rank-width is equivalent to clique-width [OS06].

Proposition 1.2 ([OS06]) *For an undirected graph* G, $rwd(G) \leq cwd(G) \leq 2^{rwd(G)+1} - 1$.

In [Oum05b] Oum defines the notion of *vertex-minor of undirected graphs* derived from the works of Bouchet on circle graphs [Bou88].

Definition 1.10 Let G be an undirected graph and $x \in V_G$. The undirected graph obtained by applying a *local complementation at* x to G is

$$G * x = \langle V_G, E_G \triangle \{yz \mid xy, xz \in E_G, z \neq y\} \rangle.$$

An undirected graph H is *locally equivalent to* an undirected graph G if H can be obtained by applying a sequence of local complementations to G. We say that H is a *vertex-minor* of G if H can be obtained by applying a sequence of vertex deletions and local complementations to G.

The graph $G * x$ is obtained by edge-complementing the sub-graph of G induced by the vertices adjacent to x. We now recall the following by Oum.

Lemma 1.1 ([Oum05b]) *Let G be an undirected graph. If H is locally equivalent to G then the rank-width of H is equal to the rank-width of G. If H is a vertex-minor of G then the rank-width of H is at most the rank-width of G.*

Example 1.3 Figure 6 shows the undirected graph $G * e$ where G is the undirected graph on Figure 3 (vii). We can verify that the layout on Figure 5 is also a layout for $G * e$ of same branch-width because $\rho_{G*e}(\{e, f, g\}) = rk(A') = 2$ where

$$A' = \begin{array}{c|c|c|c|c} & a & b & c & d \\ \hline e & 1 & 0 & 0 & 0 \\ \hline f & 1 & 1 & 0 & 0 \\ \hline g & 1 & 0 & 0 & 0 \end{array}$$

$$G * e$$
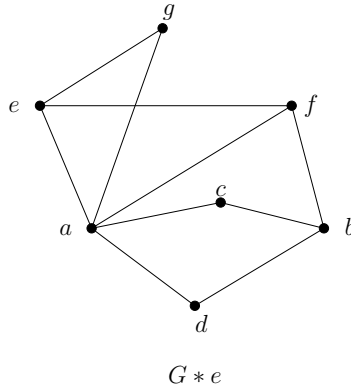
Figure 6: Local complementation at e of the undirected graph on Figure 3 (vii).

Undirected graphs of rank-width at most k are characterized by a finite list of excluded vertex-minors as stated in the following.

Theorem 1.1 ([Oum05b]) *For each k there is a finite list $\mathcal{C}_k$ of undirected graphs having at most $(6^{k+1} - 1)/5$ vertices such that an undirected graph G has rank-width at most k if and only if no undirected graph in $\mathcal{C}_k$ is isomorphic to a vertex-minor of G.*

As for clique-width, checking the rank-width of a graph is NP-complete [HOSG08] when k is part of the input. However, for fixed k there exists a cubic-time FPT algorithm that decides if a given graph has rank-width at most k due to Oum and Hliněný. (For fixed k, several approximation algorithms for recognizing undirected graphs of rank-width at most k had been given by Oum et al. [Oum05a, OS06].)

Theorem 1.2 ([HO07]) *For fixed k there exists an $O(n^3)$-time algorithm that for an undirected graph G with $|V_G| = n$, either outputs a layout of the function ρ_G of branch-width at most k or confirms that the rank-width of G is larger than k.*

Contrary to that of clique-width, the definition of rank-width of undirected graphs is combinatorial. An algebraic characterization of rank-width of undirected graphs has been given in terms of algebraic graph operations [CK09]. We will give the proof in Chapter 4 in a more general context.

As for clique-width, we will extend the notion of rank-width of undirected graphs to undirected A-edge-colored graphs.

Definition 1.11 Let A be a finite set of colors. If G is an undirected A-edge-colored graph then, for each $a \in A$ we let G_a be the sub-graph of G consisting of V_G and its a-colored edges, i.e., $G_a = \langle V_G, E_G^a \rangle$. A *layout* of a graph G is a pair $(T, \mathcal{L})$ where T is a sub-cubic tree and $\mathcal{L} : V_G \to N_T^{(1)}$ is a bijection. For every undirected A-edge-colored graph G we let

$$rwd_A(G) = \min \left\{ \max_{a \in A} \{bwd(\rho_{G_a}, T, \mathcal{L})\} \mid (T, \mathcal{L}) \text{ is a layout of } G \right\}.$$

Example 1.4 Let $A = \{a, b\}$. An undirected A-edge-colored graph G is shown on Figure 7 (iii). The graphs G_a and G_b and, a layout $(T, \mathcal{L})$ of G are shown on Figure 7 (i), (ii) and (iv). One can verify that $(T, \mathcal{L})$ is a layout of branch-width 2 for ρ_{G_a} and ρ_{G_b}.

Figure 7: An A-edge-colored graph G where $A = \{a, b\}$ and a layout of G.

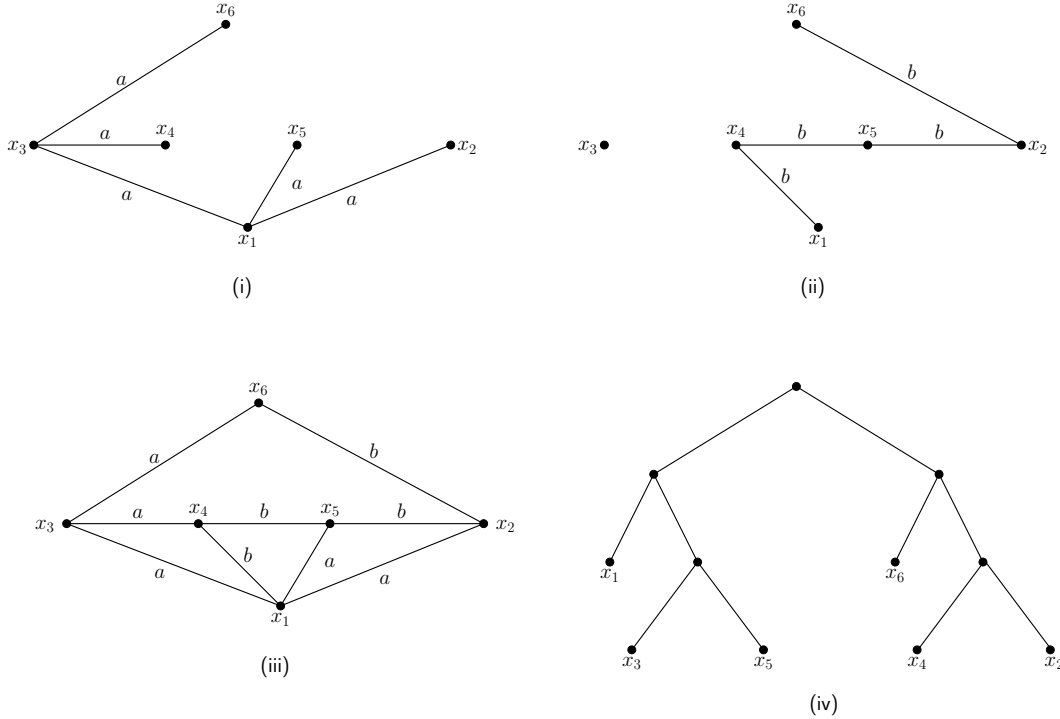## 1.3 Monadic Second Order Logic

In this section we introduce the notions of *first order logic* and of *monadic second order logic*. The terminologies are from [BC06, Cou93].

Let Σ be a relational signature and $\mathfrak{A}$ be a relational Σ-structure. We will use lower case variables $x, y, z, \ldots$, called *FO variables*, to denote elements in A and upper case variables $X, Y, Z, \ldots$, called *set variables*, to denote subsets of A. The Σ-*atomic formulas* are $x = y$, $x \in X$ and $R(x_1, \ldots, x_{ar(R)})$ for any $R \in \Sigma$. A *first order formula over* Σ (FO_Σ formula for short) is a formula formed from Σ-atomic formulas with Boolean connectives $\wedge$, $\vee$, $\neg$, $\Rightarrow$ and *element quantifications* $\exists x$, $\forall x$. A *monadic second order formula over* Σ (MSO_Σ formula for short) is formed from FO_Σ formulas with *set quantifications* $\exists X$, $\forall X$. If the context is clear, we will omit the subscript Σ. An occurrence of a variable which is not under the scope of a quantifier is called a *free variable*; a formula without free variables is called a *sentence*. If χ is a set of lower and upper case variables, we denote by $MSO_\Sigma(\chi)$ (resp. $FO_\Sigma(\chi)$) the set of MSO_Σ formulas (resp. FO_Σ formulas) with free variables in χ. If the free variables of a formula φ are $x_1, \ldots, x_m, Y_1, \ldots, Y_q$ we will write $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$.

If P is a property of relational Σ-structures, we write $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ to mean that P depends on elements of the domains $x_1, \ldots, x_m$ and sets of elements of the domains $Y_1, \ldots, Y_q$. A property $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ of relational Σ-structures is *MSO-definable* (resp. *FO-definable*) if there exists an MSO (resp. FO) formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ such that a relational Σ-structure $\mathfrak{A}$ satisfies P if and only if φ is TRUE in the relational Σ-structure $\mathfrak{A}$. A formula, MSO as well as FO, is *quantifier-free* if the formula contains no quantifier. And we denote by QF_Σ the set of quantifier-free formulas in FO_Σ. For every positive integer m notice that, up to a decidable equivalence that refines logical equivalence, the set $QF_\Sigma(\{x_1, \ldots, x_m\})$ is finite [CW05].

We will denote by MS_1 the set of $MSO_{\{E\}}$ formulas and by MS_2 the set of $MSO_{\{inc\}}$ formulas. The property stating that a graph contains an Hamiltonian cycle is MS_2-definable but not MS_1-definable (this can be proved by showing that if Hamiltonicity is MS_1-definable, then the language $\{a^n b^n\}$ is regular). Thus we can express more properties with MS_2 formulas than with MS_1 formulas. Here is an example of an FO formula that expresses that a vertex x has degree at most 4:

$$\forall y_1 \forall y_2 \forall y_3 \forall y_4 \forall y_5 \left(\bigwedge_{1 \leq i \leq 5} E(x, y_i) \Rightarrow \bigvee_{1 \leq i < j \leq 5} y_i = y_j \right)$$

The following MS_1 formula expresses that a graph is 3-colorable.

$$\exists X_1 \exists X_2 \exists X_3 \left(\bigwedge_{1 \leq i < j \leq 3} \text{``} X_i \cap X_j = \emptyset \text{''} \ \wedge \ \forall x \bigvee_{i \in [3]} x \in X_i \ \wedge \ \forall x \forall y \bigwedge_{i \in [3]} (x \in X_i \wedge y \in X_i \Rightarrow \neg E(x, y)) \right).$$

The formulas that express properties of graphs when we do not care about the structure that represents them are called MS or FO formulas. We use the usual notation $\mathfrak{A} \models \varphi$ to say that the formula φ is TRUE in the relational structure $\mathfrak{A}$. The *quantifier-rank* of a formula φ, denoted by $qr(\varphi)$, is defined inductively as follows:

$$
\begin{aligned}
qr(\varphi) &= 0 & &\text{if } \varphi \text{ is an atomic formula,} \\
qr(\varphi \sigma \psi) &= \max\{qr(\varphi), qr(\psi)\} & &\text{if } \sigma \in \{\wedge, \vee\}, \\
qr(\sigma x \varphi) &= 1 + qr(\varphi) & &\text{if } \sigma \in \{\exists, \forall\}, \\
qr(\neg \varphi) &= qr(\varphi) & &.
\end{aligned}
$$

1.4 Graph Operations

We will now introduce graph operations in a logical point of view in terms of relational structures representing graphs. We use the definitions from [BC06].

Definition Scheme. Let Σ and Γ be two relational signatures and L be a logical language. An *L-definition scheme $\mathcal{D}$ of type $\Sigma \to \Gamma$* is a tuple $(\psi, (\theta_R)_{R \in \Gamma})$ where:

$$\psi \in L_\Sigma(\{x\}),$$
$$\theta_R \in L_\Sigma(\{x_1, \dots, x_{ar(R)}\}) \text{ for each } R \in \Gamma.$$

Let $\mathfrak{A} \in STR[\Sigma]$ and $\mathfrak{B} \in STR[\Gamma]$. We say that $\mathcal{D}$ defines the Γ-structure $\mathfrak{B}$ from $\mathfrak{A}$ if

(QF1) $B = \{a \mid \mathfrak{A} \models \psi(a)\}$,

(QF2) for each $R \in \Gamma$,

$$R_{\mathfrak{B}} = \{(a_1, \dots, a_{ar(R)}) \in B^{ar(R)} \mid \mathfrak{A} \models \theta_R(a_1, \dots, a_{ar(R)})\}.$$

The structure $\mathfrak{B}$ is uniquely determined by $\mathfrak{A}$ and $\mathcal{D}$. Therefore, we can use a functional notation and we write $\mathfrak{B} = \widehat{\mathcal{D}}(\mathfrak{A})$. Notice that if $\mathfrak{A}$ is isomorphic to $\mathfrak{A}'$ then $\widehat{\mathcal{D}}(\mathfrak{A})$ is isomorphic to $\widehat{\mathcal{D}}(\mathfrak{A}')$

We recall that the notions of *definition scheme* and of *quantifier-free operation* are defined in a more general context in [BC06]. However, the definitions given here are enough for our purposes. We recall a result that is proved in detail in [BC06].

Proposition 1.3 *Let Σ and Γ be relational signatures and let $\mathcal{D}$ be an MS-definition scheme of type $\Sigma \to \Gamma$. Then, for every MS formula $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$ on relational Γ-structures, there exist an MS formula $\varphi^{\#}(x_1, \dots, x_m, Y_1, \dots, Y_q)$ on relational Σ-structures such that for every relational Γ-structure $\mathfrak{B}$ and every relational Σ-structure $\mathfrak{A}$ with $\mathfrak{B} = \widehat{\mathcal{D}}(\mathfrak{A})$:*

$$\mathfrak{B} \models \varphi \iff \mathfrak{A} \models \varphi^{\#}$$

Graph Operation. A *quantifier-free operation γ from $STR[\Sigma]$ to $STR[\Gamma]$* is a function defined by a QF_Σ-definition scheme $\mathcal{D}$ of type $\Sigma \to \Gamma$ such that $\gamma(\mathfrak{A}) = \widehat{\mathcal{D}}(\mathfrak{A})$ for all $\mathfrak{A} \in STR[\Sigma]$.

We now give examples of quantifier-free operations. Let k be a fixed positive integer. We recall that an undirected $[k]$-colored graph is represented by the structure $\langle V_G, E_G, c_{1,G}, \dots, c_{k,G} \rangle$ where $c_{i,G}(x)$ is TRUE if and only if x has color i. It is easy to verify that $\eta_{i,j}$ (one of the graph operations that define clique-width, see Section 1.1) is defined by the following quantifier-free operation $(\psi, \theta_E, (\theta_{c_\ell})_{\ell \in [k]})$ where:

$$\psi = \text{TRUE},$$
$$\theta_E(x_1, x_2) = E(x_1, x_2) \vee \left(x_1 \neq x_2 \wedge \left((c_i(x_1) \wedge c_j(x_2)) \vee (c_j(x_1) \wedge c_i(x_2)) \right) \right),$$
$$\theta_{c_\ell}(x) = c_\ell(x) \text{ for } \ell \in [k].$$

Another example of a quantifier-free operation is the *edge-complement*, which is not a basic operation defining clique-width. However it can be used; the decidability result and algorithmic results of [CMR00, CMR01] (FPT algorithms) remain valid. Notice also that edge-addition is a particular case of quantifier-free operation.

In this way we get a unique notion covering several types of graph operations, and uniform proofs for several cases. In the works by Courcelle et al. [BC06, Cou92, Cou93, CER93, CMR00, CMR01, Mak04] it is shown how to verify MS formulas on graphs generated by terms written with quantifier-free operations and disjoint union.

Part I

Graph Classes of Bounded Rank-Width

Chapter 2

Rank-width of σ-Symmetric Matrices

Clique-Width is more general than tree-width because every class of simple graphs that has bounded tree-width has bounded clique-width [CO00, CR05]. However, contrary to tree-width, for fixed $k \geq 4$ there is no polynomial-time algorithm that checks if a given graph has clique-width at most k. If k is part of the input, the clique-width checking as well as the tree-width checking problems are NP-complete [ACP87, FRRS06].

Oum and Seymour [Oum05a, Oum05b, OS06] introduced the notion of rank-width of undirected graphs in their investigations of recognition algorithms for graphs that have clique-width at most k, for fixed k. Rank-width is equivalent to clique-width (Proposition 1.2) but, rank-width has better algorithmic properties than clique-width: undirected graphs of rank-width at most k are characterized by a finite list of excluded configurations (Theorem 1.1) and can be recognized by a cubic-time algorithm (Theorem 1.2). Moreover, this cubic-time recognition algorithm gives rise to a cubic-time approximation algorithm for recognizing undirected graphs of clique-width at most k. However, clique-width is defined for directed as well as undirected graphs. It is thus natural to ask for a similar notion of rank-width for directed graphs.

We introduce for that purpose the notion of σ-symmetric matrices which extends that of symmetric matrices (the adjacency matrix of an undirected graph is a symmetric matrix). We extend the notion of rank-width for a similar notion of rank-width and of vertex-minor to σ-symmetric matrices. We will see that our definitions of rank-width and of vertex-minor coincide with Definitions 1.9 and 1.10 when we deal with undirected graphs. We will also extend Theorem 1.1 to σ-symmetric matrices. All the results in this chapter are technical but easy adaptations of the proofs of Oum [Oum05b, Sections 4,5] and can be seen as their generalizations as we will see in Sections 2.2 and 2.3. We will see in Chapter 3 that our notion of σ-symmetric matrices is useful for the extension of the notion of rank-width to directed graphs. Oum [Oum05c] also defined a notion of vertex-minor for symmetric and *skew-symmetric* matrices, however, this notion is incomparable to the one defined in this chapter.

In Section 2.1 we introduce the notion of σ-symmetric matrices and the notion of rank-width of σ-symmetric matrices. In Section 2.2 we introduce the notion of vertex-minor of σ-symmetric matrices derived from the works of Bouchet [Bou87] and we adapt some results of Oum, particularly [Oum05b, Proposition 4.3, Lemma 4.4]. In Section 2.3 we extend Theorem

1.1 to σ-symmetric matrices by adapting the proofs of [Oum05b, Section 5].

We will denote by $+$ and $\cdot$ the binary operations of any field and by 0 and 1 the neutral elements of $+$ and $\cdot$ respectively.

2.1   Rank-Width of σ-Symmetric Matrices

See Section 1.2 for the notions of layout and of branch-width of a symmetric function. We will now introduce the σ-*symmetric matrices* and the notion of F-*rank-width*.

Definition 2.1 (σ-Symmetric Matrices) Let F be a finite field and let $\sigma : F \to F$ be an automorphism. A V-matrix M over F for some finite set V is said σ-*symmetric* if $M_x^y = \sigma(M_y^x)$ for every $x, y \in V$.

Notice that σ-symmetric matrices are different from *skew-symmetric* matrices. We assume σ to be an automorphism, that means that $\sigma(0) = 0$ and $\sigma(1) = 1$ and in a skew-symmetric matrix M if $M_x^y = 1$, then $M_y^x = -1$ and there is no automorphism that maps 1 to -1. However, any symmetric matrix M is a σ-symmetric matrix since $M_x^y = M_y^x$ and we let σ be the identity automorphism. We let rk be the rank-function of matrices.

Definition 2.2 (Cut-Rank Function) Let F be a finite field and let $\sigma : F \to F$ be an automorphism. The *cut-rank function* of a σ-symmetric V-matrix M is the function $\rho_M^F : 2^V \to \mathbb{N}$ where for all $X \subseteq V$, we have $\rho_M^F(X) = rk(M_X^{\overline{X}})$.

Let A_G be the adjacency V_G-matrix of an undirected graph G. One can easily verify that the function ρ_G defined in Definition 1.9 is the same as $\rho_{A_G}^{GF(2)}$ and is then the cut-rank function of A_G over $GF(2)$. We now prove that ρ_M^F is symmetric and submodular. We first recall the *submodular inequality* of the matrix rank-function [Oum05b].

Proposition 2.1 *[Oum05b, Proposition 4.1] Let M be an (R, C)-matrix over a field F. Then for all $X_1, Y_1 \subseteq R$ and $X_2, Y_2 \subseteq C$, we have:*

$$rk\left(M_{X_1}^{X_2}\right) + rk\left(M_{Y_1}^{Y_2}\right) \geq rk\left(M_{X_1 \cup Y_1}^{X_2 \cap Y_2}\right) + rk\left(M_{X_1 \cap Y_1}^{X_2 \cup Y_2}\right).$$

Lemma 2.1 *Let F be a finite field and let $\sigma : F \to F$ be an automorphism. Then for every σ-symmetric matrix M, the function ρ_M^F is symmetric and submodular.*

Proof. Let M be a σ-symmetric V-matrix for some finite set V. The first statement is clear. By definition $\rho_M^F(X) = rk(M_X^{\overline{X}}) = rk(M_{\overline{X}}^X)$ for all $X \subseteq V$. Then one applies Proposition 2.1 to get the second statement.   ∎

We can now define the F-*rank-width of σ-symmetric matrices.*

Definition 2.3 (F-Rank-Width) Let F be a finite field and let $\sigma : F \to F$ be an automorphism. A *layout* of a σ-symmetric matrix M is a layout of ρ_M^F. The F-*rank-width of M*, denoted by $rwd^F(M)$, is the branch-width of the function ρ_M^F.

Since the rank-width of an undirected graph G is the branch-width of the function ρ_G and since $\rho_G = \rho_{A_G}^{GF(2)}$, Definition 2.3 coincides with Definition 1.9 when we deal with undirected graphs represented by their adjacency matrix over $GF(2)$.

For any σ-symmetric V-matrix M, we can construct an undirected graph $G(M)$ where $V_{G(M)} = V$ and there exists an edge between x and y in $V_{G(M)}$ if and only if $M_x^y \neq 0$. M is said *connected* if and only if $G(M)$ is connected. M_X^X is a *connected component* of M if $G(M)[X]$ is a connected component of $G(M)$ for $X \subseteq V$. It is then clear that the F-rank-width of M is the maximum of the F-rank-widths of its connected components.

2.2 Vertex-Minor

In order to extend the theory of *isotropic systems* [Bou88] to directed graphs, Bouchet [Bou87] generalized the notion of *local complementation* and of *locally equivalent* to directed graphs. He defined the local complementation at x of G as the directed graph represented by the matrix A_G' over $GF(2)$ where $(A_G')_z^y = (A_G)_z^y + (A_G)_z^x \cdot (A_G)_x^y$. When we deal with undirected graphs this definition coincides with the one of local complementation of undirected graphs (see Definition 1.10). The use of local complementation of directed graphs allows Bouchet [Bou87] to give in particular a cubic-time algorithm for finding a split[1] in a directed graph. Later Fon-Der-Flaass proved that two n-vertex graphs G and H are locally equivalent if and only if a system of n^2 equations with $3n$ indeterminates has a solution in $GF(2)$ [FDF96]. We extend this definition of local complementation to σ-symmetric matrices. We will use it in Chapter 3 in order to define a notion of vertex-minor for directed graphs.

Let p be a prime number. A field F has *characteristic* p if for every $a \in F$ we have $\underbrace{a + \cdots + a}_{p} = 0$ (see [LN97] for more informations on finite fields). A field of characteristic p is noted $GF(p^r)$ where $r \geq 1$ and p^r is its number of elements.

Definition 2.4 (Lc-Complementation) Let F be a finite field and let $\sigma : F \to F$ be an automorphism. Let M be a σ-symmetric V-matrix and let x be in V. The V-matrix obtained by applying an *lc-complementation at x to M* is $M * x$ where for all x_1, x_2 in V,

$$(M * x)_{x_1}^{x_2} = \begin{cases} M_{x_1}^{x_2} + M_{x_1}^x \cdot M_x^{x_2} & \text{if } x_1 \neq x_2 \text{ and } x \notin \{x_1, x_2\}, \\ M_{x_1}^{x_2} & \text{Otherwise.} \end{cases}$$

It is worth noticing that if $M_x^y = 0$, then the rows and the columns of M and of $M * x$ indexed by y are equal. Also the rows and columns of M and of $M * x$ indexed by x are equal. We say that N is *lc-equivalent* to M if N can be obtained by applying a sequence of lc-complementations to M. We call N a *vertex-minor* of M if $N = (M')_X^X$ where $X \subseteq V$ and M' is lc-equivalent to M.

One can easily verify from Definition 1.10 that if an undirected graph H is isomorphic to $G * x$ where G is an undirected graph and $x \in V_G$, then the following holds

$$(A_H)_{x_1}^{x_2} = \begin{cases} (A_G)_{x_1}^{x_2} + (A_G)_{x_1}^x \cdot (A_G)_x^{x_2} & \text{if } x_1 \neq x_2, \\ (A_G)_{x_1}^{x_2} & \text{Otherwise.} \end{cases}$$

---

[1]Let G be a directed graph. A bipartition (X, Y) of V_G is a *split* if and only if $|X|, |Y| \geq 2$ and $rk((A_G)_X^Y) = 1$ [Bou87].

Then an undirected graph H is a vertex-minor, as defined in Definition 1.10, of an undirected graph G if and only if A_H is a vertex-minor of A_G. We now prove some properties of vertex-minors of σ-symmetric matrices.

Lemma 2.2 *Let F be a finite field and let $\sigma : F \to F$ be an automorphism. Let M be a σ-symmetric V-matrix and let x be in V. Then*

1. *$M * x$ is a σ-symmetric V-matrix.*

2. *$M * x * \cdots * x = M$ if F is of characteristic p (we apply the local complementation at x p times consecutively).*

Proof. *1.* Let $N = M * x$. It is sufficient to prove that $N_{x_2}^{x_1} = \sigma(N_{x_1}^{x_2})$ for any $x_1, x_2 \in V$, $x_1 \neq x_2$.

$$
\begin{aligned}
N_{x_2}^{x_1} &= M_{x_2}^{x_1} + M_{x_2}^{x} \cdot M_{x}^{x_1} \\
&= \sigma(M_{x_1}^{x_2}) + \sigma(M_{x_1}^{x_2}) \cdot \sigma(M_{x_1}^{x}) \\
&= \sigma(M_{x_1}^{x_2}) + \sigma(M_{x_1}^{x} \cdot M_{x}^{x_2}) \\
&= \sigma(M_{x_1}^{x_2} + M_{x_1}^{x} \cdot M_{x}^{x_2}) \\
&= \sigma(N_{x_1}^{x_2}).
\end{aligned}
$$

2. For the sake of clarity we prove it when F is of characteristic 3. It is also sufficient to prove that $M_{x_1}^{x_2} = (M * x * x * x)_{x_1}^{x_2}$ for $x_1, x_2 \in V$, $x_1 \neq x_2$.

$$
\begin{aligned}
(M * x * x * x)_{x_1}^{x_2} &= (M * x * x)_{x_1}^{x_2} + (M * x * x)_{x_1}^{x} \cdot (M * x * x)_{x}^{x_2} \\
&= (M * x)_{x_1}^{x_2} + (M * x)_{x_1}^{x} \cdot (M * x)_{x}^{x_2} + (M * x)_{x_1}^{x} \cdot (M * x)_{x}^{x_2} \\
&= M_{x_1}^{x_2} + M_{x_1}^{x} \cdot M_{x}^{x_2} + M_{x_1}^{x} \cdot M_{x}^{x_2} + M_{x_1}^{x} \cdot M_{x}^{x_2} \\
&= M_{x_1}^{x_2}
\end{aligned}
$$

since $a + a + a = 0$ for all $a \in F$. ∎

Lemma 2.2 (1) proves that the lc-complementation is well-defined over σ-symmetric matrices.

Lemma 2.3 *Let F be a finite field and let $\sigma : F \to F$ be an automorphism. Let M be a σ-symmetric V-matrix for some finite set V and let x be in V. Then for every $X \subseteq V$,*

$$
\rho_M^F(X) = \rho_{M*x}^F(X).
$$

Proof. We can assume that $x \in X$ since the rank function rk is symmetric. For $y \in X$ the lc-complementation at x results in adding a multiple of the row indexed by x to the row indexed by y. This operation is repeated for all $y \in X$. In each case, the rank of the matrix does not change. Hence $\rho_M^F(X) = \rho_N^F(X)$. ∎

Proposition 2.2 *If N is lc-equivalent to M, then the F-rank-width of N is equal to the F-rank-width of M. If N is a vertex-minor of M, then the F-rank-width of N is at most the F-rank-width of M.*

Proof. The first statement is obvious by Lemma 2.3. Since taking sub-matrices does not increase the rank, it does not increase the F-rank-width. So the second statement is true. ■

It is clear that Lemma 1.1 is a corollary of Proposition 2.2. We refer to books like [Lip91] for terminologies on linear algebra. The following is an easy adaptation of [Oum05b, Proposition 4.3].

Proposition 2.3 *Let F be a finite field of characteristic 2 and let $\sigma : F \to F$ be an automorphism. Then for every σ-symmetric V-matrix M, every $x \in V$ and every $X \subseteq V - \{x\}$,*

$$\rho^F_{(M*x)\backslash x}(X) = rk \begin{pmatrix} 1 & M_x^{V-(X\cup x)} \\ M_X^x & M_X^{V-(X\cup x)} \end{pmatrix} - 1$$

Proof. Let M be a σ-symmetric V-matrix and let $x \in V$ and $X \subseteq V$. Let N be the set $\{y \mid M_x^y \neq 0\}$ and $Y = V - (X \cup x)$. We denote by $J = (M_{x_1}^x \cdot M_x^{x_2})_{x_1 \in X \cap N, x_2 \in Y \cap N}$. Then

$$\rho^F_{(M*x)\backslash x}(X) = rk((M*x)_X^Y)$$

$$= rk \begin{pmatrix} (M*x)_{X\cap N}^{Y\cap N} & (M*x)_{X\cap N}^{Y-N} \\ (M*x)_{X-N}^{Y\cap N} & (M*x)_{X-N}^{Y-N} \end{pmatrix}$$

$$= rk \begin{pmatrix} M_{X\cap N}^{Y\cap N} + J & M_{X\cap N}^{Y-N} \\ M_{X-N}^{Y\cap N} & M_{X-N}^{Y-N} \end{pmatrix}$$

$$= rk \begin{pmatrix} 1 & M_x^{Y\cap N} & M_x^{Y-N} \\ 0 & M_{X\cap N}^{Y\cap N} + J & M_{X\cap N}^{Y-N} \\ 0 & M_{X-N}^{Y\cap N} & M_{X-N}^{Y-N} \end{pmatrix} - 1$$

Recall that $M_x^y = 0$ for all $y \in Y - N$. Then, for each $x_1 \in X \cap N$:

$$M_{x_1}^x \cdot \begin{pmatrix} 1 & M_x^{Y\cap N} & M_x^{Y-N} \end{pmatrix} = \begin{pmatrix} M_{x_1}^x & J_{x_1}^{Y\cap N} & M_x^{Y-N} \end{pmatrix}.$$

Then

$$M_{x_1}^x \cdot \begin{pmatrix} 1 & M_x^{Y\cap N} & M_x^{Y-N} \end{pmatrix} + A_{x_1}^V = \begin{pmatrix} M_{x_1}^x & M_{x_1}^{Y\cap N} & M_{x_1}^{Y-N} \end{pmatrix}$$

where

$$A = \begin{pmatrix} 1 & M_x^{Y\cap N} & M_x^{Y-N} \\ 0 & M_{X\cap N}^{Y\cap N} + J & M_{X\cap N}^{Y-N} \\ 0 & M_{X-N}^{Y\cap N} & M_{X-N}^{Y-N} \end{pmatrix}$$

Then

$$rk(A) = rk \begin{pmatrix} 1 & M_x^{Y\cap N} & M_x^{Y-N} \\ M_{X\cap N}^x & M_{X\cap N}^{Y\cap N} & M_{X\cap N}^{Y-N} \\ 0 & M_{X-N}^{Y\cap N} & M_{X-N}^{Y-N} \end{pmatrix} - 1$$

$$= rk \begin{pmatrix} 1 & M_x^Y \\ M_X^x & M_X^Y \end{pmatrix} - 1.$$

Therefore $\rho^F_{(M*x)\backslash x}(X) = rk \begin{pmatrix} 1 & M^Y_x \\ M^x_X & M^Y_X \end{pmatrix} - 1$. ∎

The following lemma is thus the same as [Oum05b, Lemma 4.4], we just add it for completeness since the statement concerns σ-symmetric matrices and not only undirected graphs.

Lemma 2.4 *Let F be a finite field of characteristic 2 and let $\sigma : F \to F$ be an automorphism. Let M be a σ-symmetric V-matrix and let x be in V. Assume that (X_1, X_2) and (Y_1, Y_2) are partitions of $V - \{x\}$. Then*

$$\rho^F_{M\backslash x}(X_1) + \rho^F_{(M*x)\backslash x}(Y_1) \geq \rho^F_M(X_1 \cap Y_1) + \rho^F_M(X_2 \cap Y_2) - 1.$$

Proof. Let M' be defined such that $(M')^y_x = M^y_x$ if $x \neq y$ and $(M')^x_x = 1$. It is clear that for every $X \subseteq V$ we have $\rho^F_M(X) = \rho^F_{M'}(X)$. We have $Y_2 = V - (Y_1 \cup \{x\})$ and $X_2 = V - (X_1 \cup \{x\})$. By Proposition 2.3 we have :

$$\rho^F_{M\backslash x}(X_1) + \rho^F_{(M*x)\backslash x}(Y_1) = rk(M^{X_2}_{X_1}) + rk \begin{pmatrix} 1 & M^{Y_2}_x \\ M^x_{Y_1} & M^{Y_2}_{Y_1} \end{pmatrix} - 1$$

By definition we have $(M')^{Y_2 \cup x}_{Y_1 \cup x} = \begin{pmatrix} 1 & M^{Y_2}_x \\ M^x_{Y_1} & M^{Y_2}_{Y_1} \end{pmatrix}$. Then

$$\begin{aligned}
\rho^F_{M\backslash x}(X_1) + \rho^F_{(M*x)\backslash x}(Y_1) &= rk((M')^{X_2}_{X_1} + rk((M')^{Y_2 \cup x}_{Y_1 \cup x}) - 1 \\
&\geq rk((M')^{X_2 \cup Y_2 \cup x}_{X_1 \cap Y_1}) + rk((M')^{X_2 \cap Y_2}_{X_1 \cup Y_1 \cup x}) - 1 \quad \text{by Proposition 2.2} \\
&\geq \rho^F_{M'}(X_1 \cap Y_1) + \rho^F_{M'}(X_1 \cup Y_1 \cup x) - 1 \quad \text{by Definition of } \rho^F_{M'}
\end{aligned}$$

However, the cut-rank function $\rho^F_{M'}$ is symmetric (Lemma 2.1), i.e., $\rho^F_{M'}(X_1 \cup Y_1 \cup x) = \rho^F_{M'}(X_2 \cap Y_2)$. Then

$$\begin{aligned}
\rho^F_{M\backslash x}(X_1) + \rho^F_{(M*x)\backslash x}(Y_1) &\geq \rho^F_{M'}(X_1 \cap Y_1) + \rho^F_{M'}(X_2 \cap Y_2) - 1 \\
&\geq \rho^F_M(X_1 \cap Y_1) + \rho^F_M(X_2 \cap Y_2) - 1.
\end{aligned}$$

∎

2.3   Excluded Vertex-Minors

In this section we consider that we have fixed a **finite field F of characteristic 2 and an automorphism $\sigma : F \to F$**. We will extend Theorem 1.1 to σ-symmetric matrices over F. We will adapt the proofs by Oum [Oum05b, Section 5]. We first recall some definitions [GGRW03, Oum05b].

Let M be a σ-symmetric V-matrix and let (A, B) be a bipartition of V. A *branching of B* is a triple $(T, r, \mathcal{L})$ where T is a sub-cubic tree with a fixed node $r \in N^{(1)}_T$ and $\mathcal{L} : B \to N^{(1)}_T - \{r\}$ is a bijection. For an edge e of T and a node v of T, we let T_{ev} be the set of nodes in the component of $T \backslash e$ not containing v and we let $Y_{ev} = \mathcal{L}^{-1}(N^{(1)}_{T_{ev}})$. We say that B is *k-branched*

if there exists a branching $(T, r, \mathcal{L})$ such that for each edge e of T, we have $\rho_M^F(Y_{er}) \leq k$. It is worth noticing as in [Oum05b] that if A and B are k-branched, then the F-rank-width of M is at most k.

The following lemma is already proved in [Oum05b, Lemma 5.1] for $GF(2)$. But the proof is independent of the field. It uses the fact that the cut-rank function is symmetric, submodular and integer-valued. Since only the statement changes, we include it for completeness of the proof.

Lemma 2.5 *Let M be a σ-symmetric V-matrix of F-rank-width k. Let (A, B) be a bipartition of V such that $\rho_M^F(A) \leq k$. If there is no tripartition (A_1, A_2, A_3) of A such that $\rho_M^F(A_i) < \rho_M^F(A)$ for all $i \in [3]$, then B is k-branched.*

Proof. Assume for every tripartition (A_1, A_2, A_3) of A, we have $\rho_M^F(A_i) \geq \rho_M^F(A)$ for some $i \in [3]$.

Claim 2.1 *If (X_1, X_2) is a bipartition of V with $\rho_M^F(X_1) \leq k$, then either $\rho_M^F(B \cap X_1) \leq k$ or $\rho_M^F(B \cap X_2) \leq k$.*

Proof of Claim 2.1. Let $(A \cap X_1, A \cap X_2, \emptyset)$ be a tripartition of A. Then either $\rho_M^F(A \cap X_1) \geq \rho_M^F(A)$ or $\rho_M^F(A \cap X_2) \geq \rho_M^F(A)$; assume, $\rho_M^F(A \cap X_1) \geq \rho_M^F(A)$. By submodularity, $\rho_M^F(A \cup X_1) \leq \rho_M^F(A) + \rho_M^F(X_1) - \rho_M^F(A \cap X_1) \leq k$. So, $\rho_M^F(A \cup X_1) = \rho_M^F(B \cap X_2) \leq k$. ∎

We let $(T, \mathcal{L})$ be a layout of M of F-rank-width k. We may assume that $|V| \geq 3$ and $k > 0$, otherwise it is trivial.

Claim 2.2 *There exists a degree-3 node s of T such that for each edge e of T, $\rho_M^F(Y_{es} \cap B) \leq k$.*

Proof of Claim 2.2. We construct an orientation of T. Let e be an edge of T and let u and v be the ends of e. If $\rho_M^F(Y_{ev} \cap B) \leq k$, then we orient e from u to v. By Claim 2.1 each edge receives at least one orientation.

First assume that there exists a node v of T that is connected to every other node of T by a directed path in T. Since $k \geq 1$, each edge incident to a leaf has been oriented away from that leaf. Hence we may assume that v has degree 3. Then the claim follows with $s = v$.

Next, assume there exists no node reachable from every other node. Then there exists a pair of edges e and f and a node w on the path connecting e and f such that neither e nor f is oriented toward w. Let $Y_2 = V - (Y_{ew} \cup Y_{fw})$. Since e and f are oriented away from w, $\rho_M^F((Y_{fw} \cup Y_2) \cap B) \leq k$ and $\rho_M^F((Y_{ew} \cup Y_2) \cap B) \leq k$. By submodularity,

$$\rho_M^F(Y_{ew} \cap B) + \rho_M^F(Y_{fw} \cap B) \leq \rho_M^F((Y_{fw} \cup Y_2) \cap B) + \rho_M^F((Y_{ew} \cup Y_2) \cap B)$$
$$\leq 2k.$$

This contradicts the fact that neither e nor f is oriented toward w.   ∎

Let s be a node satisfying Claim 2.2, let e_1, e_2 and e_3 be the edges of T incident to s. Note that by assumption $\rho_M^F(Y_{e_i s} \cap A) \geq \rho_M^F(A)$ for some $i \in [3]$; assume $i = 1$. By sub-modularity,

$$
\begin{aligned}
\rho_M^F((Y_{e_2 s} \cup Y_{e_3 s}) \cap B) &= \rho_M^F(Y_{e_1 s} \cup A) \\
&\leq \rho_M^F(Y_{e_1 s}) + \rho_M^F(A) - \rho_M^F(Y_{e_1 s} \cap A) \\
&\leq \rho_M^F(Y_{e_1 s}) \leq k.
\end{aligned}
$$

Now we construct a branching $(T', r, \mathcal{L}')$ of B. Let T' be a tree obtained from the minimum subtree of T containing both e_1 and nodes in $\mathcal{L}(B)$ by subdividing e_1 with a node b, adding a node, denoted by r, of degree 1 adjacent to b (we delete all the degree-2 nodes). For each $x \in B$, we let $\mathcal{L}'(x)$ be the node of degree 1 of T' induced by $\mathcal{L}(x)$. Then $(T', r, \mathcal{L}')$ is a branching of B. By Claim 2.2 for each edge e of T', we have $\rho_M^F(Y_{er}) \leq k$. So B is k-branched. ∎

Let $g : \mathbb{N} \to \mathbb{N}$ be a function. A σ-symmetric V-matrix is called (m, g)-*connected* if for every partition (A, B) of V, $\rho_M^F(A) = \ell < m$ implies $|A| \leq g(\ell)$ or $|B| \leq g(\ell)$. This notion will help us to bound the order of the minimal σ-symmetric matrices that every σ-symmetric matrix of F-rank-width k must exclude as vertex-minors.

Lemma 2.6 *Let $f : \mathbb{N} \to \mathbb{N}$ be a non-decreasing function. Let M be a (m, f)-connected σ-symmetric V-matrix and let x be in V. Then either $M \backslash x$ or $(M * x) \backslash x$ is $(m, 2f)$-connected.*

Proof. We continue to follow the proof of [Oum05b, Lemma 5.2].

Suppose neither $M \backslash x$ nor $(M * x) \backslash x$ is $(m, 2f)$-connected. Then there are bipartitions (A_1, A_2) and (B_1, B_2) of $V - \{x\}$ such that

$$
\begin{aligned}
a &= \rho_{M \backslash x}^F(A_1) & |A_1| &> 2f(a) & |A_2| &> 2f(a) \\
b &= \rho_{(M * x) \backslash x}^F(B_1) & |B_1| &> 2f(b) & |B_2| &> 2f(b).
\end{aligned}
$$

We may assume that $a \geq b$ and that $|A_1 \cap B_1| > f(a)$. By Lemma 2.4 we have

$$
\rho_M^F(A_1 \cap B_1) + \rho_M^F(A_2 \cap B_2) \leq a + b + 1.
$$

Thus either $\rho_M^F(A_1 \cap B_1) \leq a$ or $\rho_M^F(A_2 \cap B_2) \leq b$. So by hypothesis either $|A_1 \cap B_1| \leq f(a)$ or $|A_2 \cap B_2| \leq f(b)$; suppose $|A_2 \cap B_2| \leq f(b)$. Similarly we also have either $|A_2 \cap B_1| \leq f(a)$ or $|A_1 \cap B_2| \leq f(b)$. Since $|A_1 \cap B_2| = |B_2| - |B_2 \cap A_2| > f(b)$ we have $|A_2 \cap B_1| \leq f(a)$. Then $|A_2| = |A_2 \cap B_1| + |A_2 \cap B_2| \leq f(a) + f(b) \leq 2f(a)$; a contradiction. ∎

We let $g(n) = (6^n - 1)/5$. Note that $g(0) = 0$, $g(1) = 1$ and $g(n) = 6g(n-1) + 1$ for all $n \geq 1$. We now prove that the minimal σ-symmetric matrices that have F-rank-width at least $k + 1$ are $(k + 1, g)$-connected.

Lemma 2.7 *Let $k \geq 1$ and let M be a σ-symmetric V-matrix for some finite set V. If M has F-rank-width larger than k but every proper vertex-minor of M has F-rank-width at most k, then M is $(k + 1, g)$-connected.*

Proof. We follow the proof of [Oum05b, Lemma 5.3]. We assume M connected since the F-rank-width of M is the maximum of the F-rank-width of its connected components. It is now easy to see that M is $(1, g)$-connected.

Suppose $m \leq k$ and M is (m, g)-connected but M is not $(m + 1, g)$-connected. Then there exists a partition (A, B) with $\rho_M^F(A) = m$ such that $|A| > g(m)$, $|B| > g(m)$. Also either A or B is not k-branched $(rwd^F(M) > k)$. We may assume that B is not k-branched. Let $x \in A$.

By Lemma 2.6, either $M\backslash x$ or $(M * x)\backslash x$ is $(m, 2g)$-connected; assume $M\backslash x$ is $(m, 2g)$-connected. Since $M\backslash x$ and $(M * x)\backslash x$ are proper vertex-minors of M, they both have F-rank-width at most k. Let (A_1, A_2, A_3) be a tripartition of $A - \{x\}$. Since $|A| > g(m) = 6g(m-1)+1$ there exists an $i \in [3]$ such that $|A_i| > 2g(m - 1)$. Since $M\backslash x$ is $(m, 2g)$-connected and $|A_i| > 2g(m - 1)$,

$$\rho_{M\backslash x}^F(A_i) \geq m \geq \rho_{M\backslash x}^F(A - \{x\}).$$

Therefore by Lemma 2.5, B is k-branched in $M\backslash x$. Since B is not k-branched in M there exists $W \subseteq B$ such that

$$\rho_M^F(W) = \rho_{M\backslash x}^F(W) + 1.$$

Thus, the column vectors of $M_W^{V-(W \cup x)}$ do not span M_W^x. So, the column vectors of $M_W^{V-(B \cup x)}$ do not span M_W^x. Hence, the column vectors of $M_B^{V-(B \cup x)}$ do not span M_B^x. Therefore,

$$\rho_{M\backslash x}^F(B) = \rho_M^F(B) - 1 = m - 1.$$

This implies that $|B| \leq 2g(m - 1)$ or $|A - \{x\}| \leq 2g(m - 1)$. A contradiction. ∎

We can now prove the main theorem of this chapter, which is a generalization of [Oum05b, Theorem 5.4]. Notice that, as in [GGRW03, Oum05b], Lemma 2.7 is the key lemma in the proof of the main theorem.

Theorem 2.1 (Excluded Vertex-Minors) *Let $k \geq 1$ and let M be a σ-symmetric V-matrix for some finite set V. If M has F-rank-width larger than k but every proper vertex-minor of M has F-rank-width at most k, then $|V| \leq (6^{k+1} - 1)/5$.*

Proof. Let $x \in V$. We may assume that $M\backslash x$ is $(k + 1, 2g)$-connected by Lemmas 2.6 and 2.7. Since $M\backslash x$ has F-rank-width k, there exists a biparitition (A, B) of $V - \{x\}$ such that $|A| \geq \frac{1}{3}(|V| - 1)$ and $|B| \geq \frac{1}{3}(|V| - 1)$ and $\rho_{M\backslash x}^F(A) \leq k$. By $(k + 1, 2g)$-connectivity, either $|A| \leq 2g(k)$ or $|B| \leq 2g(k)$. Therefore $|V| - 1 \leq 6g(k)$ and consequently $|V| \leq 6g(k) + 1 = g(k + 1)$. ∎

It is surprising that the bound $(6^{k+1} - 1)/5$ does not depend neither on F nor on σ, but that is because our proof technique is based on the branch-width of ρ_M^F and not on F or σ. However, the branch-width depends on F since there is no reason that the rank of a matrix is the same in two different fields. In the following corollary, which is a generalization of [Oum05b, Corollary 5.5], the set of forbidden σ-matrices as vertex-minors depends on F and σ.

Corollary 2.1 *Let F be a finite field of characteristic 2 and let $\sigma : F \to F$ be an automorphism. For each positive integer k, there is a finite list $\mathcal{C}_k^{(F,\sigma)}$ of σ-symmetric $(\ell \times \ell)$-matrices, $\ell \leq (6^{k+1} - 1)/5$ such that a σ-symmetric matrix M has F-rank-width at most k if and only if no σ-symmetric matrix in $\mathcal{C}_k^{(F,\sigma)}$ is isomorphic to a vertex-minor of M.*

Proof. We follow again the proof of [Oum05b, Corollary 5.5]. If $k < 0$ we let $\mathcal{C}_k = \emptyset$. If $k = 0$, we let $\mathcal{C}_0^{(F,\sigma)} = \{I_a \mid a \in F, a \neq 0\}$ where $I_a = \begin{pmatrix} 0 & a \\ a & 0 \end{pmatrix}$. It is clear that M has F-rank-width at most 0 if and only if M has no vertex-minor isomorphic to some $I_a \in \mathcal{C}_0^{(F,\sigma)}$. Let $k \geq 1$.

Let $\mathcal{C}_k^{(F,\sigma)}$ be the set of σ-symmetric matrices N such that $rwd^F(N) > k$ and every proper vertex-minor of N has F-rank-width at most k. By Theorem 2.1, $\mathcal{C}_k^{(F,\sigma)}$ is finite and each σ-symmetric matrix in $\mathcal{C}_k^{(F,\sigma)}$ has order $(\ell \times \ell)$ where $\ell \leq (6^{k+1} - 1)/5$.

Let M be a σ-symmetric matrix of F-rank-width at most k. Since every matrix in $\mathcal{C}_k^{(F,\sigma)}$ has F-rank-width larger than k, no matrix in $\mathcal{C}_k^{(F,\sigma)}$ is isomorphic to a vertex-minor of M.

Conversely, assume that the F-rank-width of M is larger than k and let N be a proper vertex-minor of M of minimum size such that $rwd^F(N) > k$. Then there exists a matrix $N' \in \mathcal{C}_k^{(F,\sigma)}$ isomorphic to N. ∎

We will give an upper bound on the size of $\mathcal{C}_k^{(F,\sigma)}$. Let $\mathcal{F}_\ell^{(F,\sigma)}$ be the set of σ-symmetric V-matrices over F where $|V| = \ell$. One can verify that:

$$|\mathcal{F}_\ell^{(F,\sigma)}| = |F|^{\ell-1} \times |F|^{\ell-2} \times \cdots \times |F|.$$

By Corollary 2.1, if a σ-symmetric matrix M is an excluded vertex-minor for σ-symmetric matrices of F-rank-width k, then the order of M is $\ell \times \ell$ where $\ell \leq (6^{k+1} - 1)/5$. Therefore, the size of $\mathcal{C}_k^{(F,\sigma)}$ is bounded by $\sum_{k+1 \leq j \leq \ell} |\mathcal{F}_j^{(F,\sigma)}|$ where $\ell = (6^{k+1} - 1)/5$. This upper bound is of no help for a concrete computation. Moreover, the bound $(6^{k+1} - 1)/5$ seems to be far from optimal. But, we were not able to improve it.

2.4   Conclusion

We have defined a notion of rank-width for σ-symmetric matrices over a field F and a notion of vertex-minor for σ-symmetric matrices over F. We have generalized Theorem 1.1 to σ-symmetric matrices over a finite field of characteristic 2. There are two open questions:

(Q2.1) Generalize Theorem 2.1 to σ-symmetric matrices over all finite fields.

(Q2.2) Give an algorithm to recognize if a σ-symmetric matrix is a vertex-minor of another σ-symmetric matrix.

   We will use the results of this chapter to define a notion of rank-width for directed graphs in Chapter 3.

Chapter 3

Rank-Width of Directed Graphs

There are several ways to define the notion of rank-width for directed graphs. We define two possible notions: one based on σ-symmetric matrices, called $GF(4)$-*rank-width*, and another based on a coding of directed graphs by undirected graphs, called *bi-rank-width*. We compare the two definitions in this chapter. In Section 3.1 we introduce the two notions of rank-width for directed graphs and define for each of them a notion of *vertex-minor*. We prove in Section 3.2 that the two notions are equivalent, i.e., that a class of directed graphs has bounded $GF(4)$-rank-width if and only if it has bounded bi-rank-width.

3.1   Rank-Width of Directed Graphs

3.1.1 $GF(4)$-Rank-Width

We recall that $GF(4)$ has four elements $\{0, 1, a, a^2\}$ with the property that $1 + a + a^2 = 0$ and $a^3 = 1$ and is of characteristic 2.

Definition 3.1 ($GF(4)$-Rank-Width) For a directed graph G, we denote by F_G the V_G-matrix over $GF(4)$ where:

$$(F_G)_x^y = \begin{cases} 0 & \text{iff } (x, y) \notin E_G \text{ and } (y, x) \notin E_G \\ a & \text{iff } (x, y) \in E_G \text{ and } (y, x) \notin E_G \\ a^2 & \text{iff } (y, x) \in E_G \text{ and } (x, y) \notin E_G \\ 1 & \text{iff } (x, y) \in E_G \text{ and } (y, x) \in E_G \end{cases}$$

We let $\sigma : GF(4) \to GF(4)$ be an automorphism such that $\sigma(1) = 1$, $\sigma(0) = 0$, $\sigma(a) = a^2$ and $\sigma(a^2) = a$. It is then clear that F_G is a σ-symmetric V_G-matrix. The $GF(4)$-rank-width of G, denoted by $rwd^{(4)}(G)$, is the $GF(4)$-rank-width of F_G.

Remark 3.1 Let G be an undirected graph. We denote by $\overrightarrow{G}$ the directed graph obtained from G by replacing each edge by two opposite edges. By the definition of $\overrightarrow{G}$, we have $A_G = F_{\overrightarrow{G}}$. Then $rwd^{(4)}(\overrightarrow{G}) = rwd(G)$ (see Lemma 3.2).

Example 3.1 We consider the directed graph G on Figure 4 (iii) (Chapter 1, Section 1.1). The V_G-matrix over $GF(4)$ of G is:

$$F_G = \begin{array}{c|cccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
\hline
x_1 & 0 & a & a & a^2 & a & 0 \\
x_2 & a^2 & 0 & 0 & 0 & a & a \\
x_3 & a^2 & 0 & 0 & a & 0 & a \\
x_4 & a & 0 & a^2 & 0 & a & 0 \\
x_5 & a^2 & a^2 & 0 & a^2 & 0 & 0 \\
x_6 & 0 & a^2 & a^2 & 0 & 0 & 0 \\
\end{array}$$

Figure 8 shows a layout of the function $\rho_G^{GF(4)}$ of branch-width 2. One can verify that for every pair (z,t) of vertices in G, we have $\rho_G^{GF(4)}(\{z,t\}) = 2$. Then the $GF(4)$-rank-width of G is 2.
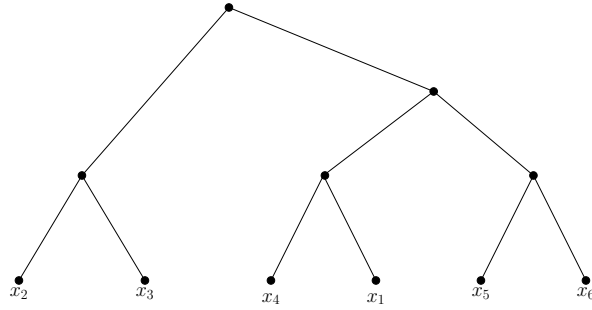
Figure 8: A layout of $\rho_G^{GF(4)}$ where G is the directed graph on Figure 4 (iii).

Let G be a directed graph and let F_G be the V_G-matrix that represents its adjacencies. Then in any layout $(T, \mathcal{L})$ of $\rho_{F_G}^{GF(4)}$, the vertices of G are in bijection with the nodes of degree 1 in T. Then a layout of $\rho_{F_G}^{GF(4)}$ measures how some bipartitions of V_G are connected by using the V_G-matrix F_G. We will now define our first notion of *vertex-minor* for directed graphs by using Definition 2.4.

Let G be a directed graph and let x be a vertex of G. An *lc-complementation of G at x* is the graph represented by the V_G-matrix $F_G * x$, noted $G * x$. We say that a directed graph H is *lc-equivalent* to a directed graph G if H can be obtained from G by a sequence of lc-complementations and H is a *vertex-minor* of G if H can be obtained from G by a sequence of lc-complementations and of vertex-deletions. Thus, a directed graph H is a vertex-minor of a directed graph G if and only if F_H is a vertex-minor of F_G. One can verify, using the definition of lc-complementation, that if $H = G * x$, then H is obtained from G by modifying the sub-graph induced on the neighbors of x as shown by Table 1[1].

Example 3.2 We consider the directed graph G on Figure 4 (iii). Figure 9 shows the directed graph obtained by applying an lc-complementation at x_4 of G. One can verify that the layout on Figure 8 of G is also a layout of $G * x_4$ and of same branch-width.

---

[1] $x \to y$ means (x,y) is an arc and not (y,x); $x \leftarrow y$ means (y,x) is an arc and not (x,y); $x \leftrightarrow y$ means (x,y) and (y,x) are arcs and $x_1 \perp x_2$ means no arc between x_1 and x_2.

G	$G * x$
$x_1 \perp x_2$	$x_1 \leftrightarrow x_2$
$x_1 \rightarrow x_2$	$x_1 \leftarrow x_2$
$x_1 \leftarrow x_2$	$x_1 \rightarrow x_2$
$x_1 \leftrightarrow x_2$	$x_1 \perp x_2$

(a)

G	$G * x$
$x_1 \perp x_2$	$x_1 \rightarrow x_2$
$x_1 \rightarrow x_2$	$x_1 \perp x_2$
$x_1 \leftarrow x_2$	$x_1 \leftrightarrow x_2$
$x_1 \leftrightarrow x_2$	$x_1 \leftarrow x_2$

(b)

Table 1: (a) Uniform Case: $x_1 \leftarrow x \rightarrow x_2$ or $x_1 \rightarrow x \leftarrow x_2$ or $x_1 \leftrightarrow x \leftrightarrow x_2$. (b) Non Uniform Case: $x_1 \leftarrow x \leftarrow x_2$ or $x_1 \rightarrow x \leftrightarrow x_2$ or $x_1 \leftrightarrow x \rightarrow x_2$.
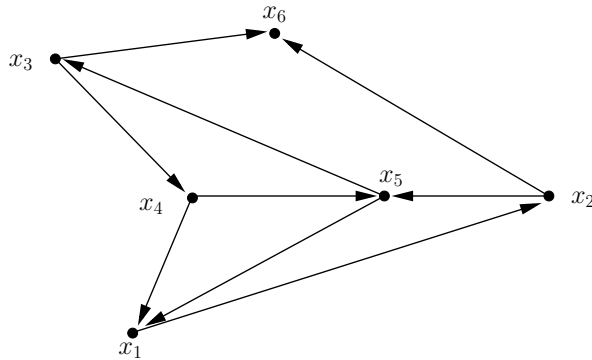
Figure 9: The directed graph $G * x_4$ where G is the directed graph on Figure 4 (iii).

Note that when G is an undirected graph, then when applying an lc-complementation of $\overrightarrow{G}$ at $x \in V_G$, the sub-graph of $\overrightarrow{G}$ induced on the neighbors of x is edge-complemented, i.e., if $z \leftrightarrow y \in V_{\overrightarrow{G}}$, then $z \perp y$ holds in $\overrightarrow{G} * x$ and vice-versa (see Table 1). Therefore, $\overrightarrow{G} * x = \overrightarrow{G * x}$.

We get this theorem as a consequence of Corollary 2.1.

Theorem 3.1 *For each k, there is a finite list $\mathcal{C}_k$ of directed graphs having at most $(6^{k+1}-1)/5$ vertices such that a directed graph G has $GF(4)$-rank-width at most k if and only if no directed graph in $\mathcal{C}_k$ is isomorphic to a vertex-minor of G.*

In a graph G, directed or not, a vertex of G is a *pendant* vertex if it has only one neighbor. A *distance hereditary graph* is an undirected graph with a single vertex or that can be obtained from a distance hereditary graph by adding an isolated vertex or a pendant vertex or by creating *twins*[2]. Oum proved that an undirected graph has rank-width at most 1 if and only if it is a distance hereditary graph [Oum05b]. We will give a similar characterization for graph classes of $GF(4)$-rank-width at most 1. Let us first give some definitions.

Definition 3.2 (Oriented Distance Hereditary Graphs) Let G be a directed graph. Two vertices x and y of G are called *dtwins* if $\rho_G^{GF(4)}(\{x,y\}) \leq 1$. A directed graph is called an *oriented distance hereditary graph* if and only if it is a directed graph with a single vertex or it can be obtained by creating dtiwns, adding an isolated vertex or adding a pendant vertex to an oriented distance hereditary graph.

---

[2]In an undirected graph G two vertices x and y are twins if for all $z \in V_G - \{x,y\}$, $E_G(x,z)$ holds if and only if $E_G(y,z)$ holds.

Example 3.3 Figure 10 shows an oriented distance hereditary graph G and a layout of $\rho_G^{GF(4)}$ of branch-width 1. The following pairs of vertices are dtiwns: $(x_4, x_3), (x_4, x_6), (x, 4, x_7)$ and (x_2, x_8).
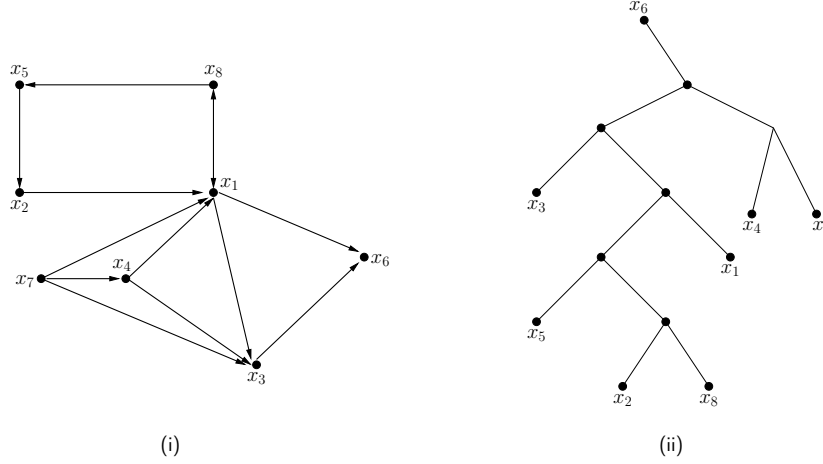
Figure 10: (i) An oriented distance hereditary graph G. (ii) A layout of the function $\rho_G^{GF(4)}$ of branch-width 1.

We will prove that directed graphs of $GF(4)$-rank-width at most 1 are exactly oriented distance hereditary graphs. Notice that if G is an oriented distance hereditary graph, the graph obtained from G by forgetting the orientations of the arcs is a distance hereditary graph and this motivates the terminology *oriented distance hereditary*. However, we do not have any characterization of oriented distance hereditary graphs in terms of *distance*, whereas distance hereditary graphs have one. We will follow the same ideas as in [Oum05b, Section 7].

Proposition 3.1 *Let G be a directed graph and let x and y be dtwins such that $G \backslash x$ has at least one arc. Then $rwd^{(4)}(G \backslash x) = rwd^{(4)}(G)$*

Proof. By definition of vertex-minor we have $rwd^{(4)}(G \backslash x) \leq rwd^{(4)}(G)$. We will prove that $rwd^{(4)}(G \backslash x) \geq rwd^{(4)}(G)$. Let $(T, \mathcal{L})$ be a layout of branch-width $k = rwd^{(4)}(G \backslash x)$ of the function $\rho_{G \backslash x}^{GF(4)}$. By definition of layouts of $\rho_{G \backslash x}^{GF(4)}$ there is a bijection $\mathcal{L}$ between $V_{G \backslash x}$ and $N_T^{(1)}$. Let $v = \mathcal{L}(y)$ and let $u \in V_T$ such that $uv \in E_T$. Let T' be obtained from T as follows:

$$V_{T'} = V_T \cup \{u', w \mid u', w \notin V_T\},$$
$$E_{T'} = (E_T - \{uv\}) \cup \{uu', u'v, u'w\}.$$

We let $\mathcal{L}' : V_G \to N_{T'}^{(1)}$ such that:

$$\mathcal{L}'(z) = \begin{cases} \mathcal{L}(z) & \text{if } z \in V_G - \{x\}, \\ w & \text{otherwise.} \end{cases}$$

It is clear that $(T', \mathcal{L}')$ is a layout of $\rho_G^{GF(4)}$. We claim that $bwd(\rho_G^{GF(4)}, T', \mathcal{L}') = bwd(\rho_{G \backslash x}^{GF(4)}, T, \mathcal{L})$.

It is clear that the branch-width of the edges $u'v$ and $u'w$ are at most 1. Since x and y are dtwins, the branch-width of the edge uu' is at most 1. Moreover, the other edges of T' are in T, then their branch-width in $(T', \mathcal{L}')$ is equal to their branch-width in $(T, \mathcal{L})$. Since $G\backslash x$ has at least one arc, we have $rwd^{(4)}(G\backslash x) \geq 1$, i.e., $bwd(\rho_G^{GF(4)}, T', \mathcal{L}') = rwd^{(4)}(G\backslash x)$. Therefore, $rwd^{(4)}(G\backslash x) \geq rwd^{(4)}(G)$. ∎

Proposition 3.2 *Let G be a directed graph such that $rwd^{(4)}(G) = 1$. Then there exist x and y such that x and y are dtwins or x is the only neighbor of y.*

Proof. Assume that $|V_G| \geq 3$, otherwise the proposition is trivially true. Let $(T, \mathcal{L})$ be a layout of the function $\rho_G^{GF(4)}$ of branch-width 1. There exists at least one node u of T adjacent with two nodes in $N_T^{(1)}$, say v and w. We let x and y such that $\mathcal{L}(x) = v$ and $\mathcal{L}(y) = w$. Let u' be the node adjacent with u and different from v and w. The partition induced by $T\backslash uu'$ is $(\{x, y\}, V_G - \{x, y\})$. Since $rwd^{(4)}(G) = 1$, the branch-width of the edge uu' is at most 1. This means that either x and y are dtwins, or y is the only neighbor of x or x is the only neighbor of y. ∎

Proposition 3.3 *A directed graph G has $GF(4)$-rank-width at most 1 if and only if G is an oriented distance hereditary graph.*

Proof. By Proposition 3.1 and Definition 3.2, an oriented distance hereditary graph has $GF(4)$-rank-width at most 1. Conversely by Proposition 3.2, if a directed graph has $GF(4)$-rank-width 1, it is an oriented distance hereditary graph. ∎

3.1.2   Bi-Rank-Width

We will now define our second notion of rank-width for directed graphs, named *bi-rank-width* and based on matrices over $GF(2)$. Let G be a directed graph and x a vertex of G. We first observe that the neighbors of each vertex x can be grouped into two groups: the set of vertices $y \in V_G$ such that $(x, y) \in E_G$ and the set of vertices $z \in V_G$ such that $(z, x) \in E_G$. We will use this observation in order to define the *bi-rank-width* of directed graphs based on matrices over $GF(2)$.

Definition 3.3 (Bi-Rank-Width) Let G be a directed graph. We let A_G be the V_G-matrix over $GF(2)$ where $(A_G)_x^y = 1$ if and only if $(x, y) \in E_G$.

For every two disjoint subsets X and Y of V_G, we let $(A_G^+)_X^Y = (A_G)_X^Y$ and $(A_G^-)_X^Y = ((A_G)_Y^X)^T$. For every $X \subseteq V_G$, we let $\rho_G^{(bi)}(X) = rk\left((A_G^+)_X^{\overline{X}}\right) + rk\left((A_G^-)_X^{\overline{X}}\right)$. It is straightforward to verify that $\rho_G^{(bi)}$ is symmetric and submodular.

The *bi-rank-width* of a directed graph G, denoted by $brwd(G)$, is defined as the branch-width of the function $\rho_G^{(bi)}$.

Example 3.4 We still let G be the directed graph on Figure 4 (iii). The following table shows the adjacency V_G-matrix A_G of G.

$$A_G = \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 0 & 1 & 1 & 0 & 1 & 0 \\ x_2 & 0 & 0 & 0 & 0 & 1 & 1 \\ x_3 & 0 & 0 & 0 & 1 & 0 & 1 \\ x_4 & 1 & 0 & 0 & 0 & 1 & 0 \\ x_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

We now illustrate matrices of the form $(A_G^+)_X^Y$ and $(A_G^-)_X^Y$. Let $X = \{x_1, x_2, x_4\}$ and $Y = \{x_3, x_4, x_6\}$. We have:

$$(A_G^+)_X^Y = \begin{array}{c|ccc} & x_3 & x_5 & x_6 \\ \hline x_1 & 1 & 1 & 0 \\ x_2 & 0 & 1 & 1 \\ x_4 & 0 & 1 & 0 \end{array}$$

$$(A_G^-)_X^Y = \begin{array}{c|ccc} & x_3 & x_5 & x_6 \\ \hline x_1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_4 & 1 & 0 & 0 \end{array}$$

Hence $\rho_G^{(bi)}(X) = 4$. One can verify that the bi-rank-width of G is 3 and the layout of $\rho_G^{GF(4)}$ on Figure 8 is also a layout of $\rho_G^{(bi)}$ of branch-width 3.

Remark 3.2 Let G be an undirected graph. We denote by $\overrightarrow{G}$ the directed graph obtained from G by replacing each edge by two opposite edges. By the definition of $\overrightarrow{G}$, for every two disjoint subsets X and Y of G, we have $(A_G^+)_X^Y = (A_G^-)_X^Y = (A_G)_X^Y$. Therefore, $brwd(\overrightarrow{G}) = 2 \cdot rwd(G)$.

We now define a notion of vertex-minor inclusion having a "good behavior" for bi-rank-width.

Definition 3.4 ([FDF96]) The *local complementation of a directed graph G* at its vertex x is the directed graph G' represented by the V_G-matrix $A_{G'}$ where:

$$(A_{G'})_{x_1}^{x_2} = \begin{cases} (A_G)_{x_1}^{x_2} + (A_G)_{x_1}^{x} \cdot (A_G)_x^{x_2} & \text{if } x_1 \neq x_2, \\ (A_G)_{x_1}^{x_2} & \text{Otherwise.} \end{cases}$$

A directed graph H is *b-locally equivalent* to a directed graph G if H can be obtained from G by a sequence of local complementations. A directed graph H is a *bvertex-minor* of a directed graph G if H can be obtained from G by a sequence of vertex deletions and local complementations.

Let H be obtained from a directed graph G by applying a local complementation at x. Then only the sub-graph induced on the neighbors of x is modified. Let x_1 and x_2 be two neighbors of x. One can verify that $(x_1, x_2) \in E_H$ if and only if $(x_1, x_2) \notin E_G$ and, $(x_1, x) \in E_G$
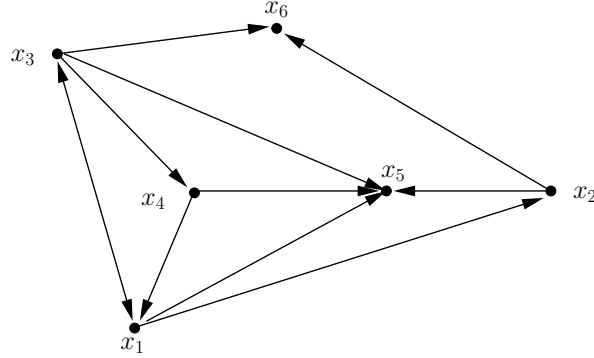
Figure 11: A directed graph obtained by applying a local complementation at x_4 on G, the directed graph on Figure 4 (iii).
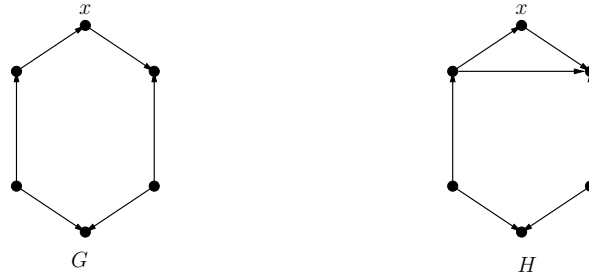
Figure 12: H is obtained from G by applying a local complementation at x and similarly G is obtained from H by applying a local complementation at x.

and $(x, x_2) \in E_G$ or $(x_1, x_2) \in E_G$ and, $(x_1, x) \notin E_G$ or $(x, x_2) \notin E_G$. It is also clear that the local complementation at x of H is G. Figures 11 and 12 illustrate local complementation of directed graphs.

We have a lemma similar to Proposition 2.2 and to Lemma 1.1.

Lemma 3.1 *If H is b-locally equivalent to G, then the bi-rank-width of H is equal to the bi-rank-width of H. If H is a bvertex-minor of G, then the bi-rank-width of H is at most the bi-rank-width of G.*

Proof. Fon-Der-Flaass [FDF96] proved that $rk((A_G^+)_X^Y)$ is invariant with respect to local complementation. Since $(A_G^-)_X^Y = (A_G^+)_Y^X$, the bi-rank-width is invariant with respect to local complementation. Since taking sub-matrices does not increase the rank, it does not increase the bi-rank-width. ∎

Remark 3.3 It is open whether for every positive integer k there exists a finite list $\mathcal{C}_k$ of directed graphs such that a directed graph has bi-rank-width at most k if and only if it does not contain any directed graph in $\mathcal{C}_k$ as a bvertex-minor. We can notice that graphs of bi-rank-width at most k are not well-quasi-ordered by bvertex-minor inclusion. In fact the class $\mathcal{F}$ of directed even cycles such that each vertex has either in-degree 2 or out-degree 2, are of bounded bi-rank-width and are

not well-quasi-ordered by bvertex-minor inclusion since none of them is a bvertex-minor of another. Figure 13 illustrates such cycles.

It is also worth noticing that a directed graph H can be a vertex-minor of a directed graph G without being a bvertex-minor of G and vice-versa as shown by Figures 13 and 14 or, Figures 9 and 11. It is open whether there exists a notion of vertex-minor inclusion on directed graphs, independent of the adjacency representations of directed graphs, that could be called *vminor*, and such that if a directed graph H is a vminor of G, then the bi-rank-width (resp. $GF(4)$-rank-width) of H is at most the bi-rank-width (resp. $GF(4)$-rank-width) of G.
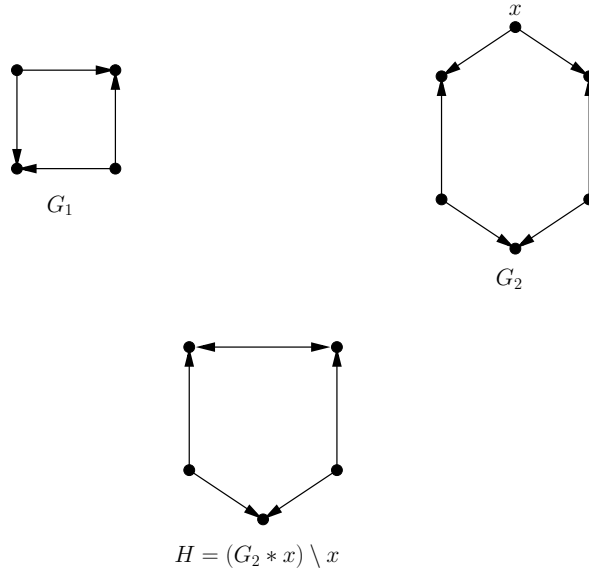
Figure 13: G_1 and G_2 are graphs in $\mathcal{F}$. H is a vertex-minor of G_2 and cannot be a bvertex-minor of G_2. In fact for any $x \in V_{G_2}$, the local complementation at x of G_2 does not change G_2.
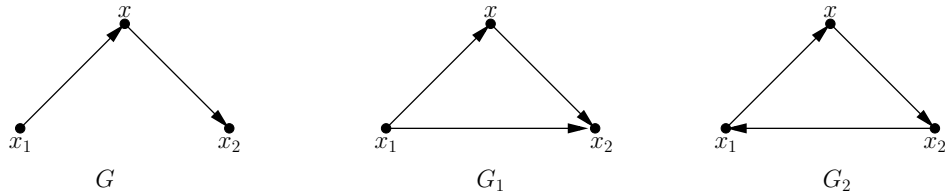
Figure 14: G_1 is a bvertex-minor of G and G_2 is a vertex-minor of G. G_1 is not isomorphic to G_2 and one can verify that we can not get G_2 as a bvertex-minor of G_1 and vice-versa.

3.2 $GF(4)$-Rank-Width and Bi-Rank-Width are Equivalent

We now prove that bi-rank-width and $GF(4)$-rank-width are equivalent complexity measures. In Chapter 5 we will prove that they are equivalent to clique-width.

Proposition 3.4 *For every directed graph G, we have*

$$rwd^{(4)}(G) \le brwd(G) \le 4 \cdot rwd^{(4)}(G).$$

Before proving the proposition, we recall some technical properties about ranks of matrices, particularly ranks of matrices with coefficients in $\{0,1\}$ over the fields $GF(2)$ and $GF(4)$.

Lemma 3.2   *(i) Let M be a matrix with entries in $\{0,1\}$. If the rank of M over $GF(2)$ is k, then the rank of M over $GF(4)$ is k.*

   (ii) if A and B are two matrices over a field F, then $rk(A + B) \le rk(A) + rk(B)$. Let $a \in F$, $a \ne 0$, then $rk(a \cdot A) = rk(A)$.

Proof. *(i)* is immediate since $GF(4)$ is an extension of $GF(2)$.

(ii) The column-bases of A and of B generate all the column-vectors of $A + B$. See for instance [Lip91]. ∎

Lemma 3.3 *For each bipartition $(X, \overline{X})$ of V_G we have*

$$(F_G)_X^{\overline{X}} = a \cdot (A_G^+)_X^{\overline{X}} + a^2 \cdot (A_G^-)_X^{\overline{X}}.$$

Proof. Consequence of the definitions of F_G, A_G^+ and A_G^-. ∎

Proof of Proposition 3.4. We first prove that $rwd^{(4)}(G) \le brwd(G)$. Assume that $brwd(G) = k$ and let $(T, \mathcal{L})$ be a layout of branch-width k of $\rho_G^{(bi)}$. We claim that $(T, \mathcal{L})$ is also a layout of branch-width at most k of $\rho_{F_G}^{GF(4)}$.

It is sufficient to prove, for each edge e of T, that $\rho_{F_G}^{GF(4)}(X^e) \le \rho_G^{(bi)}(X^e)$ in order to prove that the $GF(4)$-rank-width of $(T, \mathcal{L})$ is at most k.

By Lemmas 3.2 and 3.3, for each edge e of T, we have

$$rk\left((F_G)_{X^e}^{\overline{X^e}}\right) \le rk\left((A_G^+)_{X^e}^{\overline{X^e}}\right) + rk\left((A_G^-)_{X^e}^{\overline{X^e}}\right)$$
$$\le \rho_G^{(bi)}(X^e).$$

We can then conclude that $\rho_{F_G}^{GF(4)}(X^e) \le \rho_G^{(bi)}(X^e)$.

We now prove that $brwd(G) \le 4 \cdot rwd^{(4)}(G)$. Assume that $rwd^{(4)}(G) = k$ and let $(T, \mathcal{L})$ be a layout of branch-width k of $\rho_{F_G}^{GF(4)}$. We claim that $(T, \mathcal{L})$ is also a layout of branch-width $4k$ of $\rho_G^{(bi)}$. Let e be an edge of T and let $M_1 = (A_G^+)_{X^e}^{\overline{X^e}}$ and $M_2 = (A_G^-)_{X^e}^{\overline{X^e}}$. Let $\pi_1, \pi_2, \pi_3 : GF(4) \to GF(2)$ be such that:

$$\pi_1(b) = \begin{cases} 0 & \text{if } b \in \{0, a^2\} \\ 1 & \text{if } b = \{1, a\} \end{cases}$$

$$\pi_2(b) = \begin{cases} 0 & \text{if } b \in \{0, a\} \\ 1 & \text{if } b = \{1, a^2\} \end{cases}$$

$$\pi_3(b) = \begin{cases} 0 & \text{if } b \in \{0, 1\} \\ 1 & \text{if } b = \{a, a^2\} \end{cases}$$

It is clear that $M_j = \pi_j\left((F_G)_{Xe}^{\overline{Xe}}\right)$ for $j = 1, 2$. It is easy to see that each π_j, $j = 1, 2, 3$ is an homomorphism with respect to $+$. Moreover, for every b and c in $GF(4)$, we have

$$\pi_1(b \cdot c) = \pi_1(b) \cdot \pi_1(c) + \pi_3(b) \cdot \pi_3(c)$$
$$\pi_2(b \cdot c) = \pi_2(b) \cdot \pi_2(c) + \pi_3(b) \cdot \pi_3(c)$$
$$\pi_3(b \cdot c) = \pi_1(b) \cdot \pi_1(c) + \pi_2(b) \cdot \pi_2(c)$$

We let $v_1, \ldots, v_k$ be the column-bases of $(F_G)_{Xe}^{\overline{Xe}}$. Then for each column-vector v we have:

$$v = \sum_{i \le k} \alpha_i \cdot v_i$$

and for $j = 1, 2$ we have

$$\begin{aligned} \pi_j(v) &= \sum_{i \le k} \pi_j(\alpha_i \cdot v_i) \\ &= \sum_{i \le k} (\pi_j(\alpha_i) \cdot \pi_j(v_i) + \pi_3(\alpha_i) \cdot \pi_3(v_i)). \end{aligned}$$

Thus every column-vector of M_j is a linear combination of $2k$ vectors, i.e., $rk(M_j) \le 2k$. Therefore, $brwd(G) \le 4 \cdot rwd^{(4)}(G)$.   ∎

3.3   Conclusion

We have defined two notions of rank-width for directed graphs, named $GF(4)$-*rank-width* and *bi-rank-width*. $GF(4)$-rank-width is defined by using a coding of the adjacencies of directed graphs by σ-symmetric matrices over $GF(4)$, while bi-rank-width is defined by using a coding of directed graphs by two undirected graphs. We have defined two notions of vertex-minor inclusion for directed graphs, one related to $GF(4)$-rank-width and named *vertex-minor*, and another related to bi-rank-width, called *bvertex-minor*. While graphs of $GF(4)$-rank-width are characterized by finitely excluded vertex-minors (by using results of Chapter 2), we have no such result for bi-rank-width. We finish this chapter with some questions:

(Q3.1) Fon-Der-Flaass proved that two n-vertex graphs G and H are b-locally equivalent if and only if a system of n^2 equations with $3n$ indeterminates has a solution in $GF(2)$. Can we find a similar characterization for lc-equivalent directed graphs?

(Q3.2) Does there exist a polynomial-time algorithm to check if a directed graph is a vertex-minor (resp. bvertex-minor) of another directed graph?

(Q3.3) Can we find a CMS-characterization of the notion of vertex-minor of directed graphs? Notice that for undirected graphs Courcelle and Oum [CO07] gave a C_2MS-characterization of vertex-minor. One might think of a C_4MS-characterization in the case of directed graphs.

The analogous of question (Q3.2) for vertex-minor of undirected graphs is still open. However, Courcelle and Oum proved that it is decidable in polynomial-time when restricted to undirected graphs of bounded rank-width, by using the C_2MS-characterization of vertex-minor of undirected graphs [CO07]. In our case if we have an answer to (Q3.3) we will also be able to give an answer for (Q3.2) when restricted to classes of directed graphs of bounded $GF(4)$-rank-width by using [CMR00].

We will define in Chapter 4 some graph operations that handle algebraically graphs of small bi-rank-width (resp. $GF(4)$-rank-width). A specialization of these operations will give an exact algebraic characterization of undirected graphs of rank-width at most k as proved in [CK09].

We will prove in Chapter 5 that a class of directed graphs has bounded $GF(4)$-rank-width (resp. bi-rank-width) if and only if it has bounded clique-width and derive from the results of [HO07] that for fixed k there exists a cubic-time algorithm that given a directed graph either outputs that the bi-rank-width (resp. $GF(4)$-rank-width) is larger than k or outputs a layout of branch-width k of the function $\rho_G^{(bi)}$ (resp. $\rho_{F_G}^{GF(4)}$).

Chapter 4

Algebraic Characterization of Rank-Width

A class of undirected graphs has bounded clique-width if and only if it has bounded rank-width[1] [OS06]. However, clique-width has the advantage of being defined in terms of graph operations and the algorithmic results on clique-width are based on these operations [CMR00, CMR01, CV03]. In order to solve MS problems on an undirected graph G of small rank-width, one may transform an optimal layout of $\rho_G^{GF(2)}$ into a clique-width expression. This approach is based on results by Oum and Seymour [OS06]. However, we prove in this chapter that there is no need to transform an optimal layout of $\rho_G^{GF(2)}$ into a clique-width expression in order to solve MS problems on a graph of small rank-width because we give algebraic expressions that characterize rank-width of undirected graphs and fall into the framework of Section 1.4. We prove that an undirected graph has rank-width at most k if and only if it is the value of a term in $T(R_k, C_k)$ where R_k is a set of algebraic graph operations and C_k is a set of constants, both depending on k. We give here a more general proof in order to propose graph operations that handle algebraically classes of graphs of bounded $GF(4)$-rank-width and bi-rank-width. We first introduce the notion of *F-rank-width* for certain edge-colored graph classes where F is a finite field.

Definition 4.1 (F-Rank-Width of Graphs) Let F be a finite field and let $\sigma : F \to F$ be an automorphism. An edge-colored graph G is *over* (F, σ) if its edges are colored by elements in F and if for every pair of vertices (x, y) and every a in F, $E_G^a(x, y)$ holds if and only if $E_G^{\sigma(a)}(y, x)$ holds and for every pair of vertices (x, y) there exists a unique a in F such that $E_G^a(x, y)$ holds, hence $E_G^{\sigma(a)}(y, x)$ holds. We can then define an *adjacency matrix* of an edge-colored graph over (F, σ) as the V_G-matrix F_G over F where for every pair of vertices (x, y) we have $(F_G)_x^y = a$ if and only if $E_G^a(x, y)$ holds.

We have clearly $(F_G)_y^x = \sigma((F_G)_x^y)$, hence F_G is a σ-symmetric matrix. The *F-rank-width* of a graph G over (F, σ), denoted by $rwd^F(G)$, is the F-rank-width of F_G.

A graph H over (F, σ) is a *vertex-minor* of a graph G over (F, σ) if F_H is a vertex-minor of F_G.

---

[1]We will see in Chapter 5 that our two notions of rank-width of directed graphs are also equivalent to clique-width.

We can see an undirected graph as a $\{0,1\}$-edge-colored graph where for every pair of vertices (x,y), $E_G^1(x,y)$ holds if and only if there is an edge between x and y and $E_G^0(x,y)$ holds if and only if there is no edge between x and y. Clearly this coding of G transforms it into a graph over $(GF(2), \sigma_1)$ where $\sigma_1(0) = 0$ and $\sigma_1(1) = 1$. Hence, the $GF(2)$-rank-width of G is the same as the rank-width of ρ_G given in Definition 1.9. By using also the definition of the V_G-matrix F_G of a directed graph given in Definition 3.1, we can consider a directed graph G as an edge-colored graph over $(GF(4), \sigma_2)$ where $\sigma_2(a) = a^2$ and $\sigma_2(a^2) = a$. Then Definition 4.1 is a generalization of $GF(4)$-rank-width. We do not know any application of this generalization except that considering F-rank-width will help us to extend the operations in [CK09] in such a way that we can define operations that handle algebraically directed graph classes of small $GF(4)$-rank-width.

We assume that the set of vertices of each graph is linearly ordered. This will help us to define our operations, based on linear transformations of matrices, in an unambiguous way. Let H and G be two graphs over (F, σ). Clearly if H is obtained from G by an lc-complementation at x, i.e., F_H is obtained from F_G by an lc-complementation at x, then H is obtained from G by modifying the sub-graph induced on the neighbors of x. However, how this sub-graph is modified depends on F and σ. As a consequence of Corollary 2.1 we get the following.

Corollary 4.1 *Let F be a finite field of characteristic* 2 *and let $\sigma : F \to F$ be an automorphism. For each k, there is a finite list $\mathcal{C}_k^{(F,\sigma)}$ of graphs over (F, σ) having at most $(6^{k+1} - 1)/5$ vertices such that a graph G over (F, σ) has F-rank-width at most k if and only if no graph in $\mathcal{C}_k^{(F,\sigma)}$ is isomorphic to a vertex-minor of G.*

Let F be a finite field and let $\sigma : F \to F$ be an automorphism. We will define in Section 4.1 graph operations that handle algebraically classes of graphs over (F, σ) of small F-rank-width. In Section 4.2 we specialize these operations and give graph operations that characterize exactly undirected graphs of rank-width at most k. We also give a specialization of the former operations in Section 4.3 in order to propose graph operations that handle algebraically directed graph classes of small bi-rank-width.

4.1   Algebraic Coloring of Graphs

Let $F = \{0, 1, a_1, \ldots, a_q\}$ be a finite field and let $\sigma : F \to F$ be an automorphism. If a graph G over (F, σ) is an edge-colored graph whose edges are colored with elements of F then, it is a relational structure $\langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q} \rangle$ where $E_G^a(x,y)$ holds if and only if $(F_G)_x^y = a$ holds if and only if $E_G^{\sigma(a)}(y, x)$ holds. Since in this section all graphs are over (F, σ), unless otherwise specified, we will omit the expression "over (F, σ)" for clarity when necessary.

Let k be a positive integer. An F^k-*coloring* of a graph G is a mapping $\gamma : V_G \to F^k$ with no constraint on the values of γ for adjacent vertices[2] and an F^k-*colored graph* G is the graph $\langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q}, \gamma_G \rangle$, still denoted by G, and where γ_G is an F^k-coloring of $\langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q} \rangle$. In this way an F^k-colored graph G is a graph whose vertices are colored with colors from F^k and its edges with colors from F. A graph $G = \langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q} \rangle$ is made canonically into an F^k-colored graph with $\gamma_G(x) = (0, \cdots, 0)$ for each x. We define some operations on these graphs.

---

[2]It is worth noticing that for each $x \in V_G$, $\gamma_G(x)$ is a row vector.

Definition 4.2 (Recoloring) For a mapping $h : F^k \to F^m$ and an F^k-colored graph G, we let $Recol_h(G)$ be the F^m-colored graph $K = \langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q}, \gamma_K \rangle$ where $\gamma_K = h \circ \gamma_G$.

For each vertex x of G, the operation $Recol_h$ changes the color of x into $h(\gamma_G(x))$. It does not modify the relations, hence does not modify the colors of the edges.

Definition 4.3 (Graph products) Let $f : F^k \times F^\ell \to F$, $g : F^k \to F^m$ and $h : F^\ell \to F^m$ be arbitrary mappings. For G, F^k-colored and H, F^ℓ-colored, such that $V_G \cap V_H = \emptyset$, we let $G \otimes_{f,g,h} H$ be the F^m-colored graph $K = \langle V_G \cup V_H, E_K^0, E_K^1, E_K^{a_1}, \ldots, E_K^{a_q}, \gamma_K \rangle$ where for $\alpha \in F$,

$$E_K^\alpha = E_G^\alpha \cup E_H^\alpha \cup \{(x,y) \mid x \in V_G, y \in V_H \text{ and } f(\gamma_G(x), \gamma_H(y)) = \alpha\},$$
$$\cup \{(y,x) \mid x \in V_G, y \in V_H \text{ and } f(\gamma_G(x), \gamma_H(y)) = \sigma^{-1}(\alpha)\},$$
$$\gamma_K(x) = \begin{cases} (g \circ \gamma_G)(x) & \text{if } x \in V_G, \\ (h \circ \gamma_H)(x) & \text{if } x \in V_H. \end{cases}$$

The operations $\otimes_{f,g,h}$ adds colored edges between two disjoint graphs, that are the two arguments. This is a difference with clique-width where a single binary operation $\oplus$ is used, and $\eta_{i,j}^a$ (resp. $\alpha_{i,j}^a$) applied to $G \oplus H$ may add colored edges (resp. arcs) to G and to H. This behavior of the operation $\otimes_{f,g,h}$ are analogous to the ones of the binary operations that define *NLC-width* [Wan94] except that these later do not color edges.

Definition 4.4 (Constants) For each $u \in F^k$, we let $\mathbf{u}$ be a constant denoting a graph with one vertex colored by u and no edge. If we need to specify such a graph with a particular vertex x, we use $\mathbf{u}(x)$ instead of $\mathbf{u}$. We denote by C_k^F the set $\{\mathbf{u} \mid u \in F^1 \cup \cdots \cup F^k\}$. In some occasions we will use a constant $\emptyset_k$ to denote the empty F^k-colored graph.

Example 4.1 We let G be the directed graph on Figure 4 (iii). For $1 \le i \le 5$ we let $f_i : GF(4)^1 \times GF(4)^1 \to GF(4)$, $g_i, h_i : GF(4)^1 \to GF(4)^1$ be mappings such that:

$$f_1(1,1) = a, \qquad g_1(1) = (a), \qquad h_1(1) = (a^2),$$
$$g_2(1) = (a),$$
$$f_3(1,a) = a,$$
$$f_4(1,a) = f_4(1,1) = a^2, \qquad g_4(1) = (a), \qquad h_4(a) = (a^2)$$
$$f_5(1,1) = a^2, \; f_5(a^2, a^2) = a^2, f_5(a,1) = f_5(1,a) = f_5(1,a^2) = a.$$

For all x, y in $GF(4)^1$, if $f_i(x,y)$ is not defined, we let $f_i(x,y) = 0$ and if $g_i(x)$ (resp. $h_i(x)$) is not defined, we let $g_i(x) = (x)$ (resp. $h_i(x) = (x)$). We claim that the graph G is isomorphic to the graph defined by the term $t_5 = t_3 \otimes_{f_5, g_5, h_5} t_4$ where

$$t_1 = (\mathbf{1})(x_3) \otimes_{f_1, g_1, h_1} (\mathbf{1})(x_6),$$
$$t_3 = (\mathbf{1})(x_1) \otimes_{f_3, g_3, h_3} t_1,$$
$$t_2 = (\mathbf{1})(x_2) \otimes_{f_2, g_2, h_2} (\mathbf{1})(x_4),$$
$$t_4 = (\mathbf{1})(x_5) \otimes_{f_4, g_4, h_4} t_2.$$

The term t_1 constructs the arc $x_3 \to x_6$ and recolors x_3 into (a) and x_6 into (a^2). The term t_3 constructs the arc $x_1 \to x_3$. The term t_2 creates the two vertices x_2 and x_4 and, recolors x_2 into

(a). The term t_4 adds the arcs $x_2 \to x_5$ and $x_4 \to x_5$ and, recolors x_5 into (a) and x_2 into (a^2). The term t_5 adds the arcs $x_4 \to x_1$, $x_2 \to x_6$, $x_1 \to x_5$, $x_1 \to x_2$ and $x_3 \to x_4$. The graph constructed by t_5 is clearly isomorphic to G.

Remark 4.1   1. The disjoint union of G, F^k-colored and H, F^ℓ-colored with $k \leq \ell$ is $G \otimes_{f,g,h} H$ where $f(u,v) = 0$, $g(u) = (u, 0, \cdots, 0) \in F^\ell$ and $h(v) = v$ for all $u \in F^k$ and all $v \in F^\ell$.

2. We have $G \otimes_{f,g,h} H = H \otimes_{\tilde{f},h,g} G$ where $\tilde{f}(u,v) = \sigma(f(v,u))$.

3. The recoloring operations can actually be combined with other operations. The following rules are clear:

$$Recol_m(\mathbf{u}) = \mathbf{v} \qquad \text{if } v = m(u).$$
$$Recol_m(G \otimes_{f,g,h} H) = G \otimes_{f,m \circ g, m \circ h} H.$$
$$Recol_m(G) \otimes_{f,g,h} Recol_{m'}(H) = G \otimes_{f', g \circ m, h \circ m'} H$$

where $f'(u,v)$ is defined as $f(m(u), m'(v))$. Let us also note that

$$G \otimes_{f,g,h} \emptyset_k = Recol_g(G).$$

Let $n \in \mathbb{N}$. We let B_n^F be the finite set of operations $Recol_h$, $\otimes_{f,g,h}$ where $g : F^k \to F^m$, $h : F^\ell \to F^m$ and $f : F^k \times F^\ell \to F$ are mappings such that $k, \ell, m \leq n$. Without loss of generality we may assume $k, l, m \neq 0$. For $n \geq 1$, every term $t \in T(B_n^F, C_n^F)$ has for value an F^n-colored graph, denoted by $val(t)$, or actually the family of all graphs isomorphic to such a graph.

We now explain how such operations fit into the logical framework of [CMR00, Cou92].

Definition 4.5 (F^k-colored graphs as binary relational structures) Let us introduce unary relations $c^{i,\alpha}$ for $i \in [n]$ and $\alpha \in F$. The meaning of $c^{i,\alpha}(x) = \text{TRUE}$ will be "the i-th component of $\gamma_G(x)$ is α". Hence, an F^k-colored graph $G = \langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q}, \gamma_G \rangle$, for $k \leq n$, is described exactly by the relational structure with domain V_G that we will also denote by G:

$$\langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q}, c_G^{1,0}, \ldots, c_G^{n,0}, c_G^{1,1}, \ldots, c_G^{n,1}, \ldots, c_G^{1,a_q}, \ldots, c_G^{n,a_q} \rangle.$$

For an F^k-colored graph, the predicates $c_G^{i,\alpha}(x)$, for $k+1 \leq i \leq n$ and $\alpha \in F$, will be FALSE. Every relational structure of this form, and such that for each $\alpha \in F$, E_G^α is σ-symmetric ($E_G^\alpha(x,y)$ holds if and only if $E_G^{\sigma(\alpha)}(y,x)$ holds) and irreflexive ($E_G^\alpha(x,x)$ never holds) represents an F^k-colored graph G, $k \leq n$.

Quantifier-free operations are defined in Section 1.4.

Proposition 4.1 *For each positive integer n, we have*

1. *The operations $Recol_h$ are quantifier-free operations for any mapping $h : F^k \to F^m$, $k, m \leq n$.*

2. *The operations $\otimes_{f,g,h}$ are expressible in terms of $\oplus$ and quantifier-free operations for all mappings $f : F^k \times F^\ell \to F$, $g : F^k \to F^m$ and $h : F^\ell \to F^m$, $k, \ell, m \leq n$.*

Proof. (1) is clear.

(2) Let n be a fixed positive integer. We consider F^k-colored graphs for $k \leq n$. In addition to the unary predicates $c^{1,0}, \ldots, c^{n,a_q}$, we will use auxiliary unary ones $d^{1,0}, \ldots, d^{n,a_q}$ ($d^{i,\alpha} \notin \{c^{1,0}, \ldots, c^{n,a_q}\}$). If $K = G \otimes_{f,g,h} H$, then

$$K = \alpha(\eta^0(\eta^1(\eta^{a_1}(\ldots(\eta^{a_q}(G \oplus \beta(H))) \ldots))))$$

where β replaces in H each $c^{i,\alpha}$ by $d^{i,\alpha}$ (i.e., $d^{i,\alpha}_{\beta(H)}(x)$ holds if and only if $c^{i,\alpha}_H(x)$ holds, and then, $c^{i,\alpha}_{\beta(H)}(x)$ does not hold), η^α creates "edges" in E^α_K, by redefining $E^\alpha_G(x,y)$ with the following formula where in the definition of $E^{\alpha'}$, u and v range over F^n (we let $u[i]$ denote the i-th component of u):

$$E^\alpha_G(x,y) \vee E^\alpha_H(x,y) \vee E^{\alpha'}(x,y) \vee E^{\alpha'}(y,x)$$

and $E^{\alpha'}(x,y)$ is

$$\bigvee_{f(u,v)\in\{\alpha,\sigma^{-1}(\alpha)\}} \left(\bigwedge_{a\in F} \left(\bigwedge_{u[\ell]=a} c^{\ell,a}(x) \wedge \bigwedge_{u[\ell]\neq a} \neg c^{\ell,a}(x) \wedge \bigwedge_{v[s]=a} d^{s,a}(y) \wedge \bigwedge_{v[j]\neq a} \neg d^{s,a}(y) \right) \right).$$

The operation α performs the recolorings defined by g and h. ∎

In the case of undirected graphs considered as graphs over $(GF(2), \sigma_1)$, we consider all the couples created by the operation η^0 as non-edges. The same holds for directed graphs considered as graphs over $(GF(4), \sigma_2)$.

Theorem 4.1 *For each monadic second-order graph property P and for each $n \in \mathbb{N}$, there exists an algorithm that checks in $O(|t|)$-time for every term $t \in T(B^F_n, C^F_n)$ if the graph defined by this term satisfies P.*

Proof. This result is proved in [CMR00] for $T(F^{uc}_n, C^c_n)$ instead of $T(B^F_n, C^F_n)$, but it extends to all quantifier-free definable operations as proved in [Cou92]. The logical foundations of this result are presented in details by Makowsky in [Mak04]. ∎

Given a graph G that we know is the value of a term in $T(B^F_n, C^F_n)$, we need an algorithm that constructs this term in order to use Theorem 4.1. We will see in Chapter 5 that if a graph G is the value of a term in $T(B^F_n, C^F_n)$, then we can construct in $O(|V_G|^3)$ a term in $T(B^F_m, C^F_m)$ that defines G, $m = 2 \cdot |F|^n$.

4.2   Algebraic Operations for F-Rank-Width of Graphs

Let $F = \{0, 1, a_1, \ldots, a_q\}$ be a finite field and let $\sigma : F \to F$ be an automorphism. All the graphs in this section are over (F, σ) and for clarity we will omit the expression "over (F, σ)" when the context is clear. We specialize the operations defined in the previous section by taking advantage of the vector space structure of F^k over the field F. We denote by M^T the *transpose* of a matrix M and we let $O_{k,\ell}$ and I_k denote respectively the $(k \times \ell)$-null matrix and the $(k \times k)$-identity matrix.

Let $k \geq 1$. With an F^k-colored graph $G = \langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q}, \gamma_G \rangle$, we associate the $(V_G, [k])$-*color matrix* Γ_G, the row vectors of which are the vectors $\gamma_G(x)$ in F^k for x in V_G. We define the *color-rank* of G as the rank of Γ_G and we denote it by $crk(G)$. Clearly, $crk(G) \leq k$ if G is F^k-colored[3]. We now define specializations of the operations defined in Section 4.1.

Definition 4.6 (Linear recolorings) A recoloring $Recol_h$ is *linear* if $h : F^k \to F^m$ is a linear function, in other words, if for some $(k \times m)$-matrix N and all F^k-colored graphs G, we have by letting $H = Recol_h(G)$,

$$\Gamma_H = \Gamma_G \cdot N,$$

i.e., $\gamma_H(x) = \gamma_G(x) \cdot N$ for each x in V_G.

If $Recol_h$ and $Recol_{h'}$ are linear recolorings, described respectively by N and N', then $Recol_h \circ Recol_{h'}$ is linear and is described by $N' \cdot N$.

Definition 4.7 (Bilinear product of graphs) We consider the operations $\otimes_{f,g,h}$ where:

- $f : F^k \times F^\ell \to F$ is *bilinear*, hence defined by $f(u, v) = (u \cdot M) \cdot v^T$ where M is a $(k \times \ell)$-matrix;

- the recoloring mappings $g : F^k \to F^m$ and $h : F^\ell \to F^m$ are linear.

We order the graph $K = G \otimes_{f,g,h} H$ by preserving the orderings of V_G and V_H and letting $x < y$ for $x \in V_G$ and $y \in V_H$. In terms of products of matrices we have thus:

$$F_K = \begin{pmatrix} A_G & \Gamma_G \cdot M \cdot \Gamma_H^T \\ \sigma(\Gamma_H \cdot M^T \cdot \Gamma_G^T) & A_H \end{pmatrix},$$

$$\Gamma_K = \begin{pmatrix} \Gamma_G \cdot N \\ \Gamma_H \cdot P \end{pmatrix}$$

where M, N and P are the matrices describing f, g and h respectively. We will use in this case the notation $\otimes_{M,N,P}$ for $\otimes_{f,g,h}$.

---

[3]The color-rank of G should not be confused with its *rank*, defined as the rank of its adjacency matrix F_G over F. All ranks are relative to F.

Remark 4.2 1. If $K = G \otimes_{M,N,P} H$ is $\mathbb{B}^m$-colored, then we have:

$$(F_K)_{V_G}^{V_H} = \Gamma_G \cdot M \cdot \Gamma_H^T,$$
$$(\Gamma_K)_{V_G}^{[m]} = \Gamma_{K[V_G]} = \Gamma_G \cdot N,$$
$$(\Gamma_K)_{V_H}^{[m]} = \Gamma_{K[V_H]} = \Gamma_H \cdot P.$$

Since for all matrices,

$$rk(A \cdot B) \leq \min\{rk(A), rk(B)\},$$

we have

$$crk(K[V_G]) = rk((\Gamma_K)_{V_G}^{[m]}) \leq rk(\Gamma_G) \leq k,$$

and symmetrically,

$$crk(K[V_H]) = rk((\Gamma_K)_{V_H}^{[m]}) \leq rk(\Gamma_H) \leq \ell.$$

2. We have $K = G \otimes_{M,N,P} H = H \otimes_{M^T,P,N} G$ if $(F_K)_{V_G}^{V_H} = (F_K)_{V_H}^{V_G}$.

3. The following rules are clear:

$$Recol_Q(\mathbf{u}) = \mathbf{v} \qquad \text{if } v = u \cdot Q,$$
$$Recol_Q(G \otimes_{M,N,P} H) = G \otimes_{M,N \cdot Q,P \cdot Q} H,$$
$$Recol_Q(G) \otimes_{M,N,P} Recol_{Q'}(H) = G \otimes_{Q \cdot M \cdot Q'^T, Q \cdot N, Q' \cdot P} H,$$
$$G \otimes_{M,N,P} \emptyset_k = Recol_N(G).$$

We let $R_n^F \subseteq B_n^F$ be the set of linear recolorings $Recol_N$ and 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 $k, \ell, m \leq n$. We denote by $val(t)$ the graph defined, up to isomorphism, by a term $t \in T(R_n^F, C_n^F)$. This graph is the value of the term in the corresponding algebra. Two terms are *equivalent* if they define, up to isomorphism, the same graph.

Remark 4.3 We can transform every term $t \in T(R_n^F, C_n^F)$ into a term $t' \in T(R_n^F, C_n^F)$ where each constant $\mathbf{u} \in F^n$ and each operation $Recol_N$ or $\otimes_{M,N,P}$ is such that M, N and P are $(n \times n)$-matrices. For that, we use the following recursive rules:

$$t' = \begin{cases} (u, 0_{1,n-k}) & \text{if } t = \mathbf{u} \text{ and } u \in F^k, \\ Recol_{N'}(t_1') & \text{if } t = Recol_N(t_1), \\ t_1' \otimes_{M',N',P'} t_2' & \text{if } t = t_1 \otimes_{M,N,P} t_2. \end{cases}$$

where M, N and P are respectively $(k \times \ell)$, $(k \times m)$ and $(\ell \times m)$-matrices and

$$M' = \begin{pmatrix} M & 0_{k,n-\ell} \\ 0_{n-k,\ell} & 0_{n-k,n-\ell} \end{pmatrix} \qquad N' = \begin{pmatrix} N & 0_{k,n-m} \\ 0_{n-k,m} & 0_{n-k,n-m} \end{pmatrix} \qquad P' = \begin{pmatrix} P & 0_{\ell,n-m} \\ 0_{n-\ell,m} & 0_{n-\ell,n-m} \end{pmatrix}.$$

It is straightforward to verify that t' is equivalent to t. So, without loss of generality, we can replace R_n^F by $R_n'^F$ consisting of bilinear products $\otimes_{M,N,P}$ where M, N and P are $(n \times n)$-matrices. We also use Remark 4.2(3) showing that recolorings can be combined with bilinear products.

Our objective is to prove the following, which is the main theorem of this chapter.

Theorem 4.2 (Algebraic Characterization of Rank-Width) *Let G be a graph over (F, σ) where F_G is symmetric. Then G has F-rank-width at most n if and only if G is the value of a term in $T(R_n^F, C_n^F)$.*

We will prove it in two steps. We first prove the following stronger proposition which is the "if direction".

Proposition 4.2 *Let $G = val(t)$ where $t \in T(R_n^F, C_n^F)$. Then $rwd^F(G) \leq n$ if F_G is symmetric. Otherwise, $rwd^F(G) \leq 2n$.*

We recall that $s \bullet t = s[t/u]$ for $s \in Cxt(R_n^F, C_n^F)$, $t \in T(R_n^F, C_n^F)$ and Id is the particular context u. Before proving Proposition 4.2, we state and prove the following lemma.

Lemma 4.1 *Let $t = c \bullet t'$ where $t' \in T(R_n'^F, C_n^F)$, $c \in Cxt(R_n'^F, C_n^F) - \{Id\}$. If we let $G = val(t)$ and $H = val(t')$, then*

$$(F_G)_{V_H}^{V_G - V_H} = (\Gamma_H \cdot B \qquad \sigma(\Gamma_H \cdot B'))$$
$$\Gamma_{G[V_H]} = \Gamma_H \cdot C$$

for some matrices B, B' and C. If F_G is symmetric then, $\sigma(\Gamma_H \cdot B') = \Gamma_H \cdot B'$.

Proof. We use an induction on the structure of c. We have several cases:

(a) $c = Id \otimes_{M,N,P} t''$.
   We let $K = val(t'')$. Then $G = H \otimes_{M,N,P} K$. We have, as observed above,

$$(F_G)_{V_H}^{V_K} = \Gamma_H \cdot M \cdot \Gamma_K^T,$$
$$\Gamma_{G[V_H]} = \Gamma_H \cdot N.$$

Hence we take $B = M \cdot \Gamma_K^T, B' = 0$ and $C = N$. The last assertion is clear.

The proof is similar if $c = t'' \otimes_{M,N,P} Id$, we take $B = 0, B' = M^T \cdot \Gamma_K^T$ because $(F_G)_{V_H}^{V_K} = \sigma(\Gamma_H \cdot M^T \cdot \Gamma_K^T)$ and $C = P$. The second assertion is clear since $\sigma(\Gamma_H \cdot M^T \cdot \Gamma_K^T) = \Gamma_H \cdot M^T \cdot \Gamma_K^T$ if F_G is symmetric.

(b) $c = c' \otimes_{M,N,P} t''$ where $c' \in Cxt(R_n'^F, C_n^F) - \{Id\}$.
   We let $K = val(t'')$ and $G' = val(c' \bullet t')$. Hence, $G = G' \otimes_{M,N,P} K$. We recall that:

$$F_G = \begin{pmatrix} F_{G'} & \Gamma_{G'} \cdot M \cdot \Gamma_K^T \\ \sigma(\Gamma_K \cdot M^T \cdot \Gamma_{G'}^T) & F_K \end{pmatrix}.$$

Hence

$$(F_G)_{V_H}^{V_G - V_H} = \left((F_{G'})_{V_H}^{V_{G'} - V_H} \qquad (\Gamma_{G'} \cdot M \cdot \Gamma_K^T)_{V_H}^{V_K} \right).$$

By inductive hypothesis, $(F_{G'})_{V_H}^{V_{G'}-V_H} = (\Gamma_H \cdot B' \qquad \sigma(\Gamma_H \cdot B''))$ and $\Gamma_{G'[V_H]} = \Gamma_H \cdot C'$. We now prove that $(\Gamma_{G'} \cdot M \cdot \Gamma_K^T)_{V_H}^{V_K} = \Gamma_H \cdot C''$ for some matrices C''.

$$
\begin{aligned}
\left(\Gamma_{G'} \cdot M \cdot \Gamma_K^T\right)_{V_H}^{V_K} &= (\Gamma_{G'})_{V_H}^{[n]} \cdot M \cdot \Gamma_K^T && \text{by definition,} \\
&= \Gamma_{G'[V_H]} \cdot M \cdot \Gamma_K^T && \text{by definition,} \\
&= \Gamma_H \cdot C' \cdot M \cdot \Gamma_K^T && \text{by inductive hypothesis.}
\end{aligned}
$$

Hence $(F_G)_{V_H}^{V_G-V_H} = \left(\Gamma_H(B' \quad C' \cdot M \cdot \Gamma_K^T) \qquad \sigma(\Gamma_H \cdot B'')\right)$. We now consider $\Gamma_{G[V_H]}$. We have:

$$
\Gamma_G = \begin{pmatrix} \Gamma_{G'} \cdot N \\ \Gamma_K \cdot P \end{pmatrix}.
$$

Then $\Gamma_{G[V_H]} = (\Gamma_{G'} \cdot N)_{V_H}^{[n]} = (\Gamma_{G'})_{V_H}^{[n]} \cdot N = \Gamma_H \cdot C' \cdot N$.

If F_G is symmetric, $\sigma(\Gamma_H \cdot B'') = \Gamma_H \cdot B''$, this proves the lemma, because the case of $c = t'' \otimes_{M,N,P} c'$ is similar. ∎

We can now prove Proposition 4.2.

Proof of Proposition 4.2. Let $G = val(t)$ where $t \in T(R_n^F, C_n^F)$. We transform it into a term $\tilde{t}$ in $T(R_n^{'F}, C_n)$ with $red(\tilde{t}) = red(t)$. We take $red(\tilde{t})$ as a layout of G. We claim that the F-rank-width of this layout is at most n. It is sufficient to prove that, for each subterm t' of t,

$$
rk((F_G)_{V_{val(t')}}^{V_G-V_{val(t')}}) \leq 2n.
$$

Let t' be a subterm of t and let $H = val(t')$. By Lemma 4.1, we have $(F_G)_{V_H}^{V_G-V_H} = \left(\Gamma_H \cdot B \qquad \sigma(\Gamma_H \cdot B')\right)$. Assume first that F_G is not symmetric. It is clear that $(F_G)_{V_H}^{V_G-V_H} = \left(\Gamma_H \cdot B \qquad 0\right) + \left(0 \qquad \sigma(\Gamma_H \cdot B')\right)$, i.e., $rk((F_G)_{V_H}^{V_G-V_H}) \leq rk(\Gamma_H \cdot B) + rk(\Gamma_H \cdot B')$ because $rk(A + B) \leq rk(A) + rk(B)$ [Lip91]. Then $rk((F_G)_{V_H}^{V_G-V_H}) \leq 2n$ since H is F^n-colored and $rk(A \cdot B) \leq \min\{rk(A), rk(B)\}$.

Assume now that F_G is symmetric. By Lemma 4.1, we have $\sigma(\Gamma_H \cdot B') = \Gamma_H \cdot B'$. Hence $(F_G)_{V_H}^{V_G-V_H} = \Gamma_H(B \qquad B')$. Then $rk((F_G)_{V_H}^{V_G-V_H}) \leq n$. ∎

We have then proved that if $t \in T(R_n^F, C_n^F)$ and $G = val(t)$ is such that F_G is symmetric, then $rwd^F(G) \leq n$. For proving the converse direction, stated as Proposition 4.5, we need some technical lemmas. Let us introduce some definitions before. We write $G = H \otimes_M K$ instead of $H \otimes_{M,N,P} K$ if we do not care about the coloring of G but only about its vertices and edges. More precisely $\otimes_M$ is an abbreviation for $\otimes_{M,O,O}$ where O denotes zero-matrices. We recall that a graph without colors has all its vertices colored by a row vector $(0, \cdots, 0)$. For $X \subseteq V_G$, we denote by $\rho_G^F(X)$ the rank of $(F_G)_X^{V_G-X}$, which is by definition $\rho_{F_G}^F(X)$ (cf. the definition of F-rank-width of σ-symmetric matrices in Chapter 2).

Let G be a graph and (V_1, V_2) be a bipartition of its vertices such that $\rho_G^F(V_1) = m$. We say that vertices $x_1, \ldots, x_m$ in V_1 form a *vertex basis of $G[V_1]$ with respect to G* if their associated row vectors in $(F_G)_{V_1}^{V_2}$ are independent. Vertices $x_1, \ldots, x_p$ in V_1 with $p \geq \rho_G^F(V_1)$ form a *vertex generator of $G[V_1]$ with respect to G* if their associated row vectors generate the row vectors of $(F_G)_{V_1}^{V_2}$. We now introduce the notion of *presentation*, which will allow us to construct a term in $T(R_n^F, C_n^F)$ from a layout by induction. For $u = (u_1, \ldots, u_k)$ in F^k, we let $\sigma(u)$ be the row vector $(\sigma(u_1), \ldots, \sigma(u_k))$ in F^k and for a matrix $M = (m_{i,j})$ over F, we let $\sigma(M)$ be the matrix $(\sigma(m_{i,j}))$ over F.

Definition 4.8 (Presentation) Let $G = \langle V_G, E_G^0, E_G^1, E_G^{a_1}, \ldots, E_G^{a_q} \rangle$ and let (V_1, V_2) be a bipartition of V_G with $A = (F_G)_{V_1}^{V_2}$. Let $X = \{z_1, \ldots, z_p\} \subseteq V_1$ be a vertex generator of $G[V_1]$ with respect to G. The set of row vectors $A_{z_i}^{V_2}$ generates the same vector space as the set of all row vectors of A. Then for each $x \in V_1$, there exists a row vector $b_x = (b_{xz_1}, \ldots, b_{xz_p})$ in F^p such that

$$A_x^{V_2} = b_x \cdot \left(A_{z_1}^{V_2} + \cdots + A_{z_p}^{V_2} \right)^T$$

Clearly, for each $z \in X$, the row vector b_z is such that, for each $z' \in X$,

$$b_{zz'} = \begin{cases} 1 & \text{if } z = z', \\ 0 & \text{otherwise.} \end{cases}$$

For each $x \in V_1 - X$ and each $z \in X$, b_{xz} can be any element of F. By hypothesis, G is linearly ordered, hence V_1 is linearly ordered, the order induced by the order of G. We let $v_1, \ldots, v_n$ be the order of V_1 induced by the order of G. We let N be the (V_1, X)-matrix $\left(b_{v_1}, \ldots, b_{v_n} \right)^T$, i.e., for each i in $[n]$, the i-th row of N is b_{v_i}.

Let H' be an F^n-coloring of $G[V_1]$ such that $\gamma_{H'}(v_i) = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 at i-th position. We let H be such that:

$$H = \begin{cases} Recol_N(H') & \text{if } F_G \text{ is symmetric,} \\ Recol_{(N \quad \sigma(N))}(H') & \text{otherwise.} \end{cases}$$

We call (H, N, X) a *presentation of $G[V_1]$ relative to G*. It is clear that H is an F^p-coloring of $G[V_1]$ and $\Gamma_H = N$ if F_G is symmetric, otherwise H is an F^{2p}-coloring of $G[V_1]$ and $\Gamma_H = (N \quad \sigma(N))$.

Proposition 4.3 *Let G be a graph over (F, σ) and let (V_1, V_2) be a biparition of V_G. Let $X \subseteq V_1$ and $Y \subseteq V_2$ be vertex generators of $G[V_1]$ and $G[V_2]$ respectively, both with respect to G. Let (H, N, X) and (K, P, Y) be presentations of $G[V_1]$ and $G[V_2]$ respectively, both relative to G. Then $(F_G)_{V_1}^{V_2} = N \cdot M \cdot \sigma(P^T)$ where $M = (F_G)_X^Y$. Furthermore, $G = H \otimes_{M'} K$ where if $p = |X|$ and $q = |Y|$*

$$M' = \begin{cases} M & \text{if } F_G \text{ is symmetric,} \\ \begin{pmatrix} 0_{p,q} & M \\ 0_{p,q} & 0_{p,q} \end{pmatrix} & \text{otherwise.} \end{cases}$$

Proof. For convenience we let $A = F_G$. By definition, that (H, N, X) is a presentation of $G[V_1]$ relative to G means that $A_{V_1}^{V_2} = N \cdot A_X^{V_2}$. Similarly $A_{V_2}^{V_1} = P \cdot A_Y^{V_1}$ since (K, P, Y)

is a presentation of $G[V_2]$ relative to G. Then $A_{V_2}^X = P \cdot A_Y^X$, i.e., $A_X^{V_2} = \sigma((P \cdot A_Y^X)^T) = \sigma((A_Y^X)^T) \cdot \sigma(P^T)$. But $A_X^Y = \sigma((A_Y^X)^T)$. Hence, $A_{V_1}^{V_2} = N \cdot M \cdot \sigma(P^T)$.

We now prove that $G = H \otimes_{M'} K$. It is sufficient to prove that $A_{V_1}^{V_2} = (F_{G'})_{V_1}^{V_2}$ where $G' = H \otimes_{M'} K$. Assume first that F_G is not symmetric. By definition of a presentation, we have $\Gamma_H = \begin{pmatrix} N & \sigma(N) \end{pmatrix}$ and $\Gamma_K = \begin{pmatrix} P & \sigma(P) \end{pmatrix}$. It is now clear that $(F_{G'})_{V_1}^{V_2} = N \cdot M \cdot \sigma(P^T)$

Assume now that F_G is symmetric. By definition of a presentation, we have $\Gamma_H = N$ and $\Gamma_K = P$. Hence, $(F_{G'})_{V_1}^{V_2} = \Gamma_H \cdot M \cdot \Gamma_K^T = N \cdot M \cdot P^T = N \cdot M \cdot \sigma(P^T) = A_{V_1}^{V_2}$.   ∎

Remark 4.4   1. If $k = rk((F_G)_{V_1}^{V_2})$ in Proposition 4.3, we have necessarily $p = |X| \geq k$, $q = |Y| \geq k$. If $p = q = k$, then X and Y are vertex bases of $G[V_1]$ and $G[V_2]$ respectively, both with respect to G.

   2. If $V_1 \subseteq V$ and X is a vertex basis of $G[V_1]$ with respect to G, then there is a unique presentation (H, N, X) of $G[V_1]$ relative to G .

Example 4.2 We let G be a directed graph such that V_1 is the set $\{x_1, \ldots, x_6\}$ and V_2 is the set $\{y_1, \ldots, y_7\}$ and let $A = (F_G)_{V_1}^{V_2}$ be the following.

$$
A = \begin{array}{c|c|c|c|c|c|c|c|}
 & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \\
\hline
x_1 & a & a^2 & 1 & a & 0 & 0 & 0 \\
\hline
x_2 & 0 & 0 & 0 & a^2 & a & 1 & 1 \\
\hline
x_3 & a^2 & 0 & a & 1 & 0 & a^2 & 0 \\
\hline
x_4 & a^2 & 1 & a & 0 & a & 1 & 1 \\
\hline
x_5 & a^2 & 0 & a & a^2 & 1 & 0 & a^2 \\
\hline
x_6 & 0 & 1 & 0 & 0 & 1 & 0 & a^2 \\
\hline
\end{array}
$$

We choose $X = \{x_1, x_2, x_3\}$ and $Y = \{y_2, y_5, y_6\}$ as vertex bases of $G[V_1]$ and $G[V_2]$ respectively, both with respect to G. We have thus the following linear relations:

$$
\begin{aligned}
A_{x_4}^{V_2} &= a \cdot A_{x_1}^{V_2} + A_{x_2}^{V_2}, \\
A_{x_5}^{V_2} &= a^2 \cdot A_{x_2}^{V_2} + A_{x_3}^{V_2}, \\
A_{x_6}^{V_2} &= a \cdot A_{x_1}^{V_2} + a^2 \cdot A_{x_2}^{V_2} + A_{x_3}^{V_2}
\end{aligned}
$$

and,

$$
\begin{aligned}
A_{y_1}^{V_1} &= a \cdot A_{y_2}^{V_1} + a \cdot A_{y_5}^{V_1} + A_{y_6}^{V_1}, \\
A_{y_3}^{V_1} &= a^2 \cdot A_{y_2}^{V_1} + a^2 \cdot A_{y_5}^{V_1} + a \cdot A_{y_6}^{V_1}, \\
A_{y_4}^{V_1} &= a \cdot A_{y_2}^{V_1} + a \cdot A_{y_5}^{V_1} + a^2 \cdot A_{y_6}^{V_1}, \\
A_{y_7}^{V_1} &= a \cdot A_{y_5}^{V_1}.
\end{aligned}
$$

The corresponding $GF(4)^6$-colorings H and K of $G[V_1]$ and $G[V_2]$ are respectively defined by:

x_1	1	0	0	1	0	0
x_2	0	1	0	0	1	0
x_3	0	0	1	0	0	1
x_4	a	1	0	a^2	1	0
x_5	0	a^2	1	0	a	1
x_6	a	a^2	1	a^2	a	1

y_1	a	a	1	a^2	a^2	1
y_2	1	0	0	1	0	0
y_3	a^2	a^2	a	a	a	a^2
y_4	a	a	a^2	a^2	a^2	a
y_5	0	1	0	0	1	0
y_6	0	0	1	0	0	1
y_7	0	a	0	0	a^2	0

We have

$$M = A_X^Y = \begin{array}{|c|c|c|c|} \hline & y_2 & y_5 & y_6 \\ \hline x_1 & a^2 & 0 & 0 \\ \hline x_2 & 0 & a & 1 \\ \hline x_3 & 0 & 0 & a^2 \\ \hline \end{array} \qquad M' = \begin{pmatrix} 0_{3,3} & M \\ 0_{3,3} & 0_{3,3} \end{pmatrix}$$

We can check for an example that:

$$\gamma_H(x_4) \cdot M' \cdot \gamma_K(y_5) = \begin{pmatrix} 0 & 0 & 0 & 1 & a & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}^T = a = A_{x_4}^{y_5}$$

$$\gamma_H(x_1) \cdot M' \cdot \gamma_K(y_2) = \begin{pmatrix} 0 & 0 & 0 & a^2 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}^T = a^2 = A_{x_1}^{y_2}$$

$$\gamma_H(x_5) \cdot M' \cdot \gamma_K(y_1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a & a & 1 & a^2 & a^2 & 1 \end{pmatrix}^T = a^2 = A_{x_4}^{y_5}$$

We can now state some basic properties of presentations.

Fact 4.1 *Let G be a graph with a biparition (V_1, V_2) of V_G. Let (H, N, X) and (K, P, Y) be presentations of $G[V_1]$ and $G[V_2]$ respectively, both relative to G. Let $Z \subseteq V_1 \cup V_2$ and $M = (F_G)_X^Y$. Then*

$$G[Z] = H[Z \cap V_1] \otimes_M K[Z \cap V_2].$$

Fact 4.2 *Let G be a graph. If $X' \subseteq X \subseteq V \subseteq V_G$ and X', X are vertex generators of $G[V]$ with respect to G, then:*

$$(F_G)_V^{V_G - V} = N \cdot (F_G)_X^{V_G - V},$$
$$(F_G)_X^{V_G - V} = N' \cdot (F_G)_{X'}^{V_G - V},$$

for some (V, X)-matrix N and some (X, X')-matrix N' and

$$(F_G)_V^{V_G - V} = (N \cdot N') \cdot (F_G)_{X'}^{V_G - V}.$$

Proposition 4.4 *Let G be a graph over (F, σ) and $V \subseteq V_G$, and let (V_1, V_2) be a bipartition of V. Let (H_1, N_1, X_1) and (H_2, N_2, X_2) be presentations of $G[V_1]$ and $G[V_2]$ respectively, both relative to G. Then there exist a vertex basis $Z \subseteq X_1 \cup X_2$ of $G[V]$ and a presentation (H, N, Z) of $G[V]$ relative to G such that*

$$H = H_1 \otimes_{M, P_1, P_2} H_2$$

where M is an (X_1, X_2)-matrix, P_1 is an (X_1, Z)-matrix and P_2 is an (X_2, Z)-matrix if F_G is symmetric, otherwise M, P_1 and P_2 are respectively $(2h \times 2k), (2h \times 2\ell)$ and $(2k \times 2\ell)$-matrices where $h = |X_1|$, $k = |X_2|$ and $\ell = |Z|$.

Proof. We let $n = |V_1|$, $m = |V_2|$ and $|V_G - V| = p$. We first assume that F_G is symmetric. By Proposition 4.3, we have $G = H_1 \otimes_{M'} K$ where $M' = (F_G)_{X_1}^{X_2 \cup (V_G - V)}$ and $(K, N_2', X_2 \cup (V_G - V))$ is a presentation of $G[V_G - V_1]$ relative to G with:

$$N_2' = \begin{array}{c|c|c} \diagbox{}{} & X_2 & V_G - V \\ \hline\hline V_2 & N_2 & 0_{m,p} \\ \hline V_G - V & 0_{p,k} & I_p \end{array}$$

Hence, by Fact 4.1, $G[V] = (H_1 \otimes_{M'} K)[V_1 \cup V_2] = H_1 \otimes_M H_2$ where $M = (M')_{X_1}^{X_2}$ since

$$K[V_2] = H_2,$$
$$(N_2')_{V_2}^{X_2} = N_2,$$
$$(X_2 \cup (V_G - V)) \cap V_2 = X_2.$$

It remains to define Z, P_1 and P_2 such that (H, N, Z) is a presentation of $G[V]$ relative to G where:

$$H = H_1 \otimes_{M,P_1,P_2} H_2,$$
$$N = \begin{pmatrix} \Gamma_{H_1} \cdot P_1 \\ \Gamma_{H_2} \cdot P_2 \end{pmatrix}.$$

Let $X_1 = \{x_1, \ldots, x_h\}$, $X_2 = \{y_1, \ldots, y_k\}$ and $\ell = \rho_G^F(V)$. We let $A = (F_G)_V^{V_G - V}$.

Claim 4.1 $X_1 \cup X_2$ is a vertex generator of $G[V]$ with respect to G.

Proof of Claim 4.1. We consider the matrix $(F_G)_{V_1}^{V_G - V_1}$. Its row vectors are generated by those associated with X_1. Thus, so are those of $(F_G)_{V_1}^{V_G - V}$ which are projections of the latter ones. Similarly the row vectors of $(F_G)_{V_2}^{V_G - V}$ are generated by those associated with X_2. Hence, $X_1 \cup X_2$ is a vertex generator of $G[V_1 \cup V_2]$ with respect to G. ∎

One can thus find a vertex basis $Z \subseteq X_1 \cup X_2$. It consists of ℓ vertices. Without loss of generality, we can assume that $Z = \{x_1, \ldots, x_{h'}, y_1, \ldots, y_{k'}\}$ for some $h' \le h$ and $k' \le k$. Let $W = V_G - V$. For each s such that $h' < s \le h$, we have a vector w_s such that:

$$A_{x_s}^W = w_s \cdot \left(A_{x_1}^W, \cdots, A_{x_{h'}}^W, A_{y_1}^W, \cdots, A_{y_{k'}}^W \right)^T.$$

We let

$$P_1 = \begin{pmatrix} \begin{array}{c|c} I_{h'} & 0_{h',\ell-h'} \\ \hline \multicolumn{2}{c}{w_{h'+1}} \\ \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{w_h} \end{array} \end{pmatrix}.$$

Thus P_1 is an (X_1, Z)-matrix. Similarly, for $k' < u \leq k$, we have w'_u such that:

$$A^W_{y_u} = w'_u \cdot \left(A^W_{x_1}, \cdots, A^W_{x_{h'}}, A^W_{y_1}, \cdots, A^W_{y_{k'}} \right)^T.$$

We let

$$P_2 = \left(\begin{array}{c} \begin{array}{c|c} 0_{k', \ell - k'} & I_{k'} \end{array} \\ \hline w'_{k'+1} \\ \vdots \\ w'_k \end{array} \right).$$

It is an (X_2, Z)-matrix. We let $H = H_1 \otimes_{M, P_1, P_2} H_2$. Let $x \in V_1$ and $z \in W$. We wish to prove that:

$$A^z_x = \gamma_H(x) \cdot \left(A^z_{x_1}, \cdots, A^z_{x_{h'}}, A^z_{y_1}, \cdots, A^z_{y_{k'}} \right)^T.$$

Since (H_1, N_1, X_1) is a presentation of $G[V_1]$, we have

$$(F_G)^z_x = \gamma_{H_1}(x) \cdot \left((F_G)^z_{x_1}, \cdots, (F_G)^z_{x_h} \right)^T$$

But P_1 is defined in such a way that:

$$\begin{pmatrix} (F_G)^z_{x_1} \\ \vdots \\ (F_G)^z_{x_h} \end{pmatrix} = P_1 \cdot \left((F_G)^z_{x_1}, \cdots, (F_G)^z_{x_{h'}}, (F_G)^z_{y_1}, \cdots, (F_G)^z_{y_{k'}} \right)^T.$$

Hence,

$$(F_G)^z_x = \gamma_{H_1}(x) \cdot P_1 \cdot \left(A^z_{x_1}, \cdots, A^z_{x_{h'}}, A^z_{y_1}, \cdots, A^z_{y_{k'}} \right)^T$$

But, it is clear that $\gamma_{H_1}(x) \cdot P_1 = \gamma_H(x)$.

The proof is similar for $(F_G)^z_y$ for $y \in V_2$ and $z \in W$. It remains now to consider non symmetric σ-symmetric matrices. Let

$$M = \begin{pmatrix} 0_{h,k} & (M')^{X_2}_{X_1} \\ 0_{h,k} & 0_{h,k} \end{pmatrix} \qquad P'_1 = \begin{pmatrix} P_1 & 0_{h,\ell} \\ 0_{h,\ell} & \sigma(P_1) \end{pmatrix} \qquad P'_2 = \begin{pmatrix} P_2 & 0_{k,\ell} \\ 0_{k,\ell} & \sigma(P_2) \end{pmatrix}.$$

We let

$$H = H_1 \otimes_{M, P'_1, P'_2} H_2,$$
$$N = \begin{pmatrix} \Gamma_{H_1} \cdot P'_1 \\ \Gamma_{H_2} \cdot P'_2 \end{pmatrix}.$$

Clearly (H, N, Z) is a presentation of $G[V]$ relative to G. This terminates the proof of the proposition.   ∎

Example 4.3 Let G be a directed graph. We let V_1 be the set $\{x_1, \ldots, x_6\}$, V_2 be the set $\{y_1, \ldots, y_6\}$ and $W = V_G - V$ be the set $\{z_1, \ldots, z_5\}$. We let $A = F_G$. The matrix $A_{V_1 \cup V_2}^{V_2 \cup W}$ is (we can leave the sub-matrix $A_{V_2}^{V_2}$ undefined):

$$A = $$

	y_1	y_2	y_3	y_4	y_5	y_6	z_1	z_2	z_3	z_4	z_5
x_1	a	a^2	1	a	0	0	a^2	a^2	a^2	0	a^2
x_2	0	0	0	a^2	a	1	1	1	0	a^2	1
x_3	a^2	0	a	1	0	a^2	a	a	0	1	a
x_4	a^2	1	a	0	a	1	0	0	1	a^2	0
x_5	a^2	0	a	a^2	1	0	1	1	0	a^2	1
x_6	0	1	0	0	1	0	0	0	1	a^2	0
y_1							0	1	a^2	a	a
y_2							0	0	a^2	a	0
y_3							0	a	1	a^2	a^2
y_4							1	1	1	0	1
y_5							a	a^2	a	0	0
y_6							a^2	0	1	1	a

We can verify that we have the following linear relations between rows of $A_{V_1}^{V_2 \cup W}$ and $A_{V_2}^{V_1 \cup W}$:

$$A_{x_4}^{V_2 \cup W} = a \cdot A_{x_1}^{V_2 \cup W} + A_{x_2}^{V_2 \cup W},$$
$$A_{x_5}^{V_2 \cup W} = a^2 \cdot A_{x_2}^{V_2 \cup W} + A_{x_3}^{V_2 \cup W},$$
$$A_{x_6}^{V_2 \cup W} = a \cdot A_{x_1}^{V_2 \cup W} + a^2 \cdot A_{x_2}^{V_2 \cup W} + A_{x_3}^{V_2 \cup W}$$

and,

$$A_{y_1}^{V_1 \cup W} = a \cdot A_{y_2}^{V_1 \cup W} + a \cdot A_{y_5}^{V_1 \cup W} + A_{y_6}^{V_1 \cup W},$$
$$A_{y_3}^{V_1 \cup W} = a^2 \cdot A_{y_2}^{V_1 \cup W} + a^2 \cdot A_{y_5}^{V_1 \cup W} + a \cdot A_{y_6}^{V_1 \cup W},$$
$$A_{y_4}^{V_1 \cup W} = a \cdot A_{y_2}^{V_1 \cup W} + a \cdot A_{y_5}^{V_1 \cup W} + a^2 \cdot A_{y_6}^{V_1 \cup W},$$
$$A_{y_7}^{V_1 \cup W} = a \cdot A_{y_5}^{V_1 \cup W}.$$

Vertex bases of $G[V_1]$ and of $G[V_2]$ with respect to G are respectively $\{x_1, x_2, x_3\}$ and $\{y_2, y_5, y_6\}$. The $GF(4)^6$-colorings H and K of $G[V_1]$ and of $G[V_2]$ are the same as in Example 4.2. Among $\{x_1, x_2, x_3, y_2, y_5, y_6\}$, one can select $\{x_3, y_2, y_5\}$ as a vertex basis of $G[V_1 \cup V_2]$ with respect to G. Then we have the following linear relations:

$$A_{x_1}^{W} = a \cdot A_{x_3}^{W} + A_{y_2}^{W},$$
$$A_{x_2}^{W} = a^2 \cdot A_{x_3}^{W},$$
$$A_{y_6}^{W} = A_{x_3}^{W} + a^2 \cdot A_{y_5}^{W}.$$

We have then

$$M = \begin{pmatrix} 0_{3,3} & M' \\ 0_{3,3} & 0_{3,3} \end{pmatrix} \qquad P_1' = \begin{pmatrix} P_1 & 0_{3,3} \\ 0_{3,3} & \sigma(P_1) \end{pmatrix} \qquad P_2' = \begin{pmatrix} P_2 & 0_{3,3} \\ 0_{3,3} & \sigma(P_2) \end{pmatrix}$$

where

$$M' = \begin{array}{c|c|c|c} & y_2 & y_5 & y_6 \\ \hline x_1 & a^2 & 0 & 0 \\ \hline x_2 & 0 & a & 1 \\ \hline x_3 & 0 & 0 & a^2 \end{array} \qquad P_1 = \begin{array}{c|c|c|c} & x_3 & y_2 & y_5 \\ \hline x_1 & a & 1 & 0 \\ \hline x_2 & a^2 & 0 & 0 \\ \hline x_3 & 0 & 0 & 1 \end{array} \qquad P_2 = \begin{array}{c|c|c|c} & x_3 & y_2 & y_5 \\ \hline y_2 & 0 & 1 & 0 \\ \hline y_5 & 0 & 0 & 1 \\ \hline y_6 & 1 & 0 & a^2 \end{array}$$

One can verify for an example that $\gamma_H(x_1) \cdot P_1' = \begin{pmatrix} a & 1 & 0 & a^2 & 1 & 0 \end{pmatrix}$ and $\gamma_K(y_6) \cdot P_2' = \begin{pmatrix} 1 & 0 & a^2 & 1 & 0 & a \end{pmatrix}$.

We can now prove the converse direction of Theorem 4.2.

Proposition 4.5 *Let G be a graph over (F, σ) that has F-rank-width at most n. Then G is the value of a term t where:*

$$t \in \begin{cases} T(R_n^F, C_n^F) & \text{if } F_G \text{ is symmetric,} \\ T(R_{2n}^F, C_{2n}^F) & \text{otherwise.} \end{cases}$$

Proof. We let G be such that $rwd^F(G) \leq n$. We first assume G to be connected.

Let $(T, \mathcal{L})$ be a layout of ρ_G^F of branch-width n; we can assume T cubic since $(Red(T), \mathcal{L})$ is also a layout of branch-width n of ρ_G^F. Let us select a node s of degree 1 of T as root, and direct T accordingly, from the root towards the other vertices of degree 1. The weight of a node u of T is the number of the nodes of $T \downarrow u$, the sub-tree of the directed tree T rooted at u.

For every arc of T of the form $\overrightarrow{vu}$, we let G_u be the induced subgraph of G, the vertices of which are the leaves of $T \downarrow u$, and G^v the induced subgraph of G, the vertices of which are the leaves not in $T \downarrow u$.

Claim 4.2 *One can choose for each u different from the root of T, a presentation (H_u, N_u, X_u) of G_u with $|X_u| = r(u)$ where $r(u)$ is the width of $\overrightarrow{vu}$, and a term t_u in $T(R_n^F, C_n^F)$ if F_G is symmetric, otherwise t_u is in $T(R_{2n}^F, C_{2n}^F)$, that defines H_u, such that if u has two sons w and w', then $t_u = t_w \otimes_{M, N_1, N_2} t_{w'}$ for some matrices M, N_1 and N_2.*

Proof of Claim 4.2. By induction on the weight $w(u)$ of u.

If $w(u) = 1$, then G_u is a singleton graph. Since G is assumed connected, $r(u) = 1$. We take $t_u = (\mathbf{1})$ if F_G is symmetric, otherwise $t_u = (\mathbf{1} \ \mathbf{1})$.

Let $w(u) \neq 1$. Then u has two sons w and w' and they have smaller weights than u. By inductive hypothesis, there exist presentations (H_w, N_w, X_w) of G_w and $(H_{w'}, N_{w'}, X_{w'})$ of $G_{w'}$.

By Proposition 4.4, there exist matrices M, N_1 and N_2 such that (H_u, N_u, X_u) is a presentation

of G_u where

$$H_u = H_w \otimes_{M,N_1,N_2} H_{w'}$$

$$N_u = \begin{pmatrix} \Gamma_{H_w} \cdot N_1 \\ \Gamma_{H_{w'}} \cdot N_2 \end{pmatrix},$$

$$X_u \subseteq X_w \cup X_{w'} \text{ is a vertex basis of } G_u.$$

If F_G is symmetric, M, N_1 and N_2 have order $(r(w), r(w'))$, $(r(w), r(u))$ and $(r(w'), r(u))$. Otherwise, they are matrices of order $(2 \cdot r(w), 2 \cdot r(w'))$, $(2 \cdot r(w), 2 \cdot r(u))$ and $(2 \cdot r(w'), 2 \cdot r(u))$.

By inductive hypothesis, there also exist t_w and $t_{w'}$ that define H_w and $H_{w'}$ respectively. We let then $t_u = t_w \otimes_{M,N_1,N_2} t_{w'}$. It is clear that t defines H_u. This completes the general case. ■

We can now finish the proof of the proposition. For the case of u where $r \to u$, r being the root of T, G is defined by $t_u \oplus_{M,0,0} t$ where t_u is obtained by Claim 4.2 and

$$t = \begin{cases} (\mathbf{1}) & \text{if } F_G \text{ is symmetric,} \\ (\mathbf{1\ 1}) & \text{otherwise.} \end{cases} \qquad M = \begin{cases} (F_G)_y^{\mathcal{L}^{-1}(r)} & \text{if } F_G \text{ is symmetric,} \\ \begin{pmatrix} 0 & (F_G)_y^{\mathcal{L}^{-1}(r)} \\ 0 & 0 \end{pmatrix} & \text{otherwise.} \end{cases}$$

where y is the vertex basis of $G \backslash \mathcal{L}^{-1}(r)$ with respect of G chosen by Claim 4.2.

If G is not connected, then $G = H_1 \oplus \cdots \oplus H_k$ where $H_1, \ldots, H_k$ are connected and $rwd(H_i) \leq n$ for each i. We let $t_1, \ldots, t_k$ be terms denoting $H_1, \ldots, H_k$ respectively. Then we can take $t_1 \otimes_{0,0,0} t_2 \otimes_{0,0,0} \cdots \otimes_{0,0,0} t_k$ to denote G and $\otimes_{0,0,0}$ is equivalent to $\oplus$, the disjoint union of uncolored graphs. This ends the proof. ■

As corollaries we get the following theorems.

Theorem 4.3 ([CK09]) *An undirected graph G has rank-width at most n if and only if G is the value of a term t in $T(R_n^{GF(2)}, C_n^{GF(2)})$.*

Theorem 4.4 ([Kan08]) *Let G be a directed graph. If G has $GF(4)$-rank-width at most n, then G is the value of a term t in $T(R_{2n}^{GF(4)}, C_{2n}^{GF(4)})$. If $G = val(t)$ for $t \in T(R_n^{GF(4)}, C_n^{GF(4)})$, then $rwd^{(4)}(G) \leq 2n$.*

Remark 4.5   1. Any layout $(T, \mathcal{L})$ of G of branch-width n is made into a term t such that $Red(red(t)) = T$.

2. The procedure that transforms a layout $(T, \mathcal{L})$ of width k of a graph G into a term t in $T(R_k^F, C_k^F)$ or $T(R_{2k}^F, C_{2k}^F)$ can be done in time $O(|V_G|^2)$. Our algorithm consists in transforming each internal node u of T that partitions V_G into (V_1, V_2, W) into a binary operation $\otimes_{M,N,P}$. For that purpose, we need a vertex basis B_1 of $(F_G)_{V_1}^{V_G - V_1}$, a vertex basis B_2 of $(F_G)_{V_2}^{V_G - V_2}$ and a vertex basis B_3 of $(F_G)_{V_1 \cup V_2}^{W}$. By Proposition 4.5, we can construct a vertex basis of $(F_G)_{V_1 \cup V_2}^{W}$ by using a basis of $(F_G)_{B_1 \cup B_2}^{W}$, which is a $(2k \times |W|)$-matrix.

By using Gauss pivot algorithm [Lip91], we can find a basis of $(F_G)^W_{B_1 \cup B_2}$ in $O(k^2 \cdot |V_G|)$-time. We can then construct for each internal node of T, the matrices M, N and P in $O(k^2 \cdot |V_G|)$-time, because for each vertex x of G, the basis of $(F_G)^{V_G - \{x\}}_x$ is $\{x\}$ (G is connected). Since $|T| = O(|V_G|)$, we can construct the term t in time $O(|V_G|^2)$.

It is clear that if $t \in T(R^F_n, C^F_n)$, then $t \in T(B^F_n, C^F_n)$. We will prove in the following proposition that if $t \in T(B^F_n, C^F_n)$, then $t \in T(R^F_{|F|^n}, C^F_{|F|^n})$.

Proposition 4.6 *Let $t \in T(B^F_n, C^F_n)$, then $t \in T(R^F_{q^n}, C^F_{q^n})$ where $F = \{a_1, \ldots, a_q\}$.*

Proof. Let $t \in T(B^F_n, C^F_n)$ and without loss of generality we may assume that all operations f, g, h are defined within F^n. Let $\alpha : F^n \to [q^n]$ be a bijective function that enumerates the set of vectors in F^n. For each $u \in F^n$, we let $\dot{u} \in F^{q^n}$ with $\dot{u}[\alpha(u)] = 1$ and $\dot{u}[i] = 0$ for $i \neq \alpha(u)$. We construct an expression $t' \in T(R^F_{q^n}, C^F_{q^n})$ with these rules:

(i) if $t = \mathbf{u}$, we let $t' = \dot{\mathbf{u}}$,

(ii) if $t = Recol_h(t_1)$, we let $t' = Recol_N(t'_1)$ for some $(q^n \times q^n)$-matrix N, defined below,

(iii) if $t = t_1 \otimes_{f,g,h} t_2$, we let $t' = t'_1 \otimes_{M,N,P} t'_2$ for some $(q^n \times q^n)$-matrices M, N, P, defined below.

For Rule (ii), we let N be a $(q^n \times q^n)$-matrix such that if $h(u) = v$, then $N^{\alpha(v)}_{\alpha(u)} = 1$ and $N^{\alpha(w)}_{\alpha(u)} = 0$ for $w \neq v$. It is easy to verify that $\dot{v} = \dot{u} \cdot N$ where $v = h(u)$ for each $u \in F^n$.

For Rule (iii), we construct g and h as in h in the case of Rule (ii). To express f as a bilinear form, we let M be the $(q^n \times q^n)$-matrix such that $M^{\alpha(v)}_{\alpha(u)} = f(u, v)$ for all u, v. It is straightforward to verify that $f(u, v) = \dot{u} \cdot M \cdot \dot{v}^T$. ∎

Question 4.1 For each term $t \in T(B^F_n, C^F_n)$ can we have $t \in T(R^F_{p(n)}, C^F_{p(n)})$ for some polynomial p?

Remark 4.6 It seems better for verification purposes to use the operations in B^F_n for the following reason. The graph operations are defined for colored graphs. In order to use logical tools explained in [CMR00, DF99, FG06, Mak04], we represent the colors by unary relations and the graph operations are represented by quantifier-free operations over the signature formed with the adjacency relation and the unary relations used to describe the colored graphs. For instance, if F is a finite field, Definition 4.5 shows how to represent the colors of F^k-colored graphs and Proposition 4.1 shows how to code operations in B^F_n by quantifier-free operations. In the linear-time algorithms based on the methodology presented in [CMR00, DF99, FG06, Mak04], the hidden constants depend on the number of relations used to describe colored graphs. Hence, using less relations yields smaller (although large) constants. However, concrete implementations should be done to decide. One can use for instance MONA [HJJ+95]. Moreover, practical experience, although insufficient, shows that this is not the only point to take into account. The PhD thesis of Soguet [Sog08] presents some trials of concrete implementations.

4.3 Algebraic Operations for Bi-Rank-Width

In this section we will define graph operations that handle algebraically undirected edge-colored graphs that are not necessarily over some (F, σ) for some finite field F and automorphism $\sigma : F \to F$. We recall that if A is a finite set, we do not assume any structure on A, then an undirected A-edge-colored graph G is represented by the structure $\langle V_G, (E_G^a)_{a \in A} \rangle$. For an arbitrary edge-colored graph G, we cannot define an adjacency matrix because, for every pair of vertices (x, y), we can have two or more colors, say a and b in A and such that $E_G^a(x, y)$ and $E_G^b(x, y)$ holds.

For any finite set A, we can assume that it is linearly ordered by a relation $<_A$, by fixing any linear relation. Therefore, to any finite set A corresponds a bijection $\alpha_A : A \to |A|$ such that for every a, b in A we have $\alpha_A(a) < \alpha_A(b)$ if and only if $a <_A b$. We now extend the operations in R_n^F as follows.

Definition 4.9 Let A be a finite set and let α_A be the bijection induced by the linear relation on A and let $p = |A|$. Let $M_1, \ldots, M_p$ be $(k \times \ell)$-matrices, N and P be respectively $(k \times m)$ and $(\ell \times m)$-matrices, all over $GF(2)$. For undirected A-edge-colored graphs G, $GF(2)^k$-colored and H, $GF(2)^\ell$-colored, we let $K = G \otimes_{M_1, \ldots, M_p, N, P} H$ be the undirected $GF(2)^m$-colored A-edge-colored graph $\langle V_G \cup V_H, (E_K^a)_{a \in A}, \gamma_K \rangle$ where as usual we assume $V_G \cap V_H = \emptyset$ and for $a \in A$:

$$E_K^a = E_G^a \cup E_H^a \cup \{(x, y), (y, x) \mid \gamma_G(x) \cdot M_{\alpha_A(a)} \cdot \gamma_H(y)^T = 1\},$$

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

We let $U_n^{([p])}$ be the set of binary operations $\otimes_{M_1, \ldots, M_p, N, P}$ where the matrices M_i are $(k \times \ell)$-matrices and the matrices N, P are $(k \times \ell)$-matrices and $(\ell \times m)$-matrices for $k, \ell, m \leq n$. We obtain in this way a complexity measure on A-edge-colored graphs:

$$Rwd^{(A)}(G) = \min\{k \mid G = val(t), \ t \in T(U_k^{([p])}, C_k^{GF(2)})\}.$$

We recall that if G is an A-edge colored graph, then we denote by G_a the undirected graph $\langle V_G, E_G^a \rangle$, i.e., we have an edge xy in E_{G_a} if and only if (x, y) is in E_G^a. It is clear that if $G = val(t)$ for some term t in $T(U_k^{([p])}, C_k^{GF(2)})$ and $a \in A$, then $G_a = val(t_a)$ where $t_a \in T(R_k^{GF(2)}, C_k^{GF(2)})$ is obtained from t by replacing each operation $\otimes_{M_1, \ldots, M_p, N, P}$ by $\otimes_{M_i, N, P}$ where $\alpha_A(a) = i$. It follows that $rwd(G_a) \leq Rwd^{(A)}(G)$ for each $a \in A$.

If G is an undirected A-edge-colored graph, we have defined in Definition 1.11 the notion of rank-width for G and denote it by $rwd_A(G)$. We now relate $rwd_A(G)$ and $Rwd^{(A)}(G)$. We recall that if t is a term in $T(F, C)$, we denote by $Occ_L(t)$ the finite set of occurrences of constants in t.

Proposition 4.7 *Let A be a set of p edge-colors. For every undirected A-edge-colored graph G, we have*

$$\frac{1}{p} Rwd^{(A)}(G) \leq rwd_A(G) \leq Rwd^{(A)}(G).$$

Proof. Let $t \in T(U_k^{([p])}, C_k^{GF(2)})$ be a term defining G. Let $\mathcal{L}$ be the bijection between V_G and $Occ_L(t)$. We let $(T = red(t), \mathcal{L})$ be a layout of G. Clearly for each a in A, $(T, \mathcal{L})$ is also a layout of ρ_{G_a}. Hence, for each a, $bwd(\rho_{G_a}, T, \mathcal{L})$ is at most k since $t_a \in T(R_k^{GF(2)}, C_k^{GF(2)})$. Hence, $rwd(G) \leq \max_{a \in A}\{bwd(\rho_{G_a}, T, \mathcal{L})\} \leq k$, which proves that $rwd_A(G) \leq Rwd^{(A)}(G)$.

For the inequality $Rwd^{(A)}(G) \leq p \cdot rwd_A(G)$, let us consider a layout $(T, \mathcal{L})$ of G that witnesses $rwd_A(G) = k$. Hence, $bwd(\rho_{G_a}, T, \mathcal{L})$ is at most k for each $a \in A$. For each $a \in A$, there exists by Proposition 4.5 a term $t_a \in T(R_k^{GF(2)}, C_k^{GF(2)})$ that defines G_a. For any $a, b \in A$, we have $red(t_a) = red(t_b)$. To simplify the proof we assume that $A = \{a, b\}$ and $a < b$, i.e., $\alpha_A(a) = 1$, $\alpha_A(b) = 2$. The extension to $p > 2$ will be straightforward.

We now show how t_a and t_b can be merged into a single term (Example 4.4 illustrates it). We need a claim with an easy proof but numerous assumptions. Let $G = G_a \cup G_b$ and $H = H_a \cup H_b$. Assume that G_a and G_b have color matrices $\Gamma_{G_a}, \Gamma_{G_b}$, both of order $|V_G| \times k$. Assume that H_a and H_b have similar matrices $\Gamma_{H_a}, \Gamma_{H_b}$.

Assume now that a graph K is obtained from G and H by taking their disjoint union and adding a-edges and b-edges. We assume that the a-edges are added by the operation $G_a \otimes_{M_a, N_a, P_a} H_a$ giving K_a which also involves recolorings by N_a, P_a. Assume that the b-edges are added by $G_b \otimes_{M_b, N_b, P_b} H_b$ giving K_b.

Let us define for G the color-matrix $\Gamma_G = \begin{pmatrix} \Gamma_{G_a} & \Gamma_{G_b} \end{pmatrix}$ of order $|V_G| \times 2k$, and similarly $\Gamma_H = \begin{pmatrix} \Gamma_{H_a} & \Gamma_{H_b} \end{pmatrix}$ of order $|V_H| \times 2k$. We let finally $\Gamma_K = \begin{pmatrix} \Gamma_{K_a} & \Gamma_{K_b} \end{pmatrix}$ of order $(|V_G| + |V_H|) \times 2k$. Then we have:

Claim 4.3 $K = G \otimes_{M, M', N, P} H$ where

$$M = \begin{pmatrix} M_a & 0 \\ 0 & 0 \end{pmatrix} \qquad M' = \begin{pmatrix} 0 & 0 \\ 0 & M_b \end{pmatrix} \qquad N = \begin{pmatrix} N_a & 0 \\ 0 & N_b \end{pmatrix} \qquad P = \begin{pmatrix} P_a & 0 \\ 0 & P_b \end{pmatrix}$$

Proof of Claim 4.3. The verification is routine from the numerous assumptions. ∎

By using the fact that t_a and t_b have the same "shape", i.e., $red(t_a) = red(t_b)$, their operations can be merged by the above claim so as to form a single term t in $T(U_{2k}^{([p])}, C_{2k}^{GF(2)})$ that defines G. Note that $red(t) = red(t_a)$. ∎

Example 4.4 Let $A = \{a, b\}$ with $a < b$. An undirected A-edge-colored graph G is shown on Figure 7 (iii). The graphs G_a and G_b and, a layout $(T, \mathcal{L})$ of G are shown on Figure 7 (i), (ii) and (iv). $(T, \mathcal{L})$ is a layout of branch-width 2 for ρ_{G_a} and ρ_{G_b}. The terms t_a and t_b that constructs respectively G_a and G_b are shown on Figure 15 (i) and (ii) where, for every i in [5], we have

$f_i = \otimes_{M_{i,a},N_{i,a},P_{i,a}}$ and $g_i = \otimes_{M_{i,b},N_{i,b},P_{i,b}}$ with:

	$M_{i,a}$	$N_{i,a}$	$P_{i,a}$
f_1	(0)	$(1\ 0)$	$(0\ 1)$
f_2	(0)	$(0\ 1)$	$(1\ 0)$
f_3	$(1\ 1)$	$(1\ 0)$	$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$
f_4	(0)	$(0\ 1)$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
f_5	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

	$M_{i,b}$	$N_{i,b}$	$P_{i,b}$
g_1	(0)	(0)	(1)
g_2	(0)	$(1\ 0)$	$(0\ 1)$
g_3	(0)	$(1\ 0)$	$(0\ 1)$
g_4	$(0\ 1)$	(0)	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
g_5	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

For each i in $[5]$, we let $h_i = \otimes_{M_i,M_i',N_i,P_i}$ where

	M_i	M_i'	N_i	P_i
h_1	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
h_2	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
h_3	$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
h_4	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
h_5	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

One can verify that t effectively constructs the undirected A-edge-colored graph G.

We now explain how to consider a term in $T(U_k^{([2])}, C_k^{GF(2)})$ as one defining a directed graph. For directed graphs G, $GF(2)^n$-colored and H, $GF(2)^\ell$-colored, we let $K = G \otimes_{M_1,M_2,N,P} H$ be the directed graph $\langle V_G \cup V_H, E_K, \gamma_K \rangle$ where:

$$E_K = E_G \cup E_H \cup \{(x,y) \mid x \in V_G,\ y \in V_H \text{ and } \gamma_G(x) \cdot M_1 \cdot \gamma_H(y) = 1\} \cup$$
$$\{(y,x) \mid x \in V_G,\ y \in V_H \text{ and } \gamma_G(x) \cdot M_2 \cdot \gamma_H(y) = 1\}$$
$$\gamma_K(x) = \begin{cases} \gamma_G(x) \cdot N & \text{if } x \in V_G, \\ \gamma_H(x) \cdot P & \text{otherwise.} \end{cases}$$

By using the construction above, we can inductively construct a directed graph $val(t)$ from a term t in $T(U_k^{([2])}, C_k^{GF(2)})$. Then a term t in $T(U_k^{([2])}, C_k^{GF(2)})$ can be considered as one defining a directed graph or one defining an undirected A-edge-colored graph, depending on the used algebra. In the rest of this section, we will consider terms in $T(U_k^{([2])}, C_k^{GF(2)})$ as terms that define directed graphs.
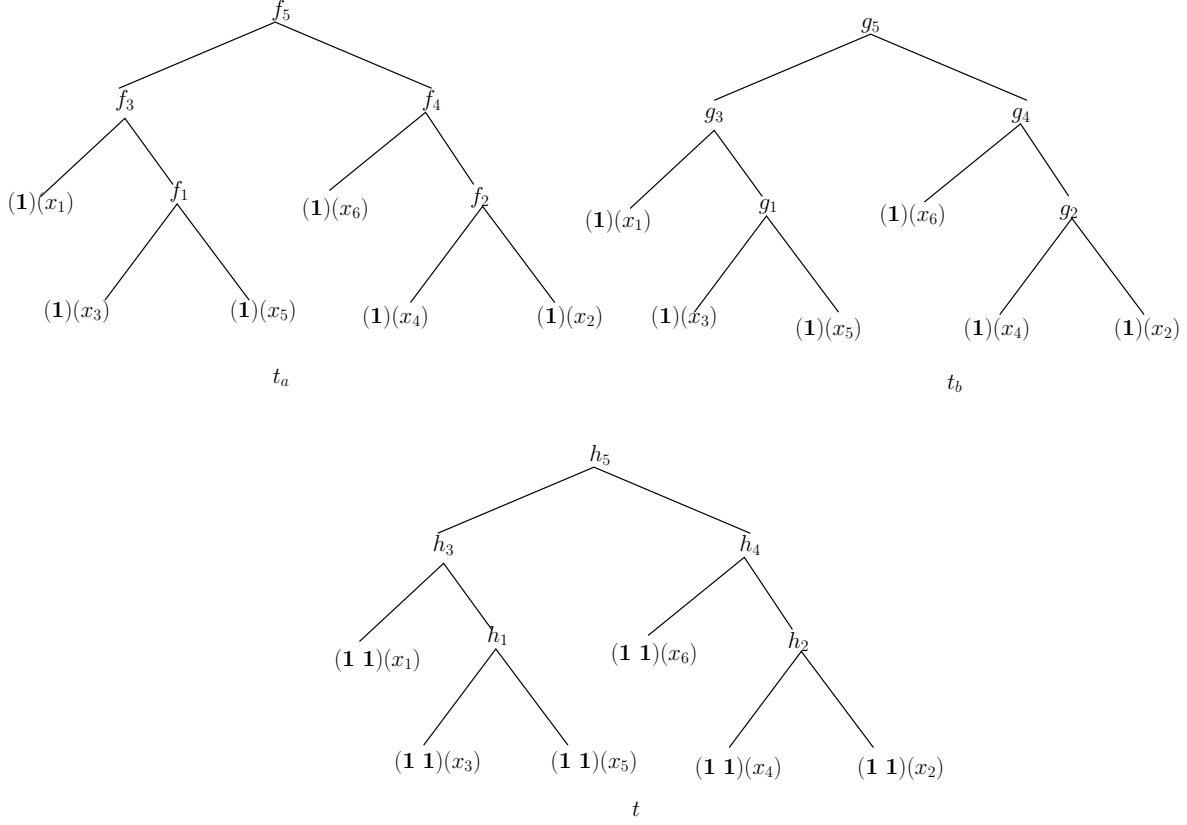
Figure 15: A term t that defines the A-edge-colored graph G on Figure 7 (iii).

We now prove that the operations in $U_k^{([2])}$ handle algebraically graphs of bi-rank-width at most k. We recall that if $(T, \mathcal{L})$ is a rooted layout and $e = (u, v)$, then $X^e = \mathcal{L}^{-1}(N_{T\downarrow v}^{(1)})$ where $T \downarrow v$ is the the sub-tree of T rooted at v induced by the set of all descendants of v (Chapter 1).

Proposition 4.8 *Let G be a graph of bi-rank-width k. Then there exists a term t in $T(U_k^{([2])}, C_k^{GF(2)})$ such that G is isomorphic to $val(t)$.*

Proof. Let $(T, \mathcal{L})$ be a layout of branch-width k of $\rho_G^{(bi)}$. We can assume that each node of T has degree 1 or 3 since $(Red(T), \mathcal{L})$ is also a layout of $\rho_G^{(bi)}$ of branch-width k. We take a node r of degree 1 as root and direct T such that each node of degree 1 is reachable from r. Hence, T defines a linear order $\leq$ on V_G. We define the two undirected graphs $G_a = \langle V_G, E_{G_a} \rangle$ and $G_b = \langle V_G, E_{G_b} \rangle$ where :

$$E_{G_a} = \{xy \mid x < y \text{ and } (x, y) \in E_G\},$$
$$E_{G_b} = \{yx \mid x < y \text{ and } (y, x) \in E_G\}.$$

It is clear that (E_{G_a}, E_{G_b}) is a bipartition of E_G. Moreover, $(T, \mathcal{L})$ is a layout of ρ_{G_a} of branch-width p where $p = \max_{e \in E_T}\{\rho_{G_a}(X^e)\}$. It is also a layout of ρ_{G_b} of branch-width q where $q = \max_{e \in E_T}\{\rho_{G_b}(X^e)\}$. We recall that $p + q = k$.

By Theorem 4.2, we can construct a term t_a in $T(R_p^{GF(2)}, C_p^{GF(2)})$ that defines G_a and similarly a term t_b in $T(R_q^{GF(2)}, C_q^{GF(2)})$ defining G_b. It is straightforward to see that $red(t_a) = red(t_b)$. By using the same ideas as in Claim 4.3, one can merge the terms t_a and t_b in order to construct a term $t \in T(U_k^{([2])}, C_k^{GF(2)})$ defining a graph isomorphic to G. ∎

We now prove a kind of converse.

Proposition 4.9 *Let* G *be a directed graph isomorphic to* $val(t)$ *where* t *is in* $T(U_k^{([2])}, C_k^{GF(2)})$. *Then* $brwd(G) \leq 2 \cdot k$.

Proof. Let $t \in T(U_k^{([2])}, C_k^{GF(2)})$ be such that G is isomorphic to $val(t)$. The term t defines a linear order $\leq$ on V_G because there exits a bijection $\mathcal{L}$ between V_G and $Occ_L(t)$ which is naturally ordered. We define the two undirected graphs $G_a = \langle V_G, E_{G_a} \rangle$ and $G_b = \langle V_G, E_{G_b} \rangle$ where :

$$E_{G_a} = \{xy \mid x < y \text{ and } (x, y) \in E_G\},$$
$$E_{G_b} = \{yx \mid x < y \text{ and } (y, x) \in E_G\}.$$

By definition (E_{G_a}, E_{G_b}) is a bipartition of E_G. We let t_a be t where we replace each operation $\otimes_{M_1, M_2, N, P}$ by $\otimes_{M_1, N, P}$ and t_b be t where we replace each operation $\otimes_{M, M_2, N, P}$ by $\otimes_{M_2, N, P}$. By the definitions of t_a, t_b, G_a and G_b it is clear that $val(t_a) = G_a$ and $val(t_b) = G_b$. We also have $red(t) = red(t_a)$ and $red(t) = red(t_b)$. Let this term be denoted by T. We claim that $(T, \mathcal{L})$ is a layout of branch-width at most $2k$ of $\rho_G^{(bi)}$. It is sufficient to prove, for each sub-term t' of t, that $\rho_G^{(bi)}(V_H) \leq 2k$ where H is isomorphic to $val(t')$.

Let t' be a subterm of t with H isomorphic to $val(t')$. By Lemma 4.1, $(A_{G_a})_{V_H}^{V_G - V_H} = \Gamma_H \cdot B$ and $(A_{G_b})_{V_H}^{V_G - V_H} = \Gamma_H \cdot B'$. But, by the definitions of G_a and of G_b, we also have $\rho_{G_a}(V_H) = rk\left((A_G^+)_{V_H}^{V_G - V_H}\right)$ and $\rho_{G_b}(V_H) = rk\left((A_G^-)_{V_H}^{V_G - V_H}\right)$. Hence,

$$\rho_G^{(bi)}(V_H) \leq rk(\Gamma_H \cdot B) + rk(\Gamma_H \cdot B')$$
$$\leq 2 \cdot rk(\Gamma_H) \leq 2k$$

since H is $GF(2)^\ell$-colored, $\ell \leq k$ and $rk(A \cdot B) \leq \min\{rk(A), rk(B)\}$. ∎

4.4 Conclusion

Let F be a finite field and let $\sigma : F \to F$ be an automorphism. We have proposed algebraic graph operations that handle algebraically graph classes over (F, σ) that have bounded F-rank-width. These graph operations are quantifier-free and then we can use the logical framework of [Cou92, CMR00, Mak04] to solve MS-definable problems on graphs generated by such operations. A specialization of these operations characterize exactly the rank-width of undirected graphs (Theorem 4.3). In the case of directed graphs, these operations do not

yield an exact algebraic characterization of $GF(4)$-rank-width (Theorem 4.4). We have also defined graph operations that handle algebraically directed graphs of bi-rank-width at most k (Propositions 4.8 and 4.9). It is open to find graph operations that characterize exactly directed graphs of $GF(4)$-rank-width at most k as follows:

> a directed graph has $GF(4)$-rank-width at most k if and only if $G = val(t)$ for some term t in $T(R_k^{(4)}, C_k^{(4)})$

where $R_k^{(4)}$ is a finite set of graph operations, $C_k^{(4)}$ is a finite set of constants, both depending on k.

Chapter 5

Recognition Algorithms

We recall that checking an upper bound to the rank-width of an undirected graph is NP-complete [HOSG08] and for fixed k, there exists a cubic-time algorithm that recognizes the undirected graphs of rank-width at most k (see Section 1.2, Theorem 1.2). In this chapter we prove that for fixed k, we can check if a directed graph has bi-rank-width or $GF(4)$-rank-width at most k. In Section 5.1 we introduce some auxiliary graph operations and compare clique-width, $GF(4)$-rank-width and bi-rank-width. We give the recognition algorithms for bi-rank-width and $GF(4)$-rank-width in Section 5.2.

5.1 Other Width Parameters and Comparisons

Our objective in this section is to prove the following proposition.

Proposition 5.1 *For every directed graph G,*

1. *$\frac{1}{2} brwd(G) \leq cwd(G) \leq 2^{brwd(G)+1} - 1$.*

2. *$rwd^{(4)}(G) \leq cwd(G) \leq 2 \cdot 4^{rwd^{(4)}(G)} - 1$.*

Before proving it, we first define some derived graph operations built from operations defining m-clique-width. We also prove some basic preliminary results. We recall that a C-colored graph $G = \langle V_G, E_G \rangle$ is defined as $G = \langle V_G, E_G, lab_G \rangle$ where $lab_G(x) \in C$ for each $x \in V_G$.

Definition 5.1 (Derived M-Clique-Width Operations Defining Undirected Graphs)
Let k be a positive integer. For $R \subseteq [k] \times [k]$, for mappings $g, h : [k] \to [k]$ and for undirected $[k]$-colored graphs G and H, we define the undirected $[k]$-colored graph $K = G \otimes_{R,g,h} H$ if G and H are disjoint (otherwise we replace H by a disjoint copy) where

$$V_K = V_G \cup V_H,$$
$$E_K = E_G \cup E_H \cup \{xy \mid x \in V_G, y \in V_H \text{ and } (lab_G(x), lab_H(y)) \in R\},$$
$$lab_K(x) = \begin{cases} g(lab_G(x)) & \text{if } x \in V_G, \\ h(lab_H(x)) & \text{if } x \in V_H. \end{cases}$$

We denote by F_k^u the set of all operations $\otimes_{R,g,h}$ where $R \subseteq [k] \times [k]$ and $g, h : [k] \to [k]$. Every term t in $T(F_k^u, C_k^c)$ denotes, up to isomorphism, an undirected graph $val(t)$.

Definition 5.2 (Derived M-Clique-Width Operations Defining Directed Graphs)
Let k be a positive integer. For $R \subseteq [k] \times [k]$, $R' \subseteq [k] \times [k]$, for mappings $g, h : [k] \to [k]$ and for directed $[k]$-colored graphs G and H, we define the directed $[k]$-colored graph $K = G \otimes_{R,R',g,h} H$ if G and H are disjoint (otherwise we replace H by a disjoint copy) where

$$V_K = V_G \cup V_H,$$
$$E_K = E_G \cup E_H \cup \{(x,y) \mid x \in V_G, y \in V_H \text{ and } (lab_G(x), lab_H(y)) \in R\}$$
$$\cup \{(y,x) \mid x \in V_G, y \in V_H \text{ and } (lab_G(x), lab_H(y)) \in R'\},$$
$$lab_K(x) = \begin{cases} g(lab_G(x)) & \text{if } x \in V_G, \\ h(lab_H(x)) & \text{if } x \in V_H. \end{cases}$$

We let F_k^d be the set of all binary operations $\otimes_{R,R',g,h}$ where $R, R' \subseteq [k] \times [k]$ and $g, h : [k] \to [k]$. Every term t in $T(F_k^d, C_k^c)$ denotes, up to isomorphism, a directed graph $val(t)$.

We introduce a last notation. Let $G = \langle V_G, E_G \rangle$ be a graph. Let $r \in T(\{*\}, \{\#\})$ and $\mathcal{L} : V_G \to Occ_L(r)$ be a bijection between V_G and $Occ_L(r)$. As in Definition 1.6, for each arc $e = (u,v)$ of r, we let X_e be the set $N_{T\downarrow v}^{(1)}$. For each edge e of r we let $Ind_G(e)$ be the cardinality of $\mathcal{L}^{-1}(X_e)/\sim_e$ where $\sim_e$ is the equivalence relation on $Y_e = \mathcal{L}^{-1}(X_e)$ defined by

$$x \sim_e y \qquad \text{if and only if} \qquad \forall z \in \overline{Y_e} \; \big(xz \in E_G \Longleftrightarrow yz \in E_G\big)$$

if G is undirected and, if G is directed, $\sim_e$ is defined by

$$x \sim_e y \qquad \text{if and only if} \qquad \forall z \in \overline{Y_e} \; \Big((x,z) \in E_G \Longleftrightarrow (y,z) \in E_G\Big)$$
$$\wedge \Big((z,x) \in E_G \Longleftrightarrow (z,y) \in E_G\Big)$$

In a way, if G is undirected and $Ind_G(e) = k$, this will induce a coloring of X_e that can be used in order to construct inductively a term in $T(F_k^u, C_k^c)$. We state this construction in the following lemma, which has an easy proof by induction.

Lemma 5.1 *Let G be an undirected graph and let r be a term in $T(\{*\}, \{\#\})$ given with a bijection between V_G and $Occ_L(r)$. If for each edge e of r, we have $Ind_G(e) \leq k$, then G is defined by a term t in $T(F_k^u, C_k^c)$ such that $red(t) = r$.*

By Lemma 5.1, if an undirected graph G has clique-width at most k, then it is the value of a term in $T(F_k^u, C_k^c)$. We will prove a similar result for directed graphs. Let us first give the main ideas. If a directed graph G has clique-width at most k, then $G = val(t)$ for some term t in $T(F_k^{dc}, C_k^c)$. This term t defines a left-right order $\leq$ on V_G which is a linear order. The idea is to partition the edges of G into two parts E_1 and E_2 such that $(x,y) \in E_1$ if and only if $x \leq y$ and $(x,y) \in E_G$ and, $(x,y) \in E_2$ if and only if $x \leq y$ and $(y,x) \in E_G$. For each $i = 1, 2$, we let G_i be the undirected graph obtained from $G[E_i]$ by forgetting the orientations of the edges. We then use Lemma 5.1 with input $red(t)$ to construct a term t_i in $T(F_k^u, C_k^c)$

and such that $G_i = val(t_i)$. We finally "glue" the two terms t_1 and t_2 in order to construct a term $\tilde{t}$ in $T(F_k^d, C_k^c)$ such that $red(\tilde{t}) = red(t_1) = red(t_2) = red(t)$. A similar idea was used to prove Proposition 4.7.

Lemma 5.2 *Let G be a directed graph. If $cwd(G) \leq k$, then $G = val(t)$ for some term t in $T(F_k^d, C_k^c)$.*

Proof of Lemma 5.2. Let t be a term in $T(F_k^{dc}, C_k^c)$ that defines G. The term t defines a linear order $\leq$ on V_G because V_G is in bijection with $Occ_L(t)$ and the occurrences of constants have a natural left-right order. We denote by $\mathcal{L}$ this bijection. We define the two undirected graphs $G^+ = \langle V_G, E_{G^+} \rangle$ and $G^- = \langle V_G, E_{G^-} \rangle$ where :

$$E_{G^+} = \{xy \mid x < y \text{ and } (x,y) \in E_G\},$$
$$E_{G^-} = \{yx \mid x < y \text{ and } (y,x) \in E_G\}.$$

Informally, for any $x, y \in V_G$, $xy \in E_{G^+}$ if and only if $x < y$ and $(x,y) \in E_G$ and $yx \in E_{G^-}$ if and only if $x < y$ and $(y,x) \in E_G$. Let $T = red(t)$. Then for each edge e of T, we have $Ind_{G^+}(e) \leq k$ and $Ind_{G^-}(e) \leq k$ since $Ind_G(e) \leq k$.

By Lemma 5.1, there exist expressions t^+ and t^- in $T(F_k^u, C_k^c)$ such that $G^+ = val(t^+)$ and $G^- = val(t^-)$ such that $red(t^+) = T$ and $red(t^-) = T$. With the help of the proof of Lemma 2.1, we can ensure that for each vertex $x \in V_G$, if $\mathcal{L}(x) = \mathbf{i}$ in t, then $\mathcal{L}(x) = \mathbf{i}$ in t^+ and in t^-. We can also ensure that a node u in T is labeled by $\otimes_{R,g,h}$ in t^+ if and only if it is labeled by some $\otimes_{R',g,h}$ in t^- with same mappings g, h. We then construct an expression $\tilde{t} \in T(F_k^d, C_k^c)$ from t^+ and t^- as follows:

$$\tilde{t} = \begin{cases} \mathbf{i} & \text{if } t^+ = t^- = \mathbf{i}, \\ \widetilde{t_1} \otimes_{R,R',g,h} \widetilde{t_2} & \text{if } t^+ = t_1^+ \otimes_{R,g,h} t_2^+ \text{ and } t^- = t_1^- \otimes_{R',g,h} t_2^-. \end{cases}$$

It is a straightforward induction to prove that $\tilde{t}$ effectively defines G. $\blacksquare$

We now explain how to prove Proposition 5.1. By Proposition 4.8, if a directed graph G has bi-rank-width at most k, then G is isomorphic to $val(t)$ for some term t in $T(U_k^{[2]}, C_k^{GF(2)})$. By Proposition 4.9, if G is a directed graph isomorphic to $val(t)$ where t is in $T(U_k^{([2])}, C_k^{GF(2)})$, then the bi-rank-width of G is at most $2k$. For proving Proposition 5.1 (1) we prove the followings.

(i) If $G = val(t)$ where $t \in T(F_k^{dc}, C_k^c)$, then $G = val(\tilde{t})$ for some term $\tilde{t}$ in $T(U_k^{([2])}, C_k^{GF(2)})$.

(ii) If $G = val(t)$ where $t \in T(U_k^{([2])}, C_k^{GF(2)})$, then $G = val(\tilde{t})$ for some term $\tilde{t}$ in $T(F_{k'}^c, C_{k'}^c)$, $k' = 2^{k+1} - 1$.

In order to prove (i) we use Lemma 5.2 which transforms a term t in $T(F_k^{dc}, C_k^c)$ into a term $\tilde{t}$ in $T(F_k^d, C_k^d)$. The idea is to show how to construct a term $\hat{t}$ in $T(U_k^{([2])}, C_k^{GF(2)})$ from $\tilde{t}$. For that purposes, we transform each color $i \in [k]$ into a color $\Omega(i) \in GF(2)^k$ and each

operation $\otimes_{R,R',g,h}$ into an operation $\otimes_{M,M',N,P}$ such that, for every i and j in $[k]$, we have (i,j) is in R (resp. $(i,j) \in R'$) if and only if $\Omega(i) \cdot M \cdot \Omega(j)^T = 1$ (resp. $\Omega(i) \cdot M' \cdot \Omega(j)^T = 1$). For the inductive construction to work, we must also guarantee that, for each i in $[k]$, we have $\Omega(g(i)) = \Omega(i) \cdot N$ and similarly $\Omega(h(i)) = \Omega(i) \cdot P$.

In order to prove (ii), we transform each color $u \in GF(2)^k$ into a color $\beta(u)$ where $\beta : GF(2)^k \to [2^k]$ is a bijective function. As in (i), we construct inductively a term $\tilde{t}$ in $T(F_{k'}^{dc}, C_k^c)$ from t in $T(U_k^{([2])}, C_k^{GF(2)})$ by showing how to transform each operation $\otimes_{M,M',N,P}$ into operations using clique-width operations. Let us explain the main idea of the induction and assume $t = t_1 \otimes_{M,M',N,P} t_2$; we let $H_1 = val(t_1)$ and $H_2 = val(t_2)$. The first step consists in replacing each color u in H_1 and H_2 by the color $\beta(u)$ and we denote the obtained graphs by $\widetilde{H_1}$ and $\widetilde{H_2}$. The idea is to replace t by $\tilde{t} = h(g(\alpha_R(\widetilde{H_1} \oplus \widetilde{H_2})))$ where α_R is a derived clique-width operation that adds the arcs between H_1 and H_2 by using the α operation of clique-width and g renames the colors in $\widetilde{H_1}$ and h renames the colors in $\widetilde{H_2}$ by using the renaming operation of clique-width and such that $g(\beta(u)) = \beta(u \cdot N))$ and similarly $h(\beta(v)) = \beta(v \cdot P)$. We assume the arcs in H_1 and in H_2 are already constructed. We now give the derived operations:

- We let α_R be the combination of operations $\alpha_{i,j}$ where $i = \beta(u)$, $j = \beta(v)$ and $u \cdot M \cdot v^T = 1$ or $u \cdot M' \cdot v^T = 1$. However, we may have $u = v$ and then $i = j$. To overcome this difficulty, we use the isomorphic copy $[2^k]'$ of $[2^k]$ and replace each color i in $\widetilde{H_2}$, by i'; α_R becomes the combination of operations $\alpha_{i,j'}$ where $i = \beta(u)$, $j = \beta(v)$ and $u \cdot M \cdot v^T = 1$ and of the operations $\alpha_{j',i}$ where $i = \beta(u)$, $j = \beta(v)$ and $u \cdot M' \cdot v^T = 1$.

- We want now to recolor the vertices in $\widetilde{H_1}$ and $\widetilde{H_2}$. If we recolor the vertices of $\widetilde{H_2}$ first, we could change their colors when recoloring the vertices of $\widetilde{H_1}$. For that purposes, we recolor them last. Since we want to rename a color i' into j and we know that we will not rename the color j (the vertices of $\widetilde{H_1}$ are already recolored), we can use a combination of the operations $\rho_{i' \to j}$ where $\beta(u) = i, \beta(v) = j$ and $u \cdot P = v$ in order to recolor the vertices of $\widetilde{H_2}$.

  But, the recoloring of the vertices in $\widetilde{H_1}$ can cause some difficulties. Indeed, for some u in $GF(2)^k$, we may have $u \cdot N = v$ and $v \cdot N = u$ (similarly for P but, since we rename colors from L' to L, this causes no difficulties). Then in order to rename $\beta(u)$ to $\beta(v)$ and $\beta(v)$ to $\beta(u)$ by using the unary operation ρ, we may need an auxiliary color, hence we will use more than k' colors. In order to overcome this second difficulty, we will rename the colors in a suitable way that ensures that we no more need any auxiliary color for the recoloring. We now show how to do that. We distinguish two cases, when g is bijective (Remark 5.1) and when it is not (Lemma 5.3).

Let L be a finite set. For a recoloring $g : L \to L$, we denote by ρ_g the operation that renames every color $i \in L$ into the color $g(i) \in L$.

Remark 5.1 Let L be a finite set of colors. Every operation ρ_g for a bijection of $g : L \to L$ can

be eliminated because of the rules:

$$\rho_g(G \oplus H) = \rho_g(G) \oplus \rho_g(H),$$
$$\rho_g(\alpha_{i,j}(G)) = \alpha_{g(i),g(j)}(\rho_g(G)),$$
$$\rho_g(\rho_{i \to j}(G)) = \rho_{g(i) \to g(j)}(\rho_g(G)),$$
$$\rho_g(i) = j \text{ if } j = g(i).$$

Lemma 5.3 *Let L be a finite set of colors. If $g : L \to L$ is not a bijection, then ρ_g is equivalent to a composition of operations $\rho_{i \to j}$, $i, j \in L$.*

Proof. By induction on the cardinality $|L|$ of L. For $|L| = 1$, the statement is trivially true since the only mapping $g : L \to L$ is a bijection.

Let $|L| \geq 2$. Let $R = \{g(i) \mid i \in L\} \subset L$ denote the range of g. Since g is not a bijection, R is a proper subset of L. Let $a \in L - R$, $L' = L - \{a\}$, and let g' be the restriction of g to L'. By construction, the range of g' is contained in L', so we may consider g' as a function $g : L' \to L'$. We have two cases:

CASE 1. g' is not a bijection. Then by induction, $\rho_{g'}$ can be expressed as a composition of operations $\rho_{i \to j}$ for $i, j \in L'$. It is easy to see that by extending such a composition with the operation $\rho_{a \to g(a)}$, we obtain a composition of operations of the form $\rho_{i \to j}$ defining ρ_g for $i, j \in L$.

CASE 2. g' is a bijection. Consider the directed graph G with the vertex set L', where there is an arc from i to j if and only if $j = g'(i)$ (if $g'(i) = i$ we introduce a loop from i to i). Clearly, every vertex in G is of out-degree 1. Moreover, since g' is a bijection, every vertex in G is of in-degree 1. It follows that the graph G is the disjoint union of cycles $C_1, \ldots, C_p$ of length 1 or more for some $p \geq 1$. Let $C \in \{C_1, \ldots, C_p\}$ and write $C = (x_1, \ldots, x_r)$. We can represent the part π_C of g' given by C (that is, $x_2 = g'(x_1), \ldots, x_r = g'(x_{r-1}), x_1 = g'(x_r)$) as a composition of operations of the form $\rho_{i \to j}$ as follows:

$$\pi_C = \rho_{a \to x_1} \circ \rho_{x_1 \to x_2} \circ \cdots \circ \rho_{x_{r-1} \to x_r} \circ \rho_{x_r \to a}.$$

Finally, using the fact that different π_{C_i}'s are independent of each other (except of the use of the auxiliary variable a), we can express ρ_g as a composition of operations of the form $\rho_{i \to j}$, as follows:

$$\pi_{C_p} \circ \cdots \circ \pi_{C_1} \circ \rho_{a \to g(a)}.$$

This completes CASE 2 and with it the proof of the lemma.   ∎

Proof of Proposition 5.1. Let G be a directed graph.

1. We first prove that, if $cwd(G) \leq k$, then G is the value of a term in $T(U_k^{([2])}, C_k^{GF(2)})$. By Lemma 5.2, if $cwd(G) = k$, then G is the value of a term t in $T(F_k^d, C_k^c)$. Let

$\Omega : [k] \to GF(2)^k$ where $\Omega(i) = (0, \cdots, 0, 1, 0, \cdots, 0)$ with 1 at i-th position. We use the following rules to transform t into $\hat{t}$ in $T(U_k^{([2])}, C_k^{GF(2)})$:

$$\hat{t} = \begin{cases} \mathbf{u} & \text{if } t = \mathbf{i} \text{ and } \Omega(i) = u, \\ \widehat{t_1} \otimes_{M,M',N,P} \widehat{t_2} & \text{if } t = t_1 \otimes_{R,R',g,h} t_2. \end{cases}$$

where M, M', N and P are $(k \times k)$-matrices over $GF(2)$ with

$$\begin{aligned} N_i^j &= 1 & \text{if and only if} & & j &= g(i), \\ P_i^j &= 1 & \text{if and only if} & & j &= h(i), \\ M_i^j &= 1 & \text{if and only if} & & (i,j) &\in R \text{ and} \\ (M')_i^j &= 1 & \text{if and only if} & & (i,j) &\in R'. \end{aligned}$$

It is straightforward to verify that $\Omega(i) \cdot N = \Omega(g(i))$, $\Omega(i) \cdot P = \Omega(h(i))$ and that $(i,j) \in R$ if and only if $\Omega(i) \cdot M \cdot \Omega(j)^T = 1$ and $(i,j) \in R'$ if and only if $\Omega(i) \cdot M' \cdot \Omega(j)^T = 1$.

We now prove that, if $G = val(t)$ where $t \in T(U_k^{([2])}, C_k^{GF(2)})$, then $cwd(G) \le 2^{k+1} - 1$. We let $\beta : GF(2)^k \to [2^k]$ be a bijective function that enumerates $GF(2)^k$. We use the following rules to transform t into $\hat{t}$, a clique-width expression of width at most $2^{k+1} - 1$ (for $S \subseteq [k] \times [k]$, we let α_S be $(\circ_{(i,j) \in S} \alpha_{i,j})$):

$$\hat{t} = \begin{cases} \mathbf{i} & \text{if } i = \beta(u) \text{ and } t = \mathbf{u}, \\ \rho_h(\rho_g(\alpha_{R'}(\alpha_R(\widehat{t_1} \oplus (\circ_{i \in B} \; \rho_{i \to i'})(\widehat{t_2}))))) & \text{if } t = t_1 \otimes_{M,M',N,P} t_2 \end{cases}$$

where $g(\beta(u)) = \beta(u \cdot N)$, $h(\beta(v)') = \beta(v \cdot P)$ for all $u, v \in GF(2)^k$ with

$$\begin{aligned} B &= \beta(GF(2)^k - \{(0 \;\; \cdots \;\; 0)\}), \\ R &= \{(\beta(u), \beta(v)') \mid u \cdot M \cdot v^T = 1\}, \\ R' &= \{(\beta(v)', \beta(u)) \mid u \cdot M' \cdot v^T = 1\}. \end{aligned}$$

The operation ρ_g is the one constructed by Lemma 5.3 and since ρ_h renames $\beta(v)'$ into $\beta(v \cdot P)$, Lemma 5.3 is not needed to construct ρ_h.

We verify by induction with Lemma 5.3 and Remark 5.1 that $val(\hat{t}) = G$ and that we use at most $2^{k+1} - 1$ colors. We recall that we do not need to rename the color $\beta((0 \;\; \cdots \;\; 0))$ into $\beta((0 \;\; \cdots \;\; 0))'$ because for all matrices N of appropriate dimension, $u \cdot N = u$ where $u = (0 \;\; \cdots \;\; 0)$.

By Propositions 4.8 and 4.9, we have

$$\frac{1}{2} brwd(G) \le cwd(G) \le 2^{brwd(G)+1} - 1.$$

2. The same technique as in [OS06, Proposition 6.3] yields the result. This concludes the proof. ∎

5.2 Recognition Algorithms for Directed Graphs

In this section, we will prove that for fixed k, there exists a cubic-time algorithm that given a directed graph G either outputs that it has bi-rank-width larger than k or outputs a layout of branch-width k of $\rho_G^{(bi)}$. We also prove that if F is a finite field and $\sigma : F \to F$ is an automorphism, then there exists a cubic-time algorithm that, given a graph over (F, σ), either outputs that it has F-rank-width larger than k or outputs a layout of the function ρ_G^F of branch-width at most k. We can then derive an approximation cubic-time algorithm for clique-width of directed graphs.

Let $f : 2^V \to \mathbb{Z}$ be a symmetric function. Oum and Hliněný [HO07] defined the notion of branch-width of a partition of V. Let $\mathcal{P}$ be a partition of V. For every subset Z of $\mathcal{P}$, we let $f^{\mathcal{P}}(Z) = f(\bigcup_{Y \in Z} Y)$. The function $f^{\mathcal{P}}$ is symmetric, and sub-modular if f is sub-modular [HO07]. We can then define the branch-width of $\mathcal{P}$ as the branch-width of $f^{\mathcal{P}}$ and a layout of $\mathcal{P}$ as a layout of $f^{\mathcal{P}}$. We will use this notion in order to give the recognition algorithms of directed graphs of bi-rank-width and $GF(4)$-rank-width at most k.

5.2.1 Recognizing Graphs of Bounded Bi-Rank-Width

Bi-rank-width is defined by using a coding of directed graphs by two undirected graphs. A "naive idea", which finally is the good one, consists in using a similar coding for recognizing the directed graphs of bi-rank-width at most k. For that purposes, we will use an idea from [HO07] which consists in duplicating each vertex x of G into $(x, 1)$ and $(x, 2)$ and then constructing an undirected bipartite graph $B(G)$ such that, if $x \to y$ in G, then we have $(x, 1) - (y, 2)$ in $B(G)$ and no more edges. Now, if we want to use recognition algorithms for undirected graphs of rank-width k, we have to show that $rwd(B(G)) = f(brwd(G))$ for some function f and show how to transform an optimal layout of the function $\rho_{B(G)}$ into an optimal layout of the function $\rho_G^{(bi)}$. However, we were not able to prove such a statement. To overcome this difficulty, we borrow another idea from [HO07] that consists in defining the branch-width of the partition $\{(x, 1), (x, 2) \mid x \in V_G\}$ of $V_{B(G)}$. We will first show that, for every subset X of V_G, we have $\rho_{B(G)}(X \times \{1, 2\}) = \rho_G^{(bi)}(X)$. As a consequence, if we can find a layout $(T, \mathcal{L})$ of $\rho_{B(G)}$ such that, for every x in V_G, we have a node v in T that is adjacent to both $\mathcal{L}((x, 1))$ and $\mathcal{L}((x, 2))$, we can transform it into a layout, of same branch-width, of $\rho_G^{(bi)}$. The difficulty is now to find such a layout that witnesses the branch-width of $\rho_G^{(bi)}$, which fortunately exists by using some results from [HO07]. We now make precise these constructions.

For each directed graph G, we let $B(G)$ be the simple undirected bipartite graph associated with G where:

$$V_{B(G)} = V_G \times \{1, 2\},$$
$$E_{B(G)} = \{\{(v, 1), (w, 2)\} \mid (v, w) \in E_G\} \cup \{\{(v, 1), (v, 2)\} \mid v \in V_G\}.$$

Figure 16 shows a directed graph G and the corresponding undirected bipartite graph $B(G)$.

Lemma 5.4 *Let G be a directed graph. For every subset X of V_G, we have*
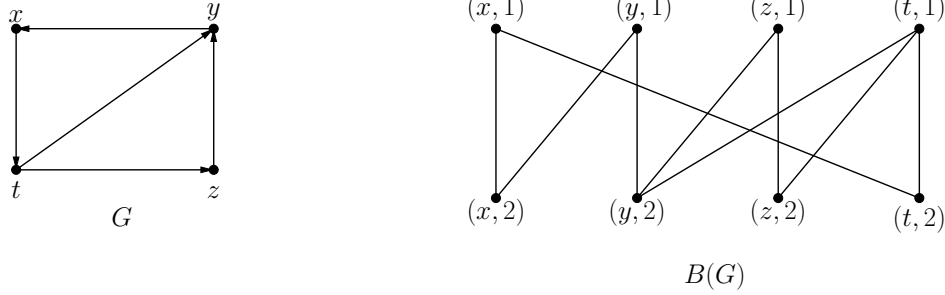
$$\rho_{B(G)}(X \times \{1, 2\}) = \rho_G^{(bi)}(X).$$

Figure 16: A directed graph G and its corresponding undirected bipartite graph $B(G)$.

Proof. From the definition of $B(G)$, we have

$$\rho_{B(G)}(X \times \{1,2\}) = rk\left((A_{B(G)})^{\overline{X} \times \{1,2\}}_{X \times \{1,2\}}\right)$$
$$= rk\left((A_{B(G)})^{\overline{X} \times \{2\}}_{X \times \{1\}}\right) + rk\left((A_{B(G)})^{\overline{X} \times \{1\}}_{X \times \{2\}}\right)$$
$$= rk\left((A_G^+)^{\overline{X}}_X\right) + rk\left((A_G^-)^{\overline{X}}_X\right).$$

We thus conclude that $\rho_{B(G)}(X \times \{1,2\}) = \rho_G^{(bi)}(X)$.   ∎

Let us discuss some consequences of the lemma above. For each $x \in V_G$, we let $P_x = \{(x,1),(x,2)\}$. We let $\Pi(G) = \{P_x \mid x \in V_G\}$. It is worth noticing that $\Pi(G)$ is a perfect matching. Lemma 5.4 implies that, for every subset Y of $\Pi(G)$, we have $\rho_{B(G)}^{\Pi(G)}(Y) = \rho_G^{(bi)}(X)$ where $Y = X \times \{1,2\}$. The following is an analogous of the one in [HO07, Corollary 7.2].

Corollary 5.1 *Let $p : V_G \to \Pi(G)$ be the bijective function such that $p(x) = P_x$. If $(T, \mathcal{L})$ is a layout of branch-width k of $\rho_{B(G)}^{\Pi(G)}$, then $(T, \mathcal{L} \circ p)$ is a layout of branch-width k of $\rho_G^{(bi)}$. Conversely if $(T, \mathcal{L})$ is a layout of branch-width k of $\rho_G^{(bi)}$, then $(T, \mathcal{L} \circ p^{-1})$ is a layout of branch-width k of $\rho_{B(G)}^{\Pi(G)}$.*

We now recall a useful result from Hliněný and Oum [HO07].

Lemma 5.5 ([HO07]) *Let k be a fixed interger. Let G be a graph with n vertices such that $\rho_{B(G)}^{\Pi(G)} = k$. Then one can construct in $O(n^3)$-time a layout of branch-width at most k of $\rho_{B(G)}^{\Pi(G)}$.*

We can give the recognition algorithm for bi-rank-width for fixed k.

Theorem 5.1 (Checking Bi-Rank-Width at most k) *For fixed k, there exists a cubic-time algorithm that for a directed graph G, either outputs a layout of branch-width at most k of $\rho_G^{(bi)}$ or confirms that the bi-rank-width of G is larger than k.*

Proof. Let G be a directed graph with n vertices. We can construct $B(G)$ in $O(n^2)$-time. We apply the algorithm of Lemma 5.5 with input $(\Pi(G), k)$. If it confirms that the branch-width of $\rho_{B(G)}^{\Pi(G)}$ is greater than k, then $brwd(G) > k$ (Lemma 5.4). If it outputs a layout of $\rho_{B(G)}^{\Pi(G)}$ of

branch-width at most k, we can transform it into a layout of $\rho_G^{(bi)}$ of branch-width at most k in $O(n)$-time by Corollary 5.1.   ∎

Corollary 5.2 *For fixed k, there exists a cubic-time approximation algorithm that, for a directed graph G, either outputs a clique-width expression of width at most $2^{2k+1} - 1$ or confirms that the clique-width of G is larger than k.*

Proof. Let G be a directed graph with n vertices. We run the algorithm of Proposition 5.1 with input G and $2k$. If it confirms that $brwd(G) > 2k$, then $cwd(G) > k$. If it outputs a layout of $\rho_G^{(bi)}$ of branch-width at most $2k$, we can transform it into a clique-width expression of width at most $2^{2k+1} - 1$ by Proposition 4.8, 4.9 and 5.1 in $O(n^2)$-time.   ∎

5.2.2   Recognizing Graphs of Bounded F-Rank-Width

Let F be a finite field and let $\sigma : F \to F$ be an automorphism. We now give, for fixed k, a recognition algorithm for graphs over (F, σ) of F-rank-width at most k (c.f. Chapter 4 for the definition of F-rank-width). Before, let us recall some useful concepts borrowed from Oum and Hliněný [HO07] that help them to give a cubic-time algorithm for recognizing undirected graphs of rank-width at most k (Theorem 1.2). We refer to Schrijver [Sch03] for our matroid terminology.

Definition 5.3 (Matroids) A pair $M = (S, \mathcal{I})$ is called a *matroid* if S is a finite set and $\mathcal{I}$ is a nonempty collection of subsets of S satisfying the following conditions

(M1) if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$,

(M2) if $I, J \in \mathcal{I}$ and $|I| < |J|$, then $I \cup \{z\} \in \mathcal{I}$ for some $z \in J - I$.

For $U \subseteq S$, a subset B of U is called a *base* of U if B is an inclusionwise maximal subset of U and belongs to $\mathcal{I}$. It is easy to see that, if B_1 and B_2 are bases of $U \subseteq S$, then B_1 and B_2 have the same size. The common size of the bases of a subset U of S is called the *rank of* U, denoted by $r_M(U)$. A set $B \subseteq S$ is a base of M if it is a base of S.

Let A be an $(m \times n)$-matrix. Let $S = \{1, \ldots, n\}$ and let $\mathcal{I}$ be the collection of all those subsets I of S such that the columns of A with index in I are linearly independent. Then $M = (S, \mathcal{I})$ is a matroid. If A has entries in a field F, then M is said *representable over F* and A is called a *representation of M*.

We now define the branch-width of matroids. Let $M = (S, \mathcal{I})$ be a matroid. We let λ_M be defined such that for every subset U of S, $\lambda_M(U) = r_M(U) + r_M(S - U) - r_M(S) + 1$ and call it the *connectivity function of M*. The function λ_M is symmetric and sub-modular [Sch03, HO07]. The *branch-width of M*, denoted by $bwd(M)$, is the branch-width of λ_M and a layout of M is a layout of λ_M.

Definition 5.4 (Partitioned Matroids [HO07]) A pair $(M, \mathcal{P})$ is called a *partitioned matroid* if $M = (S, \mathcal{I})$ is a matroid and $\mathcal{P}$ is a partition of S. A partitioned matroid $(M, \mathcal{P})$ is *representable over F* if M is representable over F. For a matroid $M = (S, I)$ and a partition $\mathcal{P}$

of S, we let $\lambda_M^{\mathcal{P}}$ be the connectivity function of the partitioned matroid $(M, \mathcal{P})$, i.e., for every $Z \subseteq \mathcal{P}$, we have $\lambda_M^{\mathcal{P}}(Z) = \lambda_M(\bigcup_{Y \in Z} Y)$. The *branch-width of* $(M, \mathcal{P})$, denoted by $bwd(M, \mathcal{P})$, is the branch-width of $\lambda_M^{\mathcal{P}}$ and a layout of $(M, \mathcal{P})$ is a layout of $\lambda_M^{\mathcal{P}}$.

We now recall the following by Hliněný and Oum.

Theorem 5.2 ([HO07]) *Let k be fixed and F be a fixed finite field. Then there exists an $O(n^3)$-time algorithm that takes as input a representable matroid $M = (S, I)$ over F given with its representation and a partition $\mathcal{P}$ of S, $|S| = n$, and outputs a layout of branch-width at most k of $\lambda_M^{\mathcal{P}}$, or confirms that the branch-width of $(M, \mathcal{P})$ is strictly greater than k.*

We can now derive our algorithm from Theorem 5.2. For a set X, we let X' be a copy of it defined as $\{x' \mid x \in X\}$. Let G be a graph over (F, σ) and let F_G be its adjacency matrix over F. Let $Bin(G)$ be the matroid on $V_G \cup V_G'$ represented by the $(V_G, V_G \cup V_G')$-matrix

$$\begin{array}{cc} V_G & V_G' \\ V_G \begin{pmatrix} I_{|V_G|} & F_G \end{pmatrix} \end{array}.$$

For each $x \in V$, we let $P_x = \{x, x'\}$ and we let $\Pi(G) = \{P_x \mid x \in V_G\}$. We now prove the following which is a counterpart of [Oum05b, Proposition 3.1].

Proposition 5.2 *Let G be a graph over (F, σ) and let $\mathcal{M} = Bin(G)$. Then for every $X \subseteq V_G$, $\lambda_{\mathcal{M}}^{\Pi(G)}(P) = 2 \cdot \rho_G^F(X) + 1$ where $P = \{P_x \mid x \in X\}$.*

Proof. For $X \subseteq V$ and $P = \{P_x \mid x \in X\}$, we have

$$\lambda_{\mathcal{M}}^{\Pi(G)}(P) = r_{\mathcal{M}}(X \cup X') + r_{\mathcal{M}}(V_G - X \cup (V_G - X)') - r_{\mathcal{M}}(V_G \cup V_G') + 1$$

$$= rk \begin{pmatrix} 0 & (F_G)_{V_G - X}^X \\ I_{|X|} & (F_G)_X^X \end{pmatrix} + rk \begin{pmatrix} 0 & (F_G)_X^{V_G - X} \\ I_{|V_G| - |X|} & (F_G)_{V_G - X}^{V_G - X} \end{pmatrix} - |V_G| + 1$$

$$= |X| + rk((F_G)_{V_G - X}^X) + |V_G - X| + rk((F_G)_X^{V_G - X} - |V_G| + 1$$

$$= 2 \cdot \rho_G^F(X) + 1.$$

∎

As a corollary we get the following.

Corollary 5.3 *Let G be a graph over (F, σ). Let $\mathcal{M} = Bin(G)$ and let $p : V_G \to \Pi(G)$ be the bijective function such that $p(x) = P_x$. If $(T, \mathcal{L})$ is a layout of $\lambda_{\mathcal{M}}^{\Pi(G)}$ of branch-width $2k + 1$, then $(T, \mathcal{L} \circ p)$ is a layout of ρ_G^F of branch-width k. Conversely, if $(T, \mathcal{L})$ is a layout of branch-width k of ρ_G^F, then $(T, \mathcal{L} \circ p^{-1})$ is a layout of branch-width $2k + 1$ of $\lambda_{\mathcal{M}}^{\Pi(G)}$.*

We can now deduce for fixed k, a recognition algorithm for graphs of F-rank-width at most k, and in particular for directed graphs of $GF(4)$-rank-width at most k.

Theorem 5.3 (Checking F-Rank-Width at most k) *For fixed k, there exists a cubic-time algorithm that, for a graph G over (F, σ), either outputs a layout of ρ_G^F of branch-width at most k or confirms that the F-rank-width of G is larger than k.*

Proof. Let G be a directed graph with n vertices. We run the algorithm of Theorem 5.2 with input $\mathcal{M} = (Bin(G), \Pi(G))$ and $2k + 1$ (Proposition 5.2). If it confirms that $bwd(Bin(G), \Pi(G)) > k$, then the F-rank-width of G is greater than k. If it outputs a layout of $\lambda_{\mathcal{M}}^{\Pi(G)}$ of branch-width at most $2k + 1$, we can transform it into a layout of ρ_G^F of branch-width at most k by Corollary 5.3. ∎

Corollary 5.4 *For fixed k, there exists a cubic-time algorithm that, for a graph G over (F, σ), either outputs a term in $T(R_{2k}^F, C_{2k}^F)$ that generates G or confirms that the F-rank-width of G is larger than k.*

We have seen in Proposition 4.6 that if a graph over (F, σ) is the value of a term t in $T(B_k^F, C_k^F)$, then it is the value of a term in $T(R_{q^k}^F, C_{q^k}^F)$ where q is the number of elements in F. Then with Corollary 5.4, for every graph G over (F, σ) that is the value of a term in $T(B_k^F, C_k^F)$, we can construct in cubic-time a term in $T(B_{2 \cdot q^k}^F, C_{2 \cdot q^k}^F)$ for it.

5.3 Conclusion

Courcelle in [Cou06a] shows how to encode directed graphs by undirected graphs by using MS-definition schemes and in both directions. This gives by using [HO07] approximation algorithms for the bi-rank-width of directed graphs. Here, we avoid such encodings and we get, by adapting the proof of Hliněný and Oum an exact algorithm. It is important not only to check that the rank-width of a graph is at most k, but also to produce an associated decomposition (here a layout). If after that, one wishes to check an MS-definable property by the methods explained in [CMR00, Mak04], based on finite automaton on terms, where automaton are built from MS formulas, he can use graph operations that handle graphs of rank-width k. The construction of such automaton is outside the scope of the present work.

We have seen that for fixed k, there exist algorithms that decide if a directed graph has bi-rank-width (resp. $GF(4)$-rank-width) at most k and if so outputs a layout of branch-width k of $\rho_G^{(bi)}$ (resp. $\rho_G^{GF(4)}$). However these algorithms are deeply based on the results of Oum and Hliněný, especially on Theorem 5.2, which is not really implementable. It is thus a challenge to find implementable algorithms for bi-rank-width as well as $GF(4)$-rank-width. It is also a challenge, for fixed k, to find a polynomial-time algorithm that, given a graph, either outputs that it has clique-width larger than k, or outputs a clique-width expression that uses at most k colors.

Chapter 6

Balanced Graph Expressions

For algorithmic purposes, it is useful and sometimes crucial to have for given graphs tree-structurings based on trees of logarithmic height hence, in our case to have a-balanced binary terms, i.e, with syntactic trees of height at most $a(\log(n)+1)$ where n is the number of nodes and a is a constant. For instance, the labeling schemes of [CV03, CT07] are defined as follows and use such notion.

- Given a binary term t that represents a graph G (the leaves of the term are in bijection with the vertices of G), we run an automaton on G hence, each node of t is assigned a state of the automaton.

- Each vertex x of G is assigned a label that is, roughly, a sequence of states met on the unique path from the root of t to the leaf of t that represents the vertex x.

Then in order to have labels of size $O(\log(|V_G|)$ in the labeling schemes considered in [CV03, CT07], it is important that the given term has logarithmic height.

Another practical use of balanced terms is the design of parallel algorithms. This is done for instance by Bodlaender for the design of parallel algorithms in order to construct minimum-width tree-decompositions of graphs or to solve some NP-complete problems [Bod88, BH98]. Balanced terms play also an important role in succinct representations of graphs, particularly succinct representations of graphs by Boolean functions as considered in [MR06, NW05][1]. Bodlaender [Bod88] proved that every graph of tree-width k admits a 2-balanced binary tree-decomposition of width at most $3k + 2$. Courcelle and Twigg [CT07] proved that every graph of m-clique-width k admits a 6-balanced m-clique-width expression of width at most $2k$. However, the proofs of Bodlaender and Courcelle et al. used the same ideas. We give a unifying framework that covers several particular cases. In particular, we prove that every undirected graph of rank-width k admits a 3-balanced layout witnessing rank-width.

---

[1]As in [CV03, CT07] these labelings are defined as sequences of values met on the unique path from the root to a leaf, or more precisely at distance 1 of the nodes of this path.

6.1   General Framework

The notions of many-sorted algebras are presented in [Cou96, Wec92, Wir]. We present here the needed definitions. We recall the notions of binary signature for the purposes of this chapter.

Let $\mathcal{S}$ be a countable set whose elements are called *sorts*. A *binary $\mathcal{S}$-signature* is a pair (F, C) where F is a set of binary function symbols, each of them having a type $s_1 \times s_2 \to s$ where $s_1, s_2, s \in \mathcal{S}$, and C is a set of nullary symbols, each of them having a type s in $\mathcal{S}$. A nullary symbol is called a *constant*. We say that a binary $\mathcal{T}$-signature (F, C) is a sub-signature of a binary $\mathcal{S}$-signature (F', C') if $\mathcal{T} \subseteq \mathcal{S}$, $F \subseteq F'$, $C \subseteq C'$ and the types of the elements of F and C are the same for (F', C') and for (F, C). We recall that $T(F, C)$ denotes the set of finite well-formed terms written with symbols from $F \cup C$ (see Chapter 1). We denote by $T(F, C)_s$ the set of finite well-formed terms of sort s (the sort of a term in $T(F, C)$ is that of its first symbol in prefix notation). Let χ be a set of $\mathcal{S}$-*sorted variables*, i.e., each variable x in χ has a sort s in $\mathcal{S}$. We denote by $T(F, C \cup \chi)$ the set of well-formed terms written with symbols in $F \cup (C \cup \chi)$.

Let (F, C) be a binary $\mathcal{S}$-signature. An (F, C)-*algebra* is an element $\mathcal{M} = \langle (M_s)_{s \in \mathcal{S}}, (f_{\mathcal{M}})_{f \in F}, (b_{\mathcal{M}})_{c \in C} \rangle$, where for each s in $\mathcal{S}$, M_s is a nonempty set, called the *domain of sort s of $\mathcal{M}$*, for each $f \in F$ of type $s_1 \times s_2 \to s$, $f_M : M_{s_1} \times M_{s_2} \to M_s$ is a total mapping and for each $b \in C$ of type s, $b_{\mathcal{M}}$ is a constant of sort s, i.e., is a member of M_s.

Definition 6.1 (Height of a Term) Let (F, C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$. For every term t in $T(F, C)$ we let $ht(t)$, called the *height* of t, be defined inductively as follows:

$$ht(t) = \begin{cases} 1 & \text{if } t = c \in C \\ 1 + \max\{ht(t_1), ht(t_2)\} & \text{if } t = f(t_1, t_2) \end{cases}$$

Let a be a positive real number. A term t in $T(F, C)$ is *a-balanced* if $ht(t) = a \cdot \log(|t| + 1)$.

The definition of a balanced term is meaningful in the case $|t| = 1$. All logarithms are in base 2.

Definition 6.2 (Equivalence of Terms) Let (F, C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$ and let $\mathscr{C}$ be a class of (F, C)-algebras. Two terms t and t' in $T(F, C)$ are *equivalent with respect to $\mathscr{C}$* if, for every (F, C)-algebra $\mathcal{M}$ in $\mathscr{C}$, we have $val_{\mathcal{M}}(t) = val_{\mathcal{M}}(t')$. We denote by $\simeq_{\mathscr{C}}$ the equivalence of terms with respect to the class of (F, C)-algebras $\mathscr{C}$. We omit the subscript $\mathscr{C}$ when $\mathscr{C}$ is implicitly assumed.

We can now introduce some definitions and basic properties before stating and proving the main theorem of this chapter. See Definition 1.1 for the definition of contexts. We first define the notion of *special terms* introduced by Courcelle and Vanicat in [CV03].

Definition 6.3 (Special Terms) Let (F, C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$ and let S be the set $T(F \cup \{\circ, \bullet\}, C \cup \{Id\})$. We let S_c and S_t be the least subsets of S

such that:

$$S_t = S_c \bullet S_t \ \cup \ f(S_t, S_t) \ \cup \ b,$$
$$S_c = S_c \circ S_c \ \cup \ f(S_t, S_c) \ \cup \ f(S_c, S_t) \ \cup \ f(S_t, Id) \ \cup \ f(Id, S_t)$$

with rules for each f in F and each b in C. We denote them by $SPE_t(F, C)$ and $SPE_c(F, C)$ if we need to specify F and C. Note that $Id \notin S_t \cup S_c$.

The notions of context and the operations $\circ$ and $\bullet$ extend clearly in presence of sorts. We have actually several operations $\circ$, $\bullet$ and several constants Id depending on sorts, but we will overlook this technical point.

For every term $c \in SPE_t(F, C) \cup SPE_c(F, C)$, we let $Eval(c)$ be defined inductively as follows:

$$Eval(c) = \begin{cases} b \in C & \text{if } c = b, \\ f(Eval(t_1), Eval(t_2)) \in T(F, C) & \text{if } t = f(t_1, t_2) \text{ and } t_1, t_2 \in S_t, \\ Eval(c_1) \bullet Eval(t) \in T(F, C) & \text{if } c = c_1 \bullet t \text{ and } c_1 \in S_C, t \in S_t, \\ f(u, Eval(t)) \in Cxt(F, C) & \text{if } c = f(Id, t) \text{ and } t \in S_t, \\ f(Eval(t), u) \in Cxt(F, C) & \text{if } c = f(t, Id) \text{ and } t \in S_t, \\ f(Eval(c_1), Eval(t)) \in Cxt(F, C) & \text{if } c = f(c_1, t) \text{ and } c_1 \in S_c, t \in S_t \\ f(Eval(t), Eval(c_1)) \in Cxt(F, C) & \text{if } c = f(t, c_1) \text{ and } c_1 \in S_c, t \in S_t, \\ Eval(c_1) \circ Eval(c_2) \in Cxt(F, C) & \text{if } c = c_1 \circ c_2 \text{ and } c_1, c_2 \in S_c. \end{cases}$$

We can now define what we mean by *commutativity* of a function in a binary $\mathcal{S}$-signature.

Definition 6.4 (Commutativity) Let $\mathcal{S}$ be a countable set of sorts. A binary $\mathcal{S}$-signature (F, C) is *commutative* with respect to a class of (F, C)-algebras $\mathscr{C}$ (that will be implicitly assumed in most cases) if for every $f \in F$ of type $s_1 \times s_2 \rightarrow s$, there exists a function $\tilde{f}$ in F of type $s_2 \times s_1 \rightarrow s$ such that:

$$\tilde{f}_{\mathcal{M}}(x, y) = f_{\mathcal{M}}(y, x) \tag{1}$$

for all $\mathcal{M} \in \mathscr{C}$, all $x \in M_{s_1}$ and all $y \in M_{s_2}$. If a binary $\mathcal{S}$-signature is not commutative, we let $F' \subseteq F$ be the set of functions f for which we do not have $\tilde{f}$ satisfying (1), then we can enrich F into $\widehat{F} = F \cup \{\hat{f} \mid f \in F'\}$ and define

$$\hat{f}_{\mathcal{M}}(x, y) = f_{\mathcal{M}}(y, x) \text{ if } f \in F'.$$

It is clear that $\widehat{F}$ is commutative with respect to the considered class $\mathscr{C}$. It is finite if F is, and the set of sorts is the same.

The commutativity of the binary operations of the signature plays a significant role in our framework. Commutative binary operations allow us to rearrange terms. We now define a decomposition of terms using the notion of *comb-term* and the commutativity property.

Definition 6.5 (Comb-Term) Let (F, C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$ and let χ be an $\mathcal{S}$-sorted set of variables. A *comb-term* is a term in $T(F, C \cup \chi)$ of the form

$$q = f_1(x_1, f_2(x_2, \ldots, f_n(x_n, x_{n+1})) \ldots)$$

where $x_1, \ldots, x_{n+1}$ are in χ. It contains no constant. We denote it also by $q(x_1, \ldots, x_n, x_{n+1})$ in order to specify the list of variables and the order in which they occur.

Definition 6.6 (Comb-Decomposition) Let (F, C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$ and let χ be an $\mathcal{S}$-sorted set of variables. The *comb-decomposition* of a term $t \in T(F, C) - C$ is the unique writing of t as $q(t_1, \ldots, t_n, b)$ where $q(x_1, \ldots, x_{n+1})$ is a comb-term, $b \in C$ and for each i in $[n]$, the term t_i is in $T(F, C)$.

We now define a notion of comb-decomposition for contexts in $SPE_c(F, C)$ and it makes sense only if F is commutative. For every $c \in Cxt(F, C) - \{Id\}$, we define $Comb(c)$ and $seq(c)$ inductively as follows:

(CS1) $Comb(c) = f(x_1, u)$ and $seq(c) = (t)$ if $c = f(t, Id)$.

(CS2) $Comb(c) = Comb(c')$ and $seq(c) = seq(c')$ if $c = f(c_1, t)$ and $c' = \tilde{f}(t, c_1)$.

(CS3) $Comb(c) = f(x_1, q(x_2, \ldots, x_{n+1}, u))$ and $seq(c) = (t) \cdot seq(c')$ if $c = f(t, c_1)$, $c_1 \neq Id$ and $Comb(c') = q(x_1, \ldots, x_n, u)$.

(CS4) $Comb(c) = q'(x_1, \ldots, x_p, q''(x_{p+1}, \ldots, x_{n+p}, u))$ and $seq(c) = seq(c') \cdot seq(c'')$ if $c = c' \circ c''$ (so that $c' \neq Id$, $c'' \neq Id$), $Comb(c') = q'(x_1, \ldots, x_p, u)$ and $Comb(c'') = q''(x_1, \ldots, x_n, u)$.

Example We let $c = f(g(u, a), h(b, d))$. Then $c \simeq \tilde{f}(h(b, d), \tilde{g}(a, u))$ and $Comb(c) = \tilde{f}(x_1, \tilde{g}(x_2, u))$, $seq(c) = (h(b, d), a)$.

We now give easy properties for $Comb(c)$ and $seq(c)$.

Fact 6.1 If $Comb(c) = q(x_1, \ldots, x_n, u)$ and $seq(c) = (t_1, \ldots, t_n)$ then

$$c \simeq q(t_1, \ldots, t_n, u).$$
$$Eval(c) \simeq q(Eval(t_1), \ldots, Eval(t_n), u).$$

Proof. By induction on the structure of c.   ∎

In the following, we will extend the equivalence relation $\simeq$ by letting

$$Eval(t) \simeq t \text{ and } Eval(c) \simeq c$$

for terms in $SPE_t(F, C) \cup SPE_c(F, C)$. The notion of *flexibility*, which is defined below, generalizes the notion of associativity.

Definition 6.7 (Flexibility) Let $\mathcal{S}$ be a countable set of sorts and let χ be an $\mathcal{S}$-sorted set of variables. We let (F', C') and (F, C) be two binary $\mathcal{S}$-signatures such that $(F, C) \subseteq (F', C')$ and we let $\mathscr{C}$ be a set of (F', C')-algebras. We say that (F', C') is (F, C)-*flexible* if the following conditions hold:

(F1) F and F' are commutative.

(F2) There exist three mappings: $q \mapsto \hat{q}$, $q \mapsto f^q$ and $(q, q') \mapsto f^{q,q'}$ which satisfy the following properties:

   (F2.1) If $q(x_1, u)$ is the comb-term $g(x_1, u)$, then $\hat{q} = x_1$ and $f^q = g$.

   (F2.2) For every comb-term $q(x_1, \ldots, x_n, u)$ in $T(F, C \cup \chi)$ with $n \geq 2$, $\hat{q}(x_1, \ldots, x_n)$ is a comb-term in $T(F', C \cup \chi)$, $f^q \in F'$ and $q \simeq_{\mathscr{C}} f^q(\hat{q}, u)$.

   (F2.3) For every two comb-terms $q(x_1, \ldots, x_p, u)$ and $q'(x_1, \ldots, x_n, u)$ in $T(F, C \cup \chi)$, we have $f^{q,q'} \in F'$ and

$$\widehat{q''} \simeq_{\mathscr{C}} f^{q,q'}(\hat{q}(x_1, \ldots, x_p), \widehat{q'}(x_{p+1}, \ldots, x_{p+n}))$$

   where $q'' = q(x_1, \ldots, x_p, q'(x_{p+1}, \ldots, x_{p+n}, u))$.

Remark 6.1   1. If $q = g(x_1, u)$ is a comb-term, then Property (F2.2) also holds from the definitions of $\hat{q}$ and f^q.

   2. If $F = \{f\}$ and f is associative and commutative (with respect to $\mathscr{C}$), then (F, C) is (F, C)-flexible. It is clear that every term t in $T(F, C)$ is equivalent to a term $\tilde{t}$ in $T(F, C)$ of height $\lceil \log(|t|+1) \rceil$. The notion of flexibility generalizes this condition and will give a similar result.

Our objective is to prove the following.

Theorem 6.1 (Balanced Terms) *Let (F', C') be an (F, C)-flexible binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$. Every term t in $T(F, C)$ of size n is equivalent to a 3-balanced term t' in $T(F', C')$. Moreover, this term can be constructed in $O(n \cdot \log(n))$-time, if we assume that $\hat{q}$, f^q, $f^{q,q'}$ can be constructed in $O(|q|)$-time.*

The proof of this theorem consists in transforming a term in $T(F, C)$ into a 3-balanced term in $SPE_t(F, C)$. Then we transform the obtained term in $SPE_t(F, C)$ into a term in $T(F', C')$ of same height by replacing each binary operation $\circ$ or $\bullet$ by a binary operation in F'. The use of $\bullet$ and of $\circ$ is from [CV03]. The notion of flexibility is new.

For terms t in $SPE_t(F, C) \cup SPE_c(F, C)$ we denote by $|t|_{FC}$ the number of occurrences of symbols from $F \cup C$, by $|t|_0$ the number of occurrences of $\circ$ and $\bullet$, and, by $|t|_{Id}$ the number of occurrences of Id. It is clear from the recursive equations defining special terms that:

$$\begin{aligned}
|t| &= |t|_{FC} + |t|_0 + |t|_{Id} && \text{if } t \in SPE_t(F, C) \cup SPE_c(F, C), \\
|t|_{Id} &= |t|_0 && \text{if } t \in SPE_t(F, C), \\
|c|_{Id} &= |c|_0 + 1 && \text{if } c \in SPE_c(F, C).
\end{aligned}$$

The following is proved in [CV03]. Since F is a set of binary functions, each term in $T(F, C)$ has odd size.

Proposition 6.1 *[CV03, Lemmas 1 and 2] Let (F, C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$.*

   1. *Every term $t \in T(F, C)$ of size $n = 2p + 1$, for $p \geq 1$, can be written $t = c_1 \bullet f(t_1, t_2)$ where:*

> *1.1 $c_1 \in Cxt(F,C)$ with $|c_1| \leq p$ and is of maximal size with this property.*
> *1.2 $t_i \in T(F,C)$ with $|t_i| \leq p+1$ for each $i = 1,2$.*

2. *Every context $c \in Cxt(F,C)$ of size $n = 2p+1$, for $p \geq 1$, can be written $c = c_1 \circ f(c_2, t_1)$ or $c = c_1 \circ f(t_1, c_2)$ where:*

> *2.1 $c_1, c_2 \in Cxt(F,C)$ with $|c_1| \leq p$ is of maximal size with this property and then $|c_2| \leq p+1$.*
> *2.2 $t_1 \in T(F,C)$ with $|t_1| \leq 2p-1$.*

Remark 6.2 Let $t = f(s_1, s_2)$, $p = (|s_1| + |s_2|)/2$. If $|s_1| - 2 \leq |s_2| \leq |s_1| + 2$, then in case *(1)* of the proposition we must take $c_1 = u$. If $|s_1| = |s_2| + 2$, then the "larger context" $c'_1 = f(u, s_2)$ has size $2 + |s_2| \geq p+1$ since $p + 1 = (|s_1| + |s_2|)/2 + 1 = |s_2| + 2$, hence c_1 is of maximal size $\leq p$. If $|s_1| = |s_2|$ or if $|s_2| = |s_1| + 2$ the same argument works.

Similarly if $c = f(c', s_2)$ and $|c'| \leq |s_2| + 2$ (in particular if $c' = u$) we must take $c_1 = u$ to satisfy *(2)*. Taking the "larger context" $c'_1 = f(u, s_2)$ would necessitate $|c'_1| \leq p$ that is $2 + |s_2| \leq (|c'| + |s_2|)/2$, i.e., $|c'| \geq |s_2| + 4$.

We now show how to transform a term in $T(F,C)$ into a term in $SPE_t(F,C)$. A more careful proof than the one of [CV03, Theorem 1] gives the following result.

Theorem 6.2 *Let (F,C) be a binary $\mathcal{S}$-signature for some countable set of sorts $\mathcal{S}$. For every term t in $T(F,C) - C$, one can construct a term t^b in $SPE_t(F,C)$ such that $|t^b|_{FC} = |t|_{FC} = |t|$, $Eval(t^b) = t$, $ht(t^b) \leq 3 \cdot \log(|t| - 1)$ and $|t^b| \leq 2 \cdot |t| - 1$. This term can be constructed in $O(|t| \cdot \log(|t|))$-time.*

Proof. We construct t^b by induction on the structure of $t \in T(F,C)$. For that purpose, we will simultaneously construct for $c \in Cxt(F,C)$, a context c^b in $SPE_c(F,C)$ such that $Eval(c^b) = c$, $|c^b|_{FC} = |c|_{FC}$, $ht(c^b) \leq 3 \log(|c| - 1) + 2$ and $|c^b| \leq 2 \cdot |c| - 1$. Note that $|c| = |c|_{FC} + 1$.

CASE 1. $t \in T(F,C)$ and have size $|t| = 2p + 1$.

CASE 1.1.   If $|t| = 3$, then we let $t^b = t$. It is clear that $ht(t^b) = 2 < 3 \log(|t| - 1)$ and $|t^b| = |t| \leq 2 \cdot |t| - 1$.

CASE 1.2.   If $|t| = 2p + 1 > 3$. Then we use Proposition 6.1 and we write $t = c_1 \bullet f(t_1, t_2)$.

CASE 1.2.1. $c_1 = u$. This means that $\left| |t_1| - |t_2| \right| \leq 2$. Assume on the contrary that $|t_1| \geq |t_2| + 4$, then $|f(u, t_2)| = 2 + |t_2| \leq p = (|t_1| + |t_2|)/2$ and c_1 is not of maximal size such that (1.1) of Proposition 6.1 holds, because it can be replaced by a "larger context", e.g., $f(u, t_2)$. It is clear now that $|p_1 - p_2| \leq 1$ where $|t_i| = 2p_i - 1$ for $i = 1,2$. We can then deduce for $i = 1,2$ that $2p_i \leq |t| - 1$. We let $t^b = f(t_1^b, t_2^b)$. We first prove that $ht(t^b) \leq 3 \log(|t| - 1)$.

If we assume for $i = 1,2$ that $t_i \notin C$, then we have by inductive hypothesis:

$$1 + ht(t_i^b) \leq 1 + 3 \log(2p_i)$$
$$\leq 1 + 3 \log((|t| - 1)/2)$$
$$= -2 + 3 \log(|t| - 1) < 3 \log(|t| - 1).$$

And since $ht(t^b) = \max\{1 + ht(t^b_1), 1 + ht(t^b_2)\}$, we have $ht(t^b) < 3\log(|t| - 1)$. We now prove that $|t^b| \le |t| - 1$.

$$
\begin{aligned}
|t^b| &= |t^b_1| + |t^b_2| + 1 \\
&\le 2|t_1| - 1 + 2|t_2| - 1 + 1 \\
&\le 2(|t_1| + |t_2|) - 1 < 2 \cdot |t| - 1.
\end{aligned}
$$

The particular case $|t_1| = 1$ implies $|t_2| = 3$, $|t| = 5$ and $t^b = t$. Then $ht(t) = 3 < 3\log(4) = 6$.

CASE 1.2.2. $c_1 \ne u$. We have $|c_1| \le p$, $|t_i| \le p + 1$. We let $t^b = c^b_1 \bullet f(t^b_1, t^b_2)$. We must prove that:

$$
\begin{aligned}
1 + ht(c^b_1) &\le 3\log(2p) \\
2 + ht(t^b_i) &\le 3\log(2p) \qquad\qquad\qquad\qquad \text{for } i = 1, 2
\end{aligned}
$$

By inductive hypothesis we have:

$$
\begin{aligned}
1 + ht(c^b_1) &\le 1 + 3\log(p - 1) + 2 \le 3\log(p) + 3 = 3\log(2p) \\
2 + ht(t^b_i) &\le 2 + 3\log(p) < 3\log(2p) \qquad\qquad\qquad \text{for } i = 1, 2.
\end{aligned}
$$

We now prove that $|t^b| \le 2|t| - 1$. We have:

$$
\begin{aligned}
|t^b| &= |c^b_1| + |t^b_1| + t^b_2| + 2 \\
&\le 2|c_1| - 1| + 2|t_1| - 1 + 2|t_2| - 1 + 2 \qquad \text{(by inductive |hypothesis)} \\
&\le 2(|c_1| + |t_1| + |t_2|) - 1 = |2 \cdot |t| - 1.
\end{aligned}
$$

CASE 2.   We now consider the case of $c \in Cxt(F, C)$ of size $n = 2p + 1$.

CASE 2.1.   If $n = 3$, then we let $c^b = c$ and the result holds, as above for CASE 1.1.

CASE 2.2.   Next we consider the case $|c| = 2p + 1 > 3$. By Proposition 6.1, we have $c = c_1 \circ f(c_2, t_1)$ or $c = c_1 \circ f(t_1, c_2)$ with c_1 of maximal size, $|c_1| \le p$. We only consider the first case (by symmetry).

CASE 2.2.1. $c_1 = u$. This means that $c = f(c_2, t_1)$, $|c_2| \le |t_1| + 2$, because, if $|c_2| \ge |t_1| + 4$, then $c_1 = u$ could be replaced by a "larger context", e.g., $f(u, t_1)$. We take $c^b = f(c^b_2, t^b_1)$. We have $|c_2| = 2p_2 + 1$, $|t_1| = 2p_1 + 1$, $p_2 \le p_1 + 1$. The proof is similar to CASE 1.2.1. We must prove that:

$$
\begin{aligned}
1 + ht(t^b_1) &\le 3\log(2p) + 2, \\
1 + ht(c^b_2) &\le 3\log(2p) + 2.
\end{aligned}
$$

We have by inductive hypothesis that

$$
1 + ht(t^b_1) \le 1 + 3\log(|t_1| - 1) \le 1 + 3\log(|c| - 1) < 3\log(|c| - 1) + 2.
$$

We also have that $1 + ht(c^b_2) \le 1 + 3\log(2p_2) + 2$. Since $4p_2 \le 2p_2 + 2(p1 + 1) = |c| - 1$, then

$$
1 + ht(c^b_2) \le 3 + 3\log((|c| - 1)/2) = 3\log(|c| - 1) < 3\log(|c| - 1) + 2.
$$

We now prove that $|c^b| \leq 2 \cdot |c| - 1$.

$$
\begin{aligned}
|c^b| &= |c_2^b| + |t_1^b| + 1 & \text{by definition,} \\
&\leq 2|c_2| - 1 + 2|t_1| - 1 + 1 & \text{by inductive hypothesis,} \\
&\leq 2(|c_2| + |t_1|) - 1 < 2 \cdot |c| - 1.
\end{aligned}
$$

CASE 2.2.2. $c_1 \neq u$. Then we let $c^b = c_1^b \circ f(c_2^b, t_1^b)$. We have $|c_1| \leq p$, $|c_2| \leq p + 1$ and $|t_1| \leq 2p - 1$. We must prove that

$$
\begin{aligned}
1 + ht(c_1^b) &\leq 3\log(2p) + 2, \\
2 + ht(c_2^b) &\leq 3\log(2p) + 2, \\
2 + ht(t_1^b) &\leq 3\log(2p) + 2.
\end{aligned}
$$

By using induction we have

$$
1 + ht(c_1^b) \leq 1 + 3\log(p - 1) + 2 < 3 + 3\log(p) = 3\log(2p).
$$

And

$$
2 + ht(c_2^b) \leq 2 + 3\log(p) + 2 = 3\log(2p) + 1.
$$

And

$$
2 + ht(t_1^b) \leq 2 + 3\log(2p - 2) < 3\log(2p) + 2.
$$

For the size of $|c^b|$ we have

$$
\begin{aligned}
|c^b| &= |c_1^b| + |c_2^b| + |t_1^b| + 2 \\
&\leq 2|c_1| - 1 + 2|c_2| - 1 + 2|t_1| - 1 + 2 \\
&\leq 2(|c_1| + |c_2| + |t_1|) - 1 = 2 \cdot |c| - 1.
\end{aligned}
$$

In all these cases, we get $|t^b|_{FC} = |t|_{FC}$ and $|c^b|_{FC} = |c|_{FC}$ by induction. We now discuss the running time of this procedure.

The decomposition of Proposition 6.1 can be found in time $O(|t|)$. Nevertheless, if t is a context, the application of Proposition 6.1 can give a big sub-term in $T(F, C)$ of size $O(a \cdot |t|)$, $a > 1/2$ and the two others of size at most $|t|/2$. Applying again Proposition 6.1 to the big sub-term will give three terms of size at most $|t|/2$. Then we can apply the same algorithm to at most five terms of size at most $|t|/2$. So, the total time is $O(|t| \cdot \log(|t|))$. This completes the proof.   ■

We can now state a proposition which shows how to transform a term in $SPE_t(F,C)$ into a term in $T(F',C')$ when (F',C') is (F,C)-flexible. The notion of flexibility makes it possible to eliminate the operations $\circ$ by using Condition (F2.3) that replaces $\circ$ by some function $f^{q,q'}$ in F'. Similarly for the operations $\bullet$ by using Conditions (F2.1) and (F2.2) which replace $\bullet$ by some function f^q in F'.

Proposition 6.2 *If (F',C') is (F,C)-flexible, then for every term t in $SPE_t(F,C)$, one can define a term $\tilde{t}$ in $T(F',C')$ that is equivalent to t and such that $|\tilde{t}|_{F'C'} = |t|_{FC}$ and $ht(\tilde{t}) \leq ht(t)$. Moreover, $\tilde{t}$ can be constructed in $O(|t|)$-time.*

We will first show how to deduce Theorem 6.1 from this proposition and Theorem 6.2.

Proof of Theorem 6.1. Let t be a term in $T(F,C)$ of size n. By using Theorem 6.2, we can construct in $O(n \cdot \log(n))$-time a term $t^b \in SPE_t(F,C)$ such that $Eval(t^b) = t$, $|t^b|_{FC} = |t|$ and $ht(t^b) \leq 3(\log(|t|) + 1)$.

By Proposition 6.2, one can transform t^b into $t' = \widetilde{t^b}$ such that $|t'| = |t'|_{F'C'} = |t^b|_{FC} = |t|$ and $ht(t') \leq ht(t^b) \leq 3 \cdot (\log(|t|) + 1)$. The term t' has exactly the same size as t, it is equivalent to t and is 3-balanced. Moreover, the construction can be done in $O(n \cdot \log(n))$-time. ∎

We can now prove Proposition 6.2.

Proof of Proposition 6.2. We define below two mappings:

$$t \mapsto \tilde{t} \qquad \text{and} \qquad c \mapsto \tilde{c}$$

where $t \in SPE_t(F,C)$ and $c \in SPE_c(F,C)$ such that $\tilde{t}, \tilde{c} \in T(F',C')$ and, $|\tilde{t}|_{F'C'} = |t|_{FC}$ and $|\tilde{c}|_{F'C'} = |c|_{FC} - 1$. We define them simultaneously by structural induction.

For $t \in SPE_t(F,C)$,

(i) If $t = b \in C$, then we let $\tilde{t} = b$.

(ii) If $t = f(t_1, t_2)$ and $f \in F$, then we let $\tilde{t} = f(\widetilde{t_1}, \widetilde{t_2})$.

(iii) If $t = c \bullet t_1$, then we let $\tilde{t} = f^q(\tilde{c}, \widetilde{t_1})$.

For $c \in SPE_c(F,C)$,

(iv) If $c = f(c', t)$ where $c' \in SPE_c(F,C)$, $t \in SPE_t(F,C)$ and $f \in F$, then we apply the following rules to $\tilde{f}(t, c')^2$, and, if $c = f(t, c')$, directly to c.

    (i') If $c = f(t, Id)$, then we let $\tilde{c} = \tilde{t}$.

    (ii') If $c = f(t, c')$ and $c' \in SPE_c(F,C)$, then we let $\tilde{c} = f^{q,q'}(\tilde{t}, \widetilde{c'})$ where $q = f(x_1, u)$ and $q' = Comb(c')$.

$^2\tilde{f}$ is defined by the commutativity condition on F.

(v) If $c = c_1 \circ c_2$, then we let $\tilde{c} = f^{q_1, q_2}(\widetilde{c_1}, \widetilde{c_2})$ where $q_1 = Comb(c_1)$ and $q_2 = Comb(c_2)$.

We now prove that $\tilde{t} \simeq t$.

Claim 6.1   *1. For every term $c \in SPE_c(F, C)$, if $Comb(c) = q(x_1, \ldots, x_n, u)$ and $seq(c) = (t_1, \ldots, t_n)$, then we have $\tilde{c} \simeq \hat{q}(t_1, \ldots, t_n)$.*

2. For every term $t \in SPE_t(F, C)$, we have $\tilde{t} \simeq t$.

Proof of Claim 6.1. We use the induction defining $\tilde{t}$ and $\tilde{c}$ for proving simultaneously *(1)* and *(2)*. Let $c \in SPE_c(F, C)$. Then

(a) The case of $c = f(c', t)$ follows by the induction hypothesis and the commutativity of F.

(b) Consider the case $c = f(t, Id)$. Then $q = Comb(c) = f(x_1, u)$, $f^q = f$, $\hat{q} = x_1$ and we have $\hat{q}(t) = t$, $\tilde{c} = \tilde{t}$ hence $\tilde{c} \simeq \hat{q}(t)$ by using *(2)* and induction.

(c) Consider the case $c = f(t, c')$, $c' \neq Id$. Let $q = Comb(c)$, $q' = Comb(c')$. We have $q = f(x_1, q'(x_2, x_3, \ldots, x_{n+1}, u))$ where we assume that q' has variables $(x_1, \ldots, x_n, u)$. Hence, $\tilde{c} = f^{r, q'}(\tilde{t}, \tilde{c'})$ where $r = f(x_1, u)$. Using induction, we have $t \simeq \tilde{t}$ and $\tilde{c'} \simeq \widehat{q'}(t_2, t_3, \ldots, t_{n+1})$ where $seq(c) = (t, t_2, \ldots, t_{n+1})$. By Condition (F2.3) of flexibility, we have

$$\hat{q} \simeq f^{r, q'}(x_1, \widehat{q'}(x_2, \ldots, x_{n+1})).$$

Hence,

$$\hat{q}(t, t_2, \ldots, t_{n+1}) \simeq f^{r, q'}(t, \widehat{q'}(t_2, \ldots, t_{n+1}))$$
$$\simeq f^{r, q'}(t, \widetilde{c'})$$
$$\simeq f^{r, q'}(\tilde{t}, \widetilde{c'})$$

using the induction (for $t \simeq \tilde{t}$). This completes the case.

(d) It remains the case $c = c_1 \circ c_2$. We let $Comb(c_i) = q_i$, $i = 1, 2$, $seq(c_1) = (t_1, \ldots, t_p)$, $seq(c_2) = (t_{p+1}, \ldots, t_n)$. Then $\tilde{c} = f^{q_1, q_2}(\widetilde{c_1}, \widetilde{c_2})$. By inductive hypothesis, $\widetilde{c_1} \simeq \widehat{q_1}(t_1, \ldots, t_p)$ and $\widetilde{c_2} \simeq \widehat{q_2}(t_{p+1}, \ldots, t_n)$. We also have $Comb(c) = q = q_1(x_1, \ldots, x_p, q_2(x_{p+1}, \ldots, x_n, u))$. By Condition (F2.3) of flexibility,

$$\hat{q}(t_1, \ldots, t_n) \simeq f^{q_1, q_2}(\widehat{q_1}(t_1, \ldots, t_p), \widehat{q_2}(t_{p+1}, \ldots, t_n)).$$

This gives $\tilde{c} \simeq \hat{q}(t_1, \ldots, t_n)$ as wanted.

Now we consider a term $t \in SPE_t(F, C)$.

(e) If $t \in C$, then $\tilde{t} = t$, hence $\tilde{t} \simeq t$.

(f) If $t = f(t_1, t_2)$, then $\tilde{t} = f(\widetilde{t_1}, \widetilde{t_2})$ and the desired equivalence follows since $t_i \simeq \widetilde{t_i}$ by inductive hypothesis.

(g) If $t = c \bullet t'$, then we have $Comb(c) = q(x_1, \ldots, x_n, u)$, $seq(c) = (t_1, \ldots, t_n)$, $t \simeq q(t_1, \ldots, t_n, t')$ and $\tilde{t} = f^q(\tilde{c}, \tilde{t'})$ by the definitions. By the inductive hypothesis, $\tilde{c} \simeq \hat{q}(t_1, \ldots, t_n)$ and by Condition (F2.2) of flexibility, $q(t_1, \ldots, t_n, t') \simeq f^q(\hat{q}(t_1, \ldots, t_n), t')$. Hence, $t \simeq \tilde{t}$ since by the inductive hypothesis $\tilde{t'} \simeq t'$. This completes the proof. ∎

The above definitions and Claim 6.1 give that $\tilde{t} \in T(F', C')$ is equivalent to t. It is also clear that $\tilde{t}$ is constructed in $O(|t|)$-time.

It remains to compare the sizes and heights of t and $\tilde{t}$, and c and $\tilde{c}$. We denote by $|t|_b$ the number of occurrences of a symbol b in a term or a context t. We can easily prove by induction that:

$$|\tilde{t}|_{F'C'} = |t|_{FC} \qquad \text{and} \qquad ht(\tilde{t}) \leq ht(t),$$
$$|\tilde{c}|_{F'C'} = |c|_{FC} - 1 \qquad \text{and} \qquad ht(\tilde{c}) \leq ht(c).$$

One can even prove that:

$$|\tilde{t}|_b = |t|_b \qquad \text{and} \qquad |\tilde{c}|_b = |c|_b \qquad \text{for each } b \in C.$$

The only case which does not yield equality of heights is case (i') in the definition of $\tilde{c}$. This ends the proof of the proposition. ∎

6.2 Applications to Graph Algebras

We apply in this section Theorem 6.1 to several graph complexity measures. It will suffice to check the flexibility condition for appropriate super-signatures of the signatures that define graphs of width at most k. As an application of Theorem 6.2 we obtain the following results collected in one theorem.

Theorem 6.3 *1. Every undirected graph of m-clique-width k is the value of a 3-balanced term in $T(F_{2k}^{um}, C_{2k}^{um})$.*

2. Every directed graph of m-clique-width k is the value of a 3-balanced term in $T(F_{3k}^{dm}, C_{3k}^{um})$.

3. Every undirected graph (resp. directed graph) of clique-width k is the value of a 3-balanced term in $T(F_{k'}^{uc}, C_{k'}^{c})$ (resp. $T(F_{k'}^{dc}, C_{k'}^{c})$) where $k' \leq k \cdot 2^k$ (resp. $k' \leq k \cdot 2^{2k}$).

4. Every graph of rank-width k is the value of a 3-balanced term in $T(R_{2k}^{GF(2)}, C_{2k}^{GF(2)})$.

5. Every graph of $GF(4)$-rank-width k is the value of a 3-balanced term in $T(R_{12k}^{GF(4)}, C_{12k}^{GF(4)})$.

   6. Every term in $T(B_k^F, C_k^F)$ is equivalent to a 3-balanced term $T(B_{k+q^k}^F, C_{k+q^k}^F)$.

   Some width measures have thus a better behavior than others with respect to the possibility of balancing terms without increasing width too much. By Theorem 6.1, we need only prove flexibility properties for each of the operations that define the widths considered in the theorem. We will prove each statement separately for clarity.

   Let u be a set of colors in $[k]$ (resp. a row vector). We denote by $u \upharpoonright n$ the set $u \cap [n]$ (resp. the restriction of u to the n first coordinates). It will be useful to use a new constant $\emptyset$ for denoting the empty graph (of any type). This constant will be eliminated at some stage of the proofs.

Proof of Theorem 6.3 (1). We will prove that $(F_{2k}^{um}, C_{2k}^{um})$ is (F_k^{um}, C_k^{um})-flexible. For clarity, we will denote the color $k + i$ as i', for each $i \in [k]$.

By definition, the signatures F_k^{um} and F_{2k}^{um} are commutative. We let $q(x_1, \ldots, x_n, u)$ be a comb-term $x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_n \otimes_n u)) \ldots)$ where $\otimes_i \in F_k^{um}$ for each i and $n \geq 2$.

Construction of $\hat{q}$ and $\otimes^q$. It is clear that the graph $q(G_1, \ldots, G_n, H)$ (for pairwise disjoint $2^{[k]}$-colored graphs $G_1, \ldots, G_n, H$) can be described as the (disjoint) union of $K = q(G_1, \ldots, G_n, \emptyset)$ and of $H' = q(\emptyset, \emptyset, \ldots, \emptyset, H)$ augmented with edges between V_K and $V_{H'} = V_H$, created by the operations of q. The objective is to have $q(G_1, \ldots, G_n, H) = K' \otimes_{R,g,h} H$ where K' is K with an additional coloring by $\delta' : V_K \to 2^{[k]'}$, that will indicate how should be linked the vertices of K and those of H.

The graph $K' = \hat{q}(G_1, \ldots, G_n)$ will be K with coloring function $\delta_{K'}$ defined by $\delta_{K'}(x) = \delta_K(x) \cup \delta'(x) \subseteq [k] \cup [k]'$. The mapping h can be defined as

$$Recol_{h_1} \circ Recol_{h_2} \circ \ldots \circ Recol_{h_n} = Recol_{h_1 \circ h_2 \circ \ldots \circ h_n}$$

where $\otimes_i = \otimes_{R_i, g_i, h_i}$ for each i. We now define δ'.

For each $i \in [k]$ and for some vertex $w \notin V_{G_1} \cup \ldots \cup V_{G_n}$, we let $\mathbf{i}(w)$ be the isolated vertex w colored by i. We let $T_i = q(G_1, \ldots, G_n, \mathbf{i}(w))$ and $N(T_i, w)$ be the set of vertices x of T_i, $x \neq w$ that are linked to w (are its neighbors). The auxiliary coloring δ' is defined by $\delta'(x) = \{i' \mid x \in N(T_i, w)\}$.

We let $\otimes^q$ be the operation $\otimes_{R,g,h_1 \circ \ldots \circ h_n}$ where $R = \{(i', i) \mid i \in [k]\}$, $g(j) = \{j\}$, $g(j') = \emptyset$ for $j \in [k]$. Letting K' be defined from $K = q(G_1, \ldots, G_n, \emptyset)$ and δ', we have

Claim 6.2 *For all $2^{[k]}$-colored graphs $G_1, \ldots, G_n$ and H, we have $q(G_1, \ldots, G_n, H) = K' \otimes^q H$.*

Proof of Claim 6.2. We let $G = q(G_1, \ldots, G_n, H)$. The vertices of G and of $K' \otimes^q H$ are the same, and they have the same colors, all in $[k]$, since g "erases" all colors from $[k]'$. The edges inside $G_1, \ldots, G_n$ and H are the same from the definitions of q, K' and $\otimes^q$. So are the edges between G_i and G_j, $j \neq i$ for the same reasons.

It remains to compare the edges between x in $V_{K'}$ and y in V_H. If xy is an edge in $K' \otimes^q H$, this means that $i' \in \delta_{K'}(x)$ and $i \in \delta_H(y)$ for some i, hence xw is in $q(G_1, \ldots, G_n, \mathbf{i}(w))$. From the definitions of the m-clique-width operations, x is linked to all y' in H such that $i \in \delta_H(y')$, and in particular to y. Hence xy in $G = q(G_1, \ldots, G_n, H)$. The proof is similar in the other direction. $\blacksquare$

Next, we must define $\hat{q} = x_1 \otimes'_1 (x_2 \otimes'_2 (\ldots (x_n \otimes'_n \emptyset) \ldots))$ in such a way that $K' = \hat{q}(G_1, \ldots, G_n)$. We recall that $\otimes_i = \otimes_{R_i, g_i, h_i}$ and we let $\otimes'_i = \otimes_{R_i, g'_i, h'_i}$ where $g'_i : [k] \to 2^{[k] \cup [k]'}$ is defined by

$$g'_i(j) = g_i(j) \cup \{l' \mid l \in [k], (j, m) \in R_i \text{ for some } m \in h_{i+1}(\ldots (h_{n-1}(h_n(l))) \ldots)\}$$

and $h'_i : [k] \cup [k]' \to 2^{[k] \cup [k]'}$ is defined by: $h'_i(j) = h_i(j)$, $h'_i(j') = \{j'\}$. Note the particular case $i = n$. We can take $R'_n = \emptyset$ instead of R_n, $h'_n(j) = \emptyset$ for all $j \in [k] \cup [k]'$.

Claim 6.3 *For all $2^{[k]}$-colored graphs $G_1, \ldots, G_n$, we have $K' = \hat{q}(G_1, \ldots, G_n)$.*

Proof of Claim 6.3. We let $K = q(G_1, \ldots, G_n, \emptyset)$ and K' be K with the additional coloring δ'. Hence, K' and $\hat{q}(G_1, \ldots, G_n)$ have the same vertices, with same colors from $[k]$ as one checks from the way the recolorings g'_i are defined.

We let $x \in G_i$ and we consider $\delta'(x) \subseteq [k]'$. Then $l' \in \delta'(x)$ if and only if x is linked to w in $q(G_1, \ldots, G_n, \mathbf{l}(w))$ which means that for some $(j, m) \in R_i$, we have $j \in \delta_{G_i}(x)$ and $m \in h_{i+1}(h_{i+2}(\ldots (h_n(l)) \ldots))$. Thus, $l' \in \delta'(x)$ implies $l' \in g'_i(j)$; hence l' is a color of x in $G_i \otimes'_i (G_{i+1} \otimes'_{i+1} (\ldots (G_n \otimes'_n \emptyset) \ldots))$. Since the recolorings h'_i do not modify the colors from $[k]'$, l' is a color of x in $\hat{q}(G_1, \ldots, G_n)$. The argument is similar in the other direction, hence the colors are the same in K' and in $\hat{q}(G_1, \ldots, G_n)$.

We now compare edges. Those with two ends in any G_i are the same in both graphs. Let $x \in V_{G_j}$ and $y \in V_{G_i}$ be linked in K', for $j < i$. This means that for some l and m, we have $(l, m) \in R_j$, $l \in \delta_{G_j}(x)$ and $m \in h_{j+1}(h_{j+2}(\ldots (h_{i-1}(g_i(\delta_{G_i}(y)))) \ldots))$. But, this edge also exists in $\hat{q}(G_1, \ldots, G_n)$ because $(l, m) \in R_j$ and $m \in h'_{j+1}(h'_{j+2}(\ldots h'_{i-1}(g'_i(\delta_{G_i}(y))) \ldots))$ since $h'_a \restriction [k] = h_a$ for each a. This concludes the proof. $\blacksquare$

Next we define the operations $\otimes^{q, q'}$ and verify Condition (F2.3) of flexibility.

Definition of the operations $\otimes^{q, q'}$. Let $q = x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_p \otimes_p u) \ldots))$ and $q' = x_{p+1} \otimes_{p+1} (x_{p+2} \otimes_{p+2} (\ldots (x_n \otimes_n u)) \ldots)$ so that $q'' = x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_n \otimes_n u)) \ldots)$. We let $\otimes^{q, q'} = \otimes_{R, g, h}$ where:

$$R = \{(i', i) \mid i \in [k]\},$$
$$g(i) = \{i\}, \quad g(i') = \{j' \mid i \in h_{q'}(j)\},$$
$$h(i) = h_q(i), \quad h(i') = \{i'\} \text{ for all } i \in [k],$$

and we recall that h_q is the composition of the recolorings defined by q and applied to u.

Claim 6.4 *For all $2^{[k]}$-colored graphs $G_1, \ldots, G_n$, we have*

$$\underbrace{\widehat{q''}(G_1, \ldots, G_n)}_{K''} = \underbrace{\hat{q}(G_1, \ldots, G_p)}_{K} \otimes^{q,q'} \underbrace{\widehat{q'}(G_{p+1}, \ldots, G_n)}_{K'}.$$

Proof of Claim 6.4. The vertices of K'' and of $K \otimes^{q,q'} K'$ are the same. We now compare the colors.

Let $x \in V_{G_i} \subseteq V_K$ with colors $\delta_K(x) \subseteq [k] \cup [k]'$. Its colors from $[k]$ are the same in K and in K'' by the various definitions. Let $j' \in \delta_{K''}(x)$. We have $x \in N(q''(G_1, \ldots, G_p, \ldots, G_n, \mathbf{j}(w)), w)$. Hence, for some $l \in h_{q'}(j)$, we have $l' \in \delta_K(x)$. Conversely, if $l' \in \delta_K(x)$ and $l \in h_{q'}(j)$, then we have $x \in N(q(G_1, \ldots, G_p, \mathbf{l}(w)), w)$. Hence, $x \in N(q''(G_1, \ldots, G_p, \ldots, G_n, \mathbf{j}(w)), w)$ and $j' \in \delta_{K''}(x)$. It follows that

$$\begin{aligned}
\delta_{K''}(x) \cap [k]' &= \{j' \mid l \in h_{q'}(j) \text{ for some } l' \in \delta_K(x)\} \\
&= \bigcup_{l' \in \delta_K(x)} \{j' \mid l \in h_{q'}(j)\} \\
&= g(\delta_K(x)) \cap [k]' \text{ since } g(l') = \{j' \mid l \in h_{q'}(j)\}.
\end{aligned}$$

We now compare the colors in K'' and in K' of $x \in V_{G_i} \subseteq V_{K'}$, $p + 1 \leq i \leq n$. Let $j \in \delta_{K'}(x) \cap [k]$. Then x has color j in $q'(G_{p+1}, \ldots, G_n, \emptyset)$, hence all colors from $h_q(j)$ in the graph $q''(G_1, \ldots, G_n, \emptyset)$, whence also in K''. Conversely, if x has colors l in K'', it must have in K' some color j such that $l \in h_q(j)$. Hence, $\delta_{K''}(x) \cap [k] = h_q(\delta_{K'}(x))$ as was to be proved. Consider now $j' \in \delta_{K'}(x)$. Then $x \in N(q'(G_{p+1}, \ldots, G_n, \mathbf{j}(w)), w)$. But, we also have $x \in N(q''(G_1, \ldots, G_n, \mathbf{j}(w)), w)$. Hence, $j' \in \delta_{K''}(x) \cap [k]'$. A similar argument gives $\delta_{K''}(x) \cap [k]' \subseteq \delta_{K'}(x) \cap [k]'$, hence $\delta_{K''}(x) \cap [k]' = \delta_{K'}(x) \cap [k]'$ as was to be proved. Hence, $\delta_{K''}(x) = h \circ \delta_{K'}(x)$.

We now compare the edges of K'' and of $K \otimes_{q,q'} K'$. Clearly, those with two ends in K or in K' are the same in both graphs. We consider $x \in V_{G_j} \subseteq V_K$ and $y \in V_{G_{j'}} \subseteq V_{K'}$, for $1 \leq j \leq p < j' \leq n$. Assume xy in K''. We also have xy in $q''(G_1, \ldots, G_n, \emptyset) = q(G_1, \ldots, G_p, \underbrace{q'(G_{p+1}, \ldots, G_n, \emptyset)}_{G'})$. We have $i \in \delta_{G_j}(x)$ and $l \in \delta_{G'}(y)$, $l' \in \delta_K(x)$ and also $l \in \delta_{K'}(y)$ since the colors from $[k]$ are the same in G' and in $\widehat{q'}(G_{p+1}, \ldots, G_n) = K'$. Hence, xy is an edge in $K \otimes^{q,q'} K'$. Conversely, if xy in $K \otimes^{q,q'} K'$, then $l' \in \delta_K(x)$, $l \in \delta_{K'}(y)$ for some l and we obtain that xy is an edge in K''. This completes the proof that $K'' = K \otimes^{q,q'} K'$.   ∎

It remains to eliminate the constant $\emptyset$. Let k be a positive integer. For any $R \subseteq [k] \times [k]$, $g, h : [k] \rightarrow 2^{[k]}$ and undirected $2^{[k]}$-colored graphs G and H, it is clear that:

1. $G \otimes_{R,g,h} \emptyset$ is equivalent to $g(G)$ where $g(G)$ is the graph G where each x is now labeled with $g \circ lab_G(x)$.

2. $g(G) \otimes_{R,g',h} H$ is equivalent to $G \otimes_{R',g' \circ g,h} H$ where $R' = \{(i,j) \mid g(i) \times j \cap R \neq \emptyset\}$.

By using the two rules above, one can eliminate the constant $\emptyset$. This finishes the proof of Theorem 6.3 (1). ∎

We now prove Theorem 6.3(2).

Proof of Theorem 6.3(2). Let G be a directed graph of m-clique-width k. Then G is the value of some term t in $T(F_k^{dm}, C_k^{um})$. Let us explain how to construct the operations f^q and $f^{q,q'}$.

Let $q(x_1, \ldots, x_n, u)$ be a comb-term $x_1 \otimes_1 (x_2 \otimes_2 (\cdots (x_n \otimes_n u) \cdots))$ where for each $i \leq n$ the operation $\otimes_i = \otimes_{R_i, R'_i, g_i, h_i}$ is in F_k^{dm}. As in Theorem 6.3(1), for each $x \in V_{G_i}$, the idea is to store in $\hat{q}(G_1, \ldots, G_n)$ not only $\{j \mid (\delta_{G_i}(x) \times h_{i+1} \circ \cdots \circ h_n(j)) \cap R_i \neq \emptyset\}$ but also $\{j \mid (\delta_{G_i}(x) \times h_{i+1} \circ \cdots \circ h_n(j)) \cap R'_i \neq \emptyset\}$. For that purpose, we will use two isomorphic copies of $\{1, \ldots, k\}$, that we denote by $\{1', \ldots, k'\}$ and $\{1'', \ldots, k''\}$. We let $\otimes^q = \otimes_{R,R',g,h}$ where

$$g(j) = \{j\}, \qquad\qquad g(j') = g(j'') = \emptyset,$$
$$R = \{(i',i) \mid i \in [k]\}, \qquad R' = \{(i'',i) \mid i \in [k]\},$$
$$h = h_1 \circ \cdots \circ h_n.$$

One can verify by using the same techniques as in Theorem 6.3(1) that for every $2^{[k]}$-colored directed graphs $G_1, \ldots, G_n$ and H, we have $q(G_1, \ldots, G_n, H) = \hat{q}(G_1, \ldots, G_n) \otimes^q H$.

Similarly, for every two comb-terms $q = x_1 \otimes_1 (x_2 \otimes_2 (\cdots (x_p \otimes_p u) \cdots))$ and $q = x_1 \otimes_{p+1} (x_{p+2} \otimes_{p+2} (\cdots (x_n \otimes_n u) \cdots))$ so that $q'' = x_1 \otimes_1 (x_2 \otimes_2 (\cdots (x_n \otimes_n u) \cdots))$, we let $\otimes^{q,q'} = \otimes_{R,R',g,h}$ where:

$$R = \{(i',i) \mid i \in [k]\}, \qquad R' = \{(i'',i) \mid i \in [k]\}$$
$$g(i) = \{i\}, \; g(i') = \{j' \mid i \in h_{q'}(j)\}, \; g(i'') = \{j'' \mid i \in h_{q'}(j)\}$$
$$h(i) = h_q(i), \; h(i') = \{i'\}, \; h(i'') = \{i''\} \text{ for all } i \in [k],$$

where $h_q = h_1 \circ \cdots \circ h_p$ and $h_{q'} = h_{p+1} \circ \cdots \circ h_n$.

Again one can verify that for every $G_1, \ldots, G_n$, we have $\widehat{q''}(G_1, \ldots, G_n) = \widehat{q}(G_1, \ldots, G_p) \otimes^{q,q'} \widehat{q'}(G_{p+1}, \ldots, G_n)$. ∎

We now prove Theorem 6.3 (3). See Definitions 5.1 and 5.2 for the definitions of the operations in F_k^u and F_k^d.

Proof of Theorem 6.3 (3). We prove it first for undirected graphs. Let G be an undirected graph of clique-width k. Then G is the value of a term in $T(F_k^{uc}, C_k^c)$. By Lemma 5.1, G is also the value of a term t in $T(F_k^u, C_k^c)$ which is an m-clique-width expression.

By the proof Theorem 6.3 (1), one can transform t into a 3-balanced term $\tilde{t}$ with the particularity that for each sub-terms t'' of $\tilde{t}$ and for each vertex $x \in V_H$, $H = val(t'')$, we have

$\delta_H(x) = \{i\} \cup L$ where $i \in [k]$ and $L \subseteq [k]'$. By Proposition 1.1, we can transform $\tilde{t}$ into a clique-width expression of width at most $k \cdot 2^k$.

For the directed case, by Lemma 5.2, if $cwd(G) = k$, then G is the value of a term t in $T(F_k^d, C_k^c)$. By the proof of Theorem 6.3(2), t can be transformed into a 3-balanced term $\tilde{t}$ in $T(F_k^{dm}, C_k^{um})$ with the particularity that for each sub-terms t'' of $\tilde{t}$ and for each vertex $x \in V_H$, $H = val(t'')$, we have $\delta_H(x) = \{i\} \cup L_1 \cup L_2$ where $i \in [k]$ and $L_1 \subseteq [k]'$ and $L_2 \subseteq [k]''$. By Proposition 1.1, we can transform $\tilde{t}$ into a clique-width expression of width at most $k \cdot 2^{2k}$. ∎

We now prove Theorem 6.3 for the case of rank-width.

Proof of Theorem 6.3 (4). We denote by $0_{k,\ell}$ the $(k \times \ell)$-null matrix and by $I_{k,\ell}$ the $(k \times \ell)$-identity matrix. We will prove that $(R_{2k}^{GF(2)}, C_{2k}^{GF(2)})$ is $(R_k^{GF(2)}, C_k^{GF(2)})$-flexible. We let $\emptyset_k$ denote the $GF(2)^k$-colored null graph.

The signatures $R_k^{GF(2)}$ and $R_{2k}^{GF(2)}$ are commutative from Remark 3.3. We let $q(x_1, \ldots, x_n, u)$ be a comb-term $x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_n \otimes_n u)) \ldots)$ where $\otimes_i = \otimes_{M_i, N_i, P_i} \in R_k^{GF(2)}$ for each i and $n \geq 2$. We recall that we can consider M_i, N_i and P_i as $(k \times k)$-matrices.

Construction of $\hat{q}$ and of $\otimes^q$. We let $\hat{q} = x_1 \otimes_1' (x_2 \otimes_2' (\ldots (x_n \otimes_n' \emptyset_k)) \ldots)$ where $\otimes_i' = \otimes_{M_i', N_i', P_i'}$ with M_i', N_i' are $(k \times 2k)$-matrices and P_i' are $(2k \times 2k)$-matrices and such that:

$$M_i' = \begin{pmatrix} M_i & 0_{k,k} \end{pmatrix} \qquad N_i' = \begin{pmatrix} N_i & M_i \cdot Q_i^T \end{pmatrix} \qquad P_i' = \begin{pmatrix} P_i & 0_{k,k} \\ 0_{k,k} & I_{k,k} \end{pmatrix}$$

where $Q_i = P_n \cdot P_{n-1} \cdot \ldots \cdot P_{i+1}$ for $i = 1, \ldots, n$. We let $\otimes^q = \otimes_{M,N,P}$ where:

$$M = \begin{pmatrix} 0_{k,k} \\ I_{k,k} \end{pmatrix} \qquad N = \begin{pmatrix} I_{k,k} \\ 0_{k,k} \end{pmatrix} \qquad P = P_n \cdot P_{n-1} \cdot \ldots \cdot P_1.$$

Claim 6.5 *For all disjoint $GF(2)^k$-colored graphs $G_1, \ldots, G_n$ and H, we have*

$$\underbrace{q(G_1, \ldots, G_n, H)}_{G} = \underbrace{\hat{q}(G_1, \ldots, G_n)}_{K} \otimes^q H.$$

Proof of Claim 6.5. We let $G' = K \otimes^q H$. Clearly, the vertices of G and of G' are the same. We first verify that $\gamma_G(x) = \gamma_{G'}(x)$ for each vertex x. Let $y \in V_H$. By definition of q, we have $\gamma_G(y) = \gamma_H(y) \cdot P_n \cdot \ldots \cdot P_1$. Then it is clear that $\gamma_G(y) = \gamma_{G'}(y)$ since $P = P_n \cdot P_{n-1} \cdot \ldots \cdot P_1$.

Now, let $x \in V_{G_i}$. We have $\gamma_G(x) = \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1$. By definition of $\hat{q}$, we have $\gamma_K(x) = \begin{pmatrix} \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1 & \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot P_{n-1} \cdot \ldots \cdot P_{i+1})^T \end{pmatrix}$. It is then clear that $\gamma_{G'}(x) = \gamma_K(x) \cdot N = \gamma_G(x)$.

We now compare the edges of G and of G'. It is clear that the edges inside $G_1, \ldots, G_n$ and H are the same in G and in G'. So are the edges between G_i and G_j, $j \neq i$. It remains to

compare the edges between $x \in V_K$ and $y \in V_H$. Let $x \in V_{G_i}$ and $y \in V_H$. It is sufficient to prove that $\gamma_K(x) \cdot M \cdot \gamma_H(y)^T = \gamma_{G_i}(x) \cdot M_i \cdot (\gamma_H(y) \cdot P_n \cdot P_{n-1} \cdot \ldots \cdot P_{i+1})^T$. We have by definition of $\hat{q}$ that $\gamma_K(x) = \big(\gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots P_1 \quad \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot P_{n-1} \cdot \ldots \cdot P_{i+1})^T\big)$. Hence,

$$\gamma_K(x) \cdot M \cdot \gamma_H(y)^T = \gamma_K(x) \cdot \begin{pmatrix} 0_{k,k} \\ I_{k,k} \end{pmatrix} \cdot \gamma_H(y)^T$$
$$= \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot \ldots \cdot P_{i+1})^T \cdot \gamma_H(y)^T.$$

This finishes the proof of the claim.   ∎

Definition of the Operations $\otimes^{q,q'}$. Let $q = x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_p \otimes_p u))\ldots)$ and $q' = x_{p+1} \otimes_{p+1} (x_{p+2} \otimes_{p+2} (\ldots (x_n \otimes_n u))\ldots)$ so that $q'' = x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_n \otimes_n u))\ldots)$. We let $\otimes^{q,q'} = \otimes_{M,N,P}$ where:

$$M = \begin{pmatrix} 0_{k,k} & 0_{k,k} \\ I_{k,k} & 0_{k,k} \end{pmatrix} \qquad N = \begin{pmatrix} I_{k,k} & 0_{k,k} \\ 0_{k,k} & Q_{q'}^T \end{pmatrix} \qquad P = \begin{pmatrix} Q_q & 0_{k,k} \\ 0_{k,k} & I_{k,k} \end{pmatrix}$$

where $Q_q = P_p \cdot P_{p-1} \cdot \ldots P_1$ and $Q_{q'} = P_n \cdot P_{n-1} \cdot \ldots \cdot P_{p+1}$.

Claim 6.6 *For all disjoint $GF(2)^k$-colored agraphs $G_1, \ldots, G_n$, we have*

$$\underbrace{\widehat{q''}(G_1, \ldots, G_n)}_{K''} = \underbrace{\hat{q}(G_1, \ldots, G_p)}_{K} \otimes^{q,q'} \underbrace{\widehat{q'}(G_{p+1}, \ldots, G_n)}_{K'}.$$

Proof of Claim 6.6. Clearly, the vertices of K'' and of $K \otimes^{q,q'} K$ are the same. We now compare their colors. Let $G = K \otimes^{q,q'} K'$. By the definition of $\widehat{q''}$, for each $x \in V_{G_i}$, for $1 \le i \le n$, we have $\gamma_{K''}(x) = \big(\gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots P_1 \quad \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot \ldots \cdot P_{i+1})^T\big)$. We now prove that $\gamma_G(x) = \gamma_{K''}(x)$.

Let $x \in V_{G_i} \subseteq V_K$, for $1 \le i \le p$. By definition of $\hat{q}$, we have $\gamma_K(x) = \big(\gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1 \quad \gamma_{G_i}(x) \cdot M_i \cdot (P_p \cdot \ldots \cdot P_{i+1})^T\big)$. Hence,

$$\gamma_G(x) = \gamma_K(x) \cdot N$$
$$= \begin{pmatrix} \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1 \\ \gamma_{G_i}(x) \cdot M_i \cdot (P_p \cdot \ldots \cdot P_{i+1})^T \end{pmatrix}^T \cdot \begin{pmatrix} I_{k,k} & 0_{k,k} \\ 0_{k,k} & Q_{q'}^T \end{pmatrix}$$
$$= \begin{pmatrix} \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1 \\ \gamma_{G_i}(x) \cdot M_i \cdot (P_p \cdot \ldots \cdot P_{i+1})^T \cdot (P_n \cdot \ldots \cdot P_{p+1})^T \end{pmatrix}$$
$$= \gamma_{K''}(x).$$

We now consider $x \in V_{G_i} \subseteq V_{K'}$, for $p + 1 \leq i \leq n$. By definition of $\widehat{q'}$, we have $\gamma_{K'}(x) = \left(\gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_{p+1} \quad \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot \ldots \cdot P_{i+1})^T \right)$. Hence,

$$
\begin{aligned}
\gamma_G(x) &= \gamma_{K'}(x) \cdot P \\
&= \begin{pmatrix} \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_{p+1} \\ \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot \ldots \cdot P_{i+1})^T \end{pmatrix} \cdot \begin{pmatrix} Q_q & 0_{k,k} \\ 0_{k,k} & I_{k,k} \end{pmatrix} \\
&= \begin{pmatrix} \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_{p+1} \cdot P_p \cdot \ldots \cdot P_1 \\ \gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot \ldots \cdot P_{i+1})^T \end{pmatrix} \\
&= \gamma_{K''}(x).
\end{aligned}
$$

We now compare the edges of K'' and of $K \otimes^{q,q'} K'$. Clearly, those with two ends in K or in K' are the same in both graphs. We consider $x \in V_{G_j} \subseteq V_K$ and $y \in V_{G_{j'}} \subseteq V_{K'}$, for $1 \leq j \leq p < j' \leq n$. By definition, we have an arc xy in K'' if and only if we have an arc in $q(G_1, \ldots, G_n, \emptyset)$. Then it is sufficient to prove that $\gamma_K(x) \cdot M \cdot \gamma_{K'}(y)^T = \gamma_{G_i}(x) \cdot M_i \cdot (\gamma_{G_j}(y) \cdot N_j \cdot P_{j-1} \cdot \ldots \cdot P_{i+1})^T$. By definition,

$$
\begin{aligned}
\gamma_K(x) \cdot M \cdot \gamma_{K'}(y)^T &= \begin{pmatrix} \gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1 \\ \gamma_{G_i}(x) \cdot M_i \cdot (P_p \cdot \ldots \cdot P_{i+1})^T \end{pmatrix}^T \cdot \begin{pmatrix} 0_{k,k} & 0_{k,k} \\ I_{k,k} & 0_{k,k} \end{pmatrix} \cdot \begin{pmatrix} \gamma_{G_j}(y) \cdot N_j \cdot P_{j-1} \cdot \ldots \cdot P_{p+1} \\ \gamma_{G_j}(y) \cdot M_j \cdot (P_n \cdot \ldots \cdot P_{j+1})^T \end{pmatrix}^T \\
&= \gamma_{G_i}(x) \cdot M_i \cdot (P_p \cdot \ldots \cdot P_{i+1})^T \cdot (\gamma_{G_j}(y) \cdot N_j \cdot P_{j-1} \cdot \ldots \cdot P_{p+1}^T) \\
&= \gamma_{G_i}(x) \cdot M_i \cdot (\gamma_{G_j}(y) \cdot N_j \cdot P_{j-1} \cdot \ldots \cdot P_{i+1})^T.
\end{aligned}
$$

This completes the proof of the claim.   ∎

By Remark 3.3, one can eliminate the constant $\emptyset_k$ in the constructed term. This terminates the proof of Theorem 6.3 (3).   ∎

We now prove Theorem 6.3 (5).

Proof of Theorem 6.3 (5). Let G be a directed graph of $GF(4)$-rank-width k. By Proposition 3.2, we have $brwd(G) \leq 4k$. Then G is the value of a term in $T(U_{4k}^{([2])}, C_{4k}^{GF(2)})$ by Proposition 4.8. It is easy to see that this term can be transformed into a 3-balanced term $\tilde{t}$ in $T(U_{12k}^{([2])}, C_{12k}^{GF(2)})$ by using the proof of Theorem 6.3 (3). Let us explain in few words the idea. Let $q(x_1, \ldots, x_n, u)$ be a comb-term $x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_n \otimes_n u) \ldots))$ where $\otimes_i = \otimes_{M_i, M_i', N_i, P_i}$. For each $x \in V_{G_i}$, the idea is to store in $\hat{q}(G_1, \ldots, G_n)$ not only $\gamma_{G_i}(x) \cdot N_i \cdot P_{i-1} \cdot \ldots \cdot P_1$ and $\gamma_{G_i}(x) \cdot M_i \cdot (P_n \cdot \ldots \cdot P_{i+1})^T$ but also $\gamma_{G_i}(x) \cdot M_i' \cdot (P_n \cdot \ldots \cdot P_{i+1})^T$. I omit all the technical proofs which are not of any interest since they are the same as in the proof of Theorem 6.3 (3).

One can verify by using Lemma 3.3 that we can transform $\tilde{t}$ into a term in $T(R_{12k}^{GF(4)}, C_{12k}^{GF(4)})$ that generates the same graph by transforming each operation $\otimes_{M_1, M_2, N, P}$ into an operation $\otimes_{M, N, P}$ where $M = a \cdot M_1 + a^2 \cdot M_2$. This terminates the proof.   ∎

We can now prove Theorem 6.3 (6).

Proof of Theorem 6.3 (6). Let F be a fixed field with q elements. Let $\alpha : F^k \to [q^k]$ be a bijective function that enumerates the set F^k. We claim that $(B^F_{k+q^k}, C^F_{k+q^k})$ is (B^F_k, C^F_k)-flexible. Recall that by Remark 3.1 (2), $(B^F_{k+q^k}, C^F_{k+q^k})$ and (B^F_k, C^F_k) are commutative. We will also use the constant $\emptyset_k$ which will be eliminated by Remark 3.1 (3).

We let $q(x_1, \ldots, x_n, u)$ be a comb-term $x_1 \otimes_1 (x_2 \otimes_2 (\ldots (x_n \otimes_n u)) \ldots)$ where $\otimes_i = \otimes_{f_i, g_i, h_i} \in B^F_k$ for each i and $n \geq 2$.

Construction of $\hat{q}$ and of $\otimes^q$. We let $\hat{q} = x_1 \otimes'_1 (x_2 \otimes'_2 (\ldots (x_n \otimes'_n \emptyset_k)) \ldots)$ where $\otimes'_i = \otimes_{f'_i, g'_i, h'_i}$ with $f'_i : F^k \times F^{k+q^k} \to \{0,1\}$, $g'_i : F^k \to F^{k+q^k}$ and $h'_i : F^{k+q^k} \to F^{k+q^k}$ are such that:

$$f'_i(u, v) = f_i(u, v \restriction k),$$

$$g'_i(u) = w \quad \text{where} \quad \begin{cases} w[j] = g_i(u)[j] & \text{if } 1 \leq j \leq k, \\ w[\alpha(z) + k] = f_i(u, v) & \text{where } v = h_{i+1}(h_{i+2}(\ldots (h_n(z))) \ldots), \end{cases}$$

$$h'_i(v) = w \quad \text{where} \quad w[j] = \begin{cases} h_i(v \restriction k)[j] & \text{if } 1 \leq j \leq k, \\ v[j] & \text{otherwise.} \end{cases}$$

Note the particular case $i = n$. We can take $f'_n(u, v) = 0$ and $h'_n(v) = (0, 0, \ldots, 0)$ for all $v \in F^{k+q^k}$.

We let $\otimes^q = \otimes_{f,g,h}$ where $f : F^{k+q^k} \times F^k \to \{0,1\}$, $g : F^{k+q^k} \to F^k$ and $h : F^k \to F^k$ are defined as follows:

$$f(u, v) = u[\alpha(v) + k],$$
$$g(u) = u \restriction k,$$
$$h(v) = h_1(h_2(\ldots (h_n(v))) \ldots).$$

Claim 6.7 *For all disjoint F^k-colored graphs $G_1, \ldots, G_n$ and H, we have*

$$\underbrace{q(G_1, \ldots, G_n, H)}_{G} = \underbrace{\hat{q}(G_1, \ldots, G_n)}_{K} \otimes^q H.$$

Proof of Claim 6.7. We let $G' = K \otimes^q H$. The proof is as for the previous similar claims. We first compare the edges between $x \in V_{G_i}$ and $y \in H$. We have:

$$\begin{aligned} xy \in E_G &\iff f_i(\gamma_{G_i}(x), h_{i+1}(h_{i+2}(\ldots (h_n(\gamma_H(y))) \ldots))) = 1 \\ &\iff g'_i(\gamma_{G_i}(x))[\alpha(\gamma_H(y)) + k] = 1 \quad \text{(by definition of } \otimes'_i) \\ &\iff \gamma_K(x)[\alpha(\gamma_H(y)) + k] = 1 \quad \text{(by definition of } \otimes'_j) \text{ for } 1 \leq j < i \\ &\iff xy \in E_{G'} \quad \text{(by definition of } \otimes^q). \end{aligned}$$

One can easily verify that x has the same color in G and in G'. ∎

Definition of the Operations $\otimes^{q,q'}$ Let $q = x_1 \otimes_1 (x_2 \otimes_2 (\dots (x_p \otimes_p u)) \dots)$ and $q' = x_1 \otimes_{p+1} (x_{p+2} \otimes_{p+2} (\dots (x_n \otimes_n u)) \dots)$ so that $q'' = x_1 \otimes_1 (x_2 \otimes_2 (\dots (x_n \otimes_n u)) \dots)$. We let $\otimes^{q,q'}$ be $\otimes_{f,g,h}$ (in the definition z denotes elements of F^k):

$$f(u,v) = u[\alpha(v \restriction k) + k],$$

$$g(u) = w \quad \text{where} \quad \begin{cases} w[j] = u[j] & \text{if } 1 \leq j \leq k, \\ w[\alpha(z) + k] = u[\alpha(v) + k] & \text{where } v = h_{q'}(z), \end{cases}$$

$$h(v) = w \quad \text{where} \quad w[j] = \begin{cases} h_q(v \restriction k)[j] & \text{if } 1 \leq j \leq k, \\ v[j] & \text{otherwise.} \end{cases}$$

and $h_q = h_1 \circ h_2 \circ \dots \circ h_p$ and $h_{q'} = h_{p+1} \circ h_{p+2} \circ \dots \circ h_n$.

Claim 6.8 *For all disjoint F^k-colored graphs $G_1, \dots, G_n$, we have*

$$\underbrace{\widehat{q''}(G_1, \dots, G_n)}_{K''} = \underbrace{\hat{q}(G_1, \dots, G_p)}_{K} \otimes^{q,q'} \underbrace{\widehat{q'}(G_{p+1}, \dots, G_n)}_{K'}.$$

Proof of Claim 6.8. It is clear that the vertices of K'' and of $G = K \otimes^{q,q'} K'$ are the same. We first compare the colors.

Let $x \in V_{G_i} \subseteq V_K$, $1 \leq i \leq p$. From the definition of $\widehat{q''}$, $\gamma_{K''}(x) = w$ where

$$w[j] = h_1(h_2(\dots (h_{i-1}(g_i(\gamma_{G_i}(x)))) \dots))[j] \qquad\qquad \text{if } 1 \leq j \leq k,$$
$$w[\alpha(z) + k] = f_i(\gamma_{G_i}(x), v) \qquad\qquad \text{where } v = h_{i+1}(h_{i+2}(\dots (h_n(z))) \dots).$$

By the definition of $\hat{q}$, we have $\gamma_K(x) = w$ where w is defined as above with $n = p$. It is then clear from the definition of $\otimes^{q,q'}$ that $\gamma_{K''}(x) \restriction k = \gamma_G(x) \restriction k$. We thus have

$$\gamma_G(x)[\alpha(z) + k] = \gamma_K(x)[\alpha(v) + k] \qquad\qquad \text{where } v = h_{q'}(z) \text{ (definition of } \otimes^{q,q'})$$
$$= f_i(\gamma_{G_i}(x), v') \qquad\qquad \text{where } v' = h_{i+1}(h_{i+2}(\dots (h_p(h_{q'}(z)))) \dots)$$
$$= \gamma_{K''}(x).$$

Let $y \in V_{G_i} \subseteq V_{K'}$, $p+1 \leq i \leq n$. By the definition of $\widehat{q''}$ we have $\gamma_{K''}(y) = w$ where

$$w[j] = h_1(h_2(\dots (h_{i-1}(g_i(\gamma_{G_i}(x)))) \dots))[j] \qquad\qquad \text{if } 1 \leq j \leq k,$$
$$w[\alpha(z) + k] = f_i(\gamma_{G_i}(y), v) \qquad\qquad \text{where } v = h_{i+1}(h_{i+2}(\dots (h_n(z))) \dots).$$

By the definition of $\widehat{q'}$, we have $\gamma_K(y) = w$ where

$$w[j] = h_{p+1}(\dots (h_{i-1}(g_i(\gamma_{G_i}(x)))) \dots)[j] \qquad\qquad \text{if } 1 \leq j \leq k,$$
$$w[\alpha(z) + k] = f_i(\gamma_{G_i}(y), v) \qquad\qquad \text{where } v = h_{i+1}(h_{i+2}(\dots (h_n(z))) \dots).$$

By the definition of $\otimes^{q,q'}$, we have $\gamma_G(y) = w$ where

$$w[j] = h_1(h_2(\dots (h_p(\gamma_{K'}(x) \restriction k)) \dots))[j] \qquad\qquad \text{if } 1 \leq j \leq k,$$
$$w[\alpha(z) + k] = f_i(\gamma_{G_i}(y), v) \qquad\qquad \text{where } v = h_{i+1}(h_{i+2}(\dots (h_n(z))) \dots).$$

Hence, $\gamma_{K''}(x) = \gamma_G(x)$.

We now compare the edges of K'' and of G. Clearly, those with two ends in K' or in K are the same in the two graphs. We consider $x \in V_{G_j} \subseteq V_K$ and $y \in V_{G_{j'}} \subseteq V_{K'}$, for $1 \leq j \leq p < j' \leq n$. Assume that xy in K''. We also have xy in

$$q''(G_1, \ldots, G_n, \emptyset_k) = q(G_1, \ldots, G_p, \underbrace{q'(G_{p+1}, \ldots, G_n, \emptyset_k)}_{G'}).$$

Then

$$
\begin{aligned}
xy \text{ in } K'' &\iff f_j(\gamma_{G_j}(x), h_{j+1}(\ldots(h_p(\gamma_{G'}(y)))\ldots)) = 1 \\
&\iff \gamma_K(x)[\alpha(\gamma_{G'}(y) \upharpoonright k) + k] = 1 \\
&\iff \gamma_K(x)[\alpha(\gamma_{K'}(y) \upharpoonright k) + k] = 1 \\
&\iff xy \text{ in } G \text{ (by definition of } \otimes^{q,q'}).
\end{aligned}
$$

This terminates the proof of the claim.   ■

By Using Remark 3.1 (3) we can remove the constant $\emptyset$ which terminates the proof of Theorem 6.3 (6).   ■

Remark 6.3 For the proof of Theorem 6.3 (6) we need colors in $F^{k+q^k} \simeq F^k \times F^{q^k}$ instead of F^{2k} because, for instance if $G = q(G_1, \ldots, G_n, H)$, for each $x \in V_{G_i}$, we need to store the set of colors $v \in F^k$ such that $f_i(\gamma_{G_i}(x), h_{i+1}(h_{i+2}(\ldots(h_n(v))\ldots)))) = 1$ to be able to construct $\otimes^q$. And $|\{u \in F^k\}| = q^k$. We do not know any algebraic structure like the lattice of subsets or the vector space on F^k that makes possible to express this information in a more compact way.

6.3 Conclusion

We have defined a framework that unifies all known balancing theorems and we give new results, especially for rank-width of undirected graphs.

In [Cou93] Courcelle defined a set of graph operations, denoted by HR_k, and a set of constants C_k^t such that a graph G has tree-width at most k if and only if G is the value of a term in $T(HR_{k+1}, C_{k+1}^t)$. By using terms in $T(HR_k, C_k^t)$, we can prove that every graph of tree-width at most k is the value of a 3-balanced term in $T(HR_{3k-1}, C_{3k-1}^t)$. However, the proof is not immediate since we need to transform the operations HR_k into binary operations suitable for our framework.

For the case of branch-width of graphs (see Definition 1.8), we can define a set of binary graph operations F_k^b in the spirit of the operations HR_k and prove that a graph has branch-width at most k if and only if it is the value of a term in $T(F_k^b, C_k^t)$. By using this term, we can prove, by using our framework, that every graph of branch-width at most k is the value of a 3-balanced term in $T(F_{2k}^b, C_{2k}^t)$.

We finish this chapter with two questions:

(Q6.1) In our balancing results we obtain 3-balanced terms by using Theorem 6.2. Can we improve the 3 in Theorem 6.2?

(Q6.2) The time complexity of our balancing results also depend on Theorem 6.2. Can we prove that the algorithm of Theorem 6.1 works in $O(n)$-time in most cases or find an alternative linear-time algorithm?

Chapter 7

Rank-width Compared to Tree-width

Tree-width is a well-known graph parameter because of the many positive results it yields. Every MS-definable property admits a linear-time algorithm on graph classes of bounded tree-width [Cou90, ALS91] and finite graph classes that have decidable MS theory have bounded tree-width. Moreover, it plays an important role in structural graph theory, particularly in the fundamental series of papers on graph minors by Robertson and Seymour [RS83]-[RS04]. Clique-width has also positive algorithmic results [CMR00] and its equivalent graph parameter rank-width has positive structural results [Oum05b, Oum05c, Oum08a]. It is thus natural to compare tree-width and, rank-width and its equivalent parameters. It is known that if an undirected graph has tree-width k, then it has clique-width at most $3 \cdot 2^{k-1} - 1$ [CR05] and if an undirected graph has rank-width ℓ, then it has clique-width at most $2^{\ell+1} - 1$ [OS06]. We can thus suspect a linear relation between tree-width and rank-width. And because rank-width and clique-width are equivalent graph parameters, this linear relation could be necessarily of the form:

> if an undirected graph has tree-width k, then it has rank-width at most $a \cdot k + b$, where a and b are fixed constants.

For proving this, we developed a technique of independent interest: simulation of edge contractions, one of the operations that yield minor inclusion, by means of local complementations and vertex deletions (the operations that define vertex-minor inclusion). We prove that if an undirected graph has tree-width k, then it has rank-width at most $4 \cdot k + 2$. However, the optimal inequality is with $k + 1$, proved in a completely different way. We reproduce a proof by Oum (private communication) for the purpose of comparison with the first and for closing the topic.

We will first recall the definitions of tree-width and its related notions in Section 7.1 and recall useful lemmas for our purposes. In Section 7.2 we prove that local complementations and vertex deletions can simulate edge contractions. As an application, we prove in Section 7.3 that the rank-width of a graph is linearly bounded in term of its tree-width. In particular, we prove a lemma relating *strong tree-width* and clique-width. In Section 7.4 we will give a proof of the tighter upper bound by Oum.

7.1 Tree-Width and Related Notions

The notions of *tree-decomposition* and of *tree-width* [RS86] was introduced by Robertson and Seymour in their graph minor series. Tree-decompositions and its related notions have been studied for the last two decades. See for instance the articles and surveys by Bodlaender et al. [Bod96, BE97, Bod05, Bod06, RS86, See85]. We will recall some definitions and needed results.

Definition 7.1 ([RS86]) A *tree-decomposition* of a graph $G = \langle V_G, E_G \rangle$ is a pair (T, f) such that $T = \langle V_T, E_T \rangle$ is a tree, f is a mapping associating with every node u of T a subset $f(u)$ of V_G such that:

(T1) $\bigcup_{u \in V_T} f(u) = V_G$,

(T2) for each edge xy or arc (x, y) of G, there exists one node u in V_T with x, y in $f(u)$,

(T3) for all $u, v, w \in V_T$, if v is on the path from u to w in T then $f(u) \cap f(w) \subseteq f(v)$.

In (T2) it is convenient, for each edge xy or arc (x, y) to choose one node u such that $x, y \in f(u)$.

The *width* of a tree-decomposition (T, f) is $\max_{u \in V_T}\{|f(u)|\} - 1$. The *tree-width* of a graph G, denoted by $twd(G)$, is the minimum width over all tree-decompositions of G.

We now recall the definition of a *strong tree-decomposition* [See85].

Definition 7.2 ([See85]) A *strong tree-decomposition* of a graph $G = \langle V_G, E_G \rangle$ is a pair (T, f) as in Definition 7.1 such that:

(S1) $\{f(u) \mid u \in V_T\}$ is a partition of V_G,

(S2) for each edge xy or arc (x, y) of G:

    (S2.1) either there exists a node u in V_T with $x, y \in f(u)$,

    (S2.2) or there exists an edge uv in E_T with $x \in f(u)$ and $y \in f(v)$ or vice-versa.

The edges xy or arcs (x, y) of type (S2.2) are called the *shared edges or arcs of G*. This notion is relative to a chosen strong tree-decomposition.

The *width* of a strong tree-decomposition (T, f) is $\max_{u \in V_T} |f(u)|$. The *strong tree-width* of a graph G, denoted by $stwd(G)$, is the minimum width over all strong tree-decompositions of G.

Let (T, f) be a rooted (strong) tree-decomposition of a graph G. We say that (T, f) is *rooted* if T is. For $u \in V_T$, we call $f(u)$ the *box* of u and we denote by $G \downarrow u$ the graph $G[\bigcup_{v \in V_{T \downarrow u}} f(v)]$. In the rest of the chapter we consider rooted tree-decompositions and rooted strong tree-decompositions.

Clique-width can be considered as more powerful than tree-width. It is known that bounded tree-width implies bounded clique-width and not vice-versa (cliques have unbounded tree-width but have clique-width 2). The following theorem gives an upper-bound on the clique-width of graphs of bounded tree-width.

Theorem 7.1 ([CR05, CO00]) *Let G be a graph. Then*

$$\begin{cases} cwd(G) \le 3 \cdot 2^{twd(G)-1} & \text{if } G \text{ is undirected,} \\ cwd(G) \le 2^{twd(G)+2} - 1 & \text{if } G \text{ is directed.} \end{cases}$$

Remark 7.1 In the proofs of Theorem 7.1, the tree of an optimal tree-decomposition is made into that of a term that uses clique-width operations. This is useful for algorithmic applications.

Theorem 7.1 combined with Proposition 1.2 gives the inequality $rwd(G) \le 3 \times 2^{twd(G)-1}$. We will improve this bound and prove that rank-width is linearly bounded in term of tree-width in the following proposition.

Proposition 7.1 *For every graph G, $rwd(G) \le 4 \times twd(G) + 2$.*

Proposition 7.1 is the first result showing that the rank-width of an undirected graph is linearly bounded in term of its tree-width. The consequence is that the bound relating clique-width and rank-width is relatively optimal. In fact Corneil and Rotics [CR05] proved that for any k, there exists a graph of tree-width k with clique-width at least $2^{\lfloor \frac{k}{2} \rfloor - 1}$.

However, Oum improves the bound relating rank-width and tree-width [Oum08b]. He proves the following (which is tight).

Proposition 7.2 *([Oum08b]) For every graph G, $rwd(G) \le twd(G) + 1$.*

We will give a proof of Proposition 7.2, based on *tangles* [RS91] and branch-width, in Section 7.4. Regarding the proofs of Propositions 7.1 and 7.2 we can ask the following question.

Question 7.1 In the proofs of Propositions 7.1 and 7.2, how an optimal tree-decomposition of an undirected graph G is transformed into a layout of the function ρ_G? Precisely, can we transform an optimal tree-decomposition of G into a layout of the function ρ_G that reaches the announced bound at most?

We will discuss on this question at the end of Section 7.4. If G is an undirected graph and $e = xy$ an edge of G, the contraction of e consists in deleting the vertex x and adding the edges $\{yz \mid z \in N_G(x) - N_G(y), \ z \ne x, y\}$. For $F \subseteq E_G$, we denote by G/F the simple graph obtained from G by contracting the edges of F. An undirected graph H is a *minor* of an undirected graph G if H can be obtained from G by applying a sequence of edge contractions, vertex deletions and edge deletions. Robertson and Seymour proved that for every k there exists a finite list $\mathcal{F}_k$ of undirected graphs such that an undirected graph G has tree-width at most k if and only if it has no minor isomorphic to an undirected graph in $\mathcal{F}_k$ [RS86]. This result is an analogous to Theorem 1.1 that deals with rank-width of undirected graphs. We will show in the next section how to simulate edge contractions by vertex-minor operations. Let us now recall and prove some lemmas.

Lemma 7.1 ([Bod98]) *Let G be a graph of tree-width k. Then G has a tree-decomposition (T, f) of width k such that:*

1. *For each $u \in V_T$, we have $|f(u)| = k + 1$.*

  2. *For each $(u, v) \in E_T$, we have $|f(u) \cap f(v)| = k$.*

Lemma 7.2 *Let (T, f) be a tree-decomposition of width k of a graph G. There exist a graph H and a strong tree-decomposition (T, g) of H of width $k + 1$ such that $G = H/F$ where F is the set of shared edges. The graph $H[F]$ is a forest.*

Proof. We let $H = (V_H, E_H)$ where:

$$\begin{aligned} V_H &= \{x_u \mid u \in V_T \text{ and } x \in f(u)\}, \\ E_H &= \{x_u x_v \mid (u, v) \in E_T \text{ and } x \in f(u) \cap f(v)\} \cup \\ &\quad \{x_u y_u \mid u \in V_T \text{ and } x, y \in f(u) \text{ and the edge } xy \text{ or the arc } (x, y) \text{ is in } f(u)\}. \end{aligned}$$

For each $u \in V_T$, we let $g(u) = \{x_u \mid x \in f(u)\}$. It is easy to verify that (T, g) is a strong tree-decomposition of H, the shared edges are the edges $\{x_u x_v \mid (u, v) \in E_T \text{ and } x \in f(u) \cap f(v)\}$. They form a set F that spans a forest and $G = H/F$ (by the definition of tree-decomposition). See Figure 17 for an example. The shared edges are dotted and marked by ε (because they are *in fine* contracted). ∎
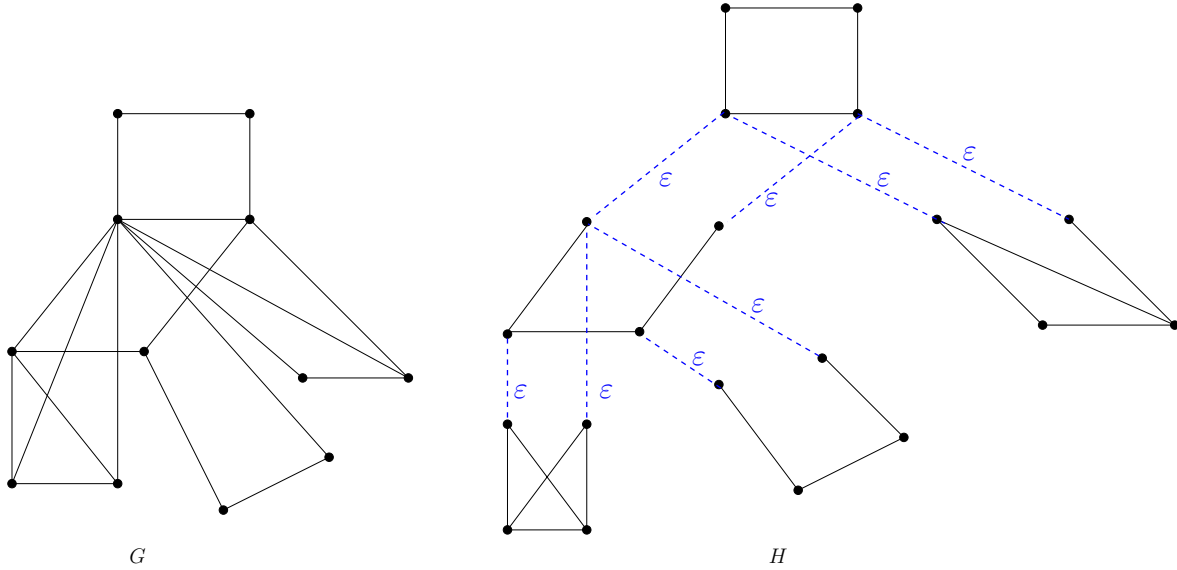
Figure 17: A graph G and the corresponding graph H. F is the set of edges labeled by ε.

7.2 Vertex-Minor Reductions and Edge Contractions

Let G be a graph. We say that $J \subseteq E_G$ is *good in G* if $G[J]$ is a forest and G/J has no loop nor multiple edges. This is equivalent to saying that in G every cycle contains at least 3 edges not in J. If a rooted forest is reduced to one arc f, we will denote it by $\{f\}$. If G is a directed graph, we denote by $und(G)$ the simple undirected graph obtained from G by omitting the direction of the arcs.

Let F be a rooted forest. We denote by V_F^{root} the set $\{x \in V_F \mid x \text{ is a root}\}$ and by V_F^{nroot} the set $V_F - V_F^{root}$, i.e, the vertices that are the targets of some arcs in F. We say that F is a *rooted forest in G* if $E_{und(F)} \subseteq E_G$, i.e., $G[E_{und(F)}]$ is a subgraph of G. We say that F is a *good rooted forest in G* if F is a rooted forest in G and $E_{und(F)}$ is good in G. Let us define two operations.

Definition 7.3 (*Lc-deletion*) Let G be a graph and $x \in V_G$. The graph obtained by applying an *lc-deletion at x to G* is $G \diamond x = (G * x) \backslash x$.

It is clear that $G \diamond x$ is a vertex-minor of G. We note that $G \diamond x \diamond y$ is not necessarily equal to $G \diamond y \diamond x$. Figure 18 gives an illustration of Definition 7.3.
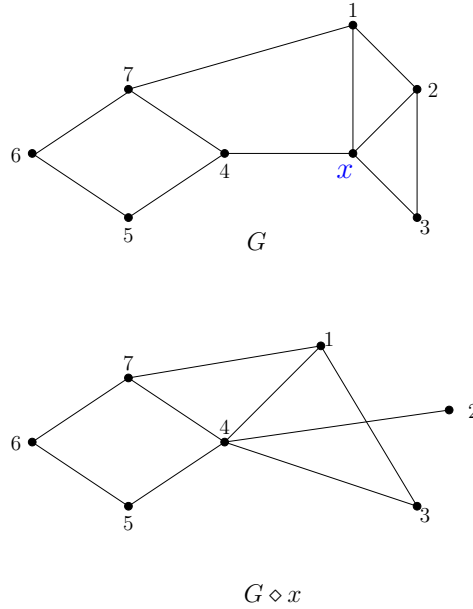
Figure 18: A graph G and the graph $G \diamond x$.

Definition 7.4 (*Local augmentation*) Let G be a graph and F be a rooted forest in G. The graph obtained by applying a *local augmentation at F to G* is $G \boxtimes F = \langle V_{G \boxtimes F}, E_{G \boxtimes F} \rangle$ where:

$$V_{G \boxtimes F} = V_G \cup \{x^t \mid x \in V_F^{nroot}\},$$
$$E_{G \boxtimes F} = E_G \cup \{x^t y \mid x \in V_F^{nroot} \text{ and } xy \in E_G \text{ and } (x,y) \notin F \text{ and } (y,x) \notin F\}$$
$$\cup \{x^t y^t \mid x, y \in V_F^{nroot} \text{ and } xy \in E_G\}.$$

x^t is a new vertex.

We illustrate the construction of Definition 7.4 with an example. Figure 19 shows a graph G, a rooted forest F in G and the graph $G \boxtimes F$. The connected components of F are T_1 induced by $\{a, b, c, d\}$ with root 1 and T_2 induced by $\{e\}$ with root 6. One can verify that we

have $V_F^{root} = \{1, 6\}$ and $V_F^{nroot} = \{2, 3, 4, 5, 7\}$. Then

$$V_{G \boxtimes F} = V_G \cup \{2^t, 3^t, 4^t, 5^t, 7^t\},$$
$$E_{G \boxtimes F} = E_G \cup \{\{2^t, 8\}, \{2^t, 9\}, \{4^t, 6\}, \{4^t, 10\}, \{4^t, 11\}, \{5^t, 12\}, \{5^t, 13\}, \{7^t, 12\}, \{7^t, 14\}\}$$
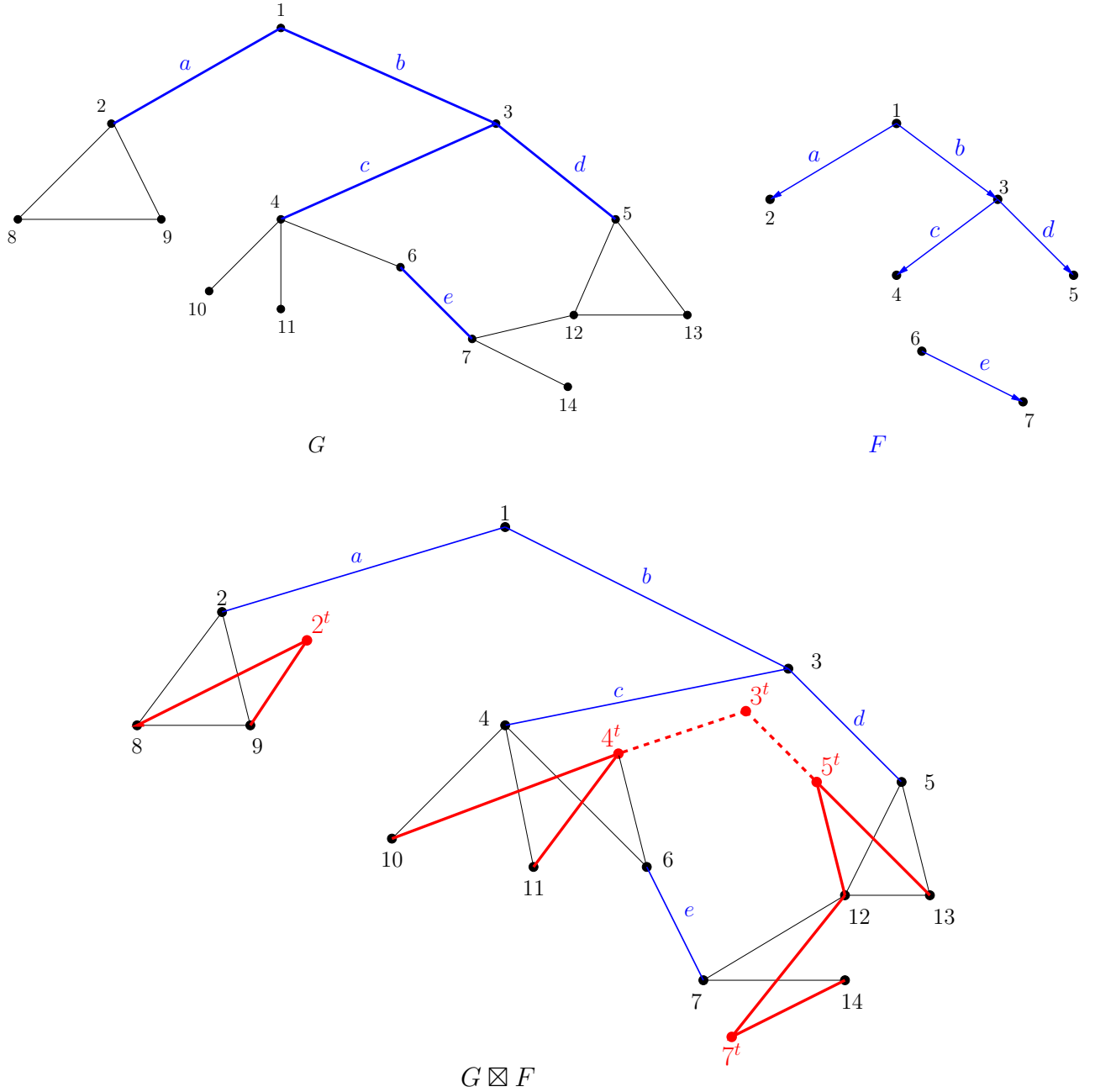$$\cup \{\{3^t, 4^t\}, \{3^t, 5^t\}\}.$$

Figure 19: A graph G, a rooted forest F in G and the graph $G \boxtimes F$.

The main result of this section is the following.

Theorem 7.2 (Vertex-Minor and Minor Relations) *Let G be a graph and F a good rooted forest in G. Then $G/E_{und(F)}$ is a vertex-minor of $G \boxtimes F$.*

In order to prove Theorem 7.2, we prove how to simulate edge contractions by vertex-minor operations. For that purpose we use the operations $\diamond$ and $\boxtimes$ defined above in this section. We begin by proving some technical lemmas.

Fact 7.1 *Let G be a graph and let $f = (y, x)$ be such that $\{f\}$ is a good rooted forest in G. Then $(G \boxtimes \{f\}) \diamond x \diamond x^t = G/e$ where $e = yx$.*

Proof. We let $N_G(x) = \{y, z_1, \ldots, z_m\}$. The effect of contracting e can be described as follows:

  (a) deletion of x and the edges incident to x,

  (b) creation of edges between y and z_i for each $i \in [m]$.

Since $\{f\}$ is a good rooted forest in G, there is no edge in G between y and any z_i for any $i \in [m]$. The effect of applying lc-deletion at x to $G \boxtimes \{f\}$ is thus:

  (1) creation of edges between y and z_i for each $i \in [m]$ (that is (b)),

  (2) creation of edges $z_i z_j$ where $z_i z_j \notin E_G$, $i \neq j$,

  (3) deletion of edges $z_i z_j \in E_G$, $i \neq j$,

  (4) deletion of x and the incident edges to x (that is (a)).

The lc-deletion applied at x links y to z_i, deletes x, but also deletes existing edges between the neighbors z_i of x (that is (3)) and creates edges in place of non-existing ones (that is (2)). Since $\{f\}$ is good we have $N_{G \boxtimes \{f\}}(x^t) = \{z_1, \ldots, z_m\}$. Therefore, an lc-deletion at x^t undoes (2) and (3) and deletes x^t and its incident edges. Then $(G \boxtimes \{f\}) \diamond x \diamond x^t = G/e$. ∎

We now generalize Fact 7.1.

Lemma 7.3 *Let G be a graph. Let F be a good rooted forest in G and let $f = (y, x)$ be an arc in F where x is a leaf. Then $(G \boxtimes F) \diamond x \diamond x^t = (G/e) \boxtimes (F - \{f\})$ where $e = yx$.*

We distinguish two cases: either y is a root or not (see Figures 20 and 21 for illustrations). For more readability we prove the two cases in two different claims.

Claim 7.1 *Let F be a good rooted forest in G and let $f = (y, x)$ be an arc in F where y is a root and x is a leaf. Then $(G \boxtimes F) \diamond x \diamond x^t = (G/e) \boxtimes (F - \{f\})$ where $e = yx$.*

Proof. Let $N_G(x) = \{y, z_1, \ldots, z_m\}$. The effect of contracting the edge yx in G can be described as follows:

(a) deletion of x and its incident edges,

(b) creation of edges between y and z_i for each $i \in [m]$.

Since F is a good rooted forest in G, y is not adjacent to any z_i in G. But in G/e, y is adjacent to all z_i. We get:

$$V_{(G/e) \boxtimes (F-\{f\})} = (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot} \text{ and } z \neq x\},$$
$$E_{(G/e) \boxtimes (F-\{f\})} = (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\}$$
$$\cup \{yz_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\}$$
$$\cup \{u^t z \mid u \in V_F^{nroot} \text{ and } u \neq x \text{ and } uz \in E_G \text{ and } (u,z), \ (z,u) \notin F\}$$
$$\cup \{u^t z^t \mid u, z \in V_F^{nroot} \text{ and } u, z \neq x \text{ and } uz \in E_G\}.$$

We have $N_{G \boxtimes F}(x) = \{z_1, \ldots, z_m\} \cup \{z_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\} \cup \{y\}$. Therefore, the effect of applying an lc-deletion at x to $G \boxtimes F$ can be described as follows:

(1) creation of edges yz_i for each $i \in [m]$ (that is (b)),

(2) creation of edges yz_i^t for each $z_i \in V_F^{nroot}$ (edges created in $(G/e) \boxtimes (F - \{f\})$),

(3) creation of edges $z_i z_j$, $z_i z_j^t$, $z_i^t z_j^t$ where $z_i z_j \notin E_G$ and $i \neq j$ and of edges $z_i z_i^t$ for each $z_i \in V_F^{nroot}$,

(4) deletion of edges $z_i z_j$, $z_i z_j^t$ and $z_i^t z_j^t$ where $z_i z_j \in E_G$ and $i \neq j$,

(5) deletion of x and its incident edges (that is (a)).

By definition, $N_{G \boxtimes F}(x^t) = \{z_1, \ldots, z_m\} \cup \{z_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\}$. Then the effect of applying an lc-deletion at x^t to $(G \boxtimes F) \diamond x$ can be described as follows:

(3') deletion of edges $z_i z_j$, $z_i z_j^t$, $z_i^t z_j^t$ where $z_i z_j \notin E_G$ and $i \neq j$, and of edges $z_i z_i^t$ for each $z_i \in V_F^{nroot}$ (in order to undo (3)),

(4') creation of edges $z_i z_j$, $z_i z_j^t$ and $z_i^t z_j^t$ where $z_i z_j \in E_G$ and $i \neq j$ (in order to undo (4)),

(5') deletion of x^t and its incident edges.

Then we have:

$$V_{(G \boxtimes F) \diamond x \diamond x^t} = (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot}\} \setminus \{x^t\},$$
$$E_{(G \boxtimes F) \diamond x \diamond x^t} = (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\}$$
$$\cup \{yz_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\}$$
$$\cup \{u^t z \mid u \in V_F^{nroot} \text{ and } u \neq x \text{ and } uz \in E_G \text{ and } (u,z), \ (z,u) \notin F\}$$
$$\cup \{u^t z^t \mid u, z \neq x \text{ and } u, z \in V_F^{nroot} \text{ and } uz \in E_G\}.$$

We thus deduce that $(G \boxtimes F) \diamond x \diamond x^t = (G/e) \boxtimes (F - \{f\})$. ∎

Figure 20: F and $\{f\}$ with y a root.

Claim 7.2 *Let F be a good rooted forest in G and let $f = (y, x)$ be an arc in F where y is a non-root and x is a leaf. Then $(G \boxtimes F) \diamond x \diamond x^t = (G/e) \boxtimes (F - \{f\})$ where $e = yx$.*

Proof. Let $N_G(x) = \{y, z_1, \ldots, z_m\}$. As in Claim 7.2 the effect of contracting the edge yx in G can be described as follows:

(a) deletion of x and its incident edges,

(b) creation of edges between y and z_i for each $i \in [m]$.

Since F is a good rooted forest in G, y is not adjacent to any z_i in G, but it is in G/e. We get:

$$
\begin{aligned}
V_{(G/e)\boxtimes(F-\{f\})} &= (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot} \text{ and } z \neq x\}, \\
E_{(G/e)\boxtimes(F-\{f\})} &= (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\} \\
&\quad \cup \{yz_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\} \\
&\quad \cup \{u^t z \mid u \in V_F^{nroot} \text{ and } u \neq x \text{ and } uz \in E_G \text{ and } (u, z), (z, u) \notin F\} \\
&\quad \cup \{u^t z^t \mid u, z \in V_F^{nroot} \text{ and } u, z \neq x \text{ and } uz \in E_G\} \\
&\quad \cup \{y^t z_i \mid i \in [m]\} \cup \{y^t z_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\}.
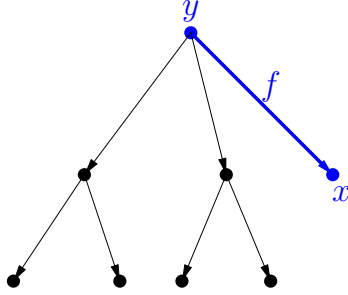\end{aligned}
$$

We have $N_{G\boxtimes F}(x) = \{z_1, \ldots, z_m\} \cup \{z_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\} \cup \{y\}$. Then the effect of applying an lc-deletion at x to $G \boxtimes F$ is the same as in Claim 7.1.

By definition, $N_{G\boxtimes F}(x^t) = \{z_1, \ldots, z_m\} \cup \{z_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\} \cup \{y^t\}$. So an lc-deletion at x^t in $(G \boxtimes F) \diamond x$ has the same effect as in Claim 7.1 with two additional steps which create two types of edges:

(1') creation of edges $y^t z_i$ for each $i \in [m]$ (edges created in $(G/e) \boxtimes (F - \{f\})$),

(2') creation of edges $y^t z_i^t$ for each $z_i \in V_F^{nroot}$ (edges created in $(G/e) \boxtimes (F - \{f\})$).

Then we have:

$$V_{(G\boxtimes F)\diamond x\diamond x^t} = (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot}\}\backslash\{x^t\},$$

$$E_{(G\boxtimes F)\diamond x\diamond x^t} = (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\}$$
$$\cup \{yz_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\}$$
$$\cup \{u^t z \mid u \in V_F^{nroot} \text{ and } u \neq x \text{ and } uz \in E_G \text{ and } (u,z),\ (z,u) \notin F\}$$
$$\cup \{u^t z^t \mid u, z \neq x \text{ and } u, z \in V_F^{nroot} \text{ and } uz \in E_G\}$$
$$\cup \{y^t z_i \mid i \in [m]\} \cup \{y^t z_i^t \mid i \in [m] \text{ and } z_i \in V_F^{nroot}\}.$$

We thus deduce that $(G \boxtimes F) \diamond x \diamond x^t = (G/e) \boxtimes (F - \{f\})$. ∎

Figure 21: F and $\{f\}$ with y not a root.

Proof of Lemma 7.3. We considered the two cases (y root or not) in Claim 7.1 and in Claim 7.2. In both cases, we have $(G \boxtimes F) \diamond x \diamond x^t = (G/e) \boxtimes (F - \{f\})$. ∎

We can now prove Theorem 7.2 by using Fact 7.1 and Lemma 7.3.

Proof of Theorem 7.2. We prove it by induction on the size of F. Let $V_F^{nroot} = \{x_1, \ldots, x_k\}$. Its elements are numbered from leaves to internal nodes in inverse topological order. We claim that $G/E_{und(F)} = (\cdots((G \boxtimes F) \diamond x_1 \diamond x_1^t) \diamond \cdots) \diamond x_k \diamond x_k^t$.

If $F = \{f\}$, let $f = (y, x_1)$. From Fact 7.1, we have $G/e = (G \boxtimes \{f\}) \diamond x_1 \diamond x_1^t$ where $e = yx_1$.

We now assume that $|F| \geq 2$ and let $F = F_1 \cup \{f\}$ where $f = (y, x_1)$ and x_1, the target of f, is a leaf. We let $e = yx_1$. By definition, we have $G/E_{und(F)} = (G/e)/E_{und(F_1)}$.

We observe that the edges incident to the vertices $x_1^t, \ldots, x_k^t$ in $G \boxtimes F$ are defined relatively to the pair (G, F) according to the definition of the operation $\boxtimes$. We also observe that F_1 is a good rooted forest in G/e and then the non-root vertices of F_1 are $x_2, \ldots, x_k$.

By Lemma 7.3, $(G \boxtimes F) \diamond x_1 \diamond x_1^t = (G/e) \boxtimes F_1$. Then the edges incident to $x_2, \ldots, x_k$ and $x_2^t, \ldots, x_k^t$ are the same in $(G \boxtimes F) \diamond x_1 \diamond x_1^t$ and in $(G/e) \boxtimes F_1$. Therefore, we get

$$(\cdots((G \boxtimes F) \diamond x_1 \diamond x_1^t) \diamond \cdots) \diamond x_k \diamond x_k^t = (\cdots(((G/e) \boxtimes F_1) \diamond x_2 \diamond x_2^t) \diamond \cdots) \diamond x_k \diamond x_k^t.$$

By the inductive hypothesis, we have

$$((G/e) \boxtimes F_1) \diamond x_2 \diamond x_2^t) \diamond \cdots) \diamond x_k \diamond x_k^t = (G/e)/E_{und(F_1)} = G/E_{und(F)}.$$

Then $G/E_{und(F)} = (\cdots ((G \boxtimes F) \diamond x_1 \diamond x_1^t) \diamond \cdots) \diamond x_k \diamond x_k^t.$ ∎

7.3 Application to Rank-Width

In this section we prove Proposition 7.1. We first prove that clique-width is linearly bounded in term of strong tree-width.

Lemma 7.4 *Let G be a graph, then $cwd(G) \leq 2 \times stwd(G) + 1$.*

Proof. Let (T, f) be a rooted strong tree-decomposition of width k of G. To prove the lemma, we introduce a binary operation. We first consider the particular case of the trees.

Let K and H be trees with one distinguished node labeled by 1 and all other nodes labeled by 0. We let $K \odot H$ be obtained from $K \oplus H$, where K and H are disjoint, by a new edge from the distinguished node of K to the one of H, and the distinguished node of K is made the distinguished one of the resulting tree. Clearly,

$$K \odot H = \rho_{2 \to 0}(\eta_{1,2}(K \oplus \rho_{1 \to 2}(H))) \tag{2}$$

All trees can be generated from the operation $\odot$ and the constant $\mathbf{1}$.

Let $n, m \leq k$. Assume now that K is a graph with distinguished vertices labeled from 1 to n, each label for one vertex. All other vertices are labeled by 0. Let H be similar with distinguished vertices labeled from 1 to m. Let t_K and t_H be terms that define respectively K and H as explained above. For $R \subseteq [n] \times [m]$, we define

$$K \odot_R H = \left(\circ_{i \in [m]} \ \rho_{i' \to 0} \right) \left(\circ_{(i,j) \in R} \ \eta_{i,j'} \right) \left(t_K \oplus \left(\circ_{i \in [m]} \ \rho_{i \to i'} \right) (t_H) \right) \tag{3}$$

Claim 7.3 *The simple loop-free undirected graphs of strong tree-width $\leq k$ are generated by the operations*

1. *$\odot_R$ for $R \subseteq [k] \times [k]$,*

2. *$\eta_{i,j}$ for $i, j \in [k]$, $i \neq j$,*

3. *and the basic graphs $\mathbf{1} \oplus \mathbf{2} \oplus \cdots \oplus \mathbf{n}$ for $1 \leq n \leq k$.*

It is clear from Claim 7.3 that $cwd(G) \leq 2k + 1$ if $stwd(G) \leq k$. ∎

We can now prove Claim 7.3.

Proof of Claim 7.3. We first color each box $f(u)$ with colors from 1 to $|f(u)|$ using a mapping γ_u, each label for one vertex (see Figure 22 for an example). We prove by induction on the number of nodes of T that for each $u \in V_T$, the graph $G \downarrow u$ labeled so that the vertices in $f(u)$ are labeled from 1 to $|f(u)|$ and all others are labeled by 0, is generated by the above operations.

Let $R_u = \{(\gamma_u(x), \gamma_u(y)) \mid x, y \in f(u) \text{ and } xy \in E_G\}$ and assume that $|f(u)| = n$. Let

$$t_u = (\circ_{(i,j) \in R_u} \eta_{i,j})(\mathbf{1} \oplus \mathbf{2} \oplus \cdots \oplus \mathbf{n}).$$

It is clear from the definition of R_u, that $val(t_u) = G[f(u)]$. If $V_T = \{u\}$, we have $G = G[f(u)]$, then the claim is verified. Now assume that $v_1, \ldots, v_p$ are the children of u ($p = 2$ in Figure 22). By the inductive hypothesis, for each child v_i of u, $G \downarrow v_i$, labeled as explained above, is generated by the above operations.

Let $R_i = \{(\gamma_u(x), \gamma_{v_i}(y)) \mid x \in f(u), \ y \in f(v_i) \text{ and } xy \in E_G\}$ for $i = 1, \ldots, p$. It is clear that $R_i \subseteq [k] \times [k]$ for $i = 1, \ldots, p$. By the definition of strong tree-decompositions and the inductive hypothesis, it only remains to add the shared edges between vertices of $f(u)$ and vertices of $f(v_i)$ for $i = 1, \ldots, p$. From the definition of R_i, if $x \in f(u)$, $y \in f(v_i)$ and $xy \in E_G$, then $(\gamma_u(x), \gamma_{v_i}(y)) \in R_i$. We let

$$t = (((val(t_u) \odot_{R_1} G \downarrow v_1 \odot_{R_2} G \downarrow v_2) \ldots) \odot_{R_p} G \downarrow v_p.$$

It is easy to verify that the above expression defines $G \downarrow u$ as wanted (see Figure 22 for an example). If u is the root of T, we have $G = G \downarrow u$. Then the claim is proved. ∎

We illustrate the proof of Lemma 7.4 with an example, taking $p = 2$. Figure 22 shows a part of a strong tree-decomposition of a graph G (the sub-tree of the strong tree-decomposition rooted at u). The node u has two children v_1 and v_2. One can verify we have:

$$R_1 = \{(1,1), (1,2), (2,2)\},$$
$$R_2 = \{(1,1), (3,1), (4,3)\},$$
$$G \downarrow u = (G[f(u)] \odot_{R_1} G \downarrow v_1) \odot_{R_2} G \downarrow v_2.$$

One can easily verify the following from the construction of Claim 7.3.

Remark 7.2 If for each $(u, v) \in E_T$ the shared edges between $f(u)$ and $f(v)$ are incident to at most $k - i$ vertices in $f(v)$, then $cwd(G) \leq 2k - i + 1$. ∎
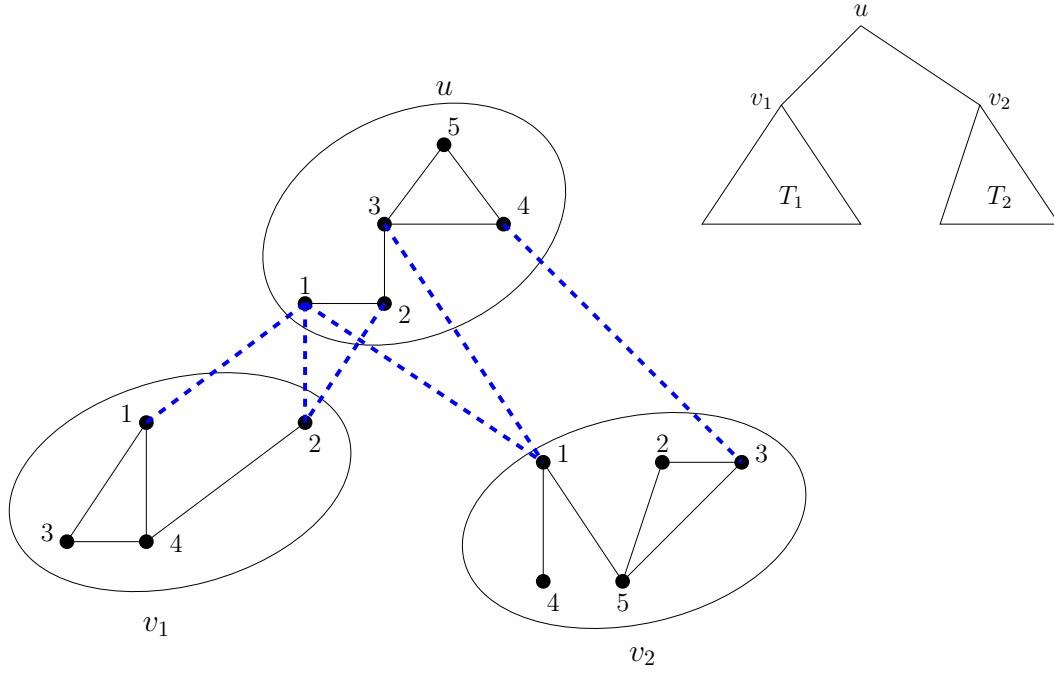
Figure 22: Illustrating the proof of Claim 7.3.

We can now prove Proposition 7.1.

Proof of Proposition 7.1. Let (T, f) be a rooted tree-decomposition of width k of G satisfying the condition of Lemma 7.1. By Lemma 7.2, we can build a graph H, a forest F and a strong tree-decomposition (T, g) of H with $G = H/E_F$. The notation g is as in Lemma 7.2. Let $\vec{F} = (V_F, E_{\vec{F}})$ where

$$E_{\vec{F}} = \{(x_u, x_v) \mid x_u x_v \in E_F \text{ and } (u, v) \in E_T\}.$$

It is clear that $und(\vec{F}) = F$. By the definition of F, H/E_F has no loops nor multiple edges, so E_F is good in H. Then $\vec{F}$ is a good rooted forest in H.

By Lemma 1.1 and Theorem 7.2, we have $rwd(G) \leq rwd(H \boxtimes \vec{F})$. We now prove that $rwd(H \boxtimes \vec{F}) \leq 4k + 2$.

Let $h(u) = g(u) \cup \{x_u^t \mid x_u \in V_{\vec{F}}^{nroot}\}$. It is easy to prove that (T, h) is a strong tree-decomposition of $H \boxtimes \vec{F}$ of width at most $2k + 1$. We have $2k + 1$ instead of $2(k + 1)$ because, for each $u \in V_T$, the size of the set $\{x_u \mid x_u \in V_{\vec{F}}^{nroot}\}$ is at most k and then the size of the set $\{x_u^t \mid x_u \in V_{\vec{F}}^{nroot}\}$ is at most k. It is easy to verify (from the definition of (T, f)) that, for each $(u, v) \in E_T$ in (T, h), the shared edges between the vertices of $h(u)$ and the vertices of $h(v)$ are incident to at most $2k$ vertices in $h(v)$. Then by Lemma 7.4 and Remark 7.2, $cwd(H \boxtimes \vec{F}) \leq 4k + 2$.

By Proposition 1.2, we have $rwd(H \boxtimes \vec{F}) \leq 4k + 2$. Then $rwd(G) \leq 4k + 2$. ∎

7.4   A Proof of Proposition 7.2

This proof is due to Oum (private communication). The notion of branch-width of an undirected graph is defined in Definition 1.8, however we recall the following notations. If G is an undirected graph, for every subset X of E_G we let T_X be the set of vertices incident to at least one edge in X and we let $\eta_G(X) = \left| \{T_X \cap T_{E_G - X}\} \right|$ where $\eta_G : 2^{E_G} \to \mathbb{N}$. We now recall the notion of *tangles* [RS91].

Let G be a graph. It is common, given a bipartition (A, B) of E_G such that $\eta_G(A) = \eta_G(B)$ is a small integer, to view one of A or B as the "small part" of the bipartition. In [RS91] the following example is given: if H is a minor of G, isomorphic to a large complete graph, then for every bipartition (A, B) such that $\eta_G(A)$ is small, exactly one of A or B has a sub-graph contracted to a vertex of H; in this case we consider the other as the "small part". Informally, the notion of *tangle* [RS91] can be seen as an axiomatization of a family of "small parts".

Definition 7.5 (Tangles [RS91]) Let V be a set and let $f : 2^V \to \mathbb{N}$ be a symmetric and sub-modular function with $f(\emptyset) = 0$. Let k be a positive integer. A collection $\mathcal{T}$ of subsets of V is called an *f-tangle of order $k + 1$* if it satisfies the following conditions:

(TA1) For all $A \subseteq V$, if $f(A) \leq k$, then either $A \in \mathcal{T}$ or $V - A \in \mathcal{T}$.

(TA2) If $A, B, C \in \mathcal{T}$, then $A \cup B \cup C \neq V$.

(TA3) For all $v \in V$, we have $V - \{v\} \notin \mathcal{T}$.

The notion of tangle can be also seen as a kind of duality to the notion of branch-width of symmetric functions, as showed by Robertson and Seymour [RS91].

Theorem 7.3 ([RS91]) *Let $f : 2^V \to \mathbb{N}$ be a symmetric and sub-modular function with $f(\emptyset) = 0$. There is no f-tangle of order $k + 1$ if and only if the branch-width of the function f is at most k.*

We can now prove the following and then Proposition 7.2.

Proposition 7.3 *Let $k \geq 2$. If an undirected graph G has rank-width at least $k + 1$, then G has branch-width at least $k + 1$.*

Proof. We assume G connected without loss of generality. Since the rank-width of G is larger than k, there exists a $\rho_G^{GF(2)}$-tangle $\mathcal{T}$ of order $k + 1$. Let $\mathcal{U} = \{X \subseteq E_G \mid \eta_G(X) \leq k \text{ and } T_X \in \mathcal{T}\}$. We claim that $\mathcal{U}$ is an η_G-tangle of order $k + 1$.

1. Suppose that $\eta_G(X) \leq k$ for a set X of edges. We need to show that either $X \in \mathcal{U}$ or $E_G - X \in \mathcal{U}$. Suppose that $X \notin \mathcal{U}$ and $E_G - X \notin \mathcal{U}$. Then $T_X \notin \mathcal{T}$. Since $\rho_G^{GF(2)}(T_X) \leq k$ and $\mathcal{T}$ is a $\rho_G^{GF(2)}$-tangle, we know that $(V_G - T_X) \in \mathcal{T}$. Similarly, we deduce that $(V_G - T_{E_G - X}) \in \mathcal{T}$. Moreover, since $\eta_G(X) \leq k$, $T_X \cap T_{E_G - X} \in \mathcal{T}$ (any set of at most k vertices belongs to a $\rho_G^{GF(2)}$-tangle of order $k + 1$). This leads a contradiction because $(V_G - T_X) \cup (T_X \cap T_{E_G - X}) \cup (V_G - T_{E_G - X}) = V_G$ which violates Condition (TA2) in Definition 7.5.

2. Suppose that $X \cup Y \cup Z = E_G$ for three sets $X, Y, Z \in \mathcal{U}$. If $v \notin T_X \cup T_Y \cup T_Z$, then v is an isolated vertex. Since G is connected, there is no such v. Thus $T_X \cup T_Y \cup T_Z = V_G$ and $T_X, T_Y, T_Z \in \mathcal{T}$. A contradiction (violation of Condition (TA2) in Definition 7.5).

3. For each edge $e \in E_G$, $\eta_G(\{e\}) \leq 2$ and therefore if $k \geq 2$, then $T_{\{e\}} \in \mathcal{T}$. So, $\{e\} \in \mathcal{U}$.

We checked all conditions of Definition 7.5 by (1)-(3). This completes the proof.   ∎

Corollary 7.1 *For every graph G, $rwd(G) \leq \max\{bwd(G), 1\}$.*

Proof. By Proposition 7.3, if the branch-width of G is larger than 1, then the rank-width of G is at most the branch-width of G. If the branch-width of G is 1, then G is a forest and therefore the rank-width of G is 1. If the branch-width of G is 0, then G is a matching and therefore the rank-width of G is 1.   ∎

Proof of Proposition 7.2. Robertson and Seymour [RS91] proved that $bwd(G) \leq twd(G) + 1$. And by Corollary 7.1, $rwd(G) \leq bwd(G)$.   ∎

We now try to give an answer to Question 7.1. One can verify that the proof of Proposition 7.1 is constructive. Given a graph G and an optimal tree-decomposition (T, f) of G, we construct a graph, denoted by $H \boxtimes F$, from (T, f) and construct a strong tree-decomposition (T, g) of $H \boxtimes F$. We then construct from (T, g) a clique-width expression t, that uses $4k + 2$ colors, of $H \boxtimes F$. Finally, by the proof of Proposition 1.2 in [OS06], t can be transformed into a layout of the function $\rho_G^{GF(2)}$ of branch-width at most $4k + 2$.

Regarding the proof of Proposition 7.2, it is not clear how to get a layout of G. If from a tree-decomposition of G of width k, we can get a layout of the function of η_G of branch-width $k + 1$ [RS91], the proof of Proposition 7.2 does not inform on how to transform a layout of the function η_G of branch-width k into a layout of the function $\rho_G^{GF(2)}$ of branch-width k. In fact, in a layout $(T, \mathcal{L})$ of the function η_G the leaves of T are in bijection with the edges of G, whereas in a layout $(T, \mathcal{L})$ of the function $\rho_G^{GF(2)}$ the leaves of T are in bijection with the vertices of G. Oum gives another proof of Proposition 7.2 in [Oum08b] that is also a non-constructive proof. Therefore, if with our proof technique we get an upper bound that is not tight, it has the advantage of being constructive, which is not the case for the known proofs of Proposition 7.2.

7.5 Conclusion

We have shown how to simulate edge contractions by duplicating certain vertices and by using vertex-minor operations. As an application of these techniques we have shown that rank-width is linearly bounded in term of tree-width.

We can also simulate edge deletions. Let G be a simple undirected graph and let $e = xy$ be an edge linking x and y in G. If we introduce a new vertex x' adjacent to x and y, we

obtain a graph G'; by applying a local complementation at x', we delete the edge e; deleting the vertex x', we get the graph G-e which is G without the edge e, and G-e is a vertex-minor of G'. We have then shown how to simulate deletion of edges by vertex-minor operations.

Oum conjectured that graphs are well-quasi-ordered by the vertex-minor relation [Oum05b, Oum05c] and proved this fact for graph classes of bounded rank-width [Oum08a]. In this chapter we prove that minor-operations can be simulated by vertex-minor operations. The techniques shown in this chapter can perhaps help to tackle the conjecture.

We now conclude this part, concerning graph classes of bounded rank-width, by two tables that summarize the known results. Because of space constraints, we use the following abbreviations.

Abbreviation	**Full Expression**
A. C	Algebraic Characterization
R. A.	Recognition Algorithm
F. C.	Forbidden Configutations
W. P.	Width Parameter
C. T.	Cubic-Time
Approx	Approximation

W. P.	**A. C.**	**R. A.**	**F. C.**
Clique-Width	YES (Definition)	Approx C. T. [OS06, HO07]	NO
Rank-Width	YES (Theorem 4.3)	Exact C. T. [HO07]	YES [Oum05b]

Table 2: Summary of the results concerning undirected graphs.

W. P.	**A. C.**	**R. A.**	**F. C.**
Clique-Width	YES (Definition)	Approx C. T. (Corollary 5.2)	NO
$GF(4)$-Rank-Width	Approx (Theorem 4.4)	Exact C. T. (Theorem 5.3)	YES (Theorem 3.1)
Bi-Rank-Width	Approx (Propositions 4.8, 4.9)	Exact C. T. (Theorem 5.1)	NO

Table 3: Summary of the results concerning directed graphs.

Part II

Labeling Schemes for Graph Classes of Unbounded Rank-Width

Labeling Schemes for Graph Classes of Unbounded Rank-Width

Decompositions and Algorithmic Results

There are many types of graph decompositions, which are useful for investigating graph structures and for algorithmic purposes. We have in particular.

(T1) Canonical graph decompositions: decomposition of connected graphs into 2-connected graphs, decomposition of 2-connected graphs into 3-connected components [Tut66], modular decomposition [Gal67] or split-decomposition [Cun82].

(T2) Graph decompositions associated with graph parameters and their corresponding or equivalent definitions by algebraic terms written with graph operations: tree-decomposition [RS86] with its associated graph parameter tree-width or rank-decomposition [Oum05b] and its associated graph parameter rank-width.

(T3) Graph decompositions that give some structural informations on certain classes of graphs: decompositions of graph classes that exclude a fixed graph as a minor [RS03, Gro07, KM07, DGK07], decompositions of perfect graphs in [CRST03] or of claw-free graphs [CS08a, CS08b].

(T4) Other decompositions driven by the search of some structural informations: the sparse decomposition of graph classes that exclude a fixed graph as a minor in [AGMW07].

The decompositions in (T1)-(T3) are mostly based on a tree describing the graphs and this has many algorithmic applications:

(T1) Decompositions in (T1) can be used as a pre-processing step in some algorithms. In [Cre07] the modular decomposition is used in the recognition of dynamic cographs, interval graphs or permutation graphs and split decomposition is used in [GP07] for the recognition of dynamic distance hereditary graphs. They can also be used for solving NP-complete problems as done in [Rao08]. We will use in Chapter 9 the decomposition of connected graphs into 2-connected graphs and the one of 2-connected graphs into 3-connected graphs as tools.

(T2) Decompositions in (T2) can yield meta-theorems for constructing polynomial-time algorithms for MS-definable properties. For instance, it is known that every MS-definable property can be checked in linear-time on graph classes of bounded tree-width

125

[Cou90, ALS91, Bod96] and can be checked in cubic-time on graph classes of bounded rank-width [CMR00, HO07].

(T3) The decomposition of graph classes that exclude a fixed graph as a minor in [RS03] can be used in order to give meta-theorems for FO-definable properties. Grohe and Flum [FG01a] proved that every FO-definable property can be checked in polynomial-time on graph classes that exclude a minor.

However, graphs arising from concrete and real problems are not usually decomposable in the above ways. Moreover, Johansson [Joh98] proved that random graphs do not have bounded clique-width (the same statement is probably true for other cases). Several methods exist in order to try to overcome this difficulty.

- One can define local versions of graph parameters or graph properties, e.g., *local tree-width* [FG01b, FG01a] or *locally excluding a minor* [DGK07].

- One can define other notions of graph decompositions a la (strong) tree-decomposition that are not based on trees, but on other graph classes as done in [Sch97, WT07].

In the second part of this thesis we will propose decompositions that are based on locally boundedness of clique-width and decompositions that are roughly *planar gluings* of graphs of small clique-width with limited overlaps. These decompositions will be used for constructing labeling schemes. Let us first review the existing algorithmic meta-theorems that concern graph classes of unbounded clique-width.

One of the first algorithmic meta-theorems concerning graph classes of unbounded clique-width is the theorem by Seese [See96] stating that every FO-definable property admits a linear-time algorithm on graph classes of bounded degree. Eppstein [Epp00] observed that many graph classes of unbounded tree-width, e.g., planar graphs, bounded degree graph classes, share a property: the tree-width of every *r-neighborhood*[1] of every vertex depends only on r. Such graph classes are said to have bounded *local tree-width*. Eppstein [Epp00] used this fact and generalized to graph classes that exclude an *apex graph*[2] as a minor the polynomial-time approximation algorithms for many NP-complete problems (minimum independent set, H-matching, ...) by Baker [Bak94] on planar graphs. Frick and Grohe [FG01b, Fri04] used the locality theorem of FO formulas by Gaifman [Gai82] and showed that every FO-definable property can be checked in almost linear-time on graph classes of bounded local tree-width, which generalized the Seese's result. However, it is not the only way to get similar algorithmic results to the ones for tree/rank-width since many graph classes do not have bounded local tree-width. For instance, Grohe et al. considered polynomial-time algorithms for FO-definable properties on graph classes that exclude a fixed graph as a minor [FG01a] or that locally exclude a minor [DGK07] and, Nešetřil and Ossona de Mendez considered polynomial-time algorithms for some FO-definable properties on graph classes of bounded expansion [NdM06a]. Note that graph classes of bounded expansion and graph classes that locally exclude a minor are incomparable. The surveys [Gro07, Kre08] reviews algorithmic results on the graph classes considered in [FG01b, FG01a, DGK07] and others.

---

[1] The *r-neighborhood* of a vertex x is the set of vertices at distance at most r from x.
[2] An *apex graph* is a graph such that for some vertex v, the *apex*, $G \setminus v$ is planar.

These decompositions can be also linked with results on graph structure. For instance, a minor-closed class of graphs has bounded local tree-width if and only it does not contain all apex-graphs as minors [Epp00, DH04a, DH04b]. There is no known such characterization for graph classes of bounded local tree/rank-width. The survey [KM07] presents some structural theorems and algorithmic applications on minor-closed classes of graphs.

The decompositions based on local boundedness of graph invariants are not the only ways to extend results by Courcelle et al. [Cou90, CMR00]. Another technique consists in generalizing the notion of (strong) tree-decomposition as considered in [Sch97, DK05]. Precisely, we can ask, in the definition of tree-decomposition (see Definition 7.1), the graph T to be in a class $\mathcal{H}$ of graphs. Hence, we have a notion of $\mathcal{H}$-*decomposition* and of $\mathcal{H}$-*width*. Ina Schiering [Sch97] considered such decompositions in a logical point of view and proved, in particular, that the satisfaction of an MS formula φ on a graph G can be translated into the satisfaction of an MS formula $\widetilde{\varphi}$ on the $\mathcal{H}$-decomposition of G. A consequence of this result is that if every MS-definable property admits a polynomial-time algorithm on $\mathcal{H}$, then every MS-definable property admits a polynomial-time algorithm on graph classes of bounded $\mathcal{H}$-width. On the other hand, Diestel and Kühn [DK05] investigated the notion of strong $\mathcal{H}$-width (see Definition 7.2 for the notion of strong tree-width). Their goal was to define a hierarchy on graphs with the minor relation in order to give more structural informations than the one stating that a class of graphs has unbounded tree-width if and only it contains all planar graphs as minors. However, it is difficult to derive important structural properties than that of sparsity with their proposed hierarchies because the proposed hierarchies either are too naive or contain too many levels [DK05].

We now review some results on labeling schemes.

Labeling Schemes

We first define formally the notion of labeling schemes and then we recall some results.

Definition 7.6 (Labeling Scheme) Let Σ be a relational structure and let $\mathfrak{A} = \langle\, \mathfrak{A}, (R_{\mathfrak{A}})_{R \in \Sigma}\, \rangle$ be a relational Σ-structure. Let $f : \mathbb{N} \to \mathbb{N}$ be a mapping. An *f-labeling of* $\mathfrak{A}$ is an injective mapping $J : A \to \{0,1\}^*$ such that for every $x \in A$, $|J(x)| \leq O(f(|A|))$. If $Y \subseteq A$, we let $J(Y)$ be the set $\{J(y) \mid y \in Y\}$. Y is defined from $J(Y)$ by injectivity. Let $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a property on relational Σ-structures. An *f-labeling scheme for P on a class of relational Σ-structures* $\mathcal{C}$ is a pair of two algorithms $(\mathcal{A}, \mathcal{B})$ where:

(L1) $\mathcal{A}$ constructs for each Σ-structure $\mathfrak{A} \in \mathcal{C}$ an f-labeling J.

(L2) For every $a_1, \ldots, a_m \in A$ and $W_1, \ldots, W_q \subseteq A$, $\mathcal{B}$ verifies whether $P(a_1, \ldots, a_m, W_1, \ldots, W_q)$ holds in $\mathfrak{A}$ by taking as input the $(m + q)$-tuple $(J(a_1), \ldots, J(a_m), J(W_1), \ldots, J(W_q))$.

For a mapping $f : \mathbb{N} \to \mathbb{N}$, a class $\mathcal{C}$ of relational Σ-structures admits an f-labeling scheme for a class of properties $\mathcal{P}$ if $\mathcal{C}$ admits an f-labeling scheme for each property $P \in \mathcal{P}$.

We will say that a class of structures $\mathcal{C}$ admits a *short labeling scheme for a property P (resp. a class of properties $\mathcal{P}$)* if there exists an f-labeling scheme for P (resp. $\mathcal{P}$) on $\mathcal{C}$ such that for every n-vertex graph G in $\mathcal{C}$, we have $n = O(exp(f(n)))$.

In distributed networks some properties are frequently asked and the nodes of the network must act locally, i.e., the nodes do not have a global knowledge of the network, in order to give an answer. For instance, when routing informations in distributed networks, each node must decide if there exists a path between itself and the destination by using its *local knowledge* of the network. One solution consists in storing the whole graph in each node. However, the sizes of the considered networks are often huge and because of space constraints we cannot store all the graph in each node. Moreover, if we use the whole network, the answer to each query often takes at least linear-time in the size of the network and we would like to reduce this time because of the sizes of the networks. One way for addressing these problems, i.e., each node acts locally and gives the answer relatively quickly, is to use labeling schemes[3]. However, we are interested in short labeling schemes because of space constraints and this raises two questions:

1. Given a property P, for what classes of graphs can we construct a short labeling scheme?

2. Given a class of graphs $\mathcal{C}$, can we characterize the properties that admit a short labeling scheme on $\mathcal{C}$?

Short labeling schemes for some particular properties are studied for many graph classes that have unbounded clique-width. For instance, it is proved in [GL07] that we can label the vertices of every n-vertex graph that exclude a fixed graph H as minor with labels of size $2\log(n) + O(\log(\log(n)))$ and checks the adjacency of two vertices by using their labels. Other properties are also studied, e.g., distance [GP03a, GP03c, GKK$^+$01], routing [AGMW07, AGM$^+$08] or connectivity [Kor07b]. Some $\log^2$-labeling schemes for distance are given in [GKK$^+$01], e.g., for graph classes of bounded tree-width or chordal graphs. However, there are some negative results. For instance, Gavoille et al. [GPPR04] showed that the minimal size of labels for distance is $\Theta(\log^2(n))$ on trees with n nodes (to be compared with the log-labeling scheme for interval graphs [GP03b]) and is $\Omega(n^{1/3})$ for n-vertex planar graphs. Korman [Kor07b] gave a $(k^2 \cdot \log)$-labeling scheme for k-vertex connectivity in all graphs.

Similarly to the meta-theorems constructing polynomial-time algorithms for MS-definable properties on graph classes of bounded tree/clique-width, Courcelle and Vanicat [CV03] gave a meta-theorem for labeling schemes of MS-definable properties, stated in the following and that will be used as a tool for certain cases of graph classes of unbounded clique-width.

Theorem 7.4 *Let k be a positive integer. Then*

1. *For every MS_1-definable property $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$, there exists a* log*-labeling scheme $(\mathcal{A}, \mathcal{B})$ for P on the class of graphs of clique-width at most k. Moreover, $\mathcal{A}$ computes the labels in $O(n^3)$-time or in $O(n \cdot \log(n))$-time if the clique-width expression of the input n-vertex graph is given.*

2. *For every MS_2-definable property $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$, there exists a* log*-labeling scheme $(\mathcal{A}, \mathcal{B})$ for P on the class of graphs of tree-width at most k. Moreover, $\mathcal{A}$ computes the labels in $O(n \cdot \log(n))$-time, n is the number of vertices of the input graph.*

---

[3]Note that if we remove the restriction of locality when answering the queries, another way is the use of *oracles*, i.e., a central node that answers each query. In general, there is a pre-computation of the network in order to answer queries as quickly as possible, more often in at most logarithmic-time, and the size of the data structure constructed by the pre-computation algorithm should be linear in the size of the network. This is done in the articles by Thorup et al. [DTCR08, PT07, Tho07, TZ05].

The proof of Theorem 7.4 combines the construction of [CV03] that works for graphs given with their decompositions, and parsing results by Bodlaender [Bod96] for tree-width and, by Hliněný , Oum and Seymour [HO07, OS06] for clique-width. Notice also that this theorem relies deeply on tree-automaton and the fact that every graph of clique-width k is generated by a 3-balanced term in $T(F_{k'}, C_{k'})$ where $k' = k \cdot 2^k$ (more precise results are given in [CV03] and Theorem 6.3). A priori, the fact that many interesting graph classes do not have bounded clique-width limits the impact of this theorem. However, we will see in the second part of this thesis that it can be useful for labeling schemes of some properties on graph classes that have bounded local clique-width.

Overview of the Results

In Chapter 8 we prove that every FO-definable property admits a log-labeling scheme on certain classes of graphs of bounded local clique-width that contain planar graphs, unit-interval graphs, graphs of bounded degree, ... We also prove that the FO-properties considered in [NdM06a] also admit log-labeling schemes on graph classes of bounded expansion.

In Chapter 9 we consider the particular property of connectivity. Precisely, we denote by $Conn(x, y, X, F)$ the graph property that expresses that x and y are connected by a path that avoids vertices in X and edges in F. We prove that this property admits a log-labeling scheme on planar graphs.

In our investigations of graph classes of unbounded clique-width, we are also interested in graph classes that are constructed by gluing graphs of small clique-width with limited overlaps. We introduce two new decompositions and associated widths in the spirit of [Sch97, Die05, WT07], one based on partitions of edges, called $\mathcal{H}$-e-decomposition and the other on partitions of vertices, called $\mathcal{H}$-v-decomposition. We prove in Chapter 10 that the property $Conn(x, y, X, \emptyset)$ admits a log-labeling scheme on some graph classes of small $\mathcal{H}$-e-width (resp. $\mathcal{H}$-v-width).

Chapter 8

Labeling Schemes for FO-definable Properties

This chapter is organized as follows. In Section 8.1 we review the locality theorem by Gaifman [Gai82] and a decomposition of some FO formulas by Frick [Fri04]. In Section 8.2 we recall the definition of *local tree-width* and we recall some results about it. In Section 8.3 we introduce the notion of *local clique-width* and of *nicely locally cwd-decomposable*, a notion similar to the one of *nicely locally tree-decomposable* introduced by Frick [Fri04]. We recall some results about local bounded clique-width for completeness, also from [Gro07]. In Section 8.4 we prove that every FO-definable property admits a log-labeling scheme on nicely locally cwd-decomposable graph classes. We conclude by some remarks in Section 8.5.

8.1 Review of Tools from Logic

Let G be a graph and let x and y be in V_G. For a positive integer r, we let $d_G(x, y)$ be the distance between x and y in G and for a subset X of V_G, we let $N_G^r(X)$, the *r-neighborhood of X*, be the set $\{y \mid d_G(x, y) \leq r \text{ for some } x \in X\}$.

Let us first recall some results by Gaifman [Gai82] and Frick [Fri04] that show how to decompose first order formulas into simpler formulas. Their decompositions are given for FO formulas without free set variables. However, those decompositions extend to FO formulas with free set variables since each free set variable can be seen as an unary relation and distance does not depend on unary relations.

Definition 8.1 (Local Formulas) Let r and t be positive integers.

1. An FO formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ is *t-local around* $(x_1, \ldots, x_m)$ if for every G, every $a_1, \ldots, a_m \in V_G$ and every $W_1, \ldots, W_q \subseteq V_G$, we have

$$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \text{ iff } G[N] \models \varphi(a_1, \ldots, a_m, W_1 \cap N, \ldots, W_q \cap N)$$

where $N = N_G^t(\{a_1, \ldots, a_m\})$.

2. An FO formula $\varphi(Y_1, \ldots, Y_q)$ is *basic (t,s)-local* if it is equivalent to a formula of the form

$$\exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \leq i < j \leq s} d(x_i, x_j) > 2t \ \wedge \ \bigwedge_{1 \leq i \leq s} \psi(x_i, Y_1, \ldots, Y_q) \right)$$

where $\psi(x, Y_1, \ldots, Y_q)$ is t-local around its unique free variable x.

For instance, the query $d(x, y) \leq 2t$ is a t-local formula around $\{x, y\}$ since two vertices x and y are at distance at most $2t$ in G if and only if they are at distance at most $2t$ in the t-neighborhood of $\{x, y\}$.

Informally, the satisfaction of a t-local formula $\varphi(\bar{x}, \overline{Y})$ around $\bar{x}$ depends only on the t-neighborhood of $\bar{x}$. And a basic (t, s)-local formula is satisfied in a graph if and only if there exist s pairwise disjoint t-neighborhoods, each of them satisfying a t-local formula around one free FO variable. We can now recall the decomposition of FO formulas into t-local and basic (t, s)-local formulas. This decomposition says roughly that a formula $\varphi(\bar{x}, \overline{Y})$ is satisfied in a graph if and only if there exist pairwise disjoint sufficiently large neighborhoods of $\bar{x}$ that satisfy some local properties.

Theorem 8.1 ([Gai82]) *Every FO formula $\varphi(\bar{x}, \overline{Y})$ with the free variables $\bar{x} = (x_1, \ldots, x_m)$ and $\overline{Y} = (Y_1, \ldots, Y_q)$ is logically equivalent to a Boolean combination $B(\varphi_1(\overline{u_1}, \overline{Y}), \ldots, \varphi_p(\overline{u_p}, \overline{Y}), \psi_1(\overline{Y}), \ldots, \psi_h(\overline{Y}))$ where:*

- *each φ_i is a t-local formula around $\overline{u_i} \subseteq \bar{x}$,*

- *each ψ_i is a basic (t', s)-local formula.*

If φ is a sentence, then only basic (t', s)-local sentences occur in the Boolean combination B. Furthermore, B can be computed effectively, and $t' \leq 7^{q-1}$, $s \leq m + q$ and $t \leq \frac{1}{2}(7^q - 1)$ where q is the quantifier-rank of φ.

In Theorem 8.1 one can hope that the number of t-local formulas and of basic (t', s)-local sentences is bounded. Unfortunately, Dawar et al. [DGKS07] showed that the number of local formulas is necessarily *explosive*, i.e., is not bounded by any function.

With Theorem 8.1 in order to give short labeling schemes for FO-definable properties on graph classes of bounded local clique-width[1] we must be able to check the validity of basic (t', s)-local sentences and also be able to give short labeling schemes for t-local formulas. However, if it is *easy* to verify basic (t', s)-local sentences on graph classes of bounded local clique-width (Lemma 8.2), short labeling schemes for t-local formulas is not so easy. Indeed, our objective is to use Theorem 7.4, which concerns graph classes of bounded clique. Therefore, we need to decompose graphs into sub-graphs of small clique-width with limited overlaps and such that the t-neighborhood of each tuple is contained in a sub-graph. But, such a decomposition is too restrictive. Frick [Fri04] met the same difficulty for the counting of solutions of FO formulas on graph classes of bounded local tree-width. To overcome this difficulty, he distinguished tuples by using the intersection of their t-neighborhoods and gave a decomposition of t-local formulas. We will see that this decomposition is also useful in the

---

[1]The notion of local clique-width and of local tree-width is given in Definition 8.2

labeling schemes for FO-definable properties on graph classes of bounded local clique-width. We now introduce this decomposition of t-local formulas.

Let $m, t \geq 1$. The *t-distance type* of an m-tuple $\bar{a}$ is the undirected graph $\varepsilon = ([m], E_\varepsilon)$ where $E_\varepsilon(i, j)$ iff $d(a_i, a_j) \leq 2t + 1$. The satisfaction of a t-distance type by an m-tuple can be expressed by a t-local formula:

$$\rho_{t,\varepsilon}(x_1, \ldots, x_m) = \bigwedge_{(i,j) \in E_\varepsilon} d(x_i, x_j) \leq 2t + 1 \ \wedge \bigwedge_{(i,j) \notin E_\varepsilon} d(x_i, x_j) > 2t + 1.$$

A *t-connected m-tuple* is an m-tuple that satisfies a connected t-distance type for m-tuples. We now recall a normal form for t-local formulas by Frick [Fri04] that uses t-distance types.

Lemma 8.1 ([Fri04]) *Let $\varphi(\bar{x}, Y_1, \ldots, Y_q)$ be a t-local formula around $\bar{x} = (x_1, \ldots, x_m)$, $m \geq 1$. For each t-distance type ε with $\varepsilon_1, \ldots, \varepsilon_p$ as connected components, one can compute a Boolean combination $F^{t,\varepsilon}(\varphi_{1,1}, \ldots, \varphi_{1,j_1}, \ldots, \varphi_{p,1}, \ldots, \varphi_{p,j_p})$ of formulas $\varphi_{i,j}$ with free variables in $\bar{x}$ and in $(Y_1, \ldots, Y_q)$ such that:*

- *the free FO variables of each $\varphi_{i,j}$ are among $\bar{x} \mid \varepsilon_i$ ($\bar{x} \mid \varepsilon_i$ is the restriction of $\bar{x}$ to ε_i),*

- *each $\varphi_{i,j}$ is t-local around $\bar{x} \mid \varepsilon_i$ ($\bar{x} \mid \varepsilon_i$ is a t-connected s-tuple where s is the size of the connected component ε_i),*

- *for each m-tuple $\bar{a}$ and each q-tuple of sets $(W_1, \ldots, W_q)$, $G \models \rho_{t,\varepsilon}(\bar{a}) \ \wedge \ \varphi(\bar{a}, W_1, \ldots, W_q)$ if and only if $G \models \rho_{t,\varepsilon}(\bar{a}) \ \wedge \ F^{t,\varepsilon}(\ldots, \varphi_{i,j}(\bar{a} \mid \varepsilon_i, W_1, \ldots, W_q), \ldots)$.*

This lemma says that given φ, a t-local formula around an m-tuple and ε, a t-distance type of m-tuples, one can find a Boolean combination $F^{t,\varepsilon}$ of t-local formulas, each around t-connected tuples, and such that if an m-tuple satisfies ε, then it satisfies φ if and only if it satisfies $F^{t,\varepsilon}$. Then given φ, a t-local formula around an m-tuple and a decomposition of a graph that guarantees that the $m \cdot (2t + 1)$-neighborhood of each vertex is contained in sub-graph of small clique-width, we can guarantee that each connected tuple of size at most m is contained in a sub-graph of small clique-width. This will allow us to give short labeling schemes for t-local formulas as proved in Lemma 8.5.

8.2 Graph Classes of Bounded Local Tree-Width

We first give the following definition.

Definition 8.2 (Local Width) Let gp-width be a graph parameter where the gp-width of a graph G is denoted by $gp(G)$.

1. The *local gp-width* of a graph G is the function $lgp^G : \mathbb{N} \to \mathbb{N}$ defined by $lgp^G(t) = \max\{gp(G[N_G^t(a)]) \mid a \in V_G\}$.

2. A class $\mathcal{C}$ of graphs has *bounded local gp-width* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $lgp^G(t) \leq f(t)$ for every $G \in \mathcal{C}$ and $t \in \mathbb{N}$.

Eppstein defined in [Epp00] the notion of *local tree-width* for graph isomorphisms and in order to generalize the algorithms by Baker for planar graphs [Bak94] to graph classes that exclude an apex graph as minor. He proved in particular that every minor-closed class of graphs has bounded local tree-width if and only if it does not contain all apex-graphs as minors. An *apex graph* is a graph such that for some vertex v, the *apex*, $G \setminus v$ is planar. Demaine et al. [DH04a, DH04b] proved that the local tree-width of every minor-closed class of graphs of bounded local tree-width is bounded by a linear function (The function given by Eppstein [Epp00] was doubly exponential).

Frick and Grohe [FG01b], by using Theorem 8.1, proved that every *FO*-definable property can be checked in almost linear-time on classes of graphs of bounded local tree-width and in linear-time on minor-closed classes of graphs of bounded local tree-width. In database theory we are in general not only interested that a property is verified, but we want also to count the set of solutions or to list the set of solutions. For instance, for every *MS*-definable property we can count the set of solutions in linear-time on graph classes of bounded tree-width [CMR01, FMR08]. We can also list the set of solutions in time proportional to the set of solutions after a linear-time pre-processing [CMR01]. Frick [Fri04] considered the counting and listing problem on graph classes of bounded local tree-width. By Theorem 8.1, for every *FO* formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$, there exist positive integers r, s and t such that the satisfiability of φ on a graph G depends only on the r-neighborhoods of $\{x_1, \ldots, x_m\}$ and on the t-neighborhoods of s vertices that are at distance at least $2t$. Then to have a chance to list or to count the set of solutions for an *FO*-definable property $P(x_1, \ldots, x_m)$ on graph classes of bounded local tree-width, an idea is to cover these graphs by sub-graphs of small tree-width, in order to use results in [CMR01], that have limited intersections, and such that the r-neighborhood of each m-tuple is contained in at least one of these sub-graphs. However, this covering is too restrictive: we cannot at the same time restrict the intersections of the covers and ask for the r-neighborhood of each m-tuple to be included in a cover, that moreover must have small tree-width. What happens if we ask for the r-neighborhood of each vertex, instead of each m-tuple, to be included in a sub-graph of small tree-width, but the union of q such sub-graphs must have tree-width that depends on q? This will allow to solve at least the case of formulas with only one free *FO* variable and for instance on planar graphs that admit such decompositions. In fact, Frick [Fri04] proved that such decompositions are enough for the counting and the listing of solutions for *FO* formulas by using Lemma 8.1. For that purpose he introduced the notion of *nicely locally tree-decomposable* graph classes and proved that for every *FO* formula, we can count the number of solutions in linear-time and list the set of solutions in time proportional to the size of the set of solutions [Fri04][2]. We will see in Section 8.4 that this notion of nicely locally tree-decomposable is also enough for labeling schemes. We now define, in a more general context, the notion of nicely locally tree-decomposable.

Definition 8.3 (Covering Graphs) Let gp-width be a graph parameter where the gp-width of a graph G is denoted by $gp(G)$. Let $r, \ell \geq 1$ and $g : \mathbb{N} \to \mathbb{N}$. An (r, ℓ, g)-*gp cover* of a graph G is a family $\mathcal{T}$ of subsets of V_G such that:

(CC1) For every $a \in V_G$, there exists a $U \in \mathcal{T}$ such that $N_G^r(a) \subseteq U$.

(CC2) For each $U \in \mathcal{T}$, there exist less than ℓ many $V \in \mathcal{T}$ such that $U \cap V \neq \emptyset$.

---

[2]Durand et al. improved this result on graph classes of bounded degree [BDG07] by using different techniques based on elimination of quantifiers.

(CC3) For each $U \in \mathcal{T}$, we have $gp(G[U]) \leq g(1)$.

An (r, ℓ, g)-gp cover is *nice* if condition (CC3) is replaced by condition (CC4) below:

(CC4) For all $U_1, \ldots, U_q \in \mathcal{T}$, for $q \geq 1$, we have $gp(G[U_1 \cup \cdots \cup U_q]) \leq g(q)$.

A class $\mathcal{C}$ of graphs is *(nicely) locally gp-decomposable* if there is a polynomial-time algorithm that given a graph $G \in \mathcal{C}$ and $r \geq 1$, computes a (nice) (r, ℓ, g)-gp cover of G for suitable ℓ and g depending on r.

As in [Fri04], we incorporate the polynomial-time requirement in order to minimize the number of notions to be used.

It is clear that if a class of graphs is nicely locally gp-decomposable, then it is locally gp-decomposable. Notice that a locally gp-decomposable class of graphs has also bounded local gp-width. We do not know if every class of graphs of bounded local tree-width is (nicely) locally tree-decomposable and it seems not. However, important classes of graphs of bounded local tree-width are nicely locally tree-decomposable.

Example 8.1   1. Every graph of bounded tree-width is obviously nicely locally tree-decomposable.

2. Frick proved that classes of graphs of bounded degree and planar graphs are nicely locally tree-decomposable [Fri04].

3. Frick and Grohe proved that every minor-closed class of graphs of bounded local tree-width is nicely locally tree-decomposable [Fri04, FG01b].

We notice that the notion of covering a graph by *blocks* by imposing some properties on the blocks is not new. For instance, Peleg [Pel93] showed that for every positive integer r, every graph can be covered by sub-graphs of small radius (depending on r) and such that the r-neighborhood of each vertex in contained in some sub-graph; this covering can be found in polynomial-time and is independent of the notions of tree-width and of clique-width. Another example of covering of graphs can be found in [AGMW07] where Abraham et al. proved that every graph that excludes a fixed graph as a minor can be decomposed into sub-graphs of small diameter and with limited overlaps.

8.3   Graph Classes of Bounded Local Clique-Width

Planar graphs, graph classes of bounded degree, apex-minor-free graph classes have bounded local tree-width and are in fact nicely locally tree-decomposable [Fri04]. However, many classes of graphs do not have bounded local tree-width. Many classes of graphs that do not have bounded tree-width have bounded clique-width. What can we say about classes of graphs of bounded *local clique-width* and graph classes that are (nicely) locally cwd-decomposable? We investigate these classes. One can easily adapt the results by Frick and Grohe [FG01b] in order to prove that every FO-definable property can be checked in polynomial-time on classes of graphs of bounded local clique-width (Lemma 8.2). There is no known minor inclusion related to clique-width, however one exists for rank-width, the vertex-minor inclusion (Section

1.2). Since rank-width is equivalent to clique-width (Proposition 1.2), one can then ask for a characterization of vertex-minor closed classes of graphs of bounded local rank-width, hence of bounded local clique-width.

We now give examples of graph classes of bounded local clique-width. For every graph G and every positive integer m, we let G^m be the graph obtained from G by adding edges between x and y whenever $d_G(x,y) \leq m$.

Fact 8.1   *(i) Each class of graphs of bounded clique-width has bounded local clique-width.*

 (ii) Each class of graphs of bounded local tree-width has bounded local clique-width. Each (nicely) locally tree-decomposable class of graphs is (nicely) locally cwd-decomposable.

 (iii) The class of unit-interval graphs has bounded local clique-width and is nicely locally cwd-decomposable.

 (iv) Let m be a positive integer. Let $\mathcal{C}$ be a class of graphs of bounded local clique-width. Then $\mathcal{C}^m = \{G^m \mid G \in \mathcal{C}\}$ has bounded local clique-width.

Proof. (i) If a graph G has clique-width k, then for every $W \subseteq V_G$, we have $cwd(G[W]) \leq k$ [CO00]. Then statement (i) is verified.

(ii) It is known that if a class of graphs has bounded tree-width, then it has bounded clique-width [CO00, CR05]. Then statement (ii) is verified.

(iii) We will use a result by Lozin in [Loz08]. We let $H_{n,m}$ be the graph $\langle V_1 \cup \cdots \cup V_n, E^1 \cup E^2 \rangle$ with nm vertices such that:

$$
\begin{aligned}
V_i &= \{v_{i,1}, \ldots, v_{i,m}\}, \\
E^1 &= \bigcup_{1 \leq i \leq n} \{v_{i,j}v_{i,\ell} \mid j, \ell \leq m, \ j \neq \ell\}, \\
E^2 &= \bigcup_{1 \leq i \leq n-1} \{v_{i,j}v_{i+1,\ell} \mid j \leq \ell \leq m\}.
\end{aligned}
$$

Each subgraph induced by V_i is a complete graph. Figure 23 shows the graph $H_{4,4}$. It is proved in [Loz08] that the clique-width of $H_{n,m}$ is at most $3n$. Moreover, every unit-interval graph with n vertices is an induced sub-graph of $H_{n,n}$ [Loz08]. These two properties will be used.

We first prove that unit-interval graphs have bounded local clique-width. Let G be a unit-interval graph with n vertices. Then for every positive integer r and every vertex x of G, the subgraph $G[N_G^r(x)]$ is isomorphic to an induced subgraph of $H_{r,n}$. Thus, for every vertex x of G and every positive integer r, $G[N_G^r(x)]$ has clique-width at most $3r$. (Bagan gives in [Bag09] another proof stating that unit-interval graphs have bounded local clique-width.)

We now prove that the class of unit-interval graphs is nicely locally cwd-decomposable. Let G be a unit-interval graph with n vertices. Hence, it is a subgraph of $H_{n,n} = \langle V_1 \cup \cdots \cup V_n, E^1, E^2 \rangle$. Without loss of generality, we may assume G connected. We can also assume that $V_G = \bigcup_{1 \leq i \leq n} V_i'$ where $V_i' = \{v_{i,i_1}, \ldots, v_{i,i_\ell}\}$ with $1 \leq i_1 \leq i_2 \leq \cdots \leq i_\ell \leq n$. For each

$1 \leq i \leq n$, we let $U_i = N_G^{r+1}(v_{i,i_1})$. We let $g : \mathbb{N} \to \mathbb{N}$ be defined by $g(q) = 3 \cdot q \cdot (r+1)$. We claim that $\{U_i \mid 1 \leq i \leq n\}$ is a nice $(r, 2r+3, g)$-cwd cover. It is clear by construction that for every $1 \leq i \leq n$ and every vertex v in V_i', the set $N_G^r(v)$ is a subset of U_i.

We now prove that for every positive integer q, if we take q subsets $U_{j_1}, \ldots, U_{j_q}$, then the subgraph $G[U_{j_1} \cup \cdots \cup U_{j_q}]$ has clique-width at most $3 \cdot q \cdot (r+1)$. Assume that $j_1 \leq j_2 \leq \cdots \leq j_q$ and let $G_1, \ldots, G_p$ be the connected components of $G[U_{j_1} \cup \cdots \cup U_{j_q}]$. We need only prove the claim for each connected component. Let G_1 be one of them. It is of the form, without loss of generality, $G[U_{j_1} \cup \cdots \cup U_{j_{\ell_1}}]$ with the property that $j_1 \leq \cdots \leq j_{\ell_1}$. Thus, G_1 is an induced subgraph of $H_{j_{\ell_1} \cdot (r+1), n}$, hence has clique-width at most $3 \cdot \ell_1 \cdot (r+1)$. Hence, the clique-width of $G[U_{j_1} \cup \cdots \cup U_{j_q}]$ is at most $3 \cdot q \cdot (r+1)$.

Let v be a vertex in V_i' for $1 \leq i \leq n$. By construction, v can only be in $U_i, U_{i-1}, \ldots, U_{i-(r+1)}, U_{i+1}, \ldots, U_{i+(r+1)}$. Thus, v is in at most $2(r+1)+1$ sets U_i. This concludes the proof.

(iv) Let us sketch the proof. Let G be a graph in $\mathcal{C}$. For every vertex x of G and every positive integer r, we have $N_{G^m}^r(x) = N_G^{rm}(x)$. One easily verifies that $G^m[N_{G^m}^r(x)] = G'[N_G^{rm}(x)]$ where $G' = (G[N_G^{(r+1)m}(x)])^m$. It is proved in [ST07] that if a graph H has clique-width k, then H^m has clique-width at most $4 \cdot (m+1)^k$. Hence, for every graph G in $\mathcal{C}$ and every positive integer r, $lcwd^{G^m}(r) \leq 4 \cdot (m+1)^{f(r(m+1))}$ where f is the function that bounds the local clique-width of graphs in $\mathcal{C}$. ∎
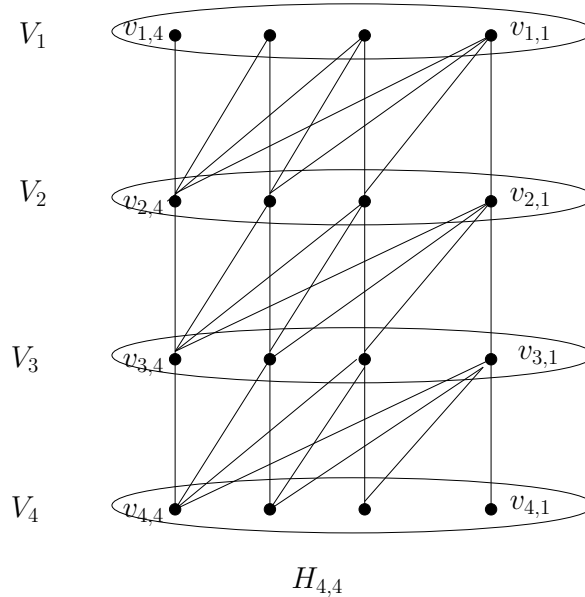

Figure 23: The graph $H_{4,4}$. Each V_i, for $1 \leq i \leq 4$, induces a clique.

We now prove that every FO formula can be checked in polynomial-time on graph classes that have bounded local clique-width. For a graph G and $W \subseteq V_G$, we let the r-kernel of W in G be the set $K_G^r(W) = \{a \in V_G \mid N_G^r(a) \subseteq W\}$.

Lemma 8.2 ([FG01b]) *Let $\mathcal{C}$ be a class of graphs of bounded local clique-width. Then for every graph G with n vertices in $\mathcal{C}$ and every FO formula φ, we can decide if G verifies φ in $O(n^4)$-time.*

Frick and Grohe proved in [FG01b, Lemma 8.3] that given a graph G which is in a class $\mathcal{C}$ of graphs of bounded local tree-width, two positive integers r and m and a subset P of V_G, one can check in linear-time if there exist $a_1, \ldots, a_m$ in P that are pairwise at distance at least r. With help of Theorem 1.2 the same thing can be proved for graph classes of bounded local clique-width, stated in the following.

Lemma 8.3 ([FG01b, Gro07]) *Let $\mathcal{C}$ be a class of graphs of bounded local clique-width and let r and m be two positive integers. Then there exists an $O(n^3)$-time algorithm that given an n-vertex graph $G \in \mathcal{C}$ and $P \subseteq V_G$, decides if there exist $a_1, \ldots, a_m$ in P such that $d_G(a_i, a_j) > r$ for all $1 \le i < j \le m$.*

We can now prove Lemma 8.2.

Proof of Lemma 8.2. It is enough to verify the statement for *FO* sentences. By Theorem 8.1 a graph verifies an *FO* sentence if and only if it verifies a Boolean combination of basic (t, s)-local sentences for some positive integers t and s. We can then only show how to verify basis (t, s)-local sentences. By definition, a basic (t, s)-local sentence φ is of the form:

$$\exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \le i < j \le s} d(x_i, x_j) > 2t \ \wedge \ \bigwedge_{1 \le i \le s} \psi(x_i) \right)$$

where $\psi(x)$ is a t-local formula around x. Let G be an n-vertex graph that is in a class $\mathcal{C}$ of graphs of bounded local clique-width and let f be the function that bounds the local clique-width of graphs in $\mathcal{C}$. For each a in V_G, we can compute $N_G^t(a)$, of size at most n, in $O(n^2)$-time. By Theorem 1.2 and Proposition 1.2, for each a in V_G, we can construct in $O(n^3)$-time a term $t \in T(F_k, C_k)$, $k \le 2^{f(t)+1}$ such that $val(t) = G[N_G^t(a)]$. We can decide if $val(t)$ verifies $\psi(a)$ in linear-time by using t [CMR00]. Therefore, we can compute the set $P = \{a \in V_G \mid G \models \varphi(a)\}$ in $O(n^4)$-time. It is clear that φ is true in G if and only if there exists $a_1, \ldots, a_m \in P$ such that $d(a_i, a_j) > 2r$ for $1 \le i < j \le m$. And this can be verified in $O(n^3)$-time by Lemma 8.3. This finishes the proof. ∎

8.4   Labeling for *FO*-definable Properties

Our objective is to prove the following.

Theorem 8.2   *1. Every FO-definable property without free set variables admits a log-labeling scheme on each locally cwd-decomposable class of graphs.*

   2. Every FO-definable property admits a log-labeling scheme on each nicely locally cwd-decomposable class of graphs.

With Theorem 8.1 and Lemma 8.2 the next step before proving Theorem 8.2 is to show that each t-local formula admits a log-labeling scheme on locally cwd-decomposable classes of graphs as shown by the lemma below.

Lemma 8.4 *Let $\mathcal{C}$ be a locally cwd-decomposable class of graphs and let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a t-local formula around $\bar{x} = (x_1, \ldots, x_m)$. Then there exists a* log-*labeling scheme $(\mathcal{A}, \mathcal{B})$ for φ on $\mathcal{C}$. Furthermore, $\mathcal{A}$ computes the labels in polynomial-time.*

Proof. Let G be in a locally cwd-decomposable class with $n = |V_G|$. We first construct the labeling and we therefore explain how to decide φ by using these labels. We let $\mathcal{T}$ be an (r, ℓ, g)-cwd cover of G where $r = m \cdot (2t + 1)$. We recall that each $G[U]$ for $U \in \mathcal{T}$ has clique-width at most $g(1)$. We assume that each $U \in \mathcal{T}$ has an index encoded as a bit string $\ulcorner U \urcorner$. For each vertex x, there exist less than ℓ many $U \in \mathcal{T}$ such that $x \in U$. Therefore, there are at most $n \cdot \ell$ sets in $\mathcal{T}$. Hence $\ulcorner U \urcorner$ has length $O(\log(n))$. We can now construct the labeling of G in order to decide φ.

By Theorem 7.4, we can label each vertex $x \in U$ with a label $L_U(x)$ of size $O(\log(n))$ and decide if $d_{G[U]}(x, y) \leq 2t + 1$ just by using $L_U(x)$ and $L_U(y)$. For each $x \in V_G$, we let:

$$L(x) = \left(\left(\ulcorner U \urcorner, L_U(x) \mid N_G^{2t+1}(x) \subseteq U \right), \left(\ulcorner U \urcorner, L_U(x) \mid x \in U \text{ and } N_G^{2t+1}(x) \nsubseteq U \right) \right).$$

It is clear that $|L(x)| = O(\log(n))$ since each x is in at most ℓ sets U in $\mathcal{T}$. Notice that by using $L(x)$ we can recover the set $\{ \ulcorner U \urcorner \mid x \in U \}$.

Let ε be a t-distance type of m-tuples with $\varepsilon_1, \ldots, \varepsilon_p$ as connected components. By Lemma 8.1, there exists a Boolean combination $F^{t,\varepsilon}(\varphi_{1,1}^\varepsilon, \ldots, \varphi_{1,j_1}^\varepsilon, \ldots, \varphi_{p,1}^\varepsilon, \ldots, \varphi_{p,j_p}^\varepsilon)$ such that for every $a_1, \ldots, a_m \in V_G$ and every $W_1, \ldots, W_q \subseteq V_G$,

$$G \models \rho_{t,\varepsilon}(\bar{a}) \,\wedge\, \varphi(\bar{a}, W_1, \ldots, W_q) \text{ iff } G \models \rho_{t,\varepsilon}(\bar{a}) \,\wedge\, F^{t,\varepsilon}(\ldots, \varphi_{i,j}^\varepsilon(\bar{a} \mid \varepsilon_i, W_1, \ldots, W_q), \ldots).$$

For each $G[U]$ and each $\varphi_{i,j}^\varepsilon$, we apply Theorem 7.4 which constructs a log-labeling $J_{i,j}^{U,\varepsilon}$ on $G[U]$ for $\varphi_{i,j}^\varepsilon$. For each vertex x, we let

$$J_\varepsilon = \left(\left(\ulcorner U \urcorner, J_{1,1}^{U,\varepsilon}(x), \ldots, J_{p,j_p}^{U,\varepsilon}(x) \right) \mid N_G^t(x) \subseteq U \right).$$

It is clear that for each ε and each $x \in V_G$, we have $|J_\varepsilon(x)| = O(\log(n))$. There exist at most $k' = 2^{m(m-1)/2}$ t-distance type graphs for m-tuples; we enumerate them by $\varepsilon^1, \ldots, \varepsilon^{k'}$. For each x, we let

$$J(x) = \left(L(x), J_{\varepsilon^1}(x), \ldots, J_{\varepsilon^{k'}}(x) \right).$$

It is clear that $|J(x)| = O(\log(n))$ and is computed in polynomial-time. Let $a_1, \ldots, a_m \in V_G$ and $W_1, \ldots, W_q \subseteq V_G$. We now explain how to decide if $G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$ by using $J(a_1), \ldots, J(a_m)$ and $J(W_1), \ldots, J(W_q)$.

$\mathcal{T}$ is an (r, ℓ, g)-cwd cover for $r = m(2t + 1)$. Then for every x and y in V_G, we have $d_G(x, y) \leq 2t + 1$ if and only if there exists a $U \in \mathcal{T}$ such that $d_{G[U]}(x, y) \leq 2t + 1$. Therefore,

by using $L(a_1), \ldots, L(a_m)$ from $J(a_1), \ldots, J(a_m)$, we can construct the t-distance type ε satisfied by $a_1, \ldots, a_m$; let $\varepsilon_1, \ldots, \varepsilon_p$ be the connected components of ε. We recover from $J(a_1), \ldots, J(a_m)$ the labels $J_\varepsilon(a_1), \ldots, J_\varepsilon(a_m)$. We use $r = m(2t + 1)$ in order to warranty that if $(a_1, \ldots, a_p)$, $p \leq m$ is a t-connected p-tuple, then there exists a $U \in \mathcal{T}$ such that $N_G^t(a_1, \ldots, a_p) \subseteq U$. Then for every $\bar{a} \mid \varepsilon_i$, by using for instance $J_\varepsilon(b)$ for $b \in \bar{a} \mid \varepsilon_i$, we can recover a set $U_i \in \mathcal{T}$ such that $N_G^t(\bar{a} \mid \varepsilon_i) \subseteq U_i$. It remains to recover $W_j \cap U_i$ for each $j \leq q$. But by using the labels $L(x)$ we can recover the set $\{\ulcorner U \urcorner \mid x \in U\}$, hence we can recover, for each $W \subseteq V_G$ and $U \in \mathcal{T}$, the set $W \cap U$. Therefore, for each $j \leq q$, we can recover $W_j \cap U_i$ from $J(W_j)$. We can then decide if G satisfies $F^{t,\varepsilon}(\varphi_{1,1}^\varepsilon(\bar{a} \mid \varepsilon_1, W_1 \cap U_1, \ldots, W_q \cap U_1), \ldots, \varphi_{p,j_p}^\varepsilon(\bar{a} \mid \varepsilon_p, W_1 \cap U_p, \ldots, W_q \cap U_p))$. And this is sufficient by Definition 8.1 and Lemma 8.1. ∎

We can now prove Theorem 8.2. We will prove the two statements separately.

Proof of Theorem 8.2 (1). Let G belong to a locally cwd-decomposable class with $|V_G| = n$. Let $P(x_1, \ldots, x_m)$ be an *FO*-definable property described by the *FO* formula $\varphi(x_1, \ldots, x_m)$. By Theorem 8.1, φ is equivalent to a Boolean combination $B(\varphi_1(\overline{u_1}), \ldots, \varphi_p(\overline{u_p}), \psi_1, \ldots, \psi_h)$ where each φ_i is a t-local formula around $\overline{u_i} \subseteq \bar{x}$ and each ψ_i is a basic (t', s)-local sentence for suitable t, t' and s.

By Lemma 8.2, we can decide in $O(n^4)$-time each sentence ψ_i. Let $b = (b_1, \ldots, b_h)$ where $b_i = 1$ if G satisfies ψ_i and 0 otherwise. By Lemma 8.4, there exists a log-labeling J_i for each $\varphi_i(\overline{u_i})$, $1 \leq i \leq p$ on G. For each x, we let $J(x) = (J_1(x), \ldots, J_p(x), b)$. It is clear that $|J(x)| = O(\log(n))$. We now explain how to decide $\varphi(a_1, \ldots, a_m)$ just by using $J(a_1), \ldots, J(a_m)$.

From $J(a_1), \ldots, J(a_m)$, we can recover the truth value of each sentence ψ_i. And by Lemma 8.4, we can decide if $G \models \varphi_i(\bar{a} \mid u_i)$ just by using J_i. Therefore, we can decide if G satisfies $B(\varphi_1(\bar{a} \mid u_1), \ldots, \varphi_p(\bar{a} \mid u_p), \psi_1, \ldots, \psi_h)$, hence if G satisfies $\varphi(a_1, \ldots, a_p)$ by using only $J(a_1), \ldots, J(a_m)$. ∎

We can now prove Theorem 8.2 (2). But before let us define the intersection graph of a *cover* of a graph G, i.e., a family $\mathcal{T}$ of subsets of V_G the union of which is V_G.

Definition 8.4 (Intersection Graph) Let G be a graph and let $\mathcal{T}$ be a cover of G. The *intersection graph of $\mathcal{T}$* is the graph $G(\mathcal{T})$ where $V_{G(\mathcal{T})} = \{x_U \mid U \in \mathcal{T}\}$ and $x_U x_V \in E_{G(\mathcal{T})}$ if and only if $U \cap V \neq \emptyset$.

It is clear that if $\mathcal{T}$ is an (r, ℓ, g)-cwd cover of a graph, then $G(\mathcal{T})$ has maximum degree at most ℓ. Let m be a positive integer, a *distance-m coloring* of a graph G is a proper coloring of G^m, where G^m is the graph with V_G as set of vertices and for every $x, y \in V_G$, $xy \in E_{G^m}$ if and only if $d_G(x, y) \leq m$. Then in a proper distance-m coloring, vertices at distance at most m have different colors. If $\Delta(G)$ is the maximum degree of a graph G, then G admits a proper coloring with $\Delta(G) + 1$ colors. Since $G(\mathcal{T})$ has maximum degree at most ℓ, then $G(\mathcal{T})$ admits a proper distance-m coloring with $\ell^{O(m)}$ colors since $G(\mathcal{T})^m$ has maximum degree at most $\ell^{O(m)}$.

We say that two sets of vertices W and W' of a graph G *touch* if $W \cap W' \neq \emptyset$ or there exists an edge between a vertex of W and one of W'. It is clear that if $W = \bigcup_{1 \leq i \leq p} W_i \subseteq V_G$ and W_i, W_j pairwise do not touch, then $G[W]$ is the disjoint union of the graphs $G[W_i]$. It follows that $cwd(G[W]) = \max\{cwd(G[W_i]) \mid 1 \leq i \leq p\}$.

Proof of Theorem 8.2 (2). Let G belong to a nicely locally cwd-decomposable class with $|V_G| = n$. By Lemma 8.4, it is sufficient to consider FO formulas $\varphi(Y_1, \ldots, Y_q)$ of the form:

$$\exists x_1 \cdots \exists x_m \left(\bigwedge_{1 \leq i < j \leq m} d(x_i, x_j) > 2t \ \wedge \ \bigwedge_{1 \leq i \leq m} \psi(x_i, Y_1, \ldots, Y_q) \right)$$

where $\psi(x, Y_1, \ldots, Y_q)$ is t-local around x. We show that there exists a log-labeling scheme for such formulas on nicely locally cwd-decomposable classes of graphs. We consider for the sake of clarity the particular case of $m = 2$.

Let $\mathcal{T}$ be a nice (r, ℓ, g)-cwd cover of G where $r = 2t + 1$. We let γ be a distance-3 coloring of the intersection graph of $\mathcal{T}$. For every 2 colors i and j, we let $G_{i,j}$ be the graph induced by the union of the sets $U \in \mathcal{T}$ of colors i and j (possibly $i = j$).

Claim 8.1 $cwd(G_{i,j}) \leq g(2)$.

Proof of Claim 8.1. Let $\mathcal{T}^2 = \{U \cup U' \mid U, U' \in \mathcal{T}, \ U \cap U' \neq \emptyset\}$. The vertex set of the graph $G_{i,j}$ is a union of sets in $\mathcal{T} \cup \mathcal{T}^2$. No two sets of this union touch: if a set $U \cup U'$ is such that $U \cap U' \neq \emptyset$ and meets some $U'' \in \mathcal{T}$ with $U'' \neq U$ and $U'' \neq U'$, then we have $\gamma(U) = i$, $\gamma(U') = j \neq i$ and U'' meets U or U'. It can have neither color i nor color j because γ is a distance-3 coloring and U, U' and U'' are pairwise at distance at most 2. Now, if there exists an edge between a vertex x in $U \cup U'$ and a vertex y in $U'' \in \mathcal{T}$, then there exists a set $W \in \mathcal{T}$ such that x and y are in W. Hence, U'' and U are at distance at most 3, similarly for U'' and U'. Thus, U'' can have neither color i nor color j. We can then conclude that $G_{i,j}$ is a disjoint union of graphs $G[U \cup U']$ with $U \cup U' \in \mathcal{T}^2$ and of graphs $G[U]$ for $U \in \mathcal{T}$ that do not touch pairwise. Since $cwd(G[U \cup U']) \leq g(2)$, we are done. ∎

Claim 8.2 *Let $x \in K_G^{2t}(U)$ and $y \in K_G^{2t}(U')$ for some $U, U' \in \mathcal{T}$. Then $d_G(x, y) > 2t$ if and only if $d_{G[U \cup U']}(x, y) > 2t$.*

Proof of Claim 8.2. It is clear that if $d_G(x, y) > 2t$, then $d_{G[U \cup U']}(x, y) > 2t$ since $d_G(x, y) \leq d_{G[U \cup U']}(x, y)$. For proving the converse direction, assume that $d_G(x, y) \leq 2t$. Then there exists in G a path of length at most $2t$ from x to y. This path is also in $G[U]$ since $x \in K_G^{2t}(U)$. Hence, it is also in $G[U \cup U']$. Therefore, $d_{G[U \cup U']} \leq 2t$. ∎

Let us now give to each vertex x of G the smallest color i such that $x \in K_G^{2t}(U)$ and $\gamma(U) = i$ and denote it by $\gamma(x)$. Hence, a vertex has one and only one color. We can then consider G as the structure $\langle V_G, E_G, c_{1G}, \ldots, c_{lG} \rangle$ where $l = \ell^{O(1)}$ and $c_{iG}(x)$ holds if and only if x has

color i. For each pair of colors (i, j), we consider the formula $\psi_{i,j}(Y_1, \ldots, Y_q)$ (possibly $i = j$):

$$\exists x \exists y \Big(d(x, y) > 2t \ \wedge \ \psi(x, Y_1, \ldots, Y_q) \ \wedge \ \psi(y, Y_1, \ldots, Y_q) \ \wedge \ c_i(x) \ \wedge \ c_j(y) \Big).$$

By Theorem 7.4, we can construct for each formula $\psi_{i,j}$ a log-labeling, $J_{i,j}$ in the graph $G_{i,j}$. We also compute the truth value $b_{i,j}$ of $\psi_{i,j}(\emptyset, \ldots, \emptyset)$ in $G_{i,j}$; we get a vector b of fixed length by concatenating all $b_{i,j}$. We let $J(x) = \big(\gamma(x), (J_{i,j}(x) \mid x \in V_{G_{i,j}}), b \big)$. It is clear that $|J(x)| = O(\log(n))$.

We now explain how to check the validity of $\varphi(W_1, \ldots, W_q)$ by using $J(W_1), \ldots, J(W_q)$. From $J(x)$, for $x \in W_i$, $1 \le i \le q$, we can recover the color of x. Then from $J(W_1), \ldots, J(W_q)$, we can determine those $G_{i,j}$ such that $V_{G_{i,j}} \cap (W_1 \cup \cdots \cup W_q) \ne \emptyset$, and check if, for one of them, $G_{i,j} \models \psi_{i,j}(W_1 \cap V_{G_{i,j}}, \ldots, W_q \cap V_{G_{i,j}})$. If one is found, we are done. Otherwise, we use the $b_{i,j}$'s to look for $G_{i,j}$ such that $G_{i,j} \models \psi_{i,j}(\emptyset, \ldots, \emptyset)$ and $(W_1 \cup \cdots \cup W_q) \cap V_{G_{i,j}} = \emptyset$. This gives the correct results because of the following facts:

- If $x \in K_G^{2t}(U)$ and $y \in K_G^{2t}(U')$ satisfy the formula ψ (possibly $U = U'$) and $d_G(x, y) > 2t$, then $d_{G_{i,j}}(x, y) > 2t$. Hence, $G_{i,j} \models \psi_{i,j}(W_1 \cap V_{G_{i,j}}, \ldots, W_q \cap V_{G_{i,j}})$ where $i = \gamma(U)$ and $j = \gamma(U')$.

- If $G_{i,j} \models \psi_{i,j}(W_1 \cap V_{G_{i,j}}, \ldots, W_q \cap V_{G_{i,j}})$ then we get $G \models \varphi(W_1, \ldots, W_q)$ by similar argument (in particular $d_{G_{i,j}}(x, y) > 2t$ implies $d_{G[U \cup U']}(x, y) > 2t$ which implies that $d_G(x, y) > 2t$ by Claim 8.2).

For $m = 1$, the proof is similar with γ a distance-2 coloring and we use $G_{i,i}$ instead of $G_{i,j}$. For $m > 2$, the proof is the same. We take for γ a distance-$(m+1)$ coloring of the intersection graph. Then we consider graphs $G_{i_1, \ldots, i_m}$ defined as (disjoint) unions of sets $U_1 \cup \cdots \cup U_m$ for $U_1, \ldots, U_m$ in $\mathcal{T}$, of respective colors $i_1, \ldots, i_m$. We can then prove that $G_{i_1, \ldots, i_m}$ has clique-width at most $g(m)$ by using similar arguments as the ones used for Claim 8.1. This terminates the proof of Theorem 8.2. ∎

Frick in [Fri04] considered the case of counting queries for *FO* formulas on nicely locally tree-decomposable classes, which consist in counting the number of solutions. We will prove in this section that if a property is *FO*-definable, then we can count the set of solutions on nicely locally cwd-decomposable classes of graphs. We define formally the notion of *counting query*.

Definition 8.5 (Counting Query) Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a formula and let G be a finite graph. For $W_1, \ldots, W_q \subseteq V_G$, we let

$$\#_G \varphi(W_1, \ldots, W_q) = \Big| \big\{ (a_1, \ldots, a_m) \in V_G^m \mid G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \big\} \Big|.$$

The *counting query of* φ consists in determining $\#_G \varphi(W_1, \ldots, W_q)$ for given $(W_1, \ldots, W_q)$. Let s be a positive integer. The *counting query of* φ *modulo* s consists in determining $\#_G \varphi(W_1, \ldots, W_q)$ modulo s for given $(W_1, \ldots, W_q)$.

The following theorem is an easy extension of Theorem 7.4.

Theorem 8.3 *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be an MS_1 formula over graphs and $k, s \in \mathbb{N}$. There exists a $\log^2$-labeling scheme (resp. $\log$-labeling scheme) on the class of graphs of clique-width at most k for the counting query of φ (resp. counting query of φ modulo s).*

We prove a similar theorem for nicely locally cwd-decomposable classes of graphs and FO formulas.

Theorem 8.4 *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be an FO formula and $s \in \mathbb{N}$. There exists a $\log^2$-labeling scheme (resp. $\log$-labeling scheme) for the counting query of φ (resp. counting query of φ modulo s) on nicely locally cwd-decomposable classes.*

We will first prove Theorem 8.4 for a particular case of t-local formulas on locally cwd-decomposable classes.

Definition 8.6 (t-Connected Formulas) A formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ is t-*connected* if for all G, all $a_1, \ldots, a_m \in V_G$ and all $W_1, \ldots, W_q \subseteq V_{G'}$,

$$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \quad \text{iff} \quad \begin{cases} \bigwedge_{1 \leq i < j \leq m} d(a_i, a_j) \leq t \text{ and} \\ G[N] \models \varphi(a_1, \ldots, a_m, W_1 \cap N, \ldots, W_q \cap N) \end{cases}$$

where $N = N_G^t(\{a_1, \ldots, a_m\})$.

Remark 8.1 Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a t-connected formula. Then for all $W \supseteq N_G^t(a_1, \ldots, a_m)$,

$$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \quad \text{iff} \quad G[W] \models \varphi(a_1, \ldots, a_m, W_1 \cap W, \ldots, W_q \cap W)$$

and because $N_G^t(\{a_1, \ldots, a_m\}) \subseteq N_G^{2t}(a_1)$

$$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \quad \text{iff} \quad G[N_G^{2t}(a_1)] \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q).$$

Lemma 8.5 *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a t-connected formula and let s be a positive integer. Then there exists a $\log^2$-labeling scheme (resp. $\log$-labeling scheme) for the counting query of φ (resp. counting query of φ modulo s) on locally cwd-decomposable classes of graphs.*

Proof. Let $\mathcal{T}$ be a $(2t, \ell, g)$-cwd cover of a locally cwd-decomposable n-vertex graph G. Let γ be a distance-2 coloring of $G(\mathcal{T})$ with $[\ell^2 + 1]$ colors.

Claim 8.3 *Let $x \in K_G^{2t}(U)$ and $y \in U'$ with $\gamma(U) = \gamma(U')$, $U \neq U'$. Then $d_G(x, y) > 2t$.*

Proof of Claim 8.3. If this is not the case, then $y \in U$ and x_U and $x_{U'}$ are adjacent in H, a contradiction since they have the same color. $\blacksquare$

For each x, we color it by i, the smallest $\gamma(U)$ such that $x \in K_G^{2t}(U)$. We can then consider G as a structure $\langle V_G, E_G, c_{1G}, \ldots, c_{lG} \rangle$ where $l = [\ell^2 + 1]$ and $c_{iG}(x)$ holds if and only if x has color i. For each $i \in [\ell^2 + 1]$, let φ_i be the formula logically equivalent to

$$\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q) \ \wedge \ c_i(x_1).$$

Then the following is clear.

Claim 8.4 $\#_G\varphi(Y_1, \ldots, Y_q) = \sum_i \#_G\varphi_i(Y_1, \ldots, Y_q)$.

We can now show that the counting query of φ admits a $\log^2$-labeling scheme on each locally cwd-decomposable class of graphs. Before let us prove the following. Let $V_i = \bigcup_{\gamma(U)=i} \{U \mid U \in \mathcal{T}\}$.

Claim 8.5 $cwd(G[V_i]) \leq g(1)$.

Proof. V_i is clearly a disjoint union of sets $U \in \mathcal{T}$. From Definition 8.3, each $G[U]$ has clique-width at most $g(1)$. Therefore $cwd(G[V_i]) \leq g(1)$. ∎

Claim 8.6 $\#_G\varphi_i(Y_1, \ldots, Y_q) = \#_{G[V_i]}\varphi_i(Y_1, \ldots, Y_q)$.

Proof of Claim 8.6. If $\varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$ holds and a_1 has color i, then $a_1 \in K_G^{2t}(U)$ for some U, $\gamma(U) = i$. Hence, $a_2, \ldots, a_m \in N_G^{2t}(a_1)$ and $G[N_G^{2t}(a_1)] \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$, hence $G[V_i] \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$.

If $G[V_i] \models \varphi_i(a_1, \ldots, a_m, W_1, \ldots, W_q)$, then $c_{iG}(a_1)$ and $\bigwedge_{1 \leq l < s \leq m} d_{G[V_i]}(a_l, a_s) \leq t$. But $d_G(a_l, a_s) = d_{G[V_i]}(a_l, a_s) = d_{G[U]}(a_l, a_s)$ where $a_1 \in U$ and $\gamma(U) = i$. And since $N_G^t(\{a_1, \ldots, a_m\}) \subseteq V_i$, then $G \models \varphi_i(a_1, \ldots, a_m, W_1, \ldots, W_q)$. ∎

By Theorem 8.3 and Claims 8.5 and 8.6, there exists a $\log^2$-labeling J_i for the counting query of each φ_i. For each $x \in V_G$, we let $J(x) = (J_1(x), \ldots, J_{\ell+1}(x))$. It is clear that $|J(x)| \leq O(\log^2(n))$ and is a $\log^2$-labeling for the counting query of φ by Claim 8.4. By Theorem 8.3, labels of size $O(\log(n))$ is sufficient for the counting query of each φ_i modulo s. ∎

We now prove Theorem 8.4.

Proof of Theorem 8.4. Let $\varphi(\bar{x}, \overline{Y})$ be an *FO* formula with free variables in $\bar{x} = (x_1, \ldots, x_m)$ and in $\overline{Y} = (Y_1, \ldots, Y_q)$. By Theorem 8.1, φ is logically equivalent to a Boolean combination of t-local formulas around $\bar{x}$ and basic (t', s)-local formulas. We have proved that each basic (t', s)-local formula admits a log-labeling scheme on nicely locally cwd-decomposable classes

of graphs in Theorem 8.2 (2). It remains then to prove that the counting query of a t-local formula admits a $\log^2$-labeling scheme on each nicely locally cwd-decomposable class of graphs.

Let $\psi(\bar{x}, Y_1, \ldots, Y_q)$ be a t-local formula around $\bar{x} = (x_1, \ldots, x_m)$. By Lemma 8.1 (see [Fri04, Sections 5,6] for technical details), we can reduce the counting query of ψ to the counting query of finitely many formulas of the form $\rho_{t,\varepsilon}(\bar{x}) \ \wedge \ \varphi'(\bar{x}, Y_1, \ldots, Y_q)$ that can be expressed as

$$\varphi'(\bar{x}, Y_1, \ldots, Y_q) = \bigwedge_{1 \le i < j \le p} d(\bar{x} \mid \varepsilon_i, \bar{x} \mid \varepsilon_j) > 2t + 1 \ \wedge \ \bigwedge_{1 \le i \le p} \varphi_i(\bar{x} \mid \varepsilon_i, Y_1, \ldots, Y_q)$$

where each φ_i is t-local and $(m \cdot (2t + 1))$-connected. We can then assume that ψ is of the form $\varphi'(\bar{x}, Y_1, \ldots, Y_q)$.

Let $\mathcal{T}$ be a nice (r, ℓ, g)-cwd cover where $r = m \cdot (2t+1)$ and let γ be a distance-$(m+1)$ coloring of the intersection graph of $\mathcal{T}$. For every m-tuple of colors $(i_1, \ldots, i_m)$, we let $G_{i_1, \ldots, i_m}$ be the graph $G[V]$ where V is the union of all sets $U \in \mathcal{T}$ such that $\gamma(U) \in \{i_1, \ldots, i_m\}$. We have then $cwd(G[V]) \le g(m)$ (same arguments as in Claim 8.1). We color each vertex with the smallest color i such that $x \in K_G^r(U)$ and $\gamma(U) = i$. Again we can consider G as a structure $\langle V_G, E_G, c_{1G}, \ldots, c_{lG} \rangle$ where $l = \ell^{O(m)}$ and $c_{iG}(x)$ holds if and only if x has color i. We let $\varphi'_{i_1, \ldots, i_p}$ be

$$\bigwedge_{1 \le i < j \le p} d(\bar{x} \mid \varepsilon_i, \bar{x} \mid \varepsilon_j) > 2t + 1 \ \wedge \ \bigwedge_{1 \le \ell \le p} (\varphi_\ell(\bar{x} \mid \varepsilon_\ell, Y_1, \ldots, Y_q) \ \wedge \ c_{i_\ell}(z_\ell))$$

where z_ℓ is the first variable of each tuple $\bar{x} \mid \varepsilon_\ell$. We have then:

Claim 8.7 $\#_G\psi(Y_1, \ldots, Y_q) = \sum\limits_{(i_1, \ldots, i_m)} \#_G\varphi'_{i_1, \ldots, i_m}(Y_1, \ldots, Y_m).$

We let $H = G_{i_1, \ldots, i_m}$. One can easily prove by using the same arguments as in the proof of Claim 8.2 that:

Claim 8.8 $d_G(\bar{x} \mid \varepsilon_i, \bar{x} \mid \varepsilon_j) > 2t + 1$ if and only if $d_H(\bar{x} \mid \varepsilon_i, \bar{x} \mid \varepsilon_j) > 2t + 1$.

It follows again that:

Claim 8.9 $\#_G\varphi'_{i_1, \ldots, i_m}(Y_1, \ldots, Y_q) = \#_H\varphi'_{i_1, \ldots, i_m}(Y_1, \ldots, Y_q).$

By Theorem 8.3, and Claims 8.7, 8.8 and 8.9, there exist a $\log^2$-labeling scheme for the counting query of each t-local formula. And a log-labeling scheme is enough for modulo counting. This finishes the proof. ∎

8.5   Other Results and Concluding Remarks

We have proved that every *FO*-definable property admits a log-labeling scheme on nicely locally cwd-decomposable classes of graphs. If we restrict the *FO*-definable properties to *FO*-definable properties without set arguments, we have a log-labeling scheme on locally cwd-decomposable classes of graphs. However, we do not know if these two classes of graphs are different since several important classes of graphs we know are nicely locally cwd-decomposable. A future work is to better understand the graphs that are (nicely) locally cwd-decomposable. It is also not known if every class of graphs of bounded local clique-width is locally cwd-decomposable. We also notice that in the definition of (nicely) locally cwd-decomposable, the "polynomial-time" appears for constructing the (nice) (r, ℓ, g)-cwd covers, hence for constructing the labels. However, this aspect has no influence on the sizes of the labels nor in the time taken for answering the queries. For completeness sake, we review here how to get similar results to the ones in Section 8.4 for other classes of graphs.

A graph has *arboricity at most* k if it is the union of k edge-disjoint forests (independently of the orientations of its edges). We prove the following.

Fact 8.2 *Let k be a positive integer and let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a quantifier-free FO formula. Then there exists a* log-*labeling scheme on the classes of graphs of arboricity at most k.*

Proof. Let G be a forest with edges anyway directed. Let us choose a root r and let $f^+, f^- : V_G \to V_G$ be mappings such that:

$f^+(u) = v$ iff $u \to v$ in G and v is on the unique undirected path between u and r.

$f^-(u) = v$ iff $u \leftarrow v$ in G and v is on the unique undirected path between u and r.

The edge relation in G is defined by:

$$E_G(u, v) \iff v = f^+(u) \ \vee \ u = f^-(v). \tag{4}$$

Let G be now a graph of arboricity k and represented by the structure $\langle V_G, E_G, P_{1G}, \ldots, P_{mG} \rangle$ where P_{iG} is an unary relation. Then G is the union of k edge-disjoint forests $F_1, \ldots, F_k$. We can, for each forest F_i, construct two mappings (f_i^+, f_i^-) such that $E_{G[F_i]}(u, v)$ holds if and only if $v = f_i^+(u) \ \vee \ u = f_i^-(v)$. We can then define the edge relation of G in a similar way as in (4) with $2k$ unary functions by letting:

$$E_G(u, v) \iff \bigvee_{i \in [k]} v = f_i^+(u) \ \vee \ u = f_i^-(v). \tag{5}$$

For each vertex x of G, we let $b(x)$ be the bit sequence of size m and such that the *i-th* bit is 1 if and only if $P_{iG}(x)$ holds. If vertices are numbered from 1 to n and $\ulcorner x \urcorner$ is the bit representation of the index of x, then we let

$$J(x) = \left(\ulcorner x \urcorner, \ulcorner f_1^+(x) \urcorner, \ulcorner f_1^-(x) \urcorner, \ldots, \ulcorner f_k^+(x) \urcorner, \ulcorner f_k^-(x) \urcorner, b(x) \right).$$

It is clear that given $J(a_1), \ldots, J(a_m)$ and $J(W_1), \ldots, J(W_q)$, we can test if G satisfies φ since by looking at the labels we can verify the adjacency of two vertices with the help of Equation (5), the equality of two vertices and then the membership of a vertex into a set, and for each vertex x, we can verify if $P_{iG}(x)$ holds by looking at the b part of its label.   ■

An example of a quantifier-free formula is $E(x, y)$ that says that x and y are adjacent. Then the adjacency has a log-labeling scheme on classes of graphs of bounded arboricity. Can we extend the results of Sections 8.4 to classes of graphs of bounded arboricity? Here is a proposition that limits the extension.

Let $\varphi_0(x, y)$ be the t-local formula that expresses that x and y are connected by a path of length at most 2:

$$\exists z (z \neq x \ \wedge \ z \neq y \ \wedge \ E(x, z) \ \wedge \ E(z, y)). \tag{6}$$

We prove the following.

Proposition 8.1 *Let $\mathcal{C}$ be the class of graphs of arboricity at most 2. Every labeling scheme $(\mathcal{A}, \mathcal{B})$ for φ_0 on $\mathcal{C}$ must use labels of size at least $\left\lfloor \sqrt{\frac{n}{2}} \right\rfloor$ for certain graphs with n vertices.*

Proof. With every (simple and undirected) graph G, we associate a graph $\widetilde{G}$ as follows:

$$V_{\widetilde{G}} = V_G \cup \{z_{x,y} \mid x, y \in V_G \text{ and } xy \in E_G\},$$
$$E_{\widetilde{G}} = \{xz_{x,y} \mid xy \in E_G\}.$$

In other words, $\widetilde{G}$ is obtained from G by subdividing each edge xy with a new vertex denoted by $z_{x,y}$. The following properties hold for $\widetilde{G}$:

(1) $V_G \subseteq V_{\widetilde{G}}$ and $|V_{\widetilde{G}}| \leq |V_G| + |E_G|$.

(2) For all $x, y \in V_G$, $xy \in E_G$ if and only if $\varphi_0(x, y)$ is true in $\widetilde{G}$.

(3) $\widetilde{G}$ is of arboricity at most 2.

The first two points are straightforward from the construction of $\widetilde{G}$. We orient each edge e of G and we get a directed graph, that we denote by $\vec{G}$. We let:

$$F_1 = \{xz_{x,y} \mid (x, y) \in E_{\vec{G}}\},$$
$$F_2 = \{z_{x,y}y \mid (x, y) \in E_{\vec{G}}\}.$$

Clearly, neither F_1 nor F_2 has a cycle in $\widetilde{G}$. Then $\widetilde{G}$ has arboricity at most 2 since (F_1, F_2) is clearly a bipartition of $E_{\widetilde{G}}$.

By using a simple counting argument, one can show that every labeling scheme for adjacency queries in simple and undirected graphs with n vertices requires some labels of size at least $\frac{1}{n} \log_2 \left(2^{\binom{n}{2}} \right) = (n-1)/2$ bits. Hence, adjacency requires labels of size $\lfloor n/2 \rfloor$ in all graphs. Using (2) above, we conclude that any labeling scheme for φ_0 on the graph family $\mathcal{F}_n = \{\widetilde{G} \mid G \text{ has } n \text{ vertices}\}$ requires labels of size at least $\left\lfloor \frac{n}{2} \right\rfloor$. Let $\widetilde{G}$ be in $\mathcal{F}_n$ and let $\widetilde{n} = |V_{\widetilde{G}}|$. Using (1), we have $\widetilde{n} \leq n + |E_G| \leq \frac{n(n+1)}{2}$, i.e., $n \geq \sqrt{2\widetilde{n}}$. Hence, any labeling scheme for φ_0 on $\mathcal{F}_n$ requires labels of size at least $\left\lfloor \frac{\sqrt{2\widetilde{n}}}{2} \right\rfloor = \left\lfloor \sqrt{\frac{\widetilde{n}}{2}} \right\rfloor$. ∎

Can we however find other graph classes of bounded arboricity, that are not locally cwd-decomposable, where we can extend the results of Section 8.4 even with a less powerful logic? We give here an example of such a class where some properties, that are definable by some particular *FO* formulas, admit a log-labeling scheme.

Nešetřil and Ossona de Mendez in their works on graph isomorphisms and on minor-closed classes of graphs, defined the class of graphs of *bounded expansion* [NdM06b, NdM06a, NdM08a]. These classes of graphs can be seen as a generalization of graphs of bounded degree and minor-closed classes of graphs. There are several equivalent definitions, we will however use the following. (Figure 24 shows inclusion relations between the many classes defined or cited in this chapter.)

Definition 8.7 (Bounded Expansion) A class $\mathcal{C}$ of graphs has *bounded expansion* if for every integer p, there exists a constant $N(\mathcal{C}, p)$ such that for every $G \in \mathcal{C}$, one can partition V_G in at most $N(\mathcal{C}, p)$ parts such that any $i \leq p$ of them induce a sub-graph of tree-width at most $i - 1$.

For $i = 1$, Definition 8.7 implies that each part is a stable set, hence the partition can be seen as a *proper vertex-coloring*. The notion of bounded expansion is new and appears in the series of papers by Nešetřil and Ossona de Mendez [NdM06b, NdM06a, NdM08a, NdM08b, NdM08c] where they unify many theorems on minor-closed classes of graphs and bounded degree graph classes. We now recall a result by Nešetřil and Ossona de Mendez [NdM06a] on properties that admit linear-time algorithms on classes of graphs of bounded expansion. Before let us define these properties.

Definition 8.8 (Bounded Formulas) An *FO* formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ is *basic bounded* if, for some $p \in \mathbb{N}$, we have the following equivalence for all graphs G, all $a_1, \ldots, a_m \in V_G$ and all $W_1, \ldots, W_q \subseteq V_G$,

$$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \text{ iff } G[X] \models \varphi(a_1, \ldots, a_m, W_1 \cap X, \ldots, W_q \cap X)$$

for some $X \subseteq V_G$ such that $|X| \leq p$ and $a_1, \ldots, a_m \in X$. (If this is true for X, then $G[Y] \models \varphi(a_1, \ldots, a_m, W_1 \cap Y, \ldots, W_q \cap Y)$ for every $Y \supseteq X$.)

An *FO* formula is *bounded* if it is a Boolean combination of basic bounded formulas.

Note that the negation of a basic bounded formula is not necessarily basic bounded, however it is bounded. For instance, the formula $d(x, y) \leq r$ is basic bounded for $p = r + 1$. Its negation is not basic bounded. Note also that the formula φ_0 in Equation (6) is also bounded.

Nešetřil and Ossona de Mendez proved the following.

Theorem 8.5 ([NdM06a]) *Let $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a graph property definable by a bounded FO formula and let $\mathcal{C}$ be a class of graphs of bounded expansion. Then there exists a linear-time algorithm that given a graph $G \in \mathcal{C}$ and P, decides whether G satisfies P or not.*

We prove the following.

Proposition 8.2 *Every graph property definable by a bounded FO formula admits a log-labeling scheme on classes of graphs of bounded expansion.*

Proof. Let G be in a class of graphs of bounded expansion with $n = |V_G|$ and let φ be a basic bounded formula with bound p. We let $N = N(\mathcal{C}, p)$. By Definition 8.7, we can partition V_G into $V_1 \uplus V_2 \uplus \cdots \uplus V_N$, $V_i \neq \emptyset$ such that for every $i \leq p$, $twd(G[V_1 \cup \cdots V_i]) \leq i - 1$.

For every $\alpha \subseteq [N]$ of size p, we let $V_\alpha = \bigcup_{i \in \alpha} V_i$; then the tree-width of $G[V_\alpha]$ is at most $p - 1$. Each vertex u belongs to less than $(N - 1)^{p-1}$ sets V_α.

Hence, a basic bounded formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ is true in G if and only if it is true in some $G[X]$ with $|X| \leq p$, hence in some $G[V_\alpha]$ such that $x_1, \ldots, x_m \in V_\alpha$. For each α, we construct a log-labeling J_α of φ on $G[V_\alpha]$ with Theorem 7.4 (recall that $twd(G[V_\alpha]) \leq p - 1$). We also determine for each α, the truth value of $\varphi(\emptyset, \ldots, \emptyset)$ in $G[V_\alpha]$. For each $x \in V_G$, we let

$$J(x) = \Big(\big((\ulcorner \alpha \urcorner, J_\alpha(x), t_\alpha) \mid x \in V_\alpha \big), \big((\ulcorner \alpha \urcorner, t_\alpha) \mid x \notin V_\alpha \big) \Big).$$

We have clearly $|J(x)| = O(\log(n))$ since the family of pairs (α, t_α) is of fixed size (depends on p). We now explain how to decide φ by using the labels only. Assume first that φ has at least one free FO variable. Given $J(a_1), \ldots, J(a_m)$, we can determine all those sets α such that V_α contains $a_1, \ldots, a_m$. Using the components $J_\alpha(\cdot)$ of $J(a_1), \ldots, J(a_m)$ and the labels in $J(W_1), \ldots, J(W_q)$, we can determine if, for some α, $G[V_\alpha] \models \varphi(a_1, \ldots, a_m, W_1 \cap V_\alpha, \ldots, W_q \cap V_\alpha)$, hence whether $G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$.

It remains to consider the case of a basic bounded formula without free FO variable, i.e., is of the form $\varphi(Y_1, \ldots, Y_q)$. From $J(W_1), \ldots, J(W_q)$, we get $D = \{\alpha \mid V_\alpha \cap (W_1 \cup \cdots \cup W_q) \neq \emptyset\}$. By using the $J_\alpha(\cdot)$ components of the labels in $J(W_1) \cup \cdots \cup J(W_q)$, we can determine if, for some $\alpha \in D$, we have $G[V_\alpha] \models \varphi(W_1 \cap V_\alpha, \ldots, W_q \cap V_\alpha)$. If one is found we, conclude positively. Otherwise, we look for some $t_\beta = \text{TRUE}$ where $\beta \notin D$. This gives the final answer.

Now assume that φ is a bounded FO formula, i.e., is a a Boolean combination of basic bounded formulas $\varphi_1, \ldots, \varphi_t$. For each basic bounded formula φ_i, we apply the procedure above to get a log-labeling J_i. For each x, we let $J(x) = (J_1(x), \ldots, J_t(x))$. It is of size $O(\log(n))$ and gives the desired result. ∎

Let H be a fixed undirected graph with p vertices. The following two FO formulas are basic bounded:

- $\varphi(x_1, \ldots, x_p) = $ "the sub-graph induced on $\{x_1, \ldots, x_p\}$ is isomorphic to H".

- $\varphi(x_1, \ldots, x_p) = $ "the sub-graph induced on $\{x_1, \ldots, x_p\}$ has a sub-graph isomorphic to H".

We finish this chapter with some open questions. Grohe et al. proved that every FO-definable property admits a polynomial-time algorithm on classes of graphs that exclude a minor [FG01a] and classes of graphs that *locally exclude a minor* [DGK07]. A class $\mathcal{C}$ of graphs *locally excludes a minor* if, for every $r \in \mathbb{N}$, there exists a graph H_r such that for every $G \in \mathcal{C}$ and every $a \in V_G$, the graph H_r is not a minor of $G[N_G^r(a)]$. We conjecture the following.

Conjecture 8.1 *Every FO-definable property admits a* log-*labeling scheme on classes of graphs that exclude a minor and classes of graphs that locally exclude a minor.*
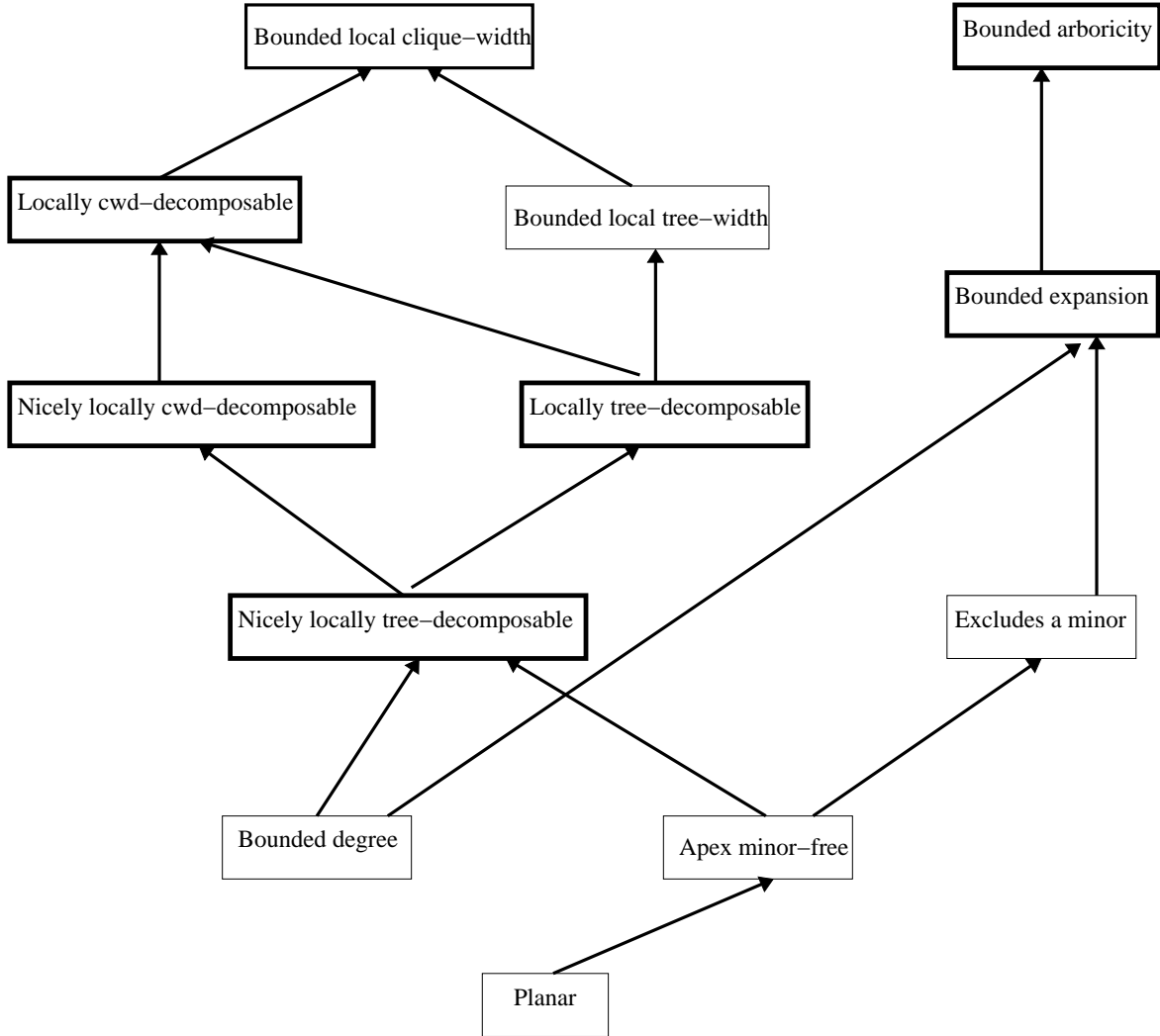

Figure 24: An arrow means an inclusion of graph classes. Bold boxes are classes studied in this chapter.

Chapter 9

Short Labeling for Connectivity Queries on planar graphs

We construct a log-labeling scheme for a specific (but important) MS-definable query on planar graphs, a class of graphs that has unbounded clique-width, but bounded local clique-width. Hence, Theorem 7.4 does not apply. However, it will be useful as a tool in our constructions.

In Section 9.1 we define some notations and explain the idea of our labeling scheme. In Section 9.2 we introduce the notion of embeddings of planar graphs and prove some basic lemmas. We introduce in Section 9.3 some tools that allow to represent properties by unary functions. We prove our main theorem for the case of 3-connected planar graphs in Section 9.4. We extend the proof to 2-connected planar graphs in Section 9.5. We finally give a proof for connected planar graphs in Section 9.6. We conclude by some remarks in Section 9.7.

9.1 Preliminaries

The graphs in this chapter are undirected, unless otherwise specified. We will sometimes write $x - y$ (resp. $x \to y$) to denote an undirected edge between x and y (resp. an arc from x to y). We denote by $E(x)$ the set of edges or arcs incident with x. See Chapter 1 for other notations or terminologies. A partial order $\leq_F$ on the nodes of a rooted forest F is defined as follows:

$$x \leq_F y \quad \text{if and only if} \quad \text{every path from a root to } x \text{ goes through } y.$$

Hence, the roots are the maximal elements.

A vertex x of a graph G is a *separating vertex* if the sub-graph $G \backslash x$ has more connected components than G. A connected graph is 2-*connected* if it has no separating vertex. A *biconnected component* of a connected graph G is a maximal 2-connected sub-graph of G (maximal with respect to inclusion). We denote by $Bcc(G)$ the set of biconnected components of G. Two vertices x and y of a graph G are *separated* by $X \subseteq V_G$ if they are in different connected components of $G \backslash X$.

A *circular sequence over a set E* is a nonempty sequence $s = (e_1, \ldots, e_n)$ of pairwise distinct elements of E. The term *circular* refers to equality: we define $(e_1, \ldots, e_n)$ and

$(e_i, \ldots, e_n, e_1, \ldots, e_{i-1})$ as equal circular sequences. If $s_1 = (e_1, \ldots, e_p)$ and $s_2 = (f_1, \ldots, f_q)$ are sequences of pairwise distinct elements of a set E, we will denote by $s_1 \bullet s_2$ the concatenation of s_1 and s_2 and by $s_1 \circ s_2$, the circular sequence, one representation of which is $s_1 \bullet s_2 = (e_1, \ldots, e_p, f_1, \ldots, f_q)$.

Definition 9.1 (Connectivity Query) Given a graph G, two vertices x and y in V_G, a set of vertices $X \subseteq V_G - \{x, y\}$ and a set of edges $F \subseteq E_G$, we let $Conn(x, y, X, F)$, called *connectivity query*, denote the graph property that expresses that x and y are connected by a path that avoids vertices in X and edges in F, i.e., x and y are connected by a path in $(G - F) \backslash X$. We write it $Conn(x, y, X)$ if $F = \emptyset$. We call the pair (X, F) the *data* of the query; its *size* is defined as $|X| + |F|$.

We now state the main theorem of this chapter.

Theorem 9.1 (Main Theorem of this Chapter) *There exists a* log*-labeling scheme* $(\mathcal{A}, \mathcal{B})$ *for the connectivity query on the class of simple undirected planar graphs. If n is the number of vertices of the input graph, $\mathcal{A}$ computes the labels in $O(n \cdot \log(n))$-time and $\mathcal{B}$ gives the answer in $O(m^2)$-time where m is the size of the data.*

We now sketch the main ideas of the proof. The principal idea is to use geometrical tools and the decomposition of connected graphs into biconnected components.

If G is a plane graph (see Definition 9.2), we denote by G^+ the planar graph obtained by adding one vertex, called *face-vertex*, in the middle of each face and edges between the face-vertex and the vertices of G incident with that face. If G is 2-connected, the graph G^+ is simple and can be embedded in the plane with integer coordinates and edges represented by straight-line segments by using Schnyder's algorithm [Sch90]; we fix such an embedding $\mathcal{E}$. For $X \subseteq V_G$, we define its *barrier*, denoted by $Bar(X)$, as a set of straight-line segments representing some edges of G^+ such that x and y in $V_G - X$ are separated by X in G if and only if they are separated in $\mathbb{R}^2$ by $Bar(X)$ (see Proposition 9.1). If from labels attached to the vertices of X we can deduce the set of straight-line segments forming $Bar(X)$, and if we also know the coordinates of x and y, then we can test whether x and y are separated in $\mathbb{R}^2$ by $Bar(X)$ in time $O(m^2)$ where m is the number of segments forming $Bar(X)$ by using geometric tools (see Definition 9.4 and Theorem 9.3).

To be able to construct $Bar(X)$ from the labels of vertices in X, we attach to each vertex z of G, not only its own pair of coordinates in the fixed embedding, but also those of a bounded number of adjacent vertices in G^+ (see Sections 9.2 and 9.3). However, this proof only works for 3-connected planar graphs, or rather for planar graphs such that every two vertices are incident with a bounded number of faces (see Section 9.4). Therefore, an additional treatment is needed for biconnected components. We will use for that the decomposition of 2-connected graphs into 3-connected components; then try to recognize simple cases where x and y are separated by one or two vertices of the given set X such that if these cases do not apply, we are reduced to connectivity queries in the plane. Happily, those simple cases are MS-definable and since the decomposition is a tree, we can use Theorem 7.4 to recognize them (see Section 9.5). Finally, we use the decomposition of connected graphs into biconnected components and again use Theorem 7.4 to recognize simple cases that, when they do not occur, reduce to some connectivity queries in the plane (see Section 9.6).

We will give the proofs for $F = \emptyset$. We will consider the case where $F \neq \emptyset$ in the proof of Theorem 9.1 (see the end of Section 9.6). Therefore from now on, by connectivity query we mean connectivity query of the form $Conn(x, y, X)$.

9.2   Plane Graphs

We recall here simple definitions and basic facts on plane graphs. See the books by Diestel [Die05] and by Mohar and Thomassen [MT01] for more detailed definitions, particularly regarding topological definitions.

Definition 9.2 (Embeddings in the Plane) A *planar embedding* of a graph $G = \langle V_G, E_G \rangle$ is a pair of mappings $\mathcal{E} = (p, s)$ such that:

(P1) the mapping $p : V_G \to \mathbb{R}^2$ associates with each vertex $x \in V_G$, the point $p(x)$ representing it in the plane,

(P2) the mapping $s : E_G \to 2^{\mathbb{R}^2}$ associates with every edge $e = xy$ a closed curve segment with ends $p(x)$ and $p(y)$, such that for every two distinct edges e and f in E_G, we have $z \in s(e) \cap s(f)$ if and only if $z = p(x)$ and x is incident with e and with f.

A planar embedding is a *straight-line embedding* if each curve $s(e)$ is a straight-line segment. A *plane graph* is the equivalence class of a planar embedding of a graph with respect to homeomorphism. We will write a plane graph G as a triple $\langle V_G, E_G, F_G \rangle$ where F_G is the set of faces. If $Y \subseteq E_G$, we let $\mathcal{E}(Y)$ be the union of curve segments $s(e)$ for $e \in Y$.

If G is a plane graph, for each $x \in V_G$, we let $E^0(x)$ be the circular sequence of edges incident with x for the anti-clockwise orientation of the plane; (e', x, e) is a *corner at* x if e' follows e in $E^0(x)$.

We can thus consider a plane graph G as a combinatorial object which consists of a graph $\langle V_G, E_G \rangle$, the circular sequence $E^0(x)$ for each $x \in V_G$ and of a corner belonging to the external face. We only consider embeddings of graphs in the plane, not in the sphere; for this reason we distinguish the external face with help of some corner. Notice that several plane graphs may have the same underlying planar graph G even if G is 3-connected. See [MT01] for more details about embeddings of graphs in the plane.

Let $\mathcal{E} = (p, s)$ be a planar embedding of a plane graph G and let C be a cycle in G. We say that a vertex $x \in V_G$, not belonging to C, is *inside* C if $p(x)$ is in the bounded component of $\mathbb{R}^2 - \mathcal{E}(C)$. We say that two distinct vertices x and y in V_G, not belonging to $Y \subseteq E_G$, are *separated by* $\mathcal{E}(Y)$ if they are in different connected components of $\mathbb{R}^2 - \mathcal{E}(Y)$. These properties do not depend on $\mathcal{E}$; they will be used for plane graphs, independently of embeddings.

We now define the planar graph G^+ associated with a plane graph G.

Definition 9.3 (Augmented Planar Graph) Let G be a connected plane graph. We associate with it a connected planar graph $G^+ = \langle V_G \cup F_G, E_G \cup E' \rangle$ where:

$$E' = \{fx \mid x \in V_G, \ f \in F_G \text{ and there exist } e, e' \in E_G \text{ such that } (e, x, e') \text{ is a corner of } f\}.$$

The graph G^+ is called the *augmented planar graph of* G and a vertex of G^+ in F_G is called a *face-vertex*.

Notice that if $G^+ = \langle V_G \cup F_G, E_G \cup E' \rangle$ we have an edge fx in E' for each corner (e, x, e') of f. Therefore, if x is a separating vertex, we may have several edges between x and f because a face f may have several corners at x.

For a simple planar graph G with n vertices, the maximum number of faces $m = 2n - 4$ is obtained when G is triangulated. Hence, G^+ has at most $3n - 4$ vertices. Every embedding $\mathcal{E} = (p, s)$ of G can be extended into an embedding $\mathcal{E}^+$ of G^+ in the following obvious way: for each $f \in F_G$ we define $p(f)$ as any point in the open subset of $\mathbb{R}^2$ corresponding to the face f and we draw curve segments between this point and the points, representing vertices in V_G, adjacent to f (see Figure 25 for an example). In general, several non homeomorphic embeddings $\mathcal{E}^+$ can be associated with $\mathcal{E}$ because the edges incident with the external face of G can be drawn in different ways, even if G is 3-connected. Hence G^+ is a planar graph (and not a plane graph) associated with a plane graph G.

Example 9.1 Figure 25 shows a plane graph G with vertices t, x, w, u, y, z, v represented by black dots and continuous edges. It also shows the graph G^+. Its face-vertices are represented by small circles. The one marked A represents the external face. There are three parallel edges between A and x, because x is the separating vertex of 3 biconnected components (there are 3 corners around x).
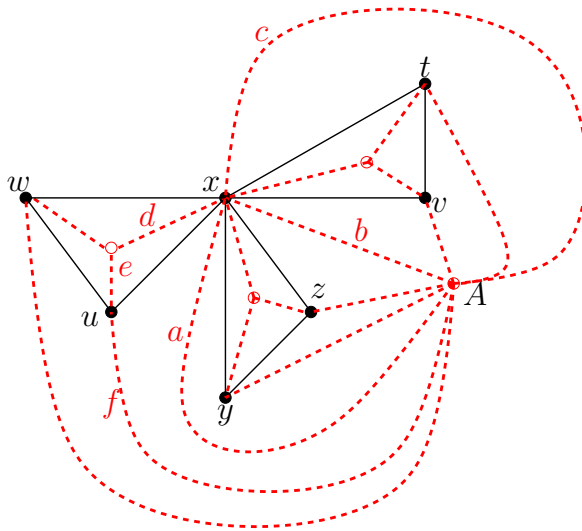
Figure 25: An augmented graph G^+.

The following lemma is straightforward to establish.

Lemma 9.1 *If G is a connected plane graph, then the planar graph G^+ is triangulated. It is simple if and only if G is 2-connected.*

We now define the notion of a *barrier* of a set of vertices. This notion will allow to transform the connectivity query to a geometrical one.

Definition 9.4 (The Barrier of a Set of Vertices) Let $G = \langle V_G, E_G, F_G \rangle$ be a plane graph and let $G^+ = \langle V_G \cup F_G, E_G \cup E' \rangle$ be its augmented graph (E' is the set of edges linking

a face-vertex to vertices of G). For $X \subseteq V_G$ we let $Bar(X)$, the *barrier* of X, be the set:

$$Bar(X) = \{fx \in E' \mid f \in F_G, x \in V_G \text{ and there exists } y \in X \text{ with } fy \in E'\}.$$

If $\mathcal{E}^+$ is a planar embedding of G^+, we let $Bar(X, \mathcal{E}^+)$ be the set $\mathcal{E}^+(Bar(X))$.

If a face f has several corners at a vertex $x \in X$, then all the edges of G^+ between f and x are in $Bar(X)$. This can happen if and only if x is a separating vertex. A vertex of X may not be the end of any edge of $Bar(X)$ (see Example 9.2). Note that $Bar(X, \mathcal{E}^+)$ is a closed (and even compact) subset of $\mathbb{R}^2$. See Examples 9.2 and 9.3 that illustrate cases of vertices separated by $Bar(X, \mathcal{E}^+)$.

Example 9.2 We use the graph G on Figure 25. Then $Bar(\{x\}) = \{a, b, c\}$. It separates u and w from y and z and from t and v. The barrier $Bar(\{y\})$ is empty. We have $Bar(\{u, x\}) = \{a, b, c, d, e, f\}$.

Example 9.3 Figure 26 shows the augmented graph H^+ of a plane graph H. It is simple since H is 2-connected. So, we can draw it with straight-lines segments. The barrier of $\{x, y\}$ consists of 6 (thick) dotted edges and separates u from v and w.
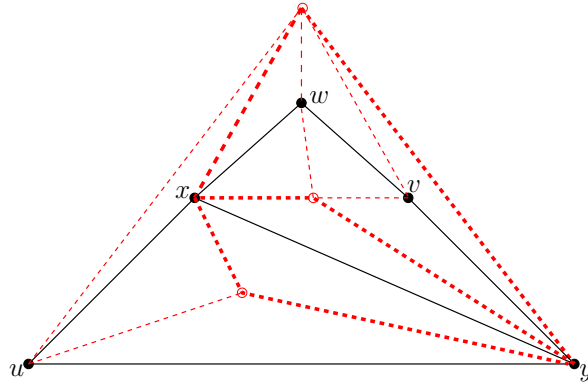
Figure 26: An augmented graph H^+.

Proposition 9.1 *Let G be a connected plane graph and let $\mathcal{E}^+ = (p, s)$ be a planar embedding of G^+. For every $X \subseteq V_G$ and $x, y \in V_G - X$, the vertices x and y are separated by X if and only if $p(x)$ and $p(y)$ are separated by $Bar(X, \mathcal{E}^+)$.*

Proof. Assume x and y connected by a path in $G \backslash X$. They are still connected by this path in G^+ and this path has no vertex in any edge of $Bar(X)$. Hence, $p(x)$ and $p(y)$ are in the same connected component of $\mathbb{R}^2 - Bar(X, \mathcal{E}^+)$.

For the converse direction, let us assume that x and y are not connected in $G \backslash X$. Since $Y \subseteq X$ implies $Bar(Y) \subseteq Bar(X)$, it is enough to prove the result for a minimal separator X of x and y. Let X be so; the set E_G can be partitioned into $E_G = E_x \cup E_y$ such that:

(a) $x \in V_{G[E_x]}$, $y \in V_{G[E_y]}$ and $V_{G[E_x]} \cap V_{G[E_y]} = X$;

(b) $G[E_x]$ and $G[E_y]$ are connected;

(c) The circular sequence of edges incident with each $z \in X$ can be written $E^\circ(z) = E_1(z) \circ E_2(z)$ where $E_1(z)$ is a sequence enumerating the set of edges $E_x \cap E(z)$ and $E_2(z)$ is similar for the set $E_y \cap E(z)$.

We modify G by replacing each vertex $z \in X$ by an edge $e_z = zz'$ and such that z is adjacent to $E_1(z)$ and z' to $E_2(z)$. Precisely, we make G into a plane graph G' with vertex set $V_{G'} = V_G \cup \{z' \mid z \in X\}$, edge set $E_{G'} = \{e_z \mid z \in X\} \cup \{wz \mid w \in V_G - X, wz \in E_1(z)\} \cup \{wz' \mid w \in V_G - X, wz \in E_2(z)\} \cup \{wz \mid w, z \in V_G - X, wz \in E_G\}$ and with circular sequences $E'^0(z)$ for each $z \in V_{G'}$ such that:

$$\begin{cases} E'^\circ(z) = E_1(z) \circ (e_z), \\ E'^\circ(z') = E_2(z) \circ (e_z) \end{cases} \qquad \text{for every } z \in X,$$

and

$$E'^\circ(z) = E^\circ(z) \qquad\qquad\qquad \text{if } z \in V_G - X.$$

It is clear that G' is a plane graph and that $E(X)$ defined as $\{e_z \mid z \in X\}$ is a minimal edge-cut of G'. Hence, $E(X)$ is a cycle in the dual plane graph G'^* (see Diestel [Die05, Proposition 4.6.1]) that furthermore separates x and y. This cycle can be written as a circular sequence of edges $(e_1, \ldots, e_p)$ for some enumeration $z_1, \ldots, z_p$ of X. (Notice that if $X = \{x_1\}$, this cycle consists of two parallel edges.) Let $f_1, \ldots, f_p$ be the faces of G' such that in G'^* we define the edge e_i as $f_i f_{i+1}$ for $1 \le i < p$, and the edge e_p as $f_p f_1$.

We denote by $\overline{f_1}, \ldots, \overline{f_p}$ the faces of G, resulting respectively from $f_1, \ldots, f_p$ by the contraction of edges e_z for all $z \in X$. It is clear that $\overline{f_i}$ is adjacent in G^+ to z_i and z_{i+1} for $i = 1, \ldots, p-1$ and that $\overline{f_p}$ is adjacent to z_p and z_1. In any embedding $\mathcal{E}^+$ of G^+, the cycle formed by the circular sequence of vertices $(z_1, \overline{f_1}, z_2, \overline{f_2}, z_3, \ldots, z_p, \overline{f_p})$ separates x and y. ∎

We illustrate the proof of Proposition 9.1 with Example 9.4.

Example 9.4 A plane graph G is shown on Figure 27. Its vertices x and y are separated by $X = \{u, v, z\}$. Figure 28 shows the result of replacing u, v, z by the edges e_u, e_v and e_z which are dotted. It also shows the edges of the cycle $E(X)$ in the dual graph G'^*. The contraction of the dotted edges gives the desired cycle in G^+ (see Figure 29).

Proposition 9.1 transforms the problem of connectivity query to a geometrical problem. We need now, given the set $Bar(X)$, be able to decide if $p(x)$ and $p(y)$ are in different connected components in $\mathbb{R}^2 - Bar(X, \mathcal{E}^+)$ where $\mathcal{E}^+ = (p, s)$ is an embedding of G^+. We will explain in Section 9.4 how to do that when the set $Bar(X, \mathcal{E}^+)$ is a set of straight-line segments which is sufficient for our purposes (see Sections 9.4, 9.5 and 9.6). We will now explain in the next section how to label each vertex of a planar graph in order to recover the set $Bar(X)$ from the labels of the vertices in X.
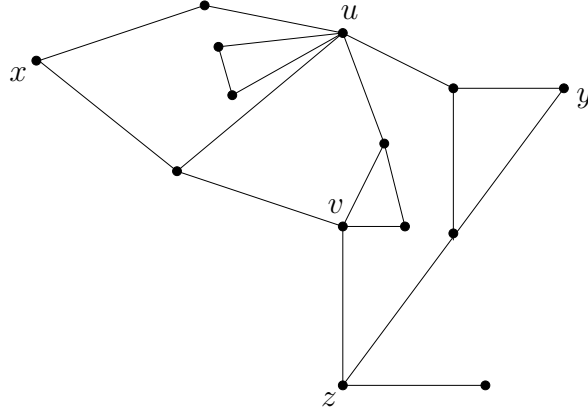
Figure 27: A plane graph G; $X = \{u, v, z\}$.
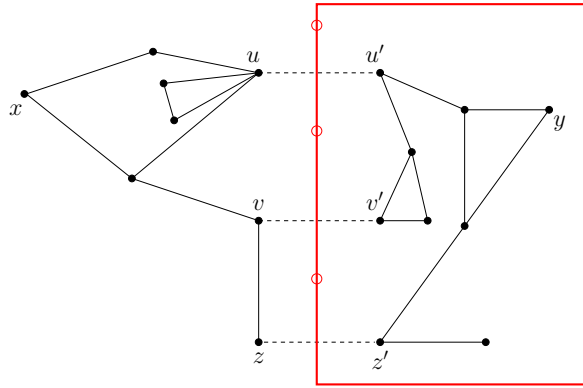
Figure 28: The plane graph G' and the cycle $E(X)$ in its dual.
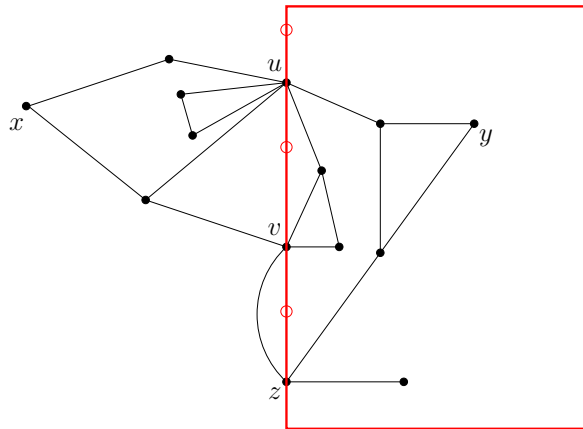
Figure 29: The plane graph obtained by contracting the edges e_z, $z \in X$.

9.3   Representation of Properties and Functions by Unary Functions

We introduce the new notion of the representation of a property with m arguments or an *m-ary* multivalued function, by a fixed number of unary functions. We will then use it in order

to label vertices of plane graphs so that we can recover $Bar(X)$ from the labels of vertices in X.

Definition 9.5 (Formulas over Unary Function Symbols) Let $\mathcal{F}$ be a finite set of unary function symbols and let $\mathcal{X}$ be a finite set of lower case variables. We denote by $\mathcal{B}(\mathcal{F},\mathcal{X})$ the set of quantifier-free formulas that are Boolean combinations of atomic formulas of the forms $x = y,\ x = f(y),\ f(x) = g(y),\ f = g$ for $x, y \in \mathcal{X}$ and $f, g \in \mathcal{F}$.

Notice that we do not allow formulas of the form $x = f(g(y))$, hence $\mathcal{B}(\mathcal{F},\mathcal{X})$ is not the set of all quantifier-free formulas over $\mathcal{F}$ and $\mathcal{X}$

Definition 9.6 (Representations by Unary Functions) Let $\mathcal{F}$ be a finite set of unary function symbols, let V be a finite set and let $\mathcal{X} = \{x_1, \ldots, x_m\}$. For each $f \in \mathcal{F}$, we let $f_V : V \to V$ be a total function and we denote by $\mathcal{F}_V$ the family $(f_V)_{f \in \mathcal{F}}$. We say that a relation $R \subseteq V^m$ is *represented by* $\mathcal{F}_V$ *and* $\varphi(x_1, \ldots, x_m) \in \mathcal{B}(\mathcal{F}, \mathcal{X})$ if:

$$R = \{(a_1, \ldots, a_m) \in V^m \mid \varphi(a_1, \ldots, a_m) \text{ holds}\}.$$

Let $\varphi(x_1, \ldots, x_m, y) \in \mathcal{B}(\mathcal{F}, \mathcal{X} \cup \{y\})$ be a disjunction of formulas of the form $\psi \wedge (y = f(x_i))$ or $\psi \wedge (y = x_i)$ with $\psi(x_1, \ldots, x_m) \in \mathcal{B}(\mathcal{F}, \mathcal{X})$. An *m-ary* multivalued function $g : V^m \to 2^V$ is *represented by* $\mathcal{F}_V$ *and* φ if for every $(a, a_1, \ldots, a_m) \in V^{m+1}$, the relation $a \in g(a_1, \ldots, a_m)$ is represented by $\mathcal{F}_V$ and φ.

Let $\mathcal{C}$ be a class of graphs where for each graph $G \in \mathcal{C}$, a set X_G is identified. Let R be an *m-ary* relation on $\mathcal{C}$ such that for every graph $G \in \mathcal{C}$, $R_G \subseteq X_G^m$ and let g be an *m-ary* multivalued function on $\mathcal{C}$ such that for every graph $G \in \mathcal{C}$, $g_G : X_G^m \to 2^{X_G^m}$. We say that R (resp. g) is *representable by k functions* if there exist a set of k unary total functions $\mathcal{F}$ and a formula $\varphi_R(x_1, \ldots, x_m) \in \mathcal{B}(\mathcal{F}, \mathcal{X})$ (resp. a formula $\varphi_g(x_1, \ldots, x_m, y) \in \mathcal{B}(\mathcal{F}, \mathcal{X} \cup \{y\})$) such that for every graph $G \in \mathcal{C}$, the relation R_G (resp. the *m-ary* multivalued function g_G) is represented by $\mathcal{F}_G$ and φ_R (resp. φ_g).

We can now use this notion for the labeling of connectivity query on planar graphs. In order to prove that a given relation on a class $\mathcal{C}$ of graphs is representable by k functions, we will always define for a fixed graph $G \in \mathcal{C}$, the functions and then give the appropriate formula. Note that if an *m-ary* multivalued function is represented by $\overline{\mathcal{F}}$ and $\varphi \in \mathcal{B}(\mathcal{F}, \{x_1, \ldots, x_m, y\})$ where $\mathcal{F}$ is a finite set of functions, then $|g(a_1, \ldots, a_m)| \leq m \cdot (|\mathcal{F}| + 1)$ for all $(a_1, \ldots, a_m) \in V^m$. Notice also that partial functions g for which $|g(a_1, \ldots, a_m)| \leq 1$ can be represented by $\overline{\mathcal{F}}$ and a formula $\varphi \in \mathcal{B}(\mathcal{F}, \{x_1, \ldots, x_m, y\})$ with help of Convention 9.1.

Convention 9.1 In all the constructions below, for every graph G, we will define partial functions $f_G : X_G \to X_G$ such that $f_G(x) \neq x$ for every x. We make them total by letting $f_G(x) = x$ instead of "$f_G(x)$ is undefined".

With this convention whenever $f(x)$ appears, we need to conjunct it with the atomic formula $f(x) \neq x$. However, for readability we will omit the atomic formulas $f(x) \neq x$.

Lemma 9.2 *The adjacency query in graphs of arboricity at most k is representable by k functions from vertices to vertices. The adjacency query and edge directions in directed graphs of arboricity at most k are representable by $2k$ functions from vertices to vertices.*

Proof. We need only consider simple graphs because we can replace a set of parallel edges by a single edge without changing adjacency. The edge set of every graph G of arboricity at most k can be partitioned into k sets $E_1, E_2, \ldots, E_k$ such that $G[E_i]$ is a forest for each $i \in [k]$, that we can assume to be rooted (we choose a root r_i and orient the edges of $G[E_i]$ appropriately). For each $i \in [k]$, we define the function $g_{iG} : V_G \to V_G$ as follows:

$$g_{iG}(x) = y \qquad \text{if and only if} \qquad y \text{ is the father of } x \text{ in } G[E_i].$$

We have clearly $g_{iG}(x) \neq x$ for every $x \in V_G$. We can therefore extend each function g_{iG} into a total one by Convention 9.1. Then x and y are adjacent if and only if the following formula holds:

$$\bigvee_{1 \leqslant i \leqslant k} (x = g_i(y) \ \vee \ y = g_i(x)).$$

For representing edge directions, we replace each function g_{iG} by two partial functions $g_{iG}^+ : V_G \to V_G$ and $g_{iG}^- : V_G \to V_G$ as follows:

$$g_{iG}^+(x) = y \qquad \text{if and only if} \qquad g_{iG}(x) = y \text{ and there is an arc } x \to y.$$
$$g_{iG}^-(x) = y \qquad \text{if and only if} \qquad g_{iG}(x) = y \text{ and there is an arc } y \to x.$$

Notice that we have $g_{iG}^+(x) = g_{iG}^-(x) = y$ if there is a pair of directed opposite edges between x and y. We use again Convention 9.1 to extend them into total functions. Again x and y are adjacent if and only if the following formula holds:

$$\bigvee_{1 \leqslant i \leqslant k} \left(x = g_i^+(y) \ \vee \ x = g_i^-(y) \right) \ \vee \ \left(y = g_i^+(x) \ \vee \ y = g_i^-(x) \right).$$

There is an arc from x to y (resp. from y to x) if φ_1 (resp. φ_2) holds:

$$\varphi_1(x, y) = \bigvee_{1 \leqslant i \leqslant k} (x = g_i^-(y) \ \vee \ y = g_i^+(x)).$$
$$\varphi_2(x, y) = \bigvee_{1 \leqslant i \leqslant k} (x = g_i^+(y) \ \vee \ y = g_i^-(x)).$$

This finishes the proof. ∎

Since every planar graph has arboricity at most 3, we have the following as a corollary.

Corollary 9.1 *The adjacency query in planar graphs is representable by* 3 *functions from vertices to vertices. The adjacency query and edge directions in directed planar graphs are representable by* 6 *functions from vertices to vertices.*

We now explain how to represent the set $Bar(X)$ by finitely many functions. However, two vertices can be adjacent to an unbounded number of faces and then given a set X, the set $Bar(X)$ could be unbounded. We will therefore explain how to do that for classes of planar graphs where each pair of vertices is incident with a bounded number of faces.

Definition 9.7 (m-Face Bounded Planar Graphs and Face Selection Functions)
Let G be a plane graph and let x and y be two distinct vertices. We let $Faces(x,y)$ be the set $\{f \in F_G \mid f$ is incident with both x and $y\}$. We say that G is *m-face bounded* if for every $x, y \in V_G$, $x \neq y$, we have $|Faces(x,y)| \leq m$. The distinct vertices x and y verifies the *same-face property*, denoted by $sf(x,y)$, if and only if $|Faces(x,y)| \geq 1$.

An *m-tuple of face selection functions* is an m-tuple $(Select_i)_{i \in [m]}$ of partial functions: $V_G \times V_G \to F_G$ such that for every two distinct vertices x and y,

$$Select_i(x,y) \neq Select_j(x,y) \qquad\qquad\qquad \text{for } i,j \in [m], \ i \neq j,$$
$$Select_i(x,y) \in Faces(x,y) \qquad\qquad\qquad\qquad \text{for all } i \in [m],$$
$$Faces(x,y) = \{Select_1(x,y), \ldots, Select_m(x,y)\} \qquad \text{if } |Faces(x,y)| \leq m.$$

For every positive integer m, the *m-tuple of face selection problem* consists in the definition of an m-tuple of face selection functions.

Note that we do not require $Select_i(x,y) = Select_i(y,x)$ for all $i \in [m]$. An example of a class of planar graphs that are 2-face bounded is the class of 2-connected graphs obtained from simple 3-connected graphs by the replacement of some edges by paths (such graphs have unique embeddings in the sphere).

Proposition 9.2 *The same-face property in plane graphs is representable by* 15 *functions. For every m, the same-face property and the m-tuple of face selection problem in plane graphs are representable by $15 + 3m$ functions.*

Proof. Let G be a simple connected plane graph and let $G^+ = \langle V_G \cup F_G, E_G \cup E' \rangle$ be its augmented planar. By Lemma 9.2, the adjacency in G^+ is represented by 3 functions that we denote g_1^+, g_2^+ and g_3^+. One can verify that for every distinct two vertices x and y, $sf(x,y)$ holds if and only if the following formula φ holds:

$$\bigvee_{1 \leq i,j \leq 3} g_i^+(x) = g_j^+(y) \in F_G \tag{7a}$$

$$\vee \bigvee_{1 \leq i,j \leq 3} g_i^+(x) \in F_G \wedge g_j^+(g_i^+(x)) = y \tag{7b}$$

$$\vee \bigvee_{1 \leq i,j \leq 3} g_i^+(y) \in F_G \wedge g_j^+(g_i^+(y)) = x \tag{7c}$$

$$\vee \ \exists f \in F_G \Big(\bigvee_{1 \leq i,j \leq 3} g_i^+(f) = x \wedge g_j^+(f) = y \Big). \tag{7d}$$

We now define a set $\mathcal{F}$ of 15 functions in order to transform φ into a formula $\varphi' \in \mathcal{B}(\mathcal{F}, \{x,y\})$. For every $i \in [3]$ we let $g'_{iG} : V_G \to F_G$ be such that for every $x \in V_G$,

$$g'_{iG}(x) = \text{ if } g_{iG}^+(x) \in F_G, \text{ then } g_{iG}^+(x) \text{ else undefined}.$$

We can then replace in φ the sub-formula $g_i^+(x) = g_j^+(y) \in F$ by $g'_i(x) = g'_j(y)$. For every $i,j \in [3]$, we let $g_{i,jG} : V_G \to V_G$ be such that for every $x \in V_G$:

$$g'_{i,jG}(x) = \text{ if } g_{iG}^+(x) \in F_G \text{ and } g_{jG}^+(g_{iG}^+(x)) \text{ is defined, then } g_{jG}^+(g_{iG}^+(x))$$

$$\qquad\qquad \text{else undefined}.$$

We can then replace in φ the sub-formulas $g_i^+(x) \in F_G \wedge g_j^+(g_i^+(x)) = y$ and $g_i^+(y) \in F_G \wedge g_j^+(g_i^+(y)) = x$ by respectively $g_{i,j}(x) = y$ and $g_{i,j}(y) = x$.

It remains now to eliminate the existential quantifier in the sub-formula (7d) in φ. For that we define an auxiliary planar graph H, with $V_H = V_G$ and $xy \in E_H$ if and only if for some $i, j \in [3]$ and $f \in F_G$, we have $g_{iG}^+(f) = x$ and $g_{jG}^+(f) = y$. We can obtain a planar embedding of H from an embedding of G because one adds to each face of G at most 3 edges. Therefore, H is a planar graph and by Lemma 9.2, the adjacency is representable by 3 functions from V_H to V_H, hence from V_G to V_G, that we denote by h_{1G}, h_{2G} and h_{3G}. Condition (7d) can thus be replaced by:

$$\bigvee_{1 \leq i \leq 3} \big(h_i(x) = y \vee h_i(y) = x\big).$$

Hence, the same-face property is representable by the 15 functions $g_i', g_{i,j}', h_i$ for $i, j \in [3]$.

We now show how to define and represent an m-tuple of face selection functions. We will use cases (7a)-(7d) that characterize the same-face property. We first observe that they are mutually exclusive in the sense that each face of $Faces(x, y)$ is specified by one and only one of them.

We first fix a linear order on F_G, therefore we can consider any subset of F_G as an ordered set, the order inherited from the order of F_G. Let x and y be 2 distinct vertices of G and let $f \in F_G$. If $f \in Faces(x, y)$, we define the (x, y)-*type* t of f as follows:

$$t = \begin{cases} (a, i, j) & \text{if } f = g_{iG}'(x) = g_{jG}'(y) \\ (b, i, j) & \text{if } f = g_{iG}'(x) \text{ and } y = g_{i,jG}'(x) \\ (c, i, j) & \text{if } f = g_{iG}'(y) \text{ and } x = g_{i,jG}'(y) \\ (d, j) & \text{if } f \text{ is the } j\text{-}th \text{ face in the ordered set } F(x, y) \subseteq Faces(x, y) \\ & \text{formed from the faces that have not type neither } (a, i, j), \\ & \text{neither } (b, i, j) \text{ neither } (c, i, j). \end{cases}$$

We have clearly $F(x, y) = F(y, x)$. Note that the (y, x)-type of f is (a, j, i) or (c, i, j) or (b, i, j) or (d, j) if its (x, y)-type is respectively (a, i, j), (b, i, j), (c, i, j) or (d, j).

For every $i \in [3]$ and every $j \geq 1$, we let $h_{i,jG} : V_G \to F_G$ be defined as follows:

$$h_{i,jG}(x) = f \text{ if } h_{iG}(x) \text{ is defined and } f \text{ is the } j\text{-}th \text{ element of } F(x, h_{iG}(x)).$$

For every $x, y \in V_G$, $x \neq y$ and $j \geq 1$, there is at most one face f of (x, y)-type (d, j) and it is characterized by the formula φ_j:

$$\bigvee_{1 \leq i \leq 3} \Big(\big(f = h_{i,j}(x) \wedge y = h_i(x)\big) \vee \big(f = h_{i,j}(y) \wedge x = h_i(y)\big)\Big). \tag{8}$$

Similarly, for each $t \in \{a, b, c\} \times [3] \times [3]$, there is at most one face f of (x, y)-type t and it is characterized by the formula $\varphi_{i,j}$:

$$\varphi_{i,j} = \begin{cases} f = g_i'(x) \wedge f = g_j'(y) & \text{if } f \text{ has } (x, y)\text{-type } (a, i, j), \\ f = g_i'(x) \wedge y = g_{i,j}'(x) & \text{if } f \text{ has } (x, y)\text{-type } (b, i, j), \\ f = g_i'(y) \ \wedge \ x = g_{i,j}'(y) & \text{if } f \text{ has } (x, y)\text{-type } (c, i, j). \end{cases} \tag{9}$$

Let us now order types lexicographically. We get thus for each pair (x, y) of distinct vertices a linear order of the set $Faces(x, y)$ (which can be different from the linear order inherited from the one of F_G). We let $Select_i(x, y)$ be the i-th element of this set and let $\mathcal{F}_m = \{g_i', g_{i,j}', h_i, h_{i,\ell} \mid i, j \in [3], \ \ell \in [m]\}$. It is clear that, for each $i \leq m$, one can express $f = Select_i(x, y)$ by a formula $\varphi(x, y) \in \mathcal{B}(\mathcal{F}_m, \{x, y\})$ which is a disjunction of the formulas $\varphi_{i,j}$ and φ_j in Equations (9) and (8). Hence, we have specified an m-tuple of face-selection functions.   ∎

Remark 9.1 With $15 + 3(m + 1)$ functions one can represent the property that two vertices x and y are incident with at most m faces. For doing so we use the negation of the formula in $\mathcal{B}(\mathcal{F}_{m+1}, \{x, y\})$ that describes the expression of $f = Select_{m+1}(x, y)$.

9.4   The Case of 2-Connected Face Bounded Plane Graphs

We prove in this section a particular case of Theorem 9.1. For every $m \geq 1$, we let $\mathcal{C}_m$ be the set of simple m-face bounded 2-connected planar graphs. The class of 2-connected planar graphs obtained from 3-connected planar graphs by replacing some edges by paths are graphs in $\mathcal{C}_2$. The class of planar graphs of degree at most d is included in $\mathcal{C}_d$. We prove the following.

Theorem 9.2 (Case of m-Face Bounded 2-Connected Planar Graphs) *There exists an $(m \cdot \log)$-labeling scheme $(\mathcal{A}, \mathcal{B})$ for connectivity query on the class of graphs $\mathcal{C}_m$. If n is the number of vertices of the input graph, $\mathcal{A}$ computes the labels in $O(m \cdot n)$-time and $\mathcal{B}$ gives the answer in $O(s^2)$-time where $s \geq 2$ is the size of the data.*

We first prove the following which a consequence of many results.

Proposition 9.3 *For every simple 2-connected planar graph with n vertices, one can construct in $O(n)$-time a corresponding plane graph G, its augmented planar graph G^+ and a straight-line embedding of G^+ with positive integer coordinates in $[3n - 6]$.*

Proof. Let G be a simple 2-connected planar graph with n vertices. We can construct in $O(n)$-time a plane graph $G = \langle V_G, E_G, F_G \rangle$ as a consequence of the well-known linear-time planarity testing algorithms (see for instance [HT74]). There is at most $2n - 4$ faces in G, hence $2n - 4$ face-vertices in the augmented planar graph G^+ of G. We can therefore construct the planar graph G^+ in $O(n)$-time (G^+ has at most $3n - 4$ vertices and at most $9n$ edges). Since G is assumed 2-connected, the planar graph G^+ is simple. We can thus construct a straight-line embedding of G^+ with coordinates in $[3n - 6]$ with the Schnyder's algorithm [Sch90].   ∎

If G is a 2-connected m-face bounded plane graph, by means of $15 + 3m$ functions we can determine, for every 2 distinct vertices z and t, the set of at most m faces incident with both z and t. Therefore, we can determine, for every set $X \subseteq V_G$, the set $Bar(X)$. By Proposition 9.3, the augmented planar graph G^+ of G admits a straight-line embedding. It remains then to show how to verify that $p(x)$ and $p(y)$ are separated by $Bar(X, \mathcal{E}^+)$ if $\mathcal{E}^+ = (p, s)$ is an embedding of G^+.

If x and y are points in $\mathbb{R}^2$, we denote by $[x, y] \subseteq \mathbb{R}^2$ the straight-line segment with ends x and y. Two straight-line segments $[x, y]$ and $[x', y']$ are *non-crossing* if $[x, y] \cap [x', y'] \subseteq \{x, y\} \cap \{x', y'\}$. A finite set Y of pairwise non-crossing straight-line segments is called a *subdivision of the plane*. The union $\bigcup Y$ of the segments in Y is a closed subset of $\mathbb{R}^2$. We need an algorithm for the following problem:

Input: A subdivision Y of the plane and 2 points p and q.
Output: Are p and q separated by $\bigcup Y$?

This problem is equivalent to the problem of identifying, for each of the point, the connected component of $\mathbb{N}^2 - \bigcup Y$ that contains it and is called in [dBKOS91, Sno04] the *planar point location problem*. Many algorithms, that use different techniques, had been proposed (see for instance the article [ST86] or the book [dBKOS91]). All these algorithms are based on the following principle: construct a data-structure from which one can identify the connected component that contains a given point. The goal is to obtain a data-structure of size linear and each query takes logarithmic-time. We recall the following one (one can also choose any other implementation).

Theorem 9.3 *[dBKOS91, Theorem 6.8] Let Y be a subdivision of the plane consisting of m straight-line segments. One can construct in $O(m \cdot \log(m))$-expected time a data structure of size $O(m)$ from which one can identify in $O(\log(m))$-time, in the worst case, the connected component in $\mathbb{N}^2 - \bigcup Y$ that contains a given point $p \in \mathbb{N}^2 - \bigcup Y$. One can therefore test in $O(\log(m))$-time whether two elements of $\mathbb{N}^2 - \bigcup Y$ are separated by $\bigcup Y$.*

We can now prove Theorem 9.2.

Proof of Theorem 9.2. Let G be a plane graph in $\mathcal{C}_m$ with n vertices. Let $\mathcal{E}^+ = (p, s)$ be a straight-line embedding of G^+ constructed by Proposition 9.3. Clearly, $|C(x)| \leq 2 \cdot \lceil \log(n) \rceil + 2 \cdot \log(3)$ for every $x \in V_{G^+}$.

By Proposition 9.2 and Convention 9.1 the m-face selection problem is representable by $p = 15 + 3m$ functions, i.e., by means of p functions $f_1, \ldots, f_p$ we can associate with every $z, t \in V_G$, $z \neq t$ the set of at most m faces incident with both. For every vertex z in V_G, we let

$$D(z) = \Big(p(z), p(f_1(z)), \ldots, p(f_p(z)) \Big).$$

We have clearly $|D(z)| = O(m \cdot \log(n))$. Let x and y be two vertices of G and let $X \subseteq V_G - \{x, y\}$ be given by their labels. We now explain how to decide if x and y are separated by X by using only their labels. By Proposition 9.2, we can associate, for every two distinct vertices z and t, the set of the at most m faces incident with both by using $f_1(z), \ldots, f_p(z), f_1(t), \ldots, f_p(t)$. Since we can recover for every $i \in [p]$ and every $z \in V_G$ the

value of $f_i(z)$ from $D(z)$, we can therefore define from the set $\{D(z) \mid z \in X\}$ the set of straight-line segments forming $Bar(X, \mathcal{E}^+)$ in $O(|X|^2)$-time ($\mathcal{E}^+$ is a straight-line embedding). By Theorem 9.3, we can construct in $O(s \cdot \log(s))$-expected time, where $s = |Bar(X)|$, a data structure and decide in $O(\log(s))$-time if $p(x)$ and $p(y)$ are separated by $Bar(X, \mathcal{E})$. And this sufficient by Proposition 9.1.

It remains to bound the size of $Bar(X)$, hence $Bar(X, \mathcal{E}^+)$ since $Bar(X) = Bar(X, \mathcal{E}^+)$. We claim that $Bar(X) \leq m \cdot (3 \cdot |X| - 6)$. To see this, consider the sub-graph $G' = G^+[Bar(X)]$. It is a plane bipartite graph with vertex set $X' \cup F$ for some $X' \subseteq X \subseteq V_G$ and $F \subseteq F_G$ (we recall that a vertex of X may not occur in $Bar(X)$). Let H be the graph with vertex set X' and an edge between z and t whenever there is $f \in F$ such that (zf, f, tf) is a corner of G'. It is clear that H is planar, that $|E_H| = |E_{G'}| = |Bar(X)|$ and that there are no more than m parallel edges in H between two vertices. It follows that $|E_H| \leq m \cdot (3 \cdot |X'| - 6) \leq m \cdot (3 \cdot |X| - 6)$. This finishes the proof.   ■

9.5 The Case of 2-Connected Planar Graphs

In this section we prove Theorem 9.1 for the case of 2-connected planar graphs. Since two vertices z and t may be incident with an unbounded number of faces, we may have in $Bar(X)$ an unbounded number of paths $z - f - t$, associated with all faces f incident with both z and t. In our labeling scheme in order to build $Bar(X, \mathcal{E}^+)$, we need the coordinates $p(f)$ of all these faces but they cannot be encoded as lists $(p(f_1), \ldots, p(f_k))$ of bounded length attached to vertices z and t. However, we can overcome this difficulty by replacing this unbounded number of paths by only one of them if there are at least 3 faces incident with both z and t. We obtain in this way the *reduced barrier* $RBar(X, \mathcal{E}^+) \subseteq Bar(X, \mathcal{E}^+)$. But in some cases, the reduced barrier cannot witness that two vertices x and y are separated by X. In order to overcome this second difficulty we borrow tools from Di Battista and Tamassia [BT96] and use the decomposition of 2-connected components into 3-connected components. The bad cases can be reduced to the geometrical tools used in Section 9.4, hence we can use Theorem 9.3 and to cases where the vertices are separated by *attachment vertices* of the 3-connected components. These later cases can be defined by MS formulas and since the decomposition into 3-connected components is a tree we can use Theorem 7.4.

In Section 9.5.1 we present the decomposition of 2-connected planar graphs into 3-connected components with help of bipolar orientations studied for instance in [dFdMR95]. In Section 9.5.2 we present the reduced barrier and prove Theorem 9.1 for the case of 2-connected planar graphs stated as follows.

Theorem 9.4 (Case of 2-Connected Planar Graphs) *There exists a* log-*labeling scheme* $(\mathcal{A}, \mathcal{B})$ *for the connectivity query on the class of* 2-*connected planar graphs. If n is the number of vertices of the input graph, $\mathcal{A}$ computes the labels in $O(n \cdot \log(n))$-time and $\mathcal{B}$ gives the answer in $O(m^2)$ if $m \geq 2$ is the size of the data.*

We recall first the definition of colored trees as relational structures. Let A be a finite set of labels and let $\mathcal{T}_A$ be the relational signature $\{E, (nlab_a)_{a \in A}, (elab_a)_{a \in A}\}$ on graphs such that for every relational $\mathcal{T}_A$-structure $T = \langle V_T, E_T, (nlab_{aT})_{a \in A}, (elab_{aT})_{a \in T} \rangle$, V_T is its set of

vertices, E_T its adjacency relation and for every $a \in A$ and every $x, y \in V_T$ we have $nlab_{aT}(x)$ if and only if x is colored by a and we have $elab_{aT}(x, y)$ if and only if the edge xy (resp. arc (x, y)) is colored by a. We denote by $\mathcal{T}(A)$ the class of relational $\mathcal{T}_A$-structures that are trees.

9.5.1   Bipolar Plane Graphs and Polar Pairs

In this section we show all the tools borrowed from [BT96] with help of the bipolar orientations of planar graphs. We prove also some technical properties that will be used in Section 9.5.2. We now define the notion of *bipolar plane graphs*. The *border of a face* in a plane graph is the set of edges that bound the face.

Definition 9.8 (Bipolar Graphs and Bipolar Plane Graphs) A *bipolar graph* is a directed graph G, without circuits, having a unique vertex of in-degree 0, called its *South pole* and denoted by $s(G)$, a unique vertex of out-degree 0, called its *North pole* and denoted by $n(G)$ and such that every vertex v in $V_{Int}(G) = V(G) - \{s(G), n(G)\}$, called an *internal vertex*, is on a directed path from $s(G)$ to $n(G)$.

A directed plane graph G is a *bipolar plane graph* if it is bipolar as a graph and has a planar embedding such that the two poles are incident with the external face.

In a bipolar plane graph G, a face $f \in F_G$ is called a *border face of G* if its border consists of two disjoint directed paths, called respectively *left-border* and *right-border* of f, from a vertex $s(f)$, called its *South pole*, to a vertex $n(f)$, called its *North pole*. If the external face of G is a border face of G, the left-border and the right-border of f are called respectively *left-border* and the *right-border* of G.

Figure 30 shows a bipolar plane graph; its left-border is the path (f_1, f_{12}) and its right-border is the path (f_{14}, f_{15}, f_{17}).

A bipolar graph with adjacent poles is 2-connected. We recall the following properties of bipolar plane graphs proved in [ET76, dFdMR95].

Lemma 9.3 *Let G be a 2-connected planar graph with n vertices. Then for every edge $e = xy$ of G, we can construct in $O(n)$-time an orientation $\overrightarrow{G}$ of G that is a bipolar plane graph with North pole x and South pole y.*

We can cite two different algorithms for constructing bipolar orientations. The first, due to Even and Tarjan [ET76], constructs for every 2-connected graph G and every edge e a bipolar orientation in $O(|V_G| + |E_G|)$-time. Their algorithm constructs the bipolar orientation by using Depth-First-Search algorithms. Since planar graphs has $O(n)$ edges, the algorithm in [ET76] constructs a bipolar orientation in $O(n)$-time. The second algorithm, due to de Fraysseix et al. [dFdMP95], takes as input a planar map of a 2-connected planar graph G with n vertices and an edge of G and constructs a bipolar orientation in $O(n)$-time. Since, a planar map of a planar graph G can be computed in linear-time [HT74] we are done. More informations on bipolar graphs and bipolar orientations of undirected graphs are given in [dFdMR95].

Lemma 9.4 ([TT86, dFdMR95]) *For every planar embedding of a 2-connected bipolar plane graph G,*

1. *The incoming arcs of each vertex $x \in V_G$ appear consecutively in the circular incidence sequence of x and so do the outgoing arcs.*

2. *Each face $f \in F_G$ is a border face of G.*

By Lemma 9.4, we can write the circular sequence of arcs incident with x as $\overrightarrow{in}(x) \circ \overrightarrow{out}(x)$ where $\overrightarrow{in}(x)$ (resp. $\overrightarrow{out}(x)$) is the sequence of incoming (resp. outgoing) arcs of x. We denote $\overrightarrow{out}(s(G))$ by $\overrightarrow{s}(G)$ and $\overrightarrow{in}(n(G))$ by $\overrightarrow{n}(G)$. We now define a way to decompose a bipolar plane graph into smaller bipolar ones by means of substitutions of edges by graphs. The disjoint union of sets is denoted by $\uplus$.

Definition 9.9 (Decomposition of Bipolar Plane Graphs) Let R be a bipolar plane graph with m edges denoted $e_1, \ldots, e_m$. Let H, $G_1, \ldots, G_m$ be bipolar plane graphs. We write $H = R(G_1, \ldots, G_m)$ if and only if the following conditions (D1)-(D5) hold:

(D1) $V_R \cap V_{Int}(G_i) = \emptyset$ and $V_{Int}(G_i) \cap V_{Int}(G_j) = \emptyset$ for all $i, j \in [m]$, $i \neq j$.

(D2) For every $e_i = (s_i, n_i) \in E_R$, we have $s_i = s(G_i)$ and $n_i = n(G_i)$ where $i \in [m]$.

(D3) $V_H = V_R \cup V_{G_1} \cup \cdots \cup V_{G_m}$.

(D4) $E_H = E_{G_1} \uplus \cdots \uplus E_{G_m}$.

(D5) For every $x \in V_H$, we have

$$\overrightarrow{in_H}(x) = \begin{cases} \overrightarrow{in_{G_i}}(x) & \text{if } x \in V_{Int}(G_i), \\ \overrightarrow{n(G_{i_1})} \bullet \cdots \bullet \overrightarrow{n(G_{i_m})} & \text{if } x \in V_R \text{ and } \overrightarrow{in_R}(x) = (e_{i_1}, \ldots, e_{i_m}). \end{cases}$$

$$\overrightarrow{out_H}(x) = \begin{cases} \overrightarrow{out_{G_i}}(x) & \text{if } x \in V_{Int}(G_i), \\ \overrightarrow{s(G_{i_1})} \bullet \cdots \bullet \overrightarrow{s(G_{i_m})} & \text{if } x \in V_R \text{ and } \overrightarrow{out_R}(x) = (e_{i_1}, \ldots, e_{i_m}). \end{cases}$$

If $H = R(G_1, \ldots, G_m)$, we say that H *decomposes into* $G_1, \ldots, G_m$. We have

$$V_{Int}(R(G_1, \ldots, G_m)) = V_{Int}(R) \uplus V_{Int}(G_1) \uplus \cdots \uplus V_{Int}(G_m).$$

A sub-graph H of a bipolar plane graph G is called a *factor* of G if:

1. H contains all directed paths in G from $s(H)$ to $n(H)$.

2. H contains all edges of G incident with a vertex of $V_{Int}(H)$.

Informally, with Conditions (D1)-(D4) we could say that H is obtained from R by the replacement of an edge e_i by the graph G_i. Clearly, by these conditions, H is bipolar with $s(H) = s(R)$ and $n(H) = n(R)$. Condition (D5) relates $H, R, G_1, \ldots, G_m$ as plane graphs, and not only as graphs as do Conditions (D1)-(D4). Condition (D5) means that planar embeddings are preserved in the replacement in R of e_i by G_i.

Note that if H is a factor of G, then there exists a bipolar plane graph R such that G results from the replacement in R of some edge e by H. We recall a particular decomposition of bipolar plane graphs.

Definition 9.10 (Parallel-Composition) Let R be a bipolar plane graph that consists of $m \geq 2$ parallel edges, from $s(R)$ to $n(R)$ such that $\overrightarrow{n}(R) = (e_1, e_2, \ldots, e_m)$ and $\overrightarrow{s}(R) = (e_m, \ldots, e_2, e_1)$. If $H = R(G_1, \ldots, G_m)$, then we call H the *parallel-composition* of $G_1, \ldots, G_m$ and we denote it by $G_1 // \cdots // G_m$.

A bipolar plane graph which is the parallel composition of bipolar plane graphs $G_1, \ldots, G_m$ is called a *//-graph*. A bipolar plane graph which is not a //-graph is called a *//-atom*.

We call a //-graph that is a factor of a bipolar plane graph a *//-factor*.

The parallel operation $//$ is associative and not necessarily commutative. We now define a notion of hierarchical decomposition of bipolar plane graphs based on parallel-composition and decomposition of bipolar plane graphs in terms of simple bipolar plane graphs. If u is a node in a rooted tree T we denote by $d(u)$ the out-degree of u, i.e., the number of children of u; we also recall that we denote by $T \downarrow u$ the sub-tree of T rooted at $u \in V_T$ (Chapter 1).

Definition 9.11 (Decomposition Tree of Bipolar Plane Graphs) A *decomposition tree* T of a bipolar plane graph G is a labeled rooted ordered tree where each internal node is labeled by a parallel operation $//$ or a simple //-atom R that is not an edge such that:

(DT1) The leaves of T are labeled by the edges of G.

(DT2) If an internal node u of T is labeled by a parallel operation, then each of its children is labeled by an edge of G or a simple //-atom that is not an edge. Such a node is called a *//-node*.

(DT3) If an internal node u of T is labeled by a simple //-atom R with $m \geq 2$ edges, then each of its children is labeled by an edge of G or a parallel operation. Such a node is called a *non-//-node*.

(DT4) $val(T) = G$ where:

$$val(T) = \begin{cases} e & \text{if } T \text{ is a leaf labeled by } e, \\ R(val(T_1), \ldots, val(T_m)) & \text{if } T = R(T_1, \ldots, T_m), \\ val(T_1) // \cdots // val(T_m) & \text{if } T = //(T_1, \ldots, T_m). \end{cases}$$

Let T be a decomposition tree of a bipolar plane graph G and let u be a node of T. We let $G(u)$ be the bipolar plane graph $val(T \downarrow u)$ and we denote $s(G(u))$ by $s(u)$ and $n(G(u))$ by $n(u)$. If u is a //-node with children, in this order, $u_1, \ldots, u_{d(u)}$, we let $F_j(u)$ be the face whose border cycle consists of the right-border of $G(u_j)$ and the left-border of $G(u_{j+1})$ for $j = 1, \ldots, d(u) - 1$.

A *polar pair* is a pair of vertices of the form $(s(u), n(u))$ for some node u of T; it is called a *//-polar pair* if u is a //-node. We say that a polar pair (z, t) *separates* x and y if $\{z, t\} \cap \{x, y\} = \emptyset$ and $(z, t) = (s(u), n(u))$ for some node u such that $x \in V_{Int}(G(u))$ and $y \notin V_{Int}(G(u))$ or vice-versa by exchanging z and t.

We illustrate Definition 9.11 with Example 9.5.

Example 9.5 A bipolar plane graph G with $V_G = \{s, n, a, b, c, d, k, m, p, q\}$ and $E_G = \{f_1, \ldots, f_{17}\}$ is shown on Figure 30. The graph G can be expressed by:

$$G = R_1\Big(f_1, f_2, \big(f_3//R_3(f_4, f_5)\big), \big(f_6//R_4(f_7, f_8)\big), \big(f_9//R_5(f_{10}, f_{11})\big)f_{12}, f_{13}\Big),$$
$$//R_2\Big(f_{14}, f_{15}, \big(f_{16}//f_{17}\big)\Big)$$

where $R_1, \ldots, R_5$ are shown on Figure 32. The corresponding tree is on Figure 31. The pairs $(s, b), (a, c), (c, b), (c, n)$ are polar, the pairs $(c, b), (a, c)$ are $//$-polar and the pairs $(s, k), (d, n)$ are not polar.
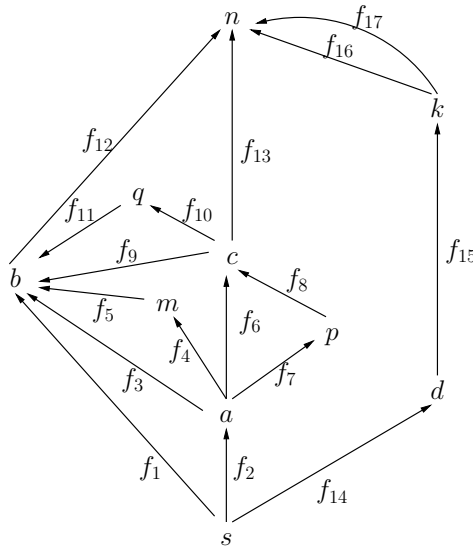
Figure 30: A bipolar plane graph.
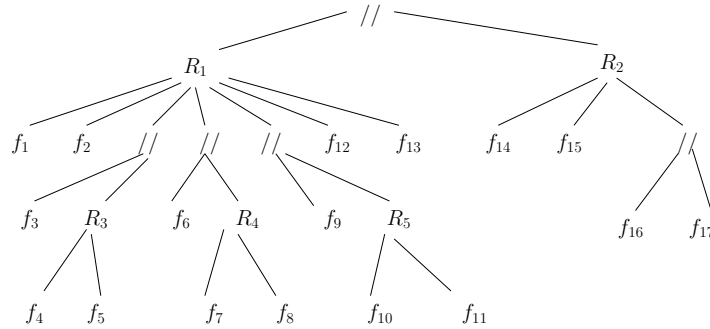
Figure 31: The decomposition tree of the graph on Figure 30.

Lemma 9.5 *Let T be a decomposition tree of a bipolar plane graph and let $R_1, \ldots, R_p$ be the $//$-atoms associated with the non-$//$-nodes of T enumerated as $u_1, \ldots, u_p$. Then $V_{Int}(G) = \biguplus_{1 \le i \le p} V_{Int}(R_i)$. The sets $V_{Int}(R_i)$ are all nonempty.*

It is proved in [BT96] with help of [HT73] that every bipolar plane graph has a unique decomposition tree computable in linear-time, that we state in the following.
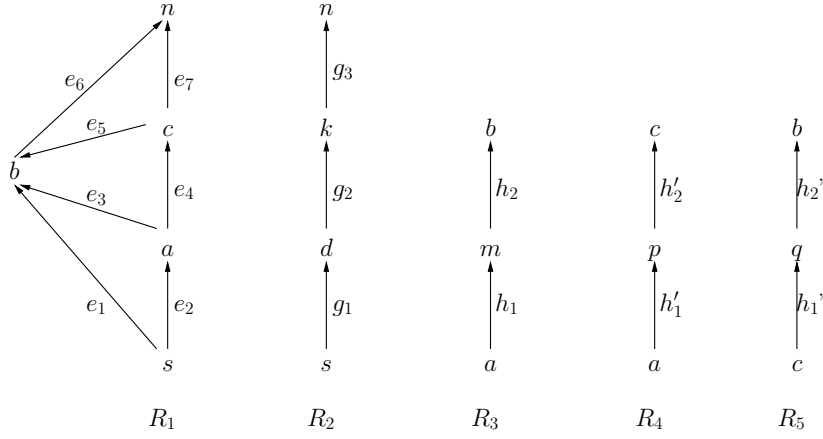
Figure 32: The graphs R_1, R_2, R_3, R_4 and R_5 (cf Figure 31 and Example 9.5).

Proposition 9.4 *Let G be a bipolar plane graph with n vertices. Then it has a unique decomposition tree T that has $O(n)$ nodes and can be computed in $O(n)$-time. Moreover, if $R_1, \ldots, R_p$ are the simple //-atoms associated with the non-//-nodes of T, then the total size of $\biguplus_{1 \le i \le p} E_{R_i}$ is $O(n)$.*

If T is the decomposition tree of a bipolar plane graph G, then the leaves of T are in bijection with the edges of G (cf. Example 9.5 and Figures 30 and 31). Hence, if $u \in V_T$ is a leaf, then $G(u)$ is an edge of G; if u is a //-node, then $G(u)$ is a //-factor of G, otherwise $G(u)$ is a //-atom of G.

The idea of our labeling scheme is to identify in the decomposition tree the polar pairs that can separate the two vertices. If these cases of separations do not apply, we use geometrical tools.

A polar pair $(s(u), n(u))$ is not a //-polar pair in the following few cases: u is a leaf and the corresponding edge is simple or it is $(s(u), n(u))$ and $G(u)$ is a //-atom. It follows that if a polar pair separates x and y, it is necessarily a //-polar pair. It is clear that if x and y are separated by a polar pair (z, t), then they are separated by the set $\{z, t\}$. Since only //-polar pairs can separate vertices, one can ask when a pair of vertices form a //-polar pair. We do not have a complete answer however, we give a sufficient condition which is enough for our purposes as we will see later.

Lemma 9.6 *Let G be a bipolar plane graph with adjacent poles and let x and y be two vertices of G. If x and y are incident with 3 common faces, then they form a //-polar pair.*

Let us first prove two technical facts.

Claim 9.1 *Let G be a bipolar plane graph with adjacent poles and let x and y be two vertices of G incident with at least 3 faces f, g and h. Then x and y are on a same border of each face f, g and h.*

Proof of Claim 9.1. Assume that x and y are not on a same border of f. Then none of them is a pole of f. Let p_1 and p_2 be respectively the left-border and the right-border of f

and assume that p_1 contains x.

CASE 1. f is the external face. By hypothesis $s(f) = s(G)$ and $n(f) = n(G)$ are adjacent. Let $C = x - f - y - g - x$, C' consisting of p_1 and the edge $s(f) - n(f)$, and C'' consisting of p_2 and the edge $s(f) - n(f)$, be 3 cycles of G^+ (note that C' and C'' are also cycles of G). Note that $s(f) - n(f) \neq p_2$ since p_2 must contain y which is not a pole of f. C has only x in common with C' and has only y in common with C''. The face g must be included in C' or in C'' and the two cases contradict the planarity of G^+ (see for instance [Cou00]).

CASE 2. f is not the external face; at least one of g and h, say g, is not the external face of G. We let p_3 and p_4 be respectively 2 shortest paths from $n(f)$ to $n(G)$ and from $s(f)$ to $s(G)$. We consider the cycle $C = x - f - y - g - x$ and the cycle C', consisting of p_1, p_3, the left-border of G and p_4. C has only x in common with C'. The face g must be included in C' and this contradicts the planarity of G^+.

Hence, x and y are on a same border of each face f, g and h.   ∎

Claim 9.2 *Let G be a bipolar plane graph with adjacent poles and let x and y be two vertices of G incident with at least 3 faces f, g and h. At least one of f, g or h has (y, x) or (x, y) as pair of poles.*

Proof of Claim 9.2. By Claim 9.1, x and y are linked by a path and we can assume without loss of generality that there exists a directed path from y to x in G. We claim that at least f, g or h has (y, x) as pair of poles.

In the plane graph G^+ we have 3 paths $y - f - x$, $y - g - x$ and $y - h - x$, and without loss of generality we have around x the circular order (xf, xg, xh). Because of planarity (see [Cou00]) we have necessarily around y the circular order (yf, yh, yg). Without loss of generality we can assume that g is inside the cycle C'' of G^+ defined as $x - f - y - h - x$. Let p_1 and p_2 be respectively the left-border and the right-border of g and assume that p_1 contains x and y.

We will prove that $x = n(g)$. If this is not the case we let x' be the vertex following x on p_1. The right-border of f (resp. the left-border of h) contains x. Let z (resp. t) be the vertex that precedes x on this border. Figure 33 shows a part of G^+ around x.

We must have around x the following cyclic order of edges: $z \to x$, $t \to x$ and $x \to x'$ by Lemma 9.4(1). We have the edge gx between $z \to x$ and $t \to x$. But we also have the edge $x'g$ in G^+. We have then two cycles, $x - g - x' \leftarrow x$ and the cycle $y \xrightarrow{*} z \to x \leftarrow t \xleftarrow{*} y$, that contradicts the planarity of G^+. Hence, $x = n(g)$ and similarly $y = s(g)$.   ∎

We can now prove Lemma 9.6.

Proof of Lemma 9.6. Let G be a bipolar plane graph with adjacent poles and let T be the decomposition tree of G. Let x and y be two vertices of G incident with 3 faces f, g and h (and possibly others). By Claims 9.1 and 9.2, at least f, g or h has (x, y) or (y, x) as pair of poles. Without loss of generality we assume that g as (y, x) as pair of poles.
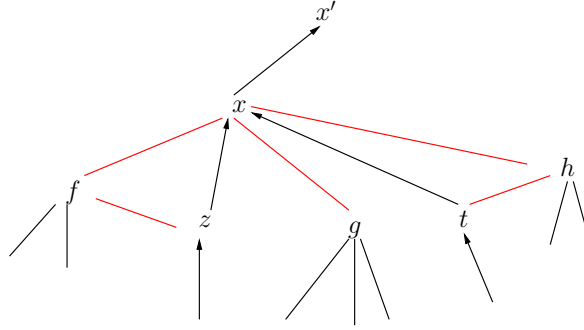
Figure 33: Illustration of Claim 9.2.

We consider the induced sub-graph $G[U]$ where U consists of x, y and of all the vertices that lie inside the cycle $x - f - y - g - x$. It is a factor of G with poles $s(g) = y$ and $n(g) = x$. The sub-graph $G[U']$ with U' defined similarly from the cycle $x - g - y - h - x$ is also a factor with the same poles. Hence, $G[U \cup U'] = G[U]//G[U']$ and is a //-factor of G. Hence, (y, x) is a //-polar pair of G.   ∎

As a consequence of Lemma 9.6 we get the following.

Lemma 9.7 *Let G be a bipolar plane graph with adjacent poles and let $m \geq 3$ be a positive integer. Two vertices x and y are incident with exactly m faces if and only if they are the poles of $G(u)$ for some //-node u such that:*

1. *either u is the root and u has m sons,*

2. *or u is not the root and it has $m - 1$ sons.*

The first step of the labeling scheme is to prove that we can decide when a pair of vertices (z, t) forms a polar pair and separates two vertices x and y by means of labels of size $O(\log(n))$, n is the number of vertices of the input graph.

Definition 9.12 (Polar-Pair Separation Query) We denote by $pps(x, y, z, t)$, called the *polar-pair separation query*, the graph property on bipolar plane graphs that expresses that the vertices x and y are separated by the polar pair (z, t).

Proposition 9.5 *There exists a* log*-labeling scheme $(\mathcal{A}, \mathcal{B})$ for the polar-pair separation query on the class of bipolar plane graphs. Moreover, if n is the number of vertices of the input graph, $\mathcal{A}$ computes the labels in $O(n \cdot \log(n))$-time and $\mathcal{B}$ gives the answer in constant-time.*

We first explain the main ideas. We want to apply Theorem 7.4 to decomposition trees. However, the decomposition tree of a bipolar plane graph does not give sufficient informations for answering the polar-pair separation query. The idea is to transform the decomposition tree T into a tree T^*, that encodes enough information about G, some nodes of which are (or correspond bijectively to) the vertices of G. Letting $u_1, \ldots, u_p$ be the non-//-nodes of T with associated graphs $R_1, \ldots, R_p$ respectively, we let a vertex x of G belonging to $V_{Int}(R_i)$ be a son of u_i. (The poles of G are represented in a special way as sons of the root.) The major problem

is to identify polar pairs. We will use auxiliary unary functions in addition to the information encoded in T^*. Consider a polar pair (z, t) with $\{z, t\} = \{s(u), n(u)\} \neq \{s(G), n(G)\}$. There are two cases (up to exchanging z and t):

CASE 1. $z, t \in V_{Int}(R_i)$ and there is an arc $z \to t$ or $t \to z$ in R_i, which is actually a place where a bipolar plane graph $G(u)$ is substituted so that $z = s(u)$ and $t = n(u)$ (Definitions 9.10 and 9.11). Since R_i is planar, we can use Corollary 9.1 and represent such arcs (and their directions) by 6 unary functions (g_i for $i = 3, \ldots, 8$). If $t \in \{g_4(z), g_6(z), g_8(z)\}$, then there is an arc $z \to t$ and we let u be a son of z in T^* with the arc $z \to u$ labeled by $i \in \{4, 6, 8\}$. If $t \in \{g_3(z), g_5(z), g_7(z)\}$, then there is an arc $t \to z$ and we let u be a son of t with the arc $t \to u$ labeled by $i \in \{3, 5, 7\}$. It follows that for a node u, son of a node z representing a vertex of G such that the arc $z \to u$ is labeled by $i \in \{3, 4, \ldots, 8\}$, we have that z and $g_i(z)$ are the poles of $G(u)$. Furthermore, z is the South pole if i is even and the North pole if i is odd.

CASE 2. $z \in V_{Int}(R_i)$, t is a pole of R_i. In this case we let $g_1(z) = t$ if t is the South pole and $g_2(z) = t$ if t is the North pole. These values of g_1 and g_2 represent respectively arcs from $t = s(R_i)$ to z and from z to $t = n(R_i)$ of R_i, to which some $G(u)$ is substituted. Similarly as in the previous case, we let in T^* the node u be a son of z (with arc $z \to u$ labeled by 1 or 2). If $z \to u$ is labeled by 1 or 2, then z is the South pole or the North pole of $G(u)$ respectively.

To conclude this informal presentation, we state that the tree T^* (to be defined formally below) belongs to $\mathcal{T}(A)$ where A is the set of labels $\{\mathbf{P}, \mathbf{N}, \mathbf{V}, 1, \ldots, 8\}$. The nodes labeled $\mathbf{V}$ correspond bijectively to the vertices of G; those labeled by $\mathbf{N}$ are the non-$//$-nodes of T (the decomposition tree of the considered graph); those labeled by $\mathbf{P}$ are some of the $//$-nodes of T. The integers $1, \ldots, 8$ are arc labels used as explained above to encode, together with functions $g_1, \ldots, g_8$, the arcs of the graphs R_i and consequently the polar pairs of G.

We now give the precise definition of T^*. It is worth noticing that if a bipolar plane graph has its poles adjacent, then in its decomposition tree T the root is a $//$-node; if $R_1, \ldots, R_p$ are the $//$-atoms associated with the non-$//$-nodes $u_1, \ldots, u_p$ of T, then $(V_{Int}(R_i))_{1 \leq i \leq p}$ is a partition of V_G (Lemma 9.5). We recall that for $i \in [p]$, if each u_i has $v_{i_1}, \ldots, v_{i_m}$ as outgoing arcs in this order, then each arc $u_i \to v_{i_j}$ corresponds to the arc e_j in R_i. An arc $z \to t$ labeled j is denoted by $z \xrightarrow{j} t$.

Definition 9.13 (The Labeled Tree T^*) Let G be a bipolar plane graph with adjacent poles and let T be its decomposition tree. We let $u_1, \ldots, u_p$ be its non-$//$-nodes with associated graphs $R_1, \ldots, R_p$. For each $i \in [p]$, we let $g_{i,j}^{\alpha}$, $j \in [3]$, $\alpha \in \{+, -\}$ be the functions that represent the directions of arcs in R_i. We let $g_1, \ldots, g_8 : V_{Int}(G) \to V_G$ be such that for each x in $V_{Int}(R_i)$,

$$
\begin{aligned}
g_1(x) &= s(R_i) & & \text{if } s(R_i) \to x, \\
g_2(x) &= n(R_i) & & \text{if } x \to n(R_i), \\
g_{2j+1}(x) &= y & & \text{if } g_{i,j}^{-}(x) = y \text{ and } y \in V_{Int}(R_i) \text{ for } j \in \{1, 2, 3\}, \\
g_{2j+2}(x) &= y & & \text{if } g_{i,j}^{+}(x) = y \text{ and } y \in V_{Int}(R_i) \text{ for } j \in \{1, 2, 3\}.
\end{aligned}
$$

The *labeled tree of G*, denoted by $T^* \in \mathcal{T}(A)$ where $A \in \{\mathbf{V}, \mathbf{P}, \mathbf{N}, 1, \ldots, 8\}$, is defined such that:

(T1) $V_{T^*} = V_G \cup \{u \in V_T \mid v \leq_T u$ for some non-$//$-node $v\}$.

(T2) A node of T^* is labeled by **V** if and only if it belongs to V_G, by **P** if and only if it is a $//$-node of T and by **N** if and only if it is a non-$//$-node of T.

(T3) $E_{T^*} = E \cup E' \cup E_1^1 \cup E_2^1 \cup \cdots \cup E_1^p \cup E_2^p$ where:

   (T3.1) $E = \{u \to v \in E_T \mid u, v \in V_{T^*}$ and u is a $//$-node and v is a non-$//$-node$\}$. They are unlabeled.

   (T3.2) $E' = \{r \xrightarrow{1} s(G), r \xrightarrow{2} n(G) \mid r$ the root of $T\}$.

   (T3.3) For each $i \in [p]$, we have

      (T3.3.a) $E_1^i = \bigcup\limits_{v \in V_{T^*}} \{u_i \to z, z \xrightarrow{j} v \mid u_i \to v \in E_T$ corresponds to $t \to z \in R_i$ and $g_j(z) = t$ with $j \in \{1, 3, 5, 7\}\}$.

      (T3.3.b) $E_2^i = \bigcup\limits_{v \in V_{T^*}} \{u_i \to z, z \xrightarrow{j} v \mid u_i \to v \in E_T$ corresponds to $z \to t \in R_i$ and $g_j(z) = t$ with $j \in \{2, 4, 6, 8\}\}$.

Remark 9.2

1. Given a bipolar plane graph G, we construct the tree T^* in linear-time:

   Step 1. The decomposition tree T of G is constructed in linear-time (Proposition 9.4).

   Step 2. We construct the functions $g_1, \ldots, g_8 : V_{Int}(G) \to V_G$ by using Corollary 9.1 and this in linear-time because by Lemma 9.5 $(V_{Int}(R_i))_{1 \leq i \leq p}$ is a partition of V_G and by Proposition 9.4 we have $\biguplus_{1 \leq i \leq p} E_{R_i}$ is $O(n)$, n is the number of vertices of G.

   Step 3. The tree T^* is then constructed by using T and $g_1, \ldots, g_8$. And this is done in linear-time since we use the arcs of T.

2. From the tree T^* and the associated functions $g_1, \ldots, g_8$, one can almost reconstruct G, but not always exactly. For an example, if in the graph G on Figure 30 (Example 9.5), one deletes the arc f_{17}, the tree T^* and the functions g_i do not change. But, the decomposition tree on Figure 31 is modified. For another example without parallel edges, let $E = (\bullet \to \bullet \to \bullet)//(\bullet \to \bullet \to \bullet)//(\bullet \to \bullet)$ be a bipolar plane graph. Then the trees T^* associated with E and the bipolar plane graph $(\bullet \to \bullet \to \bullet)//(\bullet \to \bullet \to \bullet)$ are the same. Apart from the arcs between the vertices of a polar pair, the graph G can be reconstructed from T^* and $g_1, \ldots, g_8$.

3. We will see that the arcs which are not encoded by T^* play no role in the determination of the separation of vertices by polar pairs. However, we could encode $g_1, \ldots, g_8$ in T^* by additional edges, making it into a graph. This graph would not have bounded clique-width because one can recover G from it by an MS-definition scheme. Therefore, we would not be able to use Theorem 7.4 because MS-definition schemes preserves boundedness of clique-width [Cou97, EvO97] whereas 2-connected planar graphs do not have bounded clique-width.

We illustrate Definition 9.13 with Example 9.6.

Example 9.6 We use the graph of Example 9.5. The table below shows mappings $g_1, \ldots, g_5$. The mappings g_6, g_7, g_8 are everywhere undefined. The graphs $R_1, \ldots, R_5$ are shown on Figures 32 and the decomposition tree of G is shown on Figure 31.

	R_1			R_2		R_3	R_4	R_5
	a	b	c	d	k	m	p	q
g_1	s	s		s		a	a	c
g_2		n	n		n	b	c	b
g_3		a						
g_4	c			k				
g_5		c						

The tree T^* is shown on Figure 34. For each node labeled by $\mathbf{V}$, we indicate between parentheses the corresponding vertex of G for helping to understand the construction.
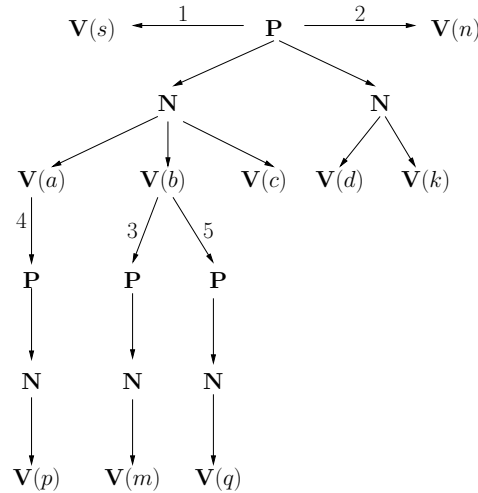
Figure 34: The tree T^* of the graph on Figure 30.

Before proving Proposition 9.5 we prove a last technical lemma on labeled trees T^* constructed in Definition 9.13. We let $ppsl(u_1, u_2, u_3, u_4)$, a property on labeled trees T^*, mean:

> u_1, u_2, u_3 are labeled by $\mathbf{V}$, u_4 is labeled by $\mathbf{P}$ or $\mathbf{N}$, $u_3 \leq_{T^*} u_4$ and $(u_1, u_2) = (s(u_4), n(u_4))$.

Claim 9.3 *Let G be a bipolar plane graph, let T^* be its labeled tree and let $\chi = \{u_1, u_2, u_3, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8, z_s, z_n\}$. Then there exists a formula $\psi(u_1, u_2, u_3, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8, z_s, z_n)$ in $MS_{\mathcal{T}_A}(\chi)$ such that for every $x, y, z \in V_G$ and every $u \in V_{T^*}$, $ppsl(x, y, z, u)$ holds if and only if:*

$$T^* \models \psi\Big(x, y, z, u, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y), s(G), n(G)\Big).$$

Proof of Claim 9.3. The condition "u_1, u_2, u_3 are labeled by $\mathbf{V}$, u_4 is labeled by $\mathbf{P}$ or $\mathbf{N}$, $u_3 \leq_{T^*} u_4$" is expressed by the following formula $\theta(u_1, u_2, u_3, u_4)$:

$(\mathbf{V}(u_1) \wedge \mathbf{V}(u_2) \wedge \mathbf{V}(u_3)) \wedge (\mathbf{N}(u_4) \vee \mathbf{P}(u_4)) \wedge$ ("there exists a directed path from u_4 to u_3").

The only difficulty is then to express the condition $(u_1, u_2) = (s(u_4), n(u_4))$. We distinguish several cases.

CASE 1. u_4 is the root or u_4 is a son of the root which is labeled by $\mathbf{P}$ (hence u_4 is labeled by $\mathbf{N}$). In this case $(u_1, u_2) = (s(u_4), n(u_4)) = (s(G), n(G))$. Then in this case for every $x, y, z \in V_G$ and $u \in V_{T^*}$, $ppsl(x, y, z, u)$ holds if and only if $T^* \models \theta(x, y, z, u) \wedge \psi_1(x, y, u, s(G), n(G))$ where $\psi_1(u_1, u_2, u_4, z_s, z_n)$ is:

$$\Big(\text{``} u_4 \text{ is the root''} \vee \text{``the father of } u_4 \text{ is the root labeled by } \mathbf{P} \text{''} \Big) \wedge (u_1 = z_s \wedge u_2 = z_n).$$

CASE 2. u_4 is not the root and is labeled by $\mathbf{P}$; hence it is not a son of the root (by the way T^* is constructed). Its father u is labeled by $\mathbf{V}$, hence is a vertex of G and u is one of the two poles of $G(u_4)$. Let $j \in [8]$ be the label of the arc $u \rightarrow u_4$. Then the other pole of $G(u_4)$ is $g_j(u)$. It follows in this case that, for every $x, y \in V_G$ and $u \in V_{T^*}$, we have $(x, y) = (s(u), n(u))$ if and only if $T^* \models \theta'(x, y, u, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y))$ where $\theta'(u_1, u_2, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8)$ is the formula:

$$\left(\text{``} u_1 \text{ is the father of } u_4 \text{''} \wedge \bigvee_{j=2,4,6,8} u_2 = x_j \right) \vee \left( \text{``} u_2 \text{ is the father of } u_4 \text{''} \wedge \bigvee_{j=1,3,5,7} u_1 = y_j \right).$$

Therefore, in this case for every $x, y, z \in V_G$ and $u \in V_{T^*}$, $ppsl(x, y, z, u)$ holds if and only if $T^* \models \theta(x, y, z, u) \wedge \psi_2(x, y, u, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y))$ where $\psi_2(u_1, u_2, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8)$ is:

$$(\text{``} u_4 \text{ is not the root''} \wedge \mathbf{P}(u_4)) \wedge \theta'(u_1, u_2, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8).$$

CASE 3. u_4 is not the root, is labeled by $\mathbf{N}$, hence its father u'' is labeled by $\mathbf{P}$ and is not the root otherwise Case 1 applies. The father u' of u'' is labeled by $\mathbf{V}$. We have $(s(u_4), n(u_4)) = (s(u''), n(u''))$ and $u' \in \{s(u_4), n(u_4)\}$ as in Case 2. It follows in this case that for every $x, y \in V_G$ and $u \in V_{T^*}$, we have $(x, y) = (s(u), n(u))$ if and only if $T^* \models \theta''(x, y, u, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y))$ where $\theta''(u_1, u_2, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8)$ is the formula:

$$\left(\text{``} u_1 \text{ is the grand-father of } u_4 \text{''} \wedge \bigvee_{j=2,4,6,8} u_2 = x_j \right) \vee \left( \text{``} u_2 \text{ is the grand-father of } u_4 \text{''} \wedge \bigvee_{j=1,3,5,7} u_1 = y_j \right).$$

Therefore, in this case for every $x, y, z \in V_G$ and $u \in V_{T^*}$, $ppsl(x, y, z, u)$ holds if and only if $T^* \models \theta(x, y, z, u) \wedge \psi_3(x, y, u, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y))$ where $\psi_2(u_1, u_2, u_4, g_1(u_1), \ldots, g_8(u_1), g_1(u_2), \ldots, g_8(u_2))$ is:

$(\text{``} u_4 \text{ is not the root''} \wedge \mathbf{N}(u_4) \wedge \text{``the father of } u_4 \text{ is not the root''}) \wedge \theta''(u_1, u_2, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8).$

The 3 cases above complete the condition $(u_1, u_2) = (s(u_4), n(u_4)$. Therefore, for every $x, y, z \in V_G$ and every $u \in V_{T^*}$, $ppsl(x, y, z, u)$ holds if and only if $T^* \models \psi(x, y, z, u, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y), s(G), n(G)$ where $\psi(u_1, u_2, u_3, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8, z_s, z_n)$ is the formula:

$$\theta \wedge (\psi_1 \vee \psi_2 \vee \psi_3).$$

This finishes the proof of the claim. ∎

We can now prove Proposition 9.5.

Proof of Proposition 9.5. Let G be a bipolar plane graph with n vertices and let T be its decomposition tree. Let $T^* \in \mathcal{T}(A)$ be the labeled tree constructed from T and G where $A = \{\mathbf{V}, \mathbf{P}, \mathbf{N}, 1 \ldots, 8\}$. Let $\chi = \{u_1, u_2, u_3, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8, z_s, z_n\}$.

For every $x, y, z, t \in V_G$, $pps(x, y, z, t)$ holds if and only if the following holds:

$$\exists u \Big(\big(ppsl(x, y, z, u) \wedge \mathbf{V}(t) \wedge t \nless_T u \big) \vee \big(ppsl(x, y, t, u) \wedge \mathbf{V}(z) \wedge z \nless_T u \big) \Big)$$

It follows from Claim 9.3 that one can build a formula $\varphi(u_1, u_2, u_3, u_4, x_1, \ldots, x_8, y_1, \ldots, y_8, z_s, z_n)$ in $MS_{\mathcal{T}_A}(\chi)$ such that $pps(x, y, z, t)$ holds if and only if

$$T^* \models \varphi \Big(x, y, z, t, g_1(x), \ldots, g_8(x), g_1(y), \ldots, g_8(y), s(G), n(G) \Big). \tag{10}$$

By Theorem 7.4, we can construct a log-labeling L for φ on T^*. Recall that the vertices of G are nodes of T^*. For every $x \in V_G$, we let

$$J(x) = (L(x), L(g_1(x)), \ldots, L(g_8(x)), L(s(G)), L(n(G))).$$

We have clearly $|J(x)| = O(\log(n))$ and by Equivalence (10), we can determine for every $x, y, z, t \in V_G$ if (z, t) is a polar pair separating x and y by using $J(x), J(y), J(z)$ and $J(t)$. This ends the proof.   ■

9.5.2   Reduced Barriers and the Proof of Theorem 9.4

We have seen with Proposition 9.5 that we can label the vertices of a bipolar plane graph G with labels of size $O(\log(|V_G|))$ and such that for every $X \subseteq V_G$ and every x and y in V_G, we can check whether x and y are separated by a polar pair $(z, t) \in X \times X$. We will see in this section that if x and y are not separated by a polar pair $(z, t) \in X \times X$, then their connectivity is reduced to the use of geometrical tools. For that purpose we need some more definitions and a proposition which extends Proposition 9.1.

Let G be a bipolar plane graph and let T be its decomposition tree. For every //-polar pair (z, t), we let $Select(z, t) = F_1(u)$, called the *Select function*, where u is the //-node such that $(z, t) = (s(u), n(u))$ ($F_1(u)$ is defined in Definition 9.11). We notice that any other face $F_j(u)$ would work for $j \leq d(u) - 1$.

Definition 9.14 (Reduced Barriers for Bipolar Plane Graphs) Let G be a bipolar plane graph with adjacent poles and let G^+ be its augmented graph. For every two distinct vertices x and y, we let

$$RBar(\{x, y\}) = \begin{cases} Bar(\{x, y\}) & \text{if } x \text{ and } y \text{ are incident with at most 2 faces,} \\ \{fx, fy \mid f = Select(x, y)\} & \text{otherwise.} \end{cases}$$

For every $X \subseteq V_G$, we let $RBar(X) = \bigcup \{RBar(\{x, y\}) \mid x, y \in X, \ x \neq y\}$ and we call it the *reduced barrier of X*.

If $\mathcal{E}^+ = (p, s)$ is an embedding of G^+, we let $RBar(X, \mathcal{E}^+)$ be the set $\{s(e) \mid e \in RBar(X)\}$.

If two distinct vertices of G, say z and t, are incident with at least 3 faces, then they form a //-polar pair, (z,t) or (t,z) by Lemma 9.6. Therefore, for every 2 distinct vertices z and t in V_G, $RBar(\{z,t\})$ is well-defined. We now prove the following which relates the connectivity of 2 vertices in $G \backslash X$ with $RBar(X)$.

Proposition 9.6 *Let G be a bipolar plane graph with adjacent poles and let $\mathcal{E}^+ = (p,s)$ be an embedding of G^+. Let $X \subseteq V_G$ and let $x, y \in V_G - X$, $x \neq y$. Then x and y are separated by X if and only if either:*

(a) x and y are separated by a polar pair $(z,t) \in X \times X$ or

(b) $p(x)$ and $p(y)$ are separated in the plane by $RBar(X, \mathcal{E}^+)$.

Proof. Let G, X, x, y be as in the statement. If (a) or (b) holds, then x and y are separated by X (for the second case, we observe that $RBar(X, \mathcal{E}^+) \subseteq Bar(X, \mathcal{E}^+)$ and we use Proposition 9.1).

Let us conversely assume that x and y are separated by X, but (a) does not hold. By Proposition 9.1, they are separated in the plane by $Bar(X, \mathcal{E}^+)$. As in the proof of Proposition 9.1 we need only prove the result for a minimal separator $Y \subseteq X$ of x and y, because if x and y are separated by $RBar(Y, \mathcal{E}^+)$, they are also by $RBar(X, \mathcal{E}^+)$. Hence, we assume that $X = \{x_1, \ldots, x_m\}$ is a minimal separator of x and y in G. We first assume that $m \geq 3$. Then $Bar(X)$ has the structure shown on Figure 35, where, for each $i \in [m]$, $\{f_{i,1}, \ldots, f_{i,p_i}\}$ is the set of faces incident with x_i and x_{i+1} (letting x_{m+1} denote also x_1).
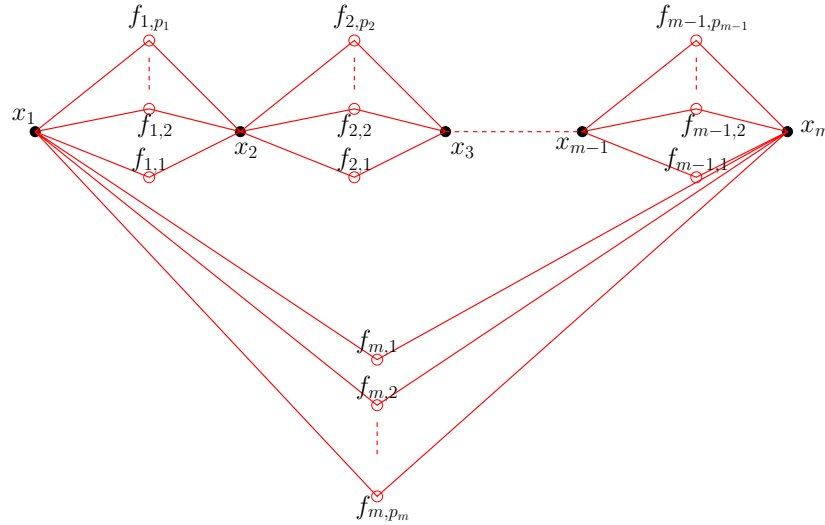
Figure 35: A barrier illustrating the proof of Proposition 9.6.

Then $RBar(X)$ is obtained from $Bar(X)$ by removing for each i such that $p_i \geq 3$ all vertices $f_{i,j}$ (and the incident edges) but one, so that $RBar(X)$ contains a cycle going through $x_1, \ldots, x_m$. If x and y are separated by $Bar(X, \mathcal{E}^+)$ and not by $RBar(X, \mathcal{E}^+)$, this means that one and only one of them is inside a cycle $x_i - f_{i,j} - x_{i+1} - f_{i,j+1} - x_i$ of $Bar(X)$ such that $f_{i,j}$ or $f_{i,j+1}$ (or both) has been removed. This implies that $p_i \geq 3$ hence that x_i and

x_{i+1} form a //-polar pair (by Lemma 9.6). Furthermore, the set of vertices that are inside this cycle are the internal vertices of $G(u_j)$ where u_j is the j-th son of the //-node u with poles x_i and x_{i+1}. Hence, x and y are separated by the polar pair (x_i, x_{i+1}) or (x_{i+1}, x_i), i.e., (a) holds. But we assumed the contrary. Hence, (b) must hold.

If $m = 2$ and $p_1 = p_2 = 1$, then $Bar(X) = RBar(X)$ hence (b) holds. If $p_1 + p_2 \geq 3$, then by Lemma 9.6, x_1 and x_2 form a polar pair. As for the case $m \geq 3$ we get that x and y are separated by $RBar(X, \mathcal{E}^+)$, otherwise (a) holds.

We cannot have $m = 1$ because the graph is assumed 2-connected. This ends the proof. ∎

The following example is an illustration of Definition 9.14 and Proposition 9.6.

Example 9.7 For clarity on Figure 36 we number faces from 1 to 8 but we do not show the edges of G^+ incident with the face-vertices $1, \ldots, 8$. The set $Bar(\{z, t\})$ contains the 4 paths $z - i - t$ for $i = 2, 6, 7, 8$. Note that (z, t) is a //-polar pair. The reduced barrier $RBar(\{z, t\})$ contains only one of them, say $z - 2 - t$. The set $RBar(\{z, t, c\})$ contains then $z - 2 - t$, $t - 3 - c$, $t - 4 - c$, $c - 5 - z$. This reduced barrier separates b and d. The edges $z - 2$ and $2 - t$ are useful for that because without them b and d are not separated. $RBar(\{a, z\}) = Bar(\{a, z\}) = \{z - 2, 2 - a\}$ and the graph $G \backslash \{a, z\}$ is connected. Note that a and z do not form a polar pair.
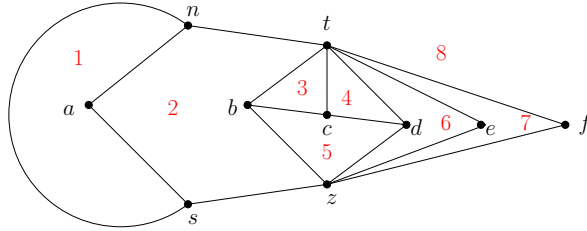
Figure 36: The graph of Example 9.7.

We now prove a last technical lemma. We let $pp(x_1, x_2)$, called the *polar-pair property*, be the property on bipolar plane graphs that expresses that the pair of vertices (x_1, x_2) is a //-polar pair.

Lemma 9.8 *The polar-pair property and the Select function in bipolar plane graphs are representable by 12 functions.*

Proof. The proof is a variant of that of Proposition 9.2. Let G be a bipolar plane graph. We let H be the simple directed graph with $V_H = V_G$ and an arc $x \to y$ if and only if (x, y) is a //-polar pair. It is planar because these arcs can be inserted without crossing in a planar embedding of G by adding this arc in a face $F_j(u)$, $j \leq d(u)$ where u is the node of T such that $(x, y) = (s(u), n(u))$. By Corollary 9.1, we can represent the directions of arcs in H with 6 functions $g_{iH}^+, g_{iH}^- : V_H \to V_H$ for $i \in [3]$ such that:

$$g_{iH}^+(x) = y \qquad \text{implies} \qquad x \to y,$$
$$g_{iH}^-(x) = y \qquad \text{implies} \qquad y \to x.$$

Each arc is represented by a unique such clause. Hence, with the 6 functions g_{iH}^α, $i \in [3]$, $\alpha \in \{+, -\}$, we can represent the polar-pair property pp.

We now define 6 partial functions h_{iG}^α for $i \in [3], \alpha \in \{+, -\}$ as follows:

$$h_{iG}^+(x) = Select(x, g_{iG}^+(x)),$$
$$h_{iG}^-(x) = Select(g_{iH}^-(x), x).$$

Then for every 2 distinct vertices $x, y \in V_G$, the face-vertex $f = Select(x, y)$ is characterized by the following formula:

$$\big(y = g_i^+(x) \wedge f = h_i^+(x)\big) \ \vee \ \big(x = g_i^-(y) \wedge f = h_i^-(y)\big).$$

Then we can represent the Select function $Select$ by the 12 functions g_i^α, h_i^α, $i \in [3]$, $\alpha \in \{+, -\}$. ∎

We can now prove Theorem 9.4.

Proof of Theorem 9.4. Let G be a 2-connected planar graph with n vertices and let G^+ be its augmented graph. By Remark 9.2 (1), we can construct the labeled tree T in $O(n)$-time. By Proposition 9.5, we can construct in $O(n \cdot \log(n))$-time a log-labeling K for the polar-pair separation query on G. By Proposition 9.3, we can construct a straight-line embedding $\mathcal{E}^+ = (p, s)$ of G^+ with coordinates in $[3n - 6]$. By Proposition 9.2 and Remark 9.1, we can represent with $24 = 15 + 3 \cdot 3$ functions $f_1, \ldots, f_{24}$ the property that 2 vertices are incident with at most 2 faces. By Lemma 9.8, we can represent with 12 functions $g_1, \ldots, g_{12}$ the polar-pair property and the Select function $Select$. Hence, for every vertex x in V_G, we let

$$J(x) = \big(p(x), p(f_1(x)), \ldots, p(f_{24}(x)), p(g_1(x)), \ldots, p(g_{12}(x)), K(x)\big).$$

It is clear that the labeling J is constructed in $O(n \cdot \log(n))$-time and for every $x \in V_G$ we have $|J(x)| = O(\log(n))$. We now explain how to decide the connectivity of 2 distinct vertices $x, y \in V_G$ in $G \backslash X$ where $X \subseteq V_G - \{x, y\}$ by using the labels.

Step 1. For every pair of vertices $(z, t) \in X \times X$, we can verify by using $p(g_1(z)), \ldots, p(g_{12}(z))$ and $p(g_1(t)), \ldots, p(g_{12}(t))$ if (z, t) or (t, z) forms a //-polar pair by Lemma 9.8 and if so, decides if it separates x and y by using $K(x), K(y), K(z)$ and $K(t)$ by Proposition 9.5. If x and y are separated by a //-polar pair $(z, t) \in X \times X$, then we can report that they are separated by X by Proposition 9.6. Otherwise, we perform Step 2.

Step 2. We can decide if $(z, t) \in X \times X$ are incident with at most 2 faces by Proposition 9.2 and Remark 9.1 and if so, determine the at most 2 faces by using $p(f_1(z)), \ldots, p(f_{24}(z))$ and $p(f_1(t)), \ldots, p(f_{24}(t))$. By Proposition 9.2, if they are incident with at least 3 faces we can recover $Select(z, t)$ by using $p(g_1(z)), \ldots, p(g_{12}(z))$ and $p(g_1(t)), \ldots, p(g_{12}(t))$ (see Lemma 9.8). Since G^+ is a straight-line embedding, we can determine $RBar(X, \mathcal{E}^+)$. Finally, we can test if x and y are separated by $RBar(X, \mathcal{E}^+)$ by using Theorem 9.3 and this is sufficient by Proposition 9.6. This gives the final answer.

Given a $//$-polar pair (z, t), we can answer if they separate x and y in constant-time (Proposition 9.5); therefore it takes $O(|X^2|)$-time to decide if x and y are separated by a $//$-polar pair in X in Step 1. In Step 2, for every pair $(z, t) \in X \times X$ we take constant-time to construct $RBar(\{z, t\})$; therefore it takes $O(|X^2|)$-time to construct $RBar(X, \mathcal{E}^+)$. By Theorem 9.3, it takes $O(m \cdot \log(m))$-expected time to decide if $p(x)$ and $p(y)$ are separated by $RBar(X, \mathcal{E}^+)$ where $m = RBar(X, \mathcal{E}^+) = O(|X|)$. This finishes the proof. ∎

9.6 The General Case

We prove in this section Theorem 9.1 and we will use for that the decomposition of connected graphs into biconnected components. We recall that we denote by $Bcc(G)$ the set of biconnected components of a connected graph G.

Definition 9.15 (The Rooted Tree $BC(G)$ of G) Let G be a connected graph. We let $W(G)$, disjoint from V_G, be a set in bijection with $Bcc(G)$ by $bcc : W(G) \to Bcc(G)$. We denote by $BC(G)$ the tree with set of nodes $V_G \cup W(G)$ and with set of edges $\{wv \mid v \in V_G, w \in W(G) \text{ and } v \in V_{bcc(w)}\}$.

We choose a vertex r of G, that belongs to a unique biconnected component, to be the root of $BC(G)$. From this choice, $BC(G)$ is directed, rooted with partial order $\leq_{BC(G)}$ and r is the greatest element. The tree $BC(G)$ is called the *rooted tree* of G and handled as the relational structure $\langle V(G) \cup W(G), \mathbf{V}, \mathbf{W}, member, root \rangle$ where for every $v, w \in V_{Bc(G)}$,

$$
\begin{array}{rll}
\mathbf{V}(v) \text{ holds} & \text{if and only if} & v \in V_G \\
\mathbf{W}(v) \text{ holds} & \text{if and only if} & v \in W(G) \\
member(v, w) \text{ holds} & \text{if and only if} & v \in V_G, \ w \in W(G) \text{ and } v \in V_{bcc(w)} \\
root(v) \text{ holds} & \text{if and only if} & v \in V_G \text{ and is the root of } BC(G).
\end{array}
$$

For each $C \in Bcc(G)$, the set V_C has a $\leq_{BC(G)}$-greatest element, called the *leader of C* and denoted by $leader(C)$. For each vertex x of G different from r, we denote by $mother(x)$ and call it the *mother of x*, the $\leq_{BC(G)}$-maximal node w of $W(G)$ such that $x \in V_{bcc(w)}$.

A vertex of G has degree at least 2 in $BC(G)$ if and only if it is a separating vertex in G.

Definition 9.16 (Problematic Biconnected Components) Let G be a connected graph and let $BC(G)$ be its rooted tree. The unique path in $BC(G)$ between 2 distinct vertices x and y in V_G is denoted by $p(x, y)$. For a subset X of V_G and two vertices x and y in $V_G - X$, we say that a biconnected component C of G is *problematic* for (x, y, X) if $bcc^{-1}(C)$ is on the path $p(x, y)$ and C contains at least 2 vertices of X.

The following lemma is clear from the definition of problematic biconnected components.

Lemma 9.9 *Let G be a connected graph and let $x, y \in V_G$ and $X \subseteq V_G - \{x, y\}$. We denote by $C_1, \ldots, C_m$ the problematic biconnected components for (x, y, X) and for every $1 \leq i \leq m-1$, we let x_i be a vertex between $bcc^{-1}(C_i)$ and $bcc^{-1}(C_{i+1})$ in $p(x, y)$; we let $x_0 = x$ and $x_m = y$. The vertices x and y are separated by X if and only if either:*

(a) $p(x, y)$ goes through some $z \in X$ or,

(b) there exists an $1 \leq i \leq m$ such that the vertices x_{i-1} and x_i are separated by $X \cap V_{C_i}$ in G.

As for the proof of Theorem 9.4 we will use Theorem 7.4 in order to identify the separating vertices and to determine the problematic biconnected components. We let the two following properties of the nodes of $BC(G)$ that are MS-definable:

$$\varphi_1(u_1, u_2, u_3) = \bigwedge_{1 \leq i < j \leq 3} \text{``} u_i \neq u_j \text{''} \wedge \bigwedge_{1 \leq i \leq 3} \mathbf{V}(u_i) \wedge \text{``} p(u_1, u_2) \text{ goes through } u_3 \text{''}.$$

$$\varphi_2(u_1, u_2, u_3, u_4, u_5) = \bigwedge_{1 \leq i < j \leq 5} \text{``} u_i \neq u_j \text{''} \wedge \bigwedge_{i=1,2,4,5} \mathbf{V}(u_i) \wedge \mathbf{W}(u_3) \wedge \bigwedge_{i=4,5} member(u_i, u_3) \wedge$$
$$\text{``} p(u_1, u_2) \text{ goes through } u_3 \text{''}.$$

For every $x, y, z \in V_G$, the property $\varphi_1(x, y, z)$ holds if and only if $p(x, y)$ goes through z. And for every $x, y, z, t \in V_G$ and every $u \in W(G)$, the property $\varphi_2(x, y, u, z, t)$ holds if and only $bcc(u)$ is a problematic biconnected component for $(x, y, \{z, t\})$. Our next aim is to prove the following proposition, stated with the notation of Definition 9.15 and Lemma 9.9.

Proposition 9.7 *Let G be a connected planar graph with n vertices. There exists a log-labeling M_0 for the properties φ_1, φ_2, $\leq$ and member on G. One can also build a log-labeling M of G such that for every $x, y \in V_G$ and $X \subseteq V_G$:*

1. *we can determine from $M(x)$, $M(y)$ and $M(X)$ whether $p(x, y)$ goes through some $z \in X$. Otherwise,*

2. *if $C_1, \ldots, C_m$ are the problematic biconnected components for (x, y, X), then we can determine in $O(|X^2|)$-time from $M(x)$, $M(y)$ and $M(X)$ the label $M_0(bcc^{-1}(C_i))$, the sets $M_0(X \cap V_{C_i})$ for i in $[m]$ and the labels $M_0(x_1), \ldots, M_0(x_{m-1})$ where the vertices $x_1, \ldots, x_{m-1}$ are leaders of some of the problematic biconnected components such that $(x_0 = x, \ x_m = y)$:*

$$Conn(x, y, X) \Longleftrightarrow \bigwedge_{1 \leq i \leq m} Conn(x_{i-1}, x_i, X \cap V_{C_i}).$$

Moreover, the labelings M and M_0 are constructed in $O(n \cdot \log(n))$-time.

Proof. The order $\leq_{BC(G)}$ is definable by an MS formula since we can define by an MS formula the existence of a directed path between two nodes of a rooted forest.

Notice that $|V_{BC(G)}| \leq 2|V_G|$. By Theorem 7.4, we can construct a log-labeling K_i of $BC(G)$ for each property φ_i, $1 \leq i \leq 2$. We can also construct log-labelings K_3 and K_4 for the properties $u_1 \leq u_2$ and $member(u_1, u_2)$ respectively. For each node u in $V_{BC(G)}$, we let $M_0(u) = (K_1(u), K_2(u), K_3(u), K_4(u))$ and for each vertex x in V_G, we let

$$M(x) = \Big(M_0(x), M_0(mother(x)), M_0(leader(mother(x))) \Big).$$

(If x is the root we mark the last two components as "undefined"). We now explain given x and y in V_G and a subset X of $V_G - \{x, y\}$, how to check the statements by using $M(x)$, $M(y)$ and $M(X)$.

For each $z \in X$, by using the K_1-parts of $M_0(x), M_0(y)$, and $M_0(z)$, we can check if $\varphi_1(x, y, z)$ holds. Hence, we can check whether $p(x, y)$ goes through some vertex in X. This test takes time $O(|X|)$.

Otherwise, let $C_1, \ldots, C_m$ be the problematic components for (x, y, X). The path $p(x, y)$ can be of 3 possible types depending on how its nodes are related under $\leq_{B(G)}$; we denote below $<_{B(G)}$ by $<$ for readability.

CASE 1. $x < C_1 < C_2 < \cdots < C_m < y$ or the same by changing $<$ into $>$,

CASE 2. $x < C_1 < C_2 < \cdots < C_p > C_{p+1} \cdots > C_m > y$,

CASE 3. $x < C_1 < C_2 < \cdots < C_p < w > C_{p+1} \cdots > C_m > y$ where w is either a vertex or a biconnected component that is not problematic.

In Case 1 we let x_i be the leader of C_i for $i = 1, \ldots, m - 1$ (in the other variant of Case 1 we interchange x and y). For Case 2, we let x_i be the leader of C_i for $i = 1, \ldots, p - 1$ and we let x_i be the leader of C_{i+1} for $i = p, \ldots, m - 1$. For Case 3, we let x_i be the leader of C_i for $i = 1, \ldots, p$ and we let x_i be the leader of C_{i+1} for $i = p + 1, \ldots, m - 1$.

If a biconnected component C is a problematic biconnected component, it must be the mother of some vertex $z \in X$. Therefore, for each pair $(z, t) \in X \times X$, by using the K_2-parts of $M_0(x), M_0(y), M_0(z), M_0(t)$ and $M_0(mother(z))$ (resp. $M_0(mother(t))$), we can decide if $bcc(mother(z))$ (resp. $bcc(mother(t))$) is a problematic biconnected component for (x, y, X). We can then determine the set $Z = \{M_0(bcc^{-1}(C_i)) \mid i = 1, \ldots, m\}$.

By using the K_3-parts of $M_0(x)$ and $M_0(y)$, we can check whether $x < y$ or $y < x$, hence we can identify if Case 1 holds. It remains to identify Cases 2 and 3. By using the K_3-parts of $M_0(x), M_0(y)$ and the set Z we can check whether Case 2 or 3 holds. If Case 2 or Case 3 holds, we can determine C_p. (Notice that with Case 3 we cannot determine w.)

We now show how to determine, for each problematic biconnected component C_i, the label $M_0(x_i)$. Since each component C_i is problematic we know at least one $z \in X \cap V_{C_i}$ such that $C_i = bcc(mother(z))$. Since $M(z)$ contains $M_0(leader(mother(z)))$ for each $z \in X$ we can determine the leaders of the problematic components, whence the desired lists $x_1, \ldots, x_{m-1}$ and $M_0(x_1), \ldots, M_0(x_{m-1})$.

It remains to determine for each problematic biconnected component C_i, the set $M_0(X \cap V_{C_i})$. By definition, if $C_i = bcc(mother(z))$ for some $z \in X$, then $X \cap V_{C_i}$ is the set of elements $t \in X$ such that $member(t, mother(z))$. By using the K_4-parts of $M_0(t)$ and of $M_0(mother(z))$, we can determine when $member(t, mother(z))$ does hold. This finishes the proof. ∎

Remark 9.3 With Proposition 9.7 we only have the labels $M_0(x_i)$ of the x_i's. However, we want to use Proposition 9.7 in such a way that it gives us other labels (typically some indices in the plane, to be defined later) of the x_i's and not only their labels M_0. For that purposes, we use the following notation. If $J : V_G \to L$ is an injective mapping, then we let for every x in V_G

$$M[J](x) = \Big(J(x), M_0(x), M_0(mother(x)), M_0(leader(mother(x))), J(leader(mother(x))) \Big).$$

It is clear that by using the labeling $M[J]$ instead of M we determine clearly the labels $(M_0(x_i), J(x_i))$ of the x_i's and if, for every x, we have $|J(x)| \leq O(f(|V_G|))$ for some function f, then $|M[J](x)| \leq O(\log(|V_G|) + f(|V_G|))$.

We now explain how to label the vertices so that we can verify $Conn(x_{i-1}, x_i, X \cap V_{C_i})$ for each i. We use the same notations as in Proposition 9.7. It is clear that $Conn(x_{i-1}, x_i, X \cap V_{C_i})$ holds if and only if $Conn(x'_{i-1}, x'_i, X \cap V_{C_i})$ holds where x'_{i-1}, denoted by $Att_G(x_{i-1}, C)$, is the first vertex of C met by $p(x_{i-1}, bcc^{-1}(C))$ in $BC(G)$ and similarly for $x'_i = Att_G(x_i, C)$. Note that for each vertex x_i the vertex $Att_G(x_i, C)$ is a separating vertex of G. Then we need for each x and a biconnected component C of G, to be able to determine the vertex $Att_G(x, C)$. One way for doing that is to store the set of separating vertices of each biconnected component C and given x, test if one of the separating vertices of C is equal to $Att_G(x, C)$. However, a biconnected component may have an unbounded number of separating vertices and we cannot encode them as a finite list of bounded size. To overcome this difficulty, we take the rooted tree and fuse the labeled trees $T^*(C)$ of all the biconnected components of G. Then we distinguish the two cases of Proposition 9.6:

1. we use logical tools in order to determine the case where $Att_G(x_{i-1}, C)$ and $Att_G(x_i, C)$ are separated by a polar pair in $X_i \times X_i$ where $X_i = X \cap V_{C_i}$. If not,

2. we prove that x_{i-1} and x_i are separated by $RBar(X \cap V_{C_i})$ if and only if they are separated by $X \cap V_{C_i}$.

Therefore, we no more need to determine the vertices $Att_G(x, C)$. We need now definitions and notations. Let us first show how to combine the trees $T^*(C)$.

Definition 9.17 (The Representing Tree $BT^*(G)$) Let G be a connected planar graph and let $BC(G)$ be its rooted tree. For each biconnected component C of G we let $n(C)$ be $leader(bcc^{-1}(C))$ and choose a vertex of C adjacent with $n(C)$, that we denote by $s(C)$. Each bi-connected component C of G is transformed into a bipolar plane graph with North pole $n(C)$ and South pole $s(C)$.

For each $C \in Bcc(G)$ we let $T^*(C)$ be the labeled tree associated to the decomposition tree of the bipolar plane graph associated with C. If C is reduced to a single edge $s(G) \to n(G)$, we let $T^*(C)$ be the tree $s(G) \xleftarrow{1} \mathbf{N} \xrightarrow{2} n(G)$.

We define $BT^*(G)$, called the *representing tree* of G, as the union of the trees $T^*(C)$ for all $C \in Bcc(G)$. These trees have in common the nodes that are vertices of G. We let $Root(C)$ be the root of $T^*(C)$. It is not in V_G.

The tree $BT^*(G)$ depends not only on the chosen orientation of $BC(G)$ but also on the bipolar orientations of the biconnected components. We illustrate Definition 9.17 with Example 9.8.

Example 9.8 Let W be the directed plane graph of Figure 37. Its biconnected components are bipolar. Letting g_3 map 4 to 14 (note in our example no other value of g_3 and no other function $g_4, \ldots, g_8$ are needed), its tree $BT^*(G)$ is shown on Figure 38.

Figure 37: A directed plane graph W.

Figure 38: The tree $BT^*(G)$ of the graph on Figure 37.

We state the following simple properties of $BT^*(G)$.

Lemma 9.10 *Let G be a planar graph and let $BT^*(G)$ be its representing tree. Then:*

(i) The graph $BT^*(G)$ is a directed tree.

(ii) The nodes of $BT^*(G)$ labeled by $\mathbf{V}$ are the vertices of G.

(iii) The nodes of $BT^*(G)$ of indegree 0 are in bijection by a function, that we will denote by Root, with $Bcc(G)$ and thus with $W(G)$.

(iv) *For each $C \in Bcc(G)$, $n(C)$ is the unique vertex x such that $Root(C) \xrightarrow{2} x$ in $BT^*(G)$
     and $s(C)$ is the unique vertex y such that $Root(C) \xrightarrow{1} y$. The nodes of $T^*(C)$ are the
     nodes of $BT^*(G)$ accessible from $Root(C)$ by a directed path and $T^*(C)$ is the sub-tree
     of $BT^*(G)$ induced on this set.*

(v) *The tree $BT^*(G)$ has $O(n)$ nodes.*

Proof. All these facts are clear from the definitions. ∎

We now define formally the vertex $Att_G(x, C)$ for every vertex x of G and every biconnected
component C of G.

Definition 9.18 (Attachment Vertices) For every vertex x of G and every biconnected com-
ponent C of G, we let

$$Att_G(x, C) = \begin{cases} x & \text{if } x \in C, \\ x' & \text{if } x \notin C \text{ and } x' \text{ is the unique vertex of } C \\ & \text{on the path in } BC(G) \text{ that links } x \text{ and } bcc^{-1}(C). \end{cases}$$

In other words $Att_G(x, C)$ is the first vertex of C on any path in G from x to some vertex
of C. We omit the sub-script when non necessary. We first need to be able to state in $BT^*(G)$
that a vertex x' is $Att_G(x, C)$. For that purposes, we prove that the rooted tree $BC(G)$ of G
can be defined from $BT^*(G)$ by an MS-definition scheme (MS-definition schemes are defined
in Section 1.4).

Lemma 9.11 *For every connected planar graph G the rooted tree $BC(G)$ of G can be defined
from the representing tree $BT^*(G)$ of G by an MS-definition scheme.*

Proof. Let $BT^*(G)$ be the representing tree of G of a connected planar graph. Let $\mathcal{D} = (\psi, \theta_{\mathbf{V}}, \theta_{\mathbf{W}}, \theta_{member}, \theta_{root}, \theta_{\leq})$ be the MS-definition scheme where:

$$\psi(u) = \text{``}u \text{ has indegree 0''} \ \vee \mathbf{V}(u)$$
$$\theta_{\mathbf{V}}(u) = \mathbf{V}(u)$$
$$\theta_{\mathbf{W}}(u) = \text{``}u \text{ has indegree 0''}$$
$$\theta_{root}(u) = \text{``maximal element of the reflexive and transitive closure of } <_0\text{''}$$
$$\theta_{member}(u_1, u_2) = \text{``}u_2 \text{ has indegree 0''} \ \wedge \mathbf{V}(u_1)$$
$$\wedge \text{``there exists a directed path from } u_2 \text{ to } u_1$$
$$\theta_{\leq}(u_1, u_2) = u_1 = u_2 \ \vee \ u_1 <_0^* u_2$$

where for all nodes u, v of $BT^*(G)$,

$$u <_0 v \quad \text{if and only if} \quad \left(\theta_{\mathbf{W}}(u) \wedge \ \theta_{\mathbf{V}}(v) \wedge u \xrightarrow{2} v \right) \ \vee (member(u, v) \ \wedge \neg(v \xrightarrow{2} u)).$$

It is clear that $\mathcal{D}$ is an MS-definition scheme that defines $BC(G)$. ∎

Lemma 9.11 combined with Proposition 1.3 says that every MS formula φ on $BC(G)$ can be translated into an MS formula $\varphi^{\#}$ on $BT^*(G)$ such that φ holds in $BC(G)$ if and only if $\varphi^{\#}$ holds in $BT^*(G)$. In particular, we get the following.

Corollary 9.2 *The properties $\varphi_1, \varphi_2, \leq$ and member on rooted trees can be translated into MS queries on representing trees, that we denote by $\varphi_1^{\#}, \varphi_2^{\#}, \leq^{\#}$ and $member^{\#}$.*

We now prove that $x' = Att_G(x, C)$ is definable by an MS formula on $BT^*(G)$.

Lemma 9.12 *Let G be a connected planar graph and let $BT^*(G)$ be the representing tree of G. There exists an MS formula $\alpha(u_1, u_2, u_3)$ relative to $BT^*(G)$ such that for any nodes u, u' and w of $BT^*(G)$, we have $BT^*(G) \models \alpha(u, u', w)$ if and only if u, u' are vertices of G, $w = Root(C)$ for some biconnected component C of G and $u' = Att_G(u, C)$.*

Proof. We let $\alpha(u_1, u_2, u_3)$ express the following: u_1 and u_2 are labeled by $\mathbf{V}$, w has in-degree 0, there is a directed path from w to u_2 and either $u_1 = u_2$ or there is an undirected path between u_1 and u_2 containing an arc $y \to u_2$ that does not belong to the path from w to u_2. From the definition of $Att_G(x, C)$ these conditions are clearly equivalent to the condition $u_2 = Att_G(u_1, C)$ where $u_3 = Root(C)$.   ∎

Example 9.9 We consider the tree on Figure 38. The nodes marked by $I, \ldots, VII$ are those of the form $Root(C)$. We have in particular $10 = Att(2, C) = Att(6, C) = Att(5, C) = Att(10, C)$ where VII$=Root(C)$. The validity of the definition of α can be checked on these examples.

We now prove that the property "$x' = Att_G(x, C)$ and $y' = Att_G(y, C)$ are separated by a polar pair in V_C" is definable by an MS formula in $BT^*(G)$. For nodes u_1, u_2, u_3, u_4 and u_5 we let $pps_1(u_1, u_2, u_3, u_4, u_5)$ mean:

> "$u_5 = Root(C)$ for some biconnected component C and (u_1, u_2) is a polar pair of C separating u_3 and u_4."

We let $pps'(u_1, u_2, u_5)$ be the property:

$$\exists u_3 \exists u_4 \Big(\alpha(u_1, u_3, u_5) \wedge \alpha(u_2, u_4, u_5) \wedge pps_1(u_1, u_2, u_3, u_4, u_5) \Big)$$

It is clear that the property $pps'(z, t, x, y, w)$ holds if and only if (z, t) is a polar pair of $bcc(w)$ separating $Att_G(x, bcc(w))$ and $Att_G(y, bcc(w))$. We now prove that there exists a log-labeling that verifies the statements of Proposition 9.7 and that is furthermore, a log-labeling for pps'.

Proposition 9.8 *Let G be a connected planar graph with n vertices and let $BT^*(G)$ be its representing tree. There exist $\log$-labelings R_0 and R on $BT^*(G)$ such that for every x, y in V_G and every subset X of $V_G - \{x, y\}$:*

(i) *we can determine from $R_0(x), R_0(y)$ and $R_0(X)$ whether $p(x, y)$ goes through some $z \in X$. Otherwise,*

(ii) if $C_1, \ldots, C_m$ are the problematic biconnected components for (x, y, X), then we can determine in $O(|X^2|)$-time from $R_0(x), R_0(y)$ and $R_0(X)$ the label $R_0(bcc^{-1}(C_i))$, the sets $R_0(X \cap V_{C_i})$ for $i \in [m]$ and $R_0(x_1), \ldots, R_0(x_{m-1})$ where the vertices $x_1, \ldots, x_{m-1}$ are leaders of some of the problematic biconnected components such that $(x_0 = x, \; x_m = y)$:

$$Conn(x, y, X) \Longleftrightarrow \bigwedge_{1 \leq i \leq m} Conn(x_{i-1}, x_i, X \cap V_{C_i}).$$

(iii) For each $i \in [m]$, by using $R_0(x_{i-1}), R_0(x_i), R_0(bcc^{-1}(C_i))$ and $R_0(X \cap V_{C_i})$, we can determine in $O(|X|^2)$-time whether $Att_G(x_{i-1}, C_i)$ and $Att_G(x_i, C_i)$ are separated by some polar pair in $(X \cap V_{C_i})^2$.

Moreover, the labelings R_0 and R are constructed in $O(n \cdot \log(n))$-time.

Proof. For every biconnected component C of G, the labeled tree $T^*(C)$ is the union of directed paths in $BT^*(G)$ originating from $Root(C)$. Then the set of nodes of $T^*(C)$ is MS-definable in $BT^*(G)$. Therefore, the query pps_1 on C can be translated into a query $pps_1^{\#}$ on $BT^*(G)$ (Proposition 1.3), hence the query pps' can be expressed by an MS formula in $BT^*(G)$. By Theorem 7.4, there exists a log-labeling for the properties $\leq^{\#}, member^{\#}, \varphi_1^{\#}, \varphi^{\#}$ and $pps_1^{\#}$. For each vertex x in V_G, we let

$$R(x) = \Big(R_0(x), R_0(mother(x)), R_0(leader(mother(x)))\Big).$$

If n is the number of vertices of G, the labeling R_0 is constructed in $O(n \cdot \log(n))$-time, hence the labeling R (Theorem 7.4). It is also clear that $|R(x)| \leq O(\log(n))$ for every x in V_G. We now explain given $R(x), R(y)$ and $R(X)$ how to verify the statements (i)-(iii).

By Lemma 9.11 and Proposition 9.7, by using $R(x), R(y)$ and $R(X)$ we can verify the statements (i)-(ii). The label $R_0(x)$ contains a log-labeling for pps'. Then by using $R_0(x_{i-1}), R_0(x_i)$ and $R_0(X \cap V_{C_i})$, we can verify statement (iii). ∎

It remains to label the vertices in order to answer queries $Conn(x, y, X)$ when X is contained in a biconnected component and X does not contain any polar pair of C that separates x and y. For that purposes we will adapt the notion of barriers and reduce such connectivity queries to the connectivity queries in the plane as in Sections 9.4 and 9.5.

Definition 9.19 (Augmented Graphs of Biconnected Components) For every graph H, we let $Spl(H) = \langle V_H, E' \rangle$ where $E' = \{xy \mid x, y \in V_H$ and x and y are adjacent in $H\}$. For every connected plane graph G, we let $G^- = Spl(G^+)$ where G^+ is the augmented graph of G.

For every connected plane graph G and every biconnected component C of G, we let $F_G(C)$ be the set $\{f \in F_G \mid$ there exist $x, y \in V_C$ and $xy \in E_C$ and $fx, fy \in E_{G^+}\}$.

Let G be a connected plane graph and let G^+ be its augmented graph. For every biconnected component C of G, we let $\mathcal{E}^-(C)$ be the restriction of $\mathcal{E}^+$ to $G^-[V_C \cup F_G(C)]$ where $\mathcal{E}^+$ is an embedding of G^+.

For every connected plane graph G and every biconnected component C of G, the following relates an embedding of C^+ with $\mathcal{E}^-(C)$.

Lemma 9.13 *Let G be a simple connected plane graph and let $\mathcal{E}^+$ be an embedding of G^+. Then $\mathcal{E}^-(C)$ is an embedding of C^+.*

Proof. Let $\mathcal{E}^-$ be the restriction of $\mathcal{E}^+$ to G^-. It is clear that $\mathcal{E}^+$ and $\mathcal{E}^-$ coincide in C and form an embedding $\mathcal{E}'$ of C. Each face $f \in F_G(C)$ defines a unique face f' of $\mathcal{E}'$. We let $\alpha : F_G(C) \to F_C$ be the mapping that maps every face $f \in F_G(C)$ into a face $\alpha(f)$.

We first prove that α is injective. Assume this is not the case and let f_1 and f_2 be 2 distinct faces in $F_G(C)$ such that $\alpha(f_1) = \alpha(f_2)$. The border Γ of f_1 (considered as a face of G) contains at least one edge of C and at least one edge not in C, that separates f_1 and f_2 in $\mathcal{E}^+$ and does not in $\mathcal{E}^-$ (since we assume $f_1 \neq f_2$ and $\alpha(f_1) = \alpha(f_2)$). Hence, Γ contains a nonempty path with no edge in C that links two distinct vertices of C, i.e., the union of this path and C is a 2-connected sub-graph of G. A contradiction because C is a biconnected component. It follows that α is injective.

We now prove that α is surjective. Let $g \in F_C$ and let $\mathcal{E}'(g)$ be the corresponding open subset of the plane associated with g in the embedding $\mathcal{E}'$. The associated embedding by $\mathcal{E}^+$ of each biconnected component of G is either in $\mathbb{R}^2 - \mathcal{E}'(g)$ or in $\mathcal{E}'(g) \cup \mathcal{E}'(\Gamma)$ where Γ is the border of g. It is clear that $\mathcal{E}'(g) - \bigcup \{\mathcal{E}(D) \mid D$ is a biconnected component of $G,\ D \neq C\}$ is $\mathcal{E}(f)$ for some face $f \in F_G(C)$ and that $g = \alpha(f)$. Hence, α is a bijection.

We finish the proof by showing that $G^-[V_C \cup F_G(C)]$ is simple. In $G^+[V_C \cup F_G(C)]$ there are several edges between f (such that $g = \alpha(f)$ as above) and a vertex x of G if some biconnected component D of G is embedded by $\mathcal{E}^+$ in $\mathcal{E}'(g) \cup \mathcal{E}'(\Gamma)$, Γ is the border of g and is such that $V_D \cap V_C = \{x\}$. In $G^-[V_C \cup F_G(C)]$ only one remains in such a case between f and x. Then the restriction of $\mathcal{E}^+$ to $G^-[V_C \cup F_G(C)]$ is an embedding of C^+. ∎

The following example is an illustration of Lemma 9.13.

Example 9.10 We consider the graph G^+ on Figure 25. It is not simple. Let G^- be obtained by deleting a and c, and let $\mathcal{E}^-$ be the corresponding planar embedding. Let C be the biconnected component with $V_C = \{x, t, v\}$. Then the restriction of $\mathcal{E}^-$ to $G^-[V_C \cup F_G(C)]$ is shown on Figure 39. It is an embedding of C^+.

We now define the reduced barriers for connected planar graphs.

Definition 9.20 (Reduced Barriers for Connected Planar Graphs) Let G be a connected plane graph, let G^+ be its augmented planar graph and let G^- be $Spl(G^+)$. For every 2 distinct vertices x and y of G, we let

$$RBar(\{x,y\}) = \begin{cases} Bar(\{x,y\}) \cap E_{G^-} & \text{if } x \text{ and } y \text{ are incident with at most 2 faces,} \\ \{fx, fy \mid f = Select(x,y)\} & \text{otherwise.} \end{cases}$$

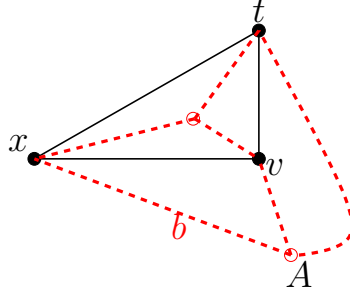For every $X \subseteq V_G$, we let $RBar(X) = \bigcup_{\substack{x,y \in X \\ x \neq y}} RBar(\{x,y\})$.

Figure 39: Illustration of Example 9.10.

The following relates $RBar(X)$ and $Bar(X)$ when X is contained in a biconnected component.

Lemma 9.14 *Let G be a plane graph and let C be a biconnected component of G. Let X be a subset of V_C and let x and y be in $V_G - X$ that either belong to V_C or are connected to V_C by paths that do not go through X and such that $Att_G(x, C)$ and $Att_G(y, C)$ are not separated by a polar pair of C in $X \times X$. Then x and y are separated in G by X if and only if $p(x)$ and $p(y)$ are separated by $RBar(X, \mathcal{E}_0)$ where $\mathcal{E}_0 = (p, s)$ is a straight-line embedding of G^-.*

Proof. Assume that x and y are separated by $RBar(X, \mathcal{E}_0)$ and let us extend $\mathcal{E}_0$ into an embedding $\mathcal{E}^+$ of G^+ with edges in $E_{G^+} - E_{G^-}$ represented by curve segments so that $\mathcal{E}^- = \mathcal{E}_0$. If x and y are separated in the plane by $RBar(X, \mathcal{E}_0)$, they are separated by $RBar(X, \mathcal{E}^-)$, hence they are also separated by X in G.

For the other direction, let x and y be separated by X in G. Let x' in $V_C - X$ be x if $x \in V_C$ or be linked to x by a path avoiding X. Let y' be defined similarly from y. Clearly, x' and y' are separated in C by X. Hence, they are separated in the plane by $RBar(X, \mathcal{E}^-(C))$ (Proposition 9.6. But by Lemma 9.13, $\mathcal{E}^-(C)$ is the embedding of C^+ defined as a restriction of $\mathcal{E}_0 = \mathcal{E}^-$. Hence x' and y' are separated by $RBar(X, \mathcal{E}_0)$ in the plane. Each of the two paths linking x to x' and y to y' avoids X, hence is in a connected component of $\mathbb{R}^2 - RBar(X, \mathcal{E}_0)$. Hence, x and y are also separated in the plane by $RBar(X, \mathcal{E}_0)$. ∎

We illustrate Lemma 9.14 with Example 9.11.

Example 9.11 We use the graph W of Example 9.8. Figure 40 shows the graph W^-. We have $F_W = \{A, B, C, \ldots, F, G, H\}$. We do not show in full all edges incident with A. Let Z be the biconnected component with $V_Z = \{1, 4, 5, 9, 14\}$, $s(Z) = 9$, $n(Z) = 1$. Then Z^+ consists of Z augmented with the following edges: $A - 1, A - 5, A - 9, \ , C - 1, C - 5, C - 14, C - 4, D - 4, D - 14, D - 5, E - 4, E - 5, E - 9, H - 1, H - 4, H - 9$. It is clear that $Z^+ = W^-[\{1, 4, 5, 9, 14, A, C, D, E, H\}]$.

Let $X = \{1, 4, 5\}$. Condition (a) of Lemma 9.9 shows that 2 and 3, and 9 and 14 are separated by X. Note that 4 and 5 form a //-polar pair. They are incident with 3 faces; 1 and 4 form also a polar pair but not a //-polar pair.

We can now prove Theorem 9.1.
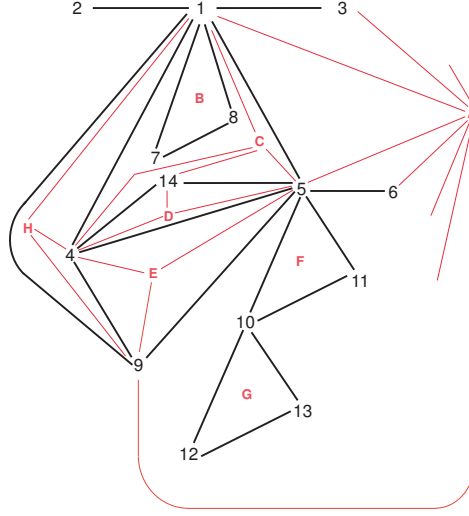
Figure 40: A graph W^-.

Proof of Theorem 9.1. We first consider connectivity queries when we only exclude vertices.

Let G be a connected planar graph with n vertices. If G is m-face bounded 2-connected, by Theorem 9.2 there exists a log-labeling J for the connectivity query on G that verifies all the statements. If G is 2-connected, by Theorem 9.4 there exists a log-labeling J for the connectivity query on G that verifies all the statements. Assume now that G is not 2-connected.

By a linear-time algorithm, we can construct the rooted tree $BC(G)$ of biconnected components. We construct the representing tree $BT^*(G)$ in linear-time (each $T^*(C)$ is computed in time linear in the number of edges of C and the number of edges of all C is exactly the number of edges of G, which is at most $3n$).

We transform G into a plane graph that we still denote by G. We can therefore construct G^+. We let $G^- = Spl(G^+)$ and we define a straight-line embedding $\mathcal{E}_0 = (p, s)$ of G^- by [Sch90]. Each vertex of G^-, i.e, each element x in $V_G \cup F_G$, has a pair of integer coordinates $p(x)$ of size at most $2 \cdot (\lceil \log(n) \rceil + \log(3))$.

By Proposition 9.2 and Remark 9.1, we can represent with 24 functions $f_1, \ldots, f_{24}$ the property that 2 vertices are incident with at most 2 faces. For every $x \in V_G$, we let

$$C_0(x) = (p(x), p(f_1(x)), \ldots, p(f_{24}(x))) \, .$$

By Lemma 9.8, in each biconnected component D of G, we can represent the polar-pair property and the Select function with 12 functions $g_1^D, \ldots, g_{12}^D$. For each $x \in V_G$ if $D = mother(x)$ we let

$$C_0'(x) = (C_1^D(x), C_1^D(leader(D)))$$

where $C_1^D(x) = (p(g_1^D(x)), \ldots, p(g_{12}^D(x)))$ (if x is the root of $BC(G)$, then $C_1^D(x)$ is "undefined"). For each $x \in V_G$, we let $R(x)$ and $R_0(x)$ be the labels of x constructed by Proposition 9.8. For each $x \in V_G$, we let $C(x) = (C_0(x), C_0'(x))$ and $J(x) = R[C](x)$.

It is clear that for every $x \in V_G$, $|J(x)| = O(\log(n))$ and is constructed in $O(n \cdot \log(n))$-time. It remains to explain now how to check the connectivity of x and y in $G \backslash X$ given $J(x), J(y)$ and $J(X)$.

By using the R-parts of $J(x), J(y)$ and $J(X)$, we can verify if $p(x,y)$ goes through some $z \in X$ (Proposition 9.8) and if it does, we can answer that x and y are disconnected by X in G by Lemma 9.9.

Otherwise, let $C_1, \ldots, C_m$ be the problematic biconnected components for (x, y, X) and let $x_1, \ldots, x_{m-1}$ that are leaders of some problematic biconnected components such that ($x_0 = x$, $x_m = y$):

$$Conn(x, y, X) \Longleftrightarrow \bigwedge_{1 \le i \le m} Conn(x_{i-1}, x_i, X \cap V_{C_i}).$$

By Proposition 9.8 and Remark 9.3; by using $R(x), R(y)$ and $R(X)$; we can determine $(R_0(x_1), C(x_1)), \ldots, (R_0(x_{m-1}), C(x_m))$ and $R_0(bcc^{-1}(C_1)), \ldots, R_0(bcc^{-1}(C_m))$ and for each $1 \le i \le m$ the set $\{(R_0(z), C(z)) \mid z \in X \cap V_{C_i}\}$. For each $1 \le i \le m$, by using $R_0(x_{i-1}), R_0(x_i)$ and $R_0(X \cap V_{C_i})$, we can check if there exists a polar pair in C_i that separates $Att_G(x_{i-1}, C_i)$ and $Att_G(x_i, C_i)$ (see Proposition 9.8). If one such i is found, then we can report that x and y are disconnected. Otherwise, by Lemma 9.14, for each $i = 1, \ldots, m$, we have $Conn(x_{i-1}, x_i, X \cap V_{C_i})$ if and only if $p(x_{i-1})$ and $p(x_i)$ are separated by $RBar(X \cap V_{C_i}, \mathcal{E}_0)$. By Theorem 9.3, we can decide if $p(x_{i-1})$ and $p(x_i)$ are separated by $RBar(X \cap V_{C_i}, \mathcal{E}_0)$ if we know $RBar(X \cap V_{C_i}, \mathcal{E}_0)$. It remains to explain how to get the ends of the edges in $RBar(X \cap V_{C_i})$ since $\mathcal{E}_0$ is a straight-line embedding.

Let z and t be 2 distinct vertices in $X \cap V_{C_i}$. If z and t are incident with at most 2 faces, then we can determine these at most 2 faces that are incident with z and t by using $C_0(z)$ and $C_0(t)$ (Proposition 9.2). Now assume that z and t are incident with at least 3 faces. At least one of them, say z, has $bcc^{-1}(C_i)$ as mother. We have 2 cases:

CASE 1. $bcc^{-1}(C_i)$ is also the mother of t. By definition of C_0', we can determine $p(g_1^{C_i}(t)), \ldots, p(g_{12}^{C_i}(t))$ and by Lemma 9.8, we can therefore determine $p(Select(z, t))$ and then $RBar(\{z, t\})$.

CASE 2. $bcc^{-1}(C_i)$ is not the mother of t and therefore t is the leader of $bcc^{-1}(C_i)$. By definition of C_0', we have stored in $C_0'(z)$ the values of $p(g_1^{C_i}(t)), \ldots, p(g_{12}^{C_i}(t))$. Again by Lemma 9.8, we can therefore determine $RBar(\{z, t\})$.

Therefore, for every $(z, t) \in X \times X$ we can determine $RBar(\{z, t\})$ and then we can determine $RBar(X)$.

We now explain how to handle queries with excluded edges. For that we transform G by subdividing each edge (or only each *unsafe* edge, for which deletion may have to be handled), i.e., by inserting a new vertex w_e on each edge e. We obtain a graph G' which is simple, connected and planar. It is clear that x and y are connected in $(G - F) \backslash X$ if and only if they are connected in $G' \backslash X'$ where $X' = X \cup \{w_e \mid e \in F\}$. Hence, we can apply to G' the above

described construction, and we obtain a log-labeling J' of the vertices of G', whereas we wish a log-labeling J of the edges and the vertices of G, since edges to delete are specified as pairs of adjacent vertices. We will again use unary functions to specify edges from pairs of vertices.

We let $g_1, g_2, g_3 : V_G \to V_G$ be the 3 functions that represent adjacency in G (see Corollary 9.1). We let $g_4, g_5, g_6 : V_G \to V_{G'}$ be the 3 functions defined as follows:

$$g_{i+3}(x) = w_e \qquad \text{if} \qquad e = x - g_i(x).$$

It is clear that for every edge $e = xy$, the vertex $w_e \in V_{G'}$ is characterized by the following formula:

$$\bigvee_{1 \leq i \leq 3} \Big(\big(w_e = g_{i+3}(x) \wedge y = g_i(x)\big) \vee \big(w_e = g_{i+3}(y) \wedge x = g_i(y)\big) \Big).$$

Therefore, the binary function $Edg : V_G \times V_G \to V_{G'}$ is representable by the functions $g_1, \ldots, g_6$. For every $x \in V_G$, we then let

$$J(x) = \big(J'(x), p(g_1(x)), p(g_2(x)), p(g_3(x)), J'(g_4(x)), J'(g_5(x)), J'(g_6(x))\big).$$

Therefore, given a pair of 2 vertices z and t such that $e = zt \in E_G$ by using the formula $J(z)$ and $J(t)$ we can determine $J'(w_e)$. Hence, for every 2 distinct vertices x and y and every $X \subseteq V_G - \{x, y\}$, and every $F \subseteq E_G$, we can verify if x and y are connected in $(G - F)\backslash X$ by using $J'(x), J'(y), J'(X)$ and $\{J'(z), J'(t) \mid zt \in F\}$.   ∎

9.7   Conclusion

For our labeling scheme we combine several tools: geometrical tools (in particular the planar point location), logical tools (we use Theorem 7.4 on trees) and the bipolar orientations of 2-connected planar graphs, particularly the decomposition tree of bipolar plane graphs defined in [BT96]. The used geometrical tools are appropriate for connectivity queries but, do not extend to distance queries. The proof we give works for planar graphs and is devoted to a very particular query. Extensions to other queries are not in view.

We will see in Chapter 10 how to extend this labeling scheme to more classes of graphs by introducing decompositions of graphs similar to the ones of Schiering et al. [Sch97, DK05, WT07]. Apart from the extension of the labeling scheme to more classes of graphs, we can ask further questions.

1. If a planar graph is modified by addition of vertices and/or edges, how can we update the labels as efficiently as possible?

2. Can we extend our labeling scheme to graph classes of bounded genus?

3. Can we propose a labeling scheme for the distance or even the approximate distance in sub-graphs of planar graphs? A labeling scheme for exact distance should use labels of size at least $\Omega(n^{1/3})$ on planar graphs with n vertices [GKK+01].

Chapter 10

Short Connectivity Query Labeling on Graph Classes of Unbounded Clique-Width

10.1 Introduction

Our objective is to extend the labeling for connectivity queries on planar graphs to more classes of graphs, particularly to certain graph classes that have bounded local clique-width. The idea is to use decompositions that reduce the problem to that for planar graphs or for other graph classes where the connectivity query admits a short labeling scheme. Furthermore, we would like these decompositions to be enough general so that we can transfer other algorithmic and structural results, e.g., boundedness of local clique-width that holds for planar graphs. For that purposes, we define two types of decompositions, based on vertex-partitions and on edge-partitions. We now introduce the main ideas of the decompositions considered in this chapter.

Decompositions Based on Vertex-Partitions

Given a partition $\{V_1, \ldots, V_m\}$ of the vertex-set of a graph G, the sub-graph induced by each V_i is connected, our idea is to construct a graph H, called *quotient graph*, that encodes the edges between the different parts. For instance, the graph H could be the adjacency graph of the parts (two parts are adjacent if and only if there exists an edge between the two parts, Figure 41 is an illustration). In this case, in order to reconstruct G from H and the parts, we color the edges of H and the vertices of the graphs $G_i = G[V_i]$ in such a way that the color of each edge of H linking V_i and V_j informs on the way to reconstruct the edges between G_i and G_j. However, if the links between two parts are too complicated, we will need many colors to encode adjacencies and this is not satisfactory. Moreover, we would like to be able to color each part efficiently (say in time proportional to the size of the part and the number of links between this part and the other parts of the partition). Then the structure of the links between parts can be used to parameterize such decompositions and we can also make some constraints on the parts or on the quotient graph. For instance, we can impose each part to be

in a class of graphs of bounded clique-width and the graph H to be planar. An example of such a decomposition is the strong tree-decomposition. However, in strong tree-decompositions the parameter is the size of the parts and not the structure of the links between parts. When a class of graphs has strong tree-width k, the links between the parts are described in a simple way (by using for instance $(k \times k)$-matrices).
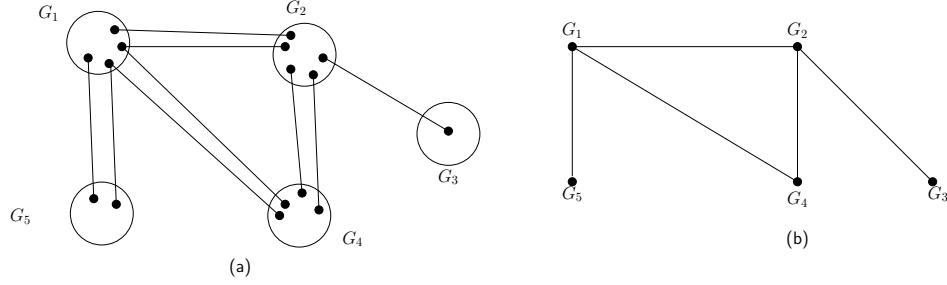
Figure 41: (a) is a partition of a graph G and (b) is the associated graph quotient.

The quotient graph as defined above (see Figure 41) is not suitable for labeling schemes of connectivity query because, informally, we may miss some paths. We will instead use a slightly different coding of the adjacencies and call this coding a *v-skeleton*. Intuitively it is defined as follows: if $x_{i_1}, \ldots, x_{m_i}$ are the vertices of V_i adjacent to vertices of other parts, we replace in the quotient graph the vertex that represents V_i by the pattern described on Figure 42 and if x_{i_s} is adjacent to x_{j_r} of V_j we add an edge between x_{i_s} and x_{j_r}. See Figure 42 for an illustration of the v-skeleton associated with H on Figure 41.
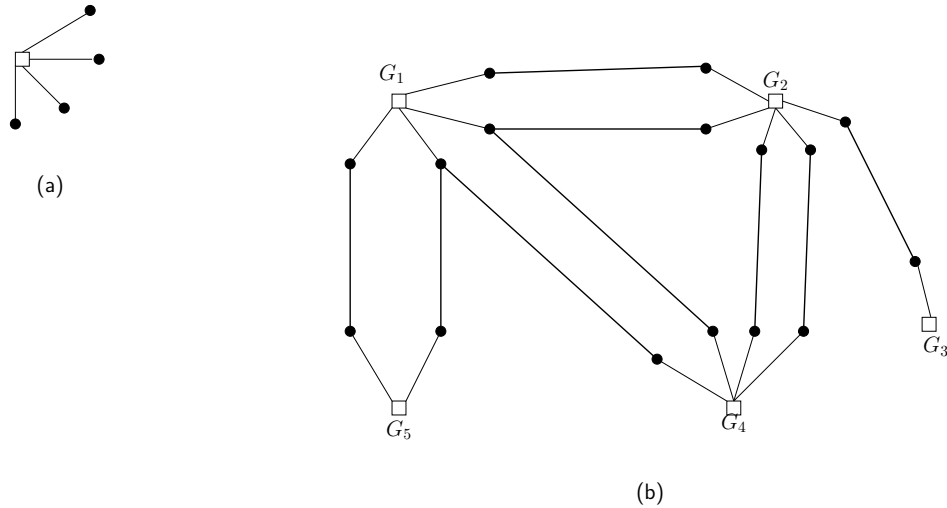
Figure 42: (a) represents the vertices of the v-skeleton associated with G_1 and (b) is the v-skeleton of the vertex-partition on Figure 41.

We choose for our notion of decomposition vertex-partitions where the edges between parts are described by bipartite graphs, of small size. Then the vertices in each part are partitioned into classes of an equivalence relation that describes how the vertices of the part are linked to vertices of the other parts. In this case, the v-skeleton has a more compact representation

(instead of coding all the vertices that are linked to vertices in other parts, we just code their equivalence classes). For instance, the split decomposition [Cun82] is a decomposition based on bipartitions of vertices where the links between the two parts are described by the bipartite graph shown on Figure 43. Note however that the split decomposition is not subsumed by our notion of decomposition because in split decomposition, adjacencies between vertices of different parts are encoded as paths in the decomposition [Cun82, Cou06b] while in our decomposition if two vertices in different parts are adjacent, their equivalence classes are adjacent.
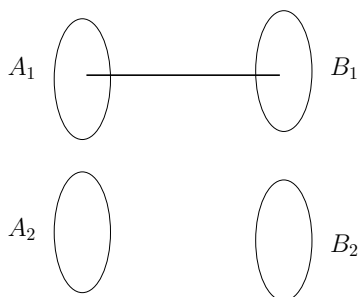
Figure 43: The bipartite graph that describes the links between splits.

Decompositions Based on Edge-Partitions

Given a partition $\{E_1, \ldots, E_m\}$ of the edge-set of a graph G, the sub-graph induced on each E_i is connected, we can define the quotient graph H as the intersection graph of the partitions where an edge $E_i E_j$ means that E_i and E_j shares at least one vertex. However, this representation is not interesting for labeling schemes of connectivity queries. We will prefer the one, called *e-skeleton* where if a part E_i shares the vertices $x_{i_1}, \ldots, x_{i_m}$ with the other parts, it is represented by a vertex linked with $x_{i_1}, \ldots, x_{i_m}$. Figure 44 illustrates it.
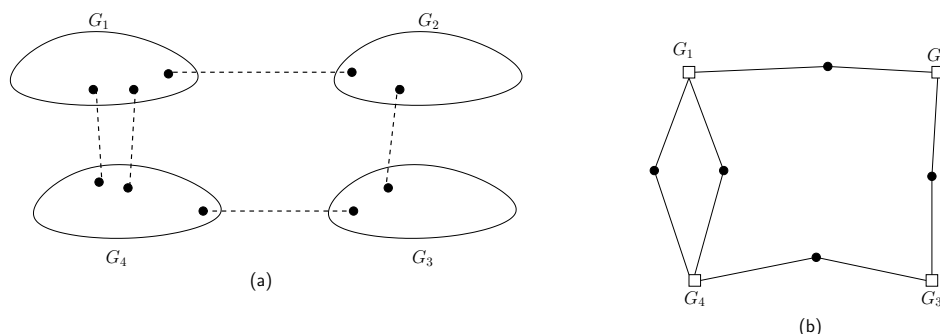
Figure 44: (a) is a partition of a graph G and (b) is the associated e-skeleton. The dashed lines means that it is the same vertex.

An example of such a decomposition is the decomposition of a connected graph into the tree of its biconnected components. As parameters, we can use the number of parts that contain a vertex or the size of the set $E_i \cap \left(\bigcup_{j \neq i} E_j \right)$ for each $i = 1, \ldots, m$. We can also impose some restrictions on the graphs induced by the parts or on the e-skeleton. Since the

e-skeleton is not necessarily a tree in our framework, we will need to bound the two possible parameters in order to get short labeling schemes for connectivity queries. An example of such a decomposition is the domino tree-decomposition [BE97] where the e-skeleton is a tree and each vertex is in at most two parts. However, general tree-decompositions are not handled by our framework because in tree-decompositions there is no bound on the number of parts that contain a vertex. Ding et al. [DOSV00, DDO$^+$04] and Nešetřil and Ossona de Mendez [NdM06b, NdM08a, NdM08b] considered decompositions based on edge-partitions, where each bounded union of parts must have small tree-width, but there are no constraints on the number of parts a vertex may belong to, or on the quotient or e-skeleton graph.

Summary of the Chapter

We define in Section 10.2 the notion of $\mathcal{H}$-*v-decomposition* and of $\mathcal{H}$-*v-width*. We give a sufficient condition for graph classes of small $\mathcal{H}$-v-width to having bounded local clique-width. In Section 10.3 we define the notion of *v-skeleton* and we prove that some graph classes of bounded $\mathcal{H}$-v-width admit a log-labeling scheme for the connectivity query. In Section 10.4 we introduce the notions of $\mathcal{H}$-*e-decomposition*, of $\mathcal{H}$-*e-width* and of *e-skeleton*. We prove in Section 10.4 that some graph classes of small $\mathcal{H}$-e-width admit a log-labeling scheme for the connectivity query.

We denote by $\mathcal{P}$ the class of undirected planar graphs and by $CWD(\leq k)$ the class of undirected graphs of clique-width at most k. For every graph G, every sub-graph H of G and every $X, Y \subseteq V_G$, we denote by $p(H, X, Y)$ the property that there exists a path between a vertex of X and a vertex of Y in H. For convenience we write $x \overset{*}{-} y$ to mean that x and y are connected by a path in some graph that is clear from the context.

10.2 $\mathcal{H}$-v-Decompositions of Undirected Graphs

We recall the definition of the notion of *k-module* [Joh03, Rao06] which generalizes the notions of module [Gal67] and of *bi-module* [dM03, Rao06].

Definition 10.1 (k-Modules of Graphs [Rao06]) Let G be a graph and let $M \subseteq V_G$. We say that M is a *k-module of G* if their exists a partition $\{M^1, \ldots, M^k\}$ of M such that for every $x, y \in M^i$ and for every $z \in V_G - M$, we have

$$E_G(x, z) \qquad \text{if and only if} \qquad E_G(y, z).$$

Let M be a subset of V_G. A *pseudo-module of M with respect to G* is a maximal subset M' of M such that for all $z \in V_G - M$, either z is adjacent to all vertices of M' or to no vertex of M'. For every $M \subseteq V_G$, we denote by $bm_G(M)$ the number of pseudo-modules of M with respect to G. If $bm_G(M) = k$, we will denote by $\{M^1, \ldots, M^k\}$ the pseudo-modules of M with respect to G.

Figure 45 shows a 4-module. On Figure 45, the sets M_3 and M_4 are pseudo-modules while M_1 and M_2 are not pseudo-modules (lack of maximality). But, $M_1 \cup M_2$ is a pseudo-module.

One observes that a module is a 1-module and a subset M of V_G is a k-module if and only if $bm_G(M) \leq k$. The notion of k-module was used by Johnson [Joh03] in her investigations
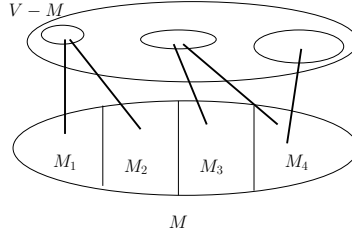
Figure 45: A 4-module in a graph.

of recognition algorithms for graph classes of clique-width at most k. She used the notion of k-module in order to define a decomposition called k-*HB decomposition*, that handles graph classes of clique-width at most k. Rao also studied the notion of k-module in [Rao06] (de Montgolfier [dM03] studied before the notion of 2-module). Rao [Rao06] gave a polynomial-time algorithm that decides if a given graph has a k-module, for fixed k and gave also a structure, computable in polynomial-time, that represents the set of k-modules in a graph. He also defined in [Rao06] a decomposition, called *décomposition k-modulaire*, that is a tree that represents some k-modules and an associated graph parameter, called *largeur modulaire*, that is equivalent to clique-width. The following properties of k-modules are proved in [Rao06].

Lemma 10.1 ([Rao06]) *Let G be a graph and let M be a subset of V_G. Then*

1. *If M is a k-module of G, then M is also a k-module of $\overline{G}$, the edge-complement of G.*

2. *If M is a k-module of G, then $V_G - M$ is a 2^k-module of G.*

For algorithmic purposes, we are interested, given a subset X of the set of vertices such that $bm_G(X) \leq \ell$, in computing the set of pseudo-modules of X. The following proves the existence of such an algorithm.

Lemma 10.2 *Let G be an undirected graph with n vertices and let $X \subseteq V_G$ such that $bm_G(X) \leq \ell$. We can compute the pseudo-modules $X^1, \ldots, X^{bm_G(X)}$ of X in $(|X| \cdot n)$-time.*

Proof. Let G be an undirected graph with n vertices and assume the vertices are numbered 1 to n and G is given with its adjacency list. For each vertex x of X, we let $l(x)$ be its set of adjacent vertices in $V_G - X$ and we can assume it ordered. We use the following algorithm:

Step 1.   Let $X^1 = \emptyset, \ldots, X^\ell = \emptyset$.

Step 2.   Let $x_1 = 0, \ldots, x_\ell = 0$.

Step 3.   For each $x \in X$,

  Step 3.1.   For $i = 1, \ldots, \ell$,
    Step 3.1.1.   If $X^i \neq \emptyset$ and $l(x) = l(x_i)$, then put x in X^i and go to Step 3.
    Step 3.1.2.   If $X^i = \emptyset$, then put x in X^i, let $x_i = x$ and go to Step 3.

Step 4. Return $X^1, \ldots, X^\ell$.

Note that at Step 3.1.1 the sets $X_1, \ldots, X_i$ are all nonempty. It is clear that x and y are in X^i if and only if $l(x) = l(x_i)$. And $l(x) = l(x_i)$ if and only if $l(x)[j] = l(x_i)[j]$ for all $j \leq n$. Then we can compare $l(x)$ and $l(x_1)$ in $O(n)$-time. Since $bm_G(X) \leq \ell$, it is clear that we will find at least one X^i such that $l(x) = l(x_i)$ for all $x \in V_G$.

Since for each x we make at most $\ell - 1$ comparisons and each comparison is done in $O(n)$-time, we compute all the pseudo-modules of X in $O(|X| \cdot n)$-time (each set $l(x)$ can be ordered by using any linear-time sorting algorithm for integers [CLR02]).   ∎

As a corollary we have the following.

Corollary 10.1 *Let G be an undirected graph with n vertices. If $\{X_1, \ldots, X_p\}$ is a partition of V_G and $bm_G(X_i) \leq \ell$ for each $i \leq p$, then we can compute the pseudo-modules of all the sets X_i in $O(n^2)$-time.*

Proof. Let G be an undirected graph with n vertices given with its adjacency list. For each X_i we can compute the pseudo-modules of X_i in $O(|X_i| \cdot n)$-time by Lemma 10.2. Then We compute the pseudo-modules of all X_i in $\left(\sum_{1 \leq i \leq p} O(|X_i| \cdot n) \right)$-time. But,

$$\sum_{1 \leq i \leq p} O(|X_i| \cdot n) = O\left(n \cdot \sum_{1 \leq i \leq p} |X_i| \right) = O(n^2)$$

because $\sum_{1 \leq i \leq p} |X_i| = n$.   ∎

We now define our first notion of graph decomposition in the spirit of [Sch97, WT07]. As in [Rao06] we will use the notion of k-module in order to define a parameter for our decomposition. Our decomposition is based on a vertex-partition and is not necessarily a tree, contrary to the one defined in [Rao06].

Definition 10.2 ($\mathcal{H}$-v-Decomposition of Graphs) Let $\mathcal{H}$ be a class of undirected graphs. An $\mathcal{H}$-v-decomposition of an undirected graph G is a pair (H, χ) where $\chi : V_H \to 2^{V_G}$ is a mapping and such that:

(HD1) the set $\{\chi(u) \mid u \in V_H\}$ is a partition of V_G and for every $u \in V_H$, the sub-graph $G[\chi(u)]$ is nonempty and connected.

(HD2) H is in $\mathcal{H}$ and if uv is in E_H, then there exists $x \in \chi(u)$, $y \in \chi(v)$ such that xy is in E_G.

(HD3) For every edge xy in E_G,

(HD3.1) either there exists $u \in V_H$ such that $x, y \in \chi(u)$ or,

(HD3.2) there exist u and v in V_H such that $x \in \chi(u)$, $y \in \chi(v)$ and $uv \in E_H$.

The *$\mathcal{H}$-v-width* of an $\mathcal{H}$-v-decomposition (H, χ), denoted by $\mathcal{H}\text{-}wd(H, \chi)$, is $\max_{u \in V_H} \{bm_G(\chi(u))\}$. The *$\mathcal{H}$-v-width* of an undirected graph, denoted by $\mathcal{H}\text{-}wd(G)$, is the minimum $\mathcal{H}$-v-width over all $\mathcal{H}$-v-decompositions of G.

If (H, χ) is an $\mathcal{H}$-v-decomposition of a graph G, we call the vertices of H *nodes* in order to distinguish them from the vertices of G.

We recall that $\mathcal{P}$ is the class of planar graphs. Figure 46 shows a $\mathcal{P}$-v-decomposition of $\mathcal{P}$-v-width 3 of an undirected graph G.
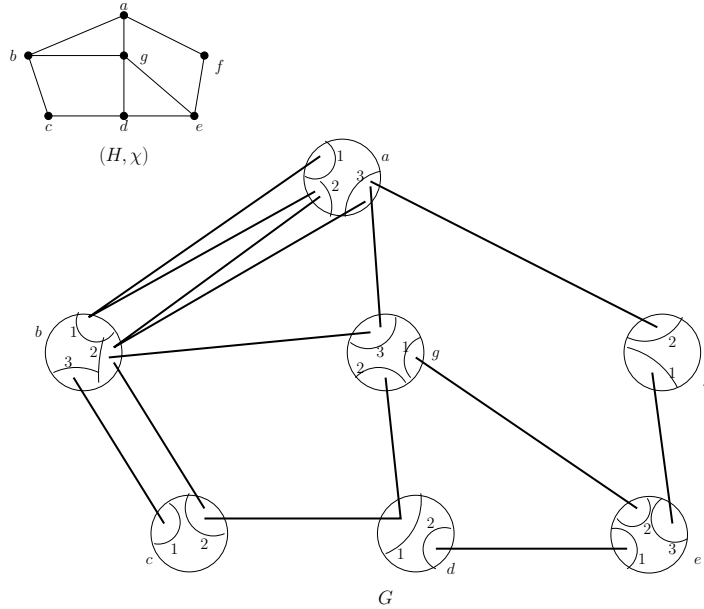
Figure 46: A graph G and a $\mathcal{P}$-v-decomposition (H, χ) of G.

If G has an $\mathcal{H}$-v-decomposition (H, χ) of $\mathcal{H}$-v-width k, we want to relate the clique-width of G with that of H and with $\max\{cwd(G[\chi(u)]) \mid u \in V_H\}$. We first define from (H, χ) an edge-colored graph, denoted by $Rel(H, \chi, G)$, and relates the clique-width of G with the tree-width of $Rel(H, \chi, G)$, and the clique-widths of the sub-graphs $G[\chi(u)]$ for $u \in V_H$.

Definition 10.3 (Edge-Colored Graphs from $\mathcal{H}$-v-Decompositions) Let $\mathcal{H}$ be a class of undirected graphs and let (H, χ) be an $\mathcal{H}$-v-decomposition of an undirected graph G of $\mathcal{H}$-v-width at most ℓ. We let $Rel(H, \chi, G)$ be the $2^{[\ell]}$-edge-colored directed graph obtained as follows:

1. As a graph it is obtained from H by orienting each edge of H.

2. Each arc (u, v) of $Rel(H, \chi, G)$ is colored by the set $\{(i, j) \mid \chi(u)^i$ and $\chi(v)^j$ are adjacent$\}$.

The orientation of H is arbitrary, however it is interesting to choose an orientation of H that minimizes the clique-width of $Rel(H, \chi, G)$. This is not easy algorithmically. Presumably, it is NP-hard.

Figure 47 shows the directed graph $Rel(H, \chi, G)$, which is an edge-colored graph. For instance the color $(1,2)$ in the list of colors of the arc $(b,a) \in E_H$ means that all vertices in $\chi(b)^1$ are adjacent with all the vertices in $\chi(a)^2$.
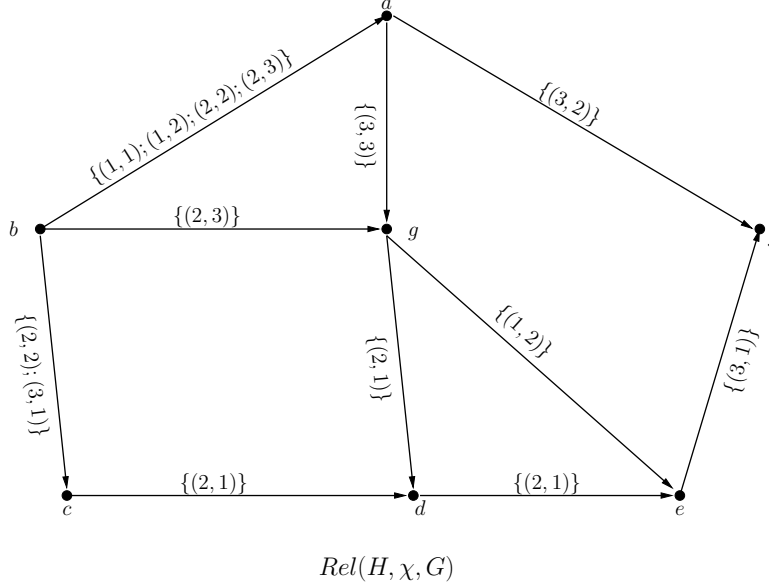
$$Rel(H, \chi, G)$$

Figure 47: The graph $Rel(H, \chi, G)$ associated with the $\mathcal{P}$-v-decomposition on Figure 46.

We now relate the tree-width of H and the clique-width of $Rel(H, \chi, G)$. The notion of clique-width of edge-colored graphs is defined in Definition 1.3, Section 1.1.

Lemma 10.3 *Let $\mathcal{H}$ be a class of undirected graphs and let (H, χ) be an $\mathcal{H}$-v-decomposition of $\mathcal{H}$-v-width at most ℓ of an undirected graph G. If $Rel(H, \chi, G)$ has tree-width at most k, then $Rel(H, \chi, G)$ has clique-width at most $2^{2 \cdot k' + 1}$ where $k' = k \cdot 2^{\ell}$.*

Proof. We can prove that if K, a C-edge-colored directed graph where C is a finite list of colors, has tree-width at most k, then K has clique-width at most $2^{2(k \cdot |C|) + 1}$ by modifying the proof of [CO00, Theorem 5.5]. ∎

We now prove the following.

Proposition 10.1 *Let $\mathcal{H}$ be a class of undirected graphs and let (H, χ) be an $\mathcal{H}$-v-decomposition of $\mathcal{H}$-v-width ℓ of G. If $twd(Rel(H, \chi, G)) = t$ and $k = max\{cwd(G[\chi(u)]) \mid u \in V_H\}$, then $cwd(G) \leq \max\{p \cdot \ell, k \cdot \ell\}$, where $p = 2^{2 \cdot (t \cdot 2^{\ell}) + 1}$.*

Before, let us prove a technical lemma.

Lemma 10.4 *Let G be a graph and let U be a subset of V_G such that $bm_G(U)$ is at most ℓ. If $cwd(G[U]) \leq k$, then there exists a term t_U such that:*

(i) $t_U \in T(F^{uc}_{k \times \ell}, C^c_{k \times \ell})$.

(ii) $G[U] = val(t_U)$.

(iii) *Every* $x \in U$ *is colored by* i *in* $val(t_U)$ *if and only if* $x \in \chi(U)^i$.

Proof of Lemma 10.4. Let $U \subseteq V_G$ be such that $bm_G(U) \leq \ell$ and $cwd(G[U]) \leq k$. Let t be a term in $T(F_k^{uc}, C_k^c)$ and such that $G[U] = val(t)$. We use the following rules to transform t into a term $\hat{t}$ in $T(F_{k \times \ell}^{uc}, C_{k \times \ell}^c)$ that verifies statements (i) and (ii) (for convenience we will write $x(i)$ to mean that $\mathbf{i}$ defines the vertex x):

$$
\hat{t} = \begin{cases}
x\big((i,j)\big) & \text{if } t = x(i) \text{ and } x \in \chi(U)^j, \\
\hat{t_1} \oplus \hat{t_2} & \text{if } t = t_1 \oplus t_2, \\
\left(\circ_{\substack{1 \leq j' \leq \ell \\ 1 \leq j'' \leq \ell}} \eta_{(i,j'),(j,j'')} \right) (\hat{t_1}) & \text{if } t = \eta_{i,j}(t_1), \\
\left(\circ_{1 \leq j' \leq \ell} \rho_{(i,j') \to (j,j'')} \right) (\hat{t_1}) & \text{if } t = \rho_{i \to j}(t_1).
\end{cases}
$$

It is straightforward to verify by induction that $val(\hat{t}) = val(t)$ and that if $x \in \chi(U)^j$ has color i in $val(t)$, then it has color (i,j) in $val(t_U)$. We let

$$
t_U = \left(\circ_{\substack{1 \leq i \leq k \\ 1 \leq j \leq \ell}} \rho_{(i,j) \to j} \right) (\hat{t}).
$$

It is clear that the term t_U verifies statements (i)-(iii). ∎

We can now prove Proposition 10.1.

Proof of Proposition 10.1. By Lemma 10.3, we have $cwd(Rel(H, \chi, G)) \leq p$ where $p = 2^{2 \cdot (t \cdot 2^\ell) + 1}$. Let t be a term in $T(F_p^C, C_p^c)$ such that $Rel(H, \chi, G) = val(t)$ and for every $u \in V_H$, we let $t_u \in T(F_{k \times \ell}^{uc}, C_{k \times \ell}^c)$ such that $G[\chi(u)] = val(t_u)$ (Lemma 10.4). For every $i \in [p]$ and every term t_u, we let $\rho^i(t_u) = \left(\circ_{1 \leq j \leq \ell} \rho_{j \to (i,j)} \right)(t_u)$. We use the following rules to transform t into a term $\hat{t}$ such that $val(\hat{t}) = G$.

$$
\hat{t} = \begin{cases}
\rho^i(t_u) & \text{if } t = u(i), \\
\hat{t_1} \oplus \hat{t_2} & \text{if } t = t_1 \oplus t_2, \\
\eta_{(i,s),(j,t)}^{s,t}(\hat{t_1}) & \text{if } t = \eta_{i,j}^{s,t}(t_1), \\
\left(\circ_{1 \leq m \leq \ell} \rho_{(i,m) \to (j,m)} \right)(\hat{t_1}) & \text{if } t = \rho_{i \to j}(t_1).
\end{cases}
$$

It is clear that $\hat{t} \in T(F_{k'}^{uc}, C_{k'}^c)$ where $k' = \max\{p \cdot \ell, k \cdot \ell\}$. It is a straightforward induction to verify that $G = val(\hat{t})$. ∎

For every undirected graph H, we let $\overline{H}_k$ denote any graph obtained from H with edges colored with colors in $2^{[k]}$ in all possible ways. For every class of undirected graphs $\mathcal{H}$, we let $\overline{\mathcal{H}}_k$ be the set $\{\overline{H}_k \mid H \in \mathcal{H}\}$. As a consequence of Proposition 10.1 we get the following.

Proposition 10.2 *Let* ℓ *and* k *be positive integers and let* $\mathcal{H}$ *be a class of undirected graphs that has bounded local tree-width. Let* $\mathcal{C}$ *be a class of undirected graphs of* $\mathcal{H}$-*v-width at most* ℓ. *If, for every* $G \in \mathcal{C}$, *there exists an* $\mathcal{H}$-*v-decomposition* (H, χ) *of* $\mathcal{H}$-*v-width at most* ℓ *such that for every* $u \in V_H$, $cwd(G[\chi(u)]) \leq k$, *then* $\mathcal{C}$ *has bounded local clique-width.*

Proof. Let $\mathcal{H}$ be a class of graphs of bounded local tree-width and let f be the function that bounds the local tree-width. Let (H, χ) be an $\mathcal{H}$-v-decomposition of $\mathcal{H}$-v-width at most ℓ of an undirected graph G in $\mathcal{C}$ such that for every $u \in V_H$, the clique-width of $G[\chi(u)]$ is at most k. One can verify that for every vertex x of G and every positive integer r, the set $N_G^r(x)$ is included in $V = \bigcup_{v \in N_H^r(u)} \chi(v)$ where x is in $\chi(u)$. By hypothesis, $Rel(H, \chi, G)$ is in $\overline{\mathcal{H}}_\ell$ and since tree-width is independent of the colors of the edges, it has its local tree-width bounded by f. The sub-graph $G[V]$ has radius at most r, hence its tree-width is bounded by $f(r)$. By Proposition 10.1, the clique-width of $G[V]$ is bounded by $\max\{p \cdot \ell, k \cdot \ell\}$ where $p = 2^{2 \cdot f(r) \cdot 2^\ell + 1}$. Hence, the clique-width of $G[N_G^r(x)]$ is bounded since $G[N_G^r(x)]$ a sub-graph of $G[V]$. $\blacksquare$

In the next section we apply $\mathcal{H}$-v-decompositions to connectivity query when we only exclude vertices.

10.3   Application of $\mathcal{H}$-v-Decompositions to Labeling Scheme for Connectivity Query

In this section we prove that certain classes of graphs of bounded $\mathcal{H}$-v-width admit a short labeling scheme for the connectivity query with excluded vertices only. The principal idea is the following:

- Let $\mathcal{H}$ and $\mathcal{D}$ be classes of undirected graphs that admit short labeling schemes for the connectivity query and assume that G has an $\mathcal{H}$-v-decomposition (H, χ) of small $\mathcal{H}$-v-width such that for all u in V_H, the sub-graph $G[\chi(u)]$ is in $\mathcal{D}$.

- We will combine the labeling scheme for graphs in $\mathcal{D}$ and the labeling scheme for graphs in $\mathcal{H}$ in order to construct a short labeling scheme for G.

However, the existence of a path between x and y in $G \backslash X$ cannot be verified directly in H. Assume for instance the following:

- in the graph $G \backslash X$ the only path between x and y should go through $\chi(v)^i$ and $\chi(v)^j$ for some v in V_H such that $\chi(v)^i = \{z\}$ and $\chi(v)^j = \{t\}$ and $X \subseteq \chi(v)$;

- moreover, there is no path in $G[\chi(v)] \backslash X$ between z and t. But, there is a path in $G \backslash X$ between z and t that does not go through $\chi(v)$.

If we use only the labeling scheme for H and the ones for all $G[\chi(u)] \in \mathcal{D}$ (that are assumed to be connected), there is no way to find the path between z and t and then between x and y. To overcome this difficulty, we will use a different coding of the $\mathcal{H}$-v-decomposition, called *v-skeleton*, that will allow us to find this path between z and t, and, instead of using a labeling scheme for H, we will use a labeling scheme for the v-skeleton. Informally, the v-skeleton is a graph with set of nodes that is the set of pseudo-modules and where there is an edge between two pseudo-modules if and only if they are adjacent in G. We define it formally now.

Definition 10.4 (v-Skeleton) Let $\mathcal{H}$ be a class of undirected graphs and let (H, χ) be an $\mathcal{H}$-v-decomposition of $\mathcal{H}$-v-width ℓ of an undirected graph G. The *v-skeleton of (H, χ)*, denoted by $Skl_{(H,\chi)}$, is the undirected bipartite graph $\langle V_{Skl_{(H,\chi)}}, E_{Skl_{(H,\chi)}} \rangle$ where:

$$V_{Skl_{(H,\chi)}} = \bigcup_{u \in V_H} \{u, u_1, \ldots, u_{bm_G(\chi(u))}\},$$

$$E_{Skl_{(H,\chi)}} = \bigcup_{u \in V_H} \{uu_i \mid 1 \leq i \leq bm_G(\chi(u))\} \ \cup \ \bigcup_{uv \in E_H} \{u_i v_j \mid (u, v) \text{ has color } (i, j)\}.$$

For every $u \in V_H$, the vertex u of $Skl(H, \chi)$ is called a *block-vertex* and for every $1 \leq i \leq bm_G(\chi(u))$, the vertex u_i is called an *attachment-vertex*.

Informally for each u of H, we create new vertices, denoted $u_1, \ldots, u_{bm_G(\chi(u))}$, associated with u and we add edges between u and its associated vertices. The edge between u_i and v_j code the edges between the pseudo-modules $G[\chi(u)^i]$ and $G[\chi(v)^j]$, forming a bipartite complete graph. Figure 48 shows the v-skeleton associated with the $\mathcal{P}$-v-decomposition on Figure 46. The blue edges represent edges between pseudo-modules and black ones represent edges between block-vertices and their associated attachment-vertices.
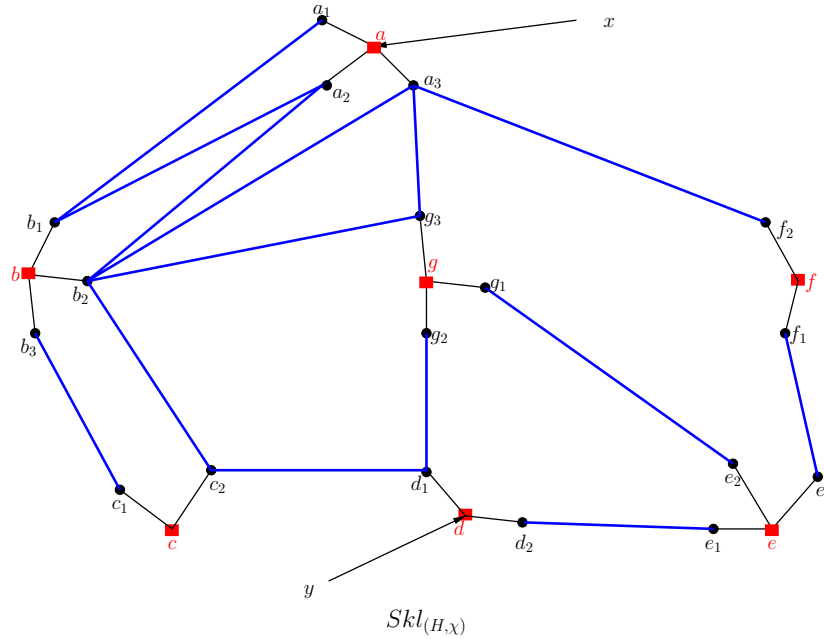
$$Skl_{(H,\chi)}$$

Figure 48: The undirected graph $Skl_{(H,\chi)}$ of the $\mathcal{P}$-v-decomposition (H, χ) on Figure 46.

Notice that with the v-skeleton we cannot still find directly the path between x and y by just using a labeling of the v-skeleton for the same reasons as explained above. However, we can construct a graph from the v-skeleton and find this path in this graph. This graph is similar to the undirected graph $BT^*(G)$ in Section 9.6, but contrary to $BT^*(G)$ (Definition 9.17) we have to construct it whenever we want to verify $Conn(x, y, X, \emptyset)$. Fortunately, its size depends only on the size of X and of the $\mathcal{H}$-v-width of the $\mathcal{H}$-v-decomposition. For that purposes, we will adapt the notion of *problematic components* of Chapter 9. We make it precise now.

Definition 10.5 (v-Problematic Block-Vertices) For every $X \subseteq V_G$, a block-vertex u is said to be *v-problematic for* X if $X \cap \chi(u) \neq \emptyset$; we let P_X be the set $\{u \mid u$ is v-problematic for $X\}$, A_X be the set $\{u_i \mid u \in P_X\}$ and we let D_X be the set $\{u_i \in A_X \mid \chi(u)^i \subseteq X\}$. We say that u_i in $A_X - D_X$ is *alive*. For every $x, y \in V_G$, every $X \subseteq V_G - \{x, y\}$ and every $u \in V_H$, we let

$$E(x, X) = \{xu_i \mid x \in \chi(u) \text{ and } p(G[\chi(u)]\backslash X, x, \chi(u)^i)\},$$

$$E(u, X) = \{u_i u_j \mid u_i, u_j \in A_X - D_X \text{ and } p(G[\chi(u)]\backslash X, \chi(u)^i, \chi(u)^j)\}$$

$$E(x, y, X) = \{u_i v_j \mid u_i, v_j \in A_{X \cup \{x,y\}} - D_X \text{ and } p(Skl(H, \chi)\backslash(P_{X \cup \{x,y\}} \cup A_{X \cup \{x,y\}}), u_i, v_j)\}$$

For every $x, y \in V_G$ and every $X \subseteq V_G - \{x, y\}$, the *problematic v-skeleton graph for* (x, y, X), denoted by $Skl_{(H,\chi)}(x, y, X)$, is the undirected graph with vertex set the set $\{x, y\} \cup (A_{X \cup \{x,y\}} - D_X)$ and edge set the set $E(x, X) \cup E(y, X) \cup E(x, y, X) \cup \bigcup_{\substack{u \in P_X \\ x, y \notin \chi(u)}} E(u, X)$.

Let (H, χ) be an $\mathcal{H}$-v-decomposition of an undirected graph G; let x and y be two vertices of G and let X be a subset of $V_G - \{x, y\}$. We let u and v be the two nodes of H such that x is in $\chi(u)$ and y is in $\chi(v)$. We explain informally what represent the edges of the undirected graph $K = Skl_{(H,\chi)}(x, y, X)$. The graph K is constructed with set of vertices the set of alive attachment-vertices and, two new vertices that represent respectively x and y and, that we still denote by x and y. We let $e = w_i - w_\ell$ be an edge in K, then we have several cases.

Case 1. It is of the form $x - u_i$ and is then included in $E(x, X)$. This edge means that there exists at least one z in $\chi(u)^i$ such that there exists a path in $G[\chi(u)]\backslash X$ between x and z. This represents *any* path in $G[\chi(u)]\backslash X$ from x and ending with a vertex in $\chi(u)^i$.

Case 2. It is of form $y - v_j$ and is then included in $E(y, X)$ and similarly to x, means that there exists at least one z' in $\chi(v)^j$ such that there is a path in $G[\chi(v)]\backslash X$ between y and z'.

Case 3. The edge e is not incident neither with x nor with y. We let s^i and s^ℓ be problematic block-vertices for $X \cup \{x, y\}$ such that $w_i = s^i_{j_i}$ and $w_\ell = s^\ell_{j_\ell}$. We have several possible meanings.

  Case 3.1. $s^i = s^\ell$. Then two cases:

    Case 3.1.1. The edge e is in $E(s^i, X)$. This represents *some* path in $G[\chi(s^i)]\backslash X$ starting from *some* vertex of G in $\chi(s^i)^{j_i}$ and ending in *some* vertex of G in $\chi(s^i)^{j_\ell}$, i.e., there exist some z_i in $\chi(s^i)^{j_i}$ and some z_ℓ in $\chi(s^i)^{j_\ell}$ such that there exists a path in $G[\chi(s^i)]\backslash X$ between z_i and z_ℓ. It is worth noticing that it may happen that there exists also some z'_i in $\chi(s^i)^{j_i}$ and some z'_ℓ in $\chi(s^i)^{j_\ell}$ such that there is no path in $G[\chi(s^i)]\backslash X$ between z'_i and z'_ℓ.

    Case 3.1.2. The edge e is in $E(x, y, X)$. This edge means there exists a path in $Skl_{(H,\chi)}$ between $w_i = s^i_{j_i}$ and $w_\ell = s^i_{j_\ell}$ that does not go through any problematic block-vertex. Since for each w in V_H, the sub-graph $G[\chi(w)]$ is connected, we have a path in $G\backslash X$ between *any* vertex z_i of G in $\chi(s^i)^{j_i}$ and *any* vertex z_ℓ of G in $\chi(s^i)^{j_\ell}$.

Case 3.2. $s^i \neq s^\ell$. The edge $w_i - w_\ell$ means there exists a path in $Skl_{(H,\chi)}$ between w_i and w_ℓ that does not go through any problematic block-vertex. Since the sub-graph $G[\chi(w)]$ of a block-vertex w is connected, we are sure that there exists a path between *any* vertex of G in $\chi(s^i)^{j_i}$ and *any* vertex of G in $\chi(s^\ell)^{j_\ell}$. These edges are included in the set $E(x, y, X)$.

It is not surprising from this informal presentation that if there exists a path in K between x and y, then there exists a path in $G\backslash X$ between x and y. We will prove that it is sufficient to look at the connectivity of x and y in K. Figure 49 shows the undirected graph $Skl_{(H,\chi)}(x,y,X)$ where x is in $\chi(a)$ and y is in $\chi(d)$. The v-problematic block-vertices are b, d and f and, d_1 is in D_X. The blue edges represent edges in $E(x,y,X)$ and black ones represent edges in $E(x,X) \cup E(y,X)$ and red ones edges in $E(u,X)$.

$$Skl_{(H,\chi)}(x,y,X)$$
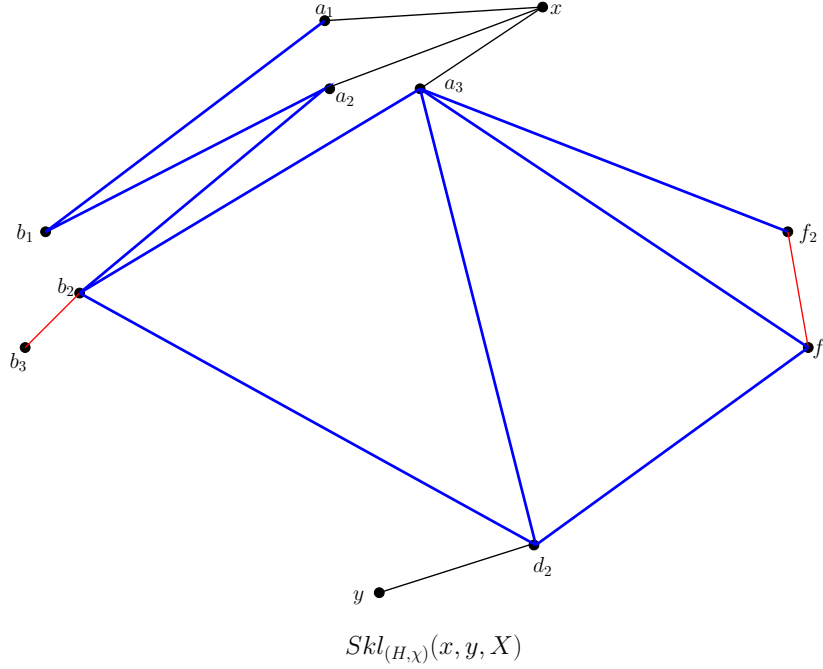
Figure 49: A problematic v-skeleton graph for (x,y,X) where (H,χ) is the $\mathcal{P}$-v-decomposition on Figure 46.

Remark 10.1 Let $\mathcal{H}$ be a class of undirected graphs and let (H,χ) be an $\mathcal{H}$-v-decomposition of $\mathcal{H}$-v-width ℓ of G. For every $x,y \in V_G$ and every $X \subseteq V_G - \{x,y\}$ the undirected graph $Skl_{(H,\chi)}(x,y,X)$ has at most $(\ell+1) \cdot |X|$ vertices. Therefore, the number of edges of $Skl_{(H,\chi)}(x,y,X)$ is at most $O(\ell^2 \cdot |X|^2)$.

We now prove that we can decide the connectivity of x and y in $G\backslash X$ by using $Skl_{(H,\chi)}(x,y,X)$. We have two cases: either x and y are in the same block-vertex (Lemma 10.6) or they belong to different block-vertices (Lemma 10.5). The proofs follow the cases presented informally above and show how to glue the paths represented by the edges in $Skl_{(H,\chi)}(x,y,X)$ in order to construct a path in $G\backslash X$ and conversely.

Lemma 10.5 *Let $\mathcal{H}$ be a class of undirected graphs and let (H,χ) be an $\mathcal{H}$-v-decomposition of an undirected graph G. For every $x,y \in V_G$ such that $x \in \chi(u), y \in \chi(v)$, $u,v \in V_H$, $u \neq v$ and every $X \subseteq V_G - \{x,y\}$, the vertices x and y are connected in $G\backslash X$ if and only if the vertices x and y are connected in $Skl_{(H,\chi)}(x,y,X)$.*

Proof. Let $K = Skl_{(H,\chi)}(x,y,X)$ and let $u \in V_H$ and $v \in V_H$ be such that $x \in \chi(u)$ and $y \in \chi(v), u \neq v$.

Assume first that x and y are connected in K and let $p = x - u_i - w_1 - w_2 - \cdots - w_p - v_j - y$ be a shortest path. The edge $x - u_i$ mean that $p(G[\chi(u)] \setminus (X \cap \chi(u)), x, \chi(u)^i)$ holds, say is of the form $x \overset{*}{-} z$ where $z \in \chi(u)^i$. The edge $u_i - w_1$ mean that there exists a path in $Skl_{(H,\chi)}$ that does not go through $P_{X \cup \{x,y\}} \cup A_{X \cup \{x,y\}}$, i.e., there exists a path in $G \setminus X$ between z and any vertex of $\chi(s^1)^{j_1}, w_1 = s^1_{j_1}$, say $z_1 \in \chi(s^1)^{j_1}$. Similarly for the edges $v_j - y$ and $w_p - v_j$, there exists a path in $G \setminus X$ between $z_p \in \chi(s^p)^{j_p}, w_p = s^p_{j_p}$ and $z' \in \chi(v)^j$ and between z' and y. We now prove that for every $1 \leq i \leq p-1$, there exists a path in $G \setminus X$ between a $z_i \in \chi(s^i)^{j_i}, w_i = s^i_{j_i}$ and a $z_{i+1} \in \chi(s^{i+1})^{j_{i+1}}, w_{i+1} = s^{i+1}_{j_{i+1}}$. For every $1 \leq i \leq p-1$, there exist 2 cases:

CASE 1. $s^i = s^{i+1}$. Then either the edge $w_i - w_{i+1}$ means that there exists a path which is contained in $G[\chi(s^i)] \setminus (X \cap \chi(s))$ or there exists a path in $Skl_{(H,\chi)}$ that does not go through $P_{X \cup \{x,y\}} \cup A_{X \cup \{x,y\}}$. In the two cases there exists a path between some vertex z_i of $\chi(s^i)^{j_i}$ and some vertex z_{i+1} of $\chi(s^i)^{j_{i+1}}$ in $G \setminus X$. Otherwise,

CASE 2. $s^i \neq s^{i+1}$. Then there exists a path in $Skl_{(H,\chi)}$ between w_i and w_{i+1} that does not go through $P_{X \cup \{x,y\}} \cup A_{X \cup \{x,y\}}$, i.e., there exists a path in $G \setminus X$ between any $z_i \in \chi(s^i)^{j_i}$ and any $z^{i+1} \in \chi(s^{i+1})^{j_{i+1}}$.

If $s^i = s^{i+1} = s^{i+2}$, then one and only of the edges $w_i - w_{i+1}$ and $w_{i+1} - w_{i+2}$ means there exists a path in $G[\chi(s^i)] \setminus (X \cap \chi(s^i))$, otherwise p is not the shortest one. For each such edge $w_i - w_{i+1}$, we choose z_i and z_{i+1} be any of the possible paths in $G[\chi(s^i)] \setminus (X \cap \chi(s^i))$. We can therefore choose the other vertices z_i of G such that we can concatenate all these paths in order to get a path in G.

Assume now that x and y are connected in $G \setminus X$ and let $w_1, \ldots, w_p$ be the attachment-vertices of the v-problematic block-vertices for X, enumerated in this order, met by a path p between x and y in $G \setminus X$ and assume that when a portion of p goes out of a block-vertex w and returns to it, there is no possible path inside the block-vertex. Let $i \leq bm_G(\chi(u))$ and $j \leq bm_G(\chi(v))$ be such that $p = x \overset{*}{-} z \overset{*}{-} z' \overset{*}{-} y$ and $z \in \chi(u)^i$ and $z' \in \chi(v)^j$. The portions $x \overset{*}{-} z$ and $z' \overset{*}{-} y$ are contained respectively in $G[\chi(u)] \setminus (X \cap \chi(u))$ and $G[\chi(v)] \setminus (X \cap \chi(v))$. Let $p = x \overset{*}{-} z \overset{*}{-} z_1 \overset{*}{-} z_2 \overset{*}{-} \cdots \overset{*}{-} z_p \overset{*}{-} z' \overset{*}{-} y$ where $z_i \in \chi(s^i)^{j_i}, w_i = s^i_{j_i}$. We have several cases.

CASE 3. $w_1 \neq u_i$ and $w_p \neq v_j$. Then u is not a v-problematic block-vertex for X and therefore $w_1 \neq u_k$ for all $k \leq bm_G(\chi(u))$. Therefore, $z \overset{*}{-} z_1$ does not go through $P_{X \cup \{x,y\}} \cup A_{X \cup \{x,y\}}$. By definition of K, this portion is represented by an edge $u_i - w_1$ in K. Similarly for w_p and v_j.

CASE 4. $w_1 = u_i$ and $w_p \neq v_j$. From Case 1 the portion $z_p \overset{*}{-} z'$ is represented by the edge $w_p - v_j$ in K. if $w_1 = u_i$, then $z_1 = z$ and this is represented by the edge $x - u_i$.

CASE 5. $w_1 \neq u_i$ and $w_p = v_j$. This is similar to Case 2 where $z \overset{*}{-} z_1$ is represented by the edge $u_i - w_1$ and $z_p \overset{*}{-} y$ is represented by $v_j - y$.

CASE 6. $w_1 = u_i$ and $w_p = v_j$. Again $z_p = z'$ and $z_1 = z$ and the portions $x \overset{*}{-} z \overset{*}{-} z_1$ and $z_p \overset{*}{-} z' \overset{*}{-} y$ are represented by the edge $x - u_i$ and $y - v_j$.

We claim now that the portions $z_i \overset{*}{-} z_{i+1}$ for $1 \le i \le p - 1$ are represented by the edges $w_i - w_{i+1}$ in K. We have two cases:

CASE 7. $s^i = s^{i+1}$. Then either the portion $z_i \overset{*}{-} z_{i+1}$ is contained in $G[\chi(s^i)] \backslash (X \cap \chi(s))$ or does not go through any vertex of problematic block vertices. In the two cases there exists an edge $w_i - w_{i+1}$ in K. Otherwise,

CASE 8. $s_i \ne s^{i+1}$. In this case the portion $z_i \overset{*}{-} z_{i+1}$ does not go through any vertex of problematic block vertices. By definition, there exists an edge $w_i - w_{i+1}$ in K.

Therefore, x and y are connected in K, which ends the proof. ∎

Lemma 10.6 *Let $\mathcal{H}$ be a class of undirected graphs and let (H, χ) be an $\mathcal{H}$-v-decomposition of $\mathcal{H}$-v-width at most ℓ of an undirected graph G. Let the vertices x and y of G be such that $x \in \chi(u), y \in \chi(u)$, $u \in V_H$ and let $X \subseteq V_G - \{x, y\}$. Assume the followings:*

(i) For each attachment-vertex u_i of u, we know $|N_{Skl_{(H,\chi)}}(u_i) \backslash D_X|$.

(ii) For each $z \in \{x, y\}$ and each u_i, we know if there exists a path in $G[\chi(u)] \backslash (X \cap \chi(u))$ between z and some vertex of $\chi(u)^i$.

(iii) For every u_i, u_j, we know if there exists a path in $G[\chi(u)] \backslash (X \cap \chi(u))$ between some vertex of $\chi(u)^i$ and some vertex of $\chi(u)^j$.

(iv) We know if there there exists a path between x and y in $G[\chi(u)] \backslash (X \cap \chi(u))$.

Then we can decide if x and y are connected by a path in $G \backslash X$ in $O(\ell^2 \cdot |X|^2)$-time.

Proof. Let $K = Skl_{(H,\chi)}$ and let x and y be connected in $G \backslash X$. Let p be a path between x and y in $G \backslash X$. Then p can only be one of the following forms:

CASE 1. p is contained in $G[\chi(u)] \backslash (X \cap \chi(u))$.

CASE 2. $p = x \overset{*}{-} z - z_1 - z' \overset{*}{-} y$ where $z, z' \in \chi(u)^i$ for some $i \le \ell$ and $z_1 \in \chi(v)^j$, $v \ne u$ and $j \le bm_G(\chi(v))$ where $v_j \in V_K - D_X$.

CASE 3. $p = x \overset{*}{-} z_{i_1} \overset{*}{-} z_{i_2} \overset{*}{-} \cdots \overset{*}{-} z_{i_p} \overset{*}{-} y$ where $z_{i_j} \in \chi(u)^{i_j}$ and $z_{i_s} \overset{*}{-} z_{i_{s+1}}$ is either contained in $G[\chi(u)] \backslash (X \cap \chi(u))$ or does not go through u and $x \overset{*}{-} z_1$ and $z_{i_p} \overset{*}{-} y$ are contained in $G[\chi(u)] \backslash (X \cap \chi(u))$ and $z_{i_{p-1}} \overset{*}{-} z_{i_p}$ does not go through u.

CASE 4. $p = x \overset{*}{-} z_{i_1} \overset{*}{-} z_{i_2} \overset{*}{-} \cdots \overset{*}{-} z_{i_p} - z - z_{i_{p+1}} \overset{*}{-} y$ where $z_{i_j} \in \chi(u)^{i_j}$, $z_{i_s} \overset{*}{-} z_{i_{s+1}}$ is either contained in $G[\chi(u)] \backslash (X \cap \chi(u))$ or does not go through u and $x \overset{*}{-} z_1$, $z_{i_{p-1}} \overset{*}{-} z_{i_p}$ and $z_{i_{p+1}} \overset{*}{-} y$ are contained in $G[\chi(u)] \backslash (X \cap \chi(u))$ and $z \in \chi(v)^j$, $v \neq u$ and $j \leq bm_G(\chi(v))$ with $v_j \in V_K - D_X$.

Since we can decide the existence of a path between x and some vertex of $\chi(u)^i$ and for each u_i, we know $|N_{Skl_{(H,\chi)}}(u_i) \backslash D_X|$, we can decide if Case 1 or Case 2 holds. (Note that Case 2 is a special case of Case 4.)

One can verify that for every attachment-vertices u_i and u_j of u, if there exists a path in $G \backslash X$ between a vertex of $\chi(u)^i$ and a vertex of $\chi(u)^j$, this path is either included in $G[\chi(u)] \backslash (X \cap \chi(u))$ or there exists a path in $Skl_{(H,\chi)}(x, y, X)$ between u_i and u_j that does no go through u. Conversely, if there exists a path between u_i and u_j in $Skl_{(H,\chi)}(x, y, X)$ that does not go through u, then there exists a path between every vertex of $\chi(u)^i$ and every vertex of $\chi(u)^j$ in $G \backslash X$.

By using $Skl_{(H,\chi)}(x, y, X)$ we can verify if there exists a path between u_i and u_j in $Skl_{(H,\chi)}(x, y, X)$ that does not go through u in $O(|X|^2)$-time (we can search a path in a graph in $O(n + m)$-time where m is the number of edges and n the number of vertices). Since we know for every u_i, u_j if there exists a path in $G[\chi(u)] \backslash (X \cap \chi(u))$ between some vertex of $\chi(u)^i$ and some vertex of $\chi(u)^j$, we can therefore decide if Case 3 holds. We can also decide if a path of the form of Case 4 exists by using in addition $|N_{Skl_{(H,\chi)}}(u_i) \backslash D_X|$ for each u_i.

Conversely, it is clear that if a path of the Case 1-4 is found, then the vertices x and y are connected in $G \backslash X$. This ends the proof.   ∎

We now introduce the last notations. Let P_1 and P_2 be the following properties:

- $P_1(x, X_1, X)$ = "there exists a path between x and a vertex in X_1 that avoids the vertices of X".

- $P_2(X_1, X_2, X)$ = "there exists a path between a vertex in X_1 and a vertex in X_2 that avoids the vertices of X".

The constraints on the v-skeletons and on the sub-graphs induced by the block-vertices are formalized in the following definition.

Definition 10.6 (Constraint $\mathcal{H}$-v-Decompositions) Let $\mathcal{H}, \mathcal{D}_1$ and $\mathcal{D}_2$ be 3 classes of undirected graphs. We say that a class $\mathcal{C}$ of undirected graphs has an $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-v-decomposition of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-v-width at most ℓ if every graph G in $\mathcal{C}$ has an $\mathcal{H}$-v-decomposition (H, χ) of $\mathcal{H}$-v-width at most ℓ such that $Skl_{(H,\chi)} \in \mathcal{D}_1$ and for every $u \in V_H$, the sub-graph $G[\chi(u)] \in \mathcal{D}_2$.

We now state and prove the main theorem of this section.

Theorem 10.1 *Let ℓ be a positive integer and let $\mathcal{H}, \mathcal{D}_1$ and $\mathcal{D}_2$ be classes of undirected graphs. Assume the followings:*

1. *There exists an f_1-labeling scheme for the connectivity query on $\mathcal{D}_1$ and for every $G \in \mathcal{D}_1$, the f_1-labeling on G is constructed in $g_1(|V_G|)$-time.*

2. *There exists an f_2-labeling scheme for the adjacency query on $\mathcal{D}_1$ and for every $G \in \mathcal{D}_1$, the f_2-labeling on G is constructed in $g_2(|V_G|)$-time.*

3. *There exist f_3-labeling schemes for the properties P_1 and P_2 on $\mathcal{D}_2$ and for every $G \in \mathcal{D}_2$, the f_3-labelings on G are constructed in $g_3(|V_G|)$-time.*

Then there exists an $O(\ell^2 \cdot (f_1 + f_2 + f_3))$-labeling scheme $(\mathcal{A}, \mathcal{B})$ for the connectivity query on the class $\mathcal{C}$ of undirected graphs of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-v-width at most ℓ. Moreover, if for every graph $G \in \mathcal{C}$ with n vertices, we can construct an $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-decomposition of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-v-width at most ℓ in $f(n)$, then for every graph $G \in \mathcal{C}$, the algorithm $\mathcal{A}$ computes the labels in $\max\{f(n), O(n \cdot g_3(n)), g_1(n), g_2(n), O(n^2)\}$-time and $\mathcal{B}$ gives the answer in $O(\ell^2 \cdot m^2)$-time where m is the size of the data.

Proof. Let $G \in \mathcal{C}$ be an undirected graph with n vertices. We compute in $f(n)$-time an $\mathcal{H}$-v-decomposition (H, χ) of G such that $Skl_{(H,\chi)} \in \mathcal{D}_1$ and for every $u \in V_H$, $G[\chi(u)] \in \mathcal{D}_2$. It is clear by definition of an $\mathcal{H}$-v-decomposition that $|V_H| \leq n$ and then $Skl_{(H,\chi)}$ has $O(n)$ vertices. We can construct the pseudo-modules of all $\chi(u)$, $u \in V_H$. Therefore, we can construct the v-skeleton $Skl_{(H,\chi)}$ of (H, χ).

By hypothesis, there exists an f_1-labeling K for the connectivity query on $Skl_{(H,\chi)}$. By hypothesis, for each $i, j \leq \ell$ and each $u \in V_H$, we can construct f_3-labelings for $P_i = P_1(x, \chi(u)^i, X)$ and $P_{i,j} = P_2(\chi(u)^i, \chi(u)^j, X)$ on $G[\chi(u)]$, that we denote by $J_{i,u}$ and $J_{i,j,u}$. It is worth noticing that in P_i there are two free variables, x and X and, in $P_{i,j}$, there is one free variable, X. There exists an f_2-labeling K' for the adjacency query on $Skl_{(H,\chi)}$ by hypothesis. We compute the degree of each node $w \in Skl_{(H,\chi)}$ that we denote by $d(w)$. For each $u \in V_H$ and each $i \leq \ell$, we let $Card(u, i) = |\chi(u)^i|$. For every $x \in V_G$ such that $x \in \chi(u)^i$, we let

$$C(x) = \Big(Card(u, 1), \ldots, Card(u, \ell), d(u_1), \ldots, d(u_\ell)\Big)$$

$$C'(x) = \Big(K'(u_1), \ldots, K'(u_\ell), i\Big),$$

$$L(x) = \Big(J_{1,u}(x), \ldots, J_{\ell,u}(x), J_{1,1,u}(x), \ldots, J_{1,\ell,u}(x), \ldots, J_{\ell,1,u}(x), \ldots, J_{\ell,\ell,u}(x)\Big)$$

$$J(x) = \Big(K(u), K(u_1), \ldots, K(u_\ell), C(x), C'(x), L(x)\Big)$$

It is clear that $|J(x) = O(\ell^2 \cdot (f_1(n) + f_2(n) + f_3(n) + f_4(n)))$ for every $x \in V_G$. We now explain given $J(x), J(y)$ and $J(X)$ how to verify if x and y are connected in $G \backslash X$.

For each $z \in V_G$, by using the K-part of $J(z)$ we can determine the block-vertices $u \in V_H$ such that $z \in \chi(u)$. We can therefore determine the set P_X of v-problematic block-vertices for X and the block-vertices u and v such that $x \in \chi(u)$ and $y \in \chi(v)$. For each $w \in P_X \cup \{u, v\}$, we can determine the set $X_w = \{x \mid x \in \chi(w)\}$. By comparing the K-parts of $J(x)$ and of $J(y)$ we can decide if $u = v$ or $u \neq v$. Let us first construct $Skl_{(H,\chi)}(x, y, X)$. For each $w \in P_X \cup \{u, v\}$, by using $Card(u, i)$ of some vertex $z \in X_w$ and by counting the number of elements in X_w that has the color i, we can decide if $w_i \in D_X$ or not. Then we can determine the set D_X and therefore the vertices of $Skl_{(H,\chi)}(x, y, X)$.

For every u_i, we can determine if there exists a path in $G[\chi(u)] \backslash X_u$ between some vertex of $\chi(u)^i$ and x by using $J_{i,u}(x)$ and $J_{i,u}(X_u)$. Similarly for y and each v_i by using $J_{i,v}(y)$ and $J_{i,v}(X_v)$. Then we can construct $E(x, X)$ and $E(y, X)$.

For each $w \in P_X \cup \{u, v\}$, by using $J_{i,j,w}(X_w)$ we can determine if there exists a path in $G[\chi(w)] \backslash X_w$ between some vertex of $\chi(w)^i$ and some vertex of $\chi(w)^j$. Then we can construct the set $E(w, X)$. By using $K(w_i)$, $K(w_j)$ of some vertex $z \in X_w$ and $K(P_X \cup \{u, v\})$ we can also determine if there exists a path in K that avoids $P_X \cup \{u, v\}$, i.e., we can construct $E(x, y, X)$. Therefore, we can construct the graph $Skl_{(H,\chi)}(x, y, X)$ since we know how to construct its set of edges.

If $u \neq v$, then by Lemma 10.5 the vertices x and y are connected in $G \backslash X$ if and only if x and y are connected in $Skl_{(H,\chi)}(x, y, X)$.

If $u = v$, then by Lemma 10.6 it remains to know $|N_{Skl_{(H,\chi)}}(u_i) \backslash D_X|$ (D_X is the set of nodes w_i in $Skl_{(H,\chi)}$ such that $w_i = s_{j_i}^i$ and $\chi(s^i)^{j_i} \subseteq X$) for each $u_i = v_i$. By using $d(u_1), \ldots, d(u_\ell)$ in $J(x)$ we can determine the degree of each u_i. For each $w_j \in D_X$, we know $g_1(w_j), g_2(w_j)$ and $g_3(w_j)$ since there is at least one $z \in X_w$ such that $z \in \chi(w)^j$. By using $g_1(u_i), g_2(u_1), g_3(u_1)$ and $g_1(w_j), g_2(w_j), g_3(w_j)$ we can decide if u_i and w_j are adjacent or not. Therefore, by comparing $d(u_i)$ and $|\{w_j \in D_X \mid u_i \text{ and } w_j \text{ are adjacent in } Skl_{(H,\chi)}\}|$ we can determine $|N_{Skl_{(H,\chi)}}(u_i) \backslash D_X|$.

As a consequence of the assumptions, Lemmas 10.5 and 10.6, and the definition of $Skl_{(H,\chi)}(x, y, X)$, we can decide the connectivity of x and y in $G \backslash X$ in $O(\ell^2 \cdot |X|^2)$.

It remains to bound the time for constructing the labeling J. Since $Skl_{(H,\chi)}$ has $O(n)$ vertices we can construct, by hypothesis, the labeling K and K' in $g_1(n)$-time and $g_2(n)$-time respectively. For each $u \in V_H$ we construct, by hypothesis, the labelings $J_{i,u}$ and $J_{i,u,j}$ in $O(g_3(n_u))$ where n_u is the number of vertices of $G[\chi(u)]$. Then we construct the labelings $J_{i,u}$ and $J_{i,j,u}$ for all $u \in V_H$ in $O(n \cdot g_3(n))$-time. We compute the pseudo-modules of all $\chi(u)$, $u \in V_H$ in $O(n^2)$-time (Corollary 10.1). It is clear that if we know all the pseudo-modules we can determine for all u and all i the number $Card(u, i)$ in $O(n)$-time. We can construct the degree of all vertices of $Skl_{(H,\chi)}$ in $O(m)$-time (m is the number of edges) as follows:

1. Initialize the degree of all the vertices of G to 0.

2. For each edge $e = xy$ of G, increase by 1 the degree of x and y.

We can therefore construct the labeling in $\max\{f(n), O(n \cdot g_3(n)), g_1(n), g_2(n), O(n^2)\}$-time. This finishes the proof.   ∎

As a corollary of Theorems 7.4, 9.1 and 10.1 we have the following which concerns *planar gluings* of graphs of small clique-width.

Corollary 10.2 *Let k and ℓ be positive integers and let $\mathcal{H}$ be a class of undirected graphs. There exists an $O(\ell^2 \cdot f(k) \cdot \log)$-labeling scheme $(\mathcal{A}, \mathcal{B})$ for the connectivity query on the class $\mathcal{C}$*

of undirected graphs of $(\mathcal{H}, \mathcal{P}, CWD(\leq k))$-width at most ℓ where $f(k)$ is the hidden constant on the labeling for connectivity queries on graphs in $CWD(\leq k)$. Moreover, if for every $G \in \mathcal{C}$ with n vertices we can construct an $(\mathcal{H}, \mathcal{P}, CWD(\leq k))$-decomposition of $(\mathcal{H}, \mathcal{P}, CWD(\leq k))$-width at most ℓ in $f(n)$-time, then for every $G \in \mathcal{C}$ with n vertices $\mathcal{A}$ computes the labels in $\max\{f(n), O(n^4)\}$-time.

10.4 $\mathcal{H}$-e-Decompositions of Undirected Graphs

In [Sch97, WT07] the notion of tree-width is generalized in such a way that instead of imposing the decomposition to be a tree, it can be in any other class of graphs, that is specified. For instance, in [WT07] a notion of planar-decomposition and of planar-width is studied and they show that graphs embeddable in a surface have such bounded planar-width. We use the same ideas. However, in the definitions of [Sch97, WT07] the parameter is the size of the blocks and a vertex may appear in an unbounded number of parts. This definition is not suitable for our purposes because in our proof techniques we cannot handle decompositions where a vertex may appear in an unbounded number of parts. For that purposes, we introduce decompositions based on edge-partitions with a parameter that measures the number of parts that contain a vertex and the number of vertices a block shares with other parts.

We define the notions of $\mathcal{H}$-*e-decomposition* and of $\mathcal{H}$-*e-width*. The axioms of the definition are also present in [Sch97].

Definition 10.7 ($\mathcal{H}$-e-Decomposition and $\mathcal{H}$-e-Width) Let $\mathcal{H}$ be a class of undirected graphs. An $\mathcal{H}$-*e-decomposition* of an undirected graph G is a pair (H, χ) such that:

(CD1) $H \in \mathcal{H}$ and $\chi : V_H \to 2^{V_G}$.

(CD2) $\bigcup_{u \in V_H} \chi(u) = V_G$.

(CD3) For every $u \in V_H$, the sub-graph $G[\chi(u)]$ is connected.

(CD4) For every $u, v \in V_H$, if $uv \in E_H$, then $\chi(u) \cap \chi(v) \neq \emptyset$.

(CD5) For every edge $xy \in E_G$, there exists $u \in V_H$ such that $x, y \in \chi(u)$:

(CD6) For every $x \in V_G$, the sub-graph $H[\{u \mid x \in \chi(u)\}]$ is connected.

For every $u \in V_H$, we let $sh(u) = \left| \chi(u) \cap \left(\bigcup_{v \in V_H, v \neq u} \chi(v) \right) \right|$ and we say that u *shares* $sh(u)$ vertices, that we denote by $x_1(u), \ldots, x_{sh}(u)$.

The $\mathcal{H}$-*spread* of a $\mathcal{H}$-e-decomposition (H, χ), denoted by $\mathcal{H}$-$spr(H, \chi)$, is $\max\limits_{x \in V_G} \left| \{u \in V_H \mid x \in \chi(u)\} \right|$ and the $\mathcal{H}$-*block-width* of a $\mathcal{H}$-e-decomposition (H, χ), denoted by $\mathcal{H}$-$blk(H, \chi)$, is $\max\{sh(u) \mid u \in V_H\}$.

The $\mathcal{H}$-*e-width* of an $\mathcal{H}$-e-decomposition (H, χ), denoted by c-$\mathcal{H}$-$wd(H, \chi)$, is $\max\{\mathcal{H}$-$spr(H, \chi), \mathcal{H}$-$blk(H, \chi)\}$. The $\mathcal{H}$-*e-width* of an undirected graph, denoted by c-$\mathcal{H}$-$wd(G)$, is the minimum $\mathcal{H}$-e-width over all $\mathcal{H}$-e-decompositions of G.

If (H,χ) is an $\mathcal{H}$-e-decomposition of a graph G, we call the vertices of H *nodes* in order to distinguish them from the vertices of G.

Figure 50 shows an example of a $\mathcal{P}$-e-decomposition. If two nodes u and v share a vertex and are adjacent, we represent the two vertices in the two blocks and join them by a dashed line. For instance, x_2 is in $\chi(a) \cap \chi(b) \cap \chi(c) \cap \chi(g)$. There exists an edge in H between a and b, then there exists a dashed line between the x_2's in a and in b. There is no edge between a and c, then there is no dashed line between the x_2's in a and in c. We can verify that for every vertex of G, the sub-graph $H[\{u \mid x \in \chi(u)\}]$ is connected. We omit the edges in the blocks and the vertices that belong to one $\chi(u)$, for $u \in V_H$.

Figure 50: A $\mathcal{P}$-e-decomposition of $\mathcal{P}$-e-width 4 of an undirected graph G.

Remark 10.2 Let $\mathcal{H}$ be a class of undirected graphs and let (H,χ) be an $\mathcal{H}$-e-decomposition of an undirected graph G of $\mathcal{H}$-e-width at most ℓ. Then every node u of H has degree at most ℓ^2. If $sh(u,v) = |\chi(u) \cap \chi(v)| \geq 1$, then we say that u and v *share* $sh(u,v)$ vertices, that we denote by $x_1(u,v), \ldots, x_{sh(u,v)}(u,v)$.

In order to prove that some graph classes of bounded $\mathcal{H}$-e-width admits short labeling scheme for connectivity query, we follow the same proof techniques as in Section 10.3. For the same reasons, we cannot use the quotient graph and we define the notions of *e-skeleton* and of *problematic e-block-vertices*.

Definition 10.8 (e-Skeleton and Problematic e-Block-Vertices) Let $\mathcal{H}$ be a class of undirected graphs and let (H, χ) be an $\mathcal{H}$-e-decomposition of $\mathcal{H}$-e-width ℓ of an undirected graph G. For every $u \in V_H$, we let $C(u) = \bigcup_{v \in N_H(u)} \{c_1(u,v), \ldots, c_{sh(u,v)}(u,v)\}$. The *e-skeleton* of (H, χ), denoted by $eSkl_{(H,\chi)}$, is the undirected bipartite graph $\langle V_{eSkl_{(H,\chi)}}, E_{eSkl_{(H,\chi)}} \rangle$ where:

$$V_{eSkl_{(H,\chi)}} = \bigcup_{u \in V_H} (\{u\} \cup C(u)),$$

$$E_{eSkl_{(H,\chi)}} = \bigcup_{u \in V_H} \bigcup_{v \in N_H(u)} \{u - c_1(u,v), \ldots, u - c_{sh(u,v)}(u,v)\} \cup$$

$$\bigcup_{uv \in E_H} \{c_i(u,v) - c_j(v,u) \mid x_i(u,v) = x_j(v,u)\}.$$

For every node $u \in V_H$, we call $u \in V_{eSkl_{(H,\chi)}}$ an *e-block-vertex* and we call the vertices u_{v_i} *e-attachment-vertices*.

For every $X \subseteq V_G$, an e-block-vertex u is said *problematic for X* if $\chi(u) \cap X \neq \emptyset$; We let $P_X = \{u \mid u \text{ is problematic for } X\}$, $P'_X = \{u \mid \exists w (\chi(u) \cap \chi(w) \neq \emptyset \text{ and } w \in P_X\}$, $A_X = \{C(u) \mid u \in P_X\}$, $A'_X = \{C(u) \mid u \in P'_X\}$ and $D_X = \{c_i(u,v) \in A_X \cup A'_X \mid \exists z (z \in X \text{ and } z = x_i(u,v))\}$. For every $x, y \in V_G$, every $X \subseteq V_G - \{x, y\}$ and every $u \in V_H$, we let

$$E(x, X) = \bigcup_{\substack{u \in V_H \\ x \in \chi(u)}} \bigcup_{v \in N_H(u)} \{x - c_i(u,v) \mid p(G[\chi(u)] \backslash X, x, x_i(u,v))\},$$

$$E(u, X) = \{c_i(u, v_\ell) - c_j(u, v_s) \mid c_i(u, v_\ell), c_j(u, v_s) \notin D_X \text{ and}$$
$$\big(p(G[\chi(u)] \backslash X, x_i(u, v_\ell), x_j(u, v_s)) \text{ or } x_i(u, v_\ell) = x_j(u, v_s)\big)\},$$

$$E(x, y, X) = \{c_i(u, v) - c_j(w, s) \mid c_i(u, v), c_j(w, s) \notin D_X \text{ and}$$
$$\big(p(eSkl_{(H,\chi)} \backslash (P_X \cup A_X), c_i(u, v), c_j(w, s)) \text{ or } x_i(u, v) = x_j(w, s)\big)\}.$$

For every $x, y \in V_G$ and every $X \subseteq V_G - \{x, y\}$, the *problematic e-skeleton graph for (x, y, X)*, denoted by $eSkl_{(H,\chi)}(x, y, X)$, is the undirected graph $\langle V_{eSkl_{(H,\chi)}(x,y,X)}, E_{eSkl_{(H,\chi)}(x,y,X)} \rangle$ where:

$$V_{eSkl_{(H,\chi)}(x,y,X)} = \{x, y\} \cup (A'_{X \cup \{x,y\}} \cup A_{X \cup \{x,y\}}) - D_X,$$

$$E_{eSkl_{(H,\chi)}(x,y,X)} = E(x, X) \cup E(y, X) \cup E(x, y, X) \cup \bigcup_{u \in P_{X \cup \{x,y\}} \cup P'_{X \cup \{x,y\}}} E(u, X).$$

Let $\mathcal{H}, \mathcal{D}_1$ and $\mathcal{D}_2$ be 3 classes of undirected graphs. We say that a class $\mathcal{C}$ of undirected graphs has an $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-*e-decomposition* of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-*e-width* at most ℓ if every graph $G \in \mathcal{C}$ has a $\mathcal{H}$-e-decomposition (H, χ) of $\mathcal{H}$-e-width at most ℓ such that $eSkl_{(H,\chi)} \in \mathcal{D}_1$ and for every $u \in V_H$ the sub-graph $G[\chi(u)] \in \mathcal{D}_2$.

For constructing the e-skeleton, we duplicate the shared vertices and two duplicated vertices are adjacent if they represent the same vertex in G and the two e-block-vertices, to which they belong, are adjacent. For instance, on Figure 50, the e-block-vertex b shares the vertices x_2 with the e-block-vertices a, c and g and is adjacent with all of them, a is not adjacent with c and c is not adjacent with g. We create 3 copies of x_2 adjacent with b, that we denote for clarity by $x_2^{b,a}$, $x_2^{b,g}$, $x_2^{b,c}$, one copy of x_2 adjacent with c, denoted by $x_2^{c,b}$, two copies of x_2

adjacent with a, denoted by $x_1^{a,b}$ and $x_2^{a,g}$ and two copies of x_2 adjacent with g, denoted by $x_2^{g,a}$ and $x_2^{g,b}$. Then for each z and t in $\{a, b, c, g\}$, we create an edge between $x_2^{z,t}$ and $x_2^{t,z}$ if zt is an edge of H. One can verify that if $\mathcal{H}$ is planar, then the e-skeleton is also planar.

Figure 51 shows the e-skeleton associated with the $\mathcal{P}$-e-decomposition on Figure 50. Red edges represent edges between e-block-vertices and their associated e-attachment-vertices and blue edges, the edges between e-attachment-vertices that are copies of the same vertex in G.
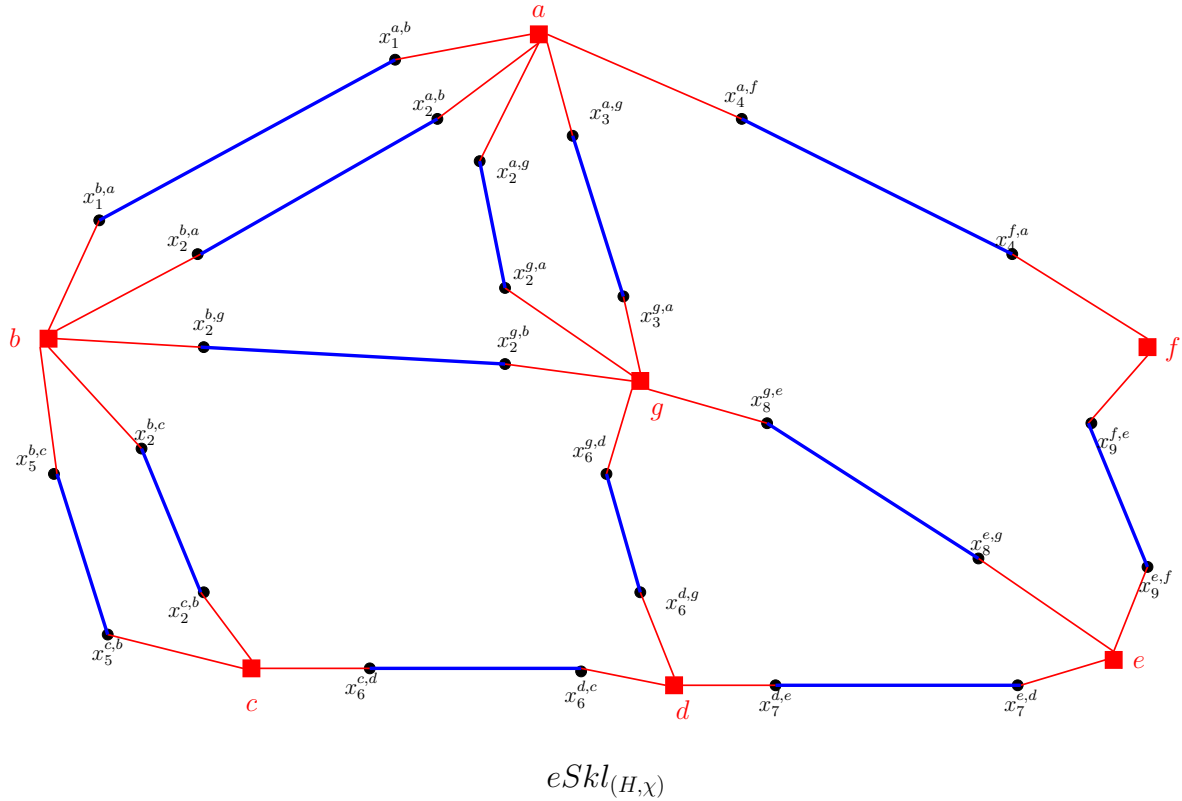

$$eSkl_{(H,\chi)}$$

Figure 51: The associated undirected graph $eSkl_{(H,\chi)}$ of the $\mathcal{P}$-e-decomposition on Figure 50.

Figure 52 shows the problematic e-skeleton graph associated with (x, y, X) where $x \in \chi(f)$, $y \in \chi(e)$, $x_9 \in X$. The graph induced on I and II are cliques. For clarity, we do not show all the edges. For instance, we do not show the edge between $x_1^{a,b}$ in I and $x_6^{g,d}$ in II; this edge can represent the path $x_1^{a,b} - a - x_3^{a,g} - x_3^{g,a} - g - x_6^{g,d}$ or any other path that does not go through e and f. Red edges represent paths inside e-block-vertices avoiding vertices in X and the blue ones represent paths in $eSkl_{(H,\chi)}$ that do not go through problematic e-block-vertices.

Remark 10.3 Notice that if (H, χ) is a $\mathcal{H}$-e-decomposition of $\mathcal{H}$-e-width at most ℓ of an undirected n-vertex graph G, then the number of vertices of $eSkl_{(H,\chi)}$ is $O(\ell^3 \cdot n)$ and for every $x, y \in V_G$ and $X \subseteq V_G - \{x, y\}$, the graph $eSkl_{(H,\chi)}(x, y, X)$ has at most $2 + \ell^3 \cdot (|X| + 2)$ nodes.

In the problematic e-skeleton, an edge w and w' either means that w and w' represent the same vertex or there exists a path that does not go through any problematic e-block-vertex or there exists a path in $G[\chi(s^i)] \backslash X$ between the vertex represented by w and the vertex
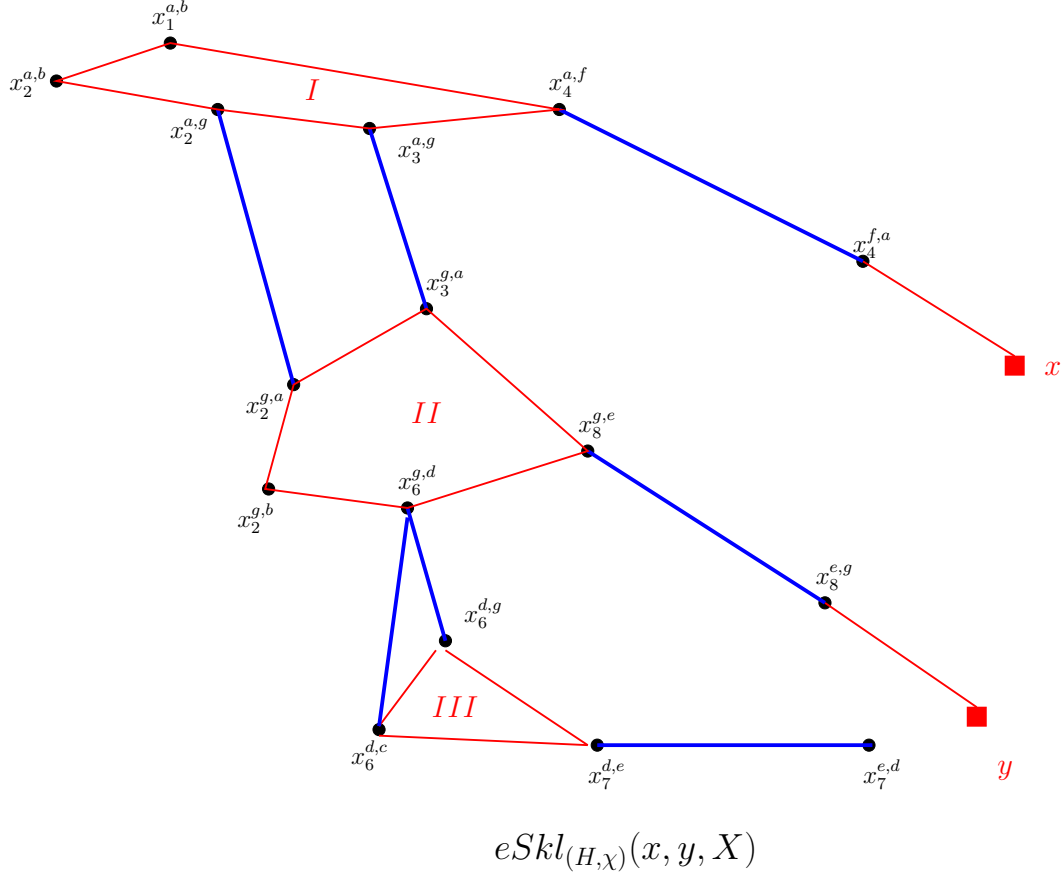
$$eSkl_{(H,\chi)}(x,y,X)$$

Figure 52: A problematic e-skeleton graph for (x,y,X) where (H,χ) is the $\mathcal{P}$-e-decomposition on Figure 50.

represented by w'. We relate the connectivity between two vertices in sub-graphs of G with their connectivity in associated problematic e-skeletons.

Lemma 10.7 *Let $\mathcal{H}$ be a class of undirected graphs and let (H,χ) be a $\mathcal{H}$-e-decomposition of $\mathcal{H}$-e-width at most ℓ of an undirected graph G. For every $x,y \in V_G$ and every $X \subseteq V_G - \{x,y\}$, the vertices x and y are connected in $G \backslash X$ if and only if the vertices x and y are connected in $eSkl_{(H,\chi)}(x,y,X)$.*

Proof. Let $K = eSkl_{(H,\chi)}(x,y,X)$ and assume first that x and y are connected in K by a path $p = x - c_i(u,s) - c_{j_1}(w_1,s_1) - \cdots - c_{j_p}(w_p,s_p) - c_j(v,s') - y$, as shortest as possible. The edge $x - c_i(u,s)$ means there exists a path in $G[\chi(u)] \backslash (X \cap \chi(u))$ between x and $x_i(u,s)$. Therefore, the edge $c_i(u,s) - c_{j_1}(w_1,s_1)$ means either $x_i(u,s) = x_{j_1}(w_1,s_1)$ or there exists a path $c_i(u,s) \overset{*}{-} c_{j_1}(w_1,s_1)$ in $eSkl_{(H,\chi)}$ that does not go through $P_X \cup A_X$, i.e., is a path $x_i(u,s) \overset{*}{-} x_{j_1}(w_1,s_1)$ in $G \backslash X$. Similarly, for the edges $c_{j_p}(w_p,s_p) - c_j(v,s')$ and $c_j(v,s') - y$, there exists a path $x_{j_p}(w_p,s_p) \overset{*}{-} x_j(v,s') \overset{*}{-} y$ in $G \backslash X$. We now prove that for every $1 \le i \le p-1$, the edge $w_i - w_{i+1}$ is a path $x_{j_i}(w_i,s_i) \overset{*}{-} x_{j_{i+1}}(w_{i+1},s_{i+1})$ in $G \backslash X$:

CASE 1. $x_{j_i}(w_i, s_i) = x_{j_{i+1}}(w_{i+1}, s_{i+1})$. Then the edge $w_i - w_{i+1}$ is an empty path in $G \backslash X$.

CASE 2. $x_{j_i}(w_i, s_i) \neq x_{j_{i+1}}(w_{i+1}, s_{i+1})$.   Then either there exists a path $x_{j_i}(w_i, s_i) \overset{*}{-} x_{j_{i+1}}(w_{i+1}, s_{i+1})$ in $G[\chi(w_i)] \backslash X = G[\chi(w_{i+1})] \backslash X$ or there exists a path in K that does not go through $P_X \cup A_X$, i.e., there exists a path $x_{j_i}(w_i, s_i) \overset{*}{-} x_{j_{i+1}}(w_{i+1}, s_{i+1})$ in $G \backslash X$.

By concatenating the paths $x \overset{*}{-} x_i(u, s) \overset{*}{-} x_{j_1}(w_1, s_1)$, $x_{j_p}(w_p, s_p) \overset{*}{-} x_j(v, s') \overset{*}{-} y$ and the paths $x_{j_i}(w_i, s_i) \overset{*}{-} x_{j_{i+1}}(w_{i+1}, s_{i+1})$, we get a path $x \overset{*}{-} y$ in $G \backslash X$.

Assume now that x and y are connected in $G \backslash X$ by a path p such that if a portion of p goes out of an e-block-vertex u from z and returns to it in t, there is no possible path between z and t in $G[\chi(u)] \backslash (X \cap \chi(u))$. Let $p = x \overset{*}{-} z_1 \overset{*}{-} z_2 \overset{*}{-} \cdots \overset{*}{-} z_m \overset{*}{-} y$ such that $z_1 = x_{j_1}(w_1, s_1), \ldots, z_m = x_{j_m}(w_m, s_m)$ and $c_{j_1}(w_1, s_1), \ldots, c_{j_m}(w_m, s_m)$ are the e-attachment-vertices of the problematic e-block-vertices met by p in this order. There is clearly an e-block-vertex u that contains x and a vertex $z = x_i(u, s)$ such that $x \overset{*}{-} x_1 = x \overset{*}{-} z \overset{*}{-} x_1$ where $x \overset{*}{-} z$ is contained in $G[\chi(u)] \backslash (X \cap \chi(u))$. Similarly for y, there is an e-block-vertex v that contains y and a vertex $z' = x_j(v, s')$ such that $x_m \overset{*}{-} y = x_m \overset{*}{-} z' \overset{*}{-} y$ where $z' \overset{*}{-} y$ is contained in $G[\chi(v)] \backslash (X \cap \chi(v))$. We have four cases:

CASE 3. $x_1 = z$ and $x_m = z'$. In this case the paths $x \overset{*}{-} z_1$ and $x_m \overset{*}{-} y$ are represented by the edges $x - c_i(u, s)$ and $y - c_j(v, s)$ in K.

CASE 4. $x_1 = z$ and $x_m \neq z'$. Then $x \overset{*}{-} x_1$ is represented by the edge $x - c_i(u, s)$ and $x_m \overset{*}{-} z'$ is a path that does not go through $P_X \cup A_X$. Then by definition of $eSkl_{(H,\chi)}$, there exists $c_{\ell_m}(r_m, t_m)$ and $c_{j'}(r', t')$ such that $x_{\ell_m}(r_m, t_m) = x_m$, $z' = x_{j'}(r', t')$ and there exists a path in $eSkl_{(H,\chi)}$ that does not go through $P_X \cup A_X$. Then the path $x_m \overset{*}{-} z \overset{*}{-} y$ is represented in K by the path $c_{j_m}(w_m, s_m) - c_{\ell_m}(r_m, t_m) - c_{j'}(r', t') - c_j(v, s') - y$.

CASE 5. $x_1 \neq z$ and $x_m = z'$. Similarly to Case 2, $x_m \overset{*}{-} y$ is represented by the edge $c_j(v, s) - y$ in $eSkl_{(H,\chi)}(x, y, X)$ and $x \overset{*}{-} x_1$ is represented by $x - c_i(u, s) - c_{i'}(r, t) - c_{\ell_1}(r_1, t_1) - c_{j_1}(w_1, s_1)$ where $x_i(u, s) = x_{i'}(r, t)$ and $x_{\ell_1}(r_1, t_1) = x_{j_1}(w_1, s_1)$.

CASE 6. $x_1 \neq z$ and $x_m \neq z'$.   The portions $x \overset{*}{-} x_1$ and $x_m \overset{*}{-} y$ are represented in $eSkl_{(H,\chi)}(x, y, X)$ respectively by $x - c_i(u, s) - c_{i'}(r, t) - c_{\ell_1}(r_1, t_1) - c_{j_1}(w_1, s_1)$ and $c_{j_m}(w_m, s_m) - c_{\ell_m}(r_m, t_m) - c_{j'}(r', t') - c_j(v, s') - y$.

It remains to show that $x_i \overset{*}{-} x_{i+1}$ is represented in $eSkl_{(H,\chi)}(x, y, X)$ by a path. We have two cases:

CASE 7.   The path $x_i \overset{*}{-} x_{i+1}$ is included in some $G[\chi(r_i)] \backslash (X \cap \chi(r_i))$ where $x_i = x_{\ell_i}(r_i, t_i)$ and $x_{i+1} = x_{\ell_{i+1}}(r_i, t_{i+1})$. By definition, this path is represented in K by the path $c_{j_i}(w_i, s_i) - c_{\ell_i}(r_i, t_i) - c_{\ell_{i+1}}(r_i, t_{i+1}) - c_{j_{i+1}}(w_{i+1}, s_{i+1})$.

CASE 8.   The path $x_i \overset{*}{-} x_{i+1}$ is not included in any e-block-vertex. Therefore, as in Cases 2-4 it is represented in K by some path $c_{j_i}(w_i, s_i) - c_{\ell_i}(r_i, t_i) - c_{\ell_{i+1}}(r_{i+1}, t_{i+1}) - c_{j_{i+1}}(w_{i+1}, s_{i+1})$ where $x_{j_i}(w_i, s_i) = x_{\ell_i}(r_i, t_i)$ and $x_{\ell_{i+1}}(r_{i+1}, t_{i+1}) = x_{j_{i+1}}(w_{i+1}, s_{i+1})$.

Therefore, if x and y are connected in $G\backslash X$ they are connected in K.   ∎

The following is the main theorem of this section.

Theorem 10.2 *Let $\mathcal{H}, \mathcal{D}_1$ and $\mathcal{D}_2$ be classes of undirected graphs and let ℓ be a positive integer. Assume the followings:*

1. *There exists an f_1-labeling scheme for the connectivity query on $\mathcal{D}_1$ and for every $G \in \mathcal{D}_1$, the f_1-labeling on G is constructed in $g_1(n)$-time.*

2. *There exists an f_2-labeling scheme for the connectivity query on $\mathcal{D}_2$ and for every $G \in \mathcal{D}_2$, the f_2-labeling on G is constructed in $g_2(n)$-time.*

Then there exists an $O(\ell^4 \cdot (f_1 + f_2))$-labeling scheme $(\mathcal{A}, \mathcal{B})$ for the connectivity query on the class of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-width at most ℓ. Moreover, if for every undirected graph $G \in \mathcal{C}$, we can construct a $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-decomposition of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-width at most ℓ in $f(n)$-time, then $\mathcal{A}$ computes the labels in $\max\{f(n), g_1(n), O(n \cdot g_2(n))\}$,-time and $\mathcal{B}$ gives the answer in $O(\ell^6 \cdot m^2)$-time where m is the size of the data.

Proof. Let G be an undirected graph with n vertices and let (H, χ) be a $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-decomposition of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-width at most ℓ. By definition of $eSkl_{(H,\chi)}$, we can construct it in $O(n)$-time. By Remark 10.2, $eSkl_{(H,\chi)}$ has $O(\ell^3 \cdot n)$ nodes. By assumption, $eSkl_{(H,\chi)} \in \mathcal{D}_1$ and for every $u \in V_H$, $G[\chi(u)] \in \mathcal{D}_2$. We assume that each vertex x of G has a bit-representation $\ulcorner x \urcorner$. Since G has n vertices this bit-representation has size $O(\log(n))$.

By hypothesis we can construct an f_1-labeling J_1 for the connectivity query on $eSkl_{(H,\chi)}$ in $g_1(n)$-time and for each $u \in V_H$ we can construct an f_2-labeling J_u for the connectivity query on $G[\chi(u)]$ in $g_2(n)$-time. For each $u \in V_H$ we let (the e-attachment-vertices of u are denoted by $c_1(u), \ldots, c_d(u)$, $d \leq \ell^2$ and we denote by $x(c_i(u))$ the vertex of G the node $c_i(u)$ represents):

$$C_u = \Big((J_u(x(c_1(u))), J_1(c_1(u)), \ulcorner x(c_1(u)) \urcorner), \ldots, (J_u(x(c_d(u))), J_1(c_d(u)), \ulcorner x(c_1(u)) \urcorner), J_1(u) \Big).$$

If w is an e-attachment-vertex of u_1 and represents a vertex x of G that is contained in $u_1, \ldots, u_s$, $s \leq \ell$, we let

$$C(w) = \Big(C_{u_1}, \ldots, C_{u_s} \Big).$$

For each $x \in V_G$ such that x is contained in $u_1, \ldots, u_s$, we let

$$L_u(x) = \Big(J_u(x), C(c_1(u)), \ldots, C(c_d(u)) \Big) \qquad \text{for } u \in \{u_1, \ldots, u_s\}.$$
$$J(x) = \Big(L_{u_1}(x), \ldots, L_{u_s}(x) \Big).$$

It is clear that $|J(x)| \leq O(\ell^4 \cdot (f_1(n) + f_2(n)))$ for every $x \in V_G$ and is computed in $\max\{f(n), g_1(n), O(n \cdot g_2(n))\}$-time. We now explain how to decide the connectivity of $x \in V_G$ and $y \in V_G$ in $G \backslash X$, $X \subseteq V_G - \{x, y\}$. By Lemma 10.7, we need only explain how to construct the problematic e-skeleton graph $eSkl_{(H,\chi)}(x, y, X)$ from $J(x), J(y)$ and $J(X)$.

From $J(z)$, $z \in V_G$, we can clearly determine the set of e-block-vertices that contain it (we can therefore determine for each u, the set $X \cap \chi(u)$). For each u that contains z, we can determine by using $L_u(z)$ the label of $J_u(z)$ and the labels of the set of e-attachment-vertices of u. For each e-attachment-vertex of u, by using its label we can determine the set of e-block-vertices with their e-attachment-vertices that contain it. We can therefore determine the labels of the nodes in $eSkl_{(H,\chi)}(x, y, X)$. Since we have all the needed labels we can determine the set of edges of $eSkl_{(H,\chi)}(x, y, X)$ (by comparing $\ulcorner x(c_i(u)) \urcorner$ and $\ulcorner x(c_j(v)) \urcorner$ we can determine if they represent the same vertex of G).   ∎

As a corollary we get the following.

Corollary 10.3 *Let $\mathcal{H}$ be a class of undirected graphs and let k and ℓ be positive integers. Let $\mathcal{D}_1, \mathcal{D}_2 \in \{CWD(\leq k), \mathcal{P}\}$ be two classes of undirected graphs. There exists a log-labeling scheme $(\mathcal{A}, \mathcal{B})$ for the connectivity query on the class of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-width at most ℓ. Moreover, if for every undirected graph $G \in \mathcal{C}$, we can construct a $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-decomposition of $(\mathcal{H}, \mathcal{D}_1, \mathcal{D}_2)$-e-width at most ℓ in $f(n)$-time, then $\mathcal{A}$ computes the labels in $\max\{f(n), O(n^2 \cdot \log(n))\}$-time.*

We now prove that $K_{3,3}$-minor-free graphs of bounded degree admits a log-labeling scheme for the connectivity query. We recall the following.

Lemma 10.8 *Let G be a $K_{3,3}$-minor-free graph with n vertices. For every edge e of G, there is a matching M in G with the following properties:*

(i) $|M| \leq \frac{1}{3} \cdot (n - 2)$.

(ii) *Each edge in M is disjoint from e.*

(iii) *Contracting M gives a planar graph.*

Moreover, we can determine the set M in $O(n)$-time.

Proof. In [WT07] it is proved that such a matching can be found in $O(n)$-time if we have as input the decomposition of G into clique-sum. In [Asa85] it is proved that this clique-sum can be found in linear-time for $K_{3,3}$-minor free graphs.   ∎

We denote by $\mathcal{G}_k$ the class of undirected graphs that have at most k vertices.

Corollary 10.4 *Every $K_{3,3}$-minor-free graph G with maximum degree $\Delta(G)$ admits a $(\mathcal{P}, \mathcal{P}, \mathcal{G}_4)$-e-decomposition of $(\mathcal{P}, \mathcal{P}, \mathcal{G}_4)$-width at most $2 \cdot \Delta(G) - 1$. Moreover, this $(\mathcal{P}, \mathcal{P}, \mathcal{G}_4)$-e-decomposition can be computed in linear-time.*

Proof. Let G be a $K_{3,3}$-minor-free graph with n vertices and let Δ be its maximum degree. By Lemma 10.8, there exists a matching M of edges of G such that their contraction gives a $\mathcal{P}$-v-decomposition (H, χ) of G such that for every $u \in V_H$, the sub-graph $G[\chi(u)]$ has at most 2 vertices. Then every node u of H has degree at most $2(\Delta - 1)$. We transform (H, χ) into a $\mathcal{P}$-e-decomposition (H', χ') of $\mathcal{P}$-spread at most $2 \cdot \Delta - 1$ and such that for each node u of H', the sub-graph $G[\chi'(u)]$ has at most 4 vertices. We transform H as follows: each edge uv of H is replaced by $u - w_{uv} - v$ where w_{uv} is a new node. For each node w of H', we let

$$\chi'(w) = \begin{cases} \chi(u) & \text{if } w \in V_H, \\ \chi(u) \cup \chi(v) & \text{if } w = w_{uv}. \end{cases}$$

It is clear that (H', χ') verifies the desired properties and for every $u \in V_H$ the sub-graph $G[\chi'(u)] \in \mathcal{G}_4$. By Remark 10.3, $eSkl_{(H', \chi')}$ is planar. It is clear that (H', χ') is constructed in linear-time. ∎

It follows the following.

Proposition 10.3 *Let d be a positive integer. Every class of $K_{3,3}$-minor-free graphs of maximum degree at most d admits a $(d^4 \cdot \log)$-labeling scheme $(\mathcal{A}, \mathcal{B})$ for the connectivity query. Moreover, if n is the number of vertices of the input graph, $\mathcal{A}$ computes the labels in $O(n \cdot \log(n))$-time.*

10.5   Conclusion

We have given two kinds of decompositions, $\mathcal{H}$-v-decomposition associated with $\mathcal{H}$-v-width and $\mathcal{H}$-e-decomposition associated with $\mathcal{H}$-e-width. We use the two notions to extend the labeling schemes for connectivity queries of planar graphs and of classes of graphs of bounded clique to some classes of graphs that have bounded $\mathcal{H}$-v-width and to some classes of graphs that have bounded $\mathcal{H}$-e-width. We have for instance identified the classes of $K_{3,3}$-minor free graphs of bounded degree as classes of graphs of bounded $\mathcal{P}$-e-width. There are several remaining questions and we can cite among others:

1. Identify more classes of graphs that have bounded $\mathcal{H}$-v-width and $\mathcal{H}$-e-width.

2. If (H, χ) is an $\mathcal{H}$-v-decomposition (resp. $\mathcal{H}$-e-decomposition) of an undirected graph G, what properties of G, e.g., labeling schemes, polynomial-time algorithms, etc., can we derive from the properties of $Skl_{(H,\chi)}$ (resp. $eSkl_{(H,\chi)}$) and of the sub-graphs $G[\chi(u)]$?

3. Given a class $\mathcal{H}$ of undirected graphs and a positive integer ℓ, how to recognize undirected graphs of $\mathcal{H}$-v-width (resp. $\mathcal{H}$-e-width) at most ℓ?

4. Find (nicely) locally cwd-decomposable graph classes that have $(\mathcal{H}, \mathcal{P}, CWD(\leq k))$-e-decompositions of bounded $\mathcal{H}$-e-width.

Conclusion

Motivated mainly by algorithmic applications, but also by structural graph questions, we have developed two themes:

1. better understanding of rank-width,

2. algorithmic meta-theorems beyond bounded clique-width and equivalent parameters.

Regarding the first theme we have given an algebraic characterization of the notion of rank-width of undirected graphs based on linear transformations of matrices. These operations allow to check MS-definable properties on graph classes of small rank-width more directly than by using clique-width operations. We propose two possible definitions for the notion of rank-width of directed graphs: one based on a coding of directed graphs by undirected ones, called bi-rank-width, and one based on a coding of directed graphs by matrices over the field $GF(4)$, called $GF(4)$-rank-width. We also derive from the works by Bouchet [Bou87] and Fon-Der-Flass [FDF96] two notions of vertex-minor, one related to bi-rank-width and the other to $GF(4)$-rank-width and, we show that directed graphs of $GF(4)$-rank-width are characterized by a finite list of excluded directed graphs as vertex-minors. This result generalizes the one by Oum [Oum05b] on undirected graphs. We also derive from the works by Hliněný and Oum a cubic-time recognition algorithm for directed graphs of $GF(4)$-rank-width (resp. bi-rank-width) at most k, which yields a cubic-time approximation algorithm for directed graphs of clique-width at most k and produces the relevant decompositions, that are necessary for algorithmic applications. As open questions, relevant to graph structure, we can cite:

(Q1) How to verify if two directed graphs are locally equivalent?

(Q2) How to decide if a directed graph is a vertex-minor of another directed graph?

(Q3) Are directed graph classes of bounded $GF(4)$-rank-width well-quasi-ordered? More generally, are directed graphs well-quasi-ordered by vertex-minor relation?

For tackling (Q2), we may hope to give a C_4MS-characterization of the vertex-minor relation. In this way we decide (Q2) for directed graph classes of small $GF(4)$-rank-width (Courcelle and Oum [CO07] gave a C_2MS-transduction of vertex-minor relation of undirected graphs and derived a cubic-time algorithm for (Q2) in the case of undirected graphs).

Many graph operations handling the same graph classes of bounded clique-width, but yielding different width parameters, have been presented. In general, the more vertex color manipulations the graph operations allow, the smallest is the corresponding width. Experimental studies for model checking of MS-definable properties on graph classes of bounded

clique-width should be conducted to obtain indications of the most efficient representations of graphs by algebraic terms. One can use for instance the MONA System [HJJ+95] as done in [Sog08].

In the second part of this thesis, we investigate labeling schemes on graph classes of bounded local clique-width. We prove that every FO-definable property admits a log-labeling scheme on certain graph classes of bounded local clique-width including bounded degree graph classes, minor-closed graph classes, unit-interval graphs. We also prove that the connectivity query also admits a log-labeling scheme on planar graphs and on graphs that are obtained by gluing graphs of small clique-width with limited overlaps. We can cite among the numerous future research topics:

(Q4) Having a better understanding of the structure of graph classes of bounded local clique-width and, a generalization of Eppstein's result concerning minor-closed graph classes of bounded local tree-width.

(Q5) Does there exist a log-labeling scheme for connectivity query on graph classes of bounded genus?

(Q6) Finding a labeling scheme on planar graphs for distances or approximate distances on sub-graphs defined by excluded vertices/edges. For exact distances, such a labeling scheme should use labels of size at least $\Omega(n^{1/3})$ for n-vertex planar graphs [GKK+01].

(Q6) Can one construct short labeling schemes where we can add vertices and edges?

All the short labeling schemes we have presented in Chapters 8 and 10 are based on combinations of several short labeling schemes into a unique one. More generally, we think interesting to study methods that combine several existing short labeling schemes for two graph classes into a unique short one, for a class of graphs obtained by combinations of graphs of the given classes.

Bibliography

[ACP87]   S. Arnborg, D.G. Corneil, and A. Proskurowski. Complexity of Finding Embeddings in a *k*-Tree. *SIAM Journal Algorithm Discrete Method*, 8:277–284, 1987.

[AGM⁺08]   I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup. Compact Name-Independent Routing with Minimum Stretch. *ACM Transactions on Algorithms*, 4(3):37, 2008.

[AGMW07]   I. Abraham, C. Gavoille, D. Malkhi, and U. Wieder. Strong-Diameter Decompositions of Minor Free Graphs. In P.B. Gibbons and C. Scheideler, editors, *Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 16–24. ACM, 2007.

[ALS91]   S. Arnborg, J. Lagergren, and D. Seese. Easy Problems for Tree-Decomposable Graphs. *Journal of Algorithms*, 12(3):308–340, 1991.

[Arn85]   S. Arnborg. Efficient Algorithms for Combinatorial Problems on Graphs with Bounded Decomposability - A Survey. *BIT*, 25(1):2–23, 1985.

[Asa85]   T. Asano. An Approach to the Sub-Graph Homeomorphism Problem. *Theoretical Computer Science*, 38(2-3):249–267, 1985.

[Bag09]   G. Bagan. *Algorithmes et Complexité des Problèmes d'Énumération pour l'Évaluation de Requêtes Logiques*. PhD thesis, Université de Caen, 2009.

[Bak94]   B.S. Baker. Approximation Algorithms for NP-Complete Problems on Planar Graphs. *Journal of the ACM*, 41(1):153–180, 1994.

[BC06]   B. Blumensath and B. Courcelle. Recognizability, Hypergraph Operations and Logical Types. *Information and Computation*, 204(6):853–919, 2006.

[BDG07]   G. Bagan, A. Durand, and E. Grandjean. First Order Queries on Structures of Bounded Degree are Computable with Constant Delay. *ACM Transactions on Computational Logic*, 8(4):Article number 21, 2007.

[BE97]   H. L. Bodlaender and J. Engelfriet. Domino Tree-Width. *Journal of Algorithms*, 24:94–123, 1997.

[BH98]   H. L. Bodlaender and T. Hagerup. Parallel Algorithms with Optimal Speedup for Bounded Tree-Width. *SIAM Journal on Computing*, 27:1725–1746, 1998.

[Bod88]   H. L. Bodlaender. NC-Algorithms for Graphs with Small Tree-Width. In J. van Leeuwen, editor, *Graph-Theoretic Concepts in Computer Science (WG)*, volume 344 of *LNCS*, pages 1–10. Springer, 1988.

[Bod96]   H. L. Bodlaender. A Linear Time Algorithm for Finding Tree-Decompositions of Small Tree-Width. *SIAM Journal on Computing*, 25:1305–1317, 1996.

[Bod98]   H. L. Bodlaender. A Partial k-Arboretum of Graphs with Bounded Tree-Width. *Theoretical Computer Science*, 209:1–45, 1998.

[Bod05]   H. L. Bodlaender. Discovering Tree-Width. In P. Vojtás, M. Bieliková, B. Charron-Bost, and O. Sýkora, editors, *Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 3381 of *LNCS*, pages 1–16. Springer, 2005.

[Bod06]   H. L. Bodlaender. Tree-Width: Characterizations, Applications and Computations. In F. V. Fomin, editor, *Graph-Theoretic Concepts in Computer Science (WG)*, volume 4271 of *LNCS*, pages 1–14. Springer, 2006.

[Bou87]   A. Bouchet. Digraph Decompositions and Eulerian Systems. *SIAM Journal Algorithm Discrete Method*, 8(3):323–337, 1987.

[Bou88]   A. Bouchet. Graphic Presentations of Isotropic Systems. *Journal of Combinatorial Theory, Series B*, 45(1):58–76, 1988.

[BT96]   G. Di Battista and R. Tamassia. On-Line Planarity Testing. *SIAM Journal on Computing*, 25(5):956–997, 1996.

[CDG⁺07]   H. Comon, M. Dauchet, R; Gilleron, F; Jacquemard, D. Lugiez, C. Löding, S. Tison, and M. Tommasi. Tree Automata Techniques and Applications, 2007.

[CER93]   B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-Rewriting Hypergraph Grammars. *Journal of Computer and System Sciences*, 46(2):218–270, 1993.

[CGK08]   B. Courcelle, C. Gavoille, and M. M. Kanté. Efficient First Order Model Checking Using Short Labels. In F.P. Preparata, X. Wu, and J. Yin, editors, *Frontiers of Algorithmics Workshop (FAW)*, volume 5059 of *LNCS*, pages 159–170. Springer, 2008.

[CGKT08]   B. Courcelle, C. Gavoille, M. M. Kanté, and A. Twigg. Connectivity Check in 3-Connected Planar Graphs with Obstacles. *Electronic Notes in Discrete Mathematics*, 31:151–155, 2008.

[CHL⁺00]   D. G. Corneil, M. Habib, J.M. Lanlignel, B. Reed, and U. Rotics. Polynomial Time Recognition of Clique-Width ≤ 3 Graphs. In G. H. Gonnet, D. Panario, and A. Viola, editors, *Latin American Symposium on Theoretical Informatics (LATIN)*, volume 1776 of *LNCS*, pages 126–134. Springer, 2000.

[CK07]   B. Courcelle and M. M. Kanté. Graph Operations Characterizing Rank-Width and Balanced Graph Expressions. In A. Brandstädt, D. Kratsch, and H. Müller, editors, *Graph Theoretic Concepts in Computer Science (WG)*, volume 4769 of *LNCS*, pages 66–75. Springer, 2007.

[CK09]   B. Courcelle and M. M. Kanté. Graph Operations Characterizing Rank-Width. *Discrete Applied Mathematics*, 157(4):627–640, 2009.

[CKS08]   F. Carrere, M.M. Kanté, and D. Soguet. Linear Algorithms on Clique-Width Bounded Graphs. Manuscript, 2008.

[CLR02]   T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press and McGraw Hill, 2nd edition, 2002.

[CMR00]   B. Courcelle, J. A. Makowsky, and U. Rotics. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory of Computing Systems*, 33(2):125–150, 2000.

[CMR01]   B. Courcelle, J. A. Makowsky, and U. Rotics. On the Fixed Parameter Complexity of Graph Enumeration Problems Definable in Monadic Second Order Logic. *Discrete Applied Mathematics*, 108(1-2):23–52, 2001.

[CO00]   B. Courcelle and S. Olariu. Upper Bounds to the Clique-Width of Graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.

[CO07]   B. Courcelle and S. Oum. Vertex-Minors, Monadic Second Order Logic, and a Conjecture by Seese. *Journal of Combinatorial Theory, Series B*, 97(1):91–126, 2007.

[Cou90]   B. Courcelle. The Monadic Second Order Logic of Graphs I: Recognizable Sets of Finite Graphs. *Information and Computation*, 85(1):12–75, 1990.

[Cou92]   B. Courcelle. The Monadic Second Order Logic of Graphs VII: Graphs as Relational Structures. *Theoretical Computer Science*, 101(1):3–33, 1992.

[Cou93]   B. Courcelle. Graph Grammars, Monadic Second Order Logic and the Theory of Graph Minors. In N. Robertson and P. Seymour, editors, *Graph Structure Theory*, volume 147 of *Contemporary Mathematics*, pages 565–590. AMS, 1993.

[Cou96]   B. Courcelle. Basic Notions of Universal Algebra for Language Theory and Graph Grammars. *Theoretical Computer Science*, 163(1-2):1–54, 1996.

[Cou97]   B. Courcelle. The Expression of Graph Properties and Graph Transformations in Monadic Second Order Logic. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformations*, pages 313–400. World Scientific, 1997.

[Cou00]   B. Courcelle. The Monadic Second Order Logic of Graphs XII: Planar Graphs and Planar Maps. *Theoretical Computer Science*, 237(1-2):1–32, 2000.

[Cou06a]   B. Courcelle. The Monadic Second Order Logic of Graphs XV: On a Conjecture by D. Seese. *Journal of Applied Logic*, 4(6):79–114, 2006.

[Cou06b]   B. Courcelle. The Monadic Second Order Logic of Graphs XVI: Canonical Graph Decompositions. *Logical Methods in Computer Science*, 2(2):Article number 2, 2006.

[CP06]   B. Crespelle and C. Paul. Fully Dynamic Recognition Algorithm and Certificate
             for Directed Cographs. *Discrete Applied Mathematics*, 154(12):1722–1741, 2006.

[CR05]   D. G. Corneil and U. Rotics. On the Relationship between Clique-Width and
             Tree-Width. *SIAM Journal on Computing*, 34:825–847, 2005.

[Cre07]   C. Crespelle. *Représentation Dynamique de Graphes*. PhD thesis, Université
             Montpellier II, 2007.

[CRST03]   M. Chudnovsky, N. Robertson, P. D. Seymour, and R. Thomas. Progress on
             Perfect Graphs. *Mathematical Programmaing, Series B*, 97:405–422, 2003.

[CS08a]   M. Chudnovsky and P. D. Seymour. Claw-Free Graphs IV: Decomposition The-
             orem. *Journal of Combinatorial Theory, Series B*, 98(5):839–938, 2008.

[CS08b]   M. Chudnovsky and P. D. Seymour. Claw-Free Graphs V: Global Structure.
             Journal of Combinatorial Theory, Series B, 98(6):1373–1410, 2008.

[CT07]   B. Courcelle and A. Twigg. Compact Forbidden-Set Routing. In W. Thomas and
             P. Weil, editors, *Symposium on Theoretical Aspects of Computer Science*, volume
             4393 of *LNCS*, pages 37–48. Springer, 2007.

[Cun82]   W. H. Cunningham. Decomposition of Directed Graphs. *SIAM Journal on Alge-
             braic and Discrete Methods*, 3:214–228, 1982.

[CV03]   B. Courcelle and R. Vanicat. Query Efficient Implementation of Graphs of
             Bounded Clique-Width. *Discrete Applied Mathematics*, 131:129–150, 2003.

[CW05]   B. Courcelle and P. Weil. The Recognizability of Sets of Graphs is a Robust
             Property. *Theoretical Computer Science*, 342(2-3):173–228, 2005.

[dBKOS91]   M. de Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational
             Geometry: Algorithms and Applications*. Springer, 1991.

[DDO+04]   M. Devos, G. Ding, B. Oporowski, D. Sanders, B. Reed, P. Seymour, and D. Ver-
             tigan. Excluding any Graph as a Minor Allows a Low Tree-Width 2-Coloring.
             Journal of Combinatorial Theory, Series B, 91(1):25–41, 2004.

[DF99]   R. Downey and M. Fellows. *Parameterized Complexity*. Springer, 1999.

[dFdMP95]   H. de Fraysseix, P. Ossona de Mendez, and J. Pach. A Left-First Search Algorithm
             for Planar Graphs. *Discrete and Computational Geometry*, 13:459–468, 1995.

[dFdMR95]   H. de Fraysseix, P. Ossona de Mendez, and P. Rosenstiehl. Bipolar Orientations
             Revisited. *Discrete Applied Mathematics*, 56(2-3):157–179, 1995.

[DG07]   Y. Dourisboure and C. Gavoille. Tree-Decompositions with Bags of Small Diam-
             eter. *Discrete Mathematics*, 307(16):2008–2029, 2007.

[DGK07]   A. Dawar, M. Grohe, and S. Kreutzer. Locally Excluding a Minor. In *Symposium
             On Logic In Computer Science (LICS)*, pages 270–279. IEEE Computer Science,
             2007.

[DGKS07]   A. Dawar, M. Grohe, S. Kreutzer, and N. Schweikardt. Model Theory Makes Formulas Large. In L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, editors, *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4596 of *LNCS*, pages 913–924. Springer, 2007.

[DH04a]   E.D. Demaine and M. Hajiaghayi. Diameter and Tree-Width in Minor-Closed Graph Families, Revisited. *Algorithmica*, 40(3):211–215, 2004.

[DH04b]   E.D. Demaine and M. Hajiaghayi. Equivalence of Local Tree-Width and Linear Local Tree-Width and its Algorithmic Applications. In *Symposium On Discrete Algorithms (SODA)*, pages 840–849. SIAM, 2004.

[Die05]   R. Diestel. *Graph Theory.* Springer, 3rd edition, 2005.

[Dir61]   G. Dirac. On Rigid Circuit Graphs. *Abhandlungen Mathematischer Seminare der Universität Hamburg*, 21:71–76, 1961.

[DK05]   R. Diestel and D. Kühn. Graph Minor Hierarchies. *Discrete Applied Mathematics*, 145(2):167–182, 2005.

[dM03]   F. de Montgolfier. *Décomposition Modulaire des Graphes: Théorie, Extensions et Algorithmes.* PhD thesis, Université Montpellier II, 2003.

[DOSV00]   G. Ding, B. Oporowski, D. Sanders, and D. Vertigan. Surfaces, Tree-Width, Clique-Minors and Partitions. *Journal of Combinatorial Theory, Series B*, 79(2):221–246, 2000.

[DTCR08]   C. Demetrescu, M. Thorup, R. A. Chowdhury, and V. Ramachandran. Oracles for Distances Avoiding a Failed Node or Link. *SIAM Journal on Computing*, 37(5):1299–1318, 2008.

[Epp00]   D. Eppstein. Diameter and Tree-Width in Minor-Closed Graph Families. *Algorithmica*, 27(3):275–291, 2000.

[ET76]   S. Even and R. E. Tarjan. Computing an (s,t)-Numbering. *Theoretical Computer Science*, 2:339–344, 1976.

[Eul59]   L. Euler. Solutio Problematis ad Geometriam situs Pertinentis. *Mémoires de l'Académie des sciences de Berlin*, pages 128–140, 1759.

[EvO97]   J. Engelfriet and V. van Oostrom. Logical Description of Context-Free Graph Languages. *Journal of Computer and System Sciences*, 55(3):489–503, 1997.

[FDF96]   D.G. Fon-Der-Flaass. Local Complementations of Simple and Directed Graphs. In A.D. Korshunov, editor, *Discrete Analysis and Operations Reserach*, pages 15–34. Kluwer Academic, 1996.

[FG01a]   J. Flum and M. Grohe. Fixed-Parameter Tractability, Definability and Model-Checking. *SIAM Journal on Computing*, 31(1):113–145, 2001.

[FG01b]   M. Frick and M. Grohe. Deciding First Order Properties of Locally Tree-Decomposable Structures. *Journal of the ACM*, 48(6):1184–1206, 2001.

[FG06]   J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[FMR08]   E. Fisher, J. A. Makowski, and E. V. Ravve. Counting Truth Assignments of For-
            mulas of Bounded Tree-Width or Clique-Width. *Discrete Applied Mathematics*,
            156(4):511–529, 2008.

[Fri04]   M. Frick. Generalized Model-Checking over Locally Tree-Decomposable Classes.
            Theory of Computing Systems, 37(1):157–191, 2004.

[FRRS06]   M. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider. Clique-Width Minimiza-
            tion is NP-Hard. In J. M. Kleinberg, editor, *Symposium on Theory Of Computing
            (STOC)*, pages 354–362. ACM, 2006.

[Gai82]   H. Gaifman. On Local and Non-Local Properties. In J. Stern, editor, *Proceedings
            of the Herbrand Symposium, Logic Colloquium'81*, pages 105–135. North-Holland
            Publishing Company, 1982.

[Gal67]   T. Gallai. Transitiv Orientierbare Graphen. *Acta Mathematica Academiae Sci-
            entiarium Hungaricae*, 18:26–66, 1967.

[GGRW03]   J. F. Geelen, A. M. H. Gerards, N. Robertson, and G. Whittle. On the Excluded
            Minors for the Matroids of Branch-Width k. *Journal of Combinatorial Theory,
            Series B*, 88(2):261–265, 2003.

[GJ79]   M.R. Garey and D.S. Johnson. *Computers and Intractability*. 1979.

[GKK⁺01]   C. Gavoille, M. Katz, N.A. Katz, D. Peleg, and C. Paul. Approximate Distance
            Labeling Schemes. In F. Meyer auf der Heide, editor, *European Symposium on
            Algorithms (ESA)*, volume 2161 of *LNCS*, pages 476–487. Springer, 2001.

[GL07]   C. Gavoille and A. Labourel. Shorter Implicit Representation for Planar Graphs
            and Bounded Tree-Width Graphs. In L. Arge, M. Hoffmann, and E. Welzl,
            editors, *European Symposium on Algorithms (ESA)*, volume 4698 of *LNCS*, pages
            582–593. Springer, 2007.

[GP03a]   C. Gavoille and C. Paul. Distance Labeling Scheme and Split Decomposition.
            Discrete Mathematics, 273(1-3):115–130, 2003.

[GP03b]   C. Gavoille and C. Paul. Optimal Distance Labeling for Interval and Circular-
            Arc Graphs. In G. Di Battista and U. Zwick, editors, *European Symposium on
            Algorithms (ESA)*, volume 2832 of *LNCS*, pages 254–265. Springer, 2003.

[GP03c]   C. Gavoille and D. Peleg. Compact and Localized Distributed Data Structures.
            Distributed Computing, 16(2-3):111–120, 2003.

[GP07]   E. Gioan and C. Paul. Dynamic Distance Hereditary Graphs Using Split Decom-
            position. In T. Tokuyama, editor, *International Symposium on Algorithms And
            Computations (ISAAC)*, volume 4835 of *LNCS*, pages 41–51. Springer, 2007.

[GPPR04]   C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance Labeling in Graphs.
            Journal of Algorithms, 53(1):85–112, 2004.

[Gro07]   M. Grohe. *Logic and Automata*, chapter Logic, Graphs and Algorithms. Amsterdam University Press, 2007.

[HJJ$^+$95] J. G. Henriksen, J. L. Jensen, M. E. Jørgensen, N. Klarlund, R. Paige, T. Rauhe, and A. Sandholm. MONA: Monadic Second Order Logic in Practice. In E. Brinksma, R. Cleaveland, K. G. Larsen, T. Margaria, and B. Steffen, editors, *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 1019 of *LNCS*, pages 89–110. Springer, 1995.

[HO07]   P. Hliněný and S. Oum. Finding Branch-Decompositions and Rank-Decompositions. In L. Arge, M. Hoffmann, and E. Welzl, editors, *European Symposium on Algorithms(ESA)*, volume 4698 of *LNCS*, pages 163–174. Springer, 2007.

[HOSG08]   P. Hliněný, S. Oum, D. Seese, and G. Gottlob. Width Parameters beyond Tree-Width and their Applications. *The Computer Journal*, 51:326–362, 2008.

[HT73]   J. E. Hopcroft and R. E. Tarjan. Dividing a Graph into Tri-Connected Components. *SIAM Journal on Computing*, 2(3):135–158, 1973.

[HT74]   J. E. Hopcroft and R. E. Tarjan. Efficient Planarity Testing. *Journal of the ACM*, 21(4):549–568, 1974.

[Joh98]   Ö. Johansson. Clique-Decomposition, NLC-Decomposition and Modular Decomposition : Relationships and Results for Random Graphs. *Congressus Numerantium*, 132:39–60, 1998.

[Joh03]   J.L. Johnson. *Polynomial-Time Recognition and Optimization Algorithms on Special Classes of Graphs*. PhD thesis, Vanderbilt University, 2003.

[Kan07]   M. M. Kanté. Vertex-minor Reductions can Simulate Edge Contraction. *Discrete Applied Mathematics*, 155(17):2328–2340, 2007.

[Kan08]   M. M. Kanté. The Rank-Width of Directed Graphs. Technical report, Université Bordeaux 1, LaBRI, 2008.

[KM07]   K. Kawarabashi and B. Mohar. Some Recent Progress and Applications in Graph Minor Theory. *Graphs and Combinatorics*, 23:1–46, 2007.

[Kor07a]   A. Korman. General Compact Labeling Schemes for Dynamic Trees. *Distributed Computing*, 20(3):179–193, 2007.

[Kor07b]   A. Korman. Labeling Schemes for Vertex Connectivity. In L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, editors, *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4596 of *LNCS*, pages 102–109. Springer, 2007.

[Kor08]   A. Korman. Improved Compact Routing Schemes for Dynamic Trees. In R. A. Bazzi and B. Patt-Shamir, editors, *Principles Of Distributed Computing (PODC)*, pages 185–194. ACM, 2008.

[KP07]   A. Korman and D. Peleg. Labeling Schemes for Weighted Dynamic Trees. *Information and Computation*, 205(12):1721–1740, 2007.

[KPR04]   A. Korman, D. Peleg, and Y. Rodeh. Labeling Schemes for Dynamic Tree Networks. *Theory of Computing Systems*, 37(1):49–75, 2004.

[Kre08]   S. Kreutzer. Algorithmic Meta-Theorems. Manuscript., 2008.

[Kru60]   J.B. Kruskal. Well-Quasi-Ordering, the Tree Theorem and Vazsonyi's Conjecture. *Transactions of the American Mathematical Society*, 95(2):210–225, 1960.

[Lab07]   A. Labourel. *Partition d'Arêtes et Représentation Implicite de Graphes*. PhD thesis, Université de Bordeaux, 2007.

[Lip91]   S. Lipschutz. *Schaum's Outline of Theory and Problems of Linear Algebra*. McGraw Hill, $2^n d$ edition, 1991.

[LN97]   R. Lidl and H. Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and its Applications, 2nd edition, 1997.

[Loz08]   V. Lozin. From Tree-Width to Clique-Width: Excluding a Unit-Interval Graph. In H. Nagamochi S. Hong and T. Fukunaga, editors, *International Symposium on Algorithms and Computation (ISAAC)*, volume 5369 of *LNCS*, pages 871–882. Springer, 2008.

[Mak04]   J. A. Makowsky. Algorithmic Uses of the Feferman-Vaught Theorem. *Annals of Pure and Applied Logic*, 126(1-3):159–213, 2004.

[MR06]   K. Meer and D. Rautenbach. On the OBDD Size for Graphs of Bounded Tree- and Clique-Width. In H.L. Bodlaender and M.A. Langstone, editors, *International Workshop on Parametrized and Exact Computation (IWPEC)*, volume 4169 of *LNCS*, pages 72–83. Springer, 2006.

[MT01]   B. Mohar and C. Thomassen. *Graphs on Surfaces*. The Johns Hopkins University Press, 2001.

[NdM06a]   J. Nešetřil and P. Ossona de Mendez. Linear-Time Low Tree-Width Partitions and Algorithmic Consequences. In J.M. Kleinberg, editor, *Symposium on Theory Of Computing (STOC)*, pages 391–400. ACM, 2006.

[NdM06b]   J. Nešetřil and P. Ossona de Mendez. Tree-Depth, Sub-Graph Coloring and Homomorphism Bounds. *European Journal of Combinatorics*, 27:1022–1041, 2006.

[NdM08a]   J. Nešetřil and P. Ossona de Mendez. Grad and Classes with Bounded Expansion I: Decompositions. *European Journal of Combinatorics*, 29:760–776, 2008.

[NdM08b]   J. Nešetřil and P. Ossona de Mendez. Grad and Classes with Bounded Expansion II: Algorithmic Aspects. *European Journal of Combinatorics*, 29:777–791, 2008.

[NdM08c]   J. Nešetřil and P. Ossona de Mendez. Grad and Classes with Bounded Expansion III: Restricted Graph Homomorphism Dualities. *European Journal of Combinatorics*, 29:1012–1024, 2008.

[NW05]   R. Nunkesser and P. Woelfel. Representation of Graphs by OBDDs. In X. Deng and D. Du, editors, *International Symposium on Algorithms And Computation (ISAAC)*, volume 3827 of *LNCS*, pages 1132–1142. Springer, 2005.

[OS06]   S. Oum and P.D. Seymour. Approximating Clique-Width and Branch-Width. *Journal of Combinatorial Theory, Series B*, 96:514–528, 2006.

[Oum05a]   S. Oum. Approximating Rank-Width and Clique-Width Quickly. In D. Kratsch, editor, *Graph-Theoretic Concepts in Computer Science (WG)*, volume 3787 of *LNCS*, pages 49–58. Springer, 2005.

[Oum05b]   S. Oum. Rank-Width and Vertex-Minors. *Journal of Combinatorial Theory, Series B*, 95:79–100, 2005.

[Oum05c]   S. Oum. Rank-Width and Well-Quasi-Ordering of Skew-Symmetric Matrices. *Electronic Notes in Discrete Mathematics*, 22:281–285, 2005.

[Oum08a]   S. Oum. Rank-Width and Well-Quasi-Ordering. *SIAM Journal on Discrete Mathematics*, 22(2):666–682, 2008.

[Oum08b]   S. Oum. Rank-Width is Less Than or Equal to Branch-Width. *Journal of Graph Theory*, 57(3):239–244, 2008.

[Oum09]   S. Oum. Excluding a Bipartite Circle Graph from Line Graphs. *Journal of Graph Theory*, 60(3):183–203, 2009.

[Pel93]   D. Peleg. Distance-Dependent Distributed Directories. *Information and Computation*, 103:270–298, 1993.

[PT07]   M. Pătraşcu and M. Thorup. Planning Fast Connectivity Updates. In *Symposium on Foundations Of Computer Science (FOCS)*, pages 263–271. IEEE Computer Society, 2007.

[Rao06]   M. Rao. *Décompositions de Graphes et Algorithmes Efficaces*. PhD thesis, Université Paul Verlaine, Metz, 2006.

[Rao08]   M. Rao. Solving some NP-Complete Problems Using Split Decomposition. *Discrete Applied Mathematics*, 156(14):2768–2780, 2008.

[RS83]   N. Robertson and P.D. Seymour. Graph Minors I : Excluding a Forest. *Journal of Combinatorial Theory B, Series B*, 35(1):39–61, 1983.

[RS86]   N. Robertson and P.D. Seymour. Graph Minors V : Excluding a Planar Graph. *Journal of Combinatorial Theory B, Series B*, 41(1):92–114, 1986.

[RS90]   N. Robertson and P.D. Seymour. Graph Minors IV : Tree-Width and Well-Quasi-Ordering. *Journal of Combinatorial Theory B, Series B*, 48(2):227–254, 1990.

[RS91]   N. Robertson and P. D. Seymour. Graph Minors X: Obstructions to Tree-Decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

[RS95]   N. Robertson and P. D. Seymour. Graph Minors XIII: The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

[RS03]   N. Robertson and P. D. Seymour. Graph Minors XVI: Excluding a Non-Planar Graph. *Journal of Combinatorial Theory, Series B*, 89(1):77–108, 2003.

[RS04]   N. Robertson and P. D. Seymour. Graph Minors XX: Wagner's Conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

[Sch90]   W. Schnyder. Embedding Planar Graphs on the Grid. In *Symposium On Discrete Algorithms (SODA)*, pages 138–148. SIAM, 1990.

[Sch97]   I. Schiering. A Hierarchical Approach to Monadic Second Order Logic over Graphs. In M. Nielsen and W. Thomas, editors, *Computer Science Logic (CSL)*, volume 1414 of *LNCS*, pages 424–440. Springer, 1997.

[Sch03]   A. Schrijver. *Cominatorial Optimization, Polyhedra and Efficiency*, volume B. Springer, 2003.

[See85]   D. Seese. Tree-Partite Graphs and the Complexity of Algorithms. In L. Budach, editor, *Fundamentals of Computation Theory (FCT)*, volume 199 of *LNCS*, pages 412–421. Springer, 1985.

[See96]   D. Seese. Linear-Time Computable Problems and First Order Descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

[Sno04]   J. Snoeyink. Point Location. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 34, pages 767–785. Chapman and Hall/CRC, 2nd edition, 2004.

[Sog08]   D. Soguet. *Génération Automatique d'Algorithmes Linéaires*. PhD thesis, Université Paris XI, Orsay, 2008.

[Spi03]   J. Spinrad. *Efficient Graph Representations*. AMS, 2003.

[ST86]   N. Sarnak and R.E. Tarjan. Planar Point Location Using Persistent Search Trees. *Communications of the ACM*, 29(7):669–679, 1986.

[ST07]   K. Suchan and I. Todinca. On Powers of Graphs of Bounded NLC-Width (Clique-Width). *Discrete Applied Mathematics*, 155(14):1885–1893, 2007.

[Tho07]   M. Thorup. Compact Oracles for Approximate Distances Around Obstacles in the Plane. In L. Arge, M. Hoffmann, and E. Welzl, editors, *European Symposium on Algorithms (ESA)*, volume 4698 of *LNCS*, pages 383–394. Springer, 2007.

[TNS82]   K. Takamizawa, T. Nishizeki, and N. Saito. Linear-Time Computability of Combinatorial Problems on Series-Parallel Graphs. *Journal of the ACM*, 29:623–641, 1982.

[TT86]   R. Tamassia and I. G. Tollis. A Unified Approach and Visibility Representation of Planar Graphs. *Discrete and Computational Geometry*, 1:321–341, 1986.

[Tut66]   W. T. Tutte. *Connectivity in Graphs*. University of Toronto Press, 1966.

[TZ05]   M. Thorup and U. Zwick. Approximate Distance Oracles. *Journal of the ACM*, 52(1):1–24, 2005.

[Wan94]   E. Wanke. *k*-NLC Graphs and Polynomial Algorithms. *Discrete Applied Mathe-matics*, 54:251–266, 1994.

[Wec92]   W. Wechler. *Universal Algebra for Computer Scientists*. Springer, 1992.

[Wir]   M. Wirsing. Algebraic Specifications. In *Handbook of Theoretical Computer Sci-ence, Volume B: Formal Models and Semantics*, pages 675–788.

[WT07]   D.R. Wood and J.A. Telle. Planar Decompositions and the Crossing Number of Graphs with an Excluded Minor. *New York Journal of Mathematics*, 13:117–146, 2007.