Graph Operations Characterizing Rank-Width

Bruno Courcelle [2,1] and Mamadou Moustapha Kanté [2,1]

Université Bordeaux 1, LaBRI, CNRS
351 Cours de la libération
33405 Talence Cedex, France.

Abstract

Graph complexity measures like *tree-width*, *clique-width* and *rank-width* are important because they yield *Fixed Parameter Tractable* algorithms. These algorithms are based on hierarchical decompositions of the considered graphs and on boundedness conditions on the graph operations that, more or less explicitly, recombine the components of decompositions into larger graphs. Rank-width is defined in a combinatorial way, based on a tree, and not in terms of graph operations. We define operations on graphs that characterize rank-width and help for the construction of Fixed Parameter Tractable algorithms, especially for problems specified in monadic second-order logic.

Key words: Clique-Width; Rank-Width; Hierarchical Decomposition; Graph Operation.

1 Introduction

Graph complexity measures like *tree-width* [29], *branch-width* [30], *clique-width* [10] and *rank-width* [27] are important parameters for the construction of polynomial algorithms. Many NP-complete properties, especially those expressible by formulas of *monadic second-order* logic (abbreviated MS logic in the sequel) have *Fixed Parameter Linear* algorithms if tree-width (equivalently branch-width) is taken as parameter and *Fixed Parameter Cubic* algorithms if clique-width (equivalently rank-width) is taken as parameter. These results are proved in the books by Downey and Fellows [15] and by Flum and Grohe

[1] Emails: `courcell@labri.fr`, `mamadou.kante@labri.fr`.

[2] Supported by the GRAAL project of "Agence Nationale pour la Recherche". B. Courcelle is member of "Institut Universitaire de France".

[18] for tree-width, by Courcelle et al. [9] with help of results by Hliněný, Oum and Seymour [20,28] for rank-width and clique-width.

These complexity measures are also interesting for the study of relations on graphs like *minor inclusion* and *vertex-minor inclusion* (see Robertson and Seymour [29], the book by Diestel [14], the article by Oum [27]).

All complexity measures defining graph "widths" are based on hierarchical decompositions. These decompositions arise in two different ways: either because the graphs are defined as the values of terms written with some kinds of "graph concatenations" or because edges and vertices are associated with the nodes of certain trees.

Clique-width and its close variants, NLC-width [31] and m-clique-width [11], are based on decompositions of the first category. Tree-width [29], branch-width [30], and rank-width [28] are decompositions of the second type. However, tree-width has an exact characterization in terms of graph operations [5, Theorem 1.1]. In this article we give one for rank-width. In all cases the width of a graph is defined as the minimal width of decomposition of a certain type of this graph, where the width of the decomposition measures how complex is the construction of the graph from the tree representing the decomposition.

Two widths, say wd and wd', are *equivalent* if the same sets of graphs have bounded width with respect to both of them. This is the case if there exist two strictly increasing functions f and $g : \mathbb{N} \to \mathbb{N}$ such that for every graph G of the considered type (simple or not, directed or not) we have $f(wd(G)) \leq wd'(G) \leq g(wd(G))$. While being equivalent, two widths may have advantages and drawbacks.

In particular, clique-width and rank-width are equivalent but clique-width has the advantage of having an algebraic definition in terms of very simple graph operations. Furthermore, this definition is the basis of the construction of algorithms for checking graph properties expressible in MS logic and for solving optimization problems expressed in MS logic [9,12,18] in linear-time in the size of the algebraic expressions defining the input graphs.

On the other hand, rank-width has a good behavior with respect to vertex-minor inclusion, so that the class of graphs of rank-width at most k is characterized by finitely many excluded vertex-minors [27]. Furthermore, the cubic-time algorithm that constructs for a given graph an algebraic expression of clique-width at most $2^{k+1}-1$ if the graph has clique-width at most k, is based on the decomposition underlying rank-width [20,28].

In this article we define algebraic operations on graphs that characterize rank-width as follows:

> a graph G 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 finite set of binary graph "concatenation" operations, C_k is a finite set of constants, both depending on k, and $T(R_k, C_k)$ is the set of finite well-formed terms constructed with R_k and C_k.

In a few words, the operations are based on coloring vertices by sets of colors $\subseteq [k] := \{1, 2, \ldots, k\}$, like in the variant of clique-width called *m-clique-width* (see [10,11]), but vertex colors are manipulated by linear transformations on the $GF(2)$ vector space $\{0,1\}^k$ rather than with set union over subsets of $\{1, \ldots, k\}$. Furthermore, edges are created between two disjoint graphs by means of bilinear forms, taking the vectors of colors as arguments. It is thus somewhat natural that they can generate (exactly) the set of graphs of rank-width at most k since the notion of rank-width is based on ranks of $GF(2)$ matrices.

That MS definable graph properties are Fixed Parameter Linear for tree-width and clique-width (assuming that graphs are given by the relevant decompositions or algebraic expressions) can be proved in a unified way because, up to some technical details, the graph operations underlying these decompositions or expressions, are expressible in terms of the following operations acting on logical structures:

- disjoint union,
- quantifier-free transformations [4,9].

The operation that replaces everywhere a vertex color a by the color b, and the one that adds edges between every vertex colored by a and every vertex colored by b are typical examples of quantifier-free transformations. General quantifier-free transformations modify logical structures by redefining certain relations by quantifier-free formulas (see [1,4,6,13] for graph algebras).

We present the graph operations that define rank-width as particular operations based on disjoint union and vertex-colorings with bounded number of distinct colors. We obtain thus a unifying framework for defining and comparing several related notions of width.

For comparing these families and the corresponding widths, we can say roughly that, the more vertex color manipulations we allow, the smallest is the corresponding width. In this respect, rank-width is "smaller" than clique-width but an even smaller width can be defined. However, all these width notions are equivalent to clique-width. But their recognition algorithms may be very different. That a width is smaller than another equivalent one, is unimportant with respect to the question: «which classes of graphs have bounded width ?», but seems useful for the construction of *Fixed Parameter Tractable* (abbreviated FPT) algorithms based on MS formulas. The reason is that, for representing graph operations based on k colors, one uses k unary relations $c_1, \ldots, c_k$ such that $c_i(x)$ means "vertex x has color i" (possibly among others). The constants in the linear algorithms based on the methodology explained in [9,15,18] depend on the sizes of the basic sets of relations used to describe the colored graphs. Hence, using less relations yields smaller (although large) constants.

However, the two possibilities for a usable implementation are the following ones (we denote by n the number of vertices of the graphs):

- either we define graphs by terms of size $p(k) \cdot n$ built with $p(k)$ basic operations (for graphs of clique-width at most k we have $p(k) = \theta(k^2)$),
- or we define them by terms of size $2n - 1$, and this is what we can do as a consequence of our main result, but built with much larger sets of operations (say $\theta(2^{k^2})$ for graphs of rank-width at most k).

Using a system like MONA [19] able to convert MS formulas into automata on terms may indicate whether one method is better than the other. Practical experience is yet insufficient to decide.

For comparing two widths, say wd and wd', a proof that for every graph G

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

consists in general in proving that every wd-decomposition of width k based on a tree T can be transformed into a wd'-decomposition of width at most $f(k)$, based on a tree T'. In most cases the trees T and T' are almost the same. Other results involving *deep reorganizations* of graph decompositions are established in [7].

The main results of this article are:

- a unified "Boolean" framework to formalize graph operations equivalent to those defining clique-width,
- an algebraic characterization of rank-width.

Summary. In Section 2 we recall the definitions of rank-width and clique-width. In Section 3 we define a general notion of graph operations based on vectorial colorings and we recall the relationships between these different widths. We prove in Section 4 that rank-width is characterized by certain graph operations based on linear transformations. We extend the results of Section 4 to edge-colored graphs in Section 5. Section 6 is a conclusion and states some open questions.

2 Notations and Definitions

We denote by $[k]$ the set $\{1, \ldots, k\}$ and by $[k]'$ the set $\{1', \ldots, k'\}$ to provide an isomorphic copy of $[k]$ to be used in some constructions. Graphs are finite, simple, loop-free and undirected unless otherwise specified. A graph G is defined as $\langle V_G, edg_G\rangle$ where $edg_G \subseteq V_G \times V_G$ is the symmetric adjacency relation. Without loss of generality we assume that V_G is always linearly ordered. This order will be used to represent edg_G by a square matrix over $GF(2)$. For a graph G and a set $U \in V_G$, we denote by $G[U]$ the sub-graph of G induced by U. We denote by 2^V the power-set of some set V.

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 $N_T^{(1)}$ the set of nodes of degree 1). We will denote T' by $Red(T)$.

Composition of multivalued functions. Let $\alpha : A \to 2^B$ and $\beta : B \to 2^C$ be two multivalued functions. We denote by $\beta \circ \alpha$ the mapping $A \to 2^C$ such that $\beta \circ \alpha(a) = \beta(\alpha(a)) = \bigcup\{\beta(b) \mid b \in \alpha(a)\}$. We also use $\circ$ for the normal composition of unary functions.

Let F be a set of unary and binary functions and C be 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.

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 $*$ and by deleting the unary symbols. Formally,

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

Contexts [12,11]. Let F and C be as above. 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. We denote by Id the particular context u. Let s be a context and t be a term or a context, we denote by $s[t/u]$ the term obtained by replacing u in s with t. We will use the notation $s \bullet t$ for $s[t/u]$ for s in $Cxt(F, C)$ and t in $T(F, C)$.

2.1 Clique-Width

We recall the definition of *clique-width* [3,6,10]. Let k be a positive integer. A *k-graph* is a graph whose vertices are colored with colors from $[k]$. Formally we define a k-graph as $G = \langle V_G, edg_G, \delta_G \rangle$ where $\delta_G(x) \in [k]$ for each $x \in V_G$ (uncolored graphs are considered as graphs whose vertices have all the same color). We recall the following operations:

(1) For k-graphs $G = \langle V_G, edg_G, \delta_G \rangle$ and $H = \langle V_H, edg_H, \delta_H \rangle$ such that $V_G \cap V_H = \emptyset$ (if not, we take a disjoint copy of H), we denote by $G \oplus H$ the k-graph $K = \langle V_G \cup V_H, edg_G \cup edg_H, \delta_K \rangle$ and call it the disjoint union of G and H where:

$$\delta_K(x) = \begin{cases} \delta_G(x) & \text{if } x \in V_G \\ \delta_H(x) & \text{if } x \in V_H. \end{cases}$$

(2) For $i, j \in [k]$ with $i \neq j$, for a k-graph $G = \langle V_G, edg_G, \delta_G \rangle$, we denote by $\eta_{i,j}(G)$ the k-graph $K = \langle V_G, edg_K, \delta_G \rangle$ where

$$edg_K = edg_G \cup \{xy \mid x, y \in V_G,\ x \neq y \text{ and } \delta_G(x) = i,\ \delta_G(y) = j\}.$$

(3) For $i, j \in [k]$ with $i \neq j$, for a k-graph $G = \langle V_G, edg_G, \delta_G \rangle$, we denote by $\rho_{i \to j}(G)$, the k-graph $K = \langle V_G, edg_G, \delta_K \rangle$ where

$$\delta_K(x) = \begin{cases} j & \text{if } \delta_G(x) = i \\ \delta_G(x) & \text{otherwise.} \end{cases}$$

(4) For $i \in [k]$, we let $\mathbf{i}$ denote a k-graph with a single vertex colored by i.

We let F_k^c be the set $\{\oplus, \eta_{i,j}, \rho_{i \to j} \mid i, j \in [k],\ i \neq j\}$ and C_k^c be the set $\{\mathbf{i} \mid i \in [k]\}$. Every term t in $T(F_k^c, C_k^c)$ denotes a k-graph $val(t)$. The *clique-width* of a graph G, denoted by $cwd(G)$, is the minimum k such that there exists a term t in $T(F_k^c, C_k^c)$ that defines G, i.e., such that G is isomorphic to $val(t)$.

The problem of checking if $cwd(G) \leq k$ for given (G, k) is NP-complete [17], polynomial if $k \leq 3$ [2]. However, thanks to [20] and [28], clique-width can be approximated in cubic-time. This is enough for constructing FPT algorithms.

Lozin and Rautenbach [24] consider the problem of finding a term t in $T(F_k^c, C_k^c)$ for k minimal and such that $red(t) = r \in T(\{*\}, \{\#\})$ where the term r is given with a bijection of the set of vertices onto the set of occurrences of $\#$. The term t must respect this bijection. They give a polynomial-time algorithm that constructs a term t with k at most twice the minimal value.

A variant of clique-width, called *m-clique-width*, has been defined and used in [10,11]. It is based on vertex-colorings where a vertex may have several colors or no color at all. Since a set of at most k colors can be considered as a single color from a set of cardinality 2^k, it is not a surprise that for every graph G, $mcwd(G) \leq cwd(G) \leq 2^{mcwd(G)+1} - 1$, where $mcwd(G)$ denotes the m-clique-width of G. For our algebraic characterization of rank-width, we will also use such colorings with several colors and graph operations that use and transform such colorings in a more powerful way than those defining m-clique-width. We will put our definitions in a convenient formal framework defined in Section 3.

2.2 Rank-Width

We now recall the definition of *rank-width* [27]. For an (R, C)-matrix $M = (m_{ij} \mid i \in R, j \in C)$ over some field, if $X \subseteq R$, $Y \subseteq C$, we let $M[X, Y]$ denote the sub-matrix $(m_{ij} \mid i \in X, j \in Y)$. The *order* of an (R, C)-matrix is $|R| \times |C|$. We will sometimes denote an (R, C)-matrix by its order when the context is clear.

For a graph G, we let A_G be its adjacency (V_G, V_G)-matrix over $GF(2)$. We assume that the vertex set of each graph G is linearly ordered, for instance by a numbering of vertices. From such a linear order, one defines the $(n \times n)$-adjacency matrix A_G of G, where $n = |V_G|$.

Cut-rank functions. Let $G = \langle V_G, edg_G \rangle$ be a graph. We define the *cut-rank* function ρ_G of G by letting $\rho_G(X) = rk(A_G[X, V_G \backslash X])$ for $X \subseteq V_G$, where rk is the matrix rank function. We let $\rho_G(\emptyset) = \rho_G(V_G) = 0$.

Rank-width. A *layout* of a graph G is a pair (T, L) of a sub-cubic tree T and a bijective function $L : V_G \to N_T^{(1)}$. For an edge e of T, the connected components of $T \backslash e$ induce a bipartition of $N_T^{(1)}$, hence a bipartition (X_e, Y_e) of V_G. The width of an edge e of a layout (T, L) is $\rho_G(X_e) = \rho_G(Y_e)$. The width of a layout (T, L), denoted by $rwd(G, T, L)$, is the maximum width over all edges of T. The *rank-width* of G, denoted by $rwd(G)$, is the minimum width over all layouts of G.

Remark 2.1 *If (T, L) is a layout of G then $(Red(T), L)$ is also a layout of G, and of same width. We can thus assume that if (T, L) is a layout of G then, T is a cubic tree, i.e., each node of T has degree 1 or 3.*

The notion of rank-width was introduced by Oum and Seymour in their investigations of recognition algorithms for graphs of bounded clique-width [28]. The notion of rank-width and of clique-width are equivalent in the sense that a class of simple undirected graphs has bounded rank-width if and only if it has bounded clique-width because $rwd(G) \leq cwd(G) \leq 2^{rwd(G)+1} - 1$ [28]. Hliněný and Oum give in [20] an $O(f(k) \cdot n^3)$-time algorithm that for every undirected graph G with n vertices and every $k \in \mathbb{N}$, checks whether $rwd(G)$ is at most k and if the answer is positive, produces a layout of width at most k. If $rwd(G) > k$, then $cwd(G) > k$ and if $rwd(G) \leq k$ one obtains a term t in $T(F^c_{k'}, C^c_{k'})$ where $k' = 2^{k+1} - 1$, that defines G.

To solve problems definable in MS logic on graphs of bounded rank-width, one can use this term and apply the techniques of Courcelle et al. [9]. In this paper, we give an algebraic characterization of rank-width, which will allow us to solve MS definable problems without transforming the layout of width k into a clique-width expression of width $2^{k+1} - 1$ because an equally useful algebraic expression can be derived.

We recall below the relations between clique-width, rank-width and tree-width [29]. Corneil and Rotics [3] showed that for every k there exists an infinite family of graphs of tree-width k that have clique-width at least $2^{\lfloor \frac{k}{2} \rfloor - 1}$. Combined with the proposition below this shows that the comparison $cwd(G) \leq 2^{rwd(G)+1} - 1$ is essentially optimal. We denote by $twd(G)$ the tree-width of a graph G.

Proposition 2.2 *Let G be an (uncolored) undirected graph. Then*

(1) $cwd(G) \leq 3 \cdot 2^{twd(G)-1}$.
(2) $rwd(G) \leq twd(G) + 1$.

Proof.

(1) See Corneil and Rotics [3].
(2) See Oum [26]. Another proof by Kanté [23] gives $rwd(G) \leq 4 \cdot twd(G) + 2$. □

3 Multiple Colorings and Logically Defined Graph Operations

Handling multiple colorings of vertices with k colors is clearly the same thing as handling colorings with colors in $\{0,1\}^k$. This vectorial approach that we introduce now will be essential in our construction of graph operations characterizing rank-width.

Let k be a positive integer, $\mathbb{B} = \{0,1\}$. A $\mathbb{B}^k$*-coloring* of a graph G is a mapping $\gamma : V_G \to \mathbb{B}^k$ with no constraint on the values of γ for adjacent vertices. We consider that $x \in V_G$ has color i (among others) if and only if $\gamma(x)[i]$ (the i-th component of $\gamma(x)$) is 1. (It is worth noticing that for each $x \in V_G$, $\gamma_G(x)$ is a row vector.)

A $\mathbb{B}^k$*-colored graph* is a triple $G = \langle V_G, edg_G, \gamma_G\rangle$ where γ_G is a $\mathbb{B}^k$-coloring of $\langle V_G, edg_G\rangle$. A graph $G = \langle V_G, edg_G\rangle$ is made canonically into a $\mathbb{B}^k$-colored graph for each k, with $\gamma_G(x) = (0, \cdots, 0)$ for each x. We define some operations on these graphs.

Recoloring. For a mapping $h : \mathbb{B}^k \to \mathbb{B}^m$ and a $\mathbb{B}^k$-colored graph G, we let $Recol_h(G)$ be the $\mathbb{B}^m$-colored graph $K = \langle V_G, edg_G, \gamma_K\rangle$ where $\gamma_K = h \circ \gamma_G$.

Graph products. Let $f : \mathbb{B}^k \times \mathbb{B}^\ell \to \{0,1\}$, $g : \mathbb{B}^k \to \mathbb{B}^m$ and $h : \mathbb{B}^\ell \to \mathbb{B}^m$ be arbitrary mappings. For G, $\mathbb{B}^k$-colored and H, $\mathbb{B}^\ell$-colored, such that $V_G \cap V_H = \emptyset$, we let $G \otimes_{f,g,h} H$ be the $\mathbb{B}^m$-colored graph $K = \langle V_G \cup V_H, edg_K, \gamma_K\rangle$ where:

$$edg_K = edg_G \cup edg_H \cup \{xy \mid x \in V_G, y \in V_H \text{ and } f(\gamma_G(x), \gamma_H(y)) = 1\},$$

$$\gamma_K(x) = \begin{cases} (g \circ \gamma_G)(x) & \text{if } x \in V_G, \\ (h \circ \gamma_H)(x) & \text{if } x \in V_H. \end{cases}$$

Constants. For each $u \in \mathbb{B}^k$ 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 the set $\{\mathbf{u} \mid u \in \mathbb{B}^1 \cup \cdots \cup \mathbb{B}^k\}$. In some occasions we will use a constant $\emptyset_k$ to denote the empty $\mathbb{B}^k$-colored graph.

Remark 3.1 *(1) As in the operations by Wanke [31] (see also [21]) these operations add edges between two disjoint graphs, that are the two arguments of (many) binary operations. This is a difference with clique-width* *where a single binary operation* $\oplus$ *is used, and* $\eta_{i,j}$ *applied to* $G \oplus H$ *may add edges to* G *and to* H*.*

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

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

(4) *The recoloring operations can actually be combined with other operations. The following rules are clear:*

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

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 be the finite set of operations $Recol_h$, $\otimes_{f,g,h}$ where $g : \mathbb{B}^k \to \mathbb{B}^m$, $h : \mathbb{B}^\ell \to \mathbb{B}^m$ and $f : \mathbb{B}^k \times \mathbb{B}^\ell \to \{0,1\}$ 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 $t \in T(B_n, C_n)$ has for value a $\mathbb{B}^n$-colored graph, denoted by $val(t)$, or actually the family of all graphs isomorphic to such a graph [3].

We now explain how such operations fit into the logical framework of Courcelle et al. [9]. 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 $\mathfrak{A} =< A, (R_{\mathfrak{A}})_{R \in \Sigma} >$ where $R_{\mathfrak{A}} \subseteq A^{ar(R)}$. The set A is called the *domain* of $\mathfrak{A}$. A relational Σ-structure $\mathfrak{A} =< A, (R_{\mathfrak{A}})_{R \in \Sigma} >$ is said *binary* if $ar(R) \leq 2$ for each $R \in \Sigma$.

$\mathbb{B}^k$-colored graphs as binary relational structures. Let us introduce unary relations c_i for $i \in [n]$. The meaning of $c_i(x) = \mathbf{true}$ will be "x has color i". Hence a $\mathbb{B}^k$-colored graph $G = \langle V_G, edg_G, \gamma_G \rangle$ for $k \leq n$ is described exactly by the relational structure with domain V_G that we will also denote by G:

$$\langle V_G, edg_G, c_{1G}, \ldots, c_{nG} \rangle.$$

For a $\mathbb{B}^k$-colored graph, $k < n$ the predicates $c_{iG}(x)$ for $k + 1 \leq i \leq n$ will be **false**. Every relational structure of this form, and such that edg_G

[3] We assume terms well-written with respect to the expected types of functions. A more formal treatment would specify B_n as a many-sorted signature.

is symmetric and irreflexive ($edg_G(x,x)$ never holds) represents a $\mathbb{B}^k$-colored graph $G,\ k \leq n$.

We define the notion of quantifier-free operations suited for our purposes (see [1] for general definitions). We let $QF[\Sigma, \{x_1, \ldots, x_n\}]$ denote the set of quantifier-free formulas written with symbols in Σ and variables among $\{x_1, \ldots, x_n\}$. (Examples will be given shortly.) Up to logical equivalence, this set is finite.

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

$$\begin{aligned} \psi &\in QF[\Sigma, \{x\}], \\ \theta_R &\in QF[\Sigma, \{x_1, \ldots, x_{ar(R)}\}] \text{ for each } R \in \Gamma. \end{aligned}$$

Let $\mathfrak{A} = \langle A, (R_{\mathfrak{A}})_{R\in\Sigma}\rangle$ and $\mathfrak{B} = \langle B, (R_{\mathfrak{B}})_{R\in\Gamma}\rangle$ be respectively in $STR[\Sigma]$ and $STR[\Gamma]$. We say that $\mathcal{D}$ defines $\mathfrak{B}$ from $\mathfrak{A}$ if

(i) $B = \{a \mid \mathfrak{A} \models \psi(a)\}$,
(ii) for each $R \in \Gamma$,

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

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

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

Example The $\eta_{i,j}$ operation for simple, loop-free undirected k-graphs can be defined as a quantifier-free operation where:

$$\begin{aligned} \psi &:= \mathbf{true}, \\ \theta_{edg}(x_1,x_2) &:= edg(x_1,x_2) \vee \Big(x_1 \neq x_2 \wedge \Big(\big(c_i(x_1) \wedge c_j(x_2)\big) \vee \big(c_j(x_1) \wedge c_i(x_2)\big)\Big)\Big), \\ \theta_{c_\ell}(x) &:= c_\ell(x) \text{ for } \ell \in [k]. \end{aligned}$$

Proposition 3.2 *For each positive integer n we have:*

(1) The operations $Recol_h$ are quantifier-free operations for any mapping $h : \mathbb{B}^k \to \mathbb{B}^m,\ k, m \leq n$.

(2) *The operations $\otimes_{f,g,h}$ are expressible in terms of $\oplus$ and quantifier-free operations for any mappings $f : \mathbb{B}^k \times \mathbb{B}^\ell \to \{0,1\}$, $g : \mathbb{B}^k \to \mathbb{B}^m$ and $h : \mathbb{B}^\ell \to \mathbb{B}^m$, $k, \ell, m \leq n$.*

Proof.

(1) is clear.

(2) Let n be a fixed positive integer. We consider $\mathbb{B}^k$-colored graphs for $k \leq n$. In addition to the unary predicates $c_1, \ldots, c_n$, we will use auxiliary unary ones $d_1, \ldots, d_n$ ($d_i \notin \{c_1, \ldots, c_n\}$). We have, if $K = G \otimes_{f,g,h} H$:

$$K = \alpha(\eta(G \oplus \beta(H)))$$

where β replaces in H each c_i by d_i (i.e., $d_{i\beta(H)}(x)$ holds if and only if $c_{iH}(x)$ holds, and then, $c_{i\beta(H)}(x)$ is **false**), η creates edges, by redefining $edg(x,y)$ with the following formula where in the definition of edg', u and v range over $\mathbb{B}^n$ (we let $u[i]$ denote the i-th component of u):

$$edg(x,y) \vee edg'(x,y) \vee edg'(y,x)$$

and $edg'(x,y)$ is

$$\bigvee_{f(u,v)=1} \Big(\bigwedge_{u[i]=1} c_i(x) \ \wedge \bigwedge_{u[i]=0} \neg c_i(x) \wedge \bigwedge_{v[j]=1} d_j(y) \wedge \bigwedge_{v[j]=0} \neg d_j(y) \Big).$$

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

Theorem 3.3 *For each monadic second-order graph property P, for each $n \in \mathbb{N}$, there exists an algorithm that checks in time $O(|t|)$ for every term $t \in T(B_n, C_n)$ if the graph defined by this term satisfies P.*

Proof. This result is proved in [9] for $T(F_n^c, C_n^c)$ instead of $T(B_n, C_n)$, but it extends to all quantifier-free definable operations as proved in [4]. The logical foundations of this result are presented in detail by Makowsky in [25]. □

Remark 3.4 *It can be proved that a class of (uncolored) graphs has bounded clique-width if and only if it is defined by a subset of $T(B_n, C_n)$ for some n. The same logical tools yield Theorem 3.3 and its specialization to clique-width bounded graphs. For graphs of bounded rank-width, one needs to express them in some algebraic way, either by clique-width expressions as in [28] or by the algebraic operations to be defined below, that are particular terms in the sets $T(B_n, C_n)$.*

4 Vectorial Colorings and Rank-Width

We specialize the operations defined in the previous section by taking advantage of the vector space structure of $\mathbb{B}^k$ over the field $GF(2)$. As in Section 2.2 the vertex set of each graph is linearly ordered. 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 a $\mathbb{B}^k$-colored graph $G = \langle V_G, edg_G, \gamma_G \rangle$ we associate the (V_G, V_G)-adjacency (symmetric) matrix A_G and the $(V_G, [k])$-*color matrix* Γ_G, the row vectors of which are the vectors $\gamma_G(x)$ in $\mathbb{B}^k$ 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 $\mathbb{B}^k$-colored[4].

Linear recolorings. A recoloring $Recol_h$ is *linear* if $h : \mathbb{B}^k \longrightarrow \mathbb{B}^m$ is linear, in other words, if for some $(k \times m)$-matrix N and all $\mathbb{B}^k$-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 $N' \cdot N$.

Bilinear product of graphs. We consider the operations $\otimes_{f,g,h}$ where:

- $f : \mathbb{B}^k \times \mathbb{B}^\ell \longrightarrow \mathbb{B}$ is *bilinear*, hence defined by $f(u, v) = (u \cdot M) \cdot v^T$ where M is a $(k \times \ell)$-matrix;
- the recoloring maps $g : \mathbb{B}^k \longrightarrow \mathbb{B}^m$ and $h : \mathbb{B}^\ell \longrightarrow \mathbb{B}^m$ 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

[4] The color-rank of G should not be confused with its *rank*, defined as the rank of its adjacency matrix A_G with coefficients in $\{0, 1\}$. All ranks are relative to $GF(2)$.

have thus:

$$A_K = \begin{pmatrix} A_G & \Gamma_G \cdot M \cdot \Gamma_H^T \\ \Gamma_H \cdot M^T \cdot \Gamma_G^T & A_H \end{pmatrix}$$

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

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 $\otimes_{f,g,h}$.

Constants. We recall that for each $u \in \mathbb{B}^k$, $\mathbf{u}$ denotes the graph with a single vertex colored by u and no edge. We also let C_k be the set $\{\mathbf{u} \mid u \in \mathbb{B}^1 \cup \cdots \cup \mathbb{B}^k\}$.

Remark 4.1 *(1) If $K = G \otimes_{M,N,P} H$ is $\mathbb{B}^m$-colored, then we have:*

$$\begin{aligned} A_K[V_G, V_H] &= \Gamma_G \cdot M \cdot \Gamma_H^T, \\ \Gamma_K[V_G, [m]] &= \Gamma_{K[V_G]} = \Gamma_G \cdot N, \\ \Gamma_K[V_H, [m]] &= \Gamma_{K[V_H]} = \Gamma_H \cdot P. \end{aligned}$$

Since for all matrices we have:

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

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 $G \otimes_{M,N,P} H = H \otimes_{M^T,P,N} G$.

(3) The following rules are clear:

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

We let $R_n \subseteq B_n$ 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 $t \in T(R_n, C_n)$. 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 4.2 *We can transform every term $t \in T(R_n, C_n)$ into a term $t' \in T(R_n, C_n)$ where each constant $\mathbf{u} \in \mathbb{B}^n$ and each operation $Recol_N$ or $\otimes_{M,N,P}$ are such that M, N and P are $(n \times n)$-matrices. For that, we use the following recursive rules:*

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

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 by R_n' consisting of bilinear products $\otimes_{M,N,P}$ where M, N and P are $(n \times n)$-matrices and may assume that for each $\mathbf{u} \in C_n$, $u \in \mathbb{B}^n$. We use here also Remark 4.1(1) showing that recolorings can be combined with bilinear products.

Our objective is to prove the following which is our main theorem.

Theorem 4.3 (Main Theorem) *A graph G has rank-width at most n if and only if it is the value of a term in $T(R_n, C_n)$.*

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

Proposition 4.4 *Let $G = val(t)$ where $t \in T(R_n, C_n)$. Then $rwd(G) \leq n$.*

We recall that $s \bullet t = s[t/u]$ for $s \in Cxt(R_n, C_n)$, $t \in T(R_n, C_n)$ and Id is the particular context u. Before proving Proposition 4.4, we state and prove the following lemma.

Lemma 4.5 *Let $t = c \bullet t'$ where $t' \in T(R_n', C_n)$, $c \in Cxt(R_n', C_n) - \{Id\}$. If* *we let $G = val(t)$ and $H = val(t')$ then:*

$$\begin{aligned} A_G[V_H, V_G - V_H] &= \Gamma_H \cdot B, \\ \Gamma_{G[V_H]} &= \Gamma_H \cdot C \end{aligned}$$

for some matrices B and C.

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,

$$\begin{aligned} A_G[V_H, V_K] &= \Gamma_H \cdot M \cdot \Gamma_K^T, \\ \Gamma_{G[V_H]} &= \Gamma_H \cdot N. \end{aligned}$$

Hence we take $B = M \cdot \Gamma_K^T$ and $C = N$.

The proof is similar if $c = t'' \otimes_{M,N,P} Id$, we take $B = M^T \cdot \Gamma_K^T$ because $A_G[V_H, V_K] = \Gamma_H \cdot M^T \cdot \Gamma_K^T$ and $C = P$.

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

$$A_G = \begin{pmatrix} A_{G'} & \Gamma_{G'} \cdot M \cdot \Gamma_K^T \\ \Gamma_K \cdot M^T \cdot \Gamma_{G'}^T & A_K \end{pmatrix}$$

Hence

$$A_G[V_H, V_G - V_H] = \begin{pmatrix} A_{G'}[V_H, V_{G'} - V_H] & \left(\Gamma_{G'} \cdot M \cdot \Gamma_K^T\right)[V_H, V_K] \end{pmatrix}$$

By inductive hypothesis, $A_{G'}[V_H, V_{G'} - V_H] = \Gamma_H \cdot B'$. We now prove that $\left(\Gamma_{G'} \cdot M \cdot \Gamma_K^T\right)[V_H, V_K] = \Gamma_H \cdot C''$ for some matrix 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 $A_G[V_H, V_G - V_H] = \Gamma_H \begin{pmatrix} B' & C' \cdot M \cdot \Gamma_K^T \end{pmatrix}$.

We now consider $\Gamma_{G[V_H]}$. We have:

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

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

This proves the lemma, because the case of $c = t'' \otimes_{M,N,P} c'$ is similar. □

We can now prove Proposition 4.4.

Proof of Proposition 4.4. Let $G = val(t)$ where $t \in T(R_n, C_n)$. We transform it into a term $\tilde{t}$ in $T(R'_n, C_n)$ with $red(\tilde{t}) = red(t)$. By definition there exists a bijection L between V_G and the set of constants of $red(\tilde{t})$. We take $(red(\tilde{t}), L)$ as a layout of G. We claim that the width of this layout is at most n. Hence we have to prove that for each subterm t' of t

$$rk(A_G[V_{val(t')}, V_G - V_{val(t')}]) \leq n.$$

Let t' be a subterm of t and let $H = val(t')$. By Lemma 4.5 we have $A_G[V_H, V_G - V_H] = \Gamma_H \cdot B$. Hence $rk(A_G[V_H, V_G - V_H]) \leq n$ since each H is $\mathbb{B}^n$-colored. □

For proving the “only if direction” stated as Proposition 4.13, 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 $(0, \cdots, 0)$. We recall that for $X \subseteq V_G$ we denote by $\rho_G(X)$ the rank of $A_G[X, V_G - X]$ (cf. the definition of rank-width in Section 2.2).

Let G be a graph and (V_1, V_2) be a bipartition of its vertices such that $\rho_G(V_1) = m$. 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 $A_G[V_1, V_2]$ are independent. Vertices $x_1, \ldots, x_p$ in V_1 with $p \geq \rho_G(V_1)$ form a *vertex generator of* $G[V_1]$ *with respect to* G if their associated row vectors generate the row vectors of $A_G[V_1, V_2]$.

If A is a (X, Y)-matrix (or equivalently a $(|X| \times |Y|)$-matrix) we let $A[x, Y]$ denote the row vector of A corresponding to $x \in X$ and let $A[X, y]$ denote the column vector corresponding to $y \in Y$. We now introduce the notion of *presentation*, which will allow us to construct a term in $T(R_n, C_n)$ from a layout by induction.

Definition 4.6 (Presentation) *Let* (V_1, V_2) *be a bipartition of* V_G *with* $A = A_G[V_1, V_2]$. *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 space vector* *as the set of all row vectors of A. It follows that $A = N \cdot A[X, V_2]$ for some (V_1, X)-matrix N. We define N by $N[x, z] = b_{xz}$ if $z \in X$ and $x \notin X$ where:*

$$A[x, V_2] = b_{xz_1}A[z_1, V_2] + \cdots + b_{xz_p}A[z_p, V_2]$$

and

$$\begin{cases} b_{zz} = 1 & \textit{if } z \in X \\ b_{zz'} = 0 & \textit{if } z, z' \in X \textit{ and } z \neq z'. \end{cases}$$

Let us enumerate the elements of V_1 as $v_1, \ldots, v_n$. Let H' be a $\mathbb{B}^n$-coloring of $G[V_1]$ such that $\gamma_{H'}(v_i) = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 at i-th position and $H = Recol_N(H')$. We call (H, N, X) a presentation of $G[V_1]$ relative to G. *It is clear that H is a $\mathbb{B}^p$-coloring of $G[V_1]$ and $\Gamma_H = N$.*

Proposition 4.7 *Let G be a graph and let (V_1, V_2) be a bipartition 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 $A_G[V_1, V_2] = NMP^T$ where $M = A_G[X, Y]$ and $G = H \otimes_M K$.*

Proof. By definition, that (H, N, X) is a presentation of $G[V_1]$ relative to G means that $A_G[V_1, V_2] = N \cdot A_G[X, V_2]$. Similarly $A_G[V_2, V_1] = P \cdot A_G[Y, V_1]$ since (K, P, Y) is a presentation of $G[V_2]$ relative to G. Then $A_G[V_2, X] = P \cdot A_G[Y, X]$, i.e., $A_G[X, V_2] = A_G[X, Y] \cdot P^T$. We let $M = A_G[X, Y]$. Hence $A_G[V_1, V_2] = NMP^T$.

We now prove that $G = H \otimes_M K$. It is sufficient to prove that $A_G[V_1, V_2] = A_{G'}[V_1, V_2]$ where $G' = H \otimes_M K$. By definition of a presentation, $\Gamma_H = N$ and $\Gamma_K = P$. Hence $A_{G'}[V_1, V_2] = \Gamma_H \cdot M \cdot \Gamma_K^T = NMP^T = A_G[V_1, V_2]$. □

Remark 4.8 *(1) If $k = rk(A_G[V_1, V_2])$ in Proposition 4.7, 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 $V_1 \subset V$ 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 We let G be such that $V_1 = \{a, b, c, d\}$, $V_2 = \{\alpha, \beta, \gamma, \delta, \mu\}$ and

$$A_G[V_1, V_2] = A = \begin{array}{|c|ccc|cc|}\hline & \alpha & \beta & \gamma & \delta & \mu \\ \hline a & 1 & 1 & 0 & 0 & 1 \\ b & 1 & 0 & 0 & 1 & 0 \\ c & 0 & 0 & 1 & 0 & 1 \\ \hline d & 1 & 1 & 1 & 0 & 0 \\ \hline \end{array}$$

We choose $\{a, b, c\}$ and $\{\alpha, \beta, \gamma\}$ as vertex bases of $G[V_1]$ and $G[V_2]$ respectively. We have thus $A[d, V_2] = A[a, V_2] + A[c, V_2]$, and, $A^T[\delta, V_1] = A^T[\alpha, V_1] + A^T[\beta, V_1]$ and $A^T[\mu, V_1] = A^T[\beta, V_1] + A^T[\gamma, V_1]$.

The corresponding $\mathbb{B}^3$-colorings of $G[V_1]$ and $G[V_2]$ are respectively defined by:

$$\begin{array}{|c|ccc|}\hline a & 1 & 0 & 0 \\ \hline b & 0 & 1 & 0 \\ \hline c & 0 & 0 & 1 \\ \hline d & 1 & 0 & 1 \\ \hline \end{array} \qquad \begin{array}{|c|ccc|}\hline \alpha & 1 & 0 & 0 \\ \hline \beta & 0 & 1 & 0 \\ \hline \gamma & 0 & 0 & 1 \\ \hline \delta & 1 & 1 & 0 \\ \hline \mu & 0 & 1 & 1 \\ \hline \end{array}$$

The matrix M is $A[\{a, b, c\}, \{\alpha, \beta, \gamma\}]$. We can check for an example that $A[d, \mu] = (1, 0, 1) \cdot M \cdot (0, 1, 1)^T = (1, 1, 1) \cdot (0, 1, 1)^T = 1 + 1 = 0$.

We can now state some basic properties of presentations.

Fact 4.9 *Let G be a graph with a bipartition (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 $M = A_G[X, Y]$. Then:*

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

Fact 4.10 *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:*

$$\begin{aligned} A_G[V, V_G - V] &= N \cdot A_G[X, V_G - V], \\ A_G[X, V_G - V] &= N' \cdot A_G[X', V_G - V], \end{aligned}$$

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

$$A_G[V, V_G - V] = (N \cdot N') \cdot A_G[X', V_G - V].$$

Proposition 4.11 *Let $V \subseteq V_G$ and (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 a (X_1, X_2)-matrix, P_1 is a (X_1, Z)-matrix and P_2 is a (X_2, Z)-matrix.

Proof. We let $h = |X_1|$, $k = |X_2|$, $n = |V_1|$, $m = |V_2|$ and $|V_G - V| = p$. By Proposition 4.7 we have $G = H_1 \otimes_{M'} K$ where $M' = A_G[X_1, X_2 \cup (V_G - V)]$ and $(K, N_2', X_2 \cup (V_G - V))$ is a presentation of $G[V_G - V_1]$ relative to G with:

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

Hence by Fact 4.9 $G[V] = (H_1 \otimes_{M'} K)\,[V_1 \cup V_2] = H_1 \otimes_M H_2$ where $M = M'[X_1, X_2]$ since

$$\begin{aligned} K[V_2] &= H_2 \\ N_2'[V_2, X_2] &= N_2 \\ (X_2 \cup (V_G - V)) \cap V_2 &= X_2. \end{aligned}$$

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:

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

Let $X_1 = \{x_1, \ldots, x_h\}$, $X_2 = \{y_1, \ldots, y_k\}$ and $\ell = \rho_G(V)$. We let $A = A_G[V, V_G - V]$.

Claim 4.12 *$X_1 \cup X_2$ is a vertex generator of $G[V]$ with respect to G.*

Proof of Claim 4.12. We consider the matrix $A_G[V_1, V_G - V_1]$. Its row vectors are generated by those associated with X_1. Thus, so are those of $A_G[V_1, V_G - V]$ which are projections of the latter ones. Similarly the row vectors of $A_G[V_2, V_G - V]$ 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 $h' \leq h$ and $k' \leq k$. Let $W = V_G - V$. For each s such that $h' < s \leq h$, we have a vector w_s such that the row vector $A[x_s, W]$ is:

$$A[x_s, W] = w_s \cdot \Big(A[x_1, W], \cdots, A[x_{h'}, W], A[y_1, W], \cdots, A[y_{k'}, W]\Big)^T$$

We let

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

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

$$A[y_u, W] = w'_u \cdot \Big(A[x_1, W], \cdots, A[x_{h'}, W], A[y_1, W], \cdots, A[y_{k'}, W]\Big)^T$$

We let

$$P_2 = \left(\begin{array}{c|c} 0_{k',\ell-k'} & I_{k'} \\ \hline \multicolumn{2}{c}{w'_{k'+1}} \\ \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{w'_k} \end{array}\right).$$

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:

$$A[x, z] = \gamma_H(x) \cdot \Big(A[x_1, z], \cdots, A[x_{h'}, z], A[y_1, z], \cdots, A[y_{k'}, z]\Big)^T$$

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

$$A_G[x, z] = \gamma_{H_1}(x) \cdot \Big(A_G[x_1, z], \cdots, A_G[x_h, z]\Big)^T$$

But P_1 is defined in such a way that:

$$\begin{pmatrix} A_G[x_1,z] \\ \vdots \\ A_G[x_h,z] \end{pmatrix} = P_1 \cdot \Big(A_G[x_1,z],\cdots,A_G[x_{h'},z],A_G[y_1,z],\cdots,A_G[y_{k'},z]\Big)^T$$

Hence

$$A_G[x,z] = \gamma_{H_1}(x)\cdot P_1 \cdot \Big(A[x_1,z],\cdots,A[x_{h'},z],A[y_1,z],\cdots,A[y_{k'},z]\Big)^T$$

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

The proof is similar for $A_G[y,z]$ for $y \in V_2$ and $z \in W$. This terminates the proof of the proposition. □

Example We let $V_1 = \{1,2,\ldots,5\}$, $V_2 = \{a,b,c,d,e\}$, $W = V_G - V = \{\alpha,\beta,\gamma,\delta\}$. The matrix $A_G[V_1 \cup V_2, V_2 \cup W]$ is:

$$A_G = \begin{array}{|c|ccc|cc|c|c|c|c|}
\hline
 & a & b & c & d & e & \alpha & \beta & \gamma & \delta \\
\hline
1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
2 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
3 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
\hline
4 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
5 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
\hline
a & & & & & & 0 & 1 & 1 & 1 \\
\hline
b & & & & & & 1 & 1 & 1 & 0 \\
\hline
c & & & & & & 1 & 0 & 1 & 1 \\
\hline
d & & & & & & 1 & 0 & 0 & 1 \\
\hline
e & & & & & & 0 & 0 & 1 & 0 \\
\hline
\end{array}$$

We can leave undefined the sub-matrix $A_G[V_2, V_2]$. We have the following linear relations between rows and columns of $A_G[V_1, V_2 \cup W]$ and $A_G[V_2, V_1 \cup W]$:

$$\begin{aligned} 4 &= 1 + 2 \\ 5 &= 2 + 3 \\ d &= a + b \\ e &= a + b + c \end{aligned}$$

Vertex bases are for $G[V_1] : \{1, 2, 3\}$ and for $G[V_2] : \{a, b, c\}$. Among $\{1, 2, 3, a, b, c\}$, one can select $\{1, 2, a\}$ as a vertex basis of $G[V_1 \cup V_2]$. Then we have:

$$M = \begin{array}{|c|ccc|}\hline & a & b & c \\ \hline 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 \\ 3 & 0 & 1 & 1 \\ \hline \end{array} \qquad P_1 = \begin{array}{|c|ccc|}\hline & 1 & 2 & a \\ \hline 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \\ \hline \end{array} \qquad P_2 = \begin{array}{|c|ccc|}\hline & 1 & 2 & a \\ \hline a & 0 & 0 & 1 \\ b & 1 & 0 & 1 \\ c & 0 & 1 & 1 \\ \hline \end{array}$$

We can now prove the converse direction of the main theorem.

Proposition 4.13 *Every graph of rank-width at most n is the value of a term in $T(R_n, C_n)$.*

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

Let (T, L) be a layout of G of width n (we can assume T cubic by Remark 2.1). Let us select a leaf s 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/u, the subtree of the directed tree T rooted at u.

For every edge of T of the form $\overrightarrow{vu}$, we let G_u be the induced sub-graph of G, the vertices of which are the leaves of T/u, and G^v the induced sub-graph of G, the vertices of which are the leaves not in T/u.

Claim 4.14 *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 $T(R_n, C_n)$ 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 4.14. 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 $t_u = \mathbf{1}$.

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 4.11 there exists a $(r(w) \times r(w'))$-matrix M, a $(r(w), r(u))$-matrix N_1 and a $(r(w'), r(u))$-matrix N_2 such that (H_u, N_u, X_u) is a presentation of G_u where

$$\begin{aligned} &H_u = H_w \otimes_{M,N_1,N_2} H_{w'} \\ &N_u = \begin{pmatrix} \Gamma_{H_w} \cdot N_1 \\ \Gamma_{H_{w'}} \cdot N_2 \end{pmatrix}, \\ &X_u \subseteq X_w \cup X_{w'} \text{ is a vertex basis of } G_u. \end{aligned}$$

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 and the claim. □

We can now finish the proof of the proposition. For the case of u where $r \to u$, r is the root of T, we have G^r defined by $\mathbf{1}$ and G defined by $t_u \otimes_{1,O,O} \mathbf{1}$, where t_u is obtained by Claim 4.14.

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. □

Remark 4.15 *(1) Any layout (T, L) of G of width n, where T is cubic, is made into a term t such that $Red(red(t)) = T$.*

(2) The procedure that transforms a layout (T, L) of width k of a graph G into a term t in $T(R_k, C_k)$ 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, we need a vertex basis B_1 of $A_G[V_1, V_G - V_1]$, a vertex basis B_2 of $A_G[V_2, V_G - V_2]$ and a vertex basis B_3 of $A_G[V_1 \cup V_2, W]$. By Proposition 4.11 we can construct a vertex basis of $A_G[V_1 \cup V_2, W]$ in time $f(k) \cdot |V_G|$, for some function f, by using the vertex bases B_1 and B_2. If $V_1 = \{x\}$ then $B_1 = \{x\}$ (G is connected). Hence for each node of T we can construct the matrices M, N and P in time $O(f(k) \cdot |V_G|)$. Since $|T| = O(|V_G|)$, we can construct the term t in time $O(|V_G|^2)$.

5 Edge Colored Graphs

Let $A = \{a_1, \ldots, a_p\}$ be a finite set of edge-colors. We extend some of the previous definitions and results to graphs such that each undirected edge has a color in A. We may have parallel edges with distinct colors.

For defining such graphs by clique-width expressions, we use the operations $\eta^a_{i,j}$ that add a-colored edges between i-vertices and j-vertices. The corresponding notion of clique-width follows immediately.

The operations R_k can be modified as follows. Instead of operations $\otimes_{M,N,P}$ we use operations $\otimes_{M_1,\ldots,M_p,N,P}$ where $M_1, \ldots, M_p$ are $(k \times \ell)$-matrices (see Section 4) and M_i is used to create a_i-colored edges, as does M in $\otimes_{M,N,P}$ for creating ordinary edges. We denote by $R^{(A)}_k$ the corresponding set of binary operations. We obtain in this way a complexity measure on A-colored graphs:

$$Rwd^{(A)}(G) = \min\{k \mid G = val(t),\ t \in T(R^{(A)}_k, C_k)\}$$

If G is A-colored, 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.

It is clear that if $G = val(t)$, $t \in T(R^{(A)}_k, C_k)$ and $a \in A$ then $G_a = val(t_a)$ where $t_a \in T(R_k, C_k)$ is obtained from t by replacing each operation $\otimes_{M_1,\ldots,M_p,N,P}$ by $\otimes_{M,N,P}$ where $M = M_i$ and $a = a_i$. It follows that $rwd(G_a) \leq Rwd^{(A)}(G)$ for each $a \in A$.

We extend as follows the definition of rank-width for edge-colored graphs.

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

Proposition 5.1 *Let A be a set of p edge-colors. For every edge-colored graph with edge-colors in A we have:*

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

Proof. Let $t \in T(R^{(A)}_k, C_k)$ be a term defining G, made into a layout (T, L) of G as in the proof of Theorem 4.3. Clearly (T, L) is also a layout of G_a, the one derived from t_a for each $a \in A$. Hence for each a, $rwd(G_a, T, L) \leq k$ since $t_a \in T(R_k, C_k)$. Hence $rwd(G) \leq \max_{a\in A}\{rwd(G_a, T, L)\} \leq k$, which proves that $rwd(G) \leq Rwd^{(A)}(G)$.

For the inequality $Rwd^{(A)}(G) \leq p \cdot rwd(G)$ let us consider a layout (T, L) of G that witnesses $rwd(G) = k$. Hence $rwd(G_a, T, L) \leq k$ for each $a \in A$. For each $a \in A$ there exists by Proposition 4.13 a term $t_a \in T(R_k, C_k)$ 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 $A = \{a, b\}$. The extension to $p > 2$ will be straightforward.

We now show how t_a and t_b can be merged into a single term. 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 G_a, G_b have color matrices $\Gamma_{G_a}, \Gamma_{G_b}$ both of order $(|V_G| \times k)$. Assume 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 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 5.2 *$K = G \otimes_{M,M',N,P} H$ 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 5.2. 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 $T(R_{2k}^{(A)}, C_{2k})$ that defines G. Note that $red(t) = red(t_a)$. □

6 Conclusion

Rank-width is an interesting complexity measure because it is equivalent to clique-width, and also because it is increasing for vertex-minor inclusion [27] and has a cubic-time verification algorithm [20]. Contrary to clique-width, rank-width was not initially defined in terms of graph operations but in terms of layouts and associated ranks of $GF(2)$-matrices. In order to apply the results of [9] one needs to transform the layout of a graph into a clique-width expression, and this transformation may introduce 2^{k+1} colors for graphs of rank-width k (see [28]). By the results of this article we can now find FPT algorithms for problems expressible in monadic second-order logic by transforming the layout more directly into an algebraic expression based on disjoint union and quantifier-free operations. Indeed, we define natural graph operations based on linear transformations that characterize rank-width. These operations are specializations of more general operations based on disjoint union and vertex-colorings. This allows us to compare several notions of width. However the same classes of graphs have bounded width.

The extension of these results to directed graphs will be considered in a future article [23]. The results of the extended abstract [7] about balanced terms is independent from the one presented here and will be considered in a second article [8]. We will prove that every graph G of rank-width k is defined by a term in $T(R_{2k}, C_{2k})$ of height $3 \cdot \log(|V_G| + 1)$.

Acknowledgments. We would like to thank the referees for their useful comments.

References

[1] A. Blumensath and B. Courcelle. Recognizability, Hypergraph Operations and Logical Types. *Information and Computation*, 204(6):853-919, 2006.

[2] 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 eds., *Latin American Symposium on Theoretical Informatics*, volume 1776 of *LNCS*, pages 126-134. Springer, 2000.

[3] D. G. Corneil and U. Rotics. On the Relationship between Clique-Width and Tree-Width. *SIAM Journal on Computing*, 34(4):825-847, 2005.

[4] B. Courcelle. The Monadic Second-Order Logic of Graphs VII: Graphs as Relational Structures. *Theoretical Computer Science*, 101(1):3-33, 1992.

[5] B. Courcelle. Graph Grammars, Monadic Second-Order Logic and the Theory of Graph Minors. In N. Robertson and P. Seymour eds., *Graph Structure Theory*, volume 147 of *Contemporary Mathematics*, pages 565-590. AMS, 1993.

[6] B. Courcelle. The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic. In G. Rozenberg ed., *Handbook of Graph Grammars and Computing by Graph Transformations*, pages 313-400. World Scientific, 1997.

[7] B. Courcelle and M.M. Kanté. Graph Operations Characterizing Rank-Width and Balanced Graph Expressions. In A. Brandstädt, D. Kratsch and H. Müller

eds., *Graph-Theoretic Concepts in Computer Science*, volume 4769 of *LNCS*, pages 66-75. Springer, 2007.

[8] B. Courcelle and M.M. Kanté. Balanced Graph Expressions. Manuscript, 2007.

[9] 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.

[10] B. Courcelle and S. Olariu. Upper Bounds to the Clique-Width of Graphs. *Discrete Applied Mathematics*, 101(1-3):77-114, 2000.

[11] B. Courcelle and A. Twigg. Compact Forbidden-Set Routing. In W. Thomas and P. Weil eds., *Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *LNCS*, pages 37-48. Springer, 2007.

[12] B. Courcelle and R. Vanicat. Query Efficient Implementation of Graphs of Bounded Clique-Width. *Discrete Applied Mathematics*, 131(1):129-150, 2003.

[13] B. Courcelle and P. Weil. The Recognizability of Sets of Graphs is a Robust Property. *Theoretical Computer Science*, 342(2-3):173-228, 2005.

[14] R. Diestel. *Graph Theory*. 3^{rd} Edition, Springer-Verlag, 2005. Readable for free on: *http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/*.

[15] R. Downey and M. Fellows. *Parameterized complexity*. Springer-verlag, 1999.

[16] W. Espelage, F. Gurski and E. Wanke. Deciding Clique-Width for Graphs of Bounded Tree-Width. *Journal of Graph Algorithms and Applications*, 7(2):141-180, 2003.

[17] M. Fellows, F. A. Rosamond, U. Rotics and S. Szeider. Clique-Width Minimization is NP-Hard. In J. M. Kleinberg ed., *Symposium on Theory Of Computing*, pages 354-362. ACM, 2006.

[18] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer Verlag, 2006.

[19] 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 eds., *Tools and Algorithms for Construction and Analysis of Systems*, volume 1019 of *LNCS*, pages 89-110. Springer, 1995.

[20] P. Hliněný and S. Oum. Finding Branch-Decompositions and Rank-Decompositions. In L. Arge, M. Hoffmann and E. Welzl eds, *European Symposium on Algorithms*, volume 4698 of *LNCS*, pages 163-174. Springer, 2007.

[21] Ö. Johansson. Clique-Decomposition, NLC-Decomposition and Modular Decomposition - Relationships and Results for Random Graphs. *Congressus Numerantium*, 132:39-60, 1998.

[22] M.M. Kanté. Vertex-Minor Reductions can Simulate Edge Contractions. *Discrete Applied Mathematics*, 155(17):2328-2340, 2007.

[23] M.M. Kanté. The Rank-Width of Directed Graphs. Submitted, 2008.

[24] V. Lozin and D. Rautenbach. The Relative Clique-Width of Graphs. *Journal of Combinatorial Theory, Series B*, 97(5):846-858, 2007.

[25] J. A. Makowsky. Algorithmic Uses of the Feferman-Vaught Theorem. *Annals of Pure and Applied Logic*, 126(1-3):159-213, 2004.

[26] S. Oum. Rank-width is Less Than or Equal to Branch-Width. *Journal of Graph Theory*, 57(3):239-244, 2007.

[27] S. Oum. Rank-Width and Vertex-Minors. *Journal of Combinatorial Theory, Series B*, 95(1):79-100, 2005.

[28] S. Oum and P. Seymour. Approximating Clique-Width and Branch-Width. *Journal of Combinatorial Theory, Series B*, 96(4):514-528, 2006.

[29] N. Robertson and P. Seymour. Graph Minors V: Excluding a Planar Graph. *Journal of Combinatorial Theory, Series B*, 41(1):92-114, 1986.

[30] N. Robertson and P. Seymour. Graphs minors X: Obstructions to Tree-Decompositions. *Journal of Combinatorial Theory, Series B*, 52(2):153-190, 1991.

[31] E. Wanke. k-NLC Graphs and Polynomial Algorithms. *Discrete Applied Mathematics*, 54(2-3):251-266, 1994.