On Parse Trees and Myhill–Nerode–type Tools for handling Graphs of Bounded Rank-width*

Robert Ganian and Petr Hliněný**

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
ganian@mail.muni.cz, hlineny@fi.muni.cz

December 8, 2008

Abstract. Rank-width is a structural graph measure introduced by Oum and Seymour and aimed at better handling of graphs of bounded clique-width. We propose a formal framework and tools for easy design of dynamic algorithms running directly on a rank-decomposition of a graph (on contrary to the usual approach which translates a rank-decomposition into a clique-width expression, with a possible exponential jump in the parameter). Our new approach links to a previous work of Courcelle and Kanté [WG 2007] who first proposed algebraic expressions with a so-called bilinear graph product as a better way of handling rank-decompositions.

Keywords: Parameterized algorithm, rank-width, graph colouring, Myhill–Nerode theorem.

1 Introduction

Most graph problems are known to be NP-hard in general, and yet a solution to these is needed for practical applications. One common method to provide such a solution is through restricting the input graph to have a certain structure. Often the input graphs are restricted to have bounded tree-width [21] (or branch-width), but another weaker useful structural restriction has appeared with the notion of *clique-width*, defined by Courcelle and Olariu in [9].

Now, many hard graph problems (particularly all those expressible in MS_1 logic, see Section 4) are solvable in polynomial time [8,

* This is the full extended version of a paper presented at IWOCA 2008.

** Both authors are supported by the Institute for Theoretical Computer Science ITI, project 1M0545, and the research grant GAČR 201/08/0308.

11, 18, 14], as long as the input graph has bounded clique-width and is given in the form of the "decomposition for clique-width", called a *k-expression*. A k-expression is an algebraic expression with the following four operations on vertex-labeled graphs using k labels: create a new vertex with label i; take the disjoint union of two labeled graphs; add all edges between vertices of label i and label j; and relabel all vertices with label i to have label j. However, for fixed $k > 3$, it is not known how to find a k-expression of an input graph having clique-width at most k.

Rank-width (see Section 2) is another graph complexity measure introduced in 2003 by Oum and Seymour [20, 19], aimed at providing an $f(k)$-expression of the input graph having clique-width k for some fixed function f in polynomial time. Furthermore, rank-width can be computed, together with an optimal decomposition, in time $O(n^3)$ on n-vertex graphs of bounded rank-width [17]. Since, in reality, clique-width can be up to exponentially larger than rank-width [5], it now appears desirable to design algorithms running directly on an optimal rank-decomposition rather than transforming a width-k rank-decomposition into an $f(k)$-expression, with $f(k)$ up to $2^{k+1}-1$ by [20], cf. also [5].

Unfortunately, the latter goal seems impossible in a direct way given the rather "strange nature" of a rank-decomposition, and so one has to look for suitable indirect alternatives. Courcelle and Kanté [7] in 2007 gave an alternative characterization of a rank-decomposition using bilinear product terms over multi-coloured graphs—see Section 2 and particularly Theorem 2. In our view, the latter characterization can be equivalently formulated in terms of *labeling parse trees* (rank-width parse trees of [13]), which straightforwardly leads to a new Myhill–Nerode–type characterization of finite state properties of graphs of bounded rank-width in Theorem 3, and which opens new mathematical ground for easier algorithmic design in the subsequent sections.

We now outline the structure of our paper: After providing some technical definitions and basic known results in Section 2, we state in Section 3 a useful characterization (Theorem 3) of the regular, i.e. decidable by tree automata, properties of bounded rank-width graphs. Subsequently in Section 4, we prove that any MS_1 formula (not neccessarily closed) defines a regular language over "equipped" bounded rank-width graphs. That, particularly, provides an alternative combinatorial proof of Courcelle, Makowsky, and Rotics' [8] results.

However, algorithms coming from those generic results are not much practical since their runtime dependence on the rank-width could be enormous, see a discussion in Section 5. The advantage of our approach is that the formal tools we develop here can directly produce actual algorithms for particular problems on bounded rank-width graphs with better runtime. We illustrate this in Section 5 with the fast FPT algorithm of Theorem 12 for c-colourability (fixed $c \geq 3$), and with the pseudopolynomial algorithm of Theorem 14 for the chromatic number which outperforms the previous algorithm of [18] with respect to rank-width.

2 Definitions and Basics

We consider finite simple undirected graphs by default. In this section we bring up some (maybe less known) definitions and previous claims which are the building blocks of our research. We particularly pay attention to branch- and rank-decompositions of graphs, and extend their scope to "parse trees" which are more suitable for handling of such decompositions with the tools of traditional automata theory in coming Sections 3,4.

Branch-width. A set function $f : 2^M \to \mathbb{Z}$ is called *symmetric* if $f(X) = f(M \setminus X)$ for all $X \subseteq M$. A tree is *subcubic* if all its nodes have degree at most 3. For a symmetric function $f : 2^M \to \mathbb{Z}$ on a finite set M, the branch-width of f is defined as follows.

A *branch-decomposition* of f is a pair (T, μ) of a subcubic tree T and a bijective function $\mu : M \to \{t : t \text{ is a leaf of } T\}$. For an edge e of T, the connected components of $T \setminus e$ induce a bipartition (X, Y) of the set of leaves of T. The *width* of an edge e of a branch-decomposition (T, μ) is $f(\mu^{-1}(X))$. The *width* of (T, μ) is the maximum width over all edges of T. The *branch-width* of f is the minimum of the width of all branch-decompositions of f. (If $|M| \leq 1$, then we define the branch-width of f as $f(\emptyset)$.)

A natural application of this definition is the branch-width of a graph, as introduced by Robertson and Seymour [21] along with better known tree-width, and its natural matroidal counterpart. In that case we use $M = E(G)$, and f the connectivity function of G. There is, however, another interesting application of the aforementioned general notions, in which we consider the vertex set $V(G) = M$ of a graph G as the ground set.

Rank-width. For a graph G, let $\boldsymbol{A}_G[U, W]$ be the bipartite adjacency matrix of a bipartition (U, W) of the vertex set $V(G)$ defined over the two-element field GF(2) as follows: the entry $a_{u,w}$, $u \in U$ and $w \in W$, of $\boldsymbol{A}_G[U, W]$ is 1 if and only if uw is an edge of G. The *cut-rank* function $\rho_G(U) = \rho_G(W)$ then equals the rank of $\boldsymbol{A}_G[U, W]$ over GF(2). A *rank-decomposition* and *rank-width* of a graph G is the branch-decomposition and branch-width of the cut-rank function ρ_G of G on $M = V(G)$, respectively.

The main reason for the popularity of rank-width over clique-width is the fact that there are parameterized algorithms for rank-decompositions [20, 17].

Theorem 1 (Hliněný and Oum [17]). *For every fixed t there is an $O(n^3)$-time algorithm that, for a given n-vertex graph G, either finds a rank-decomposition of G of width at most t, or confirms that the rank-width of G is more than t.*

Few rank-width examples. Any complete graph of more than one vertex has clearly rank-width 1 since any of its bipartite adjacency matrices consists of all 1s. It is similar with complete bipartite graphs if we split the decomposition along the parts. We illustrate the situation with graph cycles: while C_3 and C_4 have rank-width 1, C_5 and all longer cycles have rank-width equal 2. A rank-decomposition of, say, the cycle C_5 is shown in Fig. 1. Conversely, every subcubic tree with at least 4 leaves has an edge separating at least 2 leaves on each side, and every corresponding bipartition of C_5 gives a matrix of rank ≥ 2.

One may also mention *distance-hereditary* graphs, i.e. graphs such that the distances in any of their connected induced subgraphs are the same as in the original graph, which have been independently

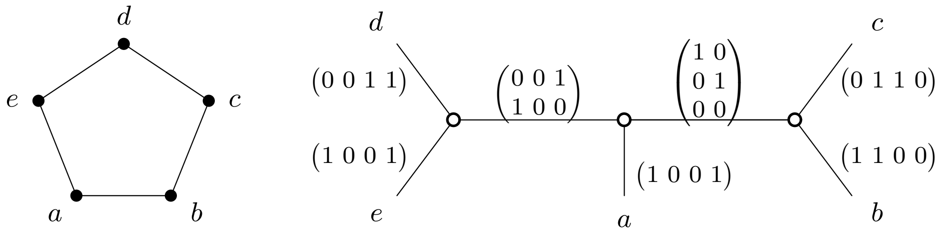

Fig. 1. A rank-decomposition of the graph cycle C_5.

studied, e.g. [4], before. It turns out that distance-hereditary graphs are exactly the graphs of rank-width one [19], and this simple fact explains many of their "nice" algorithmic properties.

Labeling parse trees. In a search for a "more suitable form" of a rank-decomposition, Courcelle and Kanté [7] defined the bilinear products of multiple-coloured graphs, and proposed algebraic expressions over these operators as an equivalent description of a rank-decomposition (cf. Theorem 2). Here we introduce (following [13]) the same idea in terms of parse trees which we propose as a more convenient notation for the results in the next sections.

A (vertex) *t-labeling* of a graph is a mapping $lab : V(G) \to 2^{L_t}$ where $L_t = \{1, 2, \ldots, t\}$ is the set of *labels* (this notion is exactly equivalent to multiple-coloured graphs of [7]). Having a graph G with an (implicitly) associated t-labeling lab, we refer to the pair (G, lab) as to a *t-labeled graph* and use notation $\bar{G}$. Notice that each vertex of a t-labeled graph may have zero, one or more labels. So even an unlabeled graph can be considered as t-labeled with no labels, and every t-labeled graph is also t'-labeled for all $t' > t$. We will often view a t-labeling of G equivalently as a mapping $V(G) \to GF(2)^t$ to the *binary vector space* of dimension t (cf. [7] again).

A *t-relabeling* is a mapping $f : L_t \to 2^{L_t}$. For a t-labeled graph $\bar{G} = (G, lab)$ we define $f(\bar{G})$ as the same graph with a vertex t-labeling $lab' = f \circ lab$. Since lab maps into subsets of L_t which are interpretable as vectors from $\mathrm{GF}(2)^t$, the relabeling f in the composition $f \circ lab$ acts as a *linear transformation* in the vector space $\mathrm{GF}(2)^t$. Informally, f is applied separately to each label in $lab(v)$ and the outcomes are summed up "modulo 2"; such as for $lab(v) = \{1, 2\}$ and $f(1) = \{1, 3, 4\}$, $f(2) = \{1, 2, 3\}$, we get $f \circ lab(v) = \{2, 4\} = \{1, 3, 4\} \,\triangle\, \{1, 2, 3\}$.

Let $\odot$ be a nullary operator creating a single new graph vertex of label $\{1\}$. For t-relabelings $f_1, f_2, g : L_t \to 2^{L_t}$ let $\otimes[g \mid f_1, f_2]$ be a binary operator, called *t-labeling join* (as bilinear product of [7]), over pairs of t-labeled graphs $\bar{G}_1 = (G_1, lab^1)$ and $\bar{G}_2 = (G_2, lab^2)$ defined as follows:

$$(G_1, lab^1) \; \otimes[g \mid f_1, f_2] \; (G_2, lab^2) \;\; = \;\; (H, lab)$$

where the graph H is constructed from the disjoint union $G_1 \dot{\cup} G_2$ by adding all edges uw, $u \in V(G_1)$ and $w \in V(G_2)$ such that $|lab^1(u) \cap g \circ lab^2(w)|$ is odd, and with the new labeling $lab(v) = f_i \circ lab^i(v)$ for $v \in V(G_i)$, $i = 1, 2$.

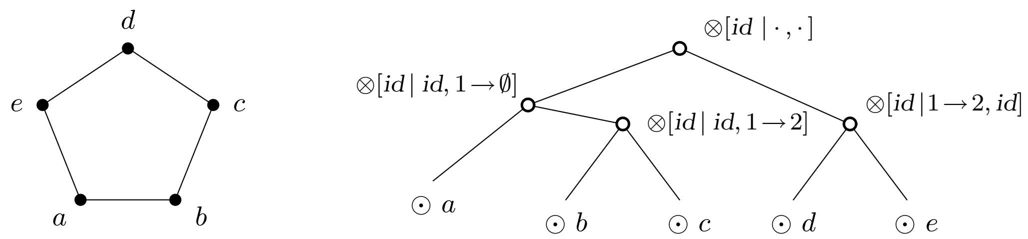

Fig. 2. An example of a labeling parse tree which generates a 2-labeled cycle C_5, with symbolic operators at the nodes (*id* denotes the relabeling preserving all labels).

A *t-labeling parse tree* T, see also [13, Definition 6.11], is a finite rooted ordered subcubic tree (with the root degree at most 2) such that

- all leaves of T contain the $\odot$ symbol, and
- each internal node of T contains one of the t-labeling join symbols.

A parse tree T then *generates* (parses) the graph G which is obtained by successive leaves-to-root applications of the operators in the nodes of T. See Fig. 2,3 for an illustration.

We make two short notes to this definition. First, the role of relabeling g in $\otimes[g \mid f_1, f_2]$ is unavoidable for Theorem 2 to hold true, but we can sometimes (when needed) avoid it as in Proposition 2 later on. Second, our definition of a parse tree allows a node with just one descendant, and in such a case the $\otimes[g \mid f_1, f_2]$ operator is (naturally) applied to the empty graph on the other side.

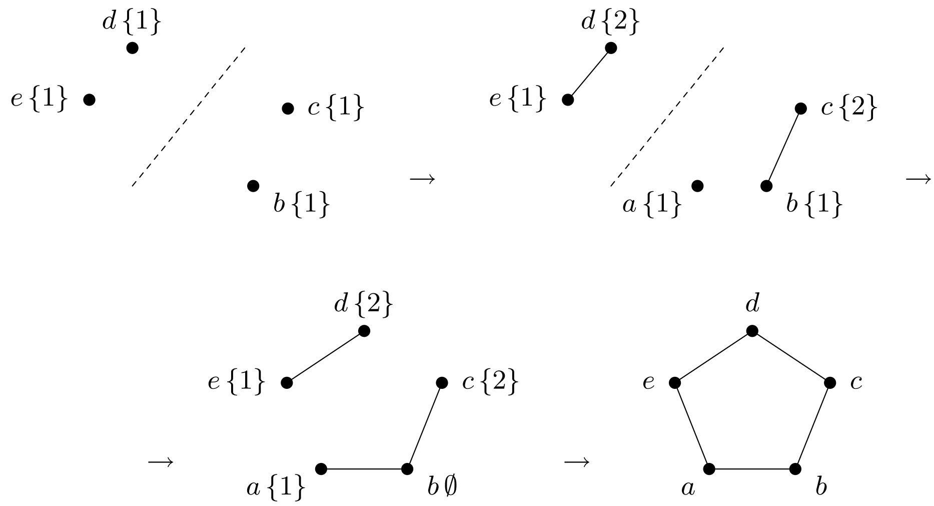

Fig. 3. "Bottom-up" generation of C_5 by the parse tree from Fig. 2.

¿From the prior work of Courcelle and Kanté we get a crucial statement:

Theorem 2 (Rank-width parsing theorem [7]). *A graph G has rank-width at most t if and only if (some labeling of) G can be generated by a t-labeling parse tree. Furthermore, a width-t rank-decomposition of G can be transformed into a t-labeling parse tree on $\Theta(|V(G)|)$ nodes in time $O(t^2 \cdot |V(G)|^2)$.*

This statement is equivalent to [7, Theorem 3.4] which reads: *G has rank-width at most t if and only if G is the value of a term over C_t and R_t*, where C_t is the set of t-labeled singletons and R_t is the set of bilinear product forms of rank at most t. A bilinear product $\otimes_{f,g,h}$ of [7] is straighforwardly equivalent to our $\otimes[f \mid g, h]$, and a t-labeled singleton vertex $f(\odot)$ can be emulated with two nodes—the $\odot$ singleton symbol with a $\otimes[\emptyset \mid f, \emptyset]$ "relabeling" parent.

Finally, time complexity bound $O(t^2 \cdot |V(G)|^2)$ for turning a rank-decomposition into a labeling parse tree is not explicit in [7], but it easily follows from an independent self-contained proof of Theorem 2 in the first author's Master thesis [13, Chapter 6].

Remark 1. We suggest that the "nearly linear" term $|V(G)|^2$ in the time complexity of Theorem 2 can be improved to linear $|E(G)|$ if one carefully reconsiders all the technical details, but that would not be profitable in our context in which we use Theorem 2 together with Theorem 1 to construct an optimal labeling parse tree of a given graph G in parameterized $O(|V(G)|^3)$ time.

3 Regularity Theorem for Rank-width

The substantial contribution of our paper lies in developing further new mathematical formalisms for easier handling of certain algorithmic problems on graphs of bounded rank-width. Our ideas are closely tied with the classical Myhill–Nerode regularity tool in automata theory—that is possible since our parse trees, for every fixed t, have nodes with symbols of a finite alphabet and hence can be used as an input for finite tree automata. Such thinking is not quite new in theory—it has been inspired by analogous machinery successfully used in [1] or [10, Chapter 6] (for graphs of bounded tree-width) and in [16] (for matroids of bounded branch-width) before. The case of rank-width, however, brings some new obstacles.

We make two simple technical remarks. First, we may need to interchange the operands of a t-labeling join which itself is not commutative. Since a t-relabeling g is a linear transformation in $GF(2)^t$, this g is determined by a square binary matrix (cf. also the bilinear product of [7]), and hence we can define a t-relabeling g^T as the matrix-transpose of the linear mapping g.

Proposition 1. *Let $\bar{G}_1, \bar{G}_2$ be t-labeled graphs and $g : L_t \to 2^{L_t}$ be a t-relabeling. If a relabeling g^T is given by the transposed linear mapping of g, then*

$$\bar{G}_1 \otimes[g \mid f_1, f_2] \; \bar{G}_2 = \bar{G}_2 \otimes[g^T \mid f_2, f_1] \; \bar{G}_1 \,.$$

Second, we shortly write $\otimes[g]$ for $\otimes[g \mid \emptyset, \emptyset]$ where $\emptyset$ stands for the relabeling $L_t \to \{\emptyset\}$ "forgetting" all vertex labels. The role of g in $\otimes[g]$ is rather technical after all, as the next immediate claim specifies (where id preserves all labels).

Proposition 2. *Let $\bar{G}_1, \bar{G}_2$ be t-labeled graphs generated by labeling parse trees T_1, T_2, and g be a t-relabeling. Then there is a tree T_2^g parsing a t-labeled graph $\bar{G}_2^g$ (actually unlabeled-equal to $\bar{G}_2$) such that*

$$\bar{G}_1 \otimes[g] \; \bar{G}_2 = \bar{G}_1 \otimes[id] \; \bar{G}_2^g \,.$$

The canonical equivalence. Let Π_t denote the finite set (alphabet) of all the t-labeling join symbols and $\odot$, and let subsequently $P_t \subseteq \Pi_t^{**}$ be the class (language) of all valid t-labeling parse trees. If $\mathcal{R}_t$ denotes the class of all unlabeled graphs of rank-width at most t and $\overline{\mathcal{R}}_t$ is the class of all t-labeled graphs parsed by the trees from P_t, then (Theorem 2) $G \in \mathcal{R}_t$ if and only if $\bar{G} \in \overline{\mathcal{R}}_t$ for some t-labeling $\bar{G}$ of G.

Let $\mathcal{D}$ be any class of graphs, and $\mathcal{D}_t = \mathcal{D} \cap \mathcal{R}_t$. In analogy to the classical theory of regular languages we define a *canonical equivalence* of $\mathcal{D}_t$, denoted by $\approx_{\mathcal{D},t}$, as follows: $\bar{G}_1 \approx_{\mathcal{D},t} \bar{G}_2$ for any $\bar{G}_1, \bar{G}_2 \in \overline{\mathcal{R}}_t$ if and only if, for all $\bar{H} \in \overline{\mathcal{R}}_t$,

$$\bar{G}_1 \otimes [id]\ \bar{H} \in \mathcal{D}_t \iff \bar{G}_2 \otimes [id]\ \bar{H} \in \mathcal{D}_t\,.$$

In informal words, the classes of $\approx_{\mathcal{D},t}$ "capture" all information we need to know about a t-labeled subgraph $\bar{G} \in \overline{\mathcal{R}}_t$ to decide membership in $\mathcal{D}$ further on in our parse tree processing (we do not need to consider arbitrary g of $\otimes[g]$ in this canonical equivalence thanks to Proposition 2).

This informal finding can be formalized as follows (cf. [13, Chapter 7]):

Theorem 3 (Rank-width regularity theorem). *Let $t \geq 1$, $\mathcal{D}$ be a graph class, and $\mathcal{D}_t = \mathcal{D} \cap \mathcal{R}_t$. The collection of all those t-labeling parse trees which generate the members of $\mathcal{D}_t$ is accepted by a finite tree automaton if, and only if, the canonical equivalence $\approx_{\mathcal{D},t}$ of $\mathcal{D}_t$ over $\overline{\mathcal{R}}_t$ is of finite index.*

Proof. Our starting point is the classical Myhill–Nerode theorem for tree automata. Let Σ^{**} denote the set of all rooted binary trees over a finite alphabet Σ. For a language $\lambda \subseteq \Sigma^{**}$ we can define a congruence $\sim_\lambda$ such that $T_1 \sim_\lambda T_2$ for $T_1, T_2 \in \Sigma^{**}$ if, and only if, $T_1 \diamond_x U \in \lambda \iff T_2 \diamond_x U \in \lambda$ where U runs over all special rooted binary trees over Σ with one distinguished leaf node x, and $T_i \diamond_x U$ results from U by replacing the leaf x with the subtree T_i. Then λ is accepted by a finite tree automaton if and only if $\sim_\lambda$ has finite index.

In our case $\Sigma = \Pi_t$, and λ are the labeling parse trees of the members of $\mathcal{D}_t$. So, to prove our theorem it is enough to show that $\approx_{\mathcal{D},t}$ has infinite index if and only if $\sim_\lambda$ has infinite index.

Suppose the former holds, i.e. there are infinitely many $\bar{G}_k \in \overline{\mathcal{R}}_t$, $k = 1, 2, \ldots$, such that for all indices $i \neq j$ there exists $\bar{H}_{i,j} \in \overline{\mathcal{R}}_t$ for which $\bar{G}_i \otimes[id]\ \bar{H}_{i,j} \in \mathcal{D}_t$ but $\bar{G}_j \otimes[id]\ \bar{H}_{i,j} \notin \mathcal{D}_t$, or vice versa. Let S_k be a labeling parse tree of $\bar{G}_k$, and $Q_{i,j}$ that of $\bar{H}_{i,j}$. We define a new parse tree $U_{i,j}$ such that the root operator is $\otimes[id \mid \emptyset, \emptyset]$, its left son is the distinguished leaf x, and its right subtree is $Q_{i,j}$. Hence the special trees $U_{i,j}$ witness that all the parse trees S_k, $k = 1, 2, \ldots$ belong to distinct classes of $\sim_\lambda$.

Conversely, suppose that the latter holds. So there are infinitely many trees $S_k \in \Pi_t^{**}$, $k = 1, 2, \ldots$, such that for each pair of indices $i \neq j$ there exists $U_{i,j}$ as above for which $S_i \diamond_x U_{i,j} \in \lambda$ but $S_j \diamond_x U_{i,j} \notin \lambda$, or vice versa. We may assume without loss of generality that $S_k \in P_t$ are valid labeling parse trees for all k. Let $\bar{G}_k$ be the graphs parsed by S_k. Using technical Lemma 1 and Proposition 2, we deduce that there exist graphs $\bar{H}_{i,j}$ such that

- the graph parsed by $S_i \diamond_x U_{i,j}$ is equal up to labeling to $\bar{G}_i \otimes[id]\ \bar{H}_{i,j} \in \mathcal{D}_t$,
- and the graph parsed by $S_j \diamond_x U_{i,j}$ equals up to labeling $\bar{G}_j \otimes[id]\ \bar{H}_{i,j} \notin \mathcal{D}_t$.

This assertion certifies that the graphs $\bar{G}_k$ indeed belong to distinct classes of our canonical equivalence $\approx_{\mathcal{D},t}$. ∎

Lemma 1. *Let T be a labeling parse tree generating an unlabeled graph G, let v be a node of T, and let T_v denote the subtree of T rooted at v. Then there exist a labeling parse tree W and a t-relabeling ℓ such that $G = \bar{G}_v \otimes[\ell]\ \bar{H}$, where $\bar{G}_v$ is the t-labeled graph parsed by T_v and $\bar{H}$ is the t-labeled graph parsed by W. Furthermore, the tree W does not depend on T_v.*

Proof. First of all, by switching the subtrees of suitable nodes of T, as in Proposition 1, we can assume that the node v is on the leftmost branch of T. Then we continue by induction on the distance between v and the root r of T. If the distance is 1, we are done: we take W the right subtree of r, and ℓ from the join operator of r. If not, then we will reduce the distance from the root to v by 1 by using the right tree rotation (at r) as in Fig. 4.

Indeed, the parse tree T' obtained from T by the rotation of Fig. 4 generates the same unlabeled graph $G' = G$ if we choose:

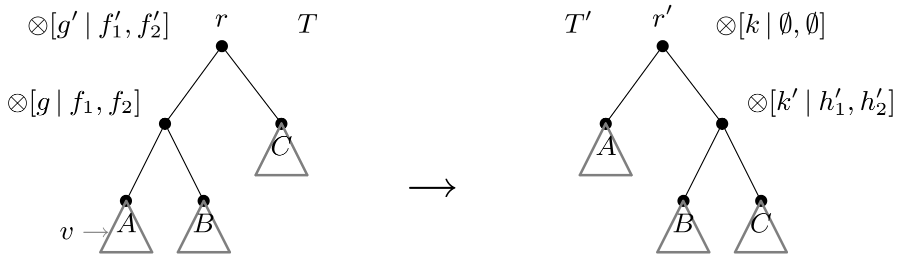

Fig. 4.

$k = id$, $k' = f_2^T \circ g'$, $h'_1 = g$, and $h'_2 = f_1^T \circ g'$, where f_i^T, $i = 1, 2$, are given by the transposed linear mapping of f_i. We leave the straightforward algebraic verification of this fact to the reader. (Notice, however, that the vertex labeling of the resulting graph G' generally cannot be preserved the same as that of G, and so such a construction can be used only at the parse-tree root.)

The proof is thus finished by induction. Since, moreover, we have not used any information about the subtree T_v in the construction, the resulting right subtree W of the root will not depend on T_v. ∎

Remark 2. Notice that the arguments used in our proof of Theorem 3 *do not* straightforwardly translate from rank-width (and labeling parse trees) to clique-width (and its k-expressions). Quite the opposite, the "only if" direction of this theorem seems not at all provable in the above way since one cannot freely choose the "root" of a k-expression. We consider that another small reason to favor rank-width over clique-width in CS applications.

3-**colourability example.** We demonstrate the use of Theorem 3 on graph 3-colourability which is a well-known NP-complete problem. Let $\mathcal{C}$ denote the class of all simple 3-colourable graphs. To construct a tree automaton accepting the labeling parse trees of the members of $\mathcal{C} \cap \mathcal{R}_t$, it is enough to identify the classes of the canonical equivalence $\approx_{\mathcal{C},t}$. We actually give below finitely many classes $\mathcal{X} = \{X_0, X_1, X_2, \dots\}$ of a refinement of $\approx_{\mathcal{C},t}$, see Proposition 3.

Assume a t-labeled graph $\bar{G} = (G, lab)$ with a proper 3-colouring χ. Let, for $i = 1, 2, 3$, $\gamma_i(\bar{G}, \chi) = \{lab(u) : u \in V(G) \wedge \chi(u) = i\}$. The set $\gamma_i(\bar{G}, \chi)$, as a set of vectors in $GF(2)^t$, generates a subspace $\langle \gamma_i(\bar{G}, \chi) \rangle \subseteq GF(2)^t$, and we define $\Gamma(\bar{G}, \chi) = \big(\langle \gamma_1(\bar{G}, \chi) \rangle, \langle \gamma_2(\bar{G}, \chi) \rangle, \langle \gamma_3(\bar{G}, \chi) \rangle\big)$. Finally, $\mathcal{X}$ is defined by

- $X_0 = \{\bar{G} : G$ is not 3-colourable$\}$, and
- $X_1, X_2, \dots, X_{i(t)}$ are the equivalence classes of $\sim$, where over t-labeled graphs $\bar{G}_1 \sim \bar{G}_2$ if and only if it holds $\{\Gamma(\bar{G}_1, \chi) :\ \chi$ is a proper 3-colouring of $G_1\} = \{\Gamma(\bar{G}_2, \chi) :\ \chi$ is a proper 3-colouring of $G_2\}$.

Now, the fact that the 3-colourability problem is efficiently solvable (even by a tree automaton) on graphs of bounded rank-width, follows from Theorems 1 and 3 and the next rather simple statement. We refer also to further Theorem 12 for a more general result.

Proposition 3. *If $\bar{G}_1$ and $\bar{G}_2$ belong to the same class of $\mathcal{X}$, then $\bar{G}_1 \approx_{\mathcal{C},t} \bar{G}_2$.*

Proof. Assume a contradiction – that, up to symmetry, there exists a t-labeled graph $\bar{H}$ such that $\bar{G}_1 \otimes [id]\ \bar{H}$ is 3-colourable while $\bar{G}_2 \otimes [id]\ \bar{H}$ is not, yet $\bar{G}_1 = (G_1, lab^1)$ and $\bar{G}_2 = (G_2, lab^2)$ belong to the same class of $\mathcal{X}$. Choosing any valid 3-colouring χ of $\bar{G}_1 \otimes [id]\ \bar{H}$; we denote by χ_1 and χ_H the restrictions of χ to the vertices of $\bar{G}_1$ and $\bar{H}$, respectively.

Since $\bar{G}_1$ and $\bar{G}_2$ both belong to the same class of $\mathcal{X}$, there must exist a 3-colouring χ_2 of $\bar{G}_2$ such that $\Gamma(\bar{G}_1, \chi_1) = \Gamma(\bar{G}_2, \chi_2)$. Now we consider a 3-colouring γ' of $\bar{G}_2 \otimes [id]\ \bar{H}$ with the vertices of $\bar{G}_2$ coloured by χ_2 and the vertices of $\bar{H}$ coloured by χ_H. What remains to argue is that γ' is a proper 3-colouring, disproving the initial assumption.

Consider any colour $j \in \{1, 2, 3\}$ and a labeling $Z \subseteq \{1 \dots t\}$ of any j-coloured vertex u in $\bar{H}$. There is no labeling $Z' \in \gamma_j(\bar{G}_1, \chi_1)$ with $|Z \cap Z'|$ odd since, otherwise, the j-coloured vertex w of $\bar{G}_1$, $lab^1(w) = Z'$, would be adjacent to j-coloured u in $\bar{G}_1 \otimes [id]\ \bar{H}$. And since $\gamma_j(\bar{G}_1, \chi_1)$ generates the same subspace of $GF(2)^t$ as $\gamma_j(\bar{G}_2, \chi_2)$, it is a matter of elementary linear algebra to verify that there is no $X \in \gamma_j(\bar{G}_2, \chi_2)$ with odd $|Z \cap X|$ either:

(4) Let Ψ be a subspace of $GF(2)^t$ generated by $y_1, \dots, y_k \in GF(2)^t$, and $z \in GF(2)^t$. If $y_i \cdot z = 0$ for $i = 1, \dots, k$, then for all $x = \alpha_1 y_1 + \cdots + \alpha_k y_k \in \Psi$ we have $x \cdot z = \alpha_1 y_1 \cdot z + \cdots + \alpha_k y_k \cdot z = 0$.

No edge between two j-coloured vertices has thus been created in $\bar{G}_2 \otimes [id]\ \bar{H}$. ∎

More involved and powerful applications of our Theorem 3 can be found in the coming sections, especially in Section 4.

4 From Regularity to MSO Properties

Monadic second-order (MSO in short) logic is a language particularly suited for description of problems on "tree-like" decompositions of graphs. Already about 20 years ago it was shown that all MSO definable properties of incidence graphs can be solved in linear time if a tree-decomposition of bounded width is given on the input [2, 6]. Analogous statement has been shown by Courcelle, Makowsky, and Rotics [8] for MSO definable properties of adjacency graphs if a k-expression (cf. clique-width) of bounded k is given on the input, and this readily extends to graphs with a given rank-decomposition of bounded width, e.g. [7, Corollary 3.3].

¿From a logic point of view, we consider an adjacency graph as a relational structure on the ground set V, with one binary predicate $edge(u, v)$. When the language of MSO logic is applied to such a graph adjacency structure, one gets a descriptional language over graphs commonly abbreviated as MS_1. For an illustration we show an MS_1 expression of the 3-colourability property of a graph:

$$\exists V_1, V_2, V_3 \Big[\quad \forall v\, (v \in V_1 \vee v \in V_2 \vee v \in V_3) \;\wedge \\ \bigwedge_{i=1,2,3} \forall v, w\, (v \notin V_i \vee w \notin V_i \vee \neg\, edge(v, w)) \Big]$$

It is also common to consider the "counting" version of MSO logic which moreover has predicates $mod_{p,q}(X)$ stating that $|X| \bmod p = q$.

To avoid possible confusion we remark that the previously mentioned stronger MSO language of incidence graphs, abbreviated as MS_2, allows to quantify also over graph edges and their sets. There are MS_2 expressible graph properties, e.g. Hamiltonicity, which are not expressible in MS_1, whilst MS_2 properties cannot be (in general) efficiently handled on graphs of bounded rank-width.

In this section we would like to show that the "MS_1"-statement of Courcelle, Makowsky, and Rotics [8] can also be set up in the scope of our Rank-width regularity Theorem 3. Briefly saying, we consider the class $\mathcal{F}$ of graphs described by an MS_1 sentence ϕ, and show by structural induction on ϕ that the canonical equivalence $\approx_{\mathcal{F},t}$ has finite index. The latter actually needs an extension of $\approx_{\mathcal{F},t}$ to an equivalence $\approx^{\sigma}_{\phi,t}$ (see below) allowing for formulas ϕ with free variables.

This new view shall not only be an elementary combinatorial alternative to the proof [8] which used MSO interpretation (transduction) of the graphs generated by k-expressions into labeled binary trees, but also leads to new Theorem 5 which could be of independent interest (see Remark 3).

Extended canonical equivalence of $\mathbf{MS}_1$ formulas. We propose an extension analogous to the previous works [1, 16], but new in the context of rank-width.

Let $Free(\phi) = Fr(\phi) \cup FR(\phi)$ be the partition of the free variables into those $Fr = Fr(\phi)$ for vertices and those $FR = FR(\phi)$ for vertex sets. We define a *partial equipment signature* of ϕ as a triple $\sigma = (Fr, FR, q)$ where $q : Fr \to \{0, 1\}$. A t-labeled graph G is *σ-partially equipped* if it has distinguished vertices and vertex sets assigned as interpretations of the free variables in σ. Formally, for each $X \in FR$ there is a distinguished subset $S_X \subseteq V(G)$, and for each $x \in Fr$ such that $q(x) = 0$ there is a distinguished vertex $v_x \in V(G)$. Nothing is assigned to variables $x \in Fr$ such that $q(x) = 1$. For σ we define a *complemented* partial equipment signature $\sigma^- = (Fr, FR, q')$ where $q'(x) = 1 - q(x)$ for all $x \in Fr$.

See that if $\bar{H}_1$ is σ-partially equipped and $\bar{H}_2$ is σ^--partially equipped, then $H = \bar{H}_1 \otimes[g]\ \bar{H}_2$ has a full and consistent interpretation for all the free variables of ϕ (hence this H is a logic model of ϕ). So, we can define equivalence $\approx^{\sigma}_{\phi,t}$ over all t-labeled σ-partially equipped graphs as follows: $\bar{G}_1 \approx^{\sigma}_{\phi,t} \bar{G}_2$ if and only if the following

$$(\bar{G}_1 \otimes[id]\ \bar{H}) \models \phi \iff (\bar{G}_2 \otimes[id]\ \bar{H}) \models \phi$$

holds for all t-labeled σ^--partially equipped graphs $\bar{H}$.

Here we have extended the meaning of $\approx^{\sigma}_{\phi,t}$ in two directions. First, by allowing free variables in ϕ we enlarge the studied universe to partially equipped graphs. Second, the universe is further enlarged by allowing all t-labeled underlying graphs – not only those from $\overline{\mathcal{R}}_t$. Yet we can prove:

Theorem 5. *Let $t \geq 1$ be fixed. Suppose ϕ is a formula in the language MS_1, and σ is a partial equipment signature for ϕ. Then $\approx^{\sigma}_{\phi,t}$ has finite index in the universe of t-labeled σ-partially equipped graphs.*

Proof. We retain the notation introduced above. The induction base is to prove the statement for the atomic formulas in MS_1: $\phi \equiv (v \in W)$, $(v = w)$, $mod_{p,q}(W)$, or $edge(u,v)$. The first three are all rather trivial cases which we skip here, and we focus on the last predicate $edge(u,v)$ (since this one actually "defines" the graph we study).

(6) Suppose $\phi \equiv edge(u,v)$. Then the index of $\approx^{\sigma}_{\phi,t}$ is one if $q(u) = q(v) = 1$, two if $q(u) = q(v) = 0$, and 2^t if $q(u) = 0$ and $q(v) = 1$ or vice versa.

In the first case both vertices u, v with a possible edge uv are interpreted in the right-hand graph $\bar{H}$, and hence no matter what $\bar{G}_1$ or $\bar{G}_2$ are, they become equivalent in $\approx^{\sigma}_{\phi,t}$. In the second case both vertices u, v are interpreted in the left-hand graphs $\bar{G}_i$, and hence there are exactly two classes formed by those graphs having and those not having u adjacent to v. It is the third case which interests us: Recalling the definition of our join operator $\otimes[id]$, we see that all information needed to decide whether some u in the left-hand graph is adjacent to a specific v in the right-hand graph is encoded in the labeling of u, and hence the 2^t possibilities there.

For the inductive step, we consider that a formula ϕ is created from shorter formula(s) in one of the following ways: $\phi \equiv \neg\psi$, $\psi \wedge \eta$, $\exists v\, \psi(v)$, or $\exists W\, \psi(W)$, where $v \in Fr(\psi)$ or $W \in FR(\psi)$ in the latter cases. One may easily express the $\vee$ or $\forall$ symbols using these. The arguments we are going to give in the rest of this proof are not completely novel—they are similar to [1] and nearly a translation of the arguments used in [16, Lemma 6.2] (unfortunately, a simple reference to that is not enough here).

We assume by induction that $\approx^{\pi}_{\psi,t}$ ($\approx^{\rho}_{\eta,t}$) has finite index, where the signature π (ρ) is inherited from σ for ψ (for η, see below the case-by-case details). The first case is quite easy to resolve:

(7) If $\phi \equiv \neg\psi$, then the equivalence $\approx^{\pi}_{\psi,t}$ is the same as $\approx^{\sigma}_{\phi,t}$.

We look at the second, only slightly more involved, case.

(8) Suppose $\phi \equiv \psi \wedge \eta$, and let π, ρ denote the restrictions of signature σ to $Free(\psi)$, $Free(\eta)$, respectively. If $\approx^{\pi}_{\psi,t}$ has index p and $\approx^{\rho}_{\eta,t}$ has index r, then $\approx^{\sigma}_{\phi,t}$ has index at most $p \cdot r$.

Consider an arbitrary pair of t-labeled σ-partially equipped graphs $\bar{G}_1 \not\approx^{\sigma}_{\phi,t} \bar{G}_2$, and an associated σ^--partially equipped graph $\bar{H}$ such that $(\bar{G}_1 \otimes[id]\ \bar{H}) \models \phi$ but $(\bar{G}_2 \otimes[id]\ \bar{H}) \not\models \phi$. Then it has to be $(\bar{G}_1 \otimes[id]\ \bar{H}) \models \psi$ (or $\models \eta$) but $(\bar{G}_2 \otimes[id]\ \bar{H}) \not\models \psi$ (or $\not\models \eta$, resp.). Hence it immediately holds that $\bar{G}_1 \not\approx^{\pi}_{\psi,t} \bar{G}_2$ or $\bar{G}_1 \not\approx^{\rho}_{\eta,t} \bar{G}_2$ with the restricted equipments, and so the equivalence classes of $\approx^{\sigma}_{\phi,t}$ are suitable unions of the classes of the "intersection" $\approx^{\pi}_{\psi,t} \cap \approx^{\rho}_{\eta,t}$.

The third case of $\exists v\, \psi(v)$ is technically more complicated, and so we first deal with the similar but easier fourth case of $\exists W\, \psi(W)$.

(9) Suppose $\phi \equiv \exists W\, \psi(W)$, and let the signature $\pi = (Fr, FR \cup \{W\}, q)$. If $\approx^{\pi}_{\psi,t}$ has index p, then $\approx^{\sigma}_{\phi,t}$ has index at most $2^p - 1$.

Again consider an arbitrary pair of t-labeled σ-partially equipped graphs $\bar{G}_1 \not\approx^{\sigma}_{\phi,t} \bar{G}_2$, and $\bar{H}$ such that $(\bar{G}_1 \otimes[id]\ \bar{H}) \models \phi$ but $(\bar{G}_2 \otimes[id]\ \bar{H}) \not\models \phi$. We shortly write $\bar{G}[W = S]$ for the π-partially equipped graph obtained from σ-partially equipped $\bar{G}$ by interpreting the variable W as $S \subseteq V(\bar{G})$. Then our assumption about $\bar{G}_1, \bar{G}_2$ means there exist $S_W \subseteq V(\bar{G}_1)$ and $S'_W \subseteq V(\bar{H})$ such that $\big(\bar{G}_1[W = S_W] \otimes[id]\ \bar{H}[W = S'_W]\big) \models \psi$, whilst $\big(\bar{G}_2[W = T_W] \otimes[id]\ \bar{H}[W = S'_W]\big) \not\models \psi$ for all $T_W \subseteq V(\bar{G}_2)$. Hence $\bar{G}_1[W = S_W] \not\approx^{\pi}_{\psi,t} \bar{G}_2[W = T_W]$.

We now, in search for a contradiction, look at the problem from the other side. Let the equivalence classes of $\approx^{\pi}_{\psi,t}$ over t-labeled π-partially equipped graphs be $\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^p$. For a σ-partially equipped graph $\bar{G}$ we define a nonempty set $Ix(\bar{G}) \subseteq \{1, 2, \ldots, p\}$ as follows: $i \in Ix(\bar{G})$ if and only if $\bar{G}[W = S] \in \mathcal{C}^i$ for some $S \subseteq V(\bar{G})$. If there were 2^p pairwise incomparable σ-partially equipped graphs in the relation $\approx^{\sigma}_{\phi,t}$, then some two of them, say $\bar{G}_1 \not\approx^{\sigma}_{\phi,t} \bar{G}_2$, would receive $Ix(\bar{G}_1) = Ix(\bar{G}_2)$ by the pigeon-hole principle. However, from the argument of the previous paragraph — $\bar{G}_1[W = S_W] \not\approx^{\pi}_{\psi,t} \bar{G}_2[W = T_W]$ for some $S_W \subseteq V(\bar{G}_1)$ and all $T_W \subseteq V(\bar{G}_2)$, we conclude that $j \in Ix(\bar{G}_1) \setminus Ix(\bar{G}_2)$ where j is such that $\bar{G}_1[W = S_W] \in \mathcal{C}^j$. This contradiction proves (9).

(10) Suppose $\phi \equiv \exists v\, \psi(v)$, and let signatures $\pi = (Fr \cup \{v\}, FR, q_1)$ and $\rho = (Fr \cup \{v\}, FR, q_2)$ where $q_1(v) = 0$ and $q_2(v) = 1$. If $\approx^{\pi}_{\psi,t}$ has index p and $\approx^{\rho}_{\psi,t}$ has index r, then $\approx^{\sigma}_{\phi,t}$ has index at most $2^p \cdot r + 1 - r$.

Notice that a ρ-partial equipment of $\bar{G}$ does not interpret the variable v in $V(G)$, and so σ-partially equipped graph $\bar{G}$ may be viewed also as ρ-partially equipped. Take an arbitrary pair of nonempty t-labeled σ-partially equipped graphs $\bar{G}_1 \not\approx^{\sigma}_{\phi,t} \bar{G}_2$, and $\bar{H}$ such that $(\bar{G}_1 \otimes[id]\ \bar{H}) \models \phi$ but $(\bar{G}_2 \otimes[id]\ \bar{H}) \not\models \phi$. Let $x_v \in V(\bar{G}_1) \cup V(\bar{H})$ be an interpretation of the variable v that satisfies ψ over $\bar{G}_1 \otimes[id]\ \bar{H}$. In particular, ψ is false over $\bar{G}_2 \otimes[id]\ \bar{H}$ here. If $x_v \in V(\bar{H})$, then immediately $\bar{G}_1 \not\approx^{\rho}_{\psi,t} \bar{G}_2$. Otherwise, $x_v \in V(\bar{G}_1)$ and we are in a situation analogous to the first paragraph of (9): $\big(\bar{G}_1[v = x_v] \otimes[id]\ \bar{H}\big) \models \psi$, whilst $\big(\bar{G}_2[v = y_v] \otimes[id]\ \bar{H}\big) \not\models \psi$ for all $y_v \in V(\bar{G}_2)$.

Again, in search for a contradiction, we look at the problem from the other side. If there are $2^p r + 2 - r$ pairwise incomparable σ-partially equipped graphs with respect to $\approx^{\sigma}_{\phi,t}$, then at least $2^p r + 1 - r = (2^p - 1)r + 1$ of those graphs are nonempty, and out of them at least 2^p belong to the same equivalence class of $\approx^{\rho}_{\psi,t}$. Let their set be denoted by $\mathcal{G}$ (Hence for each pair in $\mathcal{G}$, the latter conclusion of the previous paragraph applies). Considering the equivalence classes $\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^p$ of $\approx^{\pi}_{\psi,t}$, we again (as in 9) define a nonempty set $Ix(\bar{G}) \subseteq \{1, 2, \ldots, p\}$, for σ-partially equipped $\bar{G}$, by $i \in Ix(\bar{G})$ if and only if $\bar{G}[v = y] \in \mathcal{C}^i$ for some $y \in V(\bar{G})$. Then some pair, say $\bar{G}_1, \bar{G}_2 \in \mathcal{G}$, must satisfy $Ix(\bar{G}_1) = Ix(\bar{G}_2)$ by the pigeon-hole principle. However, that analogously contradicts the latter conclusion of the previous paragraph.

This contradiction proves (10), and thus the whole theorem. ∎

Having a closed MS_1 formula ϕ, the associated equipment signature is always empty and hence we, in conjunction with Theorem 3, easily conclude:

Corollary 1 (cf. [8, 7]). *Let $t \geq 1$. If $\mathcal{F}$ is a graph class definable in the MS_1 language, then the language of all those t-labeling parse trees which generate the members of $\mathcal{F} \cap \mathcal{R}_t$ is accepted by a finite tree automaton.*

Remark 3. Corollary 1 straightforwardly generalizes also to classes $\mathcal{F}_\phi$ defined by non-closed MS_1 formulas ϕ if we extend the universe to *equipped* t-labeling parse trees—additional labels are used (in the leaves) to encode a specific interpretation of the free variables of ϕ in these parse trees.

Solving optimization problems. Unfortunately, direct algorithmic applicability of the "MS_1" theorem (Corollary 1) is limited to pure decision problems (like 3-colourability), but many practical problems are formulated as optimization ones. And the usual way of transforming optimization problems into decision ones does not work here since MS_1 language cannot handle arbitrary numbers.

Nevertheless, there is a known solution. Arnborg, Lagergren, and Seese [2] (while studying graphs of bounded tree-width), and later Courcelle, Makowsky, and Rotics [8] (for graphs of bounded clique-width), specifically extended the expressive power of MSO logic to define so-called *Lin*EMSO optimization problems, and consequently shown existence of efficient (parameterized) algorithms for such problems in the respective cases. Briefly saying, *Lin*EMSO problems allow, in addition to ordinary MSO expressions, to compare between and optimize over linear evaluational terms.

We can achieve an analogous solution in our framework directly using Theorem 5. The basic idea is that, in a dynamic processing of the input labeling parse tree, we can keep track only of suitable "optimal" representatives of all possible interpretations of the free variables in ϕ, per each class of the extended canonical equivalence $\approx^{\sigma}_{\phi,t}$. We illustrate this idea with the following example.

Consider any MS_1 formula $\psi(X_1, \ldots, X_p)$ and an optimization problem, say,

$$\textbf{(11)} \qquad \max{}_{X_1,\ldots X_p \subseteq V(G):\, \psi(X_1,\ldots,X_p)} \; f(X_1, \ldots X_p)$$

where f is a linear evaluational function on the elements of $X_1, \ldots X_p$. Such as,

$\psi = \iota(X) \;\equiv\; \forall v, w \left(v \notin X \vee w \notin X \vee \neg\, edge(v, w)\right)$ and $f(X) = |X|$

describes the maximum independent set problem, or

$\psi = \delta(X) \;\equiv\; \forall v \exists w \left[v \in X \vee \left(w \in X \wedge edge(v, w)\right)\right]$ and $f(X) = -|X|$, is the minimum dominating set problem. Further examples like minimum independent or connected dominating set problems are easily possible.

Now we show how Theorem 5 can be employed in solving problems like (11) via dynamic programming. Let G be an input graph of rank-width t, and T its t-labeling parse tree. We denote by T_x the subtree below a node x of T, and by $\bar{G}_x$ the t-labeled subgraph of G parsed by T_x.

For any $W_1, \ldots, W_p \subseteq V(G_x)$, the σ-partially equipped graph $\bar{G}_x$ with interpretation $X_i = W_i$, $i = 1, \ldots, p$ falls into one of the (finitely many) ℓ classes of $\approx^{\sigma}_{\psi,t}$ (Theorem 5). A dynamic algorithm for solving (11) has to remember just one representative interpretation $(W^j_1, \ldots, W^j_p)$ achieving maximum $f(X_1, \ldots, X_p)$ over the j-th class of $\approx^{\sigma}_{\psi,t}$, for $j = 1, 2, \ldots, \ell$. Thanks to linearity of the objective function f, and with knowledge of the associated tree automaton (Remark 3), this information can easily be processed from leaves of T to the root in total linear time (t fixed).

5 Concrete Algorithmic Applications

As already mentioned in the introduction, the driving force of our research is to provide a framework for easier design of efficient *parameterized* algorithms running on a bounded-width rank-decomposition of a graph. The theory of parameterized complexity [10] defines a problem to be *fixed parameter tractable* with respect to an integer parameter k if it is solvable in time $O(f(k) \cdot n^c)$ where c is a constant and f is any function. The results of Theorem 1, Proposition 3 or Corollary 1 fall into this framework.

For practical applications it is good to have a "small" function f in the expression $O(f(k) \cdot n^c)$, while the previous universal Theorem 5 provides $f(k)$ as a tower of exponents generally growing with quantifier alternation in the formula, cf. (9) and (10). Obviously, we can hardly expect f to be polynomial for NP-complete problems, but say, $f(k)$ of order $2^{poly(k)}$ ("single-exponential") with reasonable coefficients can lead to practically usable algorithms when k is not big. In our context, $k = t$ is the rank-width of an input graph, and the desire is to find FPT algorithms for (some) hard problems with, at the best, a single-exponential dependency of running time on t.

This particular question has been, perhaps, the first time explicitly asked by Bui-Xuan, Telle and Vatshelle in a [yet unpublished / unavailable, 2008] manuscript, in which they provide (in a setting equivalent to prior [7]) two new algorithms for the independent and dominating set problems which run in time roughly $O(2^{\Theta(t^2)}n)$ for graphs with rank-decompositions of width at most t.

We remark that it is likely not possible to obtain an FPT algorithm for a hard problem with a single-exponential dependency on t using clique-width design techniques since [5] the clique-width parameter can reach up to $\Theta(2^{t/2})$, and so one has to work directly with a rank-decomposition. Although the FPT algorithm for a rank-decomposition in Theorem 1 has an unspecified dependency on t (at least double-exponential), in practical situations where graphs of bounded rank-width naturally occur one can hope to obtain such a decomposition for (almost) free from the specific context. Thus algorithms which are single-exponential in t are of great interest independently of Theorem 1.

Graph colourability. The c-colourability problem—whether a graph has a proper colouring using colours $1, 2, \ldots, c$—has already been mentioned for $c = 3$ and graphs of rank-width at most t in Section 3 and in [13]. The results there can be easily generalized to any fixed $c \geq 3$. However, unlike in Section 3 where we have been satisfied with a conclusion that the c-colourability problem is solvable by a tree automaton for each fixed t, here we pay close attention to running-time dependency on the parameters t and c. To actually prove single-exponential dependency on these parameters with small constants, we employ some techniques of Section 4 and dynamic programming here.

Theorem 12. *Let $c \geq 3$ and $t \geq 1$. Assume that an input graph G is given in the form of a t-labeling parse tree T. Then the c-colourability problem of G can be solved in time*

$$O\left(c\,t^3 \cdot 2^{c\,t(t+1)/2} \cdot |V(G)|\right) .$$

Proof. Let a formula $\eta_c(X_1, \ldots, X_c)$ express the claim that independent sets $X_1, \ldots, X_c$ cover all the vertices of a graph H (η_c is an MS_1 formula, but we do not directly use this fact here). In other words, $\eta_c\big(\chi^{-1}(1),\dots,\chi^{-1}(c)\big)$ certifies that $\chi : V(H) \to \{1,\dots,c\}$ is a proper c-colouring of H. We now consider a σ-partially equipped t-labeled graph $\bar{H}$ where σ equips the set variables $X_1,\dots,X_c$ in $V(H)$. For an interpretation $X_i = W_i \subseteq V(H)$, $i = 1,\dots,c$ we (analogously to the 3-colourability example in Section 3) denote by $\langle\gamma(\bar{H},W_i)\rangle$ the subspace of $GF(2)^t$ generated by the labelings of the vertices of W_i in $\bar{H}$. Then we set $\mathit{\Gamma}_c(\bar{H}) = \big(\langle\gamma(\bar{H},W_1)\,\rangle,\dots,\langle\gamma(\bar{H},W_c)\,\rangle\big)$.

By repeating the arguments of the proof of Proposition 3 we obtain that any two t-labeled σ-partially equipped graphs $\bar{H}_1,\bar{H}_2$ satisfy $\bar{H}_1 \approx^{\sigma}_{\eta_c,t} \bar{H}_2$ if $\mathit{\Gamma}_c(\bar{H}_1) = \mathit{\Gamma}_c(\bar{H}_2)$. Hence a dynamic algorithm deciding c-colourability of the input graph G has to remember just a set M of those c-tuples $\mathit{\Gamma}_c(\bar{H})$ of subspaces for which there exist interpretations $X_i = W_i \subseteq V(H)$, $i = 1,\dots,c$ such that $\eta_c(W_1,\dots,W_c)$ holds true in a particular position $\bar{H}$ of the parse tree T.

Subsequently, the number $R(c,t)$ of distinct $\mathit{\Gamma}_c(\bar{H})$ can be bounded using the following recurrence [15] for the total number $S(t)$ of subspaces of $GF(2)^t$: $S(t+1) = 2S(t) + (2^t-1)S(t-1)$. ¿From that we routinely get $S(t) \le 2^{t^2/4+t/4}$ for $t \ge 12$, and then

$$R(c,t) = S(t)^c \le 2^{ct(t+1)/4}.$$

According to the detailed description given below, the above set M can be constructed, at any particular node of T, from its two tree descendants in time

$$O(c\cdot t^3)\cdot R(c,t)^2\,, \qquad \textbf{(13)}$$

and so the runtime bound of Theorem 12 follows since T has $|V(G)|$ leaves and $\mathit{\Theta}(|V(G)|)$ internal nodes.

For the sake of completeness we separately state the details of our dynamic algorithm for the c-colourability problem. Let T_x denote the subtree at a node x of the input parse tree T, and $\bar{G}_x$ be the t-labeled subgraph of G parsed by T_x.

- For each node z of T we process information contained in the set $M_T(z) = \big\{\mathit{\Gamma}_c\big(\bar{G}_z[X_1 = \chi^{-1}(1),\dots,X_1 = \chi^{-1}(c)]\big) : \chi$ is a proper c-colouring of $G_z\big\}$.
- If z is a leaf of T, then $M_T(z)$ consists of precisely c elements of the form $(\emptyset,\dots,\langle lab(z)\rangle,\dots,\emptyset)$.

– Assume an internal node z of T with the sons x and y (if one of them does not exist in T, the analysis is only simpler). Then $\bar{G}_z = \bar{G}_x \otimes[g \mid f_1, f_2]\ \bar{G}_y$. For every $\Gamma_c(\bar{G}_x) \in M_T(x)$ and $\Gamma_c(\bar{G}_y) \in M_T(y)$, we take Ψ_x^j the j-th entry (subspace) in $\Gamma_c(\bar{G}_x)$ and Ψ_y^j the j-th entry in $\Gamma_c(\bar{G}_y)$, where $j = 1, \dots, c$.

  Each subspace Ψ is recorded by its (arbitrary) basis. With elementary linear algebra we can check in time $O(t^3)$, from g and the bases of Ψ_x^j, Ψ_y^j, whether there are points (labelings) $Z_x \in \Psi_x^j$ and $Z_y \in \Psi_y^j$ with an odd intersection $|Z_x \cap g(Z_y)|$, i.e., by (4), an edge created between two j-coloured vertices of $\bar{G}_x$ and of $\bar{G}_y$. Then we can compute, from f_1, f_2 and Ψ_x^j, Ψ_y^j, in $O(t^3)$ time the subspace $\Psi_z^j = \langle f_1(\Psi_x^j) \cup f_2(\Psi_y^j)\rangle$ generated by the labelings of the j-coloured vertices of $\bar{G}_z$. If the odd-intersection tests above succeed for all $j = 1, \dots, c$, then in time $O(c\,t^3)$ we finally obtain $\Gamma_c(\bar{G}_z) \in M_T(z)$.
– Finally, in the root r of T, we simply check whether $M_T(r)$ is nonempty.

One small remark is neccessary about accessing the sets $M_T(z)$. We build in advance an indexing structure consisting of all $2^{t(t+1)/2}$ upper-triangular binary matrices – potential bases of all the subspaces of $GF(2)^t$. We let each matrix refer to the first one in the list which generates the same subspace. Even by brute force this takes time $O\big(t^3\, 2^{t(t+1)/2}\, S(t)\big)$ which is neglectable with respect to (13). Then we implement the characteristic vector of $M_T(z)$ as a c-dimensional bit array addressed via that index, achieving constant access time to it (modulo handling “big” t^2-bit numbers). ∎

Remark 4. Theorem 12 gives an FPT algorithm which is single-exponential in the parameter t. That speedup is achieved by the crucial observation, see in (4), that it is enough to record only the subspaces generated by the labelings of vertices in each colour class, and not all the vertex labelings separately (which would lead to trivial double-exponential dependency on t).

Non-FPT (pseudo-polynomial) algorithms. Besides FPT algorithms running in time $O(f(k) \cdot n^c)$ for a constant parameter time, there exist many algorithms in the literature running in time $O(n^{f(k)})$

which also belong to the class P for any one fixed k, but we better call these "pseudo-polynomial" to stress the fact that the exponent of n can grow when choosing different value of k.

Considering as a parameter k the clique-width of a graph G, where G is given in a form of a clique-width k-expression, there are known pseudo-polynomial algorithms; e.g. [11] for Hamiltonian path and various partitioning problems like into cliques, bipartite cliques, perfect matchings, etc, or [18] for the chromatic number or edge dominating set problems, or a general framework [14] for many vertex-partitioning problems. On the other hand, it is now known [12] that likely no FPT algorithms exist for the edge dominating set, Hamiltonian cycle or chromatic problems when parameterized by clique-width (unless FPT=W[1]).

For instance, we take a look at the chromatic number algorithm of Kobler and Rotics [18] which runs in time $O(n^{f(k)})$ where $f(k) = \Theta(4^k)$. If applied to a graph of rank-width t, the runtime bound would become $O(n^{g(t)})$ where $g(t) = O(4^{2^t})$. However, by further extending our formal machinery and reusing the ideas of the proof of Theorem 12, we can improve the algorithm of [18] to achieve better running time with respect to the rank-width parameter.

Theorem 14. *Assume that an input graph G is given in the form of a t-labeling parse tree T. Then the chromatic number of G can be determined in time*

$$O\left(|V(G)|^{p(t)}\right) \text{ where } p(t) = O(2^{t(t+1)/2}) .$$

Proof. Considering an arbitrary graph H and a set family $\mathcal{N}$, we express by $\nu(\mathcal{N})$ the claim that the sets in $\mathcal{N}$ form a partition of the vertex set $V(H)$ into nonempty(!) independent sets (a proper colouring of H). ν is a second-order formula and so the formal machinery of Section 4 is not directly applicable. Neverthless, we can further extend the meaning of the canonical equivalence of Section 4 to cover also this case over t-labeled graphs equipped with $\mathcal{N}$.

Consider two set families $\mathcal{N} \subseteq 2^X, \mathcal{N}' \subseteq 2^Y$ over disjoint ground sets $X \cap Y = \emptyset$. We call a family $\mathcal{M} \subseteq 2^{X\cup Y}$ a *matching union* of $\mathcal{N}$ and $\mathcal{N}'$ if $\mathcal{M} \restriction X = \mathcal{N}$, $\mathcal{M} \restriction Y = \mathcal{N}'$, and $|\mathcal{M}| = \max(|\mathcal{N}|, |\mathcal{N}'|)$ (where $\mathcal{M} \restriction X$ denotes the family of all intersections of X with sets from $\mathcal{M}$).

Then, for any t-labeled graphs $\bar{G}_1, \bar{G}_2$ and any set families $\mathcal{N}_i \subseteq 2^{V(G_i)}$, $i = 1, 2$ forming a partition of $V(\bar{G}_i)$, we define $(\bar{G}_1, \mathcal{N}_1) \approx_{\nu,t} (\bar{G}_2, \mathcal{N}_2)$ if and only if $|\mathcal{N}_1| = |\mathcal{N}_2|$ and the following

$$\textbf{(15)} \qquad \big(\bar{G}_1 \otimes [id]\ \bar{H}\big) \models \nu(\mathcal{N}_1' \cup \mathcal{N}_H') \text{ for some } \mathcal{N}_1' \text{ matching union of } \mathcal{N}_1, \mathcal{N}_H \iff$$

$$\big(\bar{G}_2 \otimes [id]\ \bar{H}\big) \models \nu(\mathcal{N}_2' \cup \mathcal{N}_H') \text{ for some } \mathcal{N}_2' \text{ matching union of } \mathcal{N}_2, \mathcal{N}_H$$

holds for all t-labeled graphs $\bar{H}$ (disjoint from $\bar{G}_1, \bar{G}_2$), and all disjoint $\mathcal{N}_H, \mathcal{N}_H' \subseteq 2^{V(H)}$ such that $\mathcal{N}_H \cup \mathcal{N}_H'$ is a partition of $V(\bar{H})$. See that $\approx_{\nu,t}$ captures all information neccessary to decide which subcolourings of $\bar{G}_1$ extend to colourings of any larger $\bar{G}_1 \otimes [id]\ \bar{H}$, where $\mathcal{N}_H$ extends existing colour classes of $\bar{G}_1$ into $\bar{H}$ and $\mathcal{N}_H'$ gives new colour classes exclusive to $\bar{H}$.

For $W \subseteq V(\bar{H})$ we recall the subspace $\langle \gamma(\bar{H}, W) \rangle$ of $GF(2)^t$ generated by the labelings of the vertices of W in $\bar{H}$, and we define a *multiset* $\Gamma_\#(\bar{H}, \mathcal{N}) = \{\langle \gamma(\bar{H}, W) \rangle : W \in \mathcal{N}\}$. The crucial finding, actually inspired by the colouring algorithm in [18], now reads:

(16) For any t-labeled graphs $\bar{G}_1, \bar{G}_2$ and any $\mathcal{N}_i \subseteq 2^{V(G_i)}$, $i = 1, 2$ such that $\bar{G}_i \models \nu(\mathcal{N}_i)$, it holds $(\bar{G}_1, \mathcal{N}_1) \approx_{\nu,t} (\bar{G}_2, \mathcal{N}_2)$ if $\Gamma_\#(\bar{G}_1, \mathcal{N}_1) = \Gamma_\#(\bar{G}_2, \mathcal{N}_2)$.

To prove this claim, we assume $\Gamma_\#(\bar{G}_1, \mathcal{N}_1) = \Gamma_\#(\bar{G}_2, \mathcal{N}_2)$ and, cf. (15), $\big(\bar{G}_1 \otimes [id]\ \bar{H}\big) \models \nu(\mathcal{N}_1' \cup \mathcal{N}_H')$. Since $\mathcal{N}_2^+ = \mathcal{N}_2' \cup \mathcal{N}_H'$ is a partition of the vertices of $\bar{G}_2 \otimes [id]\ \bar{H}$, for all possible matching unions $\mathcal{N}_2'$, for claiming $\nu(\mathcal{N}_2^+)$ it suffices to verify that all $W \in \mathcal{N}_2^+$ are independent in the graph $\bar{G}_2 \otimes [id]\ \bar{H}$. That is trivial if $W \in \mathcal{N}_2$ or $W \in \mathcal{N}_H'$. Otherwise, we consider any bijection $b : \mathcal{N}_1 \to \mathcal{N}_2$ preserving $\langle \gamma(\bar{G}_1, W) \rangle = \langle \gamma(\bar{G}_2, b(W)) \rangle$ for all $W \in \mathcal{N}_1$, and accordingly choose

$$\mathcal{N}_2' = \big\{\, W : \ W = b\big(U \restriction V(\bar{G}_1)\big) \cup \big(U \restriction V(\bar{H})\big),\, U \in \mathcal{N}_1' \setminus \mathcal{N}_1 \,\big\}\,.$$

By (4), $W \in \mathcal{N}_2'$ induces an edge between $\bar{G}_2$ and $\bar{H}$ in $\bar{G}_2 \otimes [id]\ \bar{H}$ if and only if the corresponding $U \in \mathcal{N}_1'$ does that in $\bar{G}_1 \otimes [id]\ \bar{H}$. Therefore, for our choice of matching union $\mathcal{N}_2'$, all $W \in \mathcal{N}_2'$ are also independent and (16) follows.

Recall that $\bar{G}_z$ denotes the labeled graph parsed by the subtree of the input parse tree T rooted at $z \in V(T)$. Thanks to (16), a dynamic algorithm for the chromatic number of G has to remember only the set $M_T(z)$ of those multisets $\varGamma_\#(\bar{G}_z, \mathcal{N})$ coming from proper colour partitions $\mathcal{N}$ of $V(G_z)$, at any particular node z of T. This information is trivial to construct at each leaf of T.

Consider a node z with the sons x and y (if one of them does not exist in T, the analysis is irrelevant). Every colour partition of $\bar{G}_z = \bar{G}_x \otimes[g \mid f_1, f_2]\ \bar{G}_y$ uniquely determines colour partitions of $V(\bar{G}_x)$ and of $V(\bar{G}_y)$. So to utilize the definition of $\varGamma_\#$, we define a *matching union signature* D_t^g (depending on the "join" relabeling g): D_t^g is the bipartite graph on the vertex set $\mathcal{S} \dot\cup \mathcal{S}$ where $\mathcal{S}$ is the family of all subspaces of $GF(2)^t$, and $\varPsi\varPsi' \in \mathcal{S} \times \mathcal{S}$ is an edge of D_t^g iff no labeling $Z \in \varPsi$ has an odd intersection with any $g(Z')$, $Z' \in \varPsi'$. Each edge of D_t^g has, moreover, an integer weight (starting from 0).

Having two multisets $\varGamma_1, \varGamma_2$ of subspaces of $GF(2)^t$, and a suitably weighted signature D_t^g as above; a D_t^g-matching union $\varGamma_0$ of $\varGamma_1$ and $\varGamma_2$, with f_1, f_2 relabelings, is obtained iteratively as follows: For every edge $\varPsi_1\varPsi_2$ of positive weight m in D_t^g, the multiplicity of $\varPsi_i$ in $\varGamma_i$, $i = 1, 2$ is reduced by m, and the combined subspace $\langle f_1(\varPsi_1) \cup f_2(\varPsi_2)\rangle$ is added to $\varGamma_0$ again with multiplicity m. Remaining elements of $\varGamma_1, \varGamma_2$ (with multiplicities) are copied to $\varGamma_0$ at the end. The whole operation is defined only if each vertex of D_t^g satisfies that the sum of weights of its incident edges is at most its multiplicity in $\varGamma_1$ or $\varGamma_2$, respectively.

Now we easily conclude:

(17) A subspace $\varGamma_z$ belongs to $M_T(z)$ if and only if $\varGamma_z$ is obtained as a D_t^g-matching union of some $\varGamma_x \in M_T(x)$ and some $\varGamma_y \in M_T(y)$, for a suitably weighted matching union signature D_t^g.

So, finally, while (16) specifies what information we keep in dynamic processing of the input parse tree T, latter (17) shows how to process this information. Such a dynamic algorithm then runs in time $O\big(p(G,t)^2 \cdot q(G,t) \cdot S(t)^2 \cdot t^3 \cdot |V(G)|\big)$, where $p(G,t)$ denotes the number of possible distinct $\varGamma_\#(\bar{G}, \mathcal{N})$, and $q(G,t)$ stands for the number of distinct weightings of the matching union signature D_t^g.

For simplicity, we provide only short arguments giving rather weak (but sufficient) bounds on p, q here: $p(G,t)$ can be bounded from above by $|V(G)|^{S(t)}$ where $S(t) \le 2^{t(t+1)/4}$ (for $t \ge 12$) is the number of distinct subspaces of $GF(2)^t$ estimated in the proof of

Theorem 12—consider that the multiplicity of any subspace in the multiset $\Gamma_{\#}(\bar{H}, \mathcal{N})$ is at most the number of nonempty colour classes. With analogous arguments we also get $q(G,t) \leq |V(H)|^{S(t)^2}$. These estimates then lead to a runtime bound of order $|V(G)|^{p(t)}$ where $p(t) \leq 2S(t) + S(t)^2 + o(t^2) + 1 = O(S(t)^2) = O(2^{t(t+1)/2})$. ∎

6 Concluding Notes

We have provided a wide range of formal mathematical tools for constructing dynamic algorithms on graphs with bounded-width rank-decompositions in our paper. The employed mathematical formalism is, we believe, close also to the theoretical computer science community and suitable for designing actual algorithms. This paper elaborates on the ideas of Courcelle and Kanté [7] as well as allows better interpretation of their results. Although our examples of actual algorithms focused on graph colourability, the provided tools can, of course, be used for handling many other properties and problems in a similar manner.

For instance, among other things, we would like to suggest that Theorem 14 can straightforwardly be extended to compute the chromatic polynomial of input graphs of bounded rank-width with about the same time complexity, a task that has been done with respect to clique-width using more complicated tools in [3].

Generally, it is an interesting question (to which we do not have an answer right now) whether Theorem 5 and ideas of (11) could be used to give FPT algorithms for problems beyond the scope of the *Lin*EMSO properties [8] and of the vertex-partitioning framework [14]. Yet another suggestion of a future research would be to try to identify a fragment of MS_1 logic for which there exist FPT algorithms with a single-exponential dependency on the rank-width parameter. Design of such algorithms appears significantly more complicated than an analogous task for the tree-width parameter.

Notice, finally, in Section 5 that it is sometimes possible to find (even exponentially) better algorithms by properly utilizing dynamic programming and providing better representations of the actual equivalence classes characterizing the problem in our framework. Such a phenomenon deserves further study, too.

References

1. Abrahamson, K.A. and Fellows, M.R.: Finite Automata, Bounded Treewidth, and Well-Quasiordering. In: *Graph Structure Theory, Contemporary Mathematics 147*. American Mathematical Society (1993) 539–564.
2. Arnborg, S., Lagergren, J., and Seese, D.: Problems easy for Tree-decomposible Graphs. In: *Proc. 15th Colloq. Automata, Languages and Programming*. Volume 317 of Lecture Notes in Comput. Sci., Springer, Berlin (1988) 38–51.
3. Averbouch, I., Godlin, B., Makowsky, J.A., and Rotics, U.: Computing graph polynomials on graphs of bounded clique-width . In: *Graph-theoretic concepts in computer science*. Volume 4271 of Lecture Notes in Comput. Sci., Springer, Berlin (2006) 191–204.
4. Bandelt, H.-J. and Mulder, H.M.: Distance-hereditary graphs. *J. Combin. Theory Ser. B* **41**(2) (1986) 182–208.
5. Corneil, D.G. and Rotics, U.: On the relationship between cliquewidth and treewidth. *SIAM J. Comput.* **34**(4) (2005) 825–847.
6. B. Courcelle: The monadic second-order logic of graphs I. Recognizable sets of Finite Graphs. *Information and Computation* **85** (1990) 12–75.
7. Courcelle, B. and Kanté, M.M.: Graph Operations Characterizing Rank-Width and Balanced Graph Expressions. In: *Graph-theoretic concepts in computer science*. Volume 4769 of Lecture Notes in Comput. Sci., Berlin, Springer (2007) 66–75.
8. Courcelle, B., Makowsky, J.A., and Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.* **33**(2) (2000) 125–150.
9. Courcelle, B. and Olariu, S.: Upper bounds to the clique width of graphs. *Discrete Appl. Math.* **101**(1-3) (2000) 77–114.
10. Downey, R.G. and Fellows, M.R.: *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.
11. Espelage, W., Gurski, F., and Wanke, E.: How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In: *Graph-theoretic concepts in computer science*. Volume 2204 of Lecture Notes in Comput. Sci., Berlin, Springer (2001) 117–128.
12. Fomin, F., Golovach, P., Lokshtanov, D. and Saurabh, S.: Clique-width: On the Price of Generality. To apprear in the *Proceedings of SODA 2009*, ACM.
13. Ganian, R.: *Automata formalization for graphs of bounded rank-width*. Master thesis. Faculty of Informatics of the Masaryk University, Brno, Czech republic (2008).
14. Gerber, M.U. and Kobler, D.: Algorithms for vertex-partitioning problems on graphs with fixed clique-width. *Theoret. Comput. Sci.* **299**(1-3) (2003) 719–734.
15. Goldman, J. and Rota, G.-C.: The number of subspaces of a vector space. In: *Recent Progress in Combinatorics*, ed. W.T. Tutte. Academic Press (1969) 75–83.
16. Hliněný, P.: Branch-width, parse trees, and monadic second-order logic for matroids. *J. Combin. Theory Ser. B* **96**(3) (2006) 325–351.
17. Hliněný, P. and Oum, S.: Finding Branch-decomposition and Rank-decomposition. *SIAM J. Comput.* **38** (2008) 1012–1032.
18. Kobler, D. and Rotics, U.: Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Appl. Math.* **126**(2-3) (2003) 197–221.
19. Oum, S.: Rank-width and vertex-minors. *J. Combin. Theory Ser. B* **95**(1) (2005) 79–100.

20. Oum, S. and Seymour, P.: Approximating clique-width and branch-width. *J. Combin. Theory Ser. B* **96**(4) (2006) 514–528.
21. Robertson, N. and Seymour, P.: Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B* **52**(2) (1991) 153–190.