Discrete Applied Mathematics 158 (2010) 851–867

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

On parse trees and Myhill–Nerode-type tools for handling graphs of bounded rank-width

Robert Ganian, Petr Hliněný *

Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

ARTICLE INFO

Article history:
Received 24 February 2009
Received in revised form 26 October 2009
Accepted 29 October 2009
Available online 12 November 2009

Keywords:
Rank-width
Parameterized algorithm
Myhill–Nerode theorem
Graph colouring

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 mathematical 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). The main advantage of this framework is a fine control over the runtime dependency on the rank-width parameter. Our new approach is linked to a work of Courcelle and Kanté [7] who first proposed algebraic expressions with a so-called bilinear graph product as a better way of handling rank-decompositions, and to a parallel recent research of Bui-Xuan, Telle and Vatshelle.

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 or property (so-called parameterized algorithmics; see [11]). Often the input graphs are restricted to have bounded tree-width or branch-width, but another weaker useful structural restriction has appeared with the notion of *clique-width*, defined by Courcelle and Olariu in [10].

Now, many hard graph problems (particularly all those expressible in MS_1 logic; see Section 4) are solvable in polynomial time [9,12,21,17], 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. Unfortunately, for fixed $k > 3$, it is not known how to find a k-expression of an input graph having clique-width at most k in polynomial time.

Rank-width (see Section 2) is another graph complexity measure introduced by Oum and Seymour [23,22], 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 [20]. Since, in reality, clique-width can be up to exponentially larger than rank-width [5], it now appears desirable to design parameterized algorithms running directly on a 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 [23]; 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 [7,8] gave an alternative characterization of a rank-decomposition using bilinear product terms over multi-coloured graphs—see Section 2 and particularly Theorem 2.6. In our view, the latter characterization can be

* Corresponding author.
E-mail addresses: ganian@mail.muni.cz (R. Ganian), hlineny@fi.muni.cz (P. Hliněný).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2009.10.018

equivalently formulated in terms of *labeling parse trees* (rank-width parse trees of [14]), which straightforwardly leads to a new Myhill–Nerode-type characterization of finite state properties of graphs of bounded rank-width in Theorem 3.4. This important statement then opens new mathematical ground for easy design of better FPT algorithms in subsequent Sections 5 and 6, and has further generalizations as e.g. in [16] focusing on XP algorithms.

Recently, Bui-Xuan, Telle and Vatshelle [4] have also studied this topic in terms of H-join decompositions, cf. Remark 2.8. They, moreover, gave new FPT algorithms [4] for solving the independent set, colourability, and dominating set problems on graphs of (bounded) rank-width t in time $O(2^{\Theta(t^2)}n)$. These particular algorithms, actually the one for dominating set, have inspired us to come with a significantly enhanced formal scheme which we present in Section 5.

We now outline the structure of this paper which is a full extended version of a presentation [15] at IWOCA 2008.

After providing some technical definitions and basic known results in Section 2, we state in Section 3 a useful characterization (Theorem 3.4) 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 necessarily closed) defines a regular language over "equipped" bounded rank-width graphs. That, particularly, provides an alternative self-contained combinatorial proof of the relevant Courcelle–Makowsky–Rotics' [9] and Courcelle–Kanté's [8] results.

The main new contributions of our paper, which extend far beyond the scope of the conference version [15], are then presented in Sections 5 and 6. We provide a new formal mathematical approach to designing dynamic FPT algorithms on rank-decompositions of graphs, see Definition 5.2, which allows a much finer control over dependency of runtime on the rank-width and yet it stays very general. Applications of this new scheme are comparable with the above mentioned algorithms in [4], and they include solving new hard problems like co-colouring and acyclic colouring (for a fixed number of colours), or feedback vertex set in time $O(2^{\Theta(t^2)}n)$ for graphs of rank-width t.

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 and 4.

Branch-decompositions

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.

Definition 2.1 (*Branch-width*). 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 width over all branch-decompositions of f. If $|M| \le 1$, 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 [25] 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— Definition 2.2, in which we consider the vertex set $V(G) = M$ of a graph G as the ground set.

Rank-decompositions

For a graph G and $U, W \subseteq V(G)$, let $\boldsymbol{A}_G[U, W]$ denote the $U \times W$-submatrix of the adjacency matrix over the two-element field GF(2), i.e. the entry $a_{u,w}$, $u \in U$ and $w \in W$, of $\boldsymbol{A}_G[U, W]$ is 1 if and only if $\{u, w\}$ is an edge of G.

Definition 2.2 (*Rank-width*). The *cut-rank* function ρ_G of a graph G is defined as follows: For a bipartition (U, W) of the vertex set $V(G)$, $\rho_G(U) = \rho_G(W)$ 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 (Definition 2.1) of the cut-rank function ρ_G of G on $M = V(G)$, respectively.

Perhaps the main reason in favor of rank-width over more traditional clique-width is the fact that there are efficient parameterized algorithms for constructing rank-decompositions [23,20]:

Theorem 2.3 (*[20]*). *For every fixed t there is an $O(n^3)$-time FPT 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 studied, e.g. [3], before. It turns out that

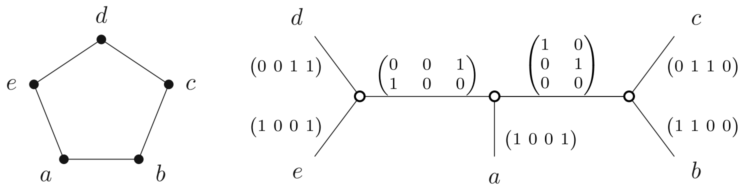


Fig. 1. A rank-decomposition of the graph cycle C_5.

distance-hereditary graphs are exactly the graphs of rank-width one [22], and this simple fact explains many of their "nice" algorithmic properties.

Labeling join

In a search for a "more suitable form" of a rank-decomposition, Courcelle and Kanté [7,8] 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 2.6). In Definitions 2.4 and 2.5 we introduce (following [14]) the same idea in terms of a labeling join and parse trees which we propose as a more convenient notation for the results in the next sections. One should note that an analogous idea also underlies H-join decompositions of Bui-Xuan, Telle and Vatshelle [4].

A *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. We will often view (cf. [7,8] again) a t-labeling of G equivalently as a mapping $V(G) \to \mathrm{GF}(2)^t$ into the *binary vector space* of dimension t.

Definition 2.4 (*Join* $\otimes$). Considering t-labeled graphs $\bar{G}_1 = (G_1, lab^1)$ and $\bar{G}_2 = (G_2, lab^2)$, a *t-labeling join* $\bar{G}_1 \otimes \bar{G}_2$ is defined on the disjoint union of G_1 and G_2 by adding all edges $\{u, v\}$ such that $|lab^1(u) \cap lab^2(v)|$ is odd, where $u \in V(G_1)$, $v \in V(G_2)$. The resulting graph is unlabeled.

Notice that, considering the scalar product $\cdot$ of vectors, $\{u, v\}$ is an edge of $\bar{G}_1 \otimes \bar{G}_2$ if and only if $lab^1(u) \cdot lab^2(v) = 1$ over GF(2).

Labeling parse trees

A *t-relabeling* is a mapping $g : L_t \to 2^{L_t}$. In linear algebra terms, a t-relabeling g is in a natural one-to-one correspondence with a *linear transformation* $g : \mathrm{GF}(2)^t \to \mathrm{GF}(2)^t$, i.e. a $t \times t$ binary matrix $\boldsymbol{T}_g$. For a t-labeled graph $\bar{G} = (G, lab)$ we define $g(\bar{G})$ as the same graph with a vertex t-labeling $lab' = g \circ lab$. Here $g \circ lab$ stands for the linear transformation g applied to the labeling lab, or equivalently $lab' = lab \times \boldsymbol{T}_g$ as matrix multiplication. Informally, g is applied separately to each label in $lab(v)$ and the outcomes are summed up "modulo 2"; e.g. for $lab(v) = \{1, 2\}$ and $g(1) = \{1, 3, 4\}$, $g(2) = \{1, 2, 3\}$, we get $g \circ lab(v) = \{2, 4\} = \{1, 3, 4\} \triangle \{1, 2, 3\}$.

Definition 2.5 (*Parse Tree*). 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 composition* – over pairs of t-labeled graphs $\bar{G}_1 = (G_1, lab^1)$ and $\bar{G}_2 = (G_2, lab^2)$ defined

$$\bar{G}_1 \otimes [g \mid f_1, f_2]\ \bar{G}_2 \;=\; \bar{H} \;=\; \big(\bar{G}_1 \otimes g(\bar{G}_2),\ lab\big)$$

where the new labeling is $lab(v) = f_i \circ lab^i(v)$ for $v \in V(G_i)$, $i = 1, 2$.

A *t-labeling parse tree* T, see also [14, 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 composition 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 Figs. 2 and 3 for an illustration of the definition, and notice that $\{u, v\}$ is an edge of $\bar{H}$ if and only if $lab^1(u) \times \boldsymbol{T}_g^T \times lab^2(v)^T = 1$ over GF(2).

We make three short notes to this definition. First, our labeling composition is clearly equivalent to the bilinear product of [7,8]. Second, the role of relabeling g in $\otimes[g \mid f_1, f_2]$ is unavoidable for Theorem 2.6 to hold true, but fortunately, it can be replaced with the generic join $\otimes$ if the resulting labeling is unimportant, cf. Proposition 3.2 later on. Third, 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.

Analogously to the work of Courcelle and Kanté we get a crucial statement:

Theorem 2.6 (*Rank-width Parsing Theorem [7,14]*). *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 $O(t^2 \cdot |V(G)|^2)$.*

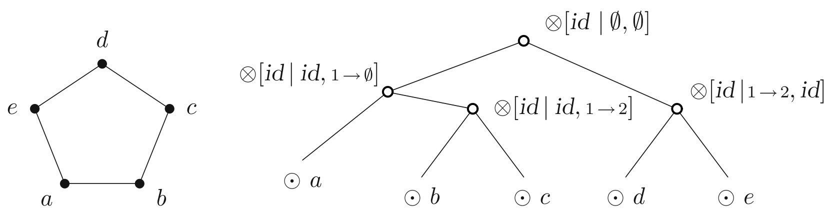


Fig. 2. An example of a labeling parse tree which generates a 2-labeled cycle C_5, with symbolic relabelings at the nodes (*id* denotes the relabeling preserving all labels, and $\emptyset$ is the relabeling "erasing" all labels).

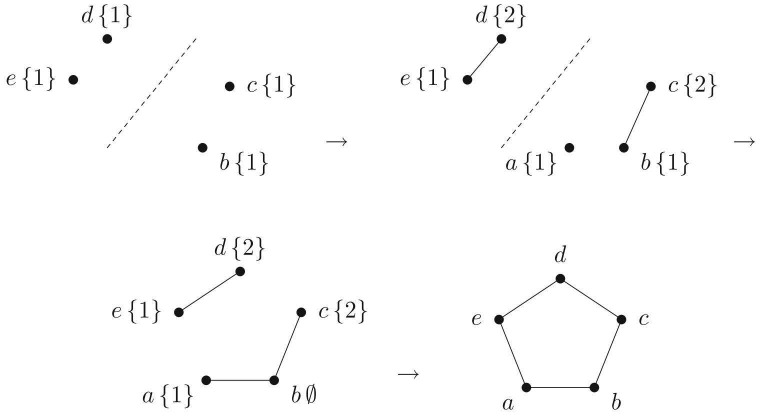


Fig. 3. "Bottom-up" generation of C_5 by the parse tree from Fig. 2.

Proof. The first part of 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 straightforwardly equivalent to our $\otimes\,[f \mid g, h]$, and a t-labeled singleton vertex $\ell(\odot)$ can be emulated with two nodes—the $\odot$ singleton symbol with a $\otimes\,[\emptyset \mid \ell, \emptyset]$ "relabeling" parent.

On the other hand, no explicit time complexity bound for turning a rank-decomposition into a labeling parse tree was given in [7,8]. While this task is clearly polynomial, our (improved) time bound of $O(t^2 \cdot |V(G)|^2)$ is nontrivial and it implicitly follows from an independent self-contained proof of the first part in the first author's Master thesis [14, Chapter 6]. For the sake of completeness, we present this whole proof in a condensed form here.

Assume (R, μ) is a rank-decomposition of G of width t, cf. Definition 2.2, such that R has no degree-2 vertices. We choose the root of the parse tree T constructed from R as subdivision of an arbitrary edge of R. Then we process the following information recursively from the leaves of T to its root. At every leaf ℓ of T, we record a single-entry matrix $\boldsymbol{L}_\ell = (1)$ and a one-row matrix $\boldsymbol{R}_\ell = \boldsymbol{A}_G[\{v\}, V(G-v)]$ defining the adjacencies of the vertex $v = \mu^{-1}(\ell)$ of G to the remaining vertices. This is, by the way, the only time we query the graph G; the rest is pure linear algebra.

Considering a node x of T we denote by T_x the subtree of T rooted at x, by G_x the subgraph of G induced by $\mu^{-1}(V(T_x))$, and by $V_x = V(G_x)$ and $V_x^{\uparrow} = V(G) \setminus V_x$. If x is not a leaf, then x has two sons y, z (left and right) in T and $V_x = V_y \cup V_z$. Suppose we have already computed $|V_y| \times t_y$- and $t_y \times |V_y^{\uparrow}|$-matrices $\boldsymbol{L}_y$ and $\boldsymbol{R}_y$ at y (we have done this above for the leaves of T), such that $t_y \le t$ and $\boldsymbol{L}_y \times \boldsymbol{R}_y = \boldsymbol{A}_G[V_y, V_y^{\uparrow}]$. Furthermore, suppose that $\boldsymbol{L}_y$ contains a unit submatrix of full rank, i.e. that $\boldsymbol{R}_y$ is a submatrix of $\boldsymbol{A}_G[V_y, V_y^{\uparrow}]$. Suppose analogously such $|V_y| \times t_z$- and $t_z \times |V_y^{\uparrow}|$-matrices $\boldsymbol{L}_z$ and $\boldsymbol{R}_z$ at z.

The task now is to construct the composition operator, cf. Definition 2.5, $\widehat{\otimes}_x = \otimes\,[g^x \mid f_1^x, f_2^x]$ at the node x. The meaning of our matrices $\boldsymbol{L}_y$ and $\boldsymbol{L}_z$ is that their row vectors form labelings of the vertices of G_y and G_z, respectively, in the parse tree T. We denote by $\boldsymbol{R}'_y$ the submatrix of $\boldsymbol{R}_y$ whose columns correspond to the vertices of V_z, and analogously $\boldsymbol{R}'_z$ of $\boldsymbol{R}_z$ with the columns corresponding to V_y. Then $\boldsymbol{L}_y \times \boldsymbol{R}'_y = \boldsymbol{A}_G[V_y, V_z] = \left(\boldsymbol{L}_z \times \boldsymbol{R}'_z\right)^T$, and so (recall the unit submatrices of $\boldsymbol{L}_y, \boldsymbol{L}_z$) both $\boldsymbol{R}'_y$ and $\boldsymbol{R}'^T_z$ are submatrices of $\boldsymbol{A}_G[V_y, V_z]$ intersecting in a $t_y \times t_z$ matrix $\boldsymbol{C}_{y,z}$. Moreover, $\boldsymbol{L}_y \times \boldsymbol{C}_{y,z} \times \boldsymbol{L}_z^T = \boldsymbol{A}_G[V_y, V_z]$, and hence the relabeling g^x simply multiplies by the matrix $\boldsymbol{C}_{y,z}^T$ (expanded with zeroes as necessary).

Remaining entries f_1^x, f_2^x of composition $\widehat{\otimes}_x$ will be constructed along with our auxiliary matrices $\boldsymbol{L}_x$ and $\boldsymbol{R}_x$ at x (again, $\boldsymbol{L}_x$ shall give labelings of the vertices of G_x). We denote by $\boldsymbol{R}''_y$ the submatrix of $\boldsymbol{R}_y$ whose columns correspond to the vertices of $V_x^{\uparrow}$, and $\boldsymbol{R}''_z$ of $\boldsymbol{R}_z$ analogously. Since $\boldsymbol{S}_{y,z} = \begin{pmatrix}\boldsymbol{R}''_y\\ \boldsymbol{R}''_z\end{pmatrix}$ is a submatrix of $\boldsymbol{A}_G[V_x, V_x^{\uparrow}]$, its rank is $t_x \le t$ by Definition 2.2. We compute, using Gaussian elimination, a submatrix $\boldsymbol{R}_x$ consisting of t_x independent rows of $\boldsymbol{S}_{y,z}$. Notice that such $\boldsymbol{R}_x$ is a basis of the row space of whole $\boldsymbol{A}_G[V_x, V_x^{\uparrow}]$.

As a byproduct of the Gaussian elimination procedure above, we get matrices $\boldsymbol{D}_y, \boldsymbol{D}_z$ such that $\boldsymbol{D}_y \times \boldsymbol{R}_x = \boldsymbol{R}''_y$ and $\boldsymbol{D}_z \times \boldsymbol{R}_x = \boldsymbol{R}''_z$. Since $\boldsymbol{L}_y \times \boldsymbol{R}''_y = \boldsymbol{A}_G[V_y, V_x^{\uparrow}]$ and $\boldsymbol{L}_z \times \boldsymbol{R}''_z = \boldsymbol{A}_G[V_z, V_x^{\uparrow}]$, we can choose $\boldsymbol{L}_x = \begin{pmatrix} \boldsymbol{L}_y \times \boldsymbol{D}_y \\ \boldsymbol{L}_z \times \boldsymbol{D}_z \end{pmatrix}$ as a $|V_x| \times t_x$-matrix fulfilling $\boldsymbol{L}_x \times \boldsymbol{R}_x = \boldsymbol{A}_G[V_x, V_x^{\uparrow}]$. This in turn means that the relabelings f_1^x, f_2^x are defined as right multiplication by the matrices $\boldsymbol{D}_y$ and $\boldsymbol{D}_z$, respectively (again expanded with zeroes as necessary). Moreover, the rows of $\boldsymbol{L}_x$, determined by the position of $\boldsymbol{R}_x$ as a submatrix in $\boldsymbol{A}_G[V_x, V_x^{\uparrow}]$, form a unit submatrix of full rank t_x in $\boldsymbol{L}_x$ as needed.

At last, we review the computational complexity of the whole process. At every node x of T, we compute matrices $\boldsymbol{R}_x$ and $\boldsymbol{L}_x$ in time $O(t^2 \cdot |V(G)|)$, and the number of nodes is $O(|V(G)|)$. □

Remark 2.7. We suggest that the "nearly linear" term $|V(G)|^2$ in time complexity of Theorem 2.6 can be improved to linear $|E(G)|$ if one works with a suitable "sparse" matrix representation and carefully reconsiders all the technical details, but such complication would not be profitable in our context in which we use Theorem 2.6 together with Theorem 2.3 to construct an optimal labeling parse tree of a given graph G in parameterized $O(|V(G)|^3)$ time.

Remark 2.8. Besides the bilinear product terms and labeling parse trees (Definition 2.5 and Theorem 2.6), another alternative characterization of rank-width has popped up recently: Bui-Xuan, Telle and Vatshelle [4] defined a so-called H-join operation, and proved that graphs of rank-width $\le k$ are exactly those having an R_k-join decomposition [4, Theorem 4.3].

While, on one hand, a bare R_k-join decomposition is essentially equivalent to a rank-decomposition (Definition 2.2), the added value of [4] which have made their approach suitable for dynamic algorithm design is in associating the nodes of a decomposition with "external module partitions" on the vertex set (where these partitions, in contrast to our parse tree approach, are constructed on-the-fly, cf. [4, Lemma 3.2]). While a transformation from a parse tree to external module partitions is straightforward and trivial, the other direction seems nontrivial and could, perhaps, be related to our proof of Theorem 2.6.

3. Regularity theorem for rank-width

Our goal is to develop further new mathematical formalisms for easier handling of certain algorithmic problems on graphs of bounded rank-width. A typical idea of a dynamic algorithm is

- to capture "all relevant information" about the studied problem through a restricted ("decomposed") part of the input, and
- to process this information bottom-up in the given "decomposition" (whatever this term means in a particular case).

Of course, besides finding a suitable "decomposition", the main question here is how to correctly specify the meaning of "relevant information". In contrast to usual problem specific or ad hoc approaches, our formal method is closely tied with the classical Myhill–Nerode regularity tool in automata theory. That is possible since our parse trees (Definition 2.5), 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 [11, Chapter 6] (for graphs of bounded tree-width) and in [19] (for matroids of bounded branch-width) before. The case of rank-width, however, brings some new obstacles.

We make two simple technical remarks. First, it could be necessary to interchange the operands of a t-labeling composition which itself is not commutative. Since a t-relabeling g is a linear transformation defined by a matrix $\boldsymbol{T}_g$ (Section 2), we can define a "transposed" t-relabeling g^T as the linear transformation by $\boldsymbol{T}_g^T$.

Proposition 3.1. *Let $\bar{G}_1, \bar{G}_2$ be t-labeled graphs and $g : L_t \to 2^{L_t}$ be a t-relabeling. If g^T is the transposed t-relabeling of g, then*

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

As we have already mentioned, the role of specific relabeling g in $\otimes [g \mid f_1, f_2]$ is rather unimportant if we do not care about the resulting labeling, cf. the next immediate claim:

Proposition 3.2. *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*

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

Hence we have got a generic summation operator $\otimes$ (the *labeling join*, Definition 2.4) making an unlabeled graph out of two labeled ones, which is an essential ingredience in the coming definitions.

The canonical equivalence

Let Π_t denote the finite set (alphabet) of all the t-labeling composition 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 $\overline{\mathcal{R}_t}$ – the *t-parsing universe* – is the class of all t-labeled graphs parsed by the trees from P_t, then (Theorem 2.6) $G \in \mathcal{R}_t$ if and only if $\bar{G} \in \overline{\mathcal{R}_t}$ for some t-labeling $\bar{G}$ of G.

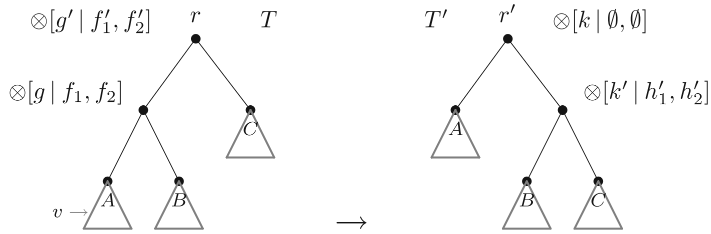


Fig. 4. Right parse tree rotation.

In analogy to the classical theory of regular languages we give:

Definition 3.3. Let $\mathcal{D}$ be any class of graphs, and $\mathcal{D}_t = \mathcal{D} \cap \mathcal{R}_t$. The *canonical equivalence* of $\mathcal{D}_t$, denoted by $\approx_{\mathcal{D},t}$, is defined 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 $\bar{H} \in \overline{\mathcal{R}}_t$,

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

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.

This informal finding can be formalized as follows (cf. [14, Chapter 7]):

Theorem 3.4 (*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 $\bar{G}_i \otimes \bar{H}_{i,j} \in \mathcal{D}_t$ but $\bar{G}_j \otimes \bar{H}_{i,j} \notin \mathcal{D}_t$, 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 3.6 and Proposition 3.2, 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 $\bar{G}_i \otimes \bar{H}_{i,j} \in \mathcal{D}_t$,
- and the graph parsed by $S_j \diamond_x U_{i,j}$ equals up to labeling $\bar{G}_j \otimes \bar{H}_{i,j} \notin \mathcal{D}_t$.

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

Corollary 3.5. *There is a natural bijection between the states of the tree automaton of Theorem 3.4 and the classes of the canonical equivalence $\approx_{\mathcal{D},t}$.*

Lemma 3.6. *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 $G = \bar{G}_v \otimes [\ell \mid \emptyset, \emptyset]\ \bar{H}$, 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 3.1, 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 composition 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: $k = id$, $k' = f_2^T \circ g'$, $h_1' = g$, 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 3.7. Notice that the arguments used in our proof of Theorem 3.4 *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 without increasing k. 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 3.4 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 rank-width-t members in $\mathcal{C} \cap \mathcal{R}_t$, it is enough to identify the classes of the canonical equivalence $\approx_{\mathcal{C},t}$ (Definition 3.3). We actually give below finitely many classes $\mathcal{X} = \{X_0, X_1, X_2, \dots\}$ of a refinement of $\approx_{\mathcal{C},t}$.

Assume a t-labeled graph $\bar{G} = (G, lab)$ with a proper 3-colouring χ. Let, for $i = 1, 2, 3$, $\gamma_i(\bar{G}, \chi) = \{lab(u) : u \in V(G) \wedge \chi(u) = i\}$. Then

- $X_0 = \{\bar{G} :\ G \text{ is not 3-colourable}\}$, and
- $X_1, X_2, \dots, X_{j(t)}$ are the equivalence classes of $\sim$, where over t-labeled graphs $\bar{G}_1 \sim \bar{G}_2$ if and only if it holds

$$\begin{aligned}&\big\{\big(\gamma_1(\bar{G}_1, \chi), \gamma_2(\bar{G}_1, \chi), \gamma_3(\bar{G}_1, \chi)\big) : \chi \text{ is a 3-colouring of } G_1\big\}\\ &\quad = \big\{\big(\gamma_1(\bar{G}_2, \chi), \gamma_2(\bar{G}_2, \chi), \gamma_3(\bar{G}_2, \chi)\big) : \chi \text{ is a 3-colouring of } G_2\big\}.\end{aligned} \tag{3.8}$$

Proposition 3.9. *If $\bar{G}_1$ and $\bar{G}_2$ belong to the same class of $\mathcal{X}$, then $\bar{G}_1 \approx_{\mathcal{C},t} \bar{G}_2$.*

The fact that the 3-colourability problem is efficiently solvable (even by a tree automaton) on graphs of bounded rank-width then follows from Theorems 2.3 and 3.4. This is also extendable to c-colourability for any fixed c.

Returning to the original motivation of this section, we can now say what "relevant information" about the 3-colourability problem we want to remember in a dynamic algorithm solving it on a labeling parse tree; it is the set (3.8). However, is this optimal? On one hand, the number of our classes $j(t)$ is a constant independent of the input size, and so it does not matter in the $O()$ notation. On the other hand, the amount of information we have to remember in (3.8) is "double-exponential" in the rank-width t, i.e. $j(t)$ is of order $exp(2^{3 \cdot 2^t})$, and that can be too much in practical applications. A much better analysis—showing that there are at most $2^{O(t^2)}$ distinct canonical classes of the c-colourability problem, can be found in [4, Section 3.3] and also in our Section 6.

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 [9] for MSO definable properties of adjacency graphs if a k-expression (cf. clique-width) of bounded k is given on the input. This readily extends to graphs with a given rank-decomposition of bounded width when this decomposition is translated to a suitable algebraic expression, cf. Courcelle–Kanté's [7] and [8, Theorems 3.3, 4.3].

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 instance, we show an MS_1 expression of the 3-colourability property of a graph:

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

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

To avoid possible confusion we remark that the 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 and Kanté [8, Theorem 3.3] can be alternatively set up in the scope of our Rank-width regularity Theorem 3.4. 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 $\approx_{\mathcal{F},t}$ (Definition 3.3) to an equivalence $\approx^{\sigma}_{\phi,t}$ (Definition 4.1) allowing for formulas ϕ with free variables.

This new view shall not only be an elementary combinatorial alternative to the proofs [9,8] which used MSO interpretation (transduction) of the graphs generated by decompositions into labeled binary trees, but also leads to new Theorem 4.2 which could be of independent interest (see Remark 4.9 and the proof of Theorem 4.13).

Extended canonical equivalence of MS_1 *formulas*

We propose an extension analogous to the previous works [1,19], 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 $\bar{G}$ 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 $x \in Fr$.

For simplicity, we will use the established notation $\bar{G}$ also to denote partially equipped labeled graphs. See that if $\bar{G}$ is σ-partially equipped and $\bar{H}$ is σ^--partially equipped, then the graph $M = \bar{G} \otimes \bar{H}$ has a full and consistent interpretation for all the free variables of ϕ (hence this M is a *logic model* for ϕ). So we can formulate:

Definition 4.1. Let ϕ be an MS_1 formula, and σ be a partial equipment signature of ϕ. The *extended canonical equivalence* $\approx^{\sigma}_{\phi,t}$ on the class of all t-labeled σ-partially equipped graphs is defined as follows:

$$\bar{G}_1 \approx^{\sigma}_{\phi,t} \bar{G}_2 \quad \text{if and only if } (\bar{G}_1 \otimes \bar{H}) \models \phi \iff (\bar{G}_2 \otimes \bar{H}) \models \phi$$

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

Comparing to Definition 3.3, we have extended $\approx^{\sigma}_{\phi,t}$ in two directions. First, by allowing free variables in ϕ we enlarge the studied universe to partially equipped graphs. Second, this universe is further enlarged by allowing all t-labeled underlying graphs—not only those as from the t-parsing universe $\overline{\mathcal{R}}_t$. Yet we can prove:

Theorem 4.2. *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: $\phi \equiv (v \in W)$, $(v = w)$, $\mathrm{mod}_{p,q}(W)$, 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).

(4.3) 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 $\{u, v\}$ 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 $\otimes$, 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 [19, 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:

(4.4) 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.

(4.5) 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 $(\bar{G}_1 \otimes \bar{H}) \models \phi$ but $(\bar{G}_2 \otimes \bar{H}) \not\models \phi$. Then it has to be $(\bar{G}_1 \otimes \bar{H}) \models \psi$ (or $\models \eta$) but $(\bar{G}_2 \otimes \bar{H}) \not\models \psi$ (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 $\exists W\, \psi(W)$.

(4.6) 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 $(\bar{G}_1 \otimes \bar{H}) \models \phi$ but $(\bar{G}_2 \otimes \bar{H}) \not\models \phi$. 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 $\left(\bar{G}_1[W = S_W] \otimes \bar{H}[W = S'_W]\right) \models \psi$, whilst $\left(\bar{G}_2[W = T_W] \otimes \bar{H}[W = S'_W]\right) \not\models \psi$ 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 (4.6).

(4.7) 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 $(\bar{G}_1 \otimes \bar{H}) \models \phi$ but $(\bar{G}_2 \otimes \bar{H}) \not\models \phi$. Let $x_v \in V(\bar{G}_1) \cup V(\bar{H})$ be an interpretation of the variable v that satisfies ψ over $\bar{G}_1 \otimes \bar{H}$. In particular, ψ is false over $\bar{G}_2 \otimes \bar{H}$ 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 (4.6): $\left(\bar{G}_1[v = x_v] \otimes \bar{H}\right) \models \psi$, whilst $\left(\bar{G}_2[v = y_v] \otimes \bar{H}\right) \not\models \psi$ for all $y_v \in V(\bar{G}_2)$.

In search for a contradiction, we again 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 4.6) 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 (4.7), 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 3.4, easily conclude:

Corollary 4.8 (*cf.* [9,8]). *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 rank-width-t members of $\mathcal{F}$ is accepted by a finite tree automaton.*

Remark 4.9. Corollary 4.8 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 4.8) 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 [9] (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, the *Lin*EMSO language allows, in addition to ordinary MSO expressions, to compare between and optimize over linear evaluational terms.

We now briefly introduce the *Lin EMSO optimization problems* as given in [9]. Consider any MS_1 formula $\psi(X_1, \ldots, X_p)$ with free set variables, and state the following problem on an input graph G:

$$opt\ \left\{f_{lin}(U_1, \ldots, U_p) : U_1, \ldots, U_p \subseteq V(G),\ G \models \psi(U_1, \ldots, U_p)\right\}, \tag{4.10}$$

where opt can be min or max, and f_{lin} is a linear evaluational function. It is

$$f_{lin}(U_1, \ldots, U_p) = \sum_{i=1}^{p} \sum_{j=1}^{m} \left(a_{i,j} \cdot \sum_{x \in U_i} f_j(x) \right) \tag{4.11}$$

where m and $a_{i,j}$ are (integer) constants and f_j are (integer) weight functions on the vertices of G. Typically f_{lin} is just a cardinality function. Such as,

$$\psi = \iota(X) \equiv \forall v, w \left(v \notin X \vee w \notin X \vee \neg edge(v, w)\right) \quad \text{and} \quad \text{“}\max |X|\text{”}$$

describes the maximum independent set problem, or

$$\psi = \delta(X) \equiv \forall v \exists w \left[v \in X \vee \left(w \in X \wedge edge(v, w)\right)\right] \quad \text{and} \quad \text{“}\min |X|\text{”} \tag{4.12}$$

is the minimum dominating set problem. Further examples like minimum independent or connected dominating set problems are easily possible.

We can achieve an analogous solution to [9] in our framework directly using Theorem 4.2. 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 ψ, per each class of the extended canonical equivalence $\approx^{\sigma}_{\psi,t}$.

Theorem 4.13 (*cf. Courcelle et al. [9]*). *Assume G is an input graph of rank-width t, and T its given t-labeling parse tree. Then the Lin EMSO optimization problem (4.10) can be solved in linear time $O(|V(G)|)$ for fixed t.*

Proof. Let $\sigma = (\emptyset, \{X_1, \ldots, X_p\}, \emptyset)$. 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 $U_1, \ldots, U_p \subseteq V(G_x)$, the σ-partially equipped graph $\bar{G}_x$ with interpretation $X_i = U_i$, $i = 1, \ldots, p$ falls into one of the (fixed number) ℓ classes of $\approx^{\sigma}_{\psi,t}$ (Theorem 4.2). A dynamic algorithm for solving (4.10) has to remember just one representative interpretation $(U^j_1, \ldots, U^j_p)$ achieving maximum $f_{lin}(X_1, \ldots, X_p)$ over the jth class of $\approx^{\sigma}_{\psi,t}$, for $j = 1, 2, \ldots, \ell$. Thanks to linearity of the objective function (4.11), and with knowledge of the associated tree automaton (Remark 4.9), this information can easily be processed from leaves of T to the root in total linear time (t fixed). □

5. Extending the regularity framework

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 [11] defines a problem to be *fixed parameter tractable* (FPT) 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 2.3, Proposition 3.9 or Theorem 4.13 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 4.2 provides $f(k)$ as a tower of exponents generally growing with quantifier alternation in the formula, cf. (4.6) and (4.7). This is, indeed, generally unavoidable for results capturing all MSO (or even FO) properties, cf. [13].

Obviously, we can hardly expect f to be polynomial for any *NP*-complete problem, 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 for rank-width by Bui-Xuan, Telle and Vatshelle in [4]. They have provided three new explicit algorithms for the independent set, c-colourability, and dominating set problems which all run in time $O(2^{\Theta(t^2)}n)$ for graphs with rank-decompositions of width at most t. While the first two are comparable with our regularity framework of Section 3 (as we discuss in Section 6), the algorithm for the dominating set problem [4, Theorems 3.13, 3.14] is most interesting for us at this point.

Consider (4.12) the MS_1 predicate $\delta(X)$ stating that X is a dominating set in G, and write shortly (X) for the equipment signature $(\emptyset, \{X\}, \emptyset)$ of δ. Then the extended canonical equivalence of $\delta(X)$ simply has too many distinct classes, and so the approach of Theorem 4.13 cannot provide a parameterized algorithm with a single-exponential dependency on t:

Proposition 5.1. *The equivalence $\approx^{(X)}_{\delta(X),t}$ (Definition 4.1) has at least 2^{2^t-1} distinct classes in the universe of all (X)-partially equipped t-labeled graphs.*

Still, Bui-Xuan, Telle and Vatshelle have managed to overcome this difficulty with a new trick—simultaneously with the current fragment of a dominating set, one should record also an "expectation" of the rest of that dominating set.

As we present next, this clever idea has a very nice generalized formalization which (we suggest) has consequences reaching far beyond the scope of [4].

Prepartitioned canonical equivalence scheme

While introducing this new concept, which enhances canonical equivalences of Section 3, we remark that it is in no specific way tied with rank-width or labeling parse trees, and so it can be used on a more general level (basically all we need is a "good" notion of parse trees, a corresponding join operator, and an analogue of Proposition 5.7).

Informally, the purpose of a prepartitioned canonical equivalence scheme is to provide a general formalism for capturing the above mentioned "expectation" of information which is not accessible yet. This is achieved by prepartitioning our universe of graphs in advance, and then restricting the scope of a canonical equivalence (the graph $\bar{H}$ displayed in Definition 4.1) to each part. We advise the reader to compare the coming technical definition with its applications in Section 6.

We consider the universe $\mathcal{U}_t^\sigma$ of all σ-partially equipped t-labeled graphs where $\sigma = (Fr, FR, q)$ is a fixed equipment signature, i.e. we allow interpretations of free element variables from Fr and free set variables from FR along with our graphs. Let π be a graph property (a predicate) with free variables from $Fr \cup FR$. More precisely, π is a property of σ-partially equipped graphs. Assume that $\mathcal{B}_t$ is, for any natural number t, an arbitrary partition of $\mathcal{U}_t^{\sigma^-}$ into nonempty parts, and for any $B \in \mathcal{B}_t$ let $\mathcal{A}_t^B$ be any partition of $\mathcal{U}_t^\sigma$. We say that a part $B' \in \mathcal{B}_t$ is *stronger* than $B \in \mathcal{B}_t$ if the partition $\mathcal{A}_t^{B'}$ is a refinement of $\mathcal{A}_t^B$ (notice that this ordering is reflexive).

Definition 5.2. A property π has a *prepartitioned canonical equivalence scheme* (abbreviated as PCE scheme) if, for all integer t, there exist partitions $\mathcal{B}_t$ and $\mathcal{A}_t^B, B \in \mathcal{B}_t$, of $\mathcal{U}_t^\sigma$ as above such that the following points are satisfied:

(i) Consider any t-labeling composition operator $\widehat{\otimes}$. For any $B, B' \in \mathcal{B}_t$ and every choice of $\bar{F}, \bar{G} \in B$ and $\bar{F}', \bar{G}' \in B'$, the graphs $\bar{F} \widehat{\otimes} \bar{F}'$ and $\bar{G} \widehat{\otimes} \bar{G}'$ belong to the same class $B_0 \in \mathcal{B}_t$. Furthermore, B_0 is stronger than B, B'.

(ii) Consider any $B_1, B_2, B_3 \in \mathcal{B}_t$ where B_1, B_2 are both stronger than B_3, any $A_1 \in \mathcal{A}_t^{B_1}$ and $A_2 \in \mathcal{A}_t^{B_2}$, and any t-labeling composition operator $\widehat{\otimes}$. For every $\bar{F}_1, \bar{G}_1 \in A_1$ and every $\bar{F}_2, \bar{G}_2 \in A_2$, the graphs $\bar{F}_1 \widehat{\otimes} \bar{F}_2$ and $\bar{G}_1 \widehat{\otimes} \bar{G}_2$ belong to the same class $A_3 \in \mathcal{A}_t^{B_3}$.

(iii) There is a constant d independent of t such that the following equivalence $\sim_\pi^{A,B}$ on A has index at most d for all choices of $B \in \mathcal{B}_t$ and $A \in \mathcal{A}_t^B$. It is $\bar{G}_1 \sim_\pi^{A,B} \bar{G}_2$ if and only if $\bar{G}_1, \bar{G}_2 \in A$ and
$$\bar{G}_1 \otimes \bar{H} \models \pi \iff \bar{G}_2 \otimes \bar{H} \models \pi \quad \text{for all } \bar{H} \in B.$$

We believe this complicated definition deserves a very informal explanation now. In the run of a dynamic algorithm, one faces input data which have already been read and processed, and remaining data which are to be accessed in future. The parts of $\mathcal{B}_t$ in a PCE scheme record our "expectation" of the remaining (future) data, and relatively to a particular $B \in \mathcal{B}_t$, the parts of $\mathcal{A}_t^B$ classify the information we remember about the processed data. Part (i) of the definition then states that our "expectations" are consistent with the composition operators we find in our parse trees. Part (ii) states that also the information we remember in $\mathcal{A}_t^B$ is consistent with the compositions, provided that our expectations are sound. That is also the only place where we use the property of stronger parts which restricts our requirements on a PCE scheme to necessary minimum. Finally, part (iii) determines that only bounded information (fixed number d of states) about the property π has to be kept in addition to our knowledge of the pairs B and $A \in \mathcal{A}_t^B$. This d is usually very small, like 1 or 2.

Definition 5.2 clearly extends the definition of a canonical equivalence $\approx_{\pi,t}^\sigma$: set trivially $\mathcal{B}_t = \{\mathcal{U}_t^{\sigma^-}\}$, and $\mathcal{A}_t$ equal to the classes of $\approx_{\pi,t}^\sigma$. Then (iii) $d = 1$.

PCE scheme and dynamic algorithms

The reason for using a PCE scheme is that in many cases (problems) we can get the partitions $\mathcal{B}_t$ and $\mathcal{A}_t^B$ with numbers of classes much smaller than the index of the associated ordinary canonical equivalence (for instance Proposition 5.1). Therefore we can give dynamic algorithms for such problems whose runtime has asymptotically much smaller dependency on t than, say, those coming from Theorem 4.13.

Actually, to avoid disturbing technical difficulties with handling a σ^--partial equipment of free element variables (cf. Section 4), we restrict our attention to the easier case of $Fr = \emptyset$, i.e. $\sigma = (\emptyset, FR, \emptyset) = \sigma^-$. That means we are going to handle labeled graphs $\mathcal{U}_t^\sigma$ which are all σ-partially equipped with interpretations of the free set variables from FR (which is enough for *Lin*EMSO optimization problems), but we remark our concept is extendable to the general case of nonempty Fr. The core new outcome of this concept is the following:

Theorem 5.3. *Let $\pi(X_1, \ldots, X_p)$ be a graph property with free set variables. Assume π has a PCE scheme consisting of partitions $\mathcal{B}_t$ and $\mathcal{A}_t^B, B \in \mathcal{B}_t$, such that each $\mathcal{A}_t^B$ is a refinement of $\mathcal{B}_t$, and denote by $b(t) = |\mathcal{B}_t|$ and $a(t) = \max\{|\mathcal{A}_t^B| : B \in \mathcal{B}_t\}$. Then any Lin EMSO optimization problem (4.10) defined via the formula $\psi = \pi$ is solvable on graphs G of rank-width $\le t$ with a given t-labeling parse tree T in time*
$$O\big(a(t)^2 \cdot b(t)^2 \cdot (c(t) + d(t)) \cdot |V(G)|\big),$$
where $c(t)$ is time needed to determine the class A_3 defined in Definition 5.2 (ii) from known B_3, A_1 and A_2, and $d(t)$ is time needed to determine the class of $\sim_\pi^{A,B}$ in Definition 5.2 (iii) to which a graph $\bar{G} \in A \subseteq \mathcal{U}_t^\sigma$ belongs.

Remark 5.4. The words "determine the class A_3 from known B_3, A_1 and A_2" should be made very clear this time. Imagine we have an "indexing" scheme for the classes of $\mathcal{B}_t$ (of $\mathcal{A}_t^B$), i.e. an assignment of the natural numbers 1, 2, ... to the classes. Then the task is to find the index of A_3 from known indices of B_3, A_1 and A_2. Since Theorem 5.3 does care about runtime dependency on t, this task is not simply a "constant operation" like in Theorem 4.13.

Proof. Let $FR = \{X_1, \ldots, X_p\}$ and $\sigma = (\emptyset, FR, \emptyset)$ be the equipment signature of π. We use a notation $w : FR \to 2^{V(G)}$ for the σ-equipment interpreting X_i as $w(X_1)$ in G, and explicitly denote such a σ-equipped graph by (G, w). Recall (4.11) the linear objective function f_{lin} of our *Lin*EMSO optimization problem. For $(\bar{G}, w) \in \mathcal{U}_t^\sigma$, we shortly write $f_{lin}(\bar{G}, w) = f_{lin}\big(w(X_1), \ldots, w(X_p)\big)$.

Our algorithm parses T in the leaves-to-root direction. At each node x of T, we remember the following information: For every $B \in \mathcal{B}_t$ and every corresponding $A \in \mathcal{A}_t^B$, and for each class D of $\sim_\pi^{A,B}$ from Definition 5.2(iii), we record (if it exists) a representative interpretation $w_x^o[A, B, D] = w_x$ which attains optimal value $f_{lin}(\bar{G}_x, w_x)$ of the objective function over all possible σ-equipments $(\bar{G}_x, w) \in D$ of the graph $\bar{G}_x$ (the subgraph parsed by the subtree below x). This is trivial at the leaves.

How is this $w_x^o[\,]$ updated at the internal nodes of T? We suppose that a node x has left son y and right son z, and carries a composition operator $\widehat{\otimes}$. We loop through all $B_x \in \mathcal{B}_t$, all $B_y \in \mathcal{B}_t$ and $A_y \in \mathcal{A}_t^{B_y}$, and all $B_z \in \mathcal{B}_t$ and $A_z \in \mathcal{A}_t^{B_z}$.

At the beginning of each iteration, we verify the "consistency of expectations" condition

$$B_y = A_z \langle\otimes B_x \quad \text{and} \quad B_z = B_x \otimes\rangle A_y, \tag{5.5}$$

which has the following formal meaning: Proposition 5.7 defines the operators $\langle\otimes$ and $\otimes\rangle$ associated with $\widehat{\otimes}$. From Definition 5.2(i) we know that the composition of two $\mathcal{B}_t$-parts is well defined, and each of A_y, A_z is a subset of a unique $\mathcal{B}_t$-part ($\mathcal{A}_t^B$ is a refinement of $\mathcal{B}_t$) to which the composition is applied. The intuition behind (5.5) is that the expectation we work with at the son y is a combination of the expectation at its parent x and the real data coming from its sibling z.

Only if (5.5) holds true, we continue with the iteration. We determine the unique $A_3 = A_x \in \mathcal{A}_t^{B_x}$ from Definition 5.2(ii) where we choose $A_1 = A_y$ and $A_2 = A_z$. Notice that we have $\bar{G}_x = \bar{G}_y \widehat{\otimes} \bar{G}_z$ by the definition of a parse tree, and that for any σ-equipments w_y, w_z, it is $f_{lin}(\bar{G}_x, w) = f_{lin}(\bar{G}_y, w_y) + f_{lin}(\bar{G}_z, w_z)$ thanks to linearity of the objective function, where $w(X_i) = w_y(X_i) \cup w_z(X_i)$. For each of the (fixed number d of) classes D_y of $\sim_\pi^{A_y,B_y}$ and each D_z of $\sim_\pi^{A_z,B_z}$, there is a unique class D_x of $\sim_\pi^{A_x,B_x}$ to which this $(\bar{G}_x, w)$ belongs, cf. Definition 5.2(iii). Thus, we now look at $w_y = w_y^o[A_y, B_y, D_y]$ and $w_z = w_z^o[A_z, B_z, D_z]$: If $f_{lin}(\bar{G}_y, w_y) + f_{lin}(\bar{G}_z, w_z)$ is better than $f_{lin}(\bar{G}_x, w_x^o[A_x, B_x, D_x])$ (or if the record does not exist yet), then we store $w_x^o[A_x, B_x, D_x] = w$.

Finally, at the root r of T, we simply check all the recorded representatives $w_r^o[A, B_0, D]$, where B_0 is the class to which the empty graph belongs, for a globally optimal true answer to our problem.

At every iteration of the above defined loops, we do a finite number of operations among which only three have runtime depending on t—checking (5.5), computing A_x, and determining D_x. The latter two can be done in time $c(t) + d(t)$ by the assumptions. Interestingly, also (say) $A_z\langle\otimes B_x$ can be computed in time $c(t)$: Set $B_1 = B_2 = B_3 = B_x$, and $A_1 = A_z$ and A_2 be any $\mathcal{A}_t^{B_z}$-part contained in B_x. Then compute A_3 of Definition 5.2(ii) and check that $A_3 \subseteq B_y$ (which is correct thanks to Definition 5.2(i)). Hence any iteration takes $O(c(t) + d(t))$.

For each of the $O(|V(G)|)$ nodes of T, we execute 5 nested loops (as above), but, actually, we can save one. After selecting B_x, B_y, and A_y, the next B_z can be determined from (5.5) as in the previous paragraph. Hence we have to do only $a(t)^2 \cdot b(t)^2$ iterations. The runtime bound follows.

We finish the proof by showing that our algorithm computes correctly. That includes two tasks. First, for a node x of T, let $\bar{G}_x$ be the subgraph parsed by the subtree below x, and $\bar{G}_x^-$ be such that $\bar{G}_x \otimes \bar{G}_x^- = G$ (which can be constructed by Lemma 3.6). We claim that if $B \in \mathcal{B}_t$ is such that $(\bar{G}_x^-, w^-) \in B$ for some σ-equipment w^- of $\bar{G}_x^-$, then for any recorded interpretation $w_x^o[A, B, D] = w_x$ at the node x, the following is true: There exists a σ-equipment w for the whole graph $\bar{G}$ such that w_x is the restriction of w to $\bar{G}_x$ and w_x^- is the restriction to $\bar{G}_x^-$, and $(\bar{G}_x, w_x) \in D \subseteq A$. That holds true at the leaves, and carries up the tree T inductively by (5.5). Notice that at the root $x = r$, it is $\bar{G}_r^- = \emptyset \in B_0$, and so the solution found in the final stage of the algorithm is admissible.

Second, we show that if there exists a σ-equipment w for $\bar{G}_x$ such that $(\bar{G}_x, w) \in D$, then $f_{lin}(\bar{G}_x, w_x^o[A, B, D]) \geq f_{lin}(\bar{G}_x, w)$ (assuming that the objective function is maximized). As $\bar{G}_x = \bar{G}_y \widehat{\otimes} \bar{G}_z$, we have $w(X_i) = w_y(X_i) \cup w_z(X_i)$ (a disjoint union) for all $X_i \in FR$. With help of Lemma 3.6 on T at $v = x$, we see that there are unique expectations $B_y, B_z \in \mathcal{B}_t$ for this B and w, and then uniquely $(\bar{G}_y, w_y) \in D_y \subseteq A_y \in \mathcal{A}_t^{B_y}$ and $(\bar{G}_z, w_z) \in D_z \subseteq A_z \in \mathcal{A}_t^{B_z}$. (5.5) holds true at this point. By induction on the depth of the parse tree we can assume that $f_{lin}(\bar{G}_y, w_y) \leq f_{lin}(\bar{G}_y, w_y^o[A_y, B_y, D_y])$, and analogously for z. Hence

$$\begin{aligned} f_{lin}(\bar{G}_x, w) &= f_{lin}(\bar{G}_y, w_y) + f_{lin}(\bar{G}_z, w_z) \\ &\leq f_{lin}(\bar{G}_y, w_y^o[A_y, B_y, D_y]) + f_{lin}(\bar{G}_z, w_z^o[A_z, B_z, D_z]) \leq f_{lin}(\bar{G}_x, w_x^o[A, B, D]), \end{aligned}$$

where the last step holds after the respective iteration of our algorithm. $\square$

Corollary 5.6. *In the setting of Theorem 5.3, assume that the partitions $\mathcal{A}_t^B = \mathcal{A}_t$ are the same for all $B \in \mathcal{B}_t$. Then the runtime bound of Theorem 5.3 can be improved to $O\big(a(t)^2 \cdot b(t) \cdot c(t) \cdot |V(G)|\big)$.*

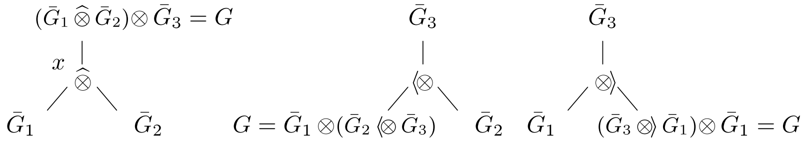
Proof. In this special case, we can select the parts $A_1, A_2 \in \mathcal{A}_t$ prior to considering B_1 and B_2, and then B_1, B_2 are uniquely determined by (5.5). $\square$

The following technical property is needed in the proof of Theorem 5.3.

Proposition 5.7. *Let $\widehat{\otimes} = \otimes[g \mid f_1, f_2]$ be a t-labeling composition operator. Then there exist t-labeling composition operators $\langle\otimes$ and $\otimes\rangle$ such that, for all t-labeled graphs $\bar{G}_1, \bar{G}_2, \bar{G}_3$, it is*

$$(\bar{G}_1 \widehat{\otimes} \bar{G}_2) \otimes \bar{G}_3 = (\bar{G}_2 \langle\otimes \bar{G}_3) \otimes \bar{G}_1 = (\bar{G}_3 \otimes\rangle \bar{G}_1) \otimes \bar{G}_2.$$

This natural statement can be perhaps better understood in the following scheme, in which we display a node x of a parse tree generating the graph G, and claim that any two branches at x can be composed together before the third one is joined, to generate the same (unlabeled) graph G.



Proof. We simply set $\langle\otimes = \otimes[f_2^T \mid g, f_1^T]$ and $\otimes\rangle = \otimes[f_1^T \mid f_2^T, g^T]$. $\square$

6. Applications in FPT algorithms

In this section we focus on several particular algorithmic problems on which we illustrate the use of our formal tools from previous sections to design parameterized algorithms of practically reasonable runtime. We take full advantage of our dual view of labelings of a graph G, on one hand as $lab : V(G) \to 2^{L_t}$ where $L_t = \{1, 2, \ldots, t\}$, and on the other hand as a mapping $lab : V(G) \to \mathrm{GF}(2)^t$ into a binary vector space (an edge $\{u, v\}$ is added in the join $\otimes$ iff $lab(u) \cdot lab(v) = 1$), cf. Section 2. This view, for instance, allows much easier handling of the situation using linear algebra tools.

One of the tools is the following classical result.

Proposition 6.1 ([18]). *The number $S(t)$ of subspaces of binary vector space $\mathrm{GF}(2)^t$ is at most $2^{t(t+1)/4} - 2$ for $t \geq 12$.*

Proof. Goldman and Rota gave [18] the exact recurrence $S(t+1) = 2S(t) + (2^t - 1)S(t-1)$. From that we routinely get $S(t) \leq 2^{t(t+1)/4} - 2$ for $t \geq 12$. $\square$

We remark that the "−2" term in this estimate is rather random—the bound works, and it is better suited for Lemma 6.2. Another potential issue in applications can be the condition $t \geq 12$, but that is "hidden" in the $O(\ldots)$ notation further on. We also have an alternative proof (without using [18]) giving a universal bound $S(t) \leq 2^{t(t+4)/4}$ with elementary tools of linear algebra.

Recall that $\iota(X)$ is the predicate stating that X (the interpretation of it) is an independent set in a graph G. We analogously define $\gamma(X)$ stating that X is a clique in G. The important relation between independence of a vertex set in a graph and the vector subspace generated by this set has been first given by Bui-Xuan, Telle and Vatshelle in [4]. We restate and extend their findings in the next lemma.

Lemma 6.2 (*cf. [4, Proposition 3.6] for part (a)*). *Consider the universe of (X)-equipped t-labeled graphs. Then the number of classes of the extended canonical equivalence (Definition 4.1)*

(a) $\approx^{(X)}_{\iota,t}$ *for $\iota(X)$ (independent set) is at most $1 + S(t)$, and*
(b) $\approx^{(X)}_{\gamma,t}$ *for $\gamma(X)$ (clique) is at most $2 + S(t+1)$.*

Proof. (a) For a subspace Σ of $\mathrm{GF}(2)^t$, we define P_Σ as the class of all those (X)-equipped t-labeled graphs $\bar{G} = (G, lab)$ such that $G \restriction X$ is independent, and that the vectors of $lab(X)$ generate the space Σ. Let $\bar{G}_1, \bar{G}_2 \in P_\Sigma$ with interpretations $X = U_1$ and $X = U_2$, respectively. Assume (Definition 4.1) that $\bar{G}_1 \otimes \bar{H} \models \iota(U_1 \cup W)$ where $X = W$ in $\bar{H}$. Then both $G_1 \restriction U_1$ and $H \restriction W$ are independent. Moreover, $G_2 \restriction U_2$ is independent since $\bar{G}_2 \in P_\Sigma$.

If $\bar{G}_2 \otimes \bar{H} \not\models \iota(U_2 \cup W)$, then there would be an edge between $v \in W$ and some vertex of U_2. So there is a vector $\alpha \in \Sigma$ such that $\alpha \cdot lab(v) = 1$ over $\mathrm{GF}(2)$. However, $\alpha \in \Sigma$ is generated as a sum $\alpha = \alpha_1 + \cdots + \alpha_c$ where $\alpha_j = lab(u_j)$ for some $u_j \in U_1, j = 1, \ldots, c$, and $\alpha_j \cdot lab(v) = 0$ by the assumption. Hence also $\alpha \cdot lab(v) = 0$, a contradiction. The conclusion is that our parts P_Σ, together with one "leftover" part, refine the classes of $\approx^{(X)}_{\iota,t}$, and (a) follows.

(b) For any t-labeled graph $\bar{G} = (G, lab)$, we define a $(t+1)$-labeled graph $\bar{G}^+ = (G, lab^+)$ such that $lab^+(v) = lab(v) \cup \{t+1\}$. Let Σ_X denote the subspace of $\mathrm{GF}(2)^t$ generated by the labelings $lab(X)$, and Σ_X^+ the subspace of $\mathrm{GF}(2)^{t+1}$ generated by $lab^+(X)$. We consequently define Q_Σ to be the class of all those (X)-equipped t-labeled graphs $\bar{G} = (G, lab)$ such that $G \restriction X$ is a clique, and that $\Sigma = \Sigma_X^+$ has the same dimension as Σ_X. Furthermore, we define Q_o as the class of all $\bar{G} = (G, lab)$ such that $G \restriction X$ is a clique, and that $\Sigma = \Sigma_X^+$ has higher dimension than Σ_X. We again claim that these parts Q_Σ, the part Q_o, and the "leftover" part, refine the classes of $\approx^{(X)}_{\gamma,t}$.

Our claim is supported by the following two facts. First, assume for our (X)-equipped $\bar{G} \in Q_o$ that Σ_X^+ has higher dimension than Σ_X, and still, $\bar{G} \otimes \bar{H} \models \gamma$ where $\bar{H}$ has a nonempty interpretation of X. Then there exist vertices $u_1, \ldots, u_p \in U$, where U is the interpretation of X in $\bar{G}$, such that $lab(u_1) + \cdots + lab(u_p) = 0$ but $lab^+(u_1) + \cdots + lab^+(u_p) = \alpha \neq 0$. Since it must be $\alpha = (0, \ldots, 0, 1)$, we have p odd. If $\bar{G} \otimes \bar{H} \models \gamma$ and β is the labeling of some vertex of X in $\bar{H}$, then $\beta \cdot (lab(u_1) + \cdots + lab(u_p)) = \beta \cdot 0 = 0$, but at the same time $\beta \cdot lab(u_1) + \cdots + \beta \cdot lab(u_p) = 1 + \cdots + 1 = 1$, a contradiction. Hence all of Q_o belong to one canonical class.

Second, if Σ_X^+ has the same dimension as Σ_X, then we can use a simple algebraic observation: An edge $\{u, v\}$, where $u \in V(G)$ and $v \in V(H)$, is created in the join $\bar{G} \otimes \bar{H}$ if and only if no edge $\{u, v\}$ is created in the join $\bar{G}^+ \otimes \bar{H}^+$. Hence in this case the partition of graphs $\bar{G}$ into Q_Σ (where $\Sigma = \Sigma_X^+$ for a particular (X)-equipped $\bar{G}$) refines the relevant canonical classes of $\approx_{\gamma,t}^{(X)}$ for the same reasons as in (a). □

Independent set, c-colourability, and extensions

As we have already briefly mentioned, Bui-Xuan, Telle and Vatshelle gave in [4, Theorem 3.10] an FPT algorithm computing the maximum independent set in a graph G with a rank-decomposition of width t in single-exponential time $O\big(2^{t(t+9)/2} \cdot t^2 \cdot |V(G)|\big)$ (this expression is translated from their "R_t-joins" to our notation).

By a combination of Lemma 6.2, Proposition 6.1, and the procedure of Theorem 4.13, we can immediately get a similar FPT algorithm for this problem with runtime $O\big(2^{t(t+1)/2} \cdot t^3 \cdot |V(G)|\big)$, where time $O(t^3)$ is needed to compute the closure of two subspaces. The slight improvement in our runtime bound has two sources; a finer analysis of $S(t)$ in Proposition 6.1, and labeling parse trees which better suit this specific algorithmic purpose.

To be very precise with the runtime bounds of both previous independent set algorithms, we should note that some time amount depending only on t is needed to build an indexing data structure for all the subspaces of $\mathrm{GF}(2)^t$. This is specified in the next claim.

Lemma 6.3. *There exists an indexing structure which allows to determine the index of a subspace Σ of $\mathrm{GF}(2)^t$ from a given set of generators in time $O(t^3)$. This structure can be built in time $O\big(2^{3t(t+1)/4} \cdot t^3\big)$.*

Proof. We build an indexing structure consisting of all $2^{t(t+1)/2}$ upper-triangular binary matrices—potential generator sets of all the subspaces of $\mathrm{GF}(2)^t$. We let each matrix refer to the first one in the list which generates the same subspace, using Gaussian elimination in time $O(t^3)$. Even by brute force this all takes time $O\big(t^3\, 2^{t(t+1)/2}\, S(t)\big)$. One access to this structure then consists of Gaussian elimination of the generator set to an upper-triangular matrix. □

It is easy to extend an independent set algorithm into one for the c-colourability problem (with fixed c). The corresponding extension by Bui-Xuan, Telle and Vatshelle in [4, Theorem 3.11] runs in time $O\big(2^{ct(t+5)/2+2t} \cdot t^{2c} \cdot |V(G)|\big)$.

Since c-colourability of a graph G means decomposability of G into c independent sets, we consider the predicate $\iota\iota(X_1, \dots, X_c)$ stating that all $X_1, \dots, X_c$ are independent in G. We apply Claim (4.5) to show that the canonical equivalence of $\iota\iota$ has at most $(1+S(t))^c$ classes, and Theorem 4.2 to prove that the MS_1 property $\tau(X_1, \dots, X_c) \equiv$ "$X_1, \dots, X_c$ is a partition of $V(G)$" has a constant number of canonical classes. In this way we can get an FPT algorithm solving c-colourability in time $O\big(2^{ct(t+1)/2} \cdot ct^3 \cdot |V(G)|\big)$. Again, we have implicitly used Lemma 6.3 for indexing the canonical classes via subspaces.

So far, we have only used part (a) of Lemma 6.2 (which seems to be a much more frequent case), but new part (b) is also useful in solving some problems. To illustrate this, we will present one new FPT algorithm for the so-called co-colouring problem (which is NP-complete in general).

A graph G is *c-co-colourable* if its vertex set can be partitioned into c parts such that each part is independent or a clique. An XP algorithm, i.e. one running in time $O(n^{f(t)})$ for c on the input, has been given for this problem in [24]. We, on the other hand, present an FPT algorithm taking c as the second parameter here. (So these two are uncomparable results.)

Theorem 6.4. *Assume G is an input graph of rank-width t, and T its given t-labeling parse tree. Then there is an FPT algorithm deciding whether G is c-co-colourable in time $O\big(2^{c(t+1)(t+2)} \cdot ct^3 \cdot |V(G)|\big)$.*

Proof. Recalling the predicates $\iota(X)$ and $\gamma(X)$ for the independent set and the clique X, respectively, and the predicate $\tau(X_1, \dots, X_c)$ expressing a vertex partition, we can write $\psi(X_1, \dots, X_c) \equiv \tau(X_1, \dots, X_c) \wedge \bigwedge_{i=1}^{c}\big(\iota(X_j) \vee \gamma(X_j)\big)$ to describe a co-colouring partition of a graph G. Let $\sigma = (\emptyset, \{X_1, \dots, X_c\}, \emptyset)$. Now the canonical partition of τ has a finite index p independent of t, and hence applying Lemma 6.2 and Claim (4.5) we get that the extended canonical equivalence $\approx_{\psi,t}^{\sigma}$ has

$$\begin{aligned} q(t) &\le p \cdot [(1+S(t))(2+S(t+1))]^c \le p \cdot (2+S(t+1))^{2c} \\ &\le p \cdot \left(2^{(t+1)(t+2)/4}\right)^{2c} = p \cdot 2^{c(t+1)(t+2)/2} \end{aligned}$$

equivalence classes.

We now apply Theorem 4.13. Although we do not have an optimization problem, we can decide the existence of $U_1, \dots, U_c$ such that $G \models \psi(U_1, \dots, U_c)$ using any, even constant, objective function in (4.10). The runtime of this algorithm is linear in $|V(G)|$, but what is the precise dependence on t? The finite tree automaton $\mathcal{A}$ associated with $\approx_{\psi,t}^{\sigma}$ has $q(t)$ states. At each node of the parse tree, we have to combine the optimal representatives of all the $\mathcal{A}$-states from the left subtree with all the $\mathcal{A}$-states from the right subtree, and the transition function of $\mathcal{A}$ can be computed in time $O(t^3)$ as $2c$ joins of pairs of subspaces of $\mathrm{GF}(t)^{t+1}$. Thus our algorithm runs in time $O\big(q(t)^2 \cdot ct^3 \cdot |V(G)|\big)$ as claimed. □

Dominating set

The single-exponential FPT algorithm for computing the minimum dominating set in a graph G with a rank-decomposition of width t, as given by Bui-Xuan, Telle and Vatshelle in [4, Theorem 3.14], has runtime $O\big(2^{3t(t+5)/4+2t} \cdot t^3 \cdot |V(G)|\big)$. As mentioned before, we have studied and generalized its core idea in Section 5; and now we show how this dominating set algorithm easily fits back into our PCE scheme formalism:

Let $\sigma = (\emptyset, \{X\}, \emptyset)$, and $\delta(X)$ be the predicate stating that X is a dominating set in the graph. Let $\mathcal{P}_t$ be the partition of all σ-equipped t-labeled graphs such that $\bar{G}$ belongs to $P_\Sigma \in \mathcal{P}_t$ if and only if the labelings of the interpretation of X generate the subspace Σ of $\mathrm{GF}(2)^t$. In Definition 5.2 we simply set $\mathcal{B}_t = \mathcal{A}_t^B = \mathcal{P}_t$. Then the points (i) and (ii) are easily satisfied. We have to verify (iii), say for $A = P_\Sigma$ and $B = P_{\Sigma'}$.

We consider $\bar{G} \in A$ and $\bar{H} \in B$, with interpretations of X as U and W, respectively. Repeating the arguments of Lemma 6.2(a), we claim that it is enough to know the space Σ' to decide whether an arbitrary vertex u of $\bar{G}$ is adjacent to at least one vertex from W in the join $\bar{G} \otimes \bar{H}$—we shortly say in such situation that $lab(u)$ is adjacent to Σ'. Hence there are precisely two classes of the equivalence $\sim_\delta^{A,B}$ from Definition 5.2(iii); the one containing all graphs $\bar{G} \in A$ such that every vertex u of $\bar{G}$ not dominated by U has $lab(u)$ adjacent to Σ', and the other one containing the rest.

By Corollary 5.6, the minimum dominating set problem can now be solved in time $O\big(2^{3t(t+1)/4} \cdot t^3 \cdot |V(G)|\big)$. Furthermore, we can easily extend this algorithm to solve the minimum independent dominating set problem, for instance.

Acyclic colouring

Finally, we are going to illustrate the full strength of PCE schemes and Theorem 5.3 on the example of acyclic colourability. A graph colouring is *acyclic* if no cycle of the graph has only two colours.

Theorem 6.5. *Assume G is an input graph of rank-width t, and T its given t-labeling parse tree. Then there is an FPT algorithm deciding whether G has an acyclic c-colouring in time $O\big(2^{5c^2t^2} \cdot c^2t^3 \cdot |V(G)|\big)$.*

Proof. We can describe an acyclic colour c-partition of a graph with the following predicate $\psi(X_1, \dots, X_c) \equiv \tau(X_1, \dots, X_c) \wedge \bigwedge_{i=1}^c \iota(X_i) \wedge \bigwedge_{i,j=1}^c \lambda(X_i, X_j)$, where $\lambda(X_1, X_2)$ means that $X_1 \cup X_2$ induces an *acyclic* subgraph. Let $\sigma = (\emptyset, \{X_1, \dots, X_c\}, \emptyset)$. Our approach is generally analogous to the previously presented results, but much more technically complicated this time.

Assume we have got a PCE scheme (Definition 5.2) for $\lambda(X, Y)$ consisting of partitions $\mathcal{B}_t$ and $\mathcal{A}_t^B$, $B \in \mathcal{B}_t$, and denote by $b(t) = |\mathcal{B}_t|$ and $a(t) = \max\{|\mathcal{A}_t^B| : B \in \mathcal{B}_t\}$. We construct $\binom{c}{2}$ isomorphic copies $(\mathcal{B}_t)_{i,j}$ and $(\mathcal{A}_t^B)_{i,j}$ of these $\mathcal{B}_t$ and $\mathcal{A}_t^B$, for each choice of a variable pair $\{X, Y\} = \{X_i, X_j\}$ over the universe of all σ-equipped graphs. The intersections of these partitions $(\mathcal{B}_t)_+ = \bigcap_{i,j=1}^c (\mathcal{B}_t)_{i,j}$ and $(\mathcal{A}_t^B)_+ = \bigcap_{i,j=1}^c (\mathcal{A}_t^B)_{i,j}$ then again form a PCE scheme. In combination with at most $(1 + S(t))^c$ canonical equivalence classes of $\iota(X_1, \dots, X_c) \equiv \bigwedge_{i=1}^c \iota(X_i)$, we finally get an FPT algorithm solving acyclic c-colouring in time

$$O\left((1 + S(t))^{2c} \cdot (a(t)b(t))^{2c(c-1)/2} \cdot c^2t^3 \cdot |V(G)|\right) \tag{6.6}$$

(the details are analogous to the previous algorithms in this section).

Hence it remains to find our PCE scheme for above mentioned $\lambda(X, Y)$. For that we will need some technical results from linear algebra. Let $\mathcal{P}_t$ be again the partition of all (X, Y)-equipped t-labeled graphs such that $\bar{G} = (G, lab)$ belongs to $P_\Sigma \in \mathcal{P}_t$ if and only if the labelings $lab(U)$ of the interpretation $U \subseteq V(G)$ of $X \cup Y$ generate the subspace Σ of $\mathrm{GF}(2)^t$.

For a space $\Sigma \subseteq \mathrm{GF}(2)^t$, let $\tilde{\Sigma}$ denote a minimal subspace of $\mathrm{GF}(2)^t$ such that $\tilde{\Sigma}$ together with the space orthogonal to Σ generates whole $\mathrm{GF}(2)^t$. Beware that $\mathrm{GF}(2)^t$ contains self-orthogonal vectors, and so we cannot simply set $\tilde{\Sigma} = \Sigma$, but these two do have the same dimension. Every t-labeling $lab(v)$ of a vertex v in $\bar{G} = (G, lab)$ can be written as $q_1\alpha + q_2\beta$, $q_1, q_2 \in \{0, 1\}$, where α is a vector orthogonal to Σ and β is from $\tilde{\Sigma}$. We then define $lab'(v) = q_2\beta$. Notice that $u \in V(G)$ is adjacent to $v \in W \subseteq V(H)$ in $\bar{G} \otimes \bar{H}$, where $\bar{H} \in \mathcal{P}_\Sigma$ and W is the interpretation of $X \cup Y$, if and only if u, v are adjacent in $(G, lab') \otimes \bar{H}$.

We say that $U \subseteq V(G)$ is *light* with respect to Σ if $G \restriction U$ is a forest, and if the following are true for the t-labeled graph $G'_U = (G, lab') \restriction U$:

- at most $2t - 1$ distinct points of $\mathrm{GF}(2)^t \setminus \{\emptyset\}$ occur as labelings in G'_U,
- no component of G'_U contains two vertices $u \neq v$ with $lab'(u) = lab'(v)$,
- at most $t - 1$ components of G'_U contain two vertices $u \neq v$ with $lab'(u) \neq \emptyset \neq lab'(v)$, and the same label pair $\{lab'(u), lab'(v)\}$ occurs in that way in at most one component of G'_U.

(6.7) Assume $\bar{H} \in P_\Sigma$ with the interpretation $W \subseteq V(H)$ of $X \cup Y$, and $U \subseteq V(G)$ such that U is not light with respect to Σ. Then $U \cup W$ is not acyclic in $\bar{G} \otimes \bar{H}$.

We choose any $W_0 \subseteq W$ such that $lab(W_0)$ is a basis of Σ, and form a matrix $\boldsymbol{A}$ from these row vectors $lab(W_0)$. For any row basis $\boldsymbol{A}'$ of $\tilde{\Sigma}$ (defined above), the product $\boldsymbol{A}' \times \boldsymbol{A}^T$ is a square nonsingular matrix, and hence it has an inverse. We set $\boldsymbol{A}_1 = \left(\boldsymbol{A}' \times \boldsymbol{A}^T\right)^{-1} \times \boldsymbol{A}'$, and so $\boldsymbol{A}_1 \times \boldsymbol{A}^T = \boldsymbol{I}$. If U is not light with respect to Σ, then one of the three conditions is violated. First, if at least t of the labelings in G'_U are not from $\boldsymbol{A}_1$ and not $\emptyset$, then the corresponding vertices in G "connect" t pairs of vertices of W_0 where $|W_0| \leq t$, and so $U \cup W$ cannot be acyclic. Second, if $lab'(u) = lab'(v)$, then u and v are adjacent to the same (at least one) vertex in W_0, and that produces a cycle in their component. The third condition follows in the same way. (6.7) is proved.

To provide a PCE scheme for $\lambda(X, Y)$, we set $\mathcal{B}_t = \mathcal{P}_t$, and for $B = P_\Sigma \in \mathcal{P}_t$ we define a partition $\mathcal{A}_t^B$ as follows. First, we take the intersection of $\mathcal{P}_t$ with the class of all those (X, Y)-equipped graphs $\bar{G}$ such that the interpretation U of $X \cup Y$ is not light with respect to Σ. Second, for the remaining graphs $\bar{G}$ with light interpretation U of $X \cup Y$, we define the *U-trace* of $\bar{G}$ as follows: Let Σ_U be the subspace generated by $lab(U)$, $L = lab'(U) \setminus \{\emptyset\}$ where $|L| \leq 2t - 1$, and $\mathcal{M}$ be the multiset of all those labeling sets $lab'(P) \setminus \{\emptyset\}$ where P is the vertex set of a component of G'_U. The *U-trace* of $\bar{G}$ is the quintuple $(\Sigma_U, L, R, S, \mathcal{C})$ where $R \subseteq L$ are the labelings that occur as singleton sets in $\mathcal{M}$ and $S \subseteq R$ are those with multiple occurrence in $\mathcal{M}$, and $\mathcal{C}$

is the set of all the (at most $t-1$ by the definition of lightness) non-singleton members of $\mathcal{M}$, i.e. those in $\mathcal{M}$ of cardinality more than one. Then $\bar{G}_1$ and $\bar{G}_2$ belong to the same class of $\mathcal{A}_t^B$ if and only if their U-traces are equal.

Notice that $\mathcal{A}_t^B$ is a refinement of $\mathcal{B}_t$ (cf. the assumptions of Theorem 5.3). Verification of parts (i) and (ii) of Definition 5.2 is quite straightforward, and so we skip it here. We just observe that a part P_Σ is stronger than a part $P_{\Sigma'}$ if Σ' is a subspace of Σ in this case. Also part (iii) holds true in this setting, even with $d=1$, as it follows from (6.7) and the next claim.

(6.8) Assume $\bar{H} \in P_\Sigma$ with the interpretation $W \subseteq V(H)$ of $X \cup Y$, and $U \subseteq V(G)$ such that U is light with respect to Σ. Then it is enough to know the U-trace of $\bar{G}$ in order to decide whether $U \cup W$ is acyclic in $\bar{G} \otimes \bar{H}$.

Let H_U be the minor of $\bar{G}'_U \otimes \bar{H}$ obtained by contracting every component of $\bar{G}'_U$ into a single vertex. Clearly, the graph H_U is fully determined by $\bar{H}$ and the U-trace of $\bar{G}$, up to possible degree-1 vertices from U. On the other hand, if $\bar{G}'_U$ is a forest, then H_U determines whether $U \cup W$ is acyclic in $\bar{G} \otimes \bar{H}$. We have verified all the conditions we need in a PCE scheme for λ.

It remains to estimate the numbers of classes in the above PCE scheme. We have $b(t) = 1 + S(t)$ (Proposition 6.1), and $a(t) \le (1+S(t)) \cdot \binom{2^t}{2t-1} \cdot 3^{2t-1} \cdot \binom{2^{2t-1}}{t-1}$ where $\binom{2^t}{2t-1}$ bounds possible choices of L, 3^{2t-1} enumerates the choices of R and S, and $\binom{2^{2t-1}}{t-1}$ is a rough estimate of choices of $\mathcal{C}$. Altogether

$$a(t) \le (1+S(t)) \cdot 2^{2t^2-t} \cdot 3^{2t-1} \cdot 2^{2t^2-3t} \le (1+S(t)) \cdot 2^{4t^2}.$$

Hence from (6.6) we get a runtime bound

$$O\left((1+S(t))^{2c+2c(c-1)} \cdot 2^{4t^2c(c-1)} \cdot c^2t^3 \cdot |V(G)|\right) \le O\left((2^{t(t+1)/4})^{2c^2} \cdot 2^{4t^2c^2} \cdot c^2t^3 \cdot |V(G)|\right)$$
$$\le O\left(2^{(t^2/2)\cdot 2c^2} \cdot 2^{4c^2t^2} \cdot c^2t^3 \cdot |V(G)|\right) = O\left(2^{5c^2t^2} \cdot c^2t^3 \cdot |V(G)|\right). \quad \square$$

On side effect of the existence of a PCE scheme for the property λ is that we can now easily find the largest induced acyclic subgraph of a given graph of bounded rank-width. The set-complement of an induced acyclic subgraph is commonly called the feedback vertex set. Hence we get the following for free.

Theorem 6.9. *Assume G is an input graph of rank-width t, and T its given t-labeling parse tree. Then there is an FPT algorithm solving the feedback vertex set problem in time $O\big(2^{5t^2} \cdot t^3 \cdot |V(G)|\big)$.* $\square$

7. 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 to the theoretical computer science community and suitable for designing actual algorithms.

It is an interesting question (to which we do not have an answer right now) whether Theorems 4.2 and 5.3 could be used to give FPT algorithms for problems beyond the scope of the *Lin*EMSO properties [9] and of the vertex-partitioning framework [17,24]. We plan to aim our future research at more general theoretical questions rather than developing particular specialized algorithms. A sound suggestion for future studies would be, for instance, to try to identify a general class of problems within the *Lin*EMSO language for which there exist FPT algorithms with a single-exponential dependency on the rank-width parameter.

Acknowledgements

We thank the anonymous referees for many helpful comments which improved the presentation of our material. Both authors have been supported by the Research intent MSM0021622419 of the Czech Ministry of Education; Petr Hliněný has moreover been supported by the grant GAČR 201/08/0308 and Robert Ganian by the grant GAČR 201/09/J021.

References

[1] K.A. Abrahamson, M.R. Fellows, Finite automata, bounded treewidth, and well-quasiordering, in: Graph Structure Theory, Contemporary Mathematics 147, American Mathematical Society, 1993, pp. 539–564.
[2] S. Arnborg, J. Lagergren, D. Seese, Problems easy for tree-decomposible graphs, in: Proc. 15th Colloq. Automata, Languages and Programming, in: Lecture Notes in Comput. Sci., vol. 317, Springer, Berlin, 1988, pp. 38–51.
[3] H.-J. Bandelt, H.M. Mulder, Distance-hereditary graphs, J. Combin. Theory Ser. B 41 (2) (1986) 182–208.
[4] B.-M. Bui-Xuan, J.A. Telle, M. Vatshelle, *H*-join decomposable graphs and algorithms with runtime single exponential in rankwidth, Discrete Appl. Math. 158 (7) (2010) 809–819.
[5] D.G. Corneil, U. Rotics, 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., Inform. Comput. 85 (1990) 12–75.
[7] B. Courcelle, M.M. Kanté, Graph operations characterizing rank-width and balanced graph expressions, in: Graph-theoretic Concepts in Computer Science, in: Lecture Notes in Comput. Sci., vol. 4769, Springer, Berlin, 2007, pp. 66–75.

[8] B. Courcelle, M.M. Kanté, Graph operations characterizing rank-width, Discrete Appl. Math. 157 (4) (2009) 627–640.
[9] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, Theory Comput. Syst. 33 (2) (2000) 125–150.
[10] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, Discrete Appl. Math. 101 (1–3) (2000) 77–114.
[11] R.G. Downey, M.R. Fellows, Parameterized Complexity, in: Monographs in Computer Science, Springer-Verlag, New York, 1999.
[12] W. Espelage, F. Gurski, E. Wanke, How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time, in: Graph-theoretic Concepts in Computer Science, in: Lecture Notes in Comput. Sci., vol. 2204, Springer, Berlin, 2001, pp. 117–128.
[13] M. Frick, M. Grohe, The complexity of first-order and monadic second-order logic revisited, Ann. Pure Appl. Logic 130 (2004) 3–31.
[14] R. Ganian, Automata formalization for graphs of bounded rank-width. Master thesis. Faculty of Informatics of the Masaryk University, Brno, Czech republic, 2008.
[15] R. Ganian, P. Hliněný, Automata approach to graphs of bounded rank-width, in: IWOCA 2008, proceedings, 2008, pp. 4–15.
[16] R. Ganian, P. Hliněný, Better polynomial algorithms on graphs of bounded rank-width, in: IWOCA 2009, in: Lecture Notes in Comput. Sci. (in press).
[17] M.U. Gerber, D. Kobler, Algorithms for vertex-partitioning problems on graphs with fixed clique-width, Theoret. Comput. Sci. 299 (1-3) (2003) 719–734.
[18] J. Goldman, G.-C. Rota, The number of subspaces of a vector space, in: W.T. Tutte. (Ed.), Recent Progress in Combinatorics, Academic Press, 1969, pp. 75–83.
[19] P. Hliněný, Branch-width, parse trees, and monadic second-order logic for matroids, J. Combin. Theory Ser. B 96 (3) (2006) 325–351.
[20] P. Hliněný, S. Oum, Finding branch-decomposition and rank-decomposition, SIAM J. Comput. 38 (2008) 1012–1032.
[21] D. Kobler, U. Rotics, Edge dominating set and colorings on graphs with fixed clique-width, Discrete Appl. Math. 126 (2-3) (2003) 197–221.
[22] S. Oum, Rank-width and vertex-minors, J. Combin. Theory Ser. B 95 (1) (2005) 79–100.
[23] S. Oum, P. Seymour, Approximating clique-width and branch-width, J. Combin. Theory Ser. B 96 (4) (2006) 514–528.
[24] M. Rao, MSOL partitioning problems on graphs of bounded treewidth and clique-width, Theoret. Comput. Sci. 377 (2007) 260–267.
[25] N. Robertson, P. Seymour, Graph minors. X. Obstructions to tree-decomposition, J. Combin. Theory Ser. B 52 (2) (1991) 153–190.