First-Order Interpretations of Bounded Expansion Classes

JAKUB GAJARSKÝ and STEPHAN KREUTZER, Technical University Berlin, Berlin, Germany
JAROSLAV NEŠETŘIL, Charles University, Prague, Czech Republic
PATRICE OSSONA DE MENDEZ, CAMS (CNRS, UMR 8557), Paris, France, and Charles University, Prague, Czech Republic
MICHAŁ PILIPCZUK, University of Warsaw, Warsaw, Poland
SEBASTIAN SIEBERTZ, University of Bremen, Bremen, Germany
SZYMON TORUŃCZYK, University of Warsaw, Warsaw, Poland

The notion of bounded expansion captures uniform sparsity of graph classes and renders various algorithmic problems that are hard in general tractable. In particular, the model-checking problem for first-order logic is fixed-parameter tractable over such graph classes. With the aim of generalizing such results to dense graphs, we introduce classes of graphs with *structurally bounded expansion*, defined as first-order transductions of classes of bounded expansion. As a first step towards their algorithmic treatment, we provide their characterization analogous to the characterization of classes of bounded expansion via low treedepth covers (or colorings), replacing treedepth by its dense analogue called shrubdepth.

CCS Concepts: • **Theory of computation → Finite Model Theory**; **Graph algorithms analysis**; • **Mathematics of computing → Graph theory**;

Additional Key Words and Phrases: Sparse graph classes, bounded expansion, first-order logic, logical interpretations, model-checking

A preliminary version of this work appeared in the proceeding of ICALP 2018 [19]. This full version contains complete proofs of all the claimed results. This work was done while S. Siebertz was affiliated with the University of Warsaw. J. Nešetřil and P. Ossona de Mendez were supported by CE-ITI P202/12/G061 of GAČR, European Associated Laboratory (LEA STRUCO), and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Synergy Grant DYNASNET, grant agreement No 810115). J. Gajarský and S. Kreutzer were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Consolidator Grant DISTRUCT, grant agreement No 648527). M. Pilipczuk and S. Siebertz were supported by the National Science Centre of Poland (NCN) via POLONEZ grant agreement UMO-2015/19/P/ST6/03998, which has received funding from the European Union's Horizon 2020 research and innovation programme (Marie Skłodowska-Curie grant agreement No. 665778). Sz. Toruńczyk was supported by the NCN grant 2016/21/D/ST6/01485.

Authors' addresses: J. Gajarský, Technical University Berlin, Germany; email: jakub.gajarsky@tu-berlin.de; S. Kreutzer, Technical University Berlin, Germany; email: stephan.kreutzer@tu-berlin.de; J. Nešetřil, Charles University, Prague, Czech Republic; email: nesetril@kam.mff.cuni.cz; P. O. de Mendez, CAMS (CNRS, UMR 8557), Paris, France, and Charles University, Prague, Czech Republic; email: pom@ehess.fr; M. Pilipczuk, University of Warsaw, Poland; email: michal.pilipczuk@mimuw.edu.pl; S. Siebertz, University of Bremen, Germany; email: siebertz@uni-bremen.de; S. Toruńczyk, University of Warsaw, Poland; email: szymtor@mimuw.edu.pl.

1529-3785/2020/07-ART29 $15.00
https://doi.org/10.1145/3382093

ACM Reference format:
Jakub Gajarský, Stephan Kreutzer, Jaroslav Nešetřil, Patrice Ossona De Mendez, Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. 2020. First-Order Interpretations of Bounded Expansion Classes. *ACM Trans. Comput. Logic* 21, 4, Article 29 (July 2020), 41 pages.
https://doi.org/10.1145/3382093

1 INTRODUCTION

The interplay of methods from logic and graph theory has led to many important results in theoretical computer science, notably in algorithmics and complexity theory. The combination of logic and algorithmic graph theory is particularly fruitful in the area of *algorithmic meta-theorems*. Algorithmic meta-theorems are results of the form: *every computational problem definable in a logic $\mathcal{L}$ can be solved efficiently on any class of structures satisfying a property $\mathscr{P}$*. In other words, these theorems show that the *model-checking problem* for the logic $\mathcal{L}$ on any class $\mathscr{C}$ satisfying $\mathscr{P}$ can be solved efficiently, where *efficiency* usually means *fixed-parameter tractability*.

The archetypal example of an algorithmic meta-theorem is Courcelle's theorem [3, 4], which states that model-checking a formula φ of *monadic second-order logic* can be solved in time $f(\varphi) \cdot n$ on every graph with n vertices that comes from a fixed class of graphs with *bounded treewidth*, for some computable function f. Seese [38] proved an analogue of Courcelle's result for the model-checking problem of first-order logic on any class of graphs with bounded degree. Following this result, the complexity of first-order model-checking on specific classes of graphs has been studied extensively in the literature. See, e.g., [9–11, 13–16, 21, 24, 25, 26, 28–30, 38, 39]. One of the main goals of this line of research is to find a structural property $\mathscr{P}$ which precisely defines those graph classes $\mathscr{C}$ for which model checking of first-order logic is tractable.

So far, research on algorithmic meta-theorems has focused predominantly on *sparse* classes of graphs, such as classes with bounded *treewidth*, *excluding a minor*, with *bounded expansion* or *nowhere dense*. The concepts of *bounded expansion* and *nowhere denseness* were introduced by Nešetřil and Ossona de Mendez with the goal of capturing the intuitive notion of *sparse* graph classes. See [36] for an extensive cover of these notions. The large number of equivalent ways in which they can be defined using either notions from combinatorics, theoretical computer science, analysis, category theory, or logic, indicate that these two concepts capture some very natural limits of good behavior and algorithmic tractability. For instance, Grohe et al. [26] proved that if $\mathscr{C}$ is a class of graphs closed under taking subgraphs, then (under standard complexity theoretic assumptions) model-checking first-order logic on $\mathscr{C}$ is fixed-parameter tractable if and only if $\mathscr{C}$ is nowhere dense (the lower bound was proved in [13]). As far as fixed-parameter tractability of first-order model-checking is concerned, this result completely solves the case for graph classes that are *closed under taking subgraphs*, which is a reasonable requirement for sparse, but not for dense graph classes.

Consequently, research in this area has shifted towards studying the dense case, which is much less understood. While there are several examples of algorithmic meta-theorems on dense classes, such as for monadic second-order logic on classes with bounded *cliquewidth* [8] or for first-order logic on *interval graphs*, *partial orders*, classes with bounded *shrubdepth* and other classes, see e.g., [17], [18], [21], [22], a general theory of meta-theorems for dense classes is still missing. Moreover, unlike in the sparse case, there is no canonical hierarchy of dense graph classes similar to the sparse case, which could guide research on algorithmic meta-theorems in the dense world.

Hence, the main research challenge for dense model-checking is not only to prove tractability results but to also develop the necessary logical and algorithmic tools. It is at least as important to define and analyze promising candidates for "structurally simple" classes of graph classes that are not necessarily sparse. This is the main motivation for the research in this article. Since bounded expansion and nowhere denseness form natural limits in the sparse case, any extension of the theory should provide notions which collapse to bounded expansion or nowhere denseness, under the additional assumption that the classes are closed under taking subgraphs. Therefore, a natural way of seeking such notions is to base them on the existing notions with bounded expansion or nowhere denseness.

In this article, we take bounded expansion classes as a starting point and study two different ways of generalizing them towards dense graph classes preserving their good properties. In particular, we define and analyze classes of graphs obtained from bounded expansion classes by means of first-order transductions, and classes of graphs obtained by generalizing another, more combinatorial characterization of bounded expansion in terms of low treedepth colorings into the dense world.

Our main structural result shows that these two very different ways of generalizing bounded expansion into the dense setting lead to the same classes of graphs. This is explained in greater detail below.

Interpretations, Transductions, and Lifts. One possible way of constructing structurally simple classes of graphs is to use logical *interpretations*, or the related concept of *transductions* studied in formal language and automata theory. In this article, we consider first-order transductions that are mild extensions of first-order interpretations. A *first-order transduction* is a transformation of relational structures specified by first-order formulas. As graphs can be represented by relational structures, we can use first-order transductions as transformations between graphs. A first-order transduction is a generalisation of the following kind of operations: (i) the definition of a relational structure "inside" another one (in model theory, this is called an *interpretation*) and (ii) the replacement of a structure $\mathbf{A}$ by the union of a fixed number of disjoint copies of $\mathbf{A}$, augmented with appropriate relations between the copies. Examples of such transductions are graph complementation and graph doubling (transformation of a graph into the disjoint union of two copies of it).

In order to increase the power of transductions we also consider *monadic lifts*, which consist in marking some subsets of vertices using new unary predicates. Monadic lifts can be seen as a way to provide some set parameters to transductions, in the spirit of Blumensath and Courcelle's monadic second-order transductions [2]. These monadic lifts will in general not be expressed using first-order logic. Formal definitions will be given in Section 2.

We postulate that if we start with a structurally simple class $\mathscr{C}$ of graphs, e.g., a class with bounded expansion or a nowhere dense class, and then study a graph class $\mathscr{D}$ that is obtained from $\mathscr{C}$ by a first-order transduction T, then $\mathscr{D}$ should still have a simple structure and thus be well behaved algorithmically as well as in terms of logic. For instance, a useful feature of transductions is that they provide a canonical way of reducing model-checking problems from the generated class $\mathscr{D}$ to the original class $\mathscr{C}$, provided that given a graph $H \in \mathscr{D}$, we can efficiently compute some graph $G \in \mathscr{C}$ that is mapped to H by the transduction T. In general, this is a very hard problem that we do not expect to be able to solve efficiently (in polynomial time). However, with a deep understanding of the structure of the classes $\mathscr{C}$ and $\mathscr{D}$, we can solve the problem in a weaker way, which is sufficient to solve model-checking problems: Instead of computing some $G \in \mathscr{C}$ that is mapped to H by T, we consider some class $\mathscr{C}'$ with structural properties close to those of $\mathscr{C}$ and a transduction T' from $\mathscr{C}'$ to $\mathscr{D}$ such that, given $H \in \mathscr{D}$ we can compute efficiently some $G \in \mathscr{C}'$ which is mapped to H by T'.

The above principle has so far been successfully applied in the setting of graph classes with bounded treewidth and monadic second-order transductions: it was shown by Courcelle [5] (see also [7], Theorem 7.47) that classes of bounded cliquewidth can be characterized as monadic second-order transductions of trees. This, combined with Oum's result [37], gives a fixed-parameter algorithm for model-checking monadic second-order logic on classes with bounded cliquewidth. More recently, the same principle has been applied to graphs with bounded degree [18], leading to a combinatorial characterization of first-order transductions of such classes, and to a model-checking algorithm.

Applying our postulate to bounded expansion classes yields the central notion of this article: a class of graphs has *structurally bounded expansion* if it is the image of a class with bounded expansion under some first-order transduction. This article is a step towards a combinatorial, algorithmic, and logical understanding of such graph classes.

Low Shrubdepth Covers. The method of transductions is one way of constructing graphs out of graphs with a limited increase of complexity resulting from the possibility of using monadic lifts in complement of transductions. A more combinatorial approach, allowing an important increase of complexity, is the method of *covers* (or *colorings*) [36]. This method can be used to provide a characterization with bounded expansion classes in terms of very simple graph classes, namely classes with *bounded treedepth*. A class of graphs has bounded treedepth if there is a bound on the length of simple paths in the graphs in the class (see Section 2 for a different but equivalent definition). A class $\mathscr{C}$ has *low treedepth covers* if, for every number $p \in \mathbb{N}$, there is a number N and a class with bounded treedepth $\mathscr{T}$ such that, for every $G \in \mathscr{C}$, the following holds: the vertex set $V(G)$ can be covered by N sets $U_1, \ldots, U_N$ so that every set $X \subseteq V(G)$ of at most p vertices is contained in some U_i, and for each $i = 1, \ldots, N$, the subgraph of G induced by U_i belongs to $\mathscr{T}$.

A consequence of a result by Nešetřil and Ossona de Mendez [34] on a related notion of *low treedepth colorings* is that a graph class has bounded expansion if, and only if, it has low treedepth covers. The covering method allows to lift algorithmic, logical, and structural properties from classes with bounded treedepth to classes with bounded expansion. For instance, this was used to show tractability of first-order model-checking on bounded expansion classes [12, 25].

An analogue of treedepth in the dense world is the concept of *shrubdepth*, introduced in [22]. Shrubdepth shares many of the good algorithmic and logical properties of treedepth. This notion is defined combinatorially, in the spirit of the definition of cliquewidth, but can be also characterized by logical means, as first-order transductions of classes with bounded treedepth or, equivalently, as first-order transductions of classes of bounded height trees. Applying the covering method to the notion of shrubdepth leads to the following definition: A class $\mathscr{C}$ of graphs has *low shrubdepth covers* if, for every number $p \in \mathbb{N}$, there is a number N and a class $\mathscr{S}$ with bounded shrubdepth such that, for every $G \in \mathscr{C}$, there is a *p-cover* of G consisting of N sets $U_1, \ldots, U_N \subseteq V(G)$, so that every set $X \subseteq V(G)$ of at most p vertices is contained in some U_i and for each $i = 1, \ldots, N$, the subgraph of G induced by U_i belongs to $\mathscr{S}$. Shrubdepth properly generalizes treedepth and consequently classes admitting low shrubdepth covers properly extend classes with bounded expansion.

The notions of low rankwidth colorings and low shrubdepth colorings were introduced in [31], where it was shown that, for every fixed $r \in \mathbb{N}$ and every class $\mathscr{C}$ with bounded expansion, the class of rth power graphs G^r of graphs from $\mathscr{C}$ (the rth power of a graph is a simple first-order transduction) admits low shrubdepth colorings. In our logical setting, it is much more convenient to work with covers instead of colorings.

Our Contributions. Our main result states that a class of graphs $\mathscr{C}$ has structurally bounded expansion if and only if $\mathscr{C}$ has bounded shrubdepth covers. That is, transductions of classes with bounded expansion are the same as classes with low shrubdepth covers (cf. Figure 1). We also prove that every class with structurally bounded expansion is transduction-equivalent to a class with bounded expansion. These results give a combinatorial characterization of structurally bounded expansion classes, which is an important step towards their algorithmic treatment.

$$\begin{tikzcd}
\textit{bounded treedepth} \arrow[r, "\text{transduction}"] \arrow[d, "\text{cover}"] & \begin{array}{c}\textit{transduction with bounded treedepth} \\ =_{(1)} \textit{bounded shrubdepth}\end{array} \arrow[d, "\text{cover}"] \\
\begin{array}{c}\textit{low treedepth covers} \\ =_{(2)} \textit{bounded expansion}\end{array} \arrow[r, "\text{transduction}"] & \begin{array}{c}\textit{low shrubdepth covers} =_{(\star)} \\ \textit{structurally bounded expansion}\end{array}
\end{tikzcd}$$

Fig. 1. The nodes in the diagram depict properties of graph classes, and the arrows depict operations on properties of graph classes. Equality (1) is by [22]. Equality (2) is by [34]. Equality ($\star$) is the main result of this article, Theorem 6.8.

One of the key ingredients of our proof is a quantifier-elimination result (Theorem 6.9) for classes with structurally bounded expansion. This result strengthens in several ways similar results for bounded expansion classes due to Dvořák et al. [12], Grohe and Kreutzer [25], and Kazana and Segoufin [30]. Our assumption is more general, as the earlier results assume that $\mathscr{C}$ has bounded expansion, and here $\mathscr{C}$ is only required to have low shrubdepth covers.

As explained earlier, the transduction method allows to reduce the model-checking problem to the problem of finding inverse images under transductions, which is a hard problem in general and depends very much on the specific transduction. On the other hand, as we show, the cover method allows the reduction of the model-checking problem for classes with low shrubdepth covers to the problem of computing a bounded shrubdepth cover of a given graph. In fact, as a consequence of our proof, in Theorem 7.2, we show that it is enough to compute a 2-cover of each graph in a structurally bounded expansion class, in order to obtain an algorithm for the model-checking problem for this class. We conjecture that such 2-covers can be computed in polynomial time from classes with structurally bounded expansion and that therefore first-order model-checking is fixed-parameter tractable on these classes. We leave this problem for future work.

Organization. In Section 2, we collect basic facts about logic, transductions, treedepth, shrubdepth, and the notion with bounded expansion. In Section 3, we provide the formal definitions of structurally bounded expansion classes and classes with low shrubdepth covers, and state the main results and their proofs using lemmas which are proved in the following three sections. We consider algorithmic aspects in Section 7 and conclude in Section 8. We include an appendix with alternative proofs of Lemma 5.4 and Lemma 6.2, which might be of interest for some readers.

2 PRELIMINARIES

We use standard graph notations. All graphs considered in this article are undirected, finite, and simple; that is, we do not allow loops or multiple edges with the same pair of endpoints. We follow the convention that the composition of an empty sequence of functions is the identity function. For an integer k, we denote the set $\{1, \ldots, k\}$ by $[k]$.

2.1 Structures, Logic, and Transductions

Structures and Logic. A *signature* Σ is a finite set of relation and function symbols, each with a prescribed arity. In this article, we do not consider signatures with function symbols with arity different from one. A Σ-*structure* $\mathbf{A}$ consists of a finite *universe* (or *domain*) $V(\mathbf{A})$ and interpretations of the symbols in the signature: each relation symbol $R \in \Sigma$, say of arity k, is interpreted as a k-ary relation $R^{\mathbf{A}} \subseteq V(\mathbf{A})^k$, whereas each function symbol f is interpreted as a function $f^{\mathbf{A}} : V(\mathbf{A}) \to V(\mathbf{A})$. For a signature Σ, we consider standard first-order logic over Σ, that is, formulas constructed using the symbols in Σ; the special relations $=, \neq$; the connectives $\vee, \wedge, \rightarrow, \leftrightarrow, \oplus, \neg$

(here $\varphi \oplus \psi$ is the negation of $\varphi \leftrightarrow \psi$); and quantification over vertices. If $\mathbf{A}$ is a structure and $X \subseteq V(\mathbf{A})$, then we define the *substructure* $\mathbf{A}[X]$ of $\mathbf{A}$ induced by X in the usual way except that, for each unary function f, we define $f^{\mathbf{A}[X]}(x) = f^{\mathbf{A}}(x)$ if $f^{\mathbf{A}}(x) \in X$ and $f^{\mathbf{A}[X]}(x) = x$, otherwise. The *Gaifman graph* of a structure $\mathbf{A}$ is the graph with vertex set $V(\mathbf{A})$ where two distinct elements $u, v \in \mathbf{A}$ are adjacent if and only if either u and v appear together in some tuple in some relation in $\mathbf{A}$, or $f(u) = v$ or $f(v) = u$ for some function f in $\mathbf{A}$. We say that a function $f : V(\mathbf{A}) \to V(\mathbf{A})$ is *guarded by* $\mathbf{A}$ if, for every $u, v \in \mathbf{A}$ such that $f(u) = v$, the edge uv is in the Gaifman graph of $\mathbf{A}$. For a formula $\varphi(x_1, \ldots, x_k)$ with k free variables and a structure $\mathbf{A}$, we define:

$$\varphi(\mathbf{A}) = \{(v_1, \ldots, v_k) \in V(\mathbf{A})^k : \mathbf{A} \models \varphi(v_1, \ldots, v_k)\}.$$

We sometimes write $\bar{x}$ for a tuple $(x_1, \ldots, x_k)$ of variables and leave it to the context to determine the length of the tuple. The above equality then rewrites as $\varphi(\mathbf{A}) = \{\bar{v} \in V(\mathbf{A})^{|\bar{x}|} : \mathbf{A} \models \varphi(\bar{v})\}$. Also, if $\bar{a} \in V(\mathbf{A})$ and $|\bar{a}| < |\bar{x}|$, we define:

$$\varphi(\bar{a}, \mathbf{A}) = \{\bar{v} \in V(\mathbf{A})^{|\bar{x}|-|\bar{a}|} : \mathbf{A} \models \varphi(\bar{a}, \bar{v})\}.$$

Graphs, Colored Graphs, and Trees. Graphs can be viewed as finite structures over the signature consisting of a binary relation symbol E, interpreted as the edge relation, in the usual way. For a finite label set Γ, by a Γ-*colored* graph, we mean a graph enriched by a unary predicate U_γ for every $\gamma \in \Gamma$. We will follow the convention that if $\mathscr{C}$ is a class of colored graphs, then we implicitly assume that all graphs in $\mathscr{C}$ are over the same fixed finite signature. A rooted forest is an acyclic graph F together with a unary predicate $R \subseteq V(F)$ selecting one root in each connected component of F. A tree is a connected forest. The *depth* of a node x in a rooted forest F is the number of vertices in the unique path between x and the root of the connected component of x in F. In particular, x is a root of F if and only if F has depth 1 in F. The depth of a forest is the largest depth of any of its nodes. The *least common ancestor* of nodes x and y in a rooted tree is the common ancestor of x and y that has the largest depth.

Monadic Lifts. A *monadic lift* of a class $\mathscr{C}$ of structures with signature σ to a class $\mathscr{C}^+$ with signature σ augmented by a finite set S of unary relation symbols is a bijection $\Lambda : \mathscr{C} \to \mathscr{C}^+$ such that $\Lambda(\mathbf{A})$ is obtained from $\mathbf{A} \in \mathscr{C}$ by adding to it a unary relation for each symbol in S.

Transductions. A *first-order transduction* T maps an input structure $\mathbf{A}$ to an output structure $\mathsf{T}(\mathbf{A})$. Transductions are defined as sequences of *atomic transductions* –relation expansion, function expansion, restriction, copying, and reduct–which are described below.

Relation Expansion. Let $\varphi(x_1, \ldots, x_k)$ be a first-order formula with k free variables and let R be a relation symbol with arity k. We denote by $\mathsf{Rel}^{\varphi \to R}$ the operation that given a structure $\mathbf{A}$ outputs the structure $\mathbf{A}$ expanded (if necessary) by the k-ary relation R interpreted as the set $\varphi(\mathbf{A})$ of all the k-tuples of elements satisfying φ in $\mathbf{A}$.

Function Expansion. Let $\varphi(x, y)$ be a first-order formula and let f be a unary function symbol. We denote by $\mathsf{Fun}^{\varphi \to f}$ the operation that given a structure $\mathbf{A}$ outputs the structure $\mathbf{A}$ expanded (if necessary) by the unary function f defined by

$$f(x) = \begin{cases} y & \text{if } \varphi(x, \mathbf{A}) = \{y\} \\ x & \text{otherwise.} \end{cases}$$

Restriction. Let $\psi(x)$ be a first-order formula. We denote by Restr^{ψ} the operation that given a structure $\mathbf{A}$ outputs the substructure $\mathbf{A}[\psi(\mathbf{A})]$ of $\mathbf{A}$ induced by $\psi(\mathbf{A})$.

Copying. Let C be a unary relation symbol and let M be a binary relation symbol. We denote by $\mathsf{Copy}^{C,M}$ the operation that given a structure $\mathbf{A}$ outputs the disjoint union of two copies of $\mathbf{A}$ expanded (if necessary) with a the unary predicate C that marks the newly created vertices (the copy vertices), and a symmetric binary relation M that connects each vertex with its copy (the matching of vertices and copy vertices).

Reduct. Let R (respectively, f) be a relation symbol (resp. a function symbol). We denote by Reduct^R (respectively, Reduct^f) the operation that given a structure $\mathbf{A}$ outputs the reduct of the structure $\mathbf{A}$ without the relation R (respectively, without the function f).

In this article, *transductions* are defined as finite sequences of (compatible) atomic transductions applied in order, and we denote by $\mathsf{S}_1; \mathsf{S}_2; \dots; \mathsf{S}_n$ the transduction T that, given a structure $\mathbf{A}$, first applies S_1 to $\mathbf{A}$, then S_2 to the output of S_1, and so on, and eventually S_n on the output of S_{n-1}. The *composition* of two transductions $\mathsf{T}_1 = \mathsf{S}_1; \dots; \mathsf{S}_a$ and $\mathsf{T}_2 = \mathsf{S}'_1; \dots; S'_b$ is the transduction $\mathsf{T}_1; \mathsf{T}_2 = \mathsf{S}_1; \dots; \mathsf{S}_a; \mathsf{S}'_1; \dots; \mathsf{S}'_b$. The empty transduction is denoted Id and it satisfies $\mathsf{Id}(\mathbf{A}) = \mathbf{A}$ for all structures $\mathbf{A}$. If $\mathscr{C}$ is a set of structures, then $\mathsf{T}(\mathscr{C})$ is defined as $\{\mathsf{T}(\mathbf{A}) : \mathbf{A} \in \mathscr{C}\}$.

A transduction is called an *interpretation* if it does not use copying. A *faithful transduction* is a transduction that does not use copying and restrictions. A *guarded transduction* is a faithful transduction that, given a structure $\mathbf{A}$, produces a structure whose Gaifman graph is a subgraph of the Gaifman graph of $\mathbf{A}$.

We emphasize the next definition, which contains some rather technical condition.

Definition 2.1. A transduction is *almost quantifier-free* if all formulas that parameterize its atomic transductions are quantifier-free and if all function expansion operations involved in atomic transductions are parameterized by quantifier-free formulas of the type $\varphi(x, y)$ where no functions are applied to y.

We use the adverb "almost" to emphasize that function expansion transductions implicitly use quantifiers.

Transduction Equivalence. We say that two transductions S and T are *equivalent* on a class $\mathscr{C}$ of structures, and write $\mathsf{S} \equiv_{\mathscr{C}} \mathsf{T}$, if $\mathsf{S}(\mathbf{A}) = \mathsf{T}(\mathbf{A})$ for all $\mathbf{A} \in \mathscr{C}$. When $\mathscr{C}$ is the class of all structures (with appropriate signature), we simply say that S and T are *equivalent* and write $\mathsf{S} \equiv \mathsf{T}$. We call two classes $\mathscr{C}$ and $\mathscr{D}$ *transduction equivalent* if there exist monadic lifts $\Lambda_{\mathscr{C}} : \mathscr{C} \to \mathscr{C}^+$ and $\Lambda_{\mathscr{D}} : \mathscr{D} \to \mathscr{D}^+$ and transductions $\mathsf{S} : \mathscr{C}^+ \to \mathscr{D}^+$ and $\mathsf{T} : \mathscr{D}^+ \to \mathscr{C}^+$ such that $\mathsf{S}; \mathsf{T} \equiv_{\mathscr{C}^+} \mathsf{Id}$ and $\mathsf{T}; \mathsf{S} \equiv_{\mathscr{D}^+} \mathsf{Id}$. Schematically, we have:

$$
\begin{tikzcd}
\mathscr{C}^+ \arrow[loop left, "\mathsf{Id}_{\mathscr{C}^+}"] \arrow[r, bend left, "\mathsf{S}"] & \mathscr{D}^+ \arrow[loop right, "\mathsf{Id}_{\mathscr{D}^+}"] \arrow[l, bend left, "\mathsf{T}"] \\
\mathscr{C} \arrow[u, "\Lambda_{\mathscr{C}}"] & \mathscr{D} \arrow[u, "\Lambda_{\mathscr{D}}"']
\end{tikzcd}
$$

We remark that function expansion operations can be simulated by relation expansion operations, defining the graphs of the functions in the obvious way. They are, however, useful for extending the expressive power of transductions in which only quantifier-free formulas are allowed.

It will sometimes be convenient to work with the encoding of rooted bounded-depth trees and forests as node sets endowed with the parent function, rather than graphs with prescribed roots. The following example shows that these two encodings are transduction equivalent.

Lemma 2.2. *Let $\mathscr{C}$ be the class of rooted forests of depth at most d, for some fixed $d \in \mathbb{N}$, with unary predicates $H_1, \ldots, H_d$ on the vertices according to their depth in the forest.*

Let $\mathscr{D}$ be the class of trees of depth at most d that are represented by the parent function π. *Then there are almost quantifier-free interpetations* $\mathsf{FunTree} : \mathscr{C} \to \mathscr{D}$ *and* $\mathsf{RelTree} : \mathscr{D} \to \mathscr{C}$ *such that* $\mathsf{FunTree}; \mathsf{RelTree} \equiv_{\mathscr{C}} \mathsf{Id}$ *and* $\mathsf{RelTree}; \mathsf{FunTree} \equiv_{\mathscr{D}} \mathsf{Id}$.

Proof. Let $\mathsf{FunTree} = \mathsf{Fun}^{\varphi \to \pi}; \mathsf{Reduct}$, where

$$\varphi(x, y) := \bigvee_{i=2}^{d} H_i(x) \wedge H_{i-1}(y) \wedge E(x, y)$$

and Reduct removes all the relations H_i as well as the relation E. Then, for $F \in \mathscr{C}$, $\mathsf{FunTree}(F)$ is clearly the same rooted forest as F encoded by the parent function π.

Now let $\mathsf{RelTree} = \mathsf{Rel}^{\eta \to E}; \mathsf{Rel}^{\nu_1 \to H_1}; \ldots; \mathsf{Rel}^{\nu_d \to H_d}; \mathsf{Reduct}^{\pi}$, where

$$\begin{aligned} \eta(x, y) &:= ((\pi(x) = y) \vee (\pi(y) = x)) \wedge (x \neq y) \\ \nu_1(x) &:= \pi(x) = x \\ \nu_i(x) &:= H_{i-1}(\pi(x)) && \text{(for } 1 < i \leqslant d\text{)} \end{aligned}$$

Then, for $\mathbf{F} \in \mathscr{D}$, $\mathsf{RelTree}(\mathbf{F})$ is clearly the same rooted forest as $\mathbf{F}$ encoded by adjacency relation E and depth predicates $H_1, \ldots, H_d$. □

Normal Forms. It will sometimes be useful to assume a certain normal form of transductions. We will need two similar, yet slightly different normal forms: one for almost quantifier-free transductions and one for general transductions. The proofs are standard, we provide them for completeness.

Lemma 2.3. *Let* T *be an almost quantifier-free transduction, let* $\mathscr{C}, \mathscr{D}$ *be classes of graphs, and let* $\Lambda : \mathscr{D} \to \mathscr{D}^+$ *be a monadic lift. Assume* $\mathsf{T}(\mathscr{C}) = \mathscr{D}$. *If* T *is an interpretation, then* $\Lambda; \mathsf{T} \equiv_{\mathscr{C}} \mathsf{T}; \Lambda$. *Otherwise, there is a monadic lift* $\Lambda' : \mathscr{C} \to \mathscr{C}^+$ *and an almost quantifier-free transduction* T', *such that we have* $\Lambda'; \mathsf{T}' \equiv_{\mathscr{C}} \mathsf{T}; \Lambda$. *In other words, the following diagram commutes:*

$$\begin{tikzcd} \mathscr{C}^+ \arrow[r, "\mathsf{T}'"] & \mathscr{D}^+ \\ \mathscr{C} \arrow[u, "\Lambda'"] \arrow[r, "\mathsf{T}"'] & \mathscr{D} \arrow[u, "\Lambda"'] \end{tikzcd}$$

Proof. We give appropriate swapping rules that allow us to arrange the atomic operations that constitute T into the desired normal form, i.e., that whenever an atomic operation is followed by a monadic lift, these two operations may be appropriately swapped. This is straightforward for all atomic operations apart from copying. In particular, monadic lifts and interpretations commute.

A copying followed by a monadic lift introducing a unary predicate X is equivalent to a monadic lift that introduces two auxiliary unary predicates X_1 and X_2, followed by the copying, then by an expansion that introduces the unary predicate X, which interprets as X_1 on the original structure and as X_2 on the copy and by a reduct that drops the predicates X_1, X_2.

Having applied the above swapping rules exhaustively, the transformation $\mathsf{T}; \Lambda$ is rewritten as $\Lambda'; \mathsf{T}'$ where Λ' is a monadic lift and T' is almost quantifier-free. □

Corollary 2.4. *The composition of any sequence of operations consisting of monadic lifts and almost quantifier-free transductions can be written as a single monadic lift followed by an almost quantifier-free transduction.*

The diagram below illustrates Corollary 2.4.

$$
\begin{tikzcd}
\mathscr{C}_1^* \arrow[rrrrdd, bend left, "\mathsf{T}"] & & & & \\
\mathscr{C}_1^+ \arrow[dr, "\mathsf{T}_1"] & \mathscr{C}_2^+ \arrow[dr, dash] & \cdots & \mathscr{C}_{k-1}^+ \arrow[dr, "\mathsf{T}_{k-1}"] & \\
\mathscr{C}_1 \arrow[u, "\Lambda_1"] \arrow[uu, bend left, "\Lambda"] & \mathscr{C}_2 \arrow[u, "\Lambda_2"] & \cdots \arrow[r] & \mathscr{C}_{k-1} \arrow[u, "\Lambda_{k-1}"] & \mathscr{C}_k
\end{tikzcd}
$$

Lemma 2.5. *Let* T *be a transduction. Then,* T *is equivalent to a transduction of the form*

$$\mathsf{C}; \mathsf{F}; \mathsf{E}; \mathsf{X}; \mathsf{R},$$

where

- C *is a sequence of copying operations;*
- F *is a sequence of function expansion operations, one for each function on the output;*
- E *is a sequence of relation expansion operations, one for each relation on the output;*
- X *is a single restriction operation; and*
- R *is a sequence of reduct operations.*

Moreover, formulas parameterizing atomic operations in $\mathsf{F}; \mathsf{E}; \mathsf{X}$ *use only relations and functions that appeared originally on input or were introduced by* C*. In particular, none of these formulas uses any function or relation introduced by an atomic operation in* $\mathsf{F}; \mathsf{E}$.

Proof. We give appropriate swapping rules that allow us to arrange the atomic operations that constitute T into the desired normal form.

We perform swapping within T so that all copying operations are put at the front of the sequence of atomic operations. It suffices to show that whenever an atomic operation is followed by copying, then the two operations may be swapped. For reducts, this is obvious; while for expansions and restrictions, one should modify the formula parameterizing the operation in a straightforward way to work on each copy separately. Thus, we have rewritten T into the form $\mathsf{C}; \mathsf{I}$ where I does not use copying and hence is an interpretation.

Now consider I. It is clear that all reduct operations can be moved to the end of the transduction, since it does no harm to have more relations in the structure. Next, we move all restriction operations to the end (before the reduct operations) by showing that each restriction operation can be swapped with any relation expansion or function expansion operation. Suppose that the restriction is parameterized by a formula ψ, and it is followed by an expansion operation (relation or function), say parameterized by a formula φ. Then, the two operations may be swapped, provided we appropriately *relativize* φ as follows: add guards to all quantifiers in φ so that they range only over elements satisfying ψ, and for every term τ used in φ add guards to check that all the intermediate elements obtained when evaluating τ satisfy ψ.

Applying these swapping rules exhaustively rewrites I into the form $\mathsf{I}'; \mathsf{X}'; \mathsf{R}$, where I' is a sequence of relation expansion and function expansion operations, X' is a sequence of restriction operations, and R is a sequence of reduct operations. We now argue that X' can be replaced with a single restriction operation X. It suffices to show how to do this for two consecutive restriction operations, say parameterized by ψ_1 and ψ_2, respectively. Then, we may replace them by one restriction operation parameterized by $\psi_1 \wedge \psi_2'$, where ψ_2' is obtained from ψ_2 by relativizing it with respect to ψ_1 just as in the previous paragraph.

We are left with treating the relation expansion and function expansion operations within I'. Whenever a formula φ parameterizing some relation expansion or function expansion operation within I' uses a relation symbol R introduced by some earlier expansion operation within I', say parameterized by formula φ', then replace all occurrences of R in φ with φ'. Similarly, if φ uses some function f that was introduced by some earlier function expansion operation within I', say using formula $\varphi'(x, y)$, then replace each usage of f in φ by appropriately quantifying the image using formula $\varphi'(x, y)$. Perform the same operations on the formula in the restriction operation X.

Having performed exhaustively the operations above, formulas parameterizing all atomic operations in $\mathsf{I}'; \mathsf{X}$ use only relations and functions that appear originally in the structure or were added by C. Hence, all expansion and function expansion operations within I' that introduce symbols that are later dropped in R can be simply removed (together with the corresponding reduct operation). It now remains to observe that all atomic operations within I' commute, so they can be sorted: first function expansions, then relation expansions. □

We remark that when T in the above lemma is almost quantifier-free, then the equivalent transduction $\mathsf{C}; \mathsf{F}; \mathsf{E}; \mathsf{X}; \mathsf{R}$ is not necessarily almost quantifier-free.

2.2 Treedepth and Shrubdepth

The *treedepth* of a graph G is the minimal depth of a rooted forest F with the same vertex set as G, such that for every edge uv of G, u is an ancestor of v, or v is an ancestor of u in F. A class $\mathscr{C}$ of graphs has *bounded treedepth* if there is a bound $d \in \mathbb{N}$ such that every graph in $\mathscr{C}$ has treedepth at most d. We shall make use of the following characterization.

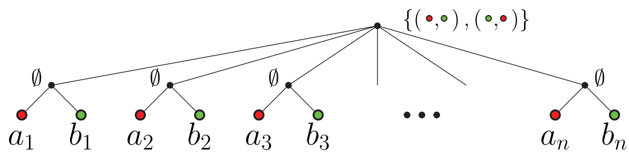
Lemma 2.6 ([33]). *A class $\mathscr{C}$ of graphs has bounded treedepth if and only if there is some number k such that no graph in $\mathscr{C}$ contains a simple path of length k. In particular, if G has treedepth at most d, then G does not contain a simple path of length 2^d.*

The notion of treedepth lifts to structures: a class $\mathscr{C}$ of structures has bounded treedepth if the class of their Gaifman graphs has bounded treedepth. In particular, a class of colored directed graphs has bounded treedepth if and only if the class of underlying uncolored undirected graphs has bounded treedepth, and applying a monadic lift to a structure does not change its treedepth.

Shrubdepth. The following notion of *shrubdepth* has been proposed in [22] as a dense analogue of treedepth. Originally, shrubdepth was defined using the notion of *tree-models*. We present an equivalent definition based on the notion of *connection models*, introduced in [22] as *m-partite cographs* with bounded depth.

A *connection model* with labels from Γ is a rooted labeled tree T where each leaf x is labeled by a label $\gamma(x) \in \Gamma$, and each non-leaf node v is labeled by a binary relation $C(v) \subseteq \Gamma \times \Gamma$. If $C(v)$ is symmetric for all non-leaf nodes v, then such a model defines a graph G on the leaves of T, in which two distinct leaves x and y are connected by an edge if and only if $(\gamma(x), \gamma(y)) \in C(v)$, where v is the least common ancestor of x and y. We say that T is a *connection model* of the resulting graph G.

Example 2.7. Fix $n \in \mathbb{N}$, and let G_n be the bi-complement of a matching of order n, i.e., the bipartite graph with nodes $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, such that a_i is adjacent to b_j if and only if $i \neq j$. A connection model for G_n is shown below:



A class of graphs $\mathscr{C}$ has *bounded shrubdepth* if there is a number $h \in \mathbb{N}$ and a finite set of labels Γ such that every graph $G \in \mathscr{C}$ has a connection model of depth at most h using labels from Γ.

A partial analog of Lemma 2.6 has been proved for induced paths in classes with bounded shrubdepth.

Lemma 2.8 ([20]). *For every class $\mathscr{C}$ of graphs with bounded shrubdepth there exists a constant $r \in \mathbb{N}$ such that no graph from $\mathscr{C}$ contains a path on more than r vertices as an induced subgraph. Consequently, for every graph $G \in \mathscr{C}$ every connected component of G has diameter at most $r - 1$.*

We can naturally extend the definition of shrubdepth to structures with an arbitrary (finite) number of binary relations by using relations C_R labeling the non-leaf nodes for each relation R. Functions can be represented by binary relations in the natural way. This allows to define classes of Σ-structures with bounded shrubdepth, for a signature Σ containing only unary and binary relation symbols and unary function symbols, as above. A class of colored graphs has bounded shrubdepth if and only if the class of underlying uncolored graphs has bounded shrubdepth. Of course, any monadic lift of a class with bounded shrubdepth has again bounded shrubdepth.

Shrubdepth of undirected graphs can be equivalently defined in terms of another graph parameter, as follows: Given a graph G and a set of vertices $W \subseteq V(G)$, the graph obtained by *flipping the adjacency within* W is the graph G' with vertices $V(G)$ and edge set which is the symmetric difference of the edge set of G and the edge set of the clique on W.

The *subset-complementation depth*, or *SC-depth*, of a graph is defined inductively as follows:

- a graph with one vertex has SC-depth 0, and
- a graph G has SC-depth at most d, where $d \geqslant 1$, if there is a set of vertices $W \subseteq V(G)$ such that in the graph obtained from G by flipping the adjacency within W all connected components have SC-depth at most $d - 1$.

Example 2.9. A star has SC-depth at most 2: flipping the adjacency within the set consisting of the vertices of degree 1 yields a clique, which in turn has SC-depth at most 1.

The notion of SC-depth leads to a natural notion of decompositions. An *SC-decomposition* of a graph G of SC-depth at most d is a rooted forest Y of depth at most d encoded using a parent function π, with leaf set $V(G)$, with unary predicates $W_1, \ldots, W_{k-1}$ on the leaves at depth $k \leqslant d$. Each connected component of Y (with restriction of the predicates $W_1, \ldots, W_{d-1}$) is an SC-decomposition of a corresponding connected component of G. When G is connected, denoting by r the root of the tree Y, the graph G' obtained by flipping the adjacency within W_1 has $Y - r$, together with the unary predicates $W_2, \ldots, W_{d-1}$ renamed as $W_1, \ldots, W_{d-2}$ as SC-decomposition. It is easily checked that one can require that every leaf has depth d. An SC-decomposition where all the leaves are at the same depth is called *normal.*

Lemma 2.10. *Let G be a graph with SC-depth at most d and let Y be a normal SC-depth decomposition of G with depth d. Then, for every two vertices $u, v \in V(G)$, we have:*

$$G \models E(u,v) \quad\iff\quad Y \models \varsigma_d(u,v), \tag{1}$$

where

$$\varsigma_d(x,y) := (x \neq y) \wedge (\pi^{d-1}(x) = \pi^{d-1}(y)) \wedge \bigoplus_{i=1}^{d-1} W_i(x) \wedge W_i(y) \wedge (\pi^{d-i}(x) = \pi^{d-i}(y)). \tag{2}$$

Proof. We prove the lemma by induction on d. If $d = 1$, then G is edgeless and (1) holds. Assume the lemma holds up to $d-1$ and let Y be a normal SC-decomposition with depth d of a graph G. As $\pi^{d-1}(u) = \pi^{d-1}(v)$ is equivalent to the fact that u and v belong to a same connected component of Y we can reduce to the case where Y and G are connected. Denote by $G \oplus W_1$ the graph obtained from G by flipping the adjacency within W_1. Let r be the root of Y. The parent function of $Y - r$ is the same as the one of Y except at the children of r. Hence, for every leaf u of Y and every $k < d-1$, the value of $\pi^k(u)$ is the same when evaluated in Y and $Y - r$. Thus, it follows from the induction hypothesis that we have u and v are adjacent in $G \oplus W_1$ if and only if $u \neq v$, $\pi^{d-2}(u) = \pi^{d-2}(v)$ and

$$Y \models \bigoplus_{i=2}^{d-1} W_i(u) \wedge W_i(v) \wedge (\pi^{d-i}(u) = \pi^{d-i}(v)).$$

From this, Equation (1) follows immediately. □

Note that while the notion of shrubdepth makes sense also for structures with binary relations and unary functions, the notion of SC-depth makes sense only for (colored) graphs.

We will make use of the following properties:

Lemma 2.11. *Let $\mathscr{C}$ be a class of (colored) graphs.*

(1) If $\mathscr{C}$ has bounded shrubdepth, then the class of all induced (colored) subgraphs of graphs from $\mathscr{C}$ also has bounded shrubdepth.
(2) $\mathscr{C}$ has bounded shrubdepth if and only if for some $d \in \mathbb{N}$ all graphs in $\mathscr{C}$ have SC-depth at most d.
(3) If $\mathscr{C}$ has bounded treedepth, then $\mathscr{C}$ has bounded shrubdepth.
(4) If $\mathscr{C}$ has bounded shrubdepth and T is a transduction that outputs colored graphs, then $\mathsf{T}(\mathscr{C})$ has bounded shrubdepth.

Proof. Property (1) follows from the definition of shrubdepth. Properties (2) and (3) follow from [22]. Property (4) was proved in [22] in the special case of one application of a relation expansion operation with a formula with two free variables that defines a symmetric relation. Let us show this property for a general transduction T. By Lemma 2.5, T can be written as a transduction of the form $\mathsf{C}; \mathsf{E}; \mathsf{X}; \mathsf{R}$, where E outputs the symmetric binary edge relation. Let $\mathscr{C}$ be a class with bounded shrubdepth. We show by induction on the structure of T that $\mathsf{T}(\mathscr{C})$ has bounded shrubdepth. It is immediate by definition that a reduct operation cannot increase the shrubdepth of a graph class. Hence, it remains to show that $(\mathsf{C}; \mathsf{E}; \mathsf{X})(\mathscr{C})$ has bounded shrubdepth. Consider a single copy operation $\mathsf{Copy}^{C,M}$ and let $G \in \mathscr{C}$. It is easy to see that a connection model for G can be modified to a connection model for $\mathsf{Copy}^{C,M}(G)$ whose height increases by one. Hence, by applying this argument for all copy operations of C, we conclude that also $\mathsf{C}(\mathscr{C})$ has bounded shrubdepth. By the result of [22], the application of E to a class with bounded shrubdepth is again a class with bounded shrubdepth. Hence, $(\mathsf{C}; \mathsf{E})(\mathscr{C})$ has bounded shrubdepth. Finally, the application of X does not increase the shrubdepth by Property (1). □

In the absence of large bi-cliques (complete bipartite graphs) a graph with bounded cliquewidth has in fact bounded treewidth [27]. The same holds also for shrubdepth and treedepth.

Lemma 2.12. *A class of (colored) graphs $\mathscr{C}$ has bounded treedepth if and only if graphs in $\mathscr{C}$ have bounded shrubdepth and exclude some fixed bi-clique as a subgraph.*

Proof. One implication is easy: by statement (3) of Lemma 2.11, every class with bounded treedepth also has bounded shrubdepth, and moreover the bi-clique $K_{s,s}$ has treedepth $s+1$, so every class with bounded treedepth excludes some bi-clique.

Now assume $\mathscr{C}$ has bounded shrubdepth and excludes some bi-clique $K_{s,s}$. It is proved in [1] that for all natural numbers s and r there is a natural number $P(r,s)$ such that any graph with a path of length $P(r,s)$ has either an induced path of length r or a bi-clique of size s. However, according to Lemma 2.8, every class with bounded shrubdepth excludes some path as an induced subgraph. It follows that the graphs in the class $\mathscr{C}$ exclude some fixed path as a subgraph and by Lemma 2.6 we conclude that $\mathscr{C}$ has bounded treedepth. □

2.3 Bounded Expansion

A graph H is a *depth-r minor* of a graph G if H can be obtained from a subgraph of G by contracting mutually disjoint connected subgraphs of radius at most r. A class $\mathscr{C}$ of graphs has *bounded expansion* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\frac{|E(H)|}{|V(H)|} \leqslant f(r)$ for every $r \in \mathbb{N}$ and every depth-r minor H of a graph from $\mathscr{C}$. Examples include the class of planar graphs, or any class of graphs with bounded maximum degree. We lift the notion with bounded expansion to classes of structures over an arbitrary fixed signature, by requiring that their class of Gaifman graphs has bounded expansion. In particular, a class of colored graphs has bounded expansion if and only if the class of underlying uncolored graphs has bounded expansion.

We will make use of the following lemma:

Lemma 2.13. *Let $\mathscr{C}$ be a class of (colored) graphs with bounded expansion and let C be a copy operation. Then $\mathsf{C}(\mathscr{C})$ has bounded expansion.*

Proof. Let $G \in \mathscr{C}$. The Gaifman graph of $\mathsf{C}(G)$ is a subgraph of the so-called *lexicographic product of G with K_2*, i.e., it is constructed from the latter by replacing every vertex with two clones of it. It is known that if a class of graphs $\mathscr{C}$ has bounded expansion, then the class of lexicographic products of graphs from $\mathscr{C}$ with any fixed graph H also has bounded expansion [34]. □

The connection between treedepth and graph classes with bounded expansion can be established via *p-treedepth colorings*. For an integer p, a function $c : V(G) \to C$ is a p-treedepth coloring if, for every $i \leqslant p$ and set $X \subseteq V(G)$ with $|c(X)| = i$, the induced graph $G[X]$ has treedepth at most i. A graph class $\mathscr{C}$ has *low treedepth colorings* if for every $p \in \mathbb{N}$ there is a number N_p such that, for every $G \in \mathscr{C}$, there exists a p-treedepth coloring $c : V(G) \to C$ with $|C| \leqslant N_p$.

Theorem 2.14 ([34]). *A class of graphs has bounded expansion if and only if it has low treedepth colorings.*

2.4 Structurally Bounded Expansion and Covers

We introduce two notions which generalize the concept of bounded expansion. First, we introduce classes with *structurally bounded expansion*. This notion arises from closing bounded expansion graph classes under transductions.

Definition 2.15. A class $\mathscr{C}$ of structures has *structurally bounded expansion* if there exists a class of colored graphs $\mathscr{D}$ with bounded expansion and a transduction T such that $\mathscr{C} = \mathsf{T}(\mathscr{D})$.

The second notion, *low shrubdepth covers*, is inspired by the low treedepth coloring characterisation with bounded expansion (see Theorem 2.14), by replacing treedepth by shrubdepth. For convenience, we formally define this in terms of *covers*.

Definition 2.16. For $p \in \mathbb{N}$ A *p-cover* of a structure $\mathbf{A}$ is a family $\mathcal{U}_{\mathbf{A}}$ of subsets of $V(\mathbf{A})$ such that if every set of at most p elements of $\mathbf{A}$ is contained in some $U \in \mathcal{U}_{\mathbf{A}}$. If $\mathscr{C}$ is a class of structures, then a *p-cover* of $\mathscr{C}$ is a family $\mathcal{U} = (\mathcal{U}_{\mathbf{A}})_{\mathbf{A}\in\mathscr{C}}$, where $\mathcal{U}_{\mathbf{A}}$ is a p-cover of $\mathbf{A}$. A 1-cover is simply called a *cover*. A cover $\mathcal{U}$ is *finite* if $\sup\{|\mathcal{U}_{\mathbf{A}}| : \mathbf{A} \in \mathscr{C}\}$ is finite. Let $\mathscr{C}[\mathcal{U}]$ denote the class structures $\{\mathbf{A}[U] : \mathbf{A} \in \mathscr{C}, U \in \mathcal{U}_{\mathbf{A}}\}$. We say that a cover $\mathcal{U}$ has *bounded treedepth* (respectively, bounded shrubdepth) if the class $\mathscr{C}[\mathcal{U}]$ has bounded treedepth (respectively, bounded shrubdepth).

Example 2.17. Let $\mathscr{T}$ be the class of trees and let $p \in \mathbb{N}$. We construct a finite p-cover $\mathcal{U}$ of $\mathscr{T}$ with bounded treedepth. Given a rooted tree T, let $\mathcal{U}_T = \{U_0, \dots, U_p\}$, where U_i is the set of vertices of T whose depth is not congruent to i modulo $p + 1$. Note that $T[U_i]$ is a forest of depth p, and that $\mathcal{U}_T$ is a p-cover of T. Hence, $\mathcal{U} = (\mathcal{U}_T)_{T\in\mathscr{T}}$ is a finite p-cover of $\mathscr{T}$ with bounded treedepth.

Example 2.18. Let $\mathscr{C}$ be the class of squares of trees: each graph $G \in \mathscr{C}$ is derived from a tree $T \in \mathscr{T}$ by making adjacent any two vertices at distance at most two (i.e., $G = T^2$). We construct a finite p-cover $\mathcal{U}$ of $\mathscr{C}$ with bounded shrubdepth.

Given a rooted tree T, let V_i (for $0 \leqslant i \leqslant 3p$) be the set of vertices of T whose depth is congruent to i modulo $3p + 1$. For $I \subseteq \{0, \dots, 3p\}$, let $V_I = \bigcup_{i\in I} V_i$, let $I^+ = \{i + j \bmod 3p + 1 : i \in I$ and $-1 \leqslant j \leqslant 1\}$, and let $U_I = (T[V_{I^+}])^2[V_I]$. As $T[V_{I^+}]$ is a forest of depth at most $3p$ and as U_I is obtained from $T[V_{I^+}]$ by a monadic lift marking V_I followed by a transduction (taking the square, then reducing to marked vertices), the family $\mathcal{U}_T = \{U_I : I \subseteq \{0, \dots, 3p\}$ and $|I| = p\}$ has bounded shrubdepth.

For every subset I of $\{0, \dots, 3p\}$ of size p, it is easily checked that the restriction of $G = T^2$ on V_I can be computed on the restriction of T on V_{I^+}, that is: $G[V_I] = (T[V_{I^+}]^2)[V_I] = U_I$, hence $\mathcal{U}_T$ is a p-cover of G.

Thus, $\mathcal{U} = (\mathcal{U}_T)_{T\in\mathscr{T}}$ is a finite p-cover of $\mathscr{C}$ with bounded shrubdepth.

In analogy to low treedepth colorings, we can now characterize graph classes with bounded expansion using covers. We say that a class $\mathscr{C}$ of graphs has *low treedepth covers* if for every $p \in \mathbb{N}$ there is a finite p-cover $\mathcal{U}$ of $\mathscr{C}$ with bounded treedepth. The following lemma follows easily from Theorem 2.14.

Lemma 2.19. *A class of graphs has bounded expansion if and only if it has low treedepth covers.*

Proof. We will prove that a graph class $\mathscr{C}$ has low treedepth colorings if and only if it has low treedepth covers. The result then follows from Theorem 2.14.

We start with the left-to-right direction. Assume $\mathscr{C}$ has low treedepth colorings. Then, for every graph $G \in \mathscr{C}$ and $p \in \mathbb{N}$, we may find a vertex coloring $\gamma : V(G) \to [N]$ using N colors where every $i \leqslant p$ color classes induce in G a subgraph of treedepth at most i; here, N depends only on p and $\mathscr{C}$. Assuming without loss of generality that $N \geqslant p$, define a p-cover $\mathcal{U}_G$ of size at most $\binom{N}{p}$ as follows: $\mathcal{U}_G = \{\gamma^{-1}(X) : X \subseteq [N], |X| = p\}$. Then, $\mathcal{U} = (\mathcal{U}_G)_{G\in\mathscr{C}}$ is a finite p-cover of $\mathscr{C}$ with bounded treedepth.

Conversely, suppose that every graph $G \in \mathscr{C}$ admits a p-cover $\mathcal{U}_G$ of size N where $G[U]$ has treedepth at most d for each $U \in \mathcal{U}_G$; here, N and d depend only on p and $\mathscr{C}$. Define a coloring $\chi : V(G) \to \mathcal{P}(\mathcal{U}_G)$ as follows: for $v \in V(G)$, let $\chi(v)$ be the set of those $U \in \mathcal{U}_G$ for which $v \in U$. Thus, χ is a coloring of $V(G)$ with at most 2^N colors. Take any p subsets $X_1, \dots, X_p \subseteq \mathcal{U}_G$ such that $\chi^{-1}(X_i) \neq \emptyset$ for each $i \in [p]$. Arbitrarily choose any $x_i \in \chi^{-1}(X_i)$. Since $\mathcal{U}_G$ is a p-cover of G, there exists $U \in \mathcal{U}_G$ such that $\{x_1, \dots, x_p\} \subseteq U$. Consequently, for each $i \in [p]$, we have that $U \in X_i$, implying $\chi^{-1}(X_i) \subseteq U$. Hence, $G[\chi^{-1}(\{X_1, \ldots, X_p\})]$ is an induced subgraph of $G[U]$, whereas the latter graph has treedepth at most d by the assumed properties of $\mathcal{U}_G$. We conclude that every p color classes in χ induce a subgraph of treedepth at most d.

It remains to refine this coloring so that we in fact obtain a coloring such that every at most $i \leqslant p$ color classes induce a subgraph of treedepth at most i. As every p color classes in χ induce a subgraph of treedepth at most d, we can fix for every p color classes I of χ a treedepth decomposition Y_I of height at most d. We define the coloring ξ such that every vertex v gets the color $\{(I, h_I) :$ I is a subset of p color classes containing v and h_I is the depth of v in the decomposition $Y_I\}$. Note that since the number of colors of χ is finite, the number of colors used by ξ is also finite.

We now prove that in the refined coloring, any $i \leqslant p$ colors in ξ have treedepth at most i. Fix any $i \leqslant p$ colors in ξ and denote the tuple of colors by J. As ξ is a refinement of χ, there exists a tuple I of at most p colors in χ which contains all vertices of $G[J]$. Furthermore, the i selected colors of J are contained in i levels of the treedepth decomposition Y_I. Taking the restriction of these i levels yields a forest of depth at most i, which is a witness that $G[J]$ has treedepth at most i. □

We now define the second notion generalizing the concept with bounded expansion. The idea is to use low shrubdepth covers instead of low treedepth covers.

Definition 2.20. A class $\mathscr{C}$ of graphs has *low shrubdepth covers* if, and only if, for every $p \in \mathbb{N}$ there is a finite p-cover $\mathcal{U}$ of $\mathscr{C}$ with bounded shrubdepth.

It is easily seen that Lemma 2.19 together with Lemma 2.11(3) imply that every class of bounded expansion has low shrubdepth covers.

3 MAIN RESULTS AND OVERVIEW

In this section, we state the main results and outline their proofs. Our main result is the following theorem.

Theorem (Theorem 6.8). *A class of (colored) graphs has structurally bounded expansion if and only if it has low shrubdepth covers.*

As a byproduct of our proof of Theorem 6.8, we obtain the following quantifier-elimination result, which we believe is of independent interest.

Theorem (Theorem 6.9). *Let $\mathscr{C}$ be a class of (colored) graphs that has low shrubdepth covers. Then for every transduction* S, *there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free transduction* T *such that* S *is equivalent to* $\Lambda; \mathsf{T}$ *on* $\mathscr{C}$.

We now outline the proof of these theorems.

The following lemma, which we prove in Section 4, intuitively shows that covers commute with almost quantifier-free transductions.

Lemma (Lemma 4.4). *If a class of (colored) graphs $\mathscr{C}$ has low shrubdepth covers and* T *is an almost quantifier-free transduction that outputs (colored) graphs, then* $\mathsf{T}(\mathscr{C})$ *also has low shrubdepth covers.*

The next ingredient in our proof, which we prove in Section 5, states that classes with low shrubdepth covers are transduction equivalent with classes with bounded expansion, using almost quantifier-free transductions.

Lemma (Lemmas 5.11 and 5.12). *Let $\mathscr{C}$ be a class of graphs with low shrubdepth covers. Then there is a class $\mathscr{D}$ with bounded expansion, a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$, an almost quantifier-free transduction* $\mathsf{S} : \mathscr{C}^+ \to \mathscr{D}$ *and an almost quantifier-free interpretation* $\mathsf{I} : \mathscr{D} \to \mathscr{C}$, *such that* $\Lambda; \mathsf{S}; \mathsf{I} \equiv_{\mathscr{C}} \mathsf{Id}$

and $\mathsf{I};\Lambda;\mathsf{S} \equiv_{\mathscr{D}} \mathsf{Id}$*, that is, such that the following diagram commutes:*

$$\begin{tikzcd}
\mathscr{C}^+ \arrow[dr, "\mathsf{S}"] & \\
\mathscr{C} \arrow[u, "\Lambda"] \arrow[loop left, "\mathsf{Id}_{\mathscr{C}}"] & \mathscr{D} \arrow[l, "\mathsf{I}"'] \arrow[loop right, "\mathsf{Id}_{\mathscr{D}}"]
\end{tikzcd}$$

In particular, $\mathscr{C}$ *and* $\mathscr{D}$ *are transduction equivalent.*

Clearly, this lemma implies that $\mathscr{C}$ has structurally bounded expansion, since $\mathscr{C} = \mathsf{I}(\mathscr{D})$ for a class of $\mathscr{D}$ with bounded expansion. Thus, we get the right-to-left implication of Theorem 6.8 as a corollary. The proofs of Lemmas 5.11 and 5.12 are presented in Section 5.

Theorem 6.9, and the remaining implication in Theorem 6.8 follow from the following result:

Lemma (Lemma 6.7). *Let* $\mathscr{C}$ *be a class of (colored) graphs with bounded expansion and let* S *be a transduction. Then,* S *is equivalent to an almost quantifier-free transduction* T *on a monadic lift of* $\mathscr{C}$*.*

We note that a statement similar to Lemma 6.7 is provided by Dvořák et al. [13] and by Grohe and Kreutzer [25]. In this article, we provide a self-contained proof using tree automata in Section A.2 in Appendix A.

From these lemmas, we can now derive our main results.

Proof of Theorem 6.8. As observed, the right-to-left implication of Theorem 6.8 follows from Lemmas 5.11 and 5.12.

For the left-to-right implication, let $\mathscr{C}$ be a class of (colored) graphs with bounded expansion and let S be a transduction that outputs (colored) graphs. By Lemma 2.19, $\mathscr{C}$ has low treedepth covers. Applying Lemma 6.7 yields a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free transduction T such that $\mathsf{S}(G) = \mathsf{T}(\Lambda(G))$ for all $G \in \mathscr{C}$, hence, $\mathscr{D} = \mathsf{S}(\mathscr{C}) = \mathsf{T}(\Lambda(\mathscr{C}))$. As $\mathscr{C}$ has low treedepth covers, also $\Lambda(\mathscr{C})$ has low treedepth covers, and hence, in particular, $\Lambda(\mathscr{C})$ has low shrubdepth covers (cf. Lemma 2.11 (3)), we may apply Lemma 4.4 to T and $\Lambda(\mathscr{C})$ to deduce that $\mathscr{D}$ has low shrubdepth covers. □

Proof of Theorem 6.9. Lemmas 5.11 and 5.12 allows to reduce the theorem to the case of classes with bounded expansion, as almost quantifier-free transductions are closed under composition. The case with bounded expansion classes is handled by Lemma 6.7. □

4 ALMOST QUANTIFIER-FREE TRANSDUCTIONS COMMUTE WITH COVERS

In this section we prove Lemma 4.4. We first define the notions of *dependency* and *support.*

Definition 4.1. Let $\varphi(x_1, \ldots, x_k)$ be a quantifier-free formula. The closure cl_φ of φ is the set of all terms (including subterms) of φ. For a structure $\mathbf{A}$ and an element v of $\mathbf{A}$ we let

$$\mathsf{cl}_\varphi^{\mathbf{A}}(v) = \{t^{\mathbf{A}}(v) : t(x) \in \mathsf{cl}_\varphi\}.$$

We say that v φ*-depends* on the elements in $\mathsf{cl}_\varphi^{\mathbf{A}}(v)$.

Note that the size of the set $\mathsf{cl}_\varphi^{\mathbf{A}}(v)$ is always bounded by a constant depending only on φ. Observe also that given elements $v_1, \ldots, v_k$, to check whether $\varphi(v_1, \ldots, v_k)$ holds in $\mathbf{A}$ it suffices to check whether it holds in the substructure of $\mathbf{A}$ induced by all elements on which $v_1, \ldots, v_k$ φ-depend.

With the auxiliary notion of dependency defined we can come to the definition of *support.*

Definition 4.2. Let $\mathbf{A}$ be a structure, $v \in V(\mathsf{T}(\mathbf{A}))$, and let S be a subset of $V(\mathbf{A})$. Let T be an almost quantifier-free transduction.

- If $\mathsf{T} = \mathsf{Copy}^{C,M}$, then v is T-*supported* by S if $v \in S$ or $\exists w \in S : M(v, w)$.
- If $\mathsf{T} = \mathsf{Restr}^{\varphi}$ or $\mathsf{T} = \mathsf{Rel}^{\varphi \to R}$, then v is T-*supported* by S if $\mathrm{cl}^{\mathbf{A}}_{\varphi}(v) \subseteq S$.
- If $\mathsf{T} = \mathsf{Fun}^{\varphi \to f}$, then v is T-*supported* by S if $\mathrm{cl}^{\mathbf{A}}_{\varphi}(v) \subseteq S$ and $|\varphi(v, \mathbf{A}) \cap S| \geqslant \min(|\varphi(v, \mathbf{A})|, 2)$.
- If T is a reduct operation, then v is T-*supported by* S if $v \in S$.
- If $\mathsf{T} = \mathsf{T}_1; \mathsf{T}_2$, where both T_1 and T_2 are almost quantifier-free, then $v \in V(\mathsf{T}(\mathbf{A}))$ is T-*supported* by S if there exists a subset $W \subseteq V(\mathsf{T}_1(\mathbf{A}))$ such that v is T_2-supported by W and each $w \in W$ is T_1-supported by S.

The notion of supporting is trivially closed under taking supersets: if v is T-supported by S, then v is also T-supported by any superset of S. For a structure $\mathbf{A}$ and an almost quantifier-free transduction T, we fix a mapping $\mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}$ that maps $v \in V(\mathbf{A})$ to a minimal subset $\mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}(v)$ that T-supports v. Note that the size of $\mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}(v)$ is bounded by a constant depending only on T; we call this constant the *complexity* of T. By extension, if $W \subseteq V(\mathbf{A})$, we define $\mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}(W) = \bigcup_{v \in W} \mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}(v)$.

We now prove that almost quantifier-free transductions are, in a certain sense, local.

Lemma 4.3. *For every almost quantifier-free transduction* T, *for every structure* $\mathbf{A}$, *for every* $W \subseteq V(\mathsf{T}(\mathbf{A}))$, *and for every* $U \subseteq V(\mathbf{A})$ *with* $U \supseteq \mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}(W)$ *we have*

$$\mathsf{T}(\mathbf{A})[W] = \mathsf{T}(\mathbf{A}[U])[W]. \tag{3}$$

Proof. We now prove by induction on the structure of T that if $W \subseteq V(\mathsf{T}(\mathbf{A}))$ and $U \subseteq V(\mathbf{A})$ are such that every $v \in W$ is T-supported by U, then (3) holds. We prove the statement for atomic operations, the inductive step is immediate by the inductive definition of T-support.

Assume T is a copy operation. Then, the fact that $v \in W$ is T-supported by U implies $v \in U$ or $w \in U$ for the unique element w in $\mathbf{A}$ of which v is a copy if $v \in V(\mathsf{T}(\mathbf{A})) \setminus V(\mathbf{A})$. Then, clearly (3) holds. Now assume that T is a restriction or a relation expansion operation, say parameterized by a quantifier-free formula $\varphi(\bar{x})$. Then, $V(\mathsf{T}(\mathbf{A})) \subseteq V(\mathbf{A})$ and hence $W \subseteq V(\mathbf{A})$. Then, $\mathrm{cl}^{\mathbf{A}}_{\varphi}(v) \subseteq U$ for all $v \in W$. As $\varphi(\bar{x})$ is quantifier-free, the truth of $\varphi(\bar{v})$ in $\mathbf{A}$ depends only on $\bigcup_{v \in \bar{v}} \mathrm{cl}^{\mathbf{A}}_{\varphi}(v) \subseteq U$. We conclude that (3) holds. The case of a reduct operation is obvious. It remains to consider the case where T is a function expansion operation, say parameterized by a quantifier-free formula $\varphi(x, y)$. Again it holds that $W \subseteq V(\mathbf{A})$. We show that, for every $v \in W$, there exists exactly one $w \in V(\mathbf{A})$ with $\mathbf{A} \models \varphi(v, w)$, if and only if there exists exactly one $w \in U$ with $\mathbf{A} \models \varphi(v, w)$. Recall that in a function expansion operation, we demand that the formula $\varphi(x, y)$ parameterizing the operation does not have function applications on y. This implies for all $v, w \in V(\mathbf{A})$ that $\mathbf{A} \models \varphi(v, w) \Leftrightarrow \mathbf{A}[\mathrm{cl}^{\mathbf{A}}_{\varphi}(v) \cup \{w\}] \models \varphi(v, w)$. Now we have $\mathrm{cl}^{\mathbf{A}}_{\varphi}(v) \subseteq U$ for all $v \in W$ and furthermore, if there exists exactly one $w \in V(\mathbf{A})$ for which $\varphi(v, w)$ holds, then $w \in U$ and if there are at least two elements $w \in V(\mathbf{A})$ for which $\varphi(v, w)$ holds, then $w \in U$ for at least two distinct such elements w. This implies (3). $\square$

Lemma 4.4. *Let* $\mathscr{C}$ *be a class of (colored) graphs with low shrubdepth covers and let* T *be an almost quantifier-free transduction that outputs (colored) graphs. Then,* $\mathsf{T}(\mathscr{C})$ *has low shrubdepth covers.*

Proof. Let c be the complexity of the transduction T. Let $p \in \mathbb{N}$, and let $\mathcal{U}$ be a finite cp-cover of $\mathscr{C}$ with bounded shrubdepth. For a graph $G \in \mathscr{C}$ and $U \in \mathcal{U}_G$, let $W_U \subseteq V(\mathsf{T}(G))$ be the set of those elements v of $\mathsf{T}(G)$ such that $\mathrm{Support}^{\mathbf{A}}_{\mathsf{T}}(v) \subseteq U$.

Define a cover $\mathcal{W} = (\mathcal{W}_{\mathsf{T}(G)})_{G \in \mathscr{C}}$ of $\mathsf{T}(\mathscr{C})$ by letting

$$\mathcal{W}_{\mathsf{T}(G)} = \{W_U : U \in \mathcal{U}_G\} \qquad \text{for every graph } G \in \mathscr{C}.$$

Clearly, $|\mathcal{W}_{\mathsf{T}(G)}| \leqslant |\mathcal{U}_G|$, so $\mathcal{W}$ is finite. Consider any p elements $w_1, \ldots, w_p$ of $\mathsf{T}(G)$. Let $S = \mathrm{Support}^{\mathsf{A}}_{\mathsf{T}}(\{w_1, \ldots, w_p\})$. Then, $|S| \leqslant cp$; hence, there exists $U \in \mathcal{U}_G$ with $S \subseteq U$. We conclude that $\{w_1, \ldots, w_p\} \subseteq W_U \in \mathcal{W}_G$; hence, $\mathcal{W}$ is a p-cover of $\mathsf{T}(\mathscr{C})$.

By assumption, $\mathscr{C}[\mathcal{U}]$ has bounded shrubdepth; hence, by Lemma 2.11(4), $\mathsf{T}(\mathscr{C}[\mathcal{U}])$ also has bounded shrubdepth. By Lemma 4.3, for each $G \in \mathscr{C}$ and $W_U \in \mathcal{W}_{\mathsf{T}(G)}$, the induced substructure $\mathsf{T}(G)[W_U]$ is equal to $\mathsf{T}(G[U])[W_U]$. As $\mathsf{T}(G[U]) \in \mathsf{T}(\mathscr{C}[\mathcal{U}])$, we get that $\mathsf{T}(G)[W_U]$ belongs to the hereditary closure of $\mathsf{T}(\mathscr{C}[\mathcal{U}])$, which also has bounded shrubdepth by Lemma 2.11(1). Thus, $\mathcal{W}$ is a bounded shrubdepth cover of $\mathsf{T}(\mathscr{C})$. □

5 BI-DEFINABILITY

In this section, we prove Lemmas 5.11 and 5.12.

5.1 Defining Connected Components in Graphs with Bounded Shrubdepth

We first prove that, for graphs in a class with bounded shrubdepth, using a monadic lift and an almost quantifier-free transduction, we can root the connected components and then define a function mapping each vertex to the root of its connected component.

Guidable Functions. We introduce the notion of *guidable* functions, as a combinatorial abstraction of functions computable by almost quantifier-free transductions.

A *guidance system* in G is any family $\mathcal{U}$ of subsets of the vertex set of G. The *size* of a guidance system $\mathcal{U}$ is the cardinality of the family $\mathcal{U}$. We say that a function $g : V(G) \to V(G)$ is *guided* by the guidance system $\mathcal{U}$ if, for every $v \in V(G)$ for which $g(v)$ is different from v, there is some $U \in \mathcal{U}$ such that $g(v)$ is the unique neighbor of v in U. Finally, a function $g : V(G) \to V(G)$ is ℓ-*guidable*, where $\ell \in \mathbb{N}$, if there is a guidance system $\mathcal{U}$ of size at most ℓ in G that such g is guided by $\mathcal{U}$. Note that if g is ℓ-guidable it is guarded by G.

The following lemmas will be useful for operating on guidable functions.

Lemma 5.1. *Let G be a graph and let $g : V(G) \to V(G)$ be a function such that the restriction $g|_C$ of g to each connected component C of G is ℓ-guidable. Then, g is ℓ-guidable.*

Proof. For each component C of G, we may find a guidance system $\mathcal{U}^C = \{U_1^C, \ldots, U_\ell^C\}$ that guides $g|_C$. Since the fact that $g|_C$ is guidable implies that, for each $v \in V(C)$ we have $g(v) \in V(C)$, we can assume $U_i^C \subseteq V(C)$ for each $i \in [\ell]$. Let U_i be the union of the U_i^C for the connected components C of G. Then, g is guided by the guidance system $\mathcal{U} = \{U_1, \ldots, U_\ell\}$. □

Lemma 5.2. *Let G be a graph and let $g_1, \ldots, g_s : V(G) \to V(G)$ be ℓ-guidable functions. Assume $g : V(G) \to V(G)$ is a function such that, for every $v \in V(G)$, there is some $i \in [s]$ such that $g(v) = g_i(v)$. Then, g is ℓs-guidable.*

Proof. Let $\mathcal{U}_i$ be a guidance system of size at most ℓ that such that g_i is guided by $\mathcal{U}_i$. Let $\mathcal{U} = \bigcup_{i=1}^s \mathcal{U}_i$. By assumption, for every $v \in V(G)$, there is some $i \in [s]$ such that $g(v) = g_i(v)$ and there is some $U \in \mathcal{U}_i \subseteq \mathcal{U}$ such that $g_i(v)$ is the unique neighbor of v in U. Hence, $\mathcal{U}$ is a guidance system of size at most ℓs that guides the function g. □

Finally, guidable functions can be computed using almost quantifier-free transductions.

Lemma 5.3. *Let $\mathscr{C}$ be a class of graphs and let $\ell \in \mathbb{N}$ be fixed. Suppose that, to each $G \in \mathscr{C}$ is associated an ℓ-guidable function $f_G : V(G) \to V(G)$. Then, there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free function expansion $\mathsf{Fun}^{\varphi \to f}$ defining a function f, such that, for every $G \in \mathscr{C}$ and all $u, v \in V(G)$, we have $\mathsf{Fun}^{\varphi \to f}(\Lambda(G)) \models f(u) = v$ if and only if $f_G(u) = v$.*

Proof. We consider the following almost quantifier-free function expansion $\mathsf{Fun}^{\varphi\to f}$ that makes use of 2ℓ new predicates $M_1,\dots,M_\ell,N_1,\dots,N_\ell$:

$$\mathsf{Fun}^{\varphi\to f},\quad \text{where}\quad \varphi(x,y) := \bigvee_{i=1}^{\ell} N_i(x)\wedge E(x,y)\wedge M_i(y).$$

We now describe how to add predicates $M_1,\dots,M_\ell,N_1,\dots,N_\ell$ to each $G\in\mathscr{C}$ to obtain the desired lift $\Lambda(G)\in\mathscr{C}^+$. Let $\mathcal{U}$ be a guidance system of size at most ℓ such that f_G is guided by $\mathcal{U}$. We label the elements of $\mathcal{U}$ has $U_1,\dots,U_\ell$ and mark elements of U_i by M_i. For each vertex u such that $f_G(u)\neq u$, pick an arbitrary set $U_i\in\mathcal{U}$ such that $f_G(u)$ is the unique neighbor of u in U_i, and mark u by N_i. It is easily checked that, for all $u,v\in V(G)$, we have

$$f_G(u)=v \quad\iff\quad \mathsf{Fun}^{\varphi\to f}(\Lambda(G))\models f(u)=v. \qquad\square$$

Spanning Forests. For a graph G and a function $g: V(G)\to V(G)$, we say that *gdefines a spanning forest of depth r* on G if g is guarded by G and the $(r-1)$-fold composition $g^{r-1}: V(G)\to V(G)$ is constant when restricted to each connected component of G. In particular, two vertices $u,v\in V(G)$ are in the same connected component of G if and only if $g^{r-1}(u)=g^{r-1}(v)$.

Constructing Guidable Choice Functions. We now prove that every total binary relation whose graph has bounded shrubdepth contains a guidable choice function.

Lemma 5.4. *Let $\mathscr{C}$ be a class of graphs with bounded shrubdepth. There exists $p\in\mathbb{N}$, such that for every $G\in\mathscr{C}$ and every disjoint subsets of vertices A and B of vertices of G, we have the following property: If every vertex in A has at least one neighbor in B, then there is a function $f: A\to B$ that is p-guidable on G.*

We found two conceptually different proofs of this result. We believe that both proofs describe complementary viewpoints on the problem, so we present both of them. To keep the presentation concise, in the main body of the article we give only one proof, using the characterization of classes with bounded shrubdepth by connection models and their close connection to *bi-cographs*. We present the second proof in Section A.1 of Appendix A, which describes an explicit greedy procedure leading to the construction of f.

We first prove the special case of Lemma 5.4 for bipartite graphs that have a connection model using two different labels α and β, where one part of G has label α and the other part has label β. Such graphs are called *bi-cographs* (cf. [23]).

Lemma 5.5. *Let G be a bi-cograph with parts A,B and with a connection model of depth d where vertices in A have label α and vertices in B have label β. Assume that every vertex in A has a neighbor in B. Then there is a function $f: A\to B$ that is d-guidable on G.*

Proof. By Lemma 5.1, it is enough to consider the case where G is connected. Let T be a connection model of depth d of G with A labeled α and B labeled β. Without loss of generality, we assume that every internal node of T has at least two daughters.

The proof proceeds by induction on d. The base case, when $d=1$, is trivial, because then every vertex of A is adjacent to every vertex of B, so picking any $w\in B$ the function $f: A\to B$, which maps every $v\in A$ to w is guided by the guidance system consisting only of $\{w\}$.

In the inductive step, assume that $d\geqslant 2$ and the statement holds for depth $d-1$. Since G is connected and the root has at least two daughters, the label $C(r)$ of the root r contains the pair (α,β).

Let $\mathcal{S}$ be the set of all the bipartite induced subgraphs H of G such that H is defined by the connection model rooted at some child of r in T. As $(\alpha,\beta)\in C(r)$, it follows that if $H_1,H_2\in\mathcal{S}$ are two distinct graphs, then every vertex with label α in H_1 is connected to every vertex with label β in H_2. We consider two cases, depending on whether $\mathcal{S}$ contains more than one graph containing a vertex with label β, or not.

In the first case, there are at least two graphs $H_1, H_2 \in \mathcal{S}$ such that H_1 and H_2 both contain a vertex with label β. Pick $w_1 \in V(H_1)$ and $w_2 \in V(H_2)$, both with label β. Then, every vertex in A is adjacent either to w_1 or to w_2. Let $f : A \to B$ be a function which maps a vertex $v \in A$ to w_1 if v is adjacent to w_1, and to w_2, otherwise. Then, f is guided by the guidance system consisting of $\{w_1\}$ and $\{w_2\}$.

In the second case, there is only one graph $H \in \mathcal{S}$ which contains a vertex with label β. Pick an arbitrary vertex w with label β in H. Notice that every vertex in $A \setminus V(H)$ is adjacent to w. The graph H has a connection model of depth $d-1$, so by inductive assumption, there is a guidance system $\mathcal{U} \subseteq \mathcal{P}(V(H))$ of size at most $d-1$ and a function $f_0 : V(H) \cap A \to V(H) \cap B$ which is guided by $\mathcal{U}$. Then, the function $f : A \to B$ which extends f_0 by mapping every vertex in $A \setminus V(H)$ to w is guided by $\mathcal{U} \cup \{\{w\}\}$. In either case, we have constructed a d-guidable function $f : A \to B$, as required. ◻

We now prove Lemma 5.4 in the general case.

Proof of Lemma 5.4. By assumption, $\mathscr{C}$ is a class of graphs with bounded shrubdepth. Let Γ be a finite set of labels and let $d \in \mathbb{N}$ be such that every graph $G \in \mathscr{C}$ has a connection model of depth d using labels from Γ. By doubling the number of labels, if necessary, we may assume that no label is used both in A and B. For $\alpha \in \Gamma$, let V_α denote the set of vertices of G which are labeled α.

Define a function $\mu : A \to \Gamma^2$ as follows: for every vertex v define $\mu(v)$ as (α, β), where α is the label of v, and $\beta \in \Gamma$ is an arbitrary label such that v has a neighbor in B with label β. Since no label occur both in A and B, we have have $\alpha \neq \beta$.

For every pair of distinct labels α, β, let $G_{\alpha,\beta}$ be the bipartite subgraph of G with parts $\mu^{-1}((\alpha, \beta))$ and $B \cap V_\beta$ with the same adjacencies between these parts as in G. Note the parts are disjoint as $\mu^{-1}((\alpha, \beta)) \subseteq A \cap V_\alpha$. Observe that $G_{\alpha,\beta}$ is a bi-cograph with a connection model of depth d, such that every vertex in $V(G_{\alpha,\beta}) \cap A$ has at least one neighbor in $V(G_{\alpha,\beta}) \cap B$. By Lemma 5.5, there is a function $f_{\alpha,\beta} : \mu^{-1}((\alpha, \beta)) \to B \cap V_\beta$ which is d-guidable in $G_{\alpha,\beta}$. Observe that $f_{\alpha,\beta}$ is also d-guidable when extended to G by fixing all the vertices not in $\mu^{-1}((\alpha, \beta))$ (the same guidance system will do).

Finally, define the function $f : A \to B$ so that if $v \in A$ and $\mu(v) = (\alpha, \beta)$, then $f(v) = f_{\alpha,\beta}(v)$. By Lemma 5.2, the function f is $(h \cdot |\Gamma|^2)$-guidable. This concludes the proof of Lemma 5.4. ◻

Constructing Guidable Spanning Forests. We are ready to show that shallow spanning forests on classes of bounded shrubdepth are definable by guidance systems.

Lemma 5.6. *For every class $\mathscr{C}$ of graphs with bounded shrubdepth, there exist constants $q, r \in \mathbb{N}$ such that, for every $G \in \mathscr{C}$, there is a function $f_G : V(G) \to V(G)$ which is q-guidable on G and defines a spanning forest of depth r on G.*

Proof. Let $\mathscr{C}$ be a class of graphs with bounded shrubdepth, and let r and p be constants provided by Lemma 2.8 and Lemma 5.4, respectively, for the class $\mathscr{C}$. Let $R_1 \subseteq V(G)$ be a set of vertices that contains exactly one vertex in each connected component C of G. By Lemma 2.8, we may assume that every vertex in G is at distance at most $r-1$ from a unique vertex in R_1. For $i \in \{2, \ldots, r\}$, let R_i be the set of vertices of G whose distance to some vertex in R_1 is equal to $i-1$. Then, the sets $R_1, \ldots, R_r$ form a partition of the vertex set of G. Furthermore, observe that, for $i \in \{2, \ldots, r\}$, every vertex of R_i has a neighbor in R_{i-1}.

Fix a number $i \in \{2, \ldots, r\}$. Apply Lemma 5.4 to R_i as A and R_{i-1} as B. This yields a function $f_i : R_i \to R_{i-1}$ that is p-guidable in $G[R_i \cup R_{i-1}]$. The function f_i extends as a p-guidable function $f_i : V(G) \to V(G)$.

Consider now the function $f_G : V(G) \to V(G)$ such that, for $u \in V(G)$, $f_G(u) = f_i(u)$ if $u \in R_i$ ($i \in \{2, \ldots, r\}$) and $f_G(u) = u$ if $u \in R_1$. By the first item of Lemma 5.2, we find that f_G is pr-guidable. By construction, f_G is guarded, and f_G^{r-1} maps every vertex $v \in V(G)$ to the unique vertex in R_1 that lies in the connected component of v. This proves that f_G defines a spanning forest of depth r on G. $\square$

Lemma 5.7. *Let $\mathscr{C}$ be a class of graphs with bounded shrubdepth and let g be a function symbol. There exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and a transduction F^c that is almost quantifier-free such that, for a given $G \in \mathscr{C}$, the output of F^c on $\Lambda(G)$ is equal to G expanded with a function c such that $c(v) = c(w)$ if and only if v and w are in the same connected component of G.*

Proof. By Lemma 5.3, there is a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and a transduction $\mathsf{Fun}^{\varphi \to f}$ that is almost quantifier-free such that, for every graph $G \in \mathscr{C}$, $\mathsf{Fun}^{\varphi \to f}(\Lambda(G))$ is the graph G expanded with the function f equal on G to the function f_G obtained from Lemma 5.6. Now let $g = f^{r-1}$ be the $(r-1)$-fold composition of f. Clearly, g can be computed by an almost quantifier-free transduction, by making use of the function f. As g is constant on every connected component of G, Lemma 5.7 follows. $\square$

5.2 Bi-Definability of Classes with Bounded Shrubdepth and Classes of Trees with Bounded Depth

We now prove that, up to transduction equivalence, classes with bounded shrubdepth are (basically) the same as classes of trees with bounded depth.

Let d be a positive integer, let $\mathscr{Y}_d$ be the class of all rooted forests with all leaves at depth d and monadic predicates $W_1, \ldots, W_{d-1}$ on the leaves, and let $\mathscr{S}_d$ be the class of all graphs with SC-depth at most d. We assume that the rooted forests in $\mathscr{Y}_d$ are encoded using the parent function π (see Lemma 2.2).

Lemma 5.8. *Let $d \geqslant 2$ be an integer. There exist almost quantifier-free transductions Build_d and SC_d with the following properties: For every subclasses $\mathscr{C}$ of $\mathscr{S}_d$ and $\mathscr{T}$ of $\mathscr{Y}_d$ such that $\mathscr{T}$ is a set consisting of exactly one normal SC-decomposition with depth d for each graph in $\mathscr{C}$, there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ such that $\Lambda; \mathsf{SC}_d; \mathsf{Build}_d \equiv_{\mathscr{C}} \mathsf{Id}$ and $\mathsf{Build}_d; \Lambda; \mathsf{SC}_d \equiv_{\mathscr{T}} \mathsf{Id}$. In other words, the following diagram commutes:*

$$\begin{tikzcd}
 & \mathscr{C}^+ \arrow[dl, "\mathsf{SC}_d"'] \\
\mathscr{T} \arrow[r, "\mathsf{Build}_d"] \arrow[loop left, "\mathsf{Id}_{\mathscr{T}}"] & \mathscr{C} \arrow[u, "\Lambda"'] \arrow[loop right, "\mathsf{Id}_{\mathscr{C}}"]
\end{tikzcd}$$

Proof. Recall that a normal SC-decomposition with depth d of a graph G is a rooted forest Y with all leaves at depth d and unary predicates $W_1, \ldots, W_{d-1}$ on the leaves. Let $\mathsf{Build}_d := \mathsf{Rel}^{\varsigma_d \to E}; \mathsf{Restr}^{\nu}; \mathsf{Reduct}^{\pi}$, where ς_d was defined in Lemma 2.10 and

$$\nu(x) := (\pi^{d-1}(x) \neq \pi^{d-2}(x)).$$

According to Lemma 2.10 the operation $\mathsf{Rel}^{\varsigma_d \to E}$ creates the edges of G, by flipping the adjacency within the sets W_i as described by the SC-decomposition. The meaning of Restr^{ν} is to restrict the graph to its subgraphs induced by the leaves (i.e., by its subset of vertices at depth d).

Then, $\mathrm{Build}_d(\mathscr{Y}_d) = \mathscr{S}_d$ and, for every normal SC-decomposition Y with height d of $G \in \mathscr{S}_d$, we have $G = \mathrm{Build}_d(Y)$.

Let us now describe a reverse transformation SC_d, which is the composition of several transformations. For each $G \in \mathscr{S}_d$, we consider the normal SC-decomposition Y of G with depth d that is in $\mathscr{T}$. We apply the following transformations to G:

A monadic lift Λ_Y. To each internal vertex w of Y at depth i, we associate a vertex of G that is a descendant of w. This vertex is marked with a unary predicate H_i (note that a vertex can be marked several times, but at most once for each value of i).

An almost quantifier-free transduction $\mathsf{Copy}^{(V_1,M_1),\dots,(V_{d-1},M_{d-1})}; \mathsf{Rel}^{\mu\to O}; \mathsf{Rel}^{\eta\to E}$, where

$$\mu(x) := \neg \bigvee_{i=0}^{d-1} V_i(x),$$
$$\eta(x,y) := E(x,y) \wedge O(x) \wedge O(y).$$

This transduction creates $d-1$ copies of each vertex, marked with predicates $V_1,\dots,V_{d-1}$ and linked to the original vertex with binary relations $M_1,\dots,M_{d-1}$. The original vertices of G are marked by predicate O, and the only E-edges kept are those between original vertices.

An almost quantifier-free transduction $\mathsf{Fun}^{\varphi_1\to\tau_1};\dots;\mathsf{Fun}^{\varphi_{d-1}\to\tau_{d-1}}$, where

$$\varphi_i(x,y) := M_i(x,y),$$

that defines involutive functions τ_i exchanging the original vertices with their copies in V_i.

A restriction transduction Restr^{ψ}, where

$$\psi(x) := \bigwedge_{i=1}^{d-1} V_i(x) \to H_i(\tau_i(x)),$$

that only keeps the copy of v in V_i if it corresponds to an internal vertex in Y, i.e., if v is marked H_i.

A sequence of d transformations $\Lambda_i; F^{c_i}; \mathsf{Rel}^{\varphi_i\to E}$ for $1 \leqslant i < d$, $\Lambda_i; \mathsf{F}^{c_i}$ obtained by Lemma 5.7, and

$$\varphi_i(x,y) := E(x,y) \oplus ((c_i(x) = c_i(y)) \wedge W_i(x) \wedge W_i(y)),$$

that flips the edges between the vertices marked W_i in a same connected component marked by c_i.

An almost quantifier-free transduction $\mathsf{Fun}^{\zeta\to\pi}$, where

$$\zeta(x,y) := O(x) \wedge (y = \tau_{d-1} \circ c_{d-1}(x)) \vee \bigvee_{i=2}^{d-1} V_i(x) \wedge (y = \tau_{i-1} \circ c_{i-1} \circ \tau_i(x)),$$

that constructs the parent function π in the SC-decomposition.

A reduct that removes all the symbols not present in the signature of $\mathscr{Y}$.

According to Corollary 2.4, this transformation is equivalent on $\mathscr{S}_d$ to the composition $\Lambda; \mathrm{SC}_d$ of a monadic lift Λ and an almost quantifier-free transduction SC_d (independent of the choice of $\mathscr{T}$). Moreover, it is clear that we have $\mathrm{SC}_d(\Lambda(Y)) = G$. □

5.3 Bi-Definability of Classes with Low Shrubdepth Covers and Classes with Bounded Expansion

We now prove that, up to transduction equivalence, classes with low shrubdepth covers are the same as classes with bounded expansion. As noted earlier, this will finish the proof of the right-to-left implication in Theorem 6.8.

Definition 5.9. Let $\mathscr{C}$ be a class of graphs with finite 2-cover $\mathcal{U}$ such that $\mathscr{C}[\mathcal{U}]$ has SC-depth at most d and let $\mathscr{T} \subseteq \mathscr{Y}_d$ contain exactly one normal SC-decomposition for each graph in $\mathscr{C}[\mathcal{U}]$.

The *sparsification* of $G \in \mathscr{C}$ is obtained as follows: let $\mathcal{U}_G = \{U_1, \ldots, U_N\}$ and let $Y_i \in \mathscr{F}$ be the normal SC-decomposition associated to $G[U_i]$. We rename the predicates H_j and W_j in Y_i as $H_{i,j}$ and $W_{i,j}$. Then we glue the forests Y_i on their leaves; moreover, each leaf of Y_i receives the predicate U_i. The resulting graph is the sparsification of G (with respect to $\mathcal{U}$, $\mathscr{T}$, and the numbering of $\mathcal{U}_G$).

The sparsification of $\mathscr{C}$ is the class of the sparsifications of the graphs $G \in \mathscr{C}$.

The term "sparsification" is justified by the following fact.

Lemma 5.10. *Let $\mathscr{D}$ be a sparsification of a class $\mathscr{D}$ with respect to a finite 2-cover $\mathcal{U}$ such that $\mathscr{C}[\mathcal{U}]$ has bounded shrub-depth, and let $s = \sup\{|\mathcal{U}_G| : G \in \mathscr{C}\}$.*

Then, the graphs in $\mathscr{D}$ exclude $K_{s+1,s+1}$ as a subgraph.

Proof. The class $\mathscr{D}$ is obtained by gluing N forests on their leaves (and adding some unary predicates), hence graphs in $\mathscr{D}$ are s-degenerate and in particular excludes the biclique $K_{s+1,s+1}$ as a subgraph. ☐

Lemma 5.11. *Let $\mathscr{C}$ be a class of graphs with finite 2-cover $\mathcal{U}$ such that $\mathscr{C}[\mathcal{U}]$ has bounded shrubdepth and let $\mathscr{D}$ be a sparsification of $\mathscr{C}$.*

Then there is a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$, an almost quantifier-free transduction S *and an almost quantifier-free interpretation* I*, such that the following diagram commutes:*

$$\begin{tikzcd}
\mathscr{C}^+ \arrow[dr, "\mathsf{S}"] & \\
\mathscr{C} \arrow[u, "\Lambda"] \arrow[loop below, "\mathsf{Id}_{\mathscr{C}}"] & \mathscr{D} \arrow[l, "\mathsf{I}"'] \arrow[loop below, "\mathsf{Id}_{\mathscr{D}}"]
\end{tikzcd}$$

In particular, $\mathscr{C}$ and its sparsification $\mathscr{D}$ are transduction equivalent.

Proof. Let $N = \sup\{|\mathcal{U}_G| : G \in \mathscr{C}\}$, and for $G \in \mathscr{C}$, let $\widehat{G} = G$ be the monadic lift of G by unary predicates $U_1, \ldots, U_N$ such that $\{U_1, \ldots, U_N\} = \mathcal{U}_G$. Let $\widehat{\mathscr{C}} = \{\widehat{G} : G \in \mathscr{C}\}$. Then the class $\mathscr{B} = \widehat{\mathscr{C}}[\mathcal{U}]$ has bounded shrubdepth. Let d be the maximum SC-depth of $\widehat{G} \in \mathscr{B}$, and let $\mathscr{T} \subseteq \mathscr{Y}_d$ contain exactly one normal SC-decomposition for each $\widehat{G} \in \mathscr{B}$.

According to Lemma 5.8, there exists a monadic lift $\Lambda : \mathscr{B} \to \mathscr{B}^+$ such that $\Lambda; \mathsf{SC}_d; \mathsf{Build}_d \equiv_{\mathscr{B}} \mathsf{Id}$ and $\mathsf{Build}_d; \Lambda; \mathsf{SC}_d \equiv_{\mathscr{T}} \mathsf{Id}$.

It is easy to construct an almost quantifier-free transduction S such that, for $\widehat{G} \in \widehat{\mathscr{C}}$, the structure $\mathsf{S}(\widehat{G})$ is the union of the trees $T_U = (\Lambda; \mathsf{SC}_d; \mathsf{RelTree})(G[U])$, for $U \in \mathcal{U}_G$, where the union is disjoint apart from the vertices which belong to $V(G)$ (the leaves of the trees): First, we save the adjacency relation E into a new relation E' by applying $\mathsf{Rel}^{E \to E'}$. Then, for $i = 1, \ldots, N$, we apply the transduction $\mathsf{SC}_d; \mathsf{RelTree}$ to $\Lambda(G[U_i])$, appropriately modifying all its atomic operations so that the elements outside of U_i are ignored and kept intact, we rename H_j to $H_{i,j}$ and W_j to $W_{i,j}$

(for $1 \leqslant j < d$), and then restore the adjacency relation from E' on U_i. Finally, we only keep the edge relation of the trees and, by applying a reduct, we keep only predicates U_i, $H_{i,j.}$ and $W_{i,j}$. According to Lemma 2.3, this process is equivalent to the composition of a monadic lift $\Lambda_{\mathscr{C}}$ and an almost quantifier-free transduction S. Note that the class of obtained structures is $\mathscr{D}$.

The almost quantifier-free interpretation I is obtained by first computing the parent functions π_i of each of - the forest glued together, using the fact that the set of leaves is U_i and that the levels are marked by predicates $H_{i,j}$. Then, we apply $\mathsf{Rel}^{\eta \to E}$; Restr^{ν}; Reduct, where

$$\eta(x, y) := (x \neq y) \wedge$$
$$\bigvee_{i=1}^{N} \left[U_i(x) \wedge U_i(y) \wedge \left(\pi_i^{d-1}(x) = \pi_i^{d-1}(y)\right) \wedge \bigoplus_{j=1}^{d-1} W_{i,j}(x) \wedge W_{i,j}(y) \wedge \left(\pi_i^{d-j}(x) = \pi_i^{d-j}(y)\right) \right],$$
$$\nu(x) := \bigvee_{i=1}^{N} U_i(x),$$

and where Reduct removes all symbols distinct from E (compare with the definition of Build_d). Let $\widehat{G} \in \mathscr{D}$ be the sparsification of $G \in \mathscr{C}$, and let $u, v \in V(G)$. Then, u and v will be adjacent in $\mathsf{I}(\widehat{G})$ if, for some $i \in \{1, \ldots, N\}$, they are encoded as adjacent in the SC-decomposition defined by $W_{i,j}$ and π_i on the set marked U_i. Hence, $\mathsf{I}(\widehat{G}) = G$. Thus, $\Lambda; \mathsf{S}; \mathsf{I} \equiv_C \mathsf{Id}$. □

Lemma 5.12. *Let $\mathscr{C}$ be a class of graphs with finite 2-cover $\mathcal{U}$ such that $\mathscr{C}[\mathcal{U}]$ has bounded shrubdepth and let $\mathscr{D}$ be a sparsification of $\mathscr{C}$.*

Then, $\mathscr{C}$ has low shrubdepth covers if and only if $\mathscr{D}$ has bounded expansion.

Proof. Assume $\mathscr{C}$ has low shrubdepth covers. By Lemma 4.4, since $\mathscr{D}$ is a transduction of a monadic lift of $\mathscr{C}$, it has low shrubdepth covers. According to Lemma 5.10, there is an integer s such that the graphs in $\mathscr{D}$ of exclude $K_{s,s}$ as a subgraph. By Lemma 2.12, it follows that $\mathscr{D}$ has low treedepth covers; hence, by Lemma 2.19, $\mathscr{D}$ has bounded expansion.

Now assume that $\mathscr{D}$ has bounded expansion. By Lemma 4.4, since $\mathscr{C}$ is a transduction of $\mathscr{D}$, it has low shrubdepth covers. □

6 QUANTIFIER ELIMINATION AND THE MAIN RESULTS

In this section, we prove Lemma 6.7 and deduce our two main results, namely Theorems 6.8 and 6.9.

Lemma 6.1. *Let $\mathscr{C}$ be a class of graphs that has 2-covers with bounded treedepth, and for each $G \in \mathscr{C}$, let $f_G : V(G) \to V(G)$ be a function guarded by G. Then, there is monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free function expansion $\mathsf{Fun}^{\varphi \to f}$ such that, for every $G \in \mathscr{C}$ and all $u, v \in V(G)$, we have $\mathsf{Fun}^{\varphi \to f}(\Lambda(G)) \models f(u) = v$ if and only if $f_G(u) = v$.*

Proof. We show that f_G is ℓ-guidable, for some ℓ depending only on $\mathscr{C}$. Then, the claim of the lemma follows by Lemma 5.3.

First, consider the special case when $\mathscr{C}$ is a class of treedepth d, for some $d \in \mathbb{N}$. For each $G \in \mathscr{C}$, fix a rooted forest F of depth d with $V(F) = V(G)$ such that every edge in G connects comparable nodes of F (with respect to the tree order). Label every vertex v of G by the depth of v in the forest F, using labels $\{1, \ldots, d\}$. For $1 \leqslant i \leqslant d$, let V_i be the subset of vertices at depth i in F, and let $W_i = \{f_G(u) : u \in V_i\}$. We claim that $\mathcal{U}_G = \{V_i : 1 \leqslant i \leqslant d\} \cup \{W_i : 1 \leqslant i \leqslant d\}$ is a guidance system of order $2d$ for f_G. To show this, consider a vertex $v \in V(G)$ with depth i in F, and let j be the depth of $f_G(v)$ in F. If $j < i$, then $f_G(v)$ is the unique neighbor of v in V_j, because $f_G(v)$ is an ancestor of v at depth j in F. Otherwise, $i < j$; in this case, we claim that $f_G(v)$ is the unique neighbor of v in W_i. Assume towards a contradiction that v has another neighbor $w \neq f(v)$ in W_i.

By definition of W_i, it holds that $w = f_G(u)$ for some vertex u in V_i, and this vertex u is adjacent to w because f_G is guarded by G. But this means that $u = v$, because w is adjacent to v and, for $i < j$, it holds that any vertex at depth j has at most one neighbor at depth i. This means $w = f_G(u) = f_G(v)$, which is a contradiction with our assumption that $w \neq f(v)$.

We now consider the general case, when $\mathscr{C}$ is a class that has a 2-cover $\mathcal{U}$ with bounded treedepth. Let $N = \sup\{|\mathcal{U}_G| : G \in \mathscr{C}\}$, and let d be the treedepth of the class $\mathscr{C}[\mathcal{U}]$. Let $G \in \mathscr{C}$ be a graph, $\mathcal{U}_G$ its 2-cover and $f_G : V(G) \to V(G)$ be a function that is guarded by G. Then for every $U \in \mathcal{U}_G$ the function $f_G|_U$ is $2d$-guidable by the previous case, and hence f_G is $2dN$-guidable by Lemma 5.2. □

Similarly as in [13] and [25], we first consider the restricted case of classes of colored forests with bounded depth.

Lemma 6.2. *Let $\varphi(\bar{x})$ be first-order formula and let $\mathscr{F}$ be a class of colored rooted forests with bounded depth. Let* $\mathsf{FunTree}$ *be the almost quantifier-free guarded transduction from Lemma 2.2. Then, there exists a monadic lift $\Lambda : \mathscr{F} \to \mathscr{F}^+$ and a quantifier-free formula $\varphi'(\bar{x})$ such that, for all $F \in \mathscr{F}$, we have $\varphi(F) = \varphi'(\mathsf{FunTree}(\Lambda(F))$.*

We present the proof of Lemma 6.2 in Section A.2 of Appendix A, which uses an automata based approach that is conceptually different from the approaches of [13] and [25].

The next step is to lift Lemma 6.2 to classes of structures with bounded treedepth. We first observe that classes of bounded treedepth are bi-definable with classes of forests with bounded depth using almost quantifier-free transductions. This result is similar, but much simpler to prove than Lemma 5.8, which is an analogous statement for classes with bounded shrubdepth.

The proof of the next lemma follows the well-known encoding of structures with bounded treedepth inside colored forests, where a structure $\mathbf{A} \in \mathscr{C}$ is encoded in a depth-first search forest of its Gaifman graph. Recall that a *depth first-search* (DFS) forest of a graph G is a rooted forest F which is a subgraph of G, such that every edge of G connects an ancestor with a descendant in F.

Lemma 6.3. *Let Σ be a signature and let $\mathscr{C}$ be a class of Σ-structures with bounded treedepth. There is a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$, a class $\mathscr{F}$ of colored rooted forests with bounded depth, and a pair of transductions $\mathsf{T} : \mathscr{C}^+ \to \mathscr{F}$ and $\mathsf{S} : \mathscr{F} \to \mathscr{C}$ such that $\Lambda; \mathsf{T}; \mathsf{S} \equiv_{\mathscr{C}} \mathsf{Id}$. Moreover, the transduction T is guarded, and S is almost quantifier-free and faithful. In other words, the following diagram commutes:*

$$\begin{tikzcd}
\mathscr{C}^+ \arrow[dr, "\mathsf{T}"] & \\
\mathscr{C} \arrow[u, "\Lambda"] \arrow[loop left, "\mathsf{Id}_{\mathscr{C}}"] & \mathscr{F} \arrow[l, "\mathsf{S}"'] \arrow[loop right, "\mathsf{Id}_{\mathscr{F}}"]
\end{tikzcd}$$

Proof. By Lemma 2.6, the maximal order of a path in a graph with treedepth d is 2^d. Hence, a graph G of treedepth at most d has a DFS forest of depth at most 2^d. The monadic lift Λ adds 2^d new unary predicates H_i ($i \in [2^d]$) to each $\mathbf{A} \in \mathscr{C}$, encoding the depth of vertices in a DFS-forest of the Gaifman graph of $\mathbf{A}$.

The transduction T is the composition of the following transductions:

- a quantifier-free transduction computing the adjacency relation E_G of the Gaifman graph of $\mathbf{A}$,
- the almost quantifier-free transduction $\mathsf{Fun}^{\varphi \to \pi}$, where $\varphi(x, y) := \bigvee_{i=2}^{2^d} H_i(x) \wedge H_{i-1}(y) \wedge E_G(x, y)$, computing the parent function π in the forest encoded by $H_1, \ldots, H_{2^d}$,

- for each Σ-structure $\mathbf{F}$ with $V(\mathbf{F}) = [i]$ and $i \leqslant 2^d$, the almost quantifier-free transduction $\mathsf{Rel}^{\psi_{\mathbf{F}} \to I_{\mathbf{F}}}$, where $\psi_{\mathbf{F}}(x)$ asserts that x has depth i in the DFS forest and $j \mapsto \pi^{j-1}(x)$ is an isomorphism of $\mathbf{F}$ and $\mathbf{A}[\{x, \pi(x), \ldots, \pi^i(x)\}]$.
- A transduction deleting all the relations but E_G (renamed E) and all the unary relations H_i and $I_{\mathbf{F}}$.

The almost quantifier-free transduction S is the composition of:

- the almost quantifier-free transduction $\mathsf{Fun}^{\varphi \to \pi}$, where $\varphi(x) := \bigvee_{i=2}^{2^d} H_i(x) \wedge H_{i-1}(y) \wedge E_G(x, y)$, computing the parent function π in the forest encoded by $H_1, \ldots, H_{2^d}$,
- for each relation R with arity k in the signature of $\mathbf{A}$ the quantifier-free transduction $\mathsf{Rel}^{\rho_R \to R}$, where

$$\rho_R(x_1, \ldots, x_k) := \bigvee_{\sigma \in \mathfrak{S}_k} \bigvee_{1 \leqslant h_1 < \ldots < h_k \leqslant 2^d} \bigvee_{\mathbf{F} \models R(h_{\sigma(1)}, \ldots, h_{\sigma(k)})} \bigwedge_{j=1}^{k} H_{h_{\sigma(j)}(x_j)} \wedge I_{\mathbf{F}}(x_{\sigma^{-1}(k)}),$$

  and $\mathfrak{S}_k$ denotes the permutation group on $[k]$.
- for each function f in the signature of $\mathbf{A}$ the quantifier-free transduction $\mathsf{Fun}^{\rho_f \to f}$, where

$$\rho_f(x_1, x_2) := \bigvee_{\sigma \in \mathfrak{S}_2} \bigvee_{1 \leqslant h_1 < h_2 \leqslant 2^d} \bigvee_{\mathbf{F} \models f(h_{\sigma(1)}) = h_{\sigma(2)}} H_{h_{\sigma(1)}}(x_1) \wedge H_{h_{\sigma(2)}}(x_2) \wedge I_{\mathbf{F}}(x_{\sigma^{-1}(2)}).$$

□

Using Lemma 6.3, we easily lift the quantifier-elimination result from forests with bounded depth to classes of low treedepth.

Lemma 6.4. *Let $\varphi(\bar{x})$ be first-order formula and let $\mathscr{C}$ be a class of structures with bounded treedepth. Then there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$, an almost quantifier-free guarded transduction* S *and a quantifier-free formula $\varphi'(\bar{x})$ such that, for all $\mathbf{A} \in \mathscr{C}$, we have $\varphi(\mathbf{A}) = \varphi'(\mathsf{S}(\Lambda(\mathbf{A}))$.*

Proof. Let $\Lambda_1, \mathsf{S}_1, \mathsf{T}_1$, and $\mathscr{F}$ be as in Lemma 6.3. Since $\mathsf{S}_1(\mathsf{T}_1(\Lambda_1(\mathbf{A}))) = \mathbf{A}$, there is a formula $\psi(\bar{x})$ such that for all $\mathbf{A} \in \mathscr{C}$ we have $\varphi(\mathbf{A}) = \psi(\mathsf{T}_1(\Lambda_1(\mathbf{A})))$.

Now, apply Lemma 6.2 to the class $\mathscr{F}$ and the formula $\psi(\bar{x})$, yielding a monadic lift Λ_2 and a quantifier-free formula $\varphi'(\bar{x})$, such that for all $F \in \mathscr{F}$ we have $\psi(F) = \varphi'(\mathsf{FunTree}(\Lambda_2(F)))$. By composition, for all $\mathbf{A} \in \mathscr{C}$ we have $\varphi(\mathbf{A}) = \varphi'(\mathsf{FunTree}(\Lambda_2(\mathsf{T}_1(\Lambda_1(\mathbf{A})))))$. As T_1 is faithful almost quantifier-free and does not use any unary predicate defined by Λ_2 we have $\mathsf{T}_1; \Lambda_2 = \Lambda_2; \mathsf{T}_1$. Thus we have $\varphi(\mathbf{A}) = \varphi'((\mathsf{T}_1; \mathsf{FunTree})(\Lambda_2 \circ \Lambda_1(\mathbf{A}))$. Let $\mathsf{S} := \mathsf{T}_1; \mathsf{FunTree}$ and $\Lambda := \Lambda_2 \circ \Lambda_1$. Since T_1 and FunTree are guarded so is S, and we have $\varphi(\mathbf{A}) = \varphi'(\mathsf{S}(\Lambda(\mathbf{A})))$, as required. □

Finally, we lift the quantifier elimination procedure to classes with low shrubdepth covers using Lemma 4.3 and a reasoning very similar to the proof of Lemma 4.4. Again, conceptually this lift is exactly what is happening in [13], [25], however, our approach based on covers makes it quite straightforward. The key observation is encapsulated in Lemma 6.5 below.

Recall from Section 2.1 that if we consider a structure $\mathbf{A}$, subset U of $V(\mathbf{A})$ and $a \in U$ such that $f(a) = b$, where $b \notin U$, then in $\mathbf{A}[u]$ we define $f(a)$ to be a. Note that this means that, if $f(a) = a$ in $\mathbf{A}[U]$, then this may be the case because of two different reason – either it was true that $f(a) = a$ already in $\mathbf{A}$, or $f(a) = b$ in $\mathbf{A}$ and $f(a)$ was 'redefined' because we restricted $\mathbf{A}$ to $\mathbf{A}[U]$. In what follows we want to be able to distinguish between these two cases, and to this end we introduce a transduction which marks all elements of $\mathbf{A}$ for which it holds $f(a) = a$ in $\mathbf{A}$. For a (usually implicit) signature Σ with only relations and unary functions, let MarkFix be the quantifier-free transduction $\mathsf{Rel}^{\zeta_{f_1} \to Z_{f_1}}; \ldots \mathsf{Rel}^{\zeta_{f_\ell} \to Z_{f_l}}$, where $f_1, \ldots, f_\ell$ are the function symbols in Σ and $\zeta_f(x) := (f(x) = x)$. Note that MarkFix can be seen as a monadic lift.

LEMMA 6.5. *Let $\mathscr{D}$ be a class of structures. Let $\varphi(\bar{x})$ be a quantifier-free formula with p free variables, involving c distinct terms (including subterms). Then there is a quantifier-free formula $\varphi'(\bar{x})$ such that for every structure $\mathbf{A} \in \mathscr{D}$, every c-cover $\mathcal{U}_{\mathbf{A}}$ of the Gaifman graph of* $\mathbf{A}$*, and all tuples $\bar{a} \in V(\mathbf{A})^p$, we have*

$$\mathbf{A} \models \varphi(\bar{a}) \Leftrightarrow \exists U \in \mathcal{U}_G : \bar{a} \in U^p \text{ and } \mathsf{MarkFix}(\mathbf{A})[U] \models \varphi'(\bar{a}).$$

PROOF. Let $\widehat{\varphi}$ be the conjunction, over all terms in φ (including subterms) of the form $f(t)$ of the formula $(f(t) = t) \leftrightarrow Z_f(t)$ and let $\varphi' = \varphi \wedge \widehat{\varphi}$. Note that $\widehat{\varphi}$ verifies that all the predicates Z_f mark the fixed points of f on every term evaluated by φ. Hence, for all $\bar{a}$ and every subset U of $V(\mathbf{A})$ containing $\bar{a}$, the formula $\widehat{\varphi}$ is satisfied in $\mathsf{MarkFix}(\mathbf{A})[U]$ when $\bar{x}$ is interpreted as $\bar{a}$ if and only if all the terms evaluated in $\varphi(\bar{a})$ are in U.

Thus, if there exists $U \in \mathcal{U}_G$ such that $\bar{a} \in U^p$ and $\mathsf{MarkFix}(\mathbf{A})[U] \models \varphi'(\bar{a})$, then all the terms in φ evaluate to the same values in $\mathbf{A}[U]$ and $\mathbf{A}$, and hence $\mathbf{A} \models \varphi(\bar{a})$.

Conversely, if $\mathbf{A} \models \varphi(\bar{a})$, then there exists some $U \in \mathcal{U}_G$ containing the c elements to which the terms in $\varphi(\bar{a})$ evaluate, thus $\mathsf{MarkFix}(\mathbf{A})[U] \models \widehat{\varphi}(\bar{a})$, and thus $\mathsf{MarkFix}(\mathbf{A})[U] \models \varphi'(\bar{a})$ as well. □

LEMMA 6.6. *Let $\varphi(\bar{x})$ be a formula and let $\mathscr{C}$ be a class of structures with bounded expansion. Then there is a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$, a guarded transduction* T*, which adds only unary functions, and a quantifier-free formula $\varphi'(\bar{x})$, such that for every $\mathbf{A} \in \mathscr{C}$ we have $\varphi(\mathbf{A}) = \varphi'(\mathsf{T}(\Lambda(\mathbf{A})))$.*

PROOF. The proof proceeds by induction on the structure of the formula $\varphi(\bar{x})$. In the base case, $\varphi(\bar{x})$ is a quantifier-free formula, so we may take T to be the identity transduction.

In the inductive step, we consider two cases. If φ is a Boolean combination of simpler formulas, then the statement follows immediately from the inductive assumption. Thus, it only remains to prove the claim in the case when $\varphi(\bar{x})$ is of the form $\exists y.\psi(\bar{x}, y)$, for some formula $\psi(\bar{x}, y)$.

Applying the inductive assumption to the formula $\psi(\bar{x}, y)$ yields a monadic lift $\Lambda_\psi : \mathscr{C} \to \mathscr{C}^+$, a guarded transduction T_ψ and a quantifier-free formula $\psi'(\bar{x}, y)$. Let c be the number of distinct terms (including subterms) appearing in the formula $\psi'(\bar{x}, y)$. Let $\mathscr{D} = \mathsf{T}_\psi(\Lambda_\psi(\mathscr{C}))$. As T_ψ is faithful (since it is guarded), any finite c-cover $\mathcal{U}$ of the class $\mathscr{C}^{\text{Gaifman}}$ of the Gaifman graphs of the structures in $\mathscr{C}$ is also a finite c-cover of the class $\mathscr{D}^{\text{Gaifman}}$ of the Gaifman graphs of the structures in $\mathscr{D}$. As T_ψ is guarded, it follows that if $\mathscr{C}^{\text{Gaifman}}[\mathcal{U}]$ has bounded treedepth, then so has the class $\mathscr{D}^{\text{Gaifman}}[\mathcal{U}]$.

Apply Lemma 6.5 to $\mathscr{D}$ and $\psi'(\bar{x}, y)$, yielding a formula $\psi''(\bar{x}, y)$ such that, for every structure $\mathbf{A} \in \mathscr{C}$, every tuple of vertices $(\bar{a}, b)$, and every c-cover $\mathcal{U}_{\mathbf{A}}$ of $\mathbf{A}$, the following equivalences hold:

$$\begin{aligned} \mathbf{A} \models \psi(\bar{a}, b) &\iff \mathsf{T}_\psi(\Lambda_\psi(\mathbf{A})) \models \psi'(\bar{a}, b) \\ &\iff \mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U] \models \psi''(\bar{a}, b) \text{ for some } U \in \mathcal{U}_{\mathbf{A}} \text{ containing } \bar{a}, b, \end{aligned}$$

where $\mathsf{S}_\psi := \mathsf{T}_\psi; \mathsf{MarkFix}$.

Apply Lemma 6.4 to the class $\mathscr{D}[\mathcal{U}]$ and the formula $\exists y.\psi''(\bar{x}, y)$, yielding a monadic lift Λ', a guarded transduction F, and a quantifier-free formula $\rho(\bar{x})$ such that, for every $\mathbf{B} \in \mathscr{D}[\mathcal{U}]$ and tuple $\bar{b} \in V(\mathbf{B})^{|\bar{x}|}$,

$$\mathbf{B} \models \exists y.\psi''(\bar{b}, y) \iff \mathsf{F}(\Lambda'(\mathbf{B})) \models \rho(\bar{b}).$$

□

CLAIM 1. *For each structure $\mathbf{A} \in \mathscr{C}$ and tuple $\bar{a} \in V(\mathbf{A})^{|\bar{x}|}$, the following conditions are equivalent:*

(1) $\mathbf{A} \models \exists y.\psi(\bar{a}, y)$,
(2) there is some $U \in \mathcal{U}_{\mathbf{A}}$ containing $\bar{a}$ such that $\mathsf{F}(\Lambda'(\mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U])) \models \rho(\bar{a})$.

Proof. We have the following equivalences:

$$\begin{aligned}\mathbf{A} \models \exists y.\psi(\bar{a}, y) &\iff \mathbf{A} \models \psi(\bar{a}, b) \textit{ for some } b \in V(\mathbf{A})\\ &\iff \mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U] \models \psi''(\bar{a}, b) \textit{ for some } U \in \mathcal{U}_\mathbf{A} \textit{ containing } \bar{a}, b\\ &\iff \mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U] \models \exists y.\psi''(\bar{a}, y) \textit{ for some } U \in \mathcal{U}_\mathbf{A} \textit{ containing } \bar{a}\\ &\iff \mathsf{F}(\Lambda'(\mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U])), \models \rho(\bar{a}) \textit{ for some } U \in \mathcal{U}_\mathbf{A} \textit{ containing } \bar{a}.\end{aligned}$$

This proves the claim. ⌟

Let $N = \sup\{|\mathcal{U}_\mathbf{A}| : \mathbf{A} \in \mathscr{C}\}$. For each structure $\mathbf{A} \in \mathscr{C}$, fix an enumeration $U_1, \dots, U_N$ of the cover $\mathcal{U}_\mathbf{A}$.

Claim 2. *There is a monadic lift Λ_φ, a guarded transduction S_φ, and quantifier-free formulas $\rho_1(\bar{x}), \dots, \rho_N(\bar{x})$ such that given a structure $\mathbf{A} \in \mathscr{C}$, a number $i \in [N]$ and a tuple $\bar{a}$ of elements of U_i,*

$$\mathsf{S}_\varphi(\Lambda_\varphi(\mathbf{A})) \models \rho_i(\bar{a}) \iff \mathsf{F}(\Lambda'(\mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U_i])) \models \rho(\bar{a})$$

Proof. We have

$$\mathsf{F}(\Lambda'(\mathsf{S}_\psi(\Lambda_\psi(\mathbf{A}))[U_i])) \models \rho(\bar{a}) \iff \mathsf{F}(\mathsf{S}_\psi(\Lambda' \circ \Lambda_\psi(\mathbf{A}))[U_i]) \models \rho(\bar{a}),$$

as the transduction S_ψ and the restriction to U_i involve no copy. Let $\Lambda_\mathcal{U}$ be a monadic lift that introduces unary predicates marking the sets $U_1, \dots, U_N$, and let $\Lambda_\varphi = \Lambda_\mathcal{U} \circ \Lambda' \circ \Lambda_\psi$. Let $\mathscr{C}^* = \Lambda_\varphi(\mathscr{C})$.

We construct a guarded transduction S_φ which, given a structure $\mathbf{A}^* \in \mathscr{C}^*$, first applies the guarded transduction S_ψ, then, for each such unary predicate U_i, applies to the structure $\mathsf{S}_\psi(\mathbf{A}^*)[U_i]$ the transduction F, modified so that each function symbol f is replaced by a new function symbol f^i.

Then, the formula $\rho_i(\bar{x})$ is obtained from the formula $\rho(\bar{x})$, by replacing each function symbol f by the function symbol f^i. ⌟

Let $\varphi' = \bigvee_{i=1}^N \rho_i$. Combining Claim 1 and Claim 2, we get the following equivalence:

$$\mathsf{S}_\varphi(\Lambda_\varphi(\mathbf{A})) \models \varphi'(\bar{a}) \iff \mathbf{A} \models \varphi(\bar{a}),$$

concluding the inductive step.

Lemma 6.7. *Let $\mathscr{C}$ be a class of graphs with bounded expansion and let S be a transduction. Then, there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free transduction T such that $\mathsf{S}(G) = \mathsf{T}(\Lambda(G))$ for all $G \in \mathscr{C}$.*

Proof. For simplicity, we assume that the signature produced by S consists of one relation P; lifting the proof to signatures containing more relation and function symbols is immediate. By Lemma 2.5, we may express S as

$$\mathsf{S} = \mathsf{C}; \mathsf{E}; \mathsf{X}; \mathsf{R},$$

where

- C is a sequence of copying operations,
- $\mathsf{E} = \mathsf{Rel}^{\varphi \to P}$ is a single expansion operation,
- $\mathsf{X} = \mathsf{Restr}^{\psi}$ is a single restriction operation, where ψ that does not use the symbol P, and
- R is a sequence of reduct operations that drop all relations and functions apart from P.

From Lemma 2.13, it follows that the class $\mathscr{D} = \mathsf{C}(\mathscr{C})$ of colored graphs is a class with bounded expansion. Therefore, we may apply Lemma 6.6 to it, and to the formulas $\varphi(\bar{x})$ and $\psi(x)$ considered above. This yields monadic lifts $\Lambda_\varphi : \mathscr{D} \to \mathscr{D}^+_\varphi$ and $\Lambda_\psi : \mathscr{D} \to \mathscr{D}^+_\psi$, transductions T_φ and T_ψ that are guarded and that add only unary functions, and quantifier-free formulas $\varphi'(\bar{x})$ and $\psi'(x)$ such that, for all $G \in \mathscr{D}$, we have $\varphi(G) = \varphi'(\mathsf{T}_\varphi(\Lambda_\varphi(G))$ and $\psi(G) = \psi'(\mathsf{T}_\psi(\Lambda_\psi(G))$. We may assume that the unary predicate symbols introduced by Λ_φ and Λ_ψ, as well as the function symbols introduced by T_φ and T_ψ are different; hence, we may combine the monadic lifts into $\Lambda_1 : \mathscr{D} \to \mathscr{D}^+_1$ and the transductions into the guarded transduction T_1, so that $\varphi(G) = \varphi'(\mathsf{T}_1(\Lambda_1(G))$ and $\psi(G) = \psi'(\mathsf{T}_1(\Lambda_1(G))$.

Using Lemma 6.1 on every function expansion $\mathsf{Fun}^{\varphi \to f}$ appearing in T_1 yields a monadic lift Λ_f and an almost quantifier-free function expansion $\mathsf{Fun}^{\varphi' \to f}$ that introduces the same function. Using Corollary 2.4, we can pull all monadic lifts to the front to obtain one monadic lift $\Lambda_2 : \mathscr{D} \to \mathscr{D}^+_2$ and an almost quantifier-free transduction T_2 such that, for all $G \in \mathscr{D}$, we have $\varphi'(\mathsf{T}_1(\Lambda_1(G)) = \mathsf{T}_2(\Lambda_2(G))$ and $\psi'(\mathsf{T}_1(\Lambda_1(G)) = \psi'(\mathsf{T}_2(\Lambda_2(G))$. Finally, it is easy to modify Λ_2 and T_2 to obtain a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free transduction T such that $\mathsf{S}(G) = \mathsf{T}(\Lambda(G))$ for all $G \in \mathscr{C}$ as desired. □

We now are able to prove our main results.

Theorem 6.8. *A class of (colored) graphs has structurally bounded expansion if and only if it has low shrubdepth covers.*

Proof. As observed in Section 3, the right-to-left implication of Theorem 6.8 follows from Lemmas 5.11 and 5.12. We now show the left-to-right implication.

Let $\mathscr{C}$ be a class with bounded expansion and let S be a transduction that outputs colored graphs. We show that $\mathscr{D} := \mathsf{S}(\mathscr{C})$ has low shrubdepth covers.

By Lemma 2.19, $\mathscr{C}$ has low treedepth covers. Applying Lemma 6.7 yields a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free transduction T such that $\mathsf{S}(G) = \mathsf{T}(\Lambda(G))$ for all $G \in \mathscr{C}$, hence, $\mathscr{D} = \mathsf{S}(\mathscr{C}) = \mathsf{T}(\Lambda(\mathscr{C}))$. As $\mathscr{C}$ has low treedepth covers, also $\Lambda(\mathscr{C})$ has low treedepth covers, and hence in particular $\Lambda(\mathscr{C})$ has low shrubdepth covers (cf. Lemma 2.11 (3)), we may apply Lemma 4.4 to T and $\Lambda(\mathscr{C})$ to deduce that $\mathscr{D}$ has low shrubdepth covers. □

Theorem 6.9. *Let $\mathscr{C}$ be a class of (colored) graphs that has low shrubdepth covers. Then, for every transduction* S, *there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and an almost quantifier-free transduction* T *such that* $\mathsf{S} \equiv_{\mathscr{C}} \Lambda; \mathsf{T}$.

Proof. Lemmas 5.11 and 5.12 allow to reduce the theorem to the case of classes with bounded expansion, as almost quantifier-free transductions are closed under composition. The case with bounded expansion classes is handled by Lemma 6.7. □

7 ALGORITHMIC ASPECTS

In this section, we discuss algorithmic aspects of classes with structurally bounded expansion. Essentially, given a class $\mathscr{C}$ with structurally bounded expansion, our main computational tasks are to compute on input $G \in \mathscr{C}$ a low shrubdepth cover $\mathcal{U}_G$ of G and, for each $U \in \mathcal{U}_G$, an SC-decomposition with bounded depth of $G[U]$. Then, first-order definable problems reduce to the well understood case of classes with bounded expansion. We first discuss an efficient version of Lemma 6.6 and show how to derive the model-checking result of Dvořák, Král', and Thomas for classes with bounded expansion in Section 7.1. We then discuss algorithmic aspects of classes with structurally bounded expansion in Section 7.2. Finally, in Section 7.3, we discuss our results on a non-trivial example of classes with structurally bounded expansion.

7.1 Bounded Expansion

In this section, we note that we can easily derive the result of Dvořák, Kráľ, and Thomas, by observing that the above proof of Lemma 6.6 is effective. For each class $\mathscr{C}$ with bounded expansion we can in linear time compute a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and a guarded transduction T, which is a linear time computable function, as described in the lemma.

We say that a transduction T is a *linear-time* transduction if there is an algorithm that, given a structure $\mathbf{A}$ as input, produces the structure $\mathbf{B} = \mathsf{T}(\mathbf{A})$ in linear time. Here, the structure $\mathbf{A}$ is represented using the adjacency list representation, i.e., for a colored graph, the size of the description is linear in the sum of the number of vertices and the number of edges in the graph.

We show the following effective variant of Lemma 6.6.

Lemma 7.1. *Let $\varphi(\bar{x})$ be a formula and let $\mathscr{C}$ be a class of structures with bounded expansion. Then, there is a linear time computable monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$ and a guarded linear time transduction T, which adds only unary functions, and a quantifier-free formula $\varphi'(\bar{x})$ computable from φ and Λ, such that for every $\mathbf{A} \in \mathscr{C}$, we have $\varphi(\mathbf{A}) = \varphi'(\mathsf{T}(\Lambda(\mathbf{A})))$.*

Proof. To prove Lemma 7.1, we observe that monadic lift Λ is linear time computable and the transduction T in Lemma 6.6 is a linear-time transduction. The proof follows by tracing the proof of Lemma 6.6, and observing the following:

(1) In Lemma 6.2, the monadic lift adds unary predicates, each of which can be computed in linear time, since each predicate is produced by running a deterministic threshold tree automaton on the input tree. The transduction used in the lemma is the almost quantifier-free guarded $\mathsf{FunTree}$ transduction that adds the parent function, which is, given a rooted forest, clearly linear-time computable.
(2) In Lemma 6.3, we have to compute the monadic lift and the transduction T. Computationally, this amounts to computing the Gaifman graph of the input structure and running a depth-first search on it. Both can be done in linear time on classes with bounded treedpeth.
(3) In Lemma 6.4, the produced lift is a composition of two linear-time computable lifts, and the transduction $\mathsf{S} = \mathsf{T}; \mathsf{FunTree}$ is a linear-time transduction, as a composition of two linear-time transductions.
(4) In the proof of Lemma 6.6, the nontrivial step is in the inductive step, in the case of an existential formula. In this case, the constructed monadic lift amounts to introducing unary predicates denoting the elements of a cover $\mathcal{U}_G$ and marking the fixed-points of all functions. The transduction S_φ is a linear-time transduction, assuming $\mathscr{C}$ has bounded expansion, as it amounts to applying transductions T_ψ and F which are linear-time transductions, respectively, by the inductive assumption, and by the effective version of Lemma 6.4 discussed above.
(5) If $\mathscr{C}$ has bounded expansion, then, for any fixed $p \geqslant 0$, there is a finite p-cover $\mathcal{U}$ of $\mathscr{C}$ with bounded treedepth such that $\mathcal{U}_G$ can be computed from a given $G \in \mathscr{C}$ in time $f(p) \cdot |V(G)|$, for some function f depending on $\mathscr{C}$ (the function f may not be computable). To compute $\mathcal{U}_G$, we may first compute a $g(p)$-treedepth coloring of G for some function g (as required in the proof of Lemma 2.19) and observe that it can be converted to a cover in linear time, as in the proof of Lemma 2.19. A p-treedepth coloring can be computed in linear time, cf. [12], [35], and [36]. □

The model checking result of Dvořák, Kráľ, and Thomas then follows from the linear algorithm for subgraph isomorphism problem [35].

7.2 Structurally Bounded Expansion

In this section, we give a preliminary result about efficient computability of transductions on classes with structurally bounded expansion. When we refer to the size of a structure in the algorithmic context, we refer to its total size, i.e., the sum of its universe size and the total sum of sizes of tuples in its relations.

Call a class $\mathscr{C}$ of graphs of structurally bounded expansion *efficiently decomposable* if there is a finite 2-cover $\mathcal{U}$ of $\mathscr{C}$ and a polynomial-time algorithm that, given a graph $G \in \mathscr{C}$, computes the cover $\mathcal{U}_G$. Denote by $P^{\mathscr{C}}_{\text{cov}}(n)$ a polynomial such that the above algorithm runs in $P^{\mathscr{C}}_{\text{cov}}(n)$-time.

In what follows, we will use the fact that, for each class $\mathscr{D}$ with bounded shrubdepth, there exists a $O(n^3)$-time algorithm to compute an SC-decomposition of depth at most d of H, where d is the upper bound on the depth of minimum height SC-decomposition of any graph from $\mathscr{D}$ (this bound exists by Lemma 2.11). To see this, consider a MSO formula φ which verifies whether a graph G has SC-decomposition of depth d by first guessing sets $W_1, \ldots, W_{d-1}$ and then, following the definition of SC-depth, checks whether consecutive flipping of adjacency within W_i in respective connected components eliminates all edges from G. Since $\mathscr{D}$ has bounded shrub depth, it also has bounded clique-width, and so, by the results of [37] and [6], one can evaluate the above formula on any G from $\mathscr{D}$ in cubic time (the formula will always evaluate to true), and moreover one can extract sets $W_1, \ldots, W_{d-1}$ which certify that G has SC-depth at most d and use them to construct an SC-decomposition of G.

Our result is as follows:

Theorem 7.2. *Let $\varphi(\bar{x})$ be a transduction and bet $\mathscr{C}$ be a class of graphs with structurally bounded expansion that is efficiently decomposable. Then, given a graph $G \in \mathscr{C}$ of order n, after a preprocessing step running in time $P^{\mathscr{C}}_{\text{cov}}(n) + O(n^3)$, we can output the number of tuples satisfying φ in G and enumerate them with constant delay.*

Proof. By efficient decomposability of $\mathscr{C}$, in time $P^{\mathscr{C}}_{\text{cov}}(n)$, we can compute a cover $\mathcal{U}_G$ of G (of order n), and in time $O(n^3)$, we can compute an SC-decomposition S_U of depth at most d of $G[U]$, for $U \in \mathcal{U}_G$. From this, we can compute in linear time a sparsification G' of G. Denote by $\mathscr{D}$ the class of all the sparsifications of graphs $G \in \mathscr{C}$.

According to Lemma 5.11, there exists a monadic lift $\Lambda : \mathscr{C} \to \mathscr{C}^+$, an almost quantifier-free transduction S and an almost quantifier-free interpretation I, such that the following diagram commutes:

$$\begin{tikzcd}
\mathscr{C}^+ \arrow[dr, "\mathsf{S}"] & \\
\mathscr{C} \arrow[u, "\Lambda"] \arrow[loop below, "\mathsf{Id}_{\mathscr{C}}"] & \mathscr{D} \arrow[l, "\mathsf{I}"'] \arrow[loop below, "\mathsf{Id}_{\mathscr{D}}"]
\end{tikzcd}$$

By tracing the proof of Lemma 5.11, we observe that Λ is linear-time computable from the cover $\mathcal{U}_G$ of G and the SC-decompositions S_U of depth at most d of $G[U]$, for $U \in \mathcal{U}_G$. It is also easy to verify that the transductions S and I are linear-time computable, as they are almost quantifier-free.

By assumption, $\mathscr{C}$ has low shrubdepth covers, and hence, by Lemma 5.12, $\mathscr{D}$ has bounded expansion.

We will make use of transductions S and I as follows: We have $\mathsf{I}(\mathsf{S}(\Lambda(G))) = G$ for each $G \in \mathscr{C}$, hence, $\varphi(G) = \varphi(\mathsf{I}(\mathsf{S}(\Lambda(G))))$. Define ψ from φ by replacing every relation and function defined by I (I does not use copying) by the defining formula. We then have $\varphi(G) = \psi(\mathsf{S}(\Lambda(G)))$. The class

$\mathscr{D} = \mathsf{S}(\Lambda(\mathscr{C}))$ is a class of colored graphs with bounded expansion; hence, we can now apply the counting and enumeration result of Kazana and Segoufin for classes with bounded expansion [29] to the colored graph $\mathsf{S}(\Lambda(G))$ and to the formula ψ. $\square$

We remark that, instead of efficient decomposability, we could assume that the 2-cover $\mathcal{U}_G$ of a graph G and corresponding SC-decompositions for all $U \in \mathcal{U}_G$ is given together with G as input.

7.3 A Non-Trivial Example

Let $\mathscr{P}$ be a class of graphs. We do not require membership to $\mathscr{P}$ to be effectively decidable. Let f be the graph transformation such that for every graph G the graph $f(G)$ has same vertex set as G and distinct vertices u, v are adjacent in $f(G)$ if $G[N(u) \cap N(v)] \in \mathscr{P}$. Let $\mathscr{C}$ be a class with bounded expansion and let $\mathscr{D} = f(\mathscr{C})$.

Proposition 7.3. *The class $\mathscr{D}$ has structurally bounded expansion.*

Proof. Let $\mathcal{U}$ be a depth-3 cover of $\mathscr{C}$ with bounded treedepth (say d), let $N = \sup\{|\mathcal{U}_G : G \in \mathscr{C}\}$, let $\mathscr{P}_0 = \{H \in \mathscr{P} : |V(H)| \leqslant 2^d N\}$. Let $\mathsf{P}_0 = \mathsf{Fun}^{\varphi_1 \to \pi_1}; \ldots; \mathsf{Fun}^{\varphi_N \to \pi_N}$, where $\mathsf{Fun}^{\varphi_i \to \pi_i}$ computes the parent function for a DFS tree in $G[U_i]$. Let

$$\alpha(x, y) := \bigvee_{i=1}^{N} \bigvee_{j=1}^{2^d - 1} \left(f_i^j(x) = y\right) \vee \left(f_i^j(y) = x\right),$$

$$\beta(x, y) := \bigvee_{i=1}^{N} \bigvee_{j=1}^{2^d - 1} \left(f_i^j(x) = y\right) \wedge G_{i,j}(x) \vee \left(f_i^j(y) = x\right) \wedge G_{i,j}(y),$$

and let $\gamma(x, y)$ express that the common ancestors of x and y in the union of the forest defined by $\pi_1, \ldots, \pi_i$ induce a graph isomorphic to some $H \in \mathscr{P}_0$. We define $\mathsf{T}_0 = \mathsf{P}; \mathsf{Rel}^{\psi \to E}; \mathsf{Reduct}$, where $\psi(x, y) := (x \neq y) \wedge (\beta(x, y) \vee \neg\alpha(x, y) \wedge \gamma(x, y))$.

Let Λ_0 be the lift defining unary predicates $H_{i,j}$ (vertices at depth j in the ith forest) and $G_{i,j}$ (marking vertices u such that $G[N(u) \cap N(f_i^j(u))] \in \mathscr{P}$).

Then, one checks that, for every graph $G \in \mathscr{C}$, we have $f(G) = \mathsf{T}_0(\Lambda_0(G))$. It follows that $\mathscr{D}$ has structurally bounded expansion. $\square$

Note the monadic lift Λ_0 and the transduction T_0 are not necessarily computable. However, we have some bound on the number of predicates added by Λ_0 and the quantifier rank of the formulas used in T_0. It follows that if the class $\mathscr{C}$ has known bounds for low treedepth covers we can effectively compute the bounds for the SC-depth and the size of low shrubdepth covers of $\mathscr{D}$. In particular, the sparsification of each graph $G \in \mathscr{D}$ can be computed effectively (in at most exponential time). It follows from Lemmas 5.11 and 5.12 that there exists a bounded expansion class $\mathscr{B}$ (the sparsification of $\mathscr{D}$), a computable monadic lift Λ, an almost quantifier-free transduction S and an almost quantifier-free interpretation I, such that the following diagram commutes. Note that S and I can also be effectively computed (they are explicitly defined and are linear-time computable as they are almost quantifier free).

$$\begin{tikzcd}
\mathscr{C}^+ \arrow[dr, "\mathsf{T}_0"] & \mathscr{D}^+ \arrow[dr, "\mathsf{S}"] & \\
\mathscr{C} \arrow[u, "\Lambda_0"] \arrow[r, "f"'] & \mathscr{D} \arrow[u, "\Lambda"] \arrow[loop below, "\mathsf{Id}_{\mathscr{D}}"] & \mathscr{B} \arrow[l, "\mathsf{I}"'] \arrow[loop right, "\mathsf{Id}_{\mathscr{B}}"]
\end{tikzcd}$$

8 CONCLUSION AND OPEN PROBLEMS

In this article, we have provided a natural combinatorial characterization of graph classes that are first-order transductions with bounded expansion classes of graphs. Our characterization parallels the known characterization with bounded expansion classes by the existence of low treedepth covers, by replacing the notion of treedepth by shrubdepth. We believe that we have thereby taken a big step towards solving the model-checking problem for first-order logic on classes of structurally bounded expansion.

On the structural side, we remark that transductions with bounded expansion graph classes are just the same as transductions of classes of structures with bounded expansion (i.e., classes whose Gaifman graphs or whose incidence encodings have bounded expansion). For general structures, we are lacking the analogue of Lemma 5.8; the problem is that, within the proof, we crucially use the characterization of shrubdepth via SC-depth, which works well for graphs but is unclear for structures of higher arity. Hence, the following question:

Open Problem 1. *Characterize classes of relational structures (rather than just graphs) that are transductions with bounded expansion classes.*

Theorem 7.2 witnesses the importance of the notion of efficiently decomposable structurally bounded expansion class. Hence, the following question:

Open Problem 2. *Is every structurally bounded expansion class efficiently decomposable?*

The sparsification step in Theorem 7.2 also needed to compute SC-decomposition for classes with bounded shrubdepth. We solved this by using the general machinery of Courcelle. It is likely that the time complexity can be reduced.

Open Problem 3. *Does there exist, for each integer d, a linear-time algorithm that, computes an SC-decomposition of height at most $f(d)$ of input graphs of SC-depth at most d, for some computable function f?*

Finally, observe that classes with bounded expansion can be characterized among classes with structurally bounded expansion as those which are bi-clique free. It follows, that every monotone (i.e., subgraph closed) class of structurally bounded expansion has bounded expansion. Exactly the same statement holds characterizing bounded treedepth among bounded shrubdepth, and the second item holds for treewidth vs cliquewidth. In particular, for monotone-graph classes, all pairs of notions collapse.

We do not know how to extend our results to nowhere dense classes of graphs, mainly due to the following problem:

Open Problem 4. *Do nowhere dense classes of graphs admit some quantifier-elimination procedure?*

A APPENDIX

In this appendix, we give alternative proofs of Lemma 5.4 and Lemma 6.2, which might be of interest for some readers.

A.1 Greedy Proof of Lemma 5.4

We now present the second proof of Lemma 5.4. As asserted by Lemma 2.8, graphs from a fixed class with bounded shrubdepth do not admit arbitrarily long induced paths. We need a strengthening of this statement: classes with bounded shrubdepth also exclude induced structures that roughly resemble paths, as explained in this section.

Definition A.1. Let G be a graph. A *quasi-path* of length ℓ in G is a sequence of vertices $(u_1, u_2, \ldots, u_\ell)$ satisfying the following conditions:

- $u_i u_{i+1} \in E(G)$ for all $i \in [\ell - 1]$; and
- for every odd $i \in [\ell]$ and even $j \in [\ell]$ with $j > i + 1$, we have $u_i u_j \notin E(G)$.

Note that in a quasi-path we do not restrict in any way the adjacencies between u_i and u_j when i, j have the same parity, or even when i is odd and j is even but $j < i - 1$. We now prove that classes with bounded shrubdepth do not admit long quasi-paths; note that since an induced path is also a quasi-path, the following lemma actually implies Lemma 2.8.

Lemma A.2. *For every class $\mathscr{C}$ of graphs with bounded shrubdepth, there exists a constant $q \in \mathbb{N}$ such that no graph from $\mathscr{C}$ contains a quasi-path of length q.*

Proof. It suffices to prove the following claim.

Claim 3. *There exists a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that no graph admitting a connection model of height h and using m labels contains a quasi-path of length larger than $f(h, m)$.*

The proof is by induction on h. Observe first that graphs admitting a connection model of height 0 are exactly graphs with one vertex, hence we may set $g = f(0, m) = 1$ for all $m \in \mathbb{N}$.

We now move to the induction step. Assume G admits a connection model T of height $h \geqslant 1$ where $\lambda : V(G) \to \Lambda$ is the corresponding labeling of $V(G)$ with a set Λ consisting of m labels. Call two vertices $u, v \in V(G)$ *related* if in T they are contained in the same subtree rooted at a child of the root of T; obviously this is an equivalence relation. The least common ancestor of two unrelated vertices is always the root of T, hence for any two unrelated vertices u, v, whether u and v are adjacent depends only on the label of u and the label of v.

Now suppose G admits a quasi-path $Q = (u_1, \ldots, u_\ell)$. A *block* in Q is a maximal contiguous subsequence of Q consisting of pairwise related vertices. Thus Q is partitioned into blocks, say $B_1, \ldots, B_p$ appearing in this order on Q. Observe that every block B_i either is a quasi-path itself or becomes a quasi-path after removing its first vertex. Since vertices of B_i are pairwise related, they are contained in an induced subgraph of G that admits a tree model of height $h - 1$ and using m labels, implying by the induction hypothesis that

$$\text{every block has length at most } f(h-1, m) + 1. \tag{4}$$

Next, for every non-last block B_i (i.e. $i \neq p$), let the *signature* of B_i be the following triple:

- the parity of the index of the last vertex of B_i,
- the label of the last vertex of B_i, and
- the label of its successor on Q, that is, the first vertex of B_{i+1}.

The next claim is the key step in the proof.

Claim 4. *There are no seven non-last blocks with the same signature.*

Proof. Supposing for the sake of contradiction that such seven non-last blocks exist, by taking the first, the fourth, and the seventh of them we find three non-last blocks B_i, B_j, B_k with sames signature such that $1 \leqslant i < j < k < p$ and $j - i > 2$ and $k - j > 2$. Let $1 \leqslant a < b < c < \ell$ be the indices on Q of the last vertices of B_i, B_j, B_k, respectively. By the assumption, $\lambda(u_a) = \lambda(u_b) = \lambda(u_c)$, $\lambda(u_{a+1}) = \lambda(u_{b+1}) = \lambda(u_{c+1})$, and a, b, c have the same parity. Suppose for now that a, b, c are all even; the second case will be analogous. Further, the assumptions $j - i > 2$ and $k - j > 2$ entail $b > a + 2$ and $c > b + 2$.

Observe that u_{a+1} and u_b have to be related. Indeed, u_a has the same label as u_b, while it is unrelated and adjacent to u_{a+1}. So if u_{a+1} and u_b were unrelated, then they would be adjacent as well, but this is a contradiction because $a+1$ is odd, b is even, and $a+2<b$. Similarly, u_a and u_{c+1} are related and u_b and u_{c+1} are related. By transitivity, we find that u_b and u_{b+1} are related, a contradiction.

The case when a,b,c are all odd is analogous: we similarly find that u_a is related to u_{b+1}, u_a is related to u_{c+1}, and u_b is related to u_{c+1}, implying that u_b is related to u_{b+1}, a contradiction. This concludes the proof. ⌟

Since there are $2m^2$ different signatures, Claim 4 implies that

$$\text{the number of blocks is at most } 12m^2+1. \tag{5}$$

Assertions Equation (4) and Equation (5) together imply that $\ell \leqslant (f(h-1,m)+1)(12m^2+1)$. As Q was chosen arbitrarily, we may set

$$f(h,m) := (f(h-1,m)+1)\cdot(12m^2+1).$$

This concludes the proof of Claim 3 and of Lemma A.2. □

Now Lemma 5.4 immediately follows from the following (essentially reformulated) statement.

Lemma A.3. *For every class $\mathscr{C}$ of graphs with bounded shrubdepth, there exists a constant $p \in \mathbb{N}$ such that the following holds. Suppose $G \in \mathscr{C}$ and A and B are two disjoint subsets of vertices of G such that every vertex of A has a neighbor in B. Then, there exist subsets $B_1,\ldots,B_p \subseteq B$ with the following property: for every vertex $v \in A$, there exists $i \in [p]$ such that v has exactly one neighbor in B_i.*

Proof. Call a vertex $u \in B$ a *private neighbor* of a vertex $v \in A$ is u is the only neighbor of v in B. Consider the following procedure which iteratively removes vertices from A and B until A becomes empty.

The procedure proceeds in rounds, where each round consists of two reduction steps, performed in order:

(1) *B-reduction:* As long as there exists a vertex $u \in B$ that is not a private neighbor of any $v \in A$, remove u from B.
(2) *A-reduction:* Remove all vertices from A that have exactly one neighbor in B.

Observe that, in the B-reduction step, we never remove any vertex that is a private neighbor of some vertex in A, so, during the procedure, we maintain the invariant that every vertex of A has at least one neighbor in B. Note also that, in any round, after the B-reduction step, the set B remains nonempty, due to the invariant, and then every vertex of B is a private neighbor of some vertex of A. Thus, the A-reduction step will remove at least one vertex from A per each vertex of B, so the size of A decreases in each round. Consequently, the procedure stops after a finite number of rounds, say ℓ, when A becomes empty.

Let $B_1,\ldots,B_\ell$ be subsets of the original set B such that B_i denotes B after the ith round of the procedure. Further, let $A_1,\ldots,A_\ell$ be the subsets of the original set A such that A_i comprises vertices removed from A in the ith round. Note that $A_1,\ldots,A_\ell$ form a partition of A. The following properties follow directly from the construction:

(1) Every vertex of A_i has exactly one neighbor in B_i, for each $1 \leqslant i \leqslant \ell$.
(2) Every vertex of A_i has at least two neighbors in B_{i-1}, for each $2 \leqslant i \leqslant \ell$.
(3) Every vertex of B_i has at least one neighbor in A_i, for all $1 \leqslant i \leqslant \ell$.

For Property (2), observe that otherwise such a vertex would be removed in the previous round.

Property (1) implies that subsets $B_1, \dots, B_\ell$ satisfy the property requested in the lemma statement. Hence, it suffices to show that ℓ, the number of rounds performed by the procedure, is universally bounded by some constant p depending on the class $\mathscr{C}$ only.

Take any vertex $v_\ell \in A_\ell$. By Property (1) and Property (2), it has at least two neighbors in $B_{\ell-1}$, out of which one, say u_ℓ, belongs to B_ℓ, and another, say $u_{\ell-1}$, belongs to $B_{\ell-1} - B_\ell$. Next, by Property (3), we have that $u_{\ell-1}$ has a neighbor $v_{\ell-1} \in A_{\ell-1}$. Observe that $v_{\ell-1}$ cannot be adjacent to u_ℓ, because $v_{\ell-1}$ has exactly one neighbor in $B_{\ell-1}$ by Property (1) and it is already adjacent to $u_{\ell-1} \neq u_\ell$. Again, by Property (1) and Property (2), we infer that $v_{\ell-1}$ has another neighbor $u_{\ell-2} \in B_{\ell-2} - B_{\ell-1}$. In turn, by Property (3) again, $u_{\ell-2}$ has a neighbor $v_{\ell-2} \in A_{\ell-2}$, which is non-adjacent to both $u_{\ell-1}$ and u_ℓ, because $u_{\ell-2}$ is its sole neighbor in $B_{\ell-2}$. Continuing in this manner, we find a sequence of vertices

$$(v_1, u_1, v_2, u_2, \dots, v_\ell, u_\ell)$$

with the following properties: each two consecutive vertices in the sequence are adjacent and for each $i < j$, v_i is non-adjacent to u_j. This is a quasi-path of length 2ℓ. By Lemma A.2, there is a universal bound q depending only on $\mathscr{C}$ on the length of quasi-paths in G, implying that we may take $p = \lfloor q/2 \rfloor$. □

A.2 Proof of Lemma 6.2: Quantifier Elimination on Trees with Bounded Depth

We first give a quantifier elimination procedure for colored trees of bounded depth. In the following, we consider Σ-*labeled trees*, that is, unordered rooted trees t where each node is labeled with exactly one element of Σ. We write $t(v)$ for the label of a node v in the tree t. In this section, we model trees by their parent functions, that is, we consider them as structures where the universe of the structure is the node set, there is a unary relation for each *label* from Σ, and there is one partial function that maps each node to its parent (the roots are not in the domain). A Γ-relabeling of a Σ-labeled tree t is any Γ-labeled tree whose underlying unlabeled tree is the same as that of t. As usual, a class of trees $\mathscr{T}$ has *bounded height* if there exists $h \in \mathbb{N}$ such that each tree in $\mathscr{T}$ has height at most h.

For convenience, we now regard sets of free variables of formulas, instead of traditional tuples. That is, if φ is a formula with free variables X and $\nu : X \to V(t)$ is a valuation of variables from X in a tree t, then we write $t, \nu \models \varphi$ if the formula φ is satisfied in t when its free variables are evaluated as prescribed by ν.

Our quantifier elimination procedure is provided by the following lemma, which implies Lemma 6.2.

Lemma A.4. *Let $\mathscr{T}$ be a class of Σ-labeled trees with bounded height and let φ be a first-order formula over the signature of Σ-labeled trees with free variables X. Then, there exists a finite set of labels Γ, a Γ-relabeling $\widehat{t}$ of t, and a quantifier-free formula $\widehat{\varphi}$ over the signature of Γ-labeled trees with free variables X, such that for each valuation ν of X in t we have*

$$t, \nu \models \varphi \qquad \textit{if and only if} \qquad \widehat{t}, \nu \models \widehat{\varphi}.$$

The result immediately lifts to classes of forests with bounded depth, which are modeled the same way as trees, i.e., using a unary parent function.

Corollary A.5. *The same statement as above holds for a class $\mathscr{F}$ of Σ-labeled forests with bounded height and a first-order formula ψ over the signature Σ-labeled forests.*

Proof. Let $\mathscr{F}$ be a class of Σ-labeled forests with bounded height and let ψ be a first-order formula with free variables X. Construct a class of Σ-labeled trees $\mathscr{T}$, by prepending an unlabeled root r_f to each forest f in $\mathscr{F}$, yielding a tree t_f. We may rewrite the formula ψ to a first-order formula φ such that $f, \nu \models \psi$ if and only if $t_f, \nu \models \varphi$, for every $f \in \mathscr{F}$, and every valuation ν of X in f.

Apply Lemma A.4 to $\mathscr{T}$, yielding a relabeling $\widehat{t}$ of each tree t in $\mathscr{T}$, using some finite set of labels Γ. This relabeling yields a relabeling $\widehat{f}$ of each forest $f \in \mathscr{F}$, where each non-root node v is labeled by a pair of labels: the label of v in the tree $\widehat{t_f}$, and the label of the root of $\widehat{t_f}$. Furthermore, we have $t_f, \nu \models \varphi$ if and only if $\widehat{t_f}, \nu \models \widehat{\varphi}$, for every valuation ν. Note that all quantifier-free properties involving the prepended root r_f in the Γ-labeled tree $\widehat{t_f}$ can be decoded from the labeled forest $\widehat{f}$: the unary predicates that hold in r_f are encoded in all the vertices of $\widehat{f}$, and r_f is the parent of the roots of $\widehat{f}$ (the elements for which the parent function is undefined). It follows that we may rewrite the formula $\widehat{\varphi}$ to a formula $\widehat{\psi}$ such that $\widehat{t_f}, \nu \models \widehat{\varphi}$ if and only if $\widehat{f}, \nu \models \widehat{\psi}$, for every valuation ν of X in f. Reassuming, $f, \nu \models \psi$ if and only if $\widehat{f}, \nu \models \widehat{\psi}$, for every $f \in \mathscr{F}$ and every valuation ν of X in f. $\square$

Corollary A.5 immediately implies Lemma 6.2. It remains to prove Lemma A.4. Before proving Lemma A.4, we recall some standard automata-theoretic techniques.

We define tree automata that process unordered labeled trees. Such automata process an input tree t from the leaves to the root assigning states to each node in the tree. The state assigned to the current node v depends only on the label $t(v)$ and the multiset of states labeling the children of v, where the multiplicities are counted only up to a certain fixed threshold. Because of that, we call these automata *threshold tree automata*.

We develop all the simple facts about tree automata needed for our purposes below. We refer to [32] for a general introduction. Note that what is usually considered under the notion of *tree automata* are automata which process *ordered* trees, i.e., trees where the children of each node are ordered. Tree automata collapse in expressive power to threshold tree automata in the case when they are required to be independent of the order, i.e., if $\mathcal{A}$ is a tree automaton with the property that for any two ordered trees t, t' which are isomorphic as unordered trees, either both t and t' are accepted by $\mathcal{A}$ or both t and t' are rejected by $\mathcal{A}$, then the language (i.e., set) of trees accepted by $\mathcal{A}$ is equal to the language of trees accepted by some threshold automaton. Therefore, the theory of threshold tree automata is a very simple and special case of that of tree automata. We now recall some simple facts about such automata.

Fix a set of labels Q. A Q-multiset is a multiset of elements of Q. If τ is a number and X is a Q-multiset, then by $X\downharpoonright_\tau$ we denote the maximal multiset $X' \subseteq X$ where the multiplicity of each element is at most τ. In other words, for every element whose multiplicity in X is more than τ, we put it exactly τ times to X'; all the other elements retain their multiplicities.

We define *threshold tree automata* as follows. A threshold tree automaton is a tuple $(\Sigma, Q, \tau, \delta, F)$, consisting of

- a finite input alphabet Σ;
- a finite state space Q;
- a *threshold* $\tau \in \mathbb{N}$;
- a transition relation δ, which is a finite set of rules of the form (a, X, q), where $a \in \Sigma$, $q \in Q$, and X is a Q-multiset in which each element occurs at most τ times; and
- an *accepting condition* F, which is a subset of Q.

A *run* of such an automaton over a Σ-labeled tree t is a Q-labeling $\rho : V(t) \to Q$ of t satisfying the following condition for every node x of t:

> If $t(x) = a, \rho(x) = q$ and X is the multiset of the Q-labels of the children of x in t, then $(a, X\downharpoonright_\tau, q) \in \delta$.

The automaton *accepts* a Σ-labeled tree t if it has a run ρ on t such that $\rho(r) \in F$, where r is the root of t. The *language* of a threshold tree automaton is the set of Σ-labeled trees it accepts. A language L of Σ-labeled trees is *threshold-regular* if there is a threshold tree automaton whose language is L; we also say that this automaton *recognizes* L.

An automaton is *deterministic* if for all $a \in \Sigma$ and all Q-multisets X in which each element occurs at most τ times there exists q such that $(a, X, q) \in \delta$ and whenever $(a, X, q), (a, X, q') \in \delta$, then $q = q'$. Note that a deterministic automaton has a unique run on every input tree.

The next lemma explains basic properties of threshold tree automata and follows from standard automata constructions. In the lemma, we speak about *monadic second-order logic* (MSO), which is the extension of first-order logic by quantification over unary predicates.

Lemma A.6. *The following assertions hold:*

(1) For every threshold automaton there is a deterministic threshold automaton with the same language.
(2) Threshold-regular languages are closed under boolean operations.
(3) If $f : \Sigma \to \Gamma$ is any function and L is a threshold-regular language of Σ-labeled trees, then the language $f(L)$ comprising trees obtained from trees of L by replacing each label by its image under f is also threshold-regular.
(4) For every MSO sentence φ in the language of Σ-labeled trees there is a deterministic threshold automaton $\mathcal{A}_\varphi$ whose language is the set of trees satisfying φ.

Proof. Assertion (1) follows by applying the standard powerset determinization construction. For assertion (2), it follows from assertion (1) that every threshold-regular language is recognized by a deterministic threshold tree automaton. Then, for conjunctions we may use the standard product construction and for negation we may negate the accepting condition. For assertion 3, an automaton recognizing $f(L)$ can be constructed from an automaton recognizing L by nondeterministically guessing labels from Σ consistently with the given labels from Γ, so that the guessed Σ-labeling is accepted by the automaton recognizing L. Now assertion 4 follows from 1, 2, and 3 in a standard way, because every MSO formula can be constructed from atomic formulas using boolean combinations and existential quantification (which can be regarded as a relabeling f that forgets the information about the quantified set). □

Let X be a finite set of (first-order) variables and let $\Sigma_X = \Sigma \times \mathscr{P}(X)$. Given a tree t and a partial valuation $\nu : X \rightharpoonup V(t)$, let $t \otimes \nu$ be the Σ_X-tree obtained from t, by replacing, for each node u of t, the label a of u by the pair (a, Y) where $Y = \nu^{-1}(u) \subseteq X$.

Toward the proof of Lemma A.4, consider a first-order formula φ over Σ-labeled trees with free variables X. We can easily rewrite φ to a first-order sentence ψ over Σ_X-labeled trees such that $t, \nu \models \varphi$ if and only if $t \otimes \nu \models \psi$ for every Σ-labeled tree t and valuation $\nu : X \to V(t)$. By Lemma A.6 there is a deterministic threshold automaton $\mathcal{A}_\psi$ whose language is exactly the set of Σ_X-labeled trees satisfying ψ.

Denote by Q the set of states and by K the threshold of $\mathcal{A}_\psi$, and let $M = K + |X|$. Denote by Δ the set of Q-multisets in which every element occurs at most M times.

Given a Σ-labeled tree t and a partial valuation $\nu : X \rightharpoonup V(t)$, define ρ_ν as the Q-labeling of t which is the unique run of $\mathcal{A}_\psi$ over $t \otimes \nu$. For a node u of t, let $C_\nu(u)$ be the Q-multiset defined as follows:

$$C_\nu(u) = \{\rho_\nu(w) : w \text{ is a child of } u \text{ in } t\}.$$

Define a new set of labels $\Gamma = \Sigma \times \Delta$, and a Γ-relabeling $\widehat{t}$ of t as follows: for each $u \in V(t)$, say with label $a \in \Sigma$ in t, the label of u in $\widehat{t}$ is the pair $(a, C_\emptyset(u) \downharpoonright_M)$, where $\emptyset$ is the partial valuation that leaves all variables of X unassigned. Our goal now is to prove that this relabeling $\widehat{t}$ of t satisfies the conditions expressed in Lemma A.4. To this end, given a valuation v of X in $\widehat{t}$, let $\widehat{t}|_v$ denote the Γ_X-labeled tree obtained from $\widehat{t} \otimes v$ by restricting the node set to the set of ancestors of nodes in the image $v(X)$ of v.

Lemma A.7. *There is a set of Γ_X-labeled trees $\mathcal{R}$ such that for every Σ-labeled tree t and valuation v of X in t,*

$$t, v \models \varphi \quad \text{if and only if} \quad \widehat{t}|_v \in \mathcal{R}.$$

Proof. Fix a tree t and a valuation v of X in t. We say that a node u of t is *nonempty* if it has a descendant which is in the image of v. For node u of t, define the following Q-multisets:

$$N_\emptyset(u) = \{\rho_\emptyset(w) : w \text{ is a nonempty child of } u\},$$
$$N_v(u) = \{\rho_v(w) : w \text{ is a nonempty child of } u\}.$$

Note that, since there are at most $|X|$ nonempty children of a given node u, there is a finite set Z independent of t and v such that the functions N_v and $N_\emptyset$ take values in Z. Fix a node u of t.

Claim 5. *The state $\rho_v(u)$ is uniquely determined by the label of u in $t \otimes v$, and the Q-multisets $C_\emptyset(u) \downharpoonright_M, N_\emptyset(u)$ and $N_v(u)$, i.e., there is a function $f : \Sigma_X \times \Delta \times Z \times Z \to Q$ such that for every tree t, valuation v, and node u,*

$$\rho_v(u) = f(\,\text{label of } u \text{ in } t \otimes v,\ C_\emptyset(u) \downharpoonright_M,\ N_\emptyset(u),\ N_v(u)\,). \tag{6}$$

Proof. Clearly, $N_\emptyset(u) \subseteq C_\emptyset(u)$, as multisets. Moreover, the following equality among multisets holds:

$$C_v(u) = (C_\emptyset(u) - N_\emptyset(u)) + N_v(u). \tag{7}$$

This is because the automaton $\mathcal{A}_\psi$ is deterministic and therefore $\rho_v(w) = \rho_\emptyset(w)$ for all nodes w which are not nonempty. From Equation (7), the fact that $N_\emptyset(u)$ has at most $|X|$ elements and $M = K + |X|$, it follows that

$$((C_\emptyset(u) \downharpoonright_M - N_\emptyset(u)) + N_v(u)) \downharpoonright_K = (C_v(u)) \downharpoonright_K. \tag{8}$$

By definition of the run of $\mathcal{A}_\psi$ on $t \otimes v$, the state $\rho_v(u)$ is determined by the label of u in $t \otimes v$ and by $(C_v(u)) \downharpoonright_K$. It follows from Equation (8) that $\rho_v(u)$ is uniquely determined by the label of u in $t \otimes v$, $(C_\emptyset(u)) \downharpoonright_M$, and the Q-multisets $N_\emptyset(u)$ and $N_v(u)$, proving the claim. □

From Claim 5, it follows that the state $\rho_v(r)$, where r is the root of t, depends only on the tree $\widehat{t}|_v$. Indeed, we can inductively compute the states $\rho_v(u)$ and $\rho_\emptyset(u)$, moving from the leaves of $\widehat{t}|_v$ towards the root, as follows. Suppose u is a node of $\widehat{t}|_v$ such that $\rho_v(v)$ and $\rho_\emptyset(v)$ have been computed for all the nonempty children v of u (in particular, this holds if u is a leaf of $\widehat{t}|_v$). Then, we can determine the multisets $N_v(u)$ and $N_\emptyset(u)$ using their definitions, and consequently, we can determine $\rho_v(u)$ by Equation (6), whereas $\rho_\emptyset(u)$ only depends on $C_\emptyset(u) \downharpoonright_K$ and on the label of u in t. Note that both the label of u in t and the multiset $C_\emptyset(u) \downharpoonright_K$ are encoded in the label of u in $\widehat{t}$.

As shown above, for any tree t and valuation v, the state of ρ_v at the root depends only on $\widehat{t}|_v$. On the other hand, $t, v \models \varphi$ if and only if the state of $\rho_v(r)$ at the root is an accepting state. Hence, whether or not $t, v \models \varphi$, depends only on the tree $\widehat{t}|_v$. This proves the lemma. □

Finally, we observe the following:

Lemma A.8. *For each Γ_X-labeled tree s there exists a quantifier-free formula ψ_s over the signature of Γ-labeled trees with free variables X such that the following holds: for every Γ-labeled tree t and valuation v of X in t, we have*

$$t, v \models \psi_s \qquad \text{if and only if} \qquad t|_v \text{ is isomorphic to } s.$$

Proof. Observe that the ancestors of nodes in $\nu(X)$ may be obtained by applying the parent function to them. Thus, using a quantifier-free formula, we may check whether each node of $\nu(X)$ lies at depth as prescribed by s, whether its ancestors have labels as prescribed by s, and whether the depth of the least common ancestor of every pair of nodes of $\nu(X)$ is as prescribed by s. Then, $t|_\nu$ is isomorphic to s if and only if all these conditions hold. □

With all the tools prepared, we may prove Lemma A.4.

Proof of Lemma A.4. Let $\mathcal{R}_h$ be the intersection of $\mathcal{R}$ with the class of trees of height at most h. Since each tree from $\mathcal{R}$ has at most $|X|$ leaves by definition, $\mathcal{R}_h$ is finite and its size depends only on $|X|$ and h. By Corollary A.7, it now suffices to define $\widehat{\varphi}$ as the disjunction of formulas ψ_s provided by Lemma A.8 over $s \in \mathcal{R}_h$. □

REFERENCES

[1] A. Atminas, V. V. Lozin, and I. Razgon. 2012. Linear time algorithm for computing a small biclique in graphs without long induced paths. In *Scandinavian Workshop on Algorithm Theory*. Springer, 142–152.

[2] A. Blumensath and B. Courcelle. 2010. On the Monadic second-order transduction hierarchy. *Logical Methods in Computer Science* 6, 2 (2010). DOI: https://doi.org/10.2168/LMCS-6(2:2)2010

[3] B. Courcelle. 1990. Graph rewriting: An algebraic and logic approach. In *Handbook of Theoretical Computer Science*. Vol. 2. Elsevier, Amsterdam, Chapter 5, 142–193.

[4] B. Courcelle. 1990. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Inform. and Comput.* 85 (1990), 12–75.

[5] B. Courcelle. 1992. The monadic second-order logic of graphs VII: Graphs as relational structures. *Theoretical Computer Science* 101, 1 (1992), 3–33.

[6] B. Courcelle. 2009. Linear delay enumeration and monadic second-order logic. *Discrete Applied Mathematics* 157, 12 (2009), 2675–2700. DOI: https://doi.org/10.1016/j.dam.2008.08.021 (Second Workshop on Graph Classes, Optimization, and Width Parameters).

[7] B. Courcelle and J. Engelfriet. 2012. *Graph Structure and Monadic Second-order Logic: A Language-theoretic Approach*. Vol. 138. Cambridge University Press.

[8] B. Courcelle, J. A. Makowsky, and U. Rotics. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems* 33 (2000), 125–150. Issue 2.

[9] A. Dawar, M. Grohe, and S. Kreutzer. 2007. Locally excluding a minor. In *22nd Annual IEEE Symposium on Logic in Computer Science*. 270–279.

[10] A. Durand and E. Grandjean. 2007. First-order queries on structures of bounded degree are computable with constant delay. *ACM Transactions on Computational Logic (TOCL)* 8, 4 (2007), 21.

[11] A. Durand, N. Schweikardt, and L. Segoufin. 2014. Enumerating answers to first-order queries over databases of low degree. In *33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, 121–131.

[12] Z. Dvořák, D. Kráľ, and R. Thomas. 2010. Deciding first-order properties for sparse graphs. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS'10)*. 133–142. DOI: https://doi.org/10.1109/FOCS.2010.20

[13] Z. Dvořák, D. Kráľ, and R. Thomas. 2013. Testing first-order properties for subclasses of sparse graphs. *J. ACM* 60:5 Article 36 (2013).

[14] K. Eickmeyer and K. Kawarabayashi. 2017. FO model checking on map graphs. In *21st International Symposium on Fundamentals of Computation Theory, (FCT'17)*. 204–216.

[15] J. Flum and M. Grohe. 2001. Fixed-parameter tractability, definability, and model checking. *SIAM Journal of Computing* 31 (2001), 113–145.

[16] M. Frick and M. Grohe. 2001. Deciding first-order properties of locally tree-decomposable structures. *J. ACM* 48 (2001), 1148–1206.

[17] J. Gajarský, P. Hliněný, D. Lokshtanov, J. Obdržálek, S. Ordyniak, M. S. Ramanujan, and S. Saurabh. 2015. FO model checking on posets of bounded width. In *56th Annual Symposium on Foundations of Computer Science (FOCS), 2015*. IEEE, 963–974.

[18] J. Gajarský, P. Hliněný, J. Obdržálek, D. Lokshtanov, and M. S. Ramanujan. 2016. A new perspective on FO model checking of dense graph classes. In *31st Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM, 176–184.

[19] J. Gajarský, S. Kreutzer, J. Nešetřil, P. Ossona de Mendez, M. Pilipczuk, S. Siebertz, and S. Toruńczyk. 2018. First-order interpretations of bounded expansion classes. In *45th International Colloquium on Automata, Languages, and Programming, ICALP*. 126:1–126:14.

[20] R. Ganian, P. Hliněný, J. Nešetřil, J. Obdržálek, and P. Ossona de Mendez. 2019. Shrub-depth: Capturing height of dense graphs. *Logical Methods in Computer Science* 15, 1 (2019). https://lmcs.episciences.org/5149.
[21] R. Ganian, P. Hliněný, J. Obdržálek, J. Schwartz, J. Teska, and D. Kráľ. 2013. FO model checking of interval graphs. In *International Colloquium on Automata, Languages, and Programming*. Springer, 250–262.
[22] R. Ganian, P. Hliněný, J. Nešetřil, J. Obdržálek, P. Ossona de Mendez, and R. Ramadurai. 2012. When trees grow low: Shrubs and fast MSO_1. In *MFCS 2012 (Lecture Notes in Computer Science)*, Vol. 7464. Springer-Verlag, 419–430.
[23] V. Giakoumakis and J.-M. Vanherpe. 1997. Bi-complement reducible graphs. *Adv. Appl. Math.* 18 (1997), 389–402.
[24] M. Grohe. 2001. Generalized model-checking problems for first-order logic. In *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 12–26.
[25] M. Grohe and S. Kreutzer. 2011. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics (Contemporary mathematics)*. 181–206.
[26] M. Grohe, S. Kreutzer, and S. Siebertz. 2017. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM* 64, 3 (2017), 17:1–17:32.
[27] F. Gurski and E. Wanke. 2000. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *WG 2000 (Lecture Notes in Computer Science)*, Vol. 1928. Springer, 196–205.
[28] P. Hlinený, F. Pokrývka, and B. Roy. 2017. FO model checking of geometric graphs. In *12th International Symposium on Parameterized and Exact Computation (IPEC'17)*. 19:1–19:12.
[29] W. Kazana and L. Segoufin. 2013. Enumeration of first-order queries on classes of structures with bounded expansion. In *16th International Conference on Database Theory*. 10–20.
[30] W. Kazana and L. Segoufin. 2013. Enumeration of first-order queries on classes of structures with bounded expansion. In *32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. ACM, 297–308.
[31] O. Kwon, Mi. Pilipczuk, and S. Siebertz. 2017. On low rank-width colorings. In *43rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG'17)*. 372–385.
[32] C. Löding. 2012. Basics on tree automata. In *Modern Applications of Automata Theory*. 79–110. DOI: https://doi.org/10.1142/9789814271059_0003
[33] J. Nešetřil and P. Ossona de Mendez. 2006. Tree depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics* 27, 6 (2006), 1022–1041. DOI: https://doi.org/10.1016/j.ejc.2005.01.010
[34] J. Nešetřil and P. Ossona de Mendez. 2008. Grad and classes with bounded expansion. I. Decompositions. *European Journal of Combinatorics* 29, 3 (2008), 760–776.
[35] J. Nešetřil and P. Ossona de Mendez. 2008. Grad and classes with bounded expansion. II. Algorithmic aspects. *European Journal of Combinatorics* 29, 3 (2008), 777–791. DOI: https://doi.org/10.1016/j.ejc.2006.07.014
[36] J. Nešetřil and P. Ossona de Mendez. 2012. *Sparsity (Graphs, Structures, and Algorithms)*. Algorithms and Combinatorics, Vol. 28. Springer. 465 pages.
[37] S. Oum. 2008. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms (TALG)* 5, 1 (2008), 10.
[38] D. Seese. 1996. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science* 5 (1996), 505–526.
[39] L. Segoufin and W. Kazana. 2011. First-order query evaluation on structures of bounded degree. *Logical Methods in Computer Science* 7 (2011).

Received October 2018; revised August 2019; accepted January 2020