First-order interpretations of bounded expansion classes *

Jakub Gajarský[1], Stephan Kreutzer[1], Jaroslav Nešetřil[2],
Patrice Ossona de Mendez[2,3], Michał Pilipczuk[4], Sebastian Siebertz[4], and
Szymon Toruńczyk[4]

[1]Technical University Berlin, Germany,
{jakub.gajarsky,stephan.kreutzer}@tu-berlin.de
[2]Charles University, Prague, Czech Republic,
{nesetril,patrice}@kam.mff.cuni.cz
[3]CAMS (CNRS, UMR 8557), Paris, France, pom@ehess.fr
[4]University of Warsaw, Poland,
{michal.pilipczuk,siebertz,szymtor}@mimuw.edu.pl

Abstract

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 interpretations 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 decompositions, replacing treedepth by its dense analogue called shrubdepth.

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*

*J. Nešetřil and P. Ossona de Mendez are supported by CE-ITI P202/12/G061 of GACR and European Associated Laboratory (LEA STRUCO). J. Gajarský and S. Kreutzer are 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 are 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 is supported by the NCN grant 2016/21/D/ST6/01485.

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 [1,2], which states that model-checking a formula φ of *monadic second-order logic* can be solved in time $f(\varphi) \cdot n$ on any graph with n vertices which comes from a fixed class of graphs of *bounded treewidth*, for some computable function f. Seese [33] proved an analogue of Courcelle's result for the model-checking problem of first-order logic on any class of graphs of 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. [5–7, 9–12, 15, 19, 20, 22, 24–26, 33, 34]. 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 of bounded *treewidth*, *excluding a minor* or which have *bounded expansion* or are *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 *sparseness*. See [31] 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 or logic, indicate that these two concepts capture some very natural limits of "well-behavedness" and algorithmic tractability. For instance, Grohe et al. [22] proved that if $\mathscr{C}$ is a class of graphs closed under taking subgraphs then model checking first-order logic on $\mathscr{C}$ is tractable if, and only if, $\mathscr{C}$ is nowhere dense (the lower bound was proved in [9]). As far as algorithmic meta-theorems for fixed-parameter tractability of first-order model-checking are concerned, this result completely solves the case for graph classes which 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 of bounded *cliquewidth* [3] or for first-order logic on *interval graphs*, *partial orders*, classes of bounded *shrubdepth* and other classes, see e.g. [13–15, 17], a general theory of meta-theorems for dense classes is still missing. Moreover, unlike 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 and to 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 which are not necessarily sparse. This is the main motivation for the research in this paper. Since bounded expansion and nowhere denseness form the limits for tractability of certain problems 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 of bounded expansion or nowhere denseness.

In this paper, 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 interpretations 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 and transductions. One possible way of constructing "well-behaved" and "structurally simple" classes of graphs is to use logical *interpretations*, or the related concept of *transductions* studied in formal language and automata theory. For our purpose, transductions are more convenient and we will use them in this paper. Intuitively, a *transduction* is a logically defined operation which takes a structure as input and nondeterministically produces as output a target structure. In this paper we use *first-order* transductions, which involve first-order formulas (see Section 2 for details). Two examples of such transductions are graph complementation, and the squaring operation which, given a graph G, adds an edge between every pair of vertices at distance 2 from each other.

We postulate that if we start with a "structurally simple" class $\mathscr{C}$ of graphs, e.g. a class of bounded expansion or a nowhere dense class, and then study the graph classes $\mathscr{D}$ which can be obtained from $\mathscr{C}$ by first-order transductions, then the resulting classes should still have a simple structure and thus be well-behaved algorithmically as well as in terms of logic. In other words, the resulting classes are interesting graph classes with good algorithmic and logical properties, and which are certainly not sparse in general. For instance, a useful feature of transductions is that they provide a canonical way of reducing model-checking problems from the generated classes $\mathscr{D}$ to the original class $\mathscr{C}$, provided that given a graph $H \in \mathscr{D}$, we can effectively compute some graph $G \in \mathscr{C}$ that is mapped to H by the transduction. In general, this is a hard problem, requiring a *combinatorial* understanding of the structure of the resulting classes $\mathscr{D}$.

The above principle has so far been successfully applied in the setting of graph classes of bounded treewidth and monadic second-order transductions: it was shown by Courcelle, Makowsky and Rotics [4] that transductions of classes of bounded treewidth can be combinatorially characterized as classes of bounded cliquewidth. This, combined with Oum's result [32] gives a fixed-parameter algorithm for model-checking monadic second-order logic on classes of bounded cliquewidth. More recently, the same principle, but for first-order logic, has been applied to graphs of bounded degree [14], 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 paper: a class of graphs has *structurally bounded expansion* if it is the image of a class of bounded expansion under some fixed first-order transduction. This paper 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 complex graphs out of simple graphs. A more combinatorial approach is the method of *decompositions* (or *colorings*) [31], which we reformulate below in terms of *covers*. This method can be used to provide a characterization of bounded expansion classes in terms of very simple graph classes, namely classes of *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 of bounded treedepth $\mathcal{T}$ such that for every $G \in \mathscr{C}$, 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 $\mathcal{T}$. A consequence of a result by Nešetřil and Ossona de Mendez [29] 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 decomposition method allows to lift algorithmic, logical, and structural properties from classes of bounded treedepth to classes of bounded expansion. For instance, this was used to show tractability of first-order model-checking on bounded expansion classes [8, 21].

An analogue of treedepth in the dense world is the concept of *shrubdepth*, introduced in [17]. 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 of bounded treedepth. Applying the method of decompositions 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 $\mathcal{B}$ of 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 $\mathcal{B}$. Shrubdepth properly generalizes treedepth and consequently classes admitting low shrubdepth covers properly extend bounded expansion classes.

It was observed earlier [27] that for every fixed $r \in \mathbb{N}$ and every class $\mathscr{C}$ of 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.

Our contributions. Our main result, Theorem 15, states that the two notions introduced above are the same: a class of graphs $\mathscr{C}$ has structurally bounded expansion if, and only if, it has bounded shrubdepth covers. That is, transductions of classes of bounded expansion are the same as classes with low shrubdepth covers (cf. Figure 1). This gives a combinatorial characterization of structurally bounded expansion classes, which is an important step towards their algorithmic treatment.

One of the key ingredients of our proof is a quantifier-elimination result (Theorem 16) for transductions on classes of structurally bounded expansion. This result strengthens in several ways similar results for bounded expansion classes due to Dvořák, Král', and Thomas [8], Grohe and Kreutzer [21] and Kazana and Segoufin [26]. Our assumption is more general, as they assume that $\mathscr{C}$ has bounded expansion, and here $\mathscr{C}$ is only

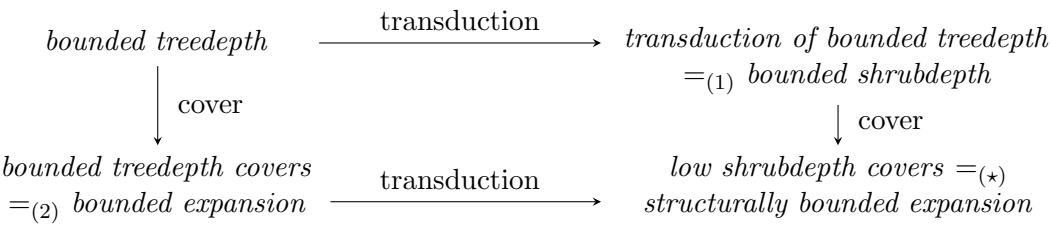


Figure 1: The nodes in the diagram depict properties of graph classes, and the arrows depict operations on properties of graph classes. Equality (1) is by [17]. Equality (2) is by [29]. Equality ($\star$) is the main result of this paper, Theorem 15.

required to have low shrubdepth covers. Also, our conclusion is stronger, as their results provide quantifier-free formulas involving some unary functions and unary predicates which are computable algorithmically, whereas our result shows that these functions can be defined using very restricted transductions. Quantifier-elimination results of this type proved to be useful for the model-checking problem on bounded expansion classes [8, 21, 26], and this is also the case here.

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 to reduce 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 40 we show that it is enough to compute a 2-cover of a given graph G from a structurally bounded expansion class, in order to obtain an algorithm for the model-checking problem for such classes. We conjecture that such an algorithm exists and that therefore first-order model-checking is fixed-parameter tractable on any class of graphs of structurally bounded expansion. We leave this problem for future work.

Organization. In Section 2 we collect basic facts about logic, transductions, treedepth, shrubdepth and the notion of 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 aim to present an easy to follow proof of our main result. For this reason, we present proofs of the key lemmas in the main body of the paper, while rather technical results that disturb the flow of ideas are presented in full detail in the appendix.

2 Preliminaries

Basic notation. We use standard graph notation. All graphs considered in this paper 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 (partial) functions is the identity function. For an integer k, we denote $[k] = \{1, \ldots, k\}$.

2.1 Structures, logic, and transductions

Structures and logic. A *signature* Σ is a finite set of relation symbols, each with prescribed arity that is a non-negative integer, and unary function symbols. A *structure* $\mathbf{A}$ over Σ consists of a finite universe $V(\mathbf{A})$ and *interpretations* of symbols from 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 *partial* function $f^{\mathbf{A}} \colon V(\mathbf{A}) \rightharpoonup V(\mathbf{A})$. We drop the superscipt when the structure is clear from the context, thus identifying each symbol with its interpretation. If $\mathbf{A}$ is a structure and $X \subseteq V(\mathbf{A})$ then we define the *substructure* of $\mathbf{A}$ induced by X in the usual way except that a unary function $f(x)$ in $\mathbf{A}$ becomes undefined on all $x \in X$ for which $f(x) \notin X$. The *Gaifman graph* of a structure $\mathbf{A}$ is the graph with vertex set $V(\mathbf{A})$ where two 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 partial function f in $\mathbf{A}$.

For a signature Σ, we consider standard first-order logic over Σ. Let us clarify the usage of function symbols. A *term* $\tau(x)$ is a finite composition of function symbols applied to a variable x. In a structure $\mathbf{A}$, given an evaluation of x, the term $\tau(x)$ either evaluates to some element of $\mathbf{A}$ in the natural sense, or is *undefined* if during the evaluation we encounter an element that does not belong to the domain of the function that is to be applied next. In first order logic over Σ we allow usage of atomic formulas of the following form:

- $R(\tau_1(x_1), \ldots, \tau_k(x_k))$ for a relation symbol R of arity k, terms $\tau_1, \ldots, \tau_k$, and variables $x_1, \ldots, x_k$;
- $\tau_1(x_1) = \tau_2(x_2)$ for terms τ_1, τ_2 and variables x_1, x_2; and
- $\mathsf{dom}_f(\tau(x))$ for term τ and variable x.

Here, the predicate $\mathsf{dom}_f(\tau(x))$ checks whether $\tau(x)$ belongs to the domain of f. The semantics are defined as usual, however an atomic formula is false if any of the terms involved is undefined. Based on these atomic formulas, the syntax and semantics of first order logic is defined in the expected way.

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 Λ, by a Λ-*colored* graph we mean a graph enriched by a unary predicate U_λ for every $\lambda \in \Lambda$. 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 distance between x and the root in the connected component of x 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.

Transductions. We now define the notion of transduction used in the sequel. A *transduction* is a special type of first-order interpretation with set parameters, which we see here (from a computational point of view) as a nondeterministic operation that maps input structures to output structures. Transductions are defined as compositions of *atomic operations* listed below.

An **extension** operation is parameterized by a first-order formula $\varphi(x_1, \ldots, x_k)$ and a relation symbol R. Given an input structure $\mathbf{A}$, it outputs the structure $\mathbf{A}$ extended by the relation R interpreted as the set of k-tuples of elements satisfying φ in $\mathbf{A}$. A **restriction** operation is parameterized by a unary formula $\psi(x)$. Applied to a structure $\mathbf{A}$ it outputs the substructure of $\mathbf{A}$ induced by all elements satisfying ψ. A **reduct** operation is parameterized by a relation symbol R, and results in removing the relation R from the input structure. **Copying** is an operation which, given a structure $\mathbf{A}$ outputs a disjoint union of two copies of $\mathbf{A}$ extended with a new unary predicate which marks the newly created vertices, and a symmetric binary relation which connects each vertex with its copy. A **function extension** operation is parameterized by a binary formula $\varphi(x, y)$ and a function symbol f, and extends a given input structure by a partial function f defined as follows: $f(x) = y$ if y is the unique vertex such that $\varphi(x, y)$ holds. Note that if there is no such y or more than one such y, then $f(x)$ is undefined. Finally, suppose σ is function that maps each structure $\mathbf{A}$ to a nonempty family $\sigma(\mathbf{A})$ of subsets of its universe. A **unary lift** operation, parameterized by σ, takes as input a structure $\mathbf{A}$ and outputs the structure $\mathbf{A}$ enriched by a unary predicate X interpreted by a nondeterministically chosen set $U \in \sigma(\mathbf{A})$.

We remark that function extension operations can be simulated by extension operations, defining the graphs of the functions in the obvious way. They are, however, useful as a means of extending the expressive power of transductions in which only quantifier-free formulas are allowed, as defined below.

Transductions are defined inductively: every atomic transduction is a transduction, and the composition of two transductions I and J is the transduction $\mathsf{I}; \mathsf{J}$ that, given a structure $\mathbf{A}$, first applies I to $\mathbf{A}$ and then J to the output $\mathsf{I}(\mathbf{A})$. A transduction is *deterministic* if it does not use unary lifts. In this case, for every input structure there is exactly one output structure. A transduction is *almost quantifier-free* if all formulas that parameterize atomic operations comprising it are quantifier-free[1], and is *deterministic almost quantifier-free* if it additionally does not use unary lifts.

[1] We use the adverb "almost" to indicate that such transductions still can access elements that are not among its free variables via functions.

If $\mathscr{C}$ is a class of structures, we write $\mathsf{I}(\mathscr{C})$ for the class which contains all possible outputs $\mathsf{I}(\mathbf{A})$ for $\mathbf{A} \in \mathscr{C}$. We say that two transductions I and J are *equivalent* on a class $\mathscr{C}$ of structures if every possible output of $\mathsf{I}(\mathbf{A})$ is also a possible output of $\mathsf{J}(\mathbf{A})$, and vice versa, for every $\mathbf{A} \in \mathscr{C}$.

It may happen that an atomic operation I is undefined for a given input structure $\mathbf{A}$. For example, for an extension operation parametrized by a first order formula φ using a relation symbol R, if the input structure $\mathbf{A}$ does not carry the symbol R, then $\mathsf{I}(\mathbf{A})$ is undefined according to the above definition. This will never occur in our constructions. However, for completeness, we may define $\mathsf{I}(\mathbf{A})$ as a fixed structure $\perp$ in such situations.

When considering a composition of atomic operations, we avoid overriding symbols by later operations, i.e., we always assume that subsequent atomic operations create relation symbols which are distinct from previously created relations symbols and also from symbols in the original signature. Since every transduction I is a composition of finitely many atomic operations, the result of I applied to a structure over a finite signature Σ will be again a structure over a finite signature Γ, which depends on Σ and I only (unless the result is undefined).

Example 1. Let $\mathscr{C}$ be the class of rooted forests of depth at most d, for some fixed $d \in \mathbb{N}$. We describe an almost quantifier-free transduction which defines the *parent function* in $\mathscr{C}$. First, using unary lifts introduce $d+1$ unary predicates $D_0, ..., D_d$, where D_i marks the vertices of the input tree which are at distance i from a root. Next, using a function extension, define a partial function f which maps a vertex v in the input tree to its parent, or is undefined in case of a root. This can be done by a quantifier-free formula, which selects those pairs x, y such that x and y are adjacent and $D_i(x)$ implies $D_{i-1}(y)$.

It will sometimes be convenient to work with the encoding of bounded-depth trees and forests as node sets endowed with the parent function, rather than graphs with prescribed roots. As seen in Example 1, these two encodings can be translated to each other by means of almost quantifier-free transductions, which render them essentially equivalent.

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 general transductions and one for almost quantifier-free transductions. The proofs are standard, for completeness, we give them in the appendix.

Lemma 2 ($\star$). *Let I be a transduction. Then I is equivalent to a transduction of the form*

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

where

- L *is a sequence of unary lifts;*
- C *is a sequence of copying operations;*

- F *is a sequence of function extension operations, one for each function on the output;*
- E *is a sequence of extension 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* $\mathsf{L};\mathsf{C}$*. In particular, none of these formulas uses any function or relation introduced by an atomic operation in* $\mathsf{F};\mathsf{E}$*.*

Lemma 3 ($\star$). *Every almost quantifier-free transduction is equivalent to an almost quantifier-free transduction that first applies a sequence of unary lifts and then applies a deterministic almost quantifier-free transduction.*

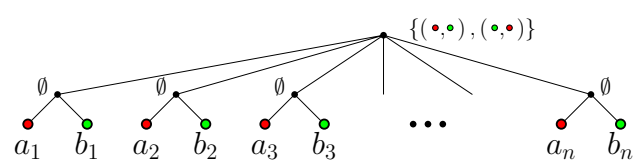
2.2 Treedepth and shrubdepth

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. Equivalently, $\mathscr{C}$ has bounded treedepth if there is some number k such that no graph in $\mathscr{C}$ contains a simple path of length k [31]. 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.

Shrubdepth. The following notion of *shrubdepth* has been proposed in [17] as a dense analogue of treedepth. Originally, shrubdepth was defined using the notion of *tree-models*. We present an equivalent definition basing on the notion of *connection models*, introduced in [17] under the name of *m-partite cographs* of bounded depth.

A *connection model* with labels from Λ is a rooted labeled tree T where each leaf x is labeled by a label $\lambda(x) \in \Lambda$, and each non-leaf node v is labeled by a (symmetric) binary relation $C(v) \subseteq \Lambda \times \Lambda$. 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 $(\lambda(x), \lambda(y)) \in C(v)$, 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 4. 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:



We can naturally extend the definition above to structures with unary functions by regarding each unary function by a binary relation selecting all (argument, value) pairs.

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

Shrubdepth 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 5. 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 tree T of depth d with leaf set $V(G)$, equipped with unary predicates $W_0, \ldots, W_d$ on the leaves. Each child s of the root in T corresponds to a connected component C_s of the graph G' obtained from G by flipping the adjacency within W_0, such that the subtree of T rooted at s, together with the unary predicates $W_1, \ldots, W_d$ restricted to $V(C_s)$, form an SC-decomposition of C_s.

We will make use of the following properties, where the first one follows from the definition of shrubdepth, and the remaining ones follow from [17].

Proposition 6. *Let $\mathscr{C}$ be a class of graphs. Then:*

1. *If $\mathscr{C}$ has bounded shrubdepth then the class of all induced 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 I is a transduction that outputs colored graphs, then $\mathsf{I}(\mathscr{C})$ has bounded shrubdepth.*

It is well-known (see [23]) that in the absence of large bi-cliques (complete bipartite graphs) a graph of bounded cliquewidth has in fact bounded treewidth. The same holds also for shrubdepth and treedepth. The lemma is proved by an easy induction on the depth of the connection models.

Lemma 7 ($\star$). *A class of 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.*

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 will use the following lemma.

Lemma 8. *Let $\mathscr{C}$ be a class of (colored) graphs of bounded expansion and let C be a copy operation. Then $\mathsf{C}(\mathscr{C})$ is a class of colored graphs of 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; see e.g., [31, Proposition 4.6]. □

The connection between treedepth and graph classes of 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 $c\colon V(G) \to C$ with $|C| \leqslant N_p$.

Theorem 9 ([29]). *A class of graphs $\mathscr{C}$ has bounded expansion if, and only if, it has low treedepth colorings.*

3 Main results

In this section we introduce two notions which generalize the concept of bounded expansion. Then we state the main results and outline the proof. First, we introduce classes of *structurally bounded expansion*. This notion arises from closing bounded expansion graph classes under transductions.

Definition 10. A class $\mathscr{C}$ of graphs has *structurally bounded expansion* if there exists a class of graphs $\mathscr{D}$ of bounded expansion and a transduction I such that $\mathscr{C} \subseteq \mathsf{I}(\mathscr{D})$.

The second notion, *low shrubdepth covers*, arises from the low treedepth coloring characterisation of bounded expansion (see Theorem 9) by replacing treedepth by its dense counterpart, shrubdepth. For convenience, we formally define this in terms of *covers*.

Definition 11. A *cover* of a graph G is a family $\mathcal{U}_G$ of subsets of $V(G)$ such that $\bigcup \mathcal{U}_G = V(G)$. A cover $\mathcal{U}_G$ is a *p-cover*, where $p \in \mathbb{N}$, if every set of at most p vertices is contained in some $U \in \mathcal{U}_G$. If $\mathscr{C}$ is a class of graphs, then a (p-)cover of $\mathscr{C}$ is a family $\mathcal{U} = (\mathcal{U}_G)_{G \in \mathscr{C}}$, where $\mathcal{U}_G$ is a (p-)cover of G. The cover $\mathcal{U}$ is *finite* if $\sup\{|\mathcal{U}_G| \colon G \in \mathscr{C}\}$ is finite. Let $\mathscr{C}[\mathcal{U}]$ denote the class of graphs $\{G[U] \colon G \in \mathscr{C}, U \in \mathcal{U}_G\}$. We say that the cover $\mathcal{U}$ has *bounded treedepth* (respectively, bounded shrubdepth) if the class $\mathscr{C}[\mathcal{U}]$ has bounded treedepth (respectively, shrubdepth).

Example 12. Let $\mathscr{T}$ be the class of trees and let $p \in \mathbb{N}$. We construct a finite p-cover $\mathcal{U}$ of $\mathscr{T}$ which has bounded treedepth. Given a rooted tree T, let $\mathcal{U}_T = \{U_0, \ldots, U_p\}$, 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 height 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 $\mathscr{T}$ of bounded treedepth.

In analogy to low treedepth colorings, we can now characterize graph classes of 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 9.

Lemma 13 ($\star$). *A class of graphs has bounded expansion if, and only if, it has low treedepth covers.*

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

Definition 14. 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 13 together with Proposition 6(3) imply that every class of bounded expansion has low shrubdepth covers. Our main result is the following theorem.

Theorem 15. *A class of graphs has structurally bounded expansion if, and only if, it has low shrubdepth covers.*

As a byproduct of our proof of Theorem 15 we obtain the following quantifier-elimination result, which we believe is of independent interest.

Theorem 16. *Let $\mathscr{C}$ be a class of colored graphs which has low shrubdepth covers. Then every transduction* I *is equivalent to some almost quantifier-free transduction* J *on $\mathscr{C}$.*

We now outline the proof of Theorem 15 and Theorem 16. Both theorems follow easily from Proposition 18 and Proposition 19 stated below. These are proved in subsequent sections.

We start with the following lemma, which intuitively shows that covers commute with almost quantifier-free transductions.

Lemma 17. *If a class of graphs $\mathscr{C}$ has low shrubdepth covers and I is an almost quantifier-free transduction that outputs colored graphs, then $\mathsf{I}(\mathscr{C})$ also has low shrubdepth covers.*

Proof (sketch). The idea is that for any almost quantifier-free transduction I there is a constant c such any induced substructure of $\mathsf{I}(G)$ on p elements depends only on an induced substructure of G of size $p \cdot c$. In particular, a $(p \cdot c)$-cover of G induces a p-cover of $\mathsf{I}(G)$. Moreover, as having bounded shrubdepth is preserved by transductions, a low shrubdepth cover of $\mathscr{C}$ induces a low shrubdepth cover of $\mathsf{I}(\mathscr{C})$. The details are presented in Section 4. □

The main novel ingredient in our proof of Theorem 15 and Theorem 16 is the following result, which intuitively states that classes with low shrubdepth covers are bi-definable with classes of bounded expansion, using almost quantifier-free transductions.

Proposition 18. *Suppose $\mathscr{C}$ is a class of graphs with low shrubdepth covers. Then there is a pair of transductions S and I, where S is almost quantifier-free and I is deterministic almost quantifier-free, such that $\mathsf{S}(\mathscr{C})$ is a class of colored graphs of bounded expansion and $\mathsf{I}(\mathsf{S}(G)) = \{G\}$ for each $G \in \mathscr{C}$.*

Clearly, Proposition 18 implies that $\mathscr{C}$ has structurally bounded expansion, since it can be obtained as a result of transduction I to a class $\mathsf{S}(\mathscr{C})$ of bounded expansion. Thus, the right-to-left implication of Theorem 15 is a corollary of the proposition. The proof of Proposition 18 is presented in Section 5. We sketch the rough idea below.

Proof (sketch). First, in Lemma 31 of Section 5.2, we prove the special case where $\mathscr{C}$ is a class of graphs of bounded shrubdepth, and for those we prove bi-definability with classes of trees of bounded depth. In particular, if $\mathscr{D}$ is a class of graphs of bounded shrubdepth, then there is a pair of almost quantifier-free transductions T, I_0 such that $\mathsf{T}(\mathscr{D})$ is a class of colored trees of bounded depth and such that $\mathsf{I}_0(\mathsf{T}(H)) = \{H\}$ for all $H \in \mathscr{D}$. Lemma 31 is the combinatorial core of this paper.

To prove Proposition 18, we lift Lemma 31 to the general case using covers, as follows. Let $\mathscr{C}$ be a class with low shrubdepth covers and let $\mathcal{U}$ be a 2-cover of $\mathscr{C}$ of bounded shrubdepth, and let N be such that $|\mathcal{U}_G| \leqslant N$ for $G \in \mathscr{C}$. We apply the bounded shrubdepth case to the class $\mathscr{D} = \mathscr{C}[\mathcal{U}]$, yielding almost quantifier-free transductions T and I_0 as above. The transduction S works as follows: given a graph $G \in \mathscr{C}$, introduce N unary predicates marking the cover $\mathcal{U}_G$ of G, and for each $U \in \mathcal{U}_G$, apply T to the induced subgraph $G[U]$ of G, yielding a colored tree $\mathsf{T}(G[U])$. Define $\mathsf{S}(G)$ as the union of the trees $\mathsf{T}(G[U])$, for $U \in \mathcal{U}_G$. As $\mathcal{U}_G$ is a 2-cover of G, G is the union of the induced graphs $G[U]$ for $U \in \mathcal{U}_G$. As each graph $G[U]$ can be recovered from the tree $\mathsf{T}(G[U])$ using the inverse transduction I_0, it follows that G can be recovered from the union $\mathsf{S}(G)$. This yields the inverse transduction I such that $\mathsf{I}(\mathsf{S}(G)) = \{G\}$. As S is almost quantifier-free by construction, it follows from Lemma 17 that $\mathsf{S}(\mathscr{C})$ is a class with low shrubdepth covers. Moreover, each graph in $\mathsf{S}(\mathscr{C})$ is a union of at most N trees, so it does not contain $K_{N+1,N+1}$ as a subgraph. It follows from Lemma 7 that the low shrubdepth cover of $\mathsf{S}(\mathscr{C})$ is in fact a low treedepth cover. Hence, $\mathsf{S}(\mathscr{C})$ has low treedepth covers, i.e., has bounded expansion. □

Theorem 16, and the remaining implication in Theorem 15 are consequences of the following result.

Proposition 19. *Let $\mathscr{C}$ be a class of graphs of bounded expansion and let* I *be a transduction. Then* I *is equivalent to an almost quantifier-free transduction* J *on* $\mathscr{C}$*.*

We note that Proposition 19 is a strengthening of similar statements provided by Dvořák et al. [9] and of Grohe and Kreutzer [21], and could be derived by a careful analysis of their proofs. In Section 6 we provide a self-contained proof, which we believe is simpler than the previous proofs, and is sketched below.

Proof (sketch). We use the characterization of bounded expansion classes as those which have low treedepth covers. We first prove Proposition 19 for forests of bounded depth. This can be handled by a direct (although slightly cumbersome) combinatorial argument, similarly as in [9]. In Appendix F.2 we present an argument using tree automata.

The statement for classes of forests of bounded depth then easily lifts to classes of bounded treedepth. Here we use the fact that in a graph of bounded treedepth it is possible to encode a depth-first search forest of bounded depth, by using unary predicates marking the depth of each node in the spanning forest.

We then lift the result from classes of bounded treedepth using covers. Specifically, suppose for simplicity that the transduction I is a single extension operation, parametrized by a formula ψ. We then proceed by induction on the structure of the formula ψ and show that it can be replaced by a quantifier-free formula, at the cost of introducing unary functions defined by an almost quantifier-free transduction.

In the inductive step, the only nontrivial case is the one of existential quantification, i.e., of formulas of the form

$$\psi(\bar{y}) = \exists x.\varphi(x, \bar{y}),$$

where $\varphi(x, \bar{y})$ may be assumed to be a quantifier-free formula involving unary functions, by inductive assumption. We consider a p-cover $\mathcal{U}$ of $\mathscr{C}$ where p is a constant such that there are at most p different terms occurring in $\varphi(x, \bar{y})$. Since $\mathscr{C}$ has bounded expansion, we may assume that the cover $\mathcal{U}$ has bounded treedepth, and that there is a constant $N \in \mathbb{N}$ such that $|\mathcal{U}_G| \leqslant N$ for all $G \in \mathscr{C}$. For a fixed graph $G \in \mathscr{C}$, the existentially quantified variable x must be in one of the sets $U \in \mathcal{U}_G$. Therefore, the formula $\psi(\bar{y})$ is equivalent to a disjunction of at most N formulas $\psi_i(\bar{y})$, for $i = 1, \ldots, N$, where each formula $\psi_i(\bar{y})$ performs existential quantification restricted to the ith set in $\mathcal{U}_G$ (where $\mathcal{U}_G$ is ordered arbitrarily). By the special case of the proposition proved for classes of bounded treedepth, $\psi_i(\bar{y})$ is equivalent to a quantifier-free formula on $\mathscr{C}[\mathcal{U}]$ (the quantifier-free formula uses unary functions introduced by almost quantifier-free transductions). Reassuming, ψ is equivalent on G to a disjunction of quantifier-free formulas involving unary functions that are introduced by almost quantifier-free transductions. This deals with the inductive step. □

We finally show how to conclude Theorem 15 and Theorem 16 from Lemma 17, Proposition 18 and Proposition 19.

Proof (of Theorem 15). As observed, the right-to-left implication of Theorem 15 follows from Proposition 18. We now show the left-to-right implication.

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

By Lemma 13, $\mathscr{C}$ has low treedepth covers. Applying Proposition 19 yields an almost quantifier-free transduction J such that $\mathsf{I}(\mathscr{C}) = \mathsf{J}(\mathscr{C})$. As $\mathscr{C}$ in particular has low shrubdepth covers (cf. Proposition 6 (3)), we may apply Lemma 17 to J and $\mathscr{C}$ to deduce that $\mathsf{J}(\mathscr{C}) = \mathsf{I}(\mathscr{C})$ has low shrubdepth covers. □

Proof (of Theorem 16). Proposition 18 allows to reduce the theorem to the case of classes of bounded expansion, as almost quantifier-free transductions are closed under composition. The case of bounded expansion classes is handled by Proposition 19. □

It remains to provide the details of the proofs of Lemma 17, Proposition 18 and Proposition 19. This is done in Section 4, Section 5 and Section 6, respectively. After that, in Section 7 we conclude with a preliminary algorithmic result concerning the model-checking problem for first-order logic on classes with structurally bounded expansion.

4 Proof of Lemma 17 *(almost quantifier-free transductions commute with covers)*

In this section we prove Lemma 17, which we restate for convenience.

Lemma 17. *If a class of graphs $\mathscr{C}$ has low shrubdepth covers and I is an almost quantifier-free transduction that outputs colored graphs, then $\mathsf{I}(\mathscr{C})$ also has low shrubdepth covers.*

We start with formulating the following lemma which states that almost quantifier-free transductions are, in a certain sense, local.

Lemma 20. *For every deterministic almost quantifier-free transduction I there is a constant $c \in \mathbb{N}$ such that the following holds. For every structure $\mathbf{A}$ and every element v of $\mathsf{I}(\mathbf{A})$ there is a set $S_v \subseteq V(\mathbf{A})$ of size at most c such that for any sets U, W with $W \subseteq V(\mathsf{I}(\mathbf{A}))$ and $U \subseteq V(\mathbf{A})$, if $U \supseteq \bigcup_{v \in W} S_v$, then*

$$\mathsf{I}(\mathbf{A})[W] = \mathsf{I}(\mathbf{A}[U])[W].$$

In order to prove the lemma, we define the following notions of *dependency* and *support*.

Definition 21. Suppose that $\tau(v) = (f_p \circ \cdots \circ f_1)(v)$ is a term. For a structure $\mathbf{A}$ carrying partial functions $f_1, \ldots, f_p$, we say that an element $v \in V(\mathbf{A})$ *τ-depends* with respect to τ on itself and all elements of the form $(f_p \circ \cdots \circ f_i)(v)$ for $i \in [p]$, whenever defined. For a quantifier-free formula $\varphi(x_1, \ldots, x_k)$, an element $v \in V(\mathbf{A})$ *φ-depends* on all elements on which v τ-depends, for any term τ appearing in φ. For an element v, the set of elements on which v φ-depends in $\mathbf{A}$ will be denoted by $\mathsf{cl}^{\mathbf{A}}_{\varphi}(v)$; note that the

size of this set 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 22. Suppose I is a deterministic almost quantifier-free transduction, and let $\mathbf{A}$ be an input structure. For an element $v \in V(\mathsf{I}(\mathbf{A}))$ and a subset $S \subseteq V(\mathbf{A})$, we now define what it means that v *is* I*-supported* by S. We first define this for atomic operations (note that unary lifts are excluded since I is assumed to be deterministic):

- If I is a reduct operation or a copy operation, then v is I-supported by S if and only if $v \in S$.
- If I is a restriction or an extension operation, say parameterized by a formula φ, then v is I-supported by S if and only if $\mathsf{cl}_\varphi^{\mathbf{A}}(v) \subseteq S$.
- Suppose I is a function extension operation, say introducing a partial function f using a binary formula $\varphi(x, y)$. Then v is I-supported by S if and only if $\mathsf{cl}_\varphi^{\mathbf{A}}(v) \subseteq S$ and the following holds:
  - if there exists exactly one $w \in V(\mathbf{A})$ for which $\varphi(v, w)$ holds, then $\mathsf{cl}_\varphi^{\mathbf{A}}(w) \subseteq S$.
  - if there are at least two elements $w \in V(\mathbf{A})$ for which $\varphi(v, w)$ holds, then $\mathsf{cl}_\varphi^{\mathbf{A}}(w) \subseteq S$ for at least two distinct such elements w.

Finally, for non-atomic deterministic almost quantifier-free transductions the notion of I-supporting is defined by induction on the structure of the transduction. Suppose I is the composition $\mathsf{I}_1; \mathsf{I}_2$ of two transductions. Then $v \in V(\mathsf{I}(\mathbf{A}))$ is I-supported by $S \subseteq V(\mathbf{A})$ if there exists a subset $T \subseteq V(\mathsf{I}_1(\mathbf{A}))$ and, for each $w \in T$, a subset $S_w \subseteq S$ such that v is I_2-supported by T and each $w \in T$ is I_1-supported by S_w.

The notion of supporting is trivially closed under taking supersets: if v is I-supported by S, then v is also I-supported by any superset of S.

Proof (of Lemma 20). By induction on the definition of an almost quantifier-free transduction I it is easy to see that for every $v \in V(\mathsf{I}(\mathbf{A}))$ there is a set $S_v \subseteq V(\mathbf{A})$ such that v is I-supported by S_v and $|S_v|$ is bounded by a constant, possibly depending on I.

By induction we also observe that if $W \subseteq V(\mathsf{I}(\mathbf{A}))$ and $U \subseteq V(\mathbf{A})$ are such that every $v \in W$ is I-supported by U then

$$\mathsf{I}(\mathbf{A})[W] = \mathsf{I}(\mathbf{A}[U])[W].$$

This proves the lemma. □

We can now prove Lemma 17.

Proof (of Lemma 17). Let $\mathscr{C}$ be a class with low shrubdepth covers and let I be an almost quantifier-free transduction that outputs colored graphs. We show that $\mathsf{I}(\mathscr{C})$ has low shrubdepth covers. By normalizing I as described in Lemma 3, we may assume that I is of the form $\mathsf{L};\mathsf{J}$, where L is a sequence of unary lifts and J is deterministic almost quantifier-free. As $\mathscr{C}$ has low shrubdepth covers, the class $\mathscr{D} = \mathsf{L}(\mathscr{C})$ also has low shrubdepth covers (this is implied by Proposition 6(4)). Moreover, $\mathsf{I}(\mathscr{C}) = \mathsf{J}(\mathscr{D})$. Therefore, it suffices to focus on the deterministic almost quantifier-free transduction J applied to the class $\mathscr{D}$. Note that $\mathscr{D}$ is a class of colored graphs, i.e., graphs with unary predicates on their vertices.

Let c be the constant provided by Lemma 20 for the transduction J. We need to find, for every $p \in \mathbb{N}$, a finite p-cover of $\mathsf{J}(\mathscr{D})$ of bounded shrubdepth, so let us fix p. Let $\mathcal{U}$ be a finite $(c \cdot p)$-cover of $\mathscr{D}$ of bounded shrubdepth. For a graph $G \in \mathscr{D}$ and $U \in \mathcal{U}_G$, let $W_U \subseteq V(\mathsf{J}(G))$ be the set of those elements v of $\mathsf{J}(G)$ such that $S_v \subseteq U$, where S_v is as obtained from Lemma 20 applied to the deterministic almost quantifier-free transduction J.

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

$$\mathcal{W}_{\mathsf{J}(G)} = \{W_U \colon U \in \mathcal{U}_G\} \qquad \text{for every graph } G \in \mathscr{D}.$$

Clearly $|\mathcal{W}_{\mathsf{J}(G)}| \leqslant |\mathcal{U}_G|$, so $\mathcal{W}$ is finite as well. We need to verify that $\mathcal{W}$ is a p-cover and that it has bounded shrubdepth.

To see that $\mathcal{W}$ is a p-cover, take any p elements $w_1, \ldots, w_p$ of $\mathsf{J}(G)$. Let $S = \bigcup_{i=1}^{p} S_{w_i}$. Then $|S| \leqslant c \cdot p$, hence there exists $U \in \mathcal{U}_G$ with $S \subseteq U$. We conclude that $\{w_1, \ldots, w_p\} \subseteq W_U \in \mathcal{W}_G$.

To see that $\mathcal{W}$ is a bounded shrubdepth cover, observe that by assumption $\mathscr{D}[\mathcal{U}]$ has bounded shrubdepth, hence by Proposition 6(4) we find that $\mathsf{J}(\mathscr{D}[\mathcal{U}])$ also has bounded shrubdepth. By Lemma 20, for each $G \in \mathscr{D}$ and $W_U \in \mathcal{W}_{\mathsf{J}(G)}$, the induced substructure $\mathsf{J}(G)[W_U]$ is equal to $\mathsf{J}(G[U])[W_U]$. Now it suffices to note that $\mathsf{J}(G[U]) \in \mathsf{J}(\mathscr{D}[\mathcal{U}])$, hence $\mathsf{J}(G)[W_U]$ belongs to the hereditary closure of $\mathsf{J}(\mathscr{D}[\mathcal{U}])$, which also has bounded shrubdepth by Proposition 6(1). □

5 Proof of Proposition 18 *(bi-definability of classes with low shrubdepth covers and classes of bounded expansion)*

In this section we prove Proposition 18, which we repeat for convenience.

Proposition 18. *Suppose $\mathscr{C}$ is a class of graphs with low shrubdepth covers. Then there is a pair of transductions S and I, where S is almost quantifier-free and I is deterministic almost quantifier-free, such that $\mathsf{S}(\mathscr{C})$ is a class of colored graphs of bounded expansion and $\mathsf{I}(\mathsf{S}(G)) = \{G\}$ for each $G \in \mathscr{C}$.*

Clearly, Proposition 18 implies that $\mathscr{C}$ has structurally bounded expansion, since it can be obtained as a result of transduction I to a class $\mathsf{S}(\mathscr{C})$ of bounded expansion. Thus, the right-to-left implication of Theorem 15 is a corollary of the proposition.

The idea of the proof of Proposition 18 is as follows. We first prove in Lemma 23 of Section 5.1 that connected components in graphs of bounded shrubdepth are definable

by almost quantifier-free transductions. We use Lemma 23 to first prove Proposition 18 for the special case where $\mathscr{C}$ is a class of graphs of bounded shrubdepth, and for those we prove bi-definability with classes of trees of bounded depth. This is done in Lemma 31 of Section 5.2. Then, we conclude the general case in Section 5.3, by lifting Lemma 31 using covers.

5.1 Defining connected components in graphs of bounded shrubdepth

The following lemma is the combinatorial core of our proof of Proposition 18.

Lemma 23. *Let $\mathscr{C}$ be a class of graphs of bounded shrubdepth. There is an almost quantifier-free transduction F such that for a given $G \in \mathscr{C}$, every output of F on G is equal to G enriched by a function $g\colon V(G) \to V(G)$ such that $g(v) = g(w)$ if and only if v and w are in the same connected component of G.*

The rest of Section 5.1 is devoted to the proof of Lemma 23.

Guidance systems. We first introduce the notions of *guidance systems* and of functions *guided* or *guidable* by them. This is a combinatorial abstraction for functions computable by almost quantifier-free transductions.

Let G be a graph. 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 partial function $f\colon V(G) \rightharpoonup V(G)$ is *guided* by the guidance system $\mathcal{U}$ if for every $x \in V(G)$ for which $f(x)$ is defined and different than x, there is some $U \in \mathcal{U}$ such that $f(x)$ is the unique neighbor of v in U. Finally, a partial function $f\colon V(G) \rightharpoonup V(G)$ 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 that f is guided by $\mathcal{U}$.

Observe that an ℓ-guidable partial function maps each vertex v from its domain to a vertex in the same connected component as v. The following lemmas will be useful for operating on guidable functions.

Lemma 24 ($\star$). *Let G be a graph and suppose $g\colon V(G) \rightharpoonup V(G)$ is a partial function such that the restriction $g|_C$ of g to each connected component C of G is ℓ-guidable. Then g is ℓ-guidable.*

Lemma 25 ($\star$). *Let G be a graph and let $g_1, \ldots, g_s\colon V(G) \rightharpoonup V(G)$ be partial functions, where g_i is ℓ-guidable for each $i \in [s]$. If $g\colon V(G) \rightharpoonup V(G)$ is a partial function such that for every $x \in V(G)$ there is some $i \in [s]$ such that $g(x) = g_i(x)$, then g is $(\ell \cdot s)$-guidable.*

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

Lemma 26 ($\star$). *Let $\mathscr{C}$ be a class of graphs and let $\ell \in \mathbb{N}$ be fixed. Suppose that each $G \in \mathscr{C}$ is equipped with an ℓ-guidable function $f_G\colon V(G) \rightharpoonup V(G)$. Then there exists an almost quantifier-free transduction which given $G \in \mathscr{C}$ has exactly one output: the graph G enriched with f_G.*

We will use the following fact stating that graphs of bounded shrubdepth do not admit long induced paths.

Lemma 27 ([16]). *For every class $\mathscr{C}$ of graphs of 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.*

Spanning forests. For a graph G and a function $g\colon V(G) \to V(G)$, we say that g *defines a spanning forest of depth r* on G if g is guarded by G and the r-fold composition $g^r\colon V(G) \to V(G)$ 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 $g^r(u) = g^r(v)$.

The following lemma states that guidance systems can define shallow spanning forests in graph classes of bounded shrubdepth.

Lemma 28. *For every class $\mathscr{C}$ of graphs of bounded shrubdepth there exist constants $q, r \in \mathbb{N}$ such that for every $G \in \mathscr{C}$ there is a function $f_G\colon V(G) \to V(G)$ which is q-guidable as a partial function on G and defines a spanning forest of depth r on G.*

We first show how Lemma 23 follows from Lemma 28.

Proof (of Lemma 23). By Lemma 26, there is an almost quantifier-free transduction I which, given a graph $G \in \mathscr{C}$ on input, constructs the function f_G obtained from Lemma 28. Now let $g = f_G^r$ be the r-fold composition of f. Clearly, g can be computed by an almost quantifier-free transduction using a single function extension operation, making use of the function f_G constructed by I. As g is constant on every connected component of G, Lemma 23 follows. □

It remains to prove Lemma 28.

Constructing guidable choice functions. Lemma 28 will follow easily from the fact that connected components of graphs of bounded shrubdepth have bounded diameter by Lemma 27, and from the following lemma, essentially stating that every total binary relation whose graph has bounded shrubdepth contains a guidable choice function.

Lemma 29. *For every class $\mathscr{C}$ of graphs of 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 is a function $f\colon A \to B$ which is p-guidable as a partial function 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 paper we give only one proof, using the characterization of classes of bounded shrubdepth using connection models, and their close connection to *bi-cographs*. We present the second proof in Appendix D.2, which provides an explicit greedy procedure leading to the construction of f.

We first prove a special case of Lemma 29 for graphs which 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. [18]).

Lemma 30. *Let G be a bi-cograph with parts A, B and with a connection model of height h where vertices in A have label α and vertices in B have label β. Suppose further that every vertex in A has a neighbor in B. Then there is a function $f\colon A \to B$ which is h-guidable as a partial function on G.*

Proof. By Lemma 24, it is enough to consider the case when G is connected. Let T be the assumed connection model of height h.

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

In the inductive step, assume that $h \geqslant 2$ and the statement holds for height $h-1$. Since G is connected, either the label $C(r)$ of the root r contains the pair (α, β), or r has only one child v. In the latter case, the subtree of T rooted at v is a connection model of G of height $h-1$, so the conclusion holds by inductive assumption. Hence, we assume that $(\alpha, \beta) \in C(r)$.

Let $\mathcal{S}$ be the set of 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 H 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 $f\colon A \to B$ 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 $V(G) - V(H)$ is adjacent to w. The graph H has a connection model of height $h-1$, so by inductive assumption, there is a guidance system $\mathcal{U} \subseteq \mathscr{P}(V(H))$ of size at most $h-1$ and a function $f_0\colon V(H) \cap A \to V(H) \cap B$ which is guided by $\mathcal{U}$. Then the function $f\colon A \to B$ which extends f_0 by mapping every vertex in $V(G) - V(H)$ to w is guided by $\mathcal{U} \cup \{\{w\}\}$. In either case, we have constructed a h-guidable function $f\colon A \to B$, as required. □

We now prove Lemma 29 in the general case.

Proof (of Lemma 29). Let $\mathscr{C}$ be a class of graphs of bounded shrubdepth. Hence, there is a finite set of labels Λ and a number $h \in \mathbb{N}$ such that every graph $G \in \mathscr{C}$ has a connection model of height h using labels from Λ. For $\alpha \in \Lambda$, let V_α denote the set of vertices of G which are labeled α.

Define a function $\mu\colon A \to \Lambda^2$ as follows: for every vertex v define $\mu(v)$ as (α, β), where α is the label of v, and $\beta \in \Lambda$ is an arbitrary label such that v has a neighbor in B with label β.

For every pair of labels α, β, consider the bipartite graph $G_{\alpha\beta}$ which is the subgraph of G consisting of $\mu^{-1}((\alpha, \beta))$ on one side and $B \cap V_\beta$ on the other side, and all edges between these sets; note that they are disjoint, as one is contained in A and second in B. Observe that $G_{\alpha\beta}$ is a bi-cograph with a connection model of height h, such that every vertex in $V(G_{\alpha\beta}) \cap A$ has a neighbor in $V(G_{\alpha\beta}) \cap B$. By Lemma 30 there is a function $f_{\alpha\beta}\colon \mu^{-1}((\alpha, \beta)) \to B \cap V_\beta$ which is h-guidable in $G_{\alpha\beta}$. Observe that $f_{\alpha\beta}$ is also h-guidable when treated as a partial function on G; it suffices to take the same guidance system, but with all its sets restricted to B.

Finally, define the function $f\colon A \to B$ so that if $v \in A$ and $\mu(v) = (\alpha, \beta)$, then $f(v) = f_{\alpha\beta}(v)$. By Lemma 25, the function f is $(h \cdot |\Lambda|^2)$-guidable. This concludes the proof of Lemma 29. □

Constructing guidable spanning forests. We are ready to complete the proof of Lemma 28 stating that shallow spanning forests on classes of bounded shrubdepth are definable by guidance systems.

Proof (of Lemma 28). Let $\mathscr{C}$ be a class of graphs of bounded shrubdepth, and let r and p be constants provided by Lemma 27 and Lemma 29, respectively, for the class $\mathscr{C}$. Let $R_0 \subseteq V(G)$ be a set of vertices which contains exactly one vertex in each connected component C of G. By Lemma 27, we may assume that every vertex in G is at distance at most r from a unique vertex in R_0. For $i = 1, \ldots, r$, let R_i be the set of vertices of G whose distance to some vertex in R_0 is equal to i. Then the sets $R_0, R_1, \ldots, R_r$ form a partition of the vertex set of G. Furthermore, observe that for $i = 1, \ldots, r$, every vertex of R_i has a neighbor in R_{i-1}.

Fix a number $i \in \{1, \ldots, r\}$. Apply Lemma 29 to R_i as A and R_{i-1} as B. This yields a function $f_i\colon R_i \to R_{i-1}$ which is p-guidable in $G[R_i \cup R_{i-1}]$. In particular, f_i is also a p-guidable partial function $f_i\colon V(G) \rightharpoonup V(G)$. Let f_0 be a partial function from $V(G)$ to $V(G)$ that fixes every vertex of R_0 and is undefined otherwise. Then f_0 is guided by the guidance system $\{R_0\}$, hence it is 1-guidable in G.

Consider now the function $f_G\colon V(G) \to V(G)$ such that for $u \in V(G)$, $f_G(u) = f_i(u)$ if $f_i(u)$ is defined for some $i \in \{0, \ldots, r\}$. By the first item of Lemma 25 we find that f_G is $p(r+1)$-guidable. By construction, f_G is guarded, and f_G^r maps every vertex $v \in V(G)$ to the unique vertex in R_0 which lies in the connected component of v. This proves that f_G defines a spanning forest of depth r on G. □

This completes the proof of Lemma 23.

5.2 Proposition 18 for classes of bounded shrubdepth

In this section, we prove Proposition 18 in the special case when $\mathscr{C}$ is a class of graphs of bounded shrubdepth:

Lemma 31. *Let $\mathcal{B}$ be a class of graphs of bounded shrubdepth. Then there is a class $\mathcal{T}$ of colored trees of bounded height and a pair of transductions T and B such that T is almost quantifier-free, B is deterministic almost quantifier-free, $\mathsf{T}(\mathcal{B}) \subseteq \mathcal{T}$, $\mathsf{B}(\mathcal{T}) \subseteq \mathcal{B}$, and*

$$\mathsf{B}(\mathsf{T}(G)) = \{G\} \quad \text{for all} \quad G \in \mathcal{B} \quad \text{and} \quad \mathsf{T}(\mathsf{B}(t)) \ni t \quad \text{for all} \quad t \in \mathcal{T}.$$

Moreover, for any $G \in \mathcal{B}$, every $t \in \mathsf{T}(G)$ is an SC-decomposition of G.

We remark that in Lemma 31, every output of the transduction T is an SC-decomposition of the input graph of bounded depth, whereas the transduction B recovers the graph from its SC-decomposition.

In other words, the lemma allows to construct the SC-decomposition of a graph from a class of graphs of bounded shrubdepth using an almost quantifier-free transduction. This argument is the combinatorial cornerstone of our approach. Conceptually, it shows that bounded-height decompositions of graphs from classes of bounded shrubdepth can be defined in a very weak logic, as essentially the whole information about the decomposition can be pushed to unary predicates on vertices (added using unary lifts), and from this information the decomposition can be reconstructed using only deterministic almost quantifier-free formulas.

We need one more auxiliary lemma which allows to apply a transduction in parallel to a disjoint union of structures. Suppose $\mathcal{K}$ is a set of structures over the same signature. The *bundling* of $\mathcal{K}$ is a structure obtained by taking the disjoint union $\bigcup \mathcal{K}$ of the structures in $\mathcal{K}$, extended with a set X disjoint from $V(\bigcup \mathcal{K})$ and a function $f \colon V(\bigcup \mathcal{K}) \to X$ such that $f(x) = f(y)$ if and only if x, y belong to the same structure in $\mathcal{K}$. We denote such a bundling by $\bigcup \mathcal{K}^X$. We now prove that an almost quantifier-free transduction working on each structure separately can be lifted to their bundling.

Lemma 32 ($\star$). *Let I be an almost quantifier-free transduction. Then there is an almost quantifier-free transduction $\mathsf{I}^\star$ such that if the input to $\mathsf{I}^\star$ is the bundling $\bigcup \mathcal{K}^X$ of $\mathcal{K}$, then $\mathsf{I}^\star(\bigcup \mathcal{K}^X)$ is the set containing the bundling of every set formed by taking one member from $\mathsf{I}(\mathbf{K})$ for each $\mathbf{K} \in \mathcal{K}$.*

We can now give a proof of Lemma 31.

Proof (of Lemma 31). Let $\mathcal{B}_d$ be the class of graphs of SC-depth at most d. We prove the statement for $\mathcal{B} = \mathcal{B}_d$, yielding appropriate transductions B_d and T_d. Observe that this implies the general case: if $\mathcal{B}$ is any class of graphs of bounded shrubdepth, then by Proposition 6(2) there is a number d such that every graph from $\mathcal{B}$ has SC-depth at most d, hence we may set $\mathsf{B} = \mathsf{B}_d$, $\mathsf{T} = \mathsf{T}_d$, and $\mathcal{T} = \mathsf{T}(\mathcal{B})$.

The proof is by induction on d. The base case, when $d = 0$, is trivial. In general, every output of T_d will be an SC-decomposition of the input graph of depth d. That is, it is a tree of height d, here encoded as a structure by providing its parent function. The leaves of this tree are exactly the original vertices of the input graph G. They are colored with d unary predicates $W_0, W_1, \ldots, W_{d-1}$, corresponding to flip sets used on consecutive levels of the SC-decomposition.

Now, given an almost quantifier-free transduction T_d we construct an almost quantifier-free transduction T_{d+1}. The transduction T_{d+1}, given a graph G, nondeterministically computes a rooted tree t_G as above in the following steps. Implementing each of them using an almost quantifier-free transduction is straightforward, and to keep the description concise, we leave the implementation details to the reader.

- Since $G \in \mathcal{B}_{d+1}$, there is a vertex subset $W \subseteq V(G)$ such that in the graph G' obtained from G by flipping the adjacency within W every connected component belongs to $\mathcal{B}_d$. Using a unary lift, introduce a unary predicate W_0 selecting the set W and compute G' by flipping the adjacency within W_0.

- Let $g\colon V(G') \to V(G')$ be the function given by Lemma 23, applied to the graph G'. Note that g can be constructed using an almost quantifier-free transduction. Using copying and restriction, create a copy X of the image of g. By composing g with the function that maps each element of the image of g to its copy (easily constructible using function extension), we construct a function $g'\colon V(G') \to X$ such that $g'(v) = g'(w)$ if and only if v and w are in the same connected component of G'. Hence, $g'\colon V(G') \to X$ defines a bundling of the set of connected components of G'.

- Apply Lemma 32 to the transduction T_d yielding a transduction $\mathsf{T}_d^\star$. Our transduction T_{d+1} now applies $\mathsf{T}_d^\star$ to the bundling given by g', resulting in a bundling of the family of colored trees t_C, for C ranging over the connected components of G'.

- Using extension, mark the roots of the trees t_C with a new unary predicate; for C ranging over the connected components of G' these are exactly elements that do not have a parent. Create new edges which join each such a root r with $g'(r)$. In effect, for every connected component C of G', all the roots of the trees t_C are appended to a new root r_C. At the end clear all unnecessary relations from the structure. Note that the obtained tree t_G retains all unary predicates $W_1, \ldots, W_d$ that were introduced by the application of the transduction $\mathsf{T}_d^\star$ to G', as well as the predicate W_0 introduced at the very beginning. All these predicates select subsets of leaves of t_G.

This concludes the description of the almost quantifier-free transduction T_{d+1}. The transduction B_{d+1} is defined similarly, and reconstructs G out of t_G recursively as follows:

- Let r be the root of t_G; it can be identified as the only vertex that does not have a parent. Remove r from the structure, thus turning t_G into a forest t'_G, where the roots of t'_G are children of r in t_G.

- Using function extension, add a function f which maps every vertex v to its unique root ancestor in t'_G. This can be done by taking f to be the d-fold composition of the parent function of t'_G with itself (assuming each root points to itself, which can be easily interpreted).

- Copy all the roots of trees in t'_G and let X be the set of those copies. Construct a function $f'\colon V(t'_G) \to X$ that maps each vertex v to the copy of $f(v)$. Observe that f' defines a bundling of the trees of t'_G.

- Apply the transduction $\mathsf{B}^\star_d$ obtained from Lemma 32 to the above bundling. This yields a bundling of the family of connected components of G', where G' is obtained from G by flipping the adjacency within W_0.

- Forgetting all elements of the structure apart from the bundled connected components of G' yields the graph G'. Construct the graph G by flipping the adjacency inside the set W_0. Note here that since the remaining vertices are exactly the leaves of the original tree t_G, the predicate W_0 is still carried by them. Finally, clean the structure from all unnecessary predicates.

It is straightforward to see that transductions T_d and B_d satisfy all the requested properties. This concludes the proof of Lemma 31. □

5.3 Proposition 18 for classes of with low shrubdepth covers

We now prove Proposition 18 in the general case. As noted earlier, this will finish the proof of the right-to-left implication in Theorem 15.

Proof (of Proposition 18). Let $\mathscr{C}$ be a class of graphs with low shrubdepth covers. We fix a finite 2-cover $\mathcal{U}$ of $\mathscr{C}$ such that $\mathscr{C}[\mathcal{U}]$ has bounded shrubdepth. Let $N = \sup\{|\mathcal{U}_G|\colon G \in \mathscr{C}\}$, and for $G \in \mathscr{C}$ let $\widehat{G}$ be the extension of G by unary predicates $U_1, \ldots, U_N$ such that $\{U_1, \ldots, U_N\} = \mathcal{U}_G$. Let $\widehat{\mathscr{C}} = \{\widehat{G}\colon G \in \mathscr{C}\}$. Then the class $\mathcal{B} = \widehat{\mathscr{C}}[\mathcal{U}]$ has bounded shrubdepth.

Apply Lemma 31 to the class $\mathscr{C}[\mathcal{U}]$, yielding almost quantifier-free transductions T and B. It is easy to construct an almost-quantifier free transduction S' such that for $G \in \mathscr{C}$, the structure $\mathsf{S}'(\widehat{G})$ is the union of the trees $T_U \in \mathsf{T}(G[U])$, one tree per each $U \in \mathcal{U}_G$, where the union is disjoint apart from the vertices which belong to $V(G)$ (the leaves of the trees). Indeed, we process $U_1, \ldots, U_N$ in order, and for each consecutive U_i we apply the transduction T to $G[U_i]$, appropriately modifying all its atomic operations so that the elements outside of U_i are ignored and kept intact. Recall all the constructed trees have depth bounded by a constant, say d.

Now obtain S from S' by precomposing with a sequence of unary lifts introducing the predicates $U_1, \ldots, U_N$, and appending the following operations. First, using extension operations introduce unary predicates $D_{i,\ell}$ for $i \in \{1, \ldots, N\}$ and $\ell \in \{0, 1, \ldots, d\}$ such that $D_{i,\ell}$ selects nodes at depth ℓ in the tree T_{U_i}. Next, using an extension operation that introduces an adjacency relation binding every pair of elements u, v such that $f(u) = v$ for some function f in the signature (the parent functions). Finally, use a sequence of reduct operations which drop all functions and non-unary relations from the signature, apart from adjacency. Thus every output of S is a colored graph.

Let $\mathscr{F} = \mathsf{S}(\mathscr{C})$. By Lemma 17, $\mathscr{F}$ has low shrubdepth covers. Furthermore, each graph $H \in \mathsf{S}(G)$ for some $G \in \mathscr{C}$ is the union of at most N trees, hence H is N-degenerate and in particular excludes the biclique $K_{N+1,N+1}$ as a subgraph.

Hence by Lemma 7 we infer that $\mathsf{S}(\mathscr{C})$ has low treedepth covers, so by Lemma 13, $\mathsf{S}(\mathscr{C})$ is a class of bounded expansion.

We are left with constructing a deterministic almost quantifier-free transduction I satisfying $\mathsf{I}(\mathsf{S}(G)) = \{G\}$. This transduction should take on input a graph $H \in \mathsf{S}(G)$ and turn it back to G. The vertex set of H consists of $V(G)$ and trees T_U for $U \in \mathcal{U}_G$, each built on top of the subset U of $V(G)$ and of depth at most d. Using predicates $D_{i,\ell}$ it is easy to use a sequence of quantifier-free function extension operations to construct, for each $U \in \mathcal{U}_G$, the parent function of T_U, thus turning the substructure induced by the nodes of T_U back into T_U. Similarly as before, it is now straightforward to construct a transduction I' that applies the transduction B to each colored tree T_U, thus turning the set of its leaves into $G[U]$. Since $\mathcal{U}$ was a 2-cover, for every edge e of G there exists $U \in \mathcal{U}_G$ that contains both endpoints of e. Hence, applying I' to the current structure recovers the graph G; this concludes the construction of I. Note that I is deterministic almost quantifier-free. □

6 Proof of Proposition 19 *(quantifier elimination for classes of bounded expansion)*

In this section we prove Proposition 19, which we repeat for convenience.

Proposition 19. *Let $\mathscr{C}$ be a class of graphs of bounded expansion and let I be a transduction. Then I is equivalent to an almost quantifier-free transduction J on $\mathscr{C}$.*

We note that Proposition 19 is a strengthening of similar statements provided by Dvořák et al. [9] and of Grohe and Kreutzer [21], and could be derived by a careful analysis of their proofs, and by using the Lemma 33 below.

For a graph G and a partial function $f\colon V(G) \rightharpoonup V(G)$, we say that f is *guarded* by G if for every vertex in the domain of f is mapped to itself or to its neighbor.

Lemma 33 ($\star$). *Let $\mathscr{C}$ be a class of graphs which has 2-covers of bounded treedepth, and for each $G \in \mathscr{C}$, let $\widehat{G}$ be the graph G extended by a partial function $f\colon V(G) \rightharpoonup V(G)$ which is guarded by G. Then there is an almost quantifier-free transduction F using only unary lifts and a single function extension such that $\mathsf{F}(G) = \widehat{G}$.*

To derive Proposition 19 from [9], one would need to prove that the unary functions constructed in their proofs can be obtained as compositions of guarded functions, and conclude using Lemma 33. Rather then doing that, below we provide a self-contained proof of Proposition 19, which we also believe is simpler than the existing proofs, among other reasons, thanks to the notion of covers. In Section 6.1 we outline how the result of Dvořák, Král', and Thomas can be deduced from our proof.

We will use the following restricted form of transductions. A *faithful transduction* is a transduction which does not use copying and restrictions. A *guarded transduction* is a faithful transduction which given a structure $\mathbf{A}$, produces a structure whose Gaifman graph is a subgraph of the Gaifman graph of $\mathbf{A}$. In the following lemmas, we identify a first-order formula $\varphi(\bar{x})$ with the transduction which inputs a structure $\mathbf{A}$ and outputs $\mathbf{A}$

extended with a single relation, consisting of those tuples $\bar{a}$ which satisfy $\varphi(\bar{x})$ in $\mathbf{A}$ (this transduction is a composition of an extension operation followed by a sequence of reduct operations which drop all the symbols from the input structure).

Lemma 34. *Let $\varphi(\bar{x})$ be a first-order formula and let $\mathscr{C}$ be a class of graphs of bounded expansion. Then there is a guarded transduction I which adds unary function and relation symbols only, and a quantifier-free formula $\varphi'(\bar{x})$, such that φ is equivalent to $\mathsf{I};\varphi'$ on $\mathscr{C}$.*

Before proving Lemma 34, we first show how to conclude Proposition 19 using it.

Proof (of Proposition 19). For simplicity we assume that the signature produced by I consists of one relation P; lifting the proof to signatures containing more relation and function symbols is immediate. By Lemma 2, we may express I as

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

where

- L is a sequence of unary lifts,
- C is a sequence of copying operations,
- E is a single extension operation introducing the final relation P using some formula $\varphi(\bar{x})$,
- X is a single universe restriction operation using some formula $\psi(x)$ that does not use symbol P, and
- R is a sequence of reduct operations that drop all relations and functions apart from P.

From Lemma 8 it follows that the class $\mathsf{C}(\mathsf{L}(\mathscr{C}))$ of colored graphs is a class of bounded expansion, and therefore, we may apply Lemma 34 to it, and to the formulas $\varphi(\bar{x})$ and $\psi(x)$ considered above.

Using Lemma 34 we replace the formulas $\varphi(\bar{x})$ and $\psi(x)$ by quantifier-free formulas, at the cost of introducing additional guarded transductions which introduce unary function and relation symbols. Using Lemma 33, every such transduction is equivalent to an almost quantifier-free transduction. Hence, the transductions E and X can be replaced in I by almost quantifier-free transductions, yielding an almost quantifier-free transduction J that is equivalent to I on $\mathscr{C}$. □

As explained, Proposition 19 together with Proposition 18 yields Theorem 16. It remains to prove Lemma 34. Similarly as in [9, 21], we first prove the statement for classes of colored forests of bounded depth:

Lemma 35 (⋆). *Let $\varphi(\bar{x})$ be a first-order formula and let $\mathscr{F}$ be a class of colored rooted forests of bounded depth. Then there is a transduction I_φ which, given a rooted forest $F \in \mathscr{F}$ extends it by the parent function of F and some unary predicates, and there exists a quantifier-free formula $\varphi'(\bar{x})$ such that φ is equivalent to $\mathsf{I}_\varphi;\varphi'$ on $\mathscr{F}$.*

Let us remark that the presented proof of Lemma 35 is based on the automata approach and is conceptually different from the ones used in [9,21]. Note that the transduction I_φ produced in Lemma 35 is in particular a guarded transduction, since the parent of a vertex in a forest is in particular a neighbor of that vertex.

The next step is to lift Lemma 35 to classes of structures of bounded treedepth. We first observe that classes of bounded treedepth are bi-definable with classes of forests of bounded depth, using almost quantifier-free transductions. This result is similar, but much simpler to prove than Lemma 31, which is an analogous statement for classes of bounded shrubdepth.

Lemma 36. *Let $\mathscr{C}$ be a class of structures of bounded treedepth. There is a pair of faithful transductions T and C and a class $\mathscr{F}$ of colored rooted forests of bounded depth such that $\mathsf{T}(\mathscr{C}) \subseteq \mathscr{F}$, $\mathsf{C}(\mathscr{F}) \subseteq \mathscr{C}$ and $\mathsf{C}(\mathsf{T}(\mathbf{A})) = \{\mathbf{A}\}$ for $\mathbf{A} \in \mathscr{C}$. Moreover, the transduction T is guarded, and C is deterministic almost quantifier-free.*

Proof. We follow the well-known encoding of structures of 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, as follows.

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.

It is known that a graph G of treedepth at most d has a DFS forest of depth at most 2^d. If $\mathbf{A}$ is a structure over a fixed signature Σ, G is its Gaifman graph and F is a DFS forest of G of depth 2^d, then $\mathbf{A}$ can be encoded in F using a bounded number of additional unary predicates by labeling every node v of F by the isomorphism type of the substructure of $\mathbf{A}$ induced by $v_1, \ldots, v_t$, where $v_1, \ldots, v_t$ are the nodes on the path from a root of F to v, $v = v_t$ and $t \leqslant 2^d$. The number of used unary predicates depends only on the signature Σ and d.

If $\mathscr{C}$ be a class of structures of treedepth at most d, then the transduction T, given a structure $\mathbf{A} \in \mathscr{C}$ outputs a DFS forest F of the Gaifman graph of $\mathbf{A}$ of depth at most 2^d, extended with unary predicates encoding $\mathbf{A}$, as described above. The structure $\mathbf{A}$ can be recovered from F (together with the unary predicates) using a deterministic almost quantifier-free transduction, which first introduces the parent function, and then uses a quantifier-free formula to determine the quantifier-free type of a tuple of vertices. □

Using Lemma 36 we easily lift the quantifier-elimination result from forests of bounded depth to classes of low treedepth.

Lemma 37. *Let $\varphi(\bar{x})$ be a first-order formula and let $\mathscr{C}$ be a class of structures of bounded treedepth. Then there is a guarded transduction I_φ and a quantifier-free formula $\varphi'(\bar{x})$ such that φ is equivalent to $\mathsf{I}_\varphi; \varphi'$ on $\mathscr{C}$.*

Proof. Let C, T and $\mathscr{F}$ be as in Lemma 36. Since $\mathsf{C}(\mathsf{T}(\mathbf{A})) = \{\mathbf{A}\}$ and C is deterministic, there is a formula $\psi(\bar{x})$ such that φ is equivalent to $\mathsf{T}; \psi$ on $\mathscr{C}$. Now, apply Lemma 35 to the class $\mathscr{F}$ and the formula $\psi(\bar{x})$, yielding a guarded transduction J and a quantifier-free formula $\psi'(\bar{x})$, such that ψ is equivalent to $\mathsf{J}; \psi'$ on $\mathscr{F}$. By composition, φ is equivalent to $\mathsf{T}; \mathsf{J}; \psi'$ on $\mathscr{C}$. Note that $\mathsf{T}; \mathsf{J}$ is a guarded transduction, since T and J are such. This proves the lemma. □

Finally, we lift the quantifier elimination procedure to classes with low shrubdepth covers using Lemma 20 and a reasoning very similar to the proof of Lemma 17. Again, conceptually this lift is exactly what is happening in [9, 21], however, our approach based on covers makes it quite straightforward. The key observation is encapsulated in the following lemma.

Lemma 38. *Let $\mathscr{D}$ be a class of structures with unary relation and function symbols only, and let $\varphi(\bar{x})$ be a quantifier-free formula with p free variables, involving c distinct terms. Then there is a quantifier-free formula $\varphi'(\bar{x})$ such that following conditions are equivalent for a structure $\mathbf{A} \in \mathscr{D}$, a $c \cdot p$-cover $\mathcal{U}_{\mathbf{A}}$ of the Gaifman graph of $\mathbf{A}$, and a p-tuple $\bar{a}$ of elements of $\mathbf{A}$:*

1. $\mathbf{A}, \bar{a} \models \varphi(\bar{x})$,
2. *there is some $U \in \mathcal{U}_G$ containing $\bar{a}$ such that $\mathbf{A}[U], \bar{a} \models \varphi'(\bar{x})$.*

Proof. We first consider the special case when $\varphi(\bar{x})$ is an atomic formula. Each term t occurring in $\varphi(\bar{x})$ defines a partial function $t_{\mathbf{A}} \colon V(\mathbf{A}) \rightharpoonup V(\mathbf{A})$ on a given structure $\mathbf{A}$, in the natural way. Let $\mathcal{T}$ denote the set of terms occurring in $\varphi(\bar{x})$. By assumption, $|\mathcal{T}| \leqslant c$. For a tuple $\bar{a} = (a_1, \ldots, a_p)$ of elements of a structure $\mathbf{A}$, denote by $\mathcal{T}_{\mathbf{A}}(\bar{a})$ the set $\{t_{\mathbf{A}}(a_i) \ : \ t \in \mathcal{T}, 1 \leqslant i \leqslant p\}$. Then $|\mathcal{T}_{\mathbf{A}}(\bar{a})| \leqslant c \cdot p$.

Since $\varphi(\bar{x})$ is an atomic formula, for any p-tuple $\bar{a}$ of elements of $\mathbf{A}$ and any set $U \subseteq V(\mathbf{A})$ containing $\mathcal{T}_{\mathbf{A}}(\bar{a})$ we have the following equivalence:

$$\mathbf{A}, \bar{a} \models \varphi(\bar{x}) \iff \mathbf{A}[U], \bar{a} \models \varphi(\bar{x}).$$

Take $\varphi'(\bar{x}) = \varphi(\bar{x})$. The equivalence of the two items then follows by assumption that $\mathcal{U}_G$ is a $p \cdot c$-cover of $\mathbf{A}$, so for every $\bar{a}$, there is some set $U \in \mathcal{U}_G$ containing $\mathcal{T}_{\mathbf{A}}(\bar{a})$.

To treat the general case of a quantifier-free formula, we take $\varphi'(\bar{x})$ to be a conjunction of $\varphi(\bar{x})$ and a formula which verifies that all the values in $\mathcal{T}_{\mathbf{A}}(\bar{a})$ are defined. We leave the details to the reader. □

We are ready to prove Lemma 34.

Proof (of Lemma 34). 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 I to be the identity transduction.

In the inductive step, we consider two cases. If $\varphi(\bar{x})$ is a boolean combination of simpler formulas, then the statement follows immediately from the inductive assumption. The interesting case is when $\varphi(\bar{x})$ is of the form $\exists y.\psi(\bar{x}, y)$, for some formula $\psi(\bar{x}, y)$. We consider this case below. Denote by p the number of free variables in the formula $\psi(\bar{x}, y)$.

Apply the inductive assumption to the formula $\psi(\bar{x}, y)$, yielding a guarded transduction I_ψ and a formula $\psi'(\bar{x}, y)$. Let c be the number of distinct terms (including subterms) appearing in the formula $\psi'(\bar{x}, y)$. Let $\mathscr{D} = \mathsf{I}_\psi(\mathscr{C})$. Note that every structure in $\mathscr{D}$ has unary function and relation symbols only, and is guarded by some graph in $\mathscr{C}$. By Lemma 17, we can pick a finite $c \cdot p$-cover $\mathcal{U}$ of $\mathscr{C}$, so that the class $\mathscr{C}[\mathcal{U}]$ has bounded treedepth. As I_ψ is guarded, it follows that also the class $\mathscr{D}[\mathcal{U}]$ has bounded treedepth.

Apply Lemma 38 to $\mathscr{D}$ and $\psi'(\bar{x}, y)$, yielding a formula $\psi''(\bar{x}, y)$ such that for every graph $G \in \mathscr{C}$, p-tuple of vertices $(\bar{a}, b)$ and the $c \cdot p$-cover $\mathcal{U}_G$ of $G \in \mathscr{C}$, the following equivalences hold:

$$\begin{aligned} G, \bar{a}, b \models \psi(\bar{x}, y) &\iff \mathsf{I}_\psi(G), \bar{a}, b \models \psi'(\bar{x}, y) \\ &\iff \mathsf{I}_\psi(G)[U], \bar{a}, b \models \psi''(\bar{x}, y) \textit{ for some } U \in \mathcal{U}_G \textit{ containing } \bar{a}, b. \end{aligned}$$

Apply Lemma 37 to the class $\mathscr{D}[\mathcal{U}]$ and the formula $\exists y.\psi''(\bar{x}, y)$, yielding a guarded transduction F and quantifier-free formula $\rho(\bar{x})$ such that for every $\mathbf{A} \in \mathscr{D}[\mathcal{U}]$ and tuple $\bar{a} \in V(\mathbf{A})^{|\bar{x}|}$,

$$\mathbf{A}, \bar{a} \models \exists y.\psi''(\bar{x}, y) \iff \mathsf{F}(\mathbf{A}), \bar{a} \models \rho(\bar{x}).$$

Claim 1. *For each graph $G \in \mathscr{C}$ and tuple $\bar{a} \in V(H)^{|\bar{x}|}$, the following conditions are equivalent:*

1. $G, \bar{a} \models \exists y.\psi(\bar{x}, y)$,
2. *there is some $U \in \mathcal{U}_G$ containing $\bar{a}$ such that $\mathsf{F}(\mathsf{I}_\psi(G)[U]), \bar{a} \models \rho(\bar{x})$.*

Proof. We have the following equivalences:

$$\begin{aligned} G, \bar{a} \models \exists y.\psi(\bar{x}, y) &\iff G, \bar{a}, b \models \psi(\bar{x}, y) \textit{ for some } b \in V(G) \\ &\iff \mathsf{I}_\psi(G)[U], \bar{a}, b \models \psi''(\bar{x}, y) \textit{ for some } U \in \mathcal{U}_G \textit{ containing } \bar{a}, b \\ &\iff \mathsf{I}_\psi(G)[U], \bar{a} \models \exists y.\psi''(\bar{x}, y) \textit{ for some } U \in \mathcal{U}_G \textit{ containing } \bar{a} \\ &\iff \mathsf{F}(\mathsf{I}_\psi(G)[U]), \bar{a} \models \rho(\bar{x}) \textit{ for some } U \in \mathcal{U}_G \textit{ containing } \bar{a}. \end{aligned}$$

This proves the claim. ⌟

Let $N = \sup\{|\mathcal{U}_G| \colon G \in \mathscr{C}\}$. For each graph $G \in \mathscr{C}$, fix an enumeration $U_1, \ldots, U_N$ of the cover $\mathcal{U}_G$.

Claim 2. *There is a guarded transduction F' and quantifier-free formulas $\rho_1(\bar{x}), \ldots, \rho_N(\bar{x})$ such that given a graph $G \in \mathscr{C}$, a number $i \in \{1, \ldots, N\}$ and a tuple $\bar{a}$ of elements of U_i,*

$$\mathsf{F}'(G), \bar{a} \models \rho_i(\bar{x}) \iff \mathsf{F}(\mathsf{I}_\psi(G)[U_i]), \bar{a} \models \rho(\bar{x}).$$

Proof. We construct a guarded transduction F' which, given a graph $G \in \mathscr{C}$, first applies the guarded transduction I_ψ, then introduces unary predicates marking the sets $U_1, \ldots, U_N$, and then, for each such unary predicate U_i, applies to the structure $\mathsf{I}_\psi(G)[U_i]$ 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. ⌟

Combining Claim 1 and Claim 2 we get the following equivalence:

$$\mathsf{F}'(G), \bar{a} \models \bigvee_{i=1}^{N} \rho_i(\bar{x}) \iff G, \bar{a} \models \varphi(\bar{x}),$$

concluding the inductive step. This finishes the proofs of Lemma 34 and Proposition 19. □

6.1 Effectivity

As a side remark, we note that we can easily derive the result of Dvořák, Král', and Thomas, by observing that the above proof of Lemma 34 is effective, and can be leveraged to construct a transduction I which is a linear time computable function.

We say that a transduction I is a *linear time* transduction if there is an algorithm which, given a structure $\mathbf{A}$ as input, produces some structure $\mathbf{B} \in \mathsf{I}(\mathbf{A})$ 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 34.

Lemma 39. *Let $\varphi(\bar{x})$ be a first-order formula and let $\mathscr{C}$ be a class of graphs of bounded expansion. Then there is a guarded transduction I which adds unary function and relation symbols only, and a quantifier-free formula $\varphi'(\bar{x})$, such that φ is equivalent to $\mathsf{I};\varphi'$ on $\mathscr{C}$. Moreover, I is a linear time transduction.*

Proof. To prove Lemma 39, we observe that the transduction I in Lemma 34 is a linear time transduction. The proof follows by tracing the proof of Lemma 34, and observing the following.

1. In Lemma 35, the constructed transduction I is a linear time transduction. This is because the transduction only adds the parent function (which is clearly linear-time computable, given a rooted forest) and some unary predicates, each of which can be computed in linear time, since each unary predicate is produced by running a deterministic threshold tree automaton on the input tree.

2. In Lemma 36, the transduction T is a linear time transduction, since it amounts to running a depth-first search on the input graph.

3. In Lemma 37, the produced transduction $\mathsf{J} = \mathsf{T};\mathsf{J}$ is a linear time transduction, as a composition of two linear time transductions.

4. In the proof of Lemma 34, the nontrivial step is in the inductive step, in the case of an existential formula. In this case, the constructed transduction F' is a linear time transduction, assuming $\mathscr{C}$ has bounded expansion, as F' amounts to introducing unary predicates denoting the elements of a cover $\mathcal{U}_G$, and applying transductions I_ψ and F which are linear time transductions, respectively, by the inductive assumption, and by the effective version of Lemma 37 discussed above.

   We note that if $\mathscr{C}$ has bounded expansion then for any fixed $p \geqslant 0$ there is a finite p-cover $\mathcal{U}$ of $\mathscr{C}$ of 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 $g(p)$-treepdepth coloring of G for some function g (as required in the proof of Lemma 13) and observe that it can be converted to a cover in linear time, as in the proof of Lemma 13. A p-treedepth coloring can be computed in linear time, cf. [8, 30, 31]. □

7 Algorithmic aspects

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 an algorithm that, given a graph $G \in \mathscr{C}$, in linear time computes the cover $\mathcal{U}_G$ and for each $U \in \mathcal{U}_G$, an SC-decomposition S_U of depth at most d of the graph $G[U]$, for some constant d depending only on $\mathscr{C}$. Our result is as follows.

Theorem 40. *Suppose* J *is a deterministic transduction and* $\mathscr{C}$ *is a class of graphs that has structurally bounded expansion and is efficiently decomposable. Then given a graph* $G \in \mathscr{C}$*, one may compute* $\mathsf{J}(G)$ *in time linear in the size of the input plus the size of the output.*

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. If only the cover is given but not the SC-decompositions, we would obtain cubic running time because bounded shrubdepth implies bounded cliquewidth and we can compute an approximate clique decomposition in cubic time [32]. Then, SC-decompositions of small height are definable in monadic second-order logic, and hence they can be computed in linear time using the result of Courcelle, Makowski and Rotics [3].

Observe that the theorem implies that we can efficiently evaluate a first-order sentence and enumerate all tuples satisfying a formula $\varphi(x_1, \ldots, x_k)$ on the given input graph, since this amounts to applying the theorem to a transduction consisting of a single extension operation. This strengthens the analogous result of Kazana and Segoufin [25] for classes of bounded expansion.

Proof (sketch). We will make use of transductions S and I constructed in the proof of Proposition 18. Recall that $\mathsf{S}(\mathscr{C})$ is a class of colored graphs of bounded expansion, I is deterministic, and $\mathsf{I}(\mathsf{S}(G)) = \{G\}$ for each $G \in \mathscr{C}$. Observe that J is equivalent to $\mathsf{S}; \mathsf{I}; \mathsf{J}$ on $\mathscr{C}$. Defining K as $\mathsf{I}; \mathsf{J}$, we get that $\mathsf{J}(G) = \mathsf{K}(\mathsf{S}(G))$ for $G \in \mathscr{C}$. Moreover, since I is deterministic, it follows that K is deterministic.

Let $G \in \mathscr{C}$ be an input graph. By efficient decomposability of $\mathscr{C}$, in linear time we can compute a cover $\mathcal{U}_G$ of G together with an SC-decomposition S_U of depth at most d of $G[U]$, for $U \in \mathcal{U}_G$. Each S_U is a colored tree, and by the construction described in the proof of Proposition 18, the trees S_U for $U \in \mathcal{U}_G$, glued along the leaves form a structure belonging to $\mathsf{S}(G)$. As $\mathsf{J}(G) = \mathsf{K}(\mathsf{S}(G))$, it suffices to apply the enumeration result of Kazana and Segoufin for classes of bounded expansion [25] to the colored graph $\mathsf{S}(G)$ and to all formulas occurring in the transduction K. $\square$

8 Conclusion

In this paper we have provided a natural combinatorial characterization of graph classes that are first-order transductions of bounded expansion classes of graphs. Our characterization parallels the known characterization of bounded expansion classes by the existence of low treedepth decompositions, 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 of bounded expansion graph classes are just the same as transductions of classes of structures of bounded expansion (i.e., classes whose Gaifman graphs or whose incidence encodings have bounded expansion). On the other hand, it remains an open question to characterize classes of relational structures, rather than just graphs, which are transductions of bounded expansion classes. We are lacking the analogue of Lemma 31; 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.

Finally, observe that classes of 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 fact that we do not know whether there exists a robust quantifier-elimination procedure for these graph classes.

References

[1] B. Courcelle. Graph rewriting: an algebraic and logic approach. In *Handbook of Theoretical Computer Science*, volume 2, chapter 5, pages 142–193. Elsevier, Amsterdam, 1990.

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

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

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

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

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

[7] A. Durand, N. Schweikardt, and L. Segoufin. Enumerating answers to first-order queries over databases of low degree. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 121–131. ACM, 2014.

[8] Z. Dvořák, D. Kráľ, and R. Thomas. Deciding first-order properties for sparse graphs. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 133–142, 2010.

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

[10] K. Eickmeyer and K. Kawarabayashi. FO model checking on map graphs. In *Proceedings of the 21st International Symposium on Fundamentals of Computation Theory, FCT 2017*, pages 204–216, 2017.

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

[12] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM*, 48:1148–1206, 2001.

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

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

[15] R. Ganian, P. Hliněný, J. Obdržálek, J. Schwartz, J. Teska, and D. Kráľ. Fo model checking of interval graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 250–262. Springer, 2013.

[16] R. Ganian, P. Hliněný, J. Nešetřil, J. Obdržálek, and P. Ossona de Mendez. When trees grow low: Shrub-depth and m-partite cographs. 2013. submitted.

[17] R. Ganian, P. Hliněný, J. Nešetřil, J. Obdržálek, P. Ossona de Mendez, and R. Ramadurai. When trees grow low: Shrubs and fast MSO_1. In *MFCS 2012*, volume 7464 of *Lecture Notes in Computer Science*, pages 419–430. Springer-Verlag, 2012.

[18] V. Giakoumakis and J.-M. Vanherpe. Bi-complement reducible graphs. *Adv. Appl. Math.*, 18:389–402, 1997.

[19] M. Grohe. Generalized model-checking problems for first-order logic. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 12–26. Springer, 2001.

[20] M. Grohe and S. Kreutzer. Methods for algorithmic meta-theorems. *Contemporary Mathematics*, 588, American Mathematical Society 2011.

[21] M. Grohe and S. Kreutzer. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics*, Contemporary mathematics, pages 181–206, 2011.

[22] M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM*, 64(3):17:1–17:32, 2017.

[23] F. Gurski and E. Wanke. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *WG 2000*, volume 1928 of *Lecture Notes in Computer Science*, pages 196–205. Springer, 2000.

[24] P. Hlinený, F. Pokrývka, and B. Roy. FO model checking of geometric graphs. In *12th International Symposium on Parameterized and Exact Computation, IPEC 2017*, pages 19:1–19:12, 2017.

[25] W. Kazana and L. Segoufin. Enumeration of first-order queries on classes of structures with bounded expansion. In *Proceedings of the 16th International Conference on Database Theory*, pages 10–20, 2013.

[26] W. Kazana and L. Segoufin. Enumeration of first-order queries on classes of structures with bounded expansion. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI symposium on Principles of database systems*, pages 297–308. ACM, 2013.

[27] O. Kwon, M. Pilipczuk, and S. Siebertz. On low rank-width colorings. In *43rd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2017*, pages 372–385, 2017.

[28] C. Löding. Basics on tree automata. In *Modern Applications of Automata Theory*, pages 79–110. 2012.

[29] J. Nešetřil and P. Ossona de Mendez. Grad and classes with bounded expansion I. decompositions. *European Journal of Combinatorics*, 29(3):760–776, 2008.

[30] J. Nešetřil and P. Ossona de Mendez. Grad and classes with bounded expansion II. algorithmic aspects. *European Journal of Combinatorics*, 29(3):777–791, 2008.

[31] J. Nešetřil and P. Ossona de Mendez. *Sparsity (Graphs, Structures, and Algorithms)*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012. 465 pages.

[32] S. Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms (TALG)*, 5(1):10, 2008.

[33] D. Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 5:505–526, 1996.

[34] L. Segoufin and W. Kazana. First-order query evaluation on structures of bounded degree. *Logical Methods in Computer Science*, 7, 2011.

A Normalization lemmas for transductions

In this section we give proofs omitted from Section 2.1.

Proof (of Lemma 2 and of Lemma 3). We give appropriate swapping rules that allow us to arrange the atomic operations comprising I into the desired normal form.

We start with putting all the unary lifts at the front of the sequence. Observe that whenever an atomic operation is followed by a unary lift, then these two operations may be appropriately swapped. This is straightforward for all atomic operations apart from copying. For this last case, observe that copying followed by a unary lift introducing a unary predicate X is equivalent to a transduction that does the following. First, using unary lifts introduce two auxiliary unary predicates X_1 and X_2, interpreted to select vertices that are supposed to be selected by X in the original universe, respectively in the copy of the universe. Then perform copying. Finally, use extension and reduct operations to appropriately interpret X and drop predicates X_1, X_2.

Having applied the above swapping rules exhaustively, the formula is rewritten into the form $\mathsf{L}; \mathsf{I}'$ where I' does not contain any lifts. Observe that if I was almost quantifier-free, then I' is deterministic almost quantifier-free. This proves Lemma 3.

Next, we perform swapping within I' so that all copying operations are put at the front of the sequence of atomic operations. Again, 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 extensions and restrictions one should modify the formula parameterizing the operation in a straightforward way to work on each copy separately. Thus we have rewritten I into the form $\mathsf{L}; \mathsf{C}; \mathsf{I}''$ where I'' does not use lifts or copying.

Now consider I''. It is clear that all reduct operations can be moved to the end of the transduction, since it does not harm to have more relations in the structure. Next, we move all restriction operations to the end (before reduct operations) by showing that each restriction operation can be swapped with any extension or function extension operation. Suppose that the restriction is parameterized by a unary formula ψ, and it is followed by an extension operation (normal 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 run 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 extension and function extension 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 extension and function extension operations within I'''. Whenever a formula φ parameterizing some extension or function extension operation

within I''' uses a relation symbol R introduced by some earlier extension 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 extension operation within I''', say using formula $\varphi'(x, y)$, then replace each usage of f in φ by appropriatiely quantifying the image using formula $\varphi'(x, y)$. Perform the same operations on the formula parameterizing 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 $\mathsf{L}; \mathsf{C}$. Hence, all extension and function extension operations within I''' which 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 extensions, then (normal) extensions. □

B Proof of Lemma 7

In this section we prove Lemma 7. One implication is easy: it is known [17] that every class of bounded treedepth also has bounded shrubdepth, and moreover the bi-clique $K_{s,s}$ has treedepth $s + 1$, so every class of bounded treedepth excludes some bi-clique.

We need to prove the reverse implication: any class of bounded shrubdepth that moreover excludes some bi-clique has bounded treedepth. We will use the following well-known characterization of classes of bounded treedepth (see [31, Theorem 13.3]).

Lemma B.41. *A class of graphs $\mathscr{C}$ has bounded treedepth if and only if there exists a number $d \in \mathbb{N}$ such that no graph from $\mathscr{C}$ contains a path on more than d vertices as a subgraph.*

By Lemma B.41 and Proposition 6(3), to prove Lemma 7 it is sufficient to prove the following.

Lemma B.42. *There exists a function $g\colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that the following holds. For all integers $h, m, s \in \mathbb{N}$, if a graph G does not contain the bi-clique $K_{s,s}$ as a subgraph and admits a connection model of height at most h using at most m labels, then G does not contain any path on more than $g(h, m, s)$ vertices as a subgraph.*

Proof. We proceed by induction on the height h. For $h = 0$, only one-vertex graphs admit a connection model of height 0, so we may set $g(0, m, s) = 1$.

For the induction step, suppose G does not contain $K_{s,s}$ as a subgraph and admits a connection model T of height $h \geqslant 1$ and using m labels. Call two vertices u and v of G *related* if they are contained in the same subtree of T rooted at a child of the root of G, and *unrelated* otherwise. Whenever u and v are unrelated, their least common ancestor is the root of T, so whether they are adjacent depends solely on the pair of their labels.

Let $P = (v_1, \ldots, v_p)$ be a path in G. A *block* on P is a maximal contiguous subpath of P consisting of vertices that are pairwise related. Thus, P breaks into blocks

$B_1, \ldots, B_q$, appearing on P in this order. Note that each block B_i is a path that is completely contained in an induced subgraph of G that admits a connection model of height $h-1$ and using m labels. Hence, by the induction hypothesis we have that each block B_i has at most $g(h-1, m, s)$ vertices.

For a non-last block B_i (i.e. $i \leqslant q$), define the *signature* of B_i as the pair of labels of the following two vertices: the last vertex of B_i and of its successor on P, that is, the first vertex of B_{i+1}. The following claim is the key point of the proof.

Claim 3. *For any signature, the number of non-last blocks with this signature is at most* $4(s-1)$.

Proof. Let $\sigma = (\lambda_1, \lambda_2)$ be the signature in question and let $\mathcal{B}$ be the set of blocks with signature σ; suppose for the sake of contradiction that $|\mathcal{B}| > 4(s-1)$. Consider the following random experiment: independently color each subtree of T rooted at a child of the root black or white, each with probability $1/2$. Call a block $B_i \in \mathcal{B}$ *split* if the last vertex of B_i is white and the first vertex of B_{i+1} is black. Since these two vertices are unrelated (by the maximality of B_i), each block B_i is split with probability $1/4$, implying that the expected number of split blocks is $|\mathcal{B}|/4 > s-1$. Hence, some run of the experiment yields a white/black coloring of subtrees rooted at children of the root of T and a set $\mathcal{S} \subseteq \mathcal{B}$ of s blocks that are split in this coloring.

Let $u_1, \ldots, u_s$ be the last vertices of blocks from $\mathcal{S}$ and $v_1, \ldots, v_s$ be their successors on the path P, respectively. By assumption, all vertices u_i have label λ_1 and all vertices v_i have label λ_2. Further, all vertices u_i are white and all vertices v_i are black, implying that u_i and v_j are unrelated for all $i, j \in [s]$. Since u_i is unrelated and adjacent to v_i, it follows that u_i is adjacent to all vertices v_j, $j \in [s]$, as these vertices are also unrelated to u_i and have the same label as v_j. We conclude that $u_1, \ldots, u_s$ and $v_1, \ldots, v_s$ form a bi-clique $K_{s,s}$ in G, a contradiction. □

Since the number of possible signatures is m^2, by Claim 3 we infer that the total number of blocks is at most $4(s-1)m^2 + 1$. As we argued, each block has at most $g(h-1, m, s)$ vertices, implying $p \leqslant (4(s-1)m^2 + 1) \cdot g(h-1, m, s)$. As P was chosen arbitrarily, we may set

$$g(h, m, s) \coloneqq (4(s-1)m^2 + 1) \cdot g(h-1, m, s).$$

This concludes the inductive proof. □

C Proof of Lemma 13

Proof (of Lemma 13). 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 9.

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 $\gamma \colon V(G) \to [N]$ 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: $\mathcal{U}_G = \{\gamma^{-1}(X) \colon X \subseteq [N], |X| = p\}$. Then $\mathcal{U} = (\mathcal{U}_G)_{G \in \mathscr{C}}$ is a finite p-cover of $\mathscr{C}$ of 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 $\chi \colon V(G) \to \mathcal{P}(\mathcal{U}_G)$ 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 2^N colors. Take any p subsets $X_1, \ldots, X_p \subseteq \mathcal{U}_G$ 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 $\{x_1, \ldots, x_p\} \subseteq U$. 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 height at most i, which is a witness that $G[J]$ has treedepth at most i. □

D Proofs of Section 5.1

In this section we present the missing proofs of Section 5.1 as well as a second proof for Lemma 29.

D.1 Guided and guidable functions

Proof (of Lemma 24). For each connected 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 $g|_C$ is undefined for vertices outside of C, we may assume that $U_i^C \subseteq V(C)$ for each $i \in [\ell]$. It follows that g is guided by the guidance system $\mathcal{U} = \{U_1, \ldots, U_\ell\}$ defined by setting U_i to be the union of U_i^C throughout connected components C of G. □

Proof (of Lemma 25). Let $\mathcal{U}_i$ be a guidance system of size at most ℓ that such that g_i is guided by $\mathcal{U}_i$. Then $\mathcal{U} = \bigcup_{i=1}^{s} \mathcal{U}_i$ is a guidance system of size at most $\ell \cdot s$. It is easy to see that $\mathcal{U}$ guides the partial function g. □

Proof (of Lemma 26). Let $\mathcal{U}$ be a guidance system of size at most ℓ such that f_G is guided by $\mathcal{U}$. For each vertex x such that $f(x)$ is a neighbor of x, pick an arbitrary set $V(x) \in \mathcal{U}$ such that $f(x)$ is the unique neighbor of x in $V(x)$.

We now present an almost quantifier-free transduction that constructs f_G. First, for each $U \in \mathcal{U}$ use a unary lift to introduce a unary predicate that selects the vertices of U. Next, introduce two unary predicates, *Null* and *Self*, which select the vertices x such that $f(x)$ is undefined or $f(x) = x$, respectively. Finally, for each $V \in \mathcal{U}$ introduce a unary predicate G_V that selects vertices x with $V(x) = V$. Now, for each $U \in \mathcal{U}$, construct the partial function d_U which maps every vertex x to its unique neighbor in U (if it exists) using the function extension operation parameterized by the formula $E(x, y) \wedge U(y)$. Finally, construct f_G using the function extension operation parameterized by the formula $\alpha(x, y)$ stating that $x \notin$ *Null* and either $x \in$ *Self* and $y = x$, or $x \in G_V$ and $y = d_U(x)$. □

D.2 Greedy proof of Lemma 29

We now present the second proof of Lemma 29. As asserted by Lemma 27, graphs from a fixed class of bounded shrubdepth do not admit arbitrarily long induced paths. We need a strengthening of this statement: classes of bounded shrubdepth also exclude induced structures that roughly resemble paths, as made precise next.

Definition D.43. 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 of 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 27.

Lemma D.44. *For every class $\mathscr{C}$ of graphs of 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 4. *There exists a function $f \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ 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 $\lambda \colon V(G) \to \Lambda$ 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

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

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 5. *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 5 implies that

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

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

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

This concludes the proof of Claim 4 and of Lemma D.44. □

Now Lemma 29 immediately follows from the following (essentially reformulated) statement.

Lemma D.45. *For every class $\mathscr{C}$ of graphs of 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 $B_1, \ldots, B_\ell$ 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

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

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 D.44, there is a universal bound q depending only on $\mathscr{C}$ on the length of quasi-paths in G, implying that we may take $p = \lfloor q/2 \rfloor$. □

E Proof of Lemma 32

Proof (of Lemma 32). It is enough to consider the case when I is an atomic operation. We assume that the input structure is a bundling $\bigcup \mathcal{K}^X$ of $\mathcal{K}$, given by a function $f \colon V(\bigcup \mathcal{K}) \to X$. Note that elements of $V(\bigcup \mathcal{K})$ can be identified in the structure as those that are in the domain of f.

Let $\sim$ be the equivalence relation on $V(\bigcup \mathcal{K})$, where $x \sim y$ if and only if $f(x) = f(y)$. Note that $\sim$ can be added to the structure by an extension operation parameterized by the formula $f(x) = f(y)$. We now consider cases depending on what atomic operation I is.

- If I is a reduct or restriction operation, then we set $\mathsf{I}^\star = \mathsf{I}$ (we may assume that a restriction does not remove elements of X by appropriate relativization, so that $\mathsf{I}^\star$ indeed outputs a bundling).

- If I is an extension operation parameterized by a quantifier-free formula $\varphi(x_1, \ldots, x_k)$, then set $\mathsf{I}^\star$ to be the extension operation parameterized by the formula $\varphi(x_1, \ldots, x_k) \wedge \bigwedge_{i,j \in [k]} (x_i \sim x_j)$.

- If I is a function extension operation parameterized by a formula $\varphi(x, y)$, then set $\mathsf{I}^\star$ to be function extension operation parameterized by the formula $\varphi(x, y) \wedge (x \sim y)$.

- If I is a copy operation, then $\mathsf{I}^\star$ is defined as the composition of a copy operation and a function extension operation that introduces a new function $f^\star$ in place of f defined as follows. We first define a function $\mathsf{origin}(x)$ as follows. Recall that when copying, we introduce a new unary predicate, say P, marking the newly created vertices and each vertex is made adjacent to its new copy. We let $\mathsf{origin}(x)$

be defined by $\psi_{\mathsf{origin}}(x, y) \coloneqq P(x) \wedge E(x, y)$. We now define $f^\star(x) = f(\mathsf{origin}(x))$. The resulting bundling is given by the function $f^\star$.

- If l is a unary lift, say parameterized by a function σ, then set $\mathsf{l}^\star$ to be the unary lift parameterized by the function $\sigma^\star$ that applies σ to each structure from $\mathcal{K}$ separately, investigates all possible ways of picking one output for each structure in $\mathcal{K}$, and returns the set of bundlings of sets formed in this way.

F Quantifier elimination

In this section we provide the missing proofs of the lemmas from Section 6.

F.1 Proof of Lemma 33

Proof (of Lemma 33). We show that if $\mathscr{C}$ is a class of graphs of bounded expansion, $G \in \mathscr{C}$ and $f\colon V(G) \rightharpoonup V(G)$ is a partial function that is guarded by G, then f is ℓ-guidable, for some ℓ depending only on $\mathscr{C}$. Then the claim of the lemma follows by Lemma 26.

First, consider the special case when $\mathscr{C}$ is a class of treedepth h, for some $h \in \mathbb{N}$. For each $G \in \mathscr{C}$, fix a forest F of depth h with $V(F) = V(G)$ such that every edge in G connects comparable nodes of F. Label every vertex v of G by the depth of v in the forest F, using labels $\{1, \ldots, h\}$. It is easy to see that the corresponding partition of $V(G)$ is a guidance system of order h for f.

Now the general case, when $\mathscr{C}$ is a class which has a 2-cover $\mathcal{U}$ of bounded treedepth. Let $N = \sup\{|\mathcal{U}_G| : G \in \mathscr{C}\}$, and let h be the treedepth of the class $\mathscr{C}[\mathcal{U}]$. Let $G \in \mathscr{C}$ be a graph and let $f\colon V(G) \to V(G)$ be a function which is guarded by G. Then $f|_U$ is h-guidable by the previous case, and hence f is $(h \cdot N)$-guidable by Lemma 25. $\square$

F.2 Proof of Lemma 35: quantifier elimination on trees of 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 $\nu\colon X \to V(t)$ 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 35.

Lemma F.46. *Let $\mathscr{T}$ be a class of Σ-labeled trees of 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*

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

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

Corollary F.47. *The same statement as above holds for a class $\mathscr{F}$ of Σ-labeled forests of 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 of 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 F.46 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. □

Corollary F.47 immediately implies Lemma 35. It remains to prove Lemma F.46. Before proving Lemma F.46, we recall some standard automata-theoretic techniques.

We define tree automata which 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 [28] 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 $X \restriction_\tau$ 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 $(a, X \restriction_\tau, q) \in \delta$.

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 F.48. *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 $f\colon \Sigma \to \Gamma$ 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 (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 $\nu\colon X \rightharpoonup V(t)$, 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 F.46, 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 $\nu\colon X \to V(t)$. By Lemma F.48(4) 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 $\nu\colon X \rightharpoonup V(t)$, 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:

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

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 $(a, C_\emptyset(u) \restriction_M)$, 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 F.46. To this end, given a valuation ν of X in $\widehat{t}$, let $\widehat{t}|_\nu$ denote the Γ_X-labeled tree obtained from $\widehat{t} \otimes \nu$ by restricting the node set to the set of ancestors of nodes in the image $\nu(X)$ of ν.

Lemma F.49. *There is a set of Γ_X-labeled trees $\mathcal{R}$ such that for every Σ-labeled tree t and valuation ν of X in t,*

$$t, \nu \models \varphi \quad \textit{if and only if} \quad \widehat{t}|_\nu \in \mathcal{R}.$$

Proof. Fix a tree t and a valuation ν 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 ν. For node u of t define the following Q-multisets:

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

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 ν such that the functions N_ν and $N_\emptyset$ take values in Z. Fix a node u of t.

Claim 6. *The state $\rho_\nu(u)$ is uniquely determined by the label of u in $t \otimes \nu$, and the Q-multisets $C_\emptyset(u) \restriction_M, N_\emptyset(u)$ and $N_\nu(u)$, i.e., there is a function $f \colon \Sigma_X \times \Delta \times Z \times Z \to Q$ such that for every tree t, valuation ν and node u,*

$$\rho_\nu(u) = f(\,\text{label of } u \text{ in } t \otimes \nu\,,\, C_\emptyset(u) \restriction_M,\, N_\emptyset(u)\,,\, N_\nu(u)\,). \tag{3}$$

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

$$C_\nu(u) = (C_\emptyset(u) - N_\emptyset(u)) + N_\nu(u). \tag{4}$$

This is because the automaton $\mathcal{A}_\psi$ is deterministic and therefore $\rho_\nu(w) = \rho_\emptyset(w)$ for all nodes w which are not nonempty. From Equation 4, the fact that $N_\emptyset(u)$ has at most $|X|$ elements and $M = K + |X|$, it follows that

$$((C_\emptyset(u) \restriction_M - N_\emptyset(u)) + N_\nu(u)) \restriction_K = (C_\nu(u)) \restriction_K . \tag{5}$$

By definition of the run of $\mathcal{A}_\psi$ on $t \otimes \nu$, the state $\rho_\nu(u)$ is determined by the label of u in $t \otimes \nu$ and by $(C_\nu(u)) \restriction_K$. It follows from Equation 5 that $\rho_\nu(u)$ is uniquely determined by the label of u in $t \otimes \nu$, $(C_\emptyset(u)) \restriction_M$, and the Q-multisets $N_\emptyset(u)$ and $N_\nu(u)$, proving the claim. □

From Claim 6 it follows that the state $\rho_\nu(r)$, where r is the root of t, depends only on the tree $\widehat{t}|_\nu$. Indeed, we can inductively compute the states $\rho_\nu(u)$ and $\rho_\emptyset(u)$, moving from the leaves of $\widehat{t}|_\nu$ towards the root, as follows. Suppose u is a node of $\widehat{t}|_\nu$ such that $\rho_\nu(v)$ 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 $\widehat{t}|_\nu$). Then, we can determine the multisets $N_\nu(u)$ and $N_\emptyset(u)$ using their definitions, and consequently, we can determine $\rho_\nu(u)$ by Equation 3, whereas $\rho_\emptyset(u)$ only depends on $C_\emptyset(u) \restriction_K$ and on the label of u in t. Note that both the label of u in t and the multiset $C_\emptyset(u) \restriction_K$ are encoded in the label of u in $\widehat{t}$.

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

Finally, we observe the following.

Lemma F.50. *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 ν of X in t, we have*

$$t, \nu \models \psi_s \qquad \textit{if and only if} \qquad t|_\nu \textit{ is isomorphic to } s.$$

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 F.46.

Proof (of Lemma F.46). 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 Lemma F.49, it now suffices to define $\widehat{\varphi}$ as the disjunction of formulas ψ_s provided by Lemma F.50 over $s \in \mathcal{R}_h$. □