BIROn - Birkbeck Institutional Research Online

Chen, Hubie and Mueller, M. (2017) One hierarchy spawns another: graph deconstructions and the complexity classification of conjunctive queries. ACM Transactions on Computational Logic 18 (4), ISSN 1529-3785.

Downloaded from: https://eprints.bbk.ac.uk/id/eprint/21940/

Usage Guidelines:
Please refer to usage guidelines at https://eprints.bbk.ac.uk/policies.html or alternatively contact lib-eprints@bbk.ac.uk.

arXiv:1307.1353v2 [cs.CC] 1 Mar 2016

One Hierarchy Spawns Another: Graph Deconstructions and the Complexity Classification of Conjunctive Queries*

Hubie Chen
Universidad del País Vasco
San Sebastián, Spain
IKERBASQUE, Bilbao, Spain
hubie.chen@ehu.es

Moritz Müller
Kurt Gödel Research Center
Universität Wien, Austria
moritz.mueller@univie.ac.at

Abstract

We study the problem of conjunctive query evaluation relative to a class of queries; this problem is formulated here as the relational homomorphism problem relative to a class of structures $\mathcal{A}$, wherein each instance must be a pair of structures such that the first structure is an element of $\mathcal{A}$. We present a comprehensive complexity classification of these problems, which strongly links graph-theoretic properties of $\mathcal{A}$ to the complexity of the corresponding homomorphism problem. In particular, we define a binary relation on graph classes, which is a preorder, and completely describe the resulting hierarchy given by this relation. This relation is defined in terms of a notion which we call *graph deconstruction* and which is a variant of the well-known notion of tree decomposition. We then use this hierarchy of graph classes to infer a complexity hierarchy of homomorphism problems which is comprehensive up to a computationally very weak notion of reduction, namely, a parameterized version of quantifier-free first-order reduction. In doing so, we obtain a significantly refined complexity classification of homomorphism problems, as well as a unifying, modular, and conceptually clean treatment of existing complexity classifications. We then present and develop the theory of Ehrenfeucht-Fraïssé-style pebble games which solve the homomorphism problems where the cores of the structures in $\mathcal{A}$ have bounded tree depth. Finally, we use our framework to classify the complexity of model checking existential sentences having bounded quantifier rank.

Keywords: Conjunctive queries, Homomorphisms, Graph decompositions, Parameterized complexity

AMS subject classifications: 05C75, 05C83, 03B70, 68Q17, 68Q19.

*An extended abstract of this work has appeared in [12].

1 Introduction

Conjunctive queries are basic and heavily studied database queries, and can be viewed logically as formulas consisting of a sequence of existentially quantified variables, followed by a conjunction of atomic formulas on those variables. Since the 1977 article of Chandra and Merlin [8], complexity-theoretic aspects of conjunctive queries have been a research subject of persistent and enduring interest which continues to the present day (as discussed and evidenced, for example, by the works [1, 30, 34, 25, 26, 27, 37, 32]). In this article, we study *conjunctive query evaluation*, which is the problem of evaluating a conjunctive query on a relational structure. Conjunctive query evaluation is indeed equivalent to a number of well-known and well-studied problems, including the homomorphism problem on relational structures, the constraint satisfaction problem, and conjunctive query containment [8, 30]. That this problem appears in many equivalent guises attests to its fundamental, primal nature, and this problem has correspondingly been approached and studied from a wide variety of perspectives and motivations.

As has been eloquently articulated in the literature [34], the employment of classical complexity notions such as polynomial-time tractability to grade the complexity of conjunctive query evaluation is not totally satisfactory: a typical scenario—for example, in the database context—is the evaluation of a relatively short query on a relatively large structure. This asymmetry between the two parts of the input suggests a notion of complexity wherein one relaxes the dependence on the query. For example, in measuring time complexity, one might allow a non-polynomial dependence on the query while enforcing a polynomial dependence on the structure. *Parameterized complexity theory* [24, 19] is a comprehensive framework for studying problems where each instance has an associated parameter, and arbitrary dependence in the parameter is permitted; in the query evaluation setting, the query (or the query length) is naturally taken to be the parameter. We use and focus on this viewpoint in the present article, and hence in this discussion.

Conjunctive query evaluation is known to be computationally intractable in general, and consequently a recurring theme in the study of this problem is the identification of structural properties of conjunctive queries that provide tractability or other computationally desirable behaviors. A well-studied framework in which to seek such properties is the family of parameterized homomorphism problems p-$\textsc{Hom}(\mathcal{A})$, for classes of relational structures $\mathcal{A}$: the problem p-$\textsc{Hom}(\mathcal{A})$ is to decide, given a relational structure $\mathbf{A}$ from $\mathcal{A}$ and another relational structure $\mathbf{B}$, whether there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$; the parameter here is the first structure $\mathbf{A}$. Studying this problem family amounts to studying conjunctive query evaluation on various classes of conjunctive queries, as it is a classical fact that each boolean conjunctive query ϕ can be bijectively represented as a structure $\mathbf{A}$ in such a way that, for any structure $\mathbf{B}$, it holds that ϕ is true on $\mathbf{B}$ if and only if $\mathbf{A}$ admits a homomorphism to $\mathbf{B}$ [8]. We will focus on the case where $\mathcal{A}$ has bounded arity, and assume this property throughout this discussion.

Known classifications.

An exact description of the tractable problems of the form p-$\textsc{Hom}(\mathcal{A})$ was presented by Grohe [27]. In particular, a known sufficient condition for fixed-parameter tractability of p-$\textsc{Hom}(\mathcal{A})$ was that the *cores* of $\mathcal{A}$ have *bounded treewidth* [16]; Grohe completed the picture by showing that for any class $\mathcal{A}$ not satisfying this condition, the problem p-$\textsc{Hom}(\mathcal{A})$ is W[1]-complete. Fixed-parameter tractability is a parameterized relaxation of polynomial-time tractability, and W[1]-hardness can be conceived of as a parameterized analog of NP-hardness. Intuitively, the *core* of a structure is an equivalent structure of minimal size.

Later, a classification of the tractable p-$\textsc{Hom}(\mathcal{A})$ problems was presented [11]. This classification is exhaustive up to parameterized logarithmic space reductions; parameterized logarithmic space (para-L) relaxes logarithmic space in a way analogous to that in which fixed-parameter tractability relaxes polynomial time. Let $\mathcal{T}$ denote the class of all trees, let $\mathcal{P}$ denote the class of all paths, and for a class of structures $\mathcal{A}$, let $\mathcal{A}^*$ be the class of structures obtainable by taking a structure $\mathbf{A}$ in $\mathcal{A}$ and giving each element its own color. The classification states that each tractable p-$\textsc{Hom}(\mathcal{A})$ problem is para-L equivalent to p-$\textsc{Hom}(\mathcal{T}^*)$, para-L equivalent to p-$\textsc{Hom}(\mathcal{P}^*)$, or decidable in para-L. The properties determining which of the three behaviors occurs are *bounded pathwidth* and *bounded tree depth*, established graph-theoretical properties.

1.1 Contributions

In this article, we present a significantly refined complexity classification of the homomorphism problems p-$\textsc{Hom}(\mathcal{A})$ which is exhaustive up to an extremely simple and computationally weak notion of reduction based on quantifier-free interpretations from first-order logic. Our classification generalizes the just-described known classifications, and indeed our present study provides a uniform, modular, and self-contained treatment thereof.

After presenting the classification, we introduce and develop the theory of Ehrenfeucht-Fraïssé-style pebble games for solving the problems lying at the lower end of our hierarchy; in doing so, we obtain a characterization of the homomorphism problems $\textsc{Hom}(\mathcal{A})$ solvable in classical logarithmic space. As a further technical contribution, we utilize our framework to analyze the complexity of model checking existential sentences.

We now give an overview of and further details on the contributions and obtained results; we refer the reader to the technical sections for precise statements.

A graph-theoretic hierarchy.

Previous work [11] related the complexity of conjunctive queries to the named graph-theoretic properties by showing that certain relationships on graph classes implied reductions for the corresponding homomorphism problems, for example:

- If $\mathcal{G}$ is a graph class and $\mathcal{M}$ is the class of minors of graphs in $\mathcal{G}$, then p-$\textsc{Hom}(\mathcal{M}^*)$ reduces to p-$\textsc{Hom}(\mathcal{G}^*)$.

- If the members of a graph class $\mathcal{G}$ have bounded width tree decompositions whose trees lie in a graph class $\mathcal{H}$, then p-Hom$(\mathcal{G}^*)$ reduces to p-Hom$(\mathcal{H}^*)$.

Another known and important reduction [28] is as follows:

- When $\mathcal{R}$ is the class of all grids and $\mathcal{G}$ is any graph class, then p-Hom$(\mathcal{G}^*)$ reduces to p-Hom$(\mathcal{R}^*)$.

We give a unified explanation for all of these reductions by defining a binary relation $\preceq$ on graph classes. This relation has the key property:

$$\text{If } \mathcal{G} \preceq \mathcal{H}, \text{ then } p\text{-Hom}(\mathcal{G}^*) \text{ reduces to } p\text{-Hom}(\mathcal{H}^*). \tag{1}$$

This key property is shown to imply the three just-named results.

The definition of the relation $\preceq$ is simple and is based on a notion which we call *graph deconstruction* and which is strongly related to and inspired by the notion of tree decomposition. When $\mathbf{G}$ and $\mathbf{H}$ are graphs, we define an $\mathbf{H}$-deconstruction of $\mathbf{G}$ to be a family $(B_h)_{h\in H}$ of subsets of the vertex set of $\mathbf{G}$ which is indexed by the vertex set H of $\mathbf{H}$ and which satisfies properties similar to those in the definition of tree decomposition (Definition 3.1); note that here, the graph $\mathbf{H}$ is not restricted to be a tree, as it is in the definition of tree decomposition. Each such deconstruction $(B_h)_{h\in H}$ has associated with it a measure called its *width* which is based on the sizes of the subsets B_h. For graph classes $\mathcal{G}$ and $\mathcal{H}$, we define $\mathcal{G} \preceq \mathcal{H}$ if and only if there is a constant w where for each graph $\mathbf{G}$ in $\mathcal{G}$, there is a graph $\mathbf{H}$ in $\mathcal{H}$ such that $\mathbf{G}$ has an $\mathbf{H}$-deconstruction of width at most w.

We describe completely the hierarchy that this relation yields on graph classes. Define $\mathcal{T}_n$ to be the class of trees of height at most n; $\mathcal{F}_n$ to be the class of forests of height at most n; and $\mathcal{L}$ to be the class of all graphs. We present the following hierarchy:

$$\mathcal{T}_0 \preceq \mathcal{F}_0 \preceq \mathcal{T}_1 \preceq \mathcal{F}_1 \preceq \cdots \preceq \mathcal{P} \preceq \mathcal{T} \preceq \mathcal{L}.$$

We prove that this hierarchy is strict and *comprehensive* in that every graph class is equivalent to exactly one of the classes in the hierarchy (Theorem 3.16). To understand the upper levels of the hierarchy, we use known excluded minor theorems. To determine the lower part of the hierarchy (below $\mathcal{P}$), we introduce a new complexity measure on graph classes $\mathcal{G}$ which we call *stack depth*: the maximum d such that all depth d trees are minors of $\mathcal{G}$; this equals the minimum depth of forests that allow for bounded width decompositions of the graphs in $\mathcal{G}$ (Lemma 3.23).

A complexity-theoretic hierarchy.

Having understood the relation $\preceq$ on graph classes, we then turn to study homomorphism problems. As mentioned, we prove that $\mathcal{G} \preceq \mathcal{H}$ implies p-Hom$(\mathcal{G}^*)$ reduces to p-Hom$(\mathcal{H}^*)$ (Theorem 5.9); this property was indeed a primary motivation for the definition of $\preceq$. We prove this with respect to a computationally weak notion of reduction that we call *quantifier-free after a precomputation (qfap)*. This notion of reduction naturally unifies and incorporates two modes of computation that have long been studied. For each parameter, this reduction provides a quantifier-free, first-order *interpretation* that defines the output instance in the input instance; interpretations as reductions have a tradition in descriptive complexity theory [29, 20], and Dawar and He studied a type of quantifier-free interpretation in the parameterized setting [17]. In addition, our reduction allows for *precomputation* on the parameter of an input instance. This follows an established schema in the definition of parameterized modes of computation: fixed-parameter tractability can be defined as polynomial time after a precomputation, and para-L can be defined as logarithmic space after a precomputation [23].

Let $\mathcal{A}$ be an arbitrary class of bounded-arity structures. We prove that p-$\textsc{Hom}(\mathcal{A})$ is equivalent under qfap-reductions to p-$\textsc{Hom}(\mathcal{G}^*)$, where $\mathcal{G}$ is the class of graphs of the cores of the structures in $\mathcal{A}$ (Theorem 5.10). By this theorem, property (1) and our description of the graph hierarchy, we obtain:

Every problem of the form p-$\textsc{Hom}(\mathcal{A})$ is equivalent, under qfap-reduction, to a problem p-$\textsc{Hom}(\mathcal{H}^*)$, where $\mathcal{H}$ is a graph class from the graph hierarchy.

This interestingly implies that, with respect to qfap-reduction, the complexity degrees attained by problems p-$\textsc{Hom}(\mathcal{A})$ are linearly ordered according to the classes in our graph hierarchy, in particular, linearly ordered in a sequence of order type $\omega + 3$.

In brief, our approach for understanding the family of problems p-$\textsc{Hom}(\mathcal{A})$ is to present a graph hierarchy and then show that this hierarchy induces a complexity hierarchy for these problems. *We wish to emphasize the unifying nature, the modularity, and the conceptual cleanliness of this approach.* Our definition and presentation of the graph hierarchy cleanly and neatly encapsulates the graph-theoretic content needed to present the complexity hierarchy. We obtain a uniform and self-contained derivation of the mentioned known classifications [28, 27, 11], which derivation we find to be clearer and simpler than those of the original works. This uniform derivation is a testament to the utility of the graph hierarchy, and we view the simple definitions of *graph deconstruction* and the relation $\preceq$, as well as the development of their basic theory, as conceptual contributions.

Our treatment strengthens the known classifications, since the problems classified as being computationally equivalent are here shown to be so under qfap-reductions; in particular, it follows from our treatment that all of the W[1]-complete problems from Grohe's theorem [27] are pairwise equivalent under qfap-reductions and hence in a very strong sense.

A proof of Grohe's theorem.

We present a modular, conceptually concise, and relatively transparent proof of Grohe's celebrated hardness result (Section 4). In particular, we make simple use of our notion of graph deconstruction, and we obtain this hardness result essentially as the composition of three readily comprehensible polynomial-time reductions.

Consistency, pebble games, and logarithmic space.

The study of so-called consistency algorithms has a long tradition in research on constraint satisfaction and homomorphism problems. Such algorithms are typically efficient and simple heuristics that can detect inconsistency (that is, that an instance is a *no* instance) and are based on local reasoning. Identifying cases of these problems where such algorithms provide a sound and complete decision procedure has been a central theme in the tractability theory of these problems (see for example [31, 16, 4, 9, 7, 3, 10]).

Seminal work of Koalitits and Vardi [31] showed that certain natural consistency algorithms could be viewed as determining the winner in certain Ehrenfeucht-Fraïssé type pebble games [20]. Since this work, there has been sustained effort devoted to presenting pebble games that solve cases of the homomorphism problem. For example, there is a study of pebble games that solve p-$\textsc{Hom}(\mathcal{A})$ when the class $\mathcal{G}$ of graphs of structures from $\mathcal{A}$ has bounded treewidth [16, 2], and also when $\mathcal{G}$ has bounded pathwidth [15].

We complete the picture by presenting natural pebble games that are shown to solve p-$\textsc{Hom}(\mathcal{A})$ when $\mathcal{G}$ has bounded tree depth (Section 6). Our pebble games are finite-round games that can be thought of as homomorphism variants of the classical Ehrenfeucht-Fraïssé game. We develop the theory of our games, showing for example that it is decidable, given a structure $\mathbf{A}$, whether or not a particular game solves the homomorphism problem on $\mathbf{A}$ (Theorem 6.1). We also show equality ± 1 between the number of pebbles needed to solve p-$\textsc{Hom}(\mathcal{A})$ and the tree depth of $\mathcal{G}$; and, between the number of rounds needed to solve p-$\textsc{Hom}(\mathcal{A})$ and the stack depth of $\mathcal{G}$ (Section 6). We believe that the latter result reinforces the suggestion that stack depth is a natural graph-theoretic measure.

We obtain a characterization of the classical homomorphism problems $\textsc{Hom}(\mathcal{A})$ decidable in classical logarithmic space: these are precisely those where the cores of structures from $\mathcal{A}$ have bounded tree depth (Section 7). This characterization is established under a natural hypothesis from parameterized space complexity.

Model checking existential sentences.

The given hierarchies, along with the notion of qfap reduction, provide a clean and comprehensive understanding of the complexity degrees of parameterized homomorphism problems. We expect that the given hierarchies can be meaningfully used to obtain a fine-grained understanding of the complexity of other problems of independent interest. We show that the hierarchy can be used to classify the complexity of model-checking existential sentences having bounded quantifier rank (Section 8).

2 Preliminaries

For $n \in \mathbb{N}$ we let $[n]$ denote $\{1, \ldots, n\}$ and understand $[0] = \emptyset$.

2.1 Structures

A *vocabulary* is a finite set of relation symbols, where each symbol R has an associated arity $\mathrm{ar}(R) \in \mathbb{N}$. A *structure* $\mathbf{B}$ *with vocabulary* σ, for short, a σ*-structure* is given by a non-empty set B called its *universe* together with an *interpretation* $R^{\mathbf{B}} \subseteq B^{\mathrm{ar}(R)}$ of R for every $R \in \sigma$. We only consider finite structures, i.e. structures with finite universe. When $\mathbf{B}$ is a σ-structure and S a non-empty subset of B, we let $\langle S \rangle^{\mathbf{B}}$ denote the σ-structure *induced* in $\mathbf{B}$ on S: it has universe S and interprets every $R \in \sigma$ by $S^{\mathrm{ar}(R)} \cap R^{\mathbf{B}}$. The class of all σ-structures is denoted by $\mathrm{STR}[\sigma]$, and the class of all structures by STR. The *product* $\mathbf{A} \times \mathbf{B}$ of two σ-structures $\mathbf{A}$ and $\mathbf{B}$ has universe $A \times B$ and interprets $R \in \sigma$ by

$$R^{\mathbf{A}\times\mathbf{B}} := \big\{((a_1, b_1), \ldots, (a_{\mathrm{ar}(R)}, b_{\mathrm{ar}(R)})) \mid \bar{a} \in R^{\mathbf{A}}, \bar{b} \in R^{\mathbf{B}}\big\}.$$

For a vocabulary σ, a (first-order) σ-formula φ is built from *atoms* $Rx_1 \cdots x_{\mathrm{ar}(R)}$ and $x = y$ where x, y and the x_i are variables and $R \in \sigma$, by means of Boolean combinations $\vee, \wedge, \neg$ and existential and universal quantification $\exists x, \forall x$. We write $\varphi(\bar{x})$ for φ to indicate that the free variables of φ are among the components of $\bar{x} = x_1 \cdots x_r$; for a σ-structure, and $\bar{a} = a_1 \cdots a_r \in A^r$ we write $\mathbf{A} \models \varphi(\bar{a})$ to indicate that $\bar{a}$ satisfies $\varphi(\bar{x})$ in $\mathbf{A}$, further we write $\varphi(\mathbf{A}) := \{\bar{a} \in A^r \mid \mathbf{A} \models \varphi(\bar{a})\}$. Formulas without free variables are *sentences*.

Two structures $\mathbf{A}$ and $\mathbf{B}$ interpreting the same vocabulary are called *similar*. In this case, a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a function $h : A \to B$ such that for every $R \in \sigma$ and every tuple $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$, it holds that $(h(a_1), \ldots, h(a_{\mathrm{ar}(R)})) \in R^{\mathbf{B}}$. We write $\mathbf{A} \xrightarrow{h} \mathbf{B}$ to indicate that such a homomorphism exists. A *partial homomorphism* h from $\mathbf{A}$ to $\mathbf{B}$ is either $\varnothing$ or a homomorphism from $\langle \mathsf{dom}(h) \rangle^{\mathbf{A}}$ to $\mathbf{B}$; by $\mathsf{dom}(h)$ we denote the domain of h and by $\mathsf{im}(h)$ its image.

A structure $\mathbf{A}$ is a *core* if all homomorphisms from $\mathbf{A}$ to itself are injective. The *core of a structure* $\mathbf{A}$ is a structure $\mathbf{B}$ such that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, $\mathbf{B}$ is a core, $B \subseteq A$, and $R^{\mathbf{B}} \subseteq R^{\mathbf{A}}$ for each symbol R. It is well-known that each finite structure has at least one core, and that all cores of a finite structure are isomorphic; for this reason, one often speaks of *the* core of a finite structure $\mathbf{A}$, which we denote here by $\mathsf{core}(\mathbf{A})$.

For example, all structures of the form $\mathbf{A}^*$ are cores. Here, $\mathbf{A}^*$ is the expansion obtained from $\mathbf{A}$ by interpreting for each $a \in A$ a new unary relation symbol C_a by $C_a^{\mathbf{A}} := \{a\}$. For a class of structures $\mathcal{A}$ we let

$$\mathcal{A}^* := \{\mathbf{A}^* \mid \mathbf{A} \in \mathcal{A}\}.$$

2.2 Graphs

In this article, a *graph* is a $\{E\}$-structure $\mathbf{G}$ for a binary relation symbol E such that $E^{\mathbf{G}}$ is irreflexive and symmetric. A graph $\mathbf{G}$ is a *subgraph* of another graph $\mathbf{H}$ if $G \subseteq H$ and $E^{\mathbf{G}} \subseteq E^{\mathbf{H}}$. Given a graph $\mathbf{G}$ we write

$$\mathsf{refl}(E^{\mathbf{G}}) := E^{\mathbf{G}} \cup \{(g, g) \mid g \in G\}.$$

When $\mathbf{A}$ is a σ-structure, we let $\mathsf{graph}(\mathbf{A})$ be the "Gaifman" graph with universe A and an edge between $a, a' \in A$ if $a \neq a'$ and there are $R \in \sigma$ and a tuple $\bar{a} \in R^{\mathbf{A}}$ such that a, a' are components of $\bar{a}$. We call $\mathbf{A}$ *connected* if $\mathsf{graph}(\mathbf{A})$ is connected. A *(connected) component* of $\mathbf{A}$ is a structure induced in $\mathbf{A}$ on a (connected) component of $\mathsf{graph}(\mathbf{A})$.

By a *rooted tree*, we mean a tree that interprets a unary relation symbol root by a set containing a single element, called the *root* of the tree. By a *rooted forest*, we mean a graph $\mathbf{G}$ where, for each component C, the graph $\langle C\rangle^{\mathbf{G}}$ is a rooted tree. When a and d are elements of a rooted tree, we say that a is an *ancestor* of d and that d is a *descendent* of a if a lies on the unique path from the root to d; if in addition $a \neq d$, we say that a is a *proper ancestor* of d and that d is a *proper descendent* of a. The *height* of a rooted tree is the maximum length (number of edges) of a path from the root to some vertex. The height of a tree $\mathbf{T}$ is the minimum height of all rootings of $\mathbf{T}$. The height of a forest $\mathbf{F}$ is the maximum height of a (connected) component of $\mathbf{F}$.

The *tree depth* [33] of a connected graph $\mathbf{G}$ is the minimum $h \in \mathbb{N}$ such that there exists a rooted tree $\mathbf{T}$ with universe $T = G$ of height $\leqslant h$ such that $E^{\mathbf{G}}$ is contained in the closure of $\mathbf{T}$. The *closure* of $\mathbf{T}$ is the set of edges (g, g') such that either g is an ancestor of g' in $\mathbf{T}$ or vice versa. For an arbitrary graph, its tree depth is defined to be the maximum tree depth taken over all components of $\mathbf{G}$.

A *tree decomposition* of a graph $\mathbf{G}$ is a tree $\mathbf{H}$ along with an H-indexed family $(B_h)_{h\in H}$ of subsets of G satisfying the following conditions:

- (Coverage) For every pair $(g, g') \in \mathsf{refl}(E^{\mathbf{G}})$, there exists $h \in H$ such that $\{g, g'\} \subseteq B_h$.
- (Connectivity) For each element $g \in G$, the set $\{h \mid g \in B_h\}$ is connected in $\mathbf{H}$.

The *width* of a tree decomposition is $\max_{h\in H} |B_h| - 1$. A *path decomposition* of a graph is a tree decomposition where the tree $\mathbf{H}$ is a path. The *treewidth* of a graph $\mathbf{G}$ is the minimum width over all tree decompositions of $\mathbf{G}$; likewise, the *pathwidth* of $\mathbf{G}$ is the minimum width over all path decompositions of $\mathbf{G}$. A class $\mathcal{G}$ of graphs has *bounded treewidth* if there exists a constant w such that each graph $\mathbf{G} \in \mathcal{G}$ has treewidth $\leqslant w$; the properties of *bounded pathwidth* and *bounded tree depth* are defined similarly.

A graph $\mathbf{M}$ is a *minor* of a graph $\mathbf{G}$ if there exists a *minor map* from $\mathbf{M}$ to $\mathbf{G}$, which is a map μ defined on M where

- for each $m \in M$, it holds that $\mu(m)$ is a non-empty, connected subset of G;
- the sets $\mu(m)$ are pairwise disjoint; and,
- for each $(m, m') \in E^{\mathbf{M}}$ there exist $g \in \mu(m)$ and $g' \in \mu(m')$ such that $(g, g') \in E^{\mathbf{G}}$.

When $\mathbf{G}$ is a graph, we use $\mathsf{minors}(\mathbf{G})$ to denote the class of all minors of $\mathbf{G}$, and we extend this notation to a class $\mathcal{G}$ setting $\mathsf{minors}(\mathcal{G}) = \bigcup_{\mathbf{G}\in\mathcal{G}} \mathsf{minors}(\mathbf{G})$.

The following theorem is known; the first two parts are due to Robertson and Seymour's graph minor series [36, 35] and the third is due to Blumensath and Courcelle [5].

Theorem 2.1. *Let $\mathcal{G}$ be a class of graphs.*

1. *(Excluded grid theorem) $\mathcal{G}$ has bounded treewidth if and only if* $\mathsf{minors}(\mathcal{G})$ *does not contain all grids.*

2. *(Excluded tree theorem) $\mathcal{G}$ has bounded pathwidth if and only if* $\mathsf{minors}(\mathcal{G})$ *does not contain all trees.*
3. *(Excluded path theorem) $\mathcal{G}$ has bounded tree depth if and only if* $\mathsf{minors}(\mathcal{G})$ *does not contain all paths.*

Proposition 2.2. *Let $\mathbf{G}$ be a connected graph, and suppose that $\mathbf{T}$ is a rooted tree with height h witnessing that $\mathbf{G}$ has tree depth $\leqslant h$, i.e. $\mathbf{T}$ has height $\leqslant h$ and its closure contains $E^{\mathbf{G}}$. Then the tree $(T, E^{\mathbf{T}})$ together with $(B_t)_{t \in T}$ is a tree decomposition of $\mathbf{G}$ of width h where $B_t = \{a \mid a \text{ is an ancestor of } t\}$.*

To prove Proposition 2.2, the key observation is the following. For any two vertices $g, g' \in G$ connected by an edge in $\mathbf{G}$, one is an ancestor of the other in $\mathbf{T}$, and if, say, g' is an ancestor of g, then $g, g' \in B_g$.

3 Graph deconstructions

In this section, we present the notion of *graph deconstruction* and develop its basic theory (Section 3.1); we introduce the measure of *stack depth* (Section 3.2); and, we state and prove our theorem presenting the graph hierarchy (Section 3.3).

3.1 Definitions and basic properties

Definition 3.1. When $\mathbf{G}$ and $\mathbf{H}$ are graphs, an $\mathbf{H}$-*deconstruction of* $\mathbf{G}$ is an H-indexed family $(B_h)_{h \in H}$ of subsets of G that satisfies the following two conditions:

– (Coverage) For each pair $(g, g') \in \mathsf{refl}(E^{\mathbf{G}})$, there exists a pair $(h, h') \in \mathsf{refl}(E^{\mathbf{H}})$ such that $\{g, g'\} \subseteq B_h \cup B_{h'}$.

– (Connectivity) For each element $g \in G$, the set $\{h \mid g \in B_h\}$ is connected in $\mathbf{H}$.

The *width* of an $\mathbf{H}$-deconstruction $(B_h)_{h \in H}$ is defined as

$$\max_{(h,h') \in \mathsf{refl}(E^{\mathbf{H}})} |B_h \cup B_{h'}|.$$

We will refer to the subsets B_h as *bags*.

Note that the definition of an $\mathbf{H}$-deconstruction is similar to that of a tree decomposition, but one important difference is that, in the definition of $\mathbf{H}$-deconstruction, it is not required that an edge $(g, g') \in E^{\mathbf{G}}$ lie inside a single bag B_h, but rather, may lie inside the union $B_h \cup B_{h'}$ of two bags where $(h, h') \in E^{\mathbf{H}}$.

Remark 3.2. Another natural way to define the width of an $\mathbf{H}$-deconstruction $(B_h)_{h \in H}$ is simply as $\max_{h \in H} |B_h|$. The theory that we develop is essentially unchanged if one adopts this alternative definition.

Example 3.3. Let $n \geqslant 1$. Let $\mathbf{H}$ be the n-by-n grid: it has vertices $[n]^2$ and an edge between (i, j) and (i', j') if $|i - i'| + |j - j'| = 1$. Any graph $\mathbf{G}$ on n vertices has an $\mathbf{H}$-deconstruction of width $\leqslant 3$. Assume without loss of generality that $G = [n]$. The desired deconstruction is $(B_{(i,j)})_{(i,j)\in H}$ defined by $B_{(i,j)} = \{i, j\}$. Coverage holds, since each pair $(i, j) \in [n]^2$ has $\{i, j\} \subseteq B_{(i,j)}$. Connectivity holds, since for each $i \in [n]$, the set $\{h \mid i \in B_h\}$ forms a cross in the grid.

Example 3.4. For any graph $\mathbf{G}$, the family $(B_g)_{g\in G}$ defined by $B_g = \{g\}$ is a $\mathbf{G}$-deconstruction of $\mathbf{G}$ of width $\leqslant 2$.

Proposition 3.5. *Let* $\mathbf{G}, \mathbf{H}$ *be graphs,* $S \subseteq G$ *connected in* $\mathbf{G}$ *and* $(B_h)_{h\in H}$ *an* $\mathbf{H}$-*deconstruction of* $\mathbf{G}$. *Then* $\{h \mid S \cap B_h \neq \varnothing\}$ *is connected in* $\mathbf{H}$.

Proof. Suppose $(g, g') \in E^{\mathbf{G}}$, and define $T_g = \{h \mid g \in B_h\}$ and $T_{g'} = \{h \mid g' \in B_h\}$. It suffices to show $T_g \cup T_{g'}$ is connected in $\mathbf{H}$. By coverage, there is $(h, h') \in \mathsf{refl}(E^{\mathbf{H}})$ such that $\{g, g'\} \subseteq B_h \cup B_{h'}$. If $h = h'$, then T_g and $T_{g'}$ share the vertex h; otherwise $(h, h') \in E^{\mathbf{H}}$ with $g \in B_h$ and $g' \in B_{h'}$, so $h \in T_g$ and $h' \in T_{g'}$, and hence $T_g \cup T_{g'}$ is connected in $\mathbf{H}$. ☐

Let $\mathcal{G}$ and $\mathcal{H}$ be classes of graphs.

- We say that $\mathcal{G}$ has $\mathcal{H}$-deconstructions of width $\leqslant k$ if for each graph $\mathbf{G} \in \mathcal{G}$, there exists a graph $\mathbf{H} \in \mathcal{H}$ such that $\mathbf{G}$ has an $\mathbf{H}$-deconstruction of width $\leqslant k$.
- We say that $\mathcal{G}$ has $\mathcal{H}$-deconstructions of bounded width if there exists $k \geqslant 1$ such that $\mathcal{G}$ has $\mathcal{H}$-deconstructions of width $\leqslant k$.

We will employ analogous terminology to discuss, for example, *nice deconstructions* which will be defined later.

Definition 3.6. We define the binary relation $\preceq$ on classes of graphs as follows: $\mathcal{G} \preceq \mathcal{H}$ if and only if $\mathcal{G}$ has $\mathcal{H}$-deconstructions of bounded width. We write $\mathcal{G} \equiv \mathcal{H}$ if $\mathcal{G} \preceq \mathcal{H}$ and $\mathcal{H} \preceq \mathcal{G}$, and we write $\mathcal{G} \precneqq \mathcal{H}$ if $\mathcal{G} \preceq \mathcal{H}$ and $\mathcal{G} \not\equiv \mathcal{H}$.

We now present some basic properties of the relation $\preceq$.

Proposition 3.7. *The relation* $\preceq$ *is reflexive and transitive.*

This implies that $\equiv$ is an equivalence relation. Throughout, we will tacitly use the fact that, if $\mathcal{G} \subseteq \mathcal{H}$, then $\mathcal{G} \preceq \mathcal{H}$.

Proof of Proposition 3.7. Reflexivity follows from Example 3.4 and transitivity from the following lemma. ☐

Lemma 3.8. *Let* $\mathbf{G}$, $\mathbf{H}$, *and* $\mathbf{I}$ *be graphs; suppose that* $\mathbf{G}$ *has an* $\mathbf{H}$-*deconstruction of width* $\leqslant v$, *and that* $\mathbf{H}$ *has an* $\mathbf{I}$-*deconstruction of width* $\leqslant w$. *Then,* $\mathbf{G}$ *has an* $\mathbf{I}$-*deconstruction of width* $\leqslant 2vw$.

PROOF. Suppose that $\mathbf{G}$ has the $\mathbf{H}$-deconstruction $(B_h)_{h\in H}$ of width $\leqslant v$, and that $\mathbf{H}$ has the $\mathbf{I}$-deconstruction $(C_i)_{i\in I}$ of width $\leqslant w$. Define the family $(C_i^+)_{i\in I}$ by $C_i^+ = \bigcup_{h\in C_i} B_h$. We claim that $(C_i^+)_{i\in I}$ is an $\mathbf{I}$-deconstruction of $\mathbf{G}$, which suffices.

To verify coverage, let $(g, g') \in \mathsf{refl}(E^{\mathbf{G}})$. By coverage of (B_h), there exists $(h, h') \in \mathsf{refl}(E^{\mathbf{H}})$ such that $\{g, g'\} \subseteq B_h \cup B_{h'}$. By coverage of (C_i), there exists $(i, i') \in \mathsf{refl}(E^{\mathbf{I}})$ such that $\{h, h'\} \subseteq C_i \cup C_{i'}$. By definition of (C_i^+), we have that $B_h \cup B_{h'} \subseteq C_i^+ \cup C_{i'}^+$, from which it follows that $\{g, g'\} \subseteq C_i^+ \cup C_{i'}^+$.

To verify connectivity, let $g \in G$ be an arbitrary element. By connectivity of (B_h), it holds that $T = \{h \mid g \in B_h\}$ is connected in $\mathbf{H}$. It follows from Proposition 3.5 that $U = \{i \mid T \cap C_i \neq \varnothing\}$ is connected in $\mathbf{I}$. We claim that $U = \{i \mid g \in C_i^+\}$. This is because

$$\begin{aligned} T \cap C_i \neq \varnothing &\iff \exists h \in C_i : h \in T \\ &\iff \exists h \in C_i : g \in B_h \\ &\iff g \in C_i^+. \end{aligned}$$

□

Proposition 3.9. *For any class of graphs $\mathcal{G}$, it holds that $\mathcal{G} \equiv \mathsf{minors}(\mathcal{G})$.*

Proof. It is clear that $\mathcal{G} \preceq \mathsf{minors}(\mathcal{G})$, since $\mathcal{G} \subseteq \mathsf{minors}(\mathcal{G})$. To show that $\mathsf{minors}(\mathcal{G}) \preceq \mathcal{G}$, we prove that when a graph $\mathbf{M}$ is a minor of a graph $\mathbf{G}$, it holds that $\mathbf{M}$ has a $\mathbf{G}$-deconstruction of width $\leqslant 2$. Let μ be a minor map from $\mathbf{M}$ to $\mathbf{G}$, and define, for all $g \in G$, the set B_g to be $\{m \mid g \in \mu(m)\}$. Clearly, for each $g \in G$, it holds that $|B_g| \leqslant 1$. We claim that $(B_g)_{g\in G}$ is a $\mathbf{G}$-deconstruction of $\mathbf{M}$. For each $m \in M$, since $\mu(m)$ is non-empty, there exists $g \in G$ such that $m \in B_g$. For each $(m, m') \in E^{\mathbf{M}}$, by definition of minor, there exists $(g, g') \in E^{\mathbf{G}}$ with $g \in \mu(m)$ and $g' \in \mu(m')$; it thus holds that $\{m, m'\} \subseteq B_g \cup B_{g'}$. For each $m \in M$, $\{g \mid m \in B_g\}$ is equal to $\mu(m)$, and is hence connected as μ is a minor map. □

We now generalize the notion of tree decomposition to arbitrary graphs, and then compare the resulting notion with the presented notion of $\mathbf{H}$-deconstruction.

When $\mathbf{G}$ and $\mathbf{H}$ are graphs, define an $\mathbf{H}$-*decomposition* of $\mathbf{G}$ to be an H-indexed family $(B_h)_{h\in H}$ of subsets of G satisfying:

– For each pair $(g, g') \in \mathsf{refl}(E^{\mathbf{G}})$, there exists $h \in H$ such that $\{g, g'\} \subseteq B_h$.

– Connectivity (as defined in Definition 3.1)

This is a natural generalization of the definition of *tree decomposition*: a tree decomposition is precisely a $\mathbf{H}$-decomposition where $\mathbf{H}$ is required to be a tree.

Proposition 3.10. *Let $\mathbf{G}$ and $\mathbf{H}$ be graphs, and let $w \geqslant 1$.*

1. *$\mathbf{H}$-decompositions are $\mathbf{H}$-deconstructions.*
2. *If $\mathbf{H}$ is a tree and $\mathbf{G}$ has an $\mathbf{H}$-deconstruction of width $\leqslant w$, then $\mathbf{G}$ has an $\mathbf{H}$-decomposition of width $< 2w$.*

Consequently, when $\mathcal{H}$ is a class of trees, it holds that $\mathcal{G} \preceq \mathcal{H}$ if and only if $\mathcal{G}$ has $\mathcal{H}$-decompositions of bounded width.

Proof. (1) is straightforwardly verified. For (2), suppose that $(B_h)_{h \in H}$ is an $\mathbf{H}$-deconstruction of $\mathbf{G}$ where each bag has size $\leqslant w$. Root the tree $\mathbf{H}$, and define $p : H \to H$ as follows: if h is the root of $\mathbf{H}$, define $p(h) = h$, and otherwise define $p(h)$ to be the parent of h. Define $(C_h)_{h \in H}$ by $C_h = B_h \cup B_{p(h)}$; it is straightforward to verify that $(C_h)_{h \in H}$ is an $\mathbf{H}$-decomposition of $\mathbf{G}$, and each of its bags has size $\leqslant 2w$. □

Remark 3.11. We find that it is cleaner to work with the notion of $\mathbf{H}$-deconstruction than to work with the notion of $\mathbf{H}$-decomposition; this is a primary reason for our focus on the notion of $\mathbf{H}$-deconstruction. Indeed, note that while reflexivity of the $\preceq$ relation is straightforward to prove, we do not know of a simple proof of reflexivity of the analogous relation defined via $\mathbf{H}$-decomposition. Note that while there exists a constant w such that each graph $\mathbf{G}$ has a $\mathbf{G}$-deconstruction of width $\leqslant w$ (in particular, one can take $w = 2$), there does not exist a constant w such that each graph $\mathbf{G}$ has a $\mathbf{G}$-decomposition of width $\leqslant w$: the bags of a $\mathbf{G}$-decomposition of width $\leqslant w$ can cover at most a number of edges that is linear in the number of vertices ($|G|\binom{w+1}{2}$ many), but graphs in general may have quadratically many edges.

3.2 Stack depth

Recall that $\mathcal{T}_d$ denotes the class of all trees of height $\leqslant d$. For $h \geqslant 0$, $k \geqslant 1$, define $\mathbf{T}_{h,k}$ to be the tree with universe $[k]^{\leqslant h}$ and with $E^{\mathbf{T}_{h,k}} = \{(t, ti), (ti, t) \mid t \in [k]^{<h}, i \in [k]\}$. Here, $[k]^{\leqslant h}$ and $[k]^{<h}$ denote the sets of strings over alphabet $[k]$ of length $\leqslant h$ and of length $< h$, respectively. Define the *stack depth* of a class $\mathcal{G}$ of graphs to be

$$\max\{h \mid \forall k \geqslant 1, \mathbf{T}_{h,k} \in \mathsf{minors}(\mathcal{G})\};$$

let it be understood that this maximum is ∞ if the set is infinite.

Proposition 3.12. *A class of graphs has bounded stack depth if and only if it has bounded pathwidth.*

Proof. Let $\mathcal{G}$ be a class of graphs. By Theorem 2.1, $\mathcal{G}$ has unbounded pathwidth if and only if $\mathcal{T} \subseteq \mathsf{minors}(\mathcal{G})$. This obviously implies infinite stack depth. Conversely, infinite stack depth implies $\mathcal{T} \subseteq \mathsf{minors}(\mathcal{G})$: each tree is a subgraph of a tree of the form $\mathbf{T}_{m,m}$, and infinite stack depth gives that $\mathbf{T}_{m,m} \in \mathsf{minors}(\mathcal{G})$ (for each $m \geqslant 1$). □

Theorem 3.13. *Suppose that $d, e \geqslant 0$ are constants and that $\mathcal{G}$ is a class of trees having stack depth d and where each tree has height $\leqslant e$. Then $\mathcal{G} \preceq \mathcal{T}_d$.*

To prove this we shall need some preparations. Let $\mathbf{M}$, $\mathbf{G}$ be rooted trees. Let us say that an $\mathbf{M}$-deconstruction $(\mu(m))_{m \in M}$ of $\mathbf{G}$ is *nice* if the following hold:

- μ is a minor map from $\mathbf{M}$ to $\mathbf{G}$.
- $g_0 \in \mu(m_0)$, where g_0 and m_0 denote the roots of $\mathbf{G}$ and $\mathbf{M}$, respectively.

– If m' is a child of m in $\mathbf{M}$, $g \in \mu(m)$, $g' \in \mu(m')$, and $(g, g') \in E^{\mathbf{G}}$, then g' is a child of g in $\mathbf{G}$.

Lemma 3.14. *Suppose that* $\mathbf{N}$*,* $\mathbf{M}$*, and* $\mathbf{G}$ *are rooted trees, that* $(\nu(n))_{n\in N}$ *is a nice* $\mathbf{N}$*-deconstruction of* $\mathbf{M}$*, and that* $(\mu(m))_{m\in M}$ *is a nice* $\mathbf{M}$*-deconstruction of* $\mathbf{G}$*. Then* $(\mu(\nu(n)))_{n\in N}$ *is a nice* $\mathbf{N}$*-deconstruction of* $\mathbf{G}$*, where here* $\mu(\nu(n))$ *denotes* $\bigcup_{m\in\nu(n)} \mu(m)$*.*

Proof. It is straightforward to verify that $n \mapsto \mu(\nu(n))$ is a minor map from $\mathbf{N}$ to $\mathbf{G}$. It follows from the proof of Lemma 3.8 that $(\mu(\nu(n)))_{n\in N}$ is an $\mathbf{N}$-deconstruction of $\mathbf{G}$. We verify that this deconstruction is nice, as follows. By the niceness of $(\mu(m))_{m\in M}$ and $(\nu(n))_{n\in N}$, we have that $g_0 \in \mu(m_0)$ and $m_0 \in \nu(n_0)$, so $g_0 \in \mu(\nu(n_0))$ (here, g_0, m_0, n_0 denote the roots of $\mathbf{G}$, $\mathbf{M}$, $\mathbf{N}$, respectively). Next, suppose that n' is a child of n in $\mathbf{N}$, that $g \in \mu(\nu(n))$, that $g' \in \mu(\nu(n'))$, and that $(g, g') \in E^{\mathbf{G}}$. There are $m, m' \in M$ such that $g \in \mu(m)$, $m \in \nu(n)$, $g' \in \mu(m')$, and $m' \in \nu(n')$. Since $n' \neq n$, we have $m \neq m'$; as $(g, g') \in E^{\mathbf{G}}$, we then have $(m, m') \in E^{\mathbf{M}}$, and from the niceness of $(\nu(n))_{n\in N}$, we have that m' is a child of m in $\mathbf{M}$. By the niceness of $(\mu(m))_{m\in M}$ then g' is a child of g. ☐

For $d \geqslant 0$ and $k \geqslant 1$, let us say that a node u of a rooted tree has property $P(d, k)$ if either $d = 0$ or $d > 0$ and u has k pairwise incomparable descendents each having property $P(d-1, k)$. (Here, we consider two nodes v, v' to be incomparable if neither is an ancestor of the other.) Let us say that a rooted tree has property $P(d, k)$ if its root has property $P(d, k)$. Observe that such a tree contains $\mathbf{T}_{d,k}$ as a minor.

Lemma 3.15. *Suppose that* $\mathbf{M}$ *and* $\mathbf{G}$ *are rooted trees and that* $(\mu(m))_{m\in M}$ *is a nice* $\mathbf{M}$*-deconstruction of* $\mathbf{G}$*. For* $d \geqslant 0$ *and* $k \geqslant 1$*, if* $\mathbf{M}$ *has property* $P(d, k)$*, then so does* $\mathbf{G}$*.*

Proof. For each $m \in M$, since $\mu(m)$ is connected, it has a unique highest element, by which we mean the element with shortest distance to the root; denote this element by $\mathsf{hi}(\mu(m))$. Suppose that m' is a child of m in $\mathbf{M}$; the parent of $\mathsf{hi}(\mu(m'))$ must, by the definition of deconstruction, lie in $\mu(m'')$ where m'' is adjacent to m'; but it must be that $m'' = m$ by the niceness of $(\mu(m))_{m\in M}$. Thus, if m' is a child of m in $\mathbf{M}$, then $\mathsf{hi}(\mu(m'))$ is a descendent of $\mathsf{hi}(\mu(m))$. It follows by induction on d that, if a node $m \in M$ has property $P(d, k)$ in $\mathbf{M}$, then $\mathsf{hi}(\mu(m))$ has property $P(d, k)$ in $\mathbf{G}$. ☐

Proof of Theorem 3.13. Let $K \geqslant 1$ be a constant. Suppose that $\mathcal{G}$ is a class of rooted trees of height $\leqslant e$ which do not have property $P(d+1, K)$; we prove that $\mathcal{G}$ has nice $\mathcal{T}_d$-deconstructions of bounded width. (This suffices, since the assumption that $\mathcal{G}$ has stack depth d implies that there is a constant $K \geqslant 1$ such that $\mathbf{T}_{d+1,K} \notin \mathsf{minors}(\mathcal{G})$, which in turn implies that the trees in $\mathcal{G}$ do not have property $P(d+1, K)$.)

We proceed by induction on d.

Case $d = 0$*:* We have that the trees in $\mathcal{G}$ do not have property $P(1, K)$. Consider a rooted tree from $\mathcal{G}$. The number of leaves is bounded above by K; since each node is the ancestor of a leaf, the total number of nodes is bounded above by $K(e+1)$. Thus $\mathcal{G}$ has nice $\mathcal{T}_0$-deconstructions of width $\leqslant K(e+1)$.

Case $d > 0$: We argue by induction on e. If $e \leqslant d$, then we have that $\mathcal{G} \subseteq \mathcal{T}_d$ and we are done (for each $\mathbf{G} \in \mathcal{G}$, use the $\mathbf{G}$-deconstruction of $\mathbf{G}$ discussed in conjunction with reflexivity in Proposition 3.7). So suppose that $e > d$.

Define a class of trees $\mathcal{G}'$ as follows: for each tree $\mathbf{G}$ in $\mathcal{G}$, and for each child c of the root of $\mathbf{G}$, if c does not have property $P(d, K)$, then place the subtree of $\mathbf{G}$ rooted at c in $\mathcal{G}'$. The trees in $\mathcal{G}'$ have bounded height and do not have property $P(d, K)$; so, by induction, there is a constant w such that $\mathcal{G}'$ has nice $\mathcal{T}_{d-1}$-deconstructions of width $\leqslant w$.

Let $\mathbf{G}$ be a tree in $\mathcal{G}$. Let $b^1, \ldots, b^L$ denote the children of the root g_0 that have property $P(d, K)$, and let $c^1, \ldots, c^Q$ denote the remaining children of the root. Since the root of $\mathbf{G}$ does not have property $P(d+1, K)$, we have that $L < K$. Let $\mathbf{G}^1, \ldots, \mathbf{G}^Q$ denote the subtrees of $\mathbf{G}$ rooted at $c^1, \ldots, c^Q$, respectively. Each $\mathbf{G}^i$ is in $\mathcal{G}'$, so for each $i \in [Q]$, there is a tree $\mathbf{T}^i \in \mathcal{T}_{d-1}$ such that $\mathbf{G}^i$ has a nice $\mathbf{T}^i$-deconstruction $(\mu^i(t))_{t \in T^i}$ of width $\leqslant w$.

Now define the tree $\mathbf{H}$ to be the minor of $\mathbf{G}$ obtained from $\mathbf{G}$ by contracting together the vertices $\{g_0, b^1, \ldots, b^L\}$ to obtain h_0, and by replacing each $\mathbf{G}^i$ with $\mathbf{T}^i$. Observe that the height of $\mathbf{H}$ is $\leqslant e - 1$. The following map μ is a minor map from $\mathbf{H}$ to $\mathbf{G}$: $\mu(h_0) = \{g_0, b^1, \ldots, b^L\}$, $\mu(t)$ is equal to $\mu^i(t)$ if $t \in T^i$, and $\mu(h) = \{h\}$ for all other vertices $h \in H$. It is straightforward to verify that $(\mu(h))_{h \in H}$ gives a nice $\mathbf{H}$-deconstruction of $\mathbf{G}$ having width $\leqslant \max(K, w)$.

Let $\mathcal{H}$ denote the class of all trees $\mathbf{H}$ obtained from $\mathbf{G} \in \mathcal{G}$ in this way. We just saw that $\mathcal{G}$ has nice $\mathcal{H}$-deconstructions of bounded width. Since $\mathcal{H}$ has height $\leqslant e - 1$ and does not have property $P(d+1, K)$ by Lemma 3.15, by induction, $\mathcal{H}$ has nice $\mathcal{T}_d$-deconstructions of bounded width. As a consequence of Lemma 3.14, we obtain that $\mathcal{G}$ has nice $\mathcal{T}_d$-deconstructions of bounded width. $\square$

3.3 Hierarchy

Recall that $\mathcal{L}, \mathcal{T}, \mathcal{P}$ denote the classes of graphs, trees and paths respectively, and $\mathcal{T}_d, \mathcal{F}_d$ denote the classes of trees respectively forests of height at most d.

Theorem 3.16 (Graph hierarchy theorem)**.** *The hierarchy*

$$\mathcal{T}_0 \precnsim \mathcal{F}_0 \precnsim \mathcal{T}_1 \precnsim \mathcal{F}_1 \precnsim \cdots \precnsim \mathcal{P} \precnsim \mathcal{T} \precnsim \mathcal{L} \qquad (*)$$

presents correct relationships, and is comprehensive in that each class of graphs is equivalent (under $\equiv$) to one of the classes therein.

We break the proof into several lemmas. We begin by observing that the established conditions of bounded treewidth, bounded pathwidth, and bounded tree depth can be formulated using the $\preceq$ relation and the defined graph classes.

Proposition 3.17. *Let $\mathcal{G}$ be a class of graphs.*

1. *$\mathcal{G}$ has bounded treewidth if and only if $\mathcal{G} \preceq \mathcal{T}$.*
2. *$\mathcal{G}$ has bounded pathwidth if and only if $\mathcal{G} \preceq \mathcal{P}$.*

3. $\mathcal{G}$ *has bounded tree depth if and only if there exists* $d \geqslant 0$ *such that* $\mathcal{G} \preceq \mathcal{F}_d$.

Proof. The first two claims follow immediately from Proposition 3.10. For the third claim, we reason as follows. For the forward direction, let d be an upper bound on the tree depth of $\mathcal{G}$, and let $\mathbf{G}$ be a graph in $\mathcal{G}$; Proposition 2.2 (along with Proposition 3.10) implies that each component of $\mathbf{G}$ has a $\mathcal{T}_d$-deconstruction of width $\leqslant d$. For the backward direction, Proposition 3.10 gives a constant w such that $\mathcal{G}$ has $\mathcal{F}_d$-decompositions of width $< w$, which implies that $\mathcal{G}$ has tree depth $\leqslant wd$ (see [5, Remark 4.3(a)]). □

In the next two lemmas, we present the negative results needed to give the hierarchy, showing that various pairs of graph classes are not related by $\preceq$.

Lemma 3.18. *Let* $\mathcal{G}$ *be a class of graphs.*

1. *If* $\mathcal{L} \not\preceq \mathcal{G}$, *then* $\mathcal{G} \preceq \mathcal{T}$.
2. *If* $\mathcal{T} \not\preceq \mathcal{G}$, *then* $\mathcal{G} \preceq \mathcal{P}$.
3. *If* $\mathcal{P} \not\preceq \mathcal{G}$, *then* $\mathcal{G}$ *has bounded tree depth.*

Proof. To show (3), assume $\mathcal{P} \not\preceq \mathcal{G}$. By Proposition 3.9, $\mathcal{P} \not\preceq \mathsf{minors}(\mathcal{G})$, so $\mathcal{P} \not\subseteq \mathsf{minors}(\mathcal{G})$, and hence $\mathcal{G}$ has bounded tree depth by Theorem 2.1.

For (2) we reason analogously: if $\mathcal{T} \not\preceq \mathcal{G}$, then $\mathcal{G}$ has bounded pathwidth by Proposition 3.9 and Theorem 2.1 and hence $\mathcal{G} \preceq \mathcal{P}$ by Proposition 3.17.

Also (1) is proved analoguously, but in order to use Theorem 2.1, we need to prove that $\mathcal{L} \preceq \mathcal{R}$, where $\mathcal{R}$ denotes the class of all grids; then $\mathcal{L} \not\preceq \mathcal{G}$ implies $\mathcal{R} \not\preceq \mathcal{G}$. That $\mathcal{L} \preceq \mathcal{R}$ follows from Example 3.3: each graph has $\mathcal{R}$-deconstructions of width $\leqslant 2$. □

Lemma 3.19. *The following relationships hold.*

1. $\mathcal{L} \not\preceq \mathcal{T}$.
2. $\mathcal{T} \not\preceq \mathcal{P}$.
3. $\mathcal{P} \not\preceq \mathcal{F}_d$, *for all* $d \geqslant 0$.

Proof. Immediate from Theorem 2.1 and Proposition 3.17. □

Lemma 3.20. *For each* $d \geqslant 0$, *the following hold.*

1. $\mathcal{F}_d \not\preceq \mathcal{T}_d$.
2. $\mathcal{T}_{d+1} \not\preceq \mathcal{F}_d$.

Proof. We prove this by induction on d. When $d = 0$, the claim that $\mathcal{F}_d \not\preceq \mathcal{T}_d$ is clear, since $\mathcal{T}_0$ contains only one-vertex trees, but $\mathcal{F}_0$ contains graphs with arbitrarily many vertices. The claim that $\mathcal{T}_{d+1} \not\preceq \mathcal{F}_d$ always follows from $\mathcal{F}_d \not\preceq \mathcal{T}_d$, as follows: the assumption $\mathcal{T}_{d+1} \preceq \mathcal{F}_d$ implies $\mathcal{T}_{d+1} \preceq \mathcal{T}_d$ by Proposition 3.5, and this implies $\mathcal{F}_d \preceq \mathcal{T}_d$.

Let $d > 0$; we need to prove that $\mathcal{F}_d \not\preceq \mathcal{T}_d$. Suppose for a contradiction that w is a constant such that $\mathcal{F}_d$ has $\mathcal{T}_d$-deconstructions of width $\leqslant w$. We show that $\mathcal{T}_d \preceq \mathcal{F}_{d-1}$, which contradicts the induction hypothesis. Given an arbitrary tree $\mathbf{T}$ in $\mathcal{T}_d$, create $w+1$ copies of it; the resulting graph has an $\mathbf{H}$-deconstruction $(B_h)_{h \in H}$ of width $\leqslant w$, with $\mathbf{H} \in \mathcal{T}_d$. Root $\mathbf{H}$ with a vertex h_0 to witness that its height is $\leqslant d$. Since the bag B_{h_0} has size $\leqslant w$, there must be a copy of $\mathbf{T}$ that is disjoint from B_{h_0}; call this the *key copy* and denote its universe by K. By Proposition 3.5, it is possible to take a subtree $\mathbf{H}'$ of $\mathbf{H}$ that excludes h_0 such that $(B_h \cap K)_{h \in H'}$ gives an $\mathbf{H}'$-deconstruction of the key copy. The graph $\mathbf{H}'$ has height $\leqslant d-1$. We thus showed that $\mathcal{T}_d$ has $\mathcal{T}_{d-1}$-deconstructions of width $\leqslant w$, implying that $\mathcal{T}_d \preceq \mathcal{F}_{d-1}$, as desired. □

Lemma 3.21. *Let $\mathcal{G}$ be a class of graphs. If $\mathcal{G}$ has bounded tree depth, then there exists a class $\mathcal{H}$ of forests of bounded height such that $\mathcal{G} \equiv \mathcal{H}$ and such that $\mathcal{G}$ and $\mathcal{H}$ have the same stack depth. (Namely, one can take $\mathcal{H}$ to be the class of all forests in* $\mathsf{minors}(\mathcal{G})$*.)*

Proof. Let $\mathcal{H}$ be as described. The class $\mathcal{H}$ has bounded height, as $\mathcal{G}$ has bounded tree depth. The classes $\mathcal{G}$ and $\mathcal{H}$ have the same tree minors, so they have the same stack depth. We have $\mathcal{H} \preceq \mathcal{G}$ by Proposition 3.9.

The proof of $\mathcal{G} \preceq \mathcal{H}$ is along the lines of that of [5, Lemma 4.8]. We provide it here for completeness. For each graph $\mathbf{G} \in \mathcal{G}$, let $\mathbf{H}$ be a forest that contains a depth-first search tree of each component of $\mathbf{G}$ (for precise details, we refer to Diestel [18], who speaks of *normal spanning trees*; see Proposition 1.5.6). Since $\mathbf{H}$ is a subgraph of $\mathbf{G}$, we have $\mathbf{H} \in \mathcal{H}$. Each component of $\mathbf{G}$ is contained in the closure of its corresponding component in $\mathbf{H}$ (in the sense of the definition of tree depth), so Proposition 2.2 and the fact that $\mathcal{H}$ has bounded height imply that $\mathcal{G} \preceq \mathcal{H}$. □

Lemma 3.22. *Let $\mathcal{G}$ be a class of forests having bounded height, and let d denote the stack depth of $\mathcal{G}$. It holds either that $\mathcal{G} \equiv \mathcal{T}_d$ or that $\mathcal{G} \equiv \mathcal{F}_d$.*

Proof. By Lemma 3.20, not both $\mathcal{G} \equiv \mathcal{T}_d$ and $\mathcal{G} \equiv \mathcal{F}_d$ can hold. Let $\mathcal{C}$ be the class of connected graphs that appear as components of graphs in $\mathcal{G}$; note that d is the stack depth of $\mathcal{C}$. By Theorem 3.13 we have $\mathcal{C} \preceq \mathcal{T}_d$ and hence $\mathcal{G} \preceq \mathcal{F}_d$

For each $k \geqslant 1$ and each graph $\mathbf{G} \in \mathcal{G}$, define $\mathbf{G}(k)$ to be the number of components of $\mathbf{G}$ having the tree $\mathbf{T}_{d,k}$ as a minor. We consider two cases.

Case 1: Suppose that, for all $k \geqslant 1$, the set $\{\mathbf{G}(k) \mid \mathbf{G} \in \mathcal{G}\}$ has infinite size. We claim that $\mathcal{G} \equiv \mathcal{F}_d$. We have to show that $\mathcal{F}_d \preceq \mathcal{G}$. For each $k \geqslant 1$, let us use $k \times \mathbf{T}_{d,k}$ to denote the graph consisting of k disjoint copies of $\mathbf{T}_{d,k}$. Then $\{k \times \mathbf{T}_{d,k} \mid k \geqslant 1\} \subseteq \mathsf{minors}(\mathcal{G})$ by assumption. Each graph in $\mathcal{F}_d$ is isomorphic to a subgraph of a graph of the form $k \times \mathbf{T}_{d,k}$. Hence, by Proposition 3.9, it holds that $\mathcal{F}_d \preceq \mathsf{minors}(\mathcal{G}) \equiv \mathcal{G}$.

Case 2: When the assumption of the first case does not hold, one can choose a sufficiently large $K \geqslant 1$ such that, for all $\mathbf{G} \in \mathcal{G}$, it holds that $\mathbf{G}(K) \leqslant K$. We claim that $\mathcal{G} \equiv \mathcal{T}_d$. That $\mathcal{T}_d \preceq \mathcal{G}$ follows from the hypothesis that d is the stack depth of $\mathcal{C}$. We show that $\mathcal{G} \preceq \mathcal{T}_d$. Let $\mathcal{H}$ be the subset of $\mathcal{C}$ that contains a graph $\mathbf{H} \in \mathcal{C}$ if and only if $\mathbf{T}_{d,K}$ is not a minor of $\mathbf{H}$. By Theorem 3.13, it holds that $\mathcal{H} \preceq \mathcal{T}_{d-1}$; let $w \geqslant 1$ be such that $\mathcal{H}$ has $\mathcal{T}_{d-1}$-deconstructions of width $\leqslant w$. Let $\mathbf{G} \in \mathcal{G}$; let $\mathbf{G}_1, \ldots, \mathbf{G}_L$ be the components of $\mathbf{G}$ having

$\mathbf{T}_{d,K}$ as a minor, and let $\mathbf{H}_1, \ldots, \mathbf{H}_M$ be the other components of $\mathbf{G}$. By the choice of K, it holds that $L \leqslant K$. Since $\mathcal{C} \preceq \mathcal{T}_d$, there exists $v \geqslant 1$ such that $\mathcal{C}$ has $\mathcal{T}_d$-deconstructions of width $\leqslant v$. Let $\mathbf{T} \in \mathcal{T}_d$ be a sufficiently large tree so that each $\mathbf{G}_i$ has a $\mathbf{T}$-deconstruction $(B_t^i)_{t \in T}$ of width $\leqslant v$; then, the disjoint union of the $\mathbf{G}_i$ has a $\mathbf{T}$-deconstruction of width $\leqslant vL$, namely, $(B_t^1 \cup \cdots \cup B_t^L)_{t \in T}$. Each $\mathbf{H}_j$ has a $\mathbf{T}_j$-deconstruction of width $\leqslant w$, where $\mathbf{T}_j \in \mathcal{T}_{d-1}$. Let $\mathbf{T}'$ be equal to $\mathbf{T}$ but augmented so that the root of each $\mathbf{T}_j$ is a child of the root of $\mathbf{T}$; we have that the height of $\mathbf{T}'$ is d. The graph $\mathbf{G}$ has a $\mathbf{T}'$-deconstruction where each bag is defined as it was in the respective $\mathbf{T}$-deconstruction or $\mathbf{T}_j$-deconstruction; this $\mathbf{T}'$-deconstruction has width $\leqslant \max(vL, w)$. ☐

The following is a consequence of the previous two lemmas.

Lemma 3.23. *Let $\mathcal{G}$ be a class of graphs having bounded tree depth, and let $d \geqslant 0$. The class $\mathcal{G}$ has stack depth d if and only if $\mathcal{G} \equiv \mathcal{T}_d$ or $\mathcal{G} \equiv \mathcal{F}_d$.*

Proof. We first prove the forward direction. Suppose that $\mathcal{G}$ has stack depth d. By Lemma 3.21, there exists a class of forests $\mathcal{H}$ having bounded height and stack depth d such that $\mathcal{G} \equiv \mathcal{H}$. By Lemma 3.22, $\mathcal{H} \equiv \mathcal{T}_d$ or $\mathcal{H} \equiv \mathcal{F}_d$, implying $\mathcal{G} \equiv \mathcal{T}_d$ or $\mathcal{G} \equiv \mathcal{F}_d$.

Conversely, suppose $\mathcal{G}$ does not have stack depth d. By bounded tree depth and Proposition 3.12, $\mathcal{G}$ has stack depth $d' \in \mathbb{N}$ for some $d' \neq d$. By the forward direction $\mathcal{G} \equiv \mathcal{T}_{d'}$ or $\mathcal{G} \equiv \mathcal{F}_{d'}$. In both cases, $\mathcal{G} \not\equiv \mathcal{T}_d$ and $\mathcal{G} \not\equiv \mathcal{F}_d$ by Lemma 3.20. ☐

Proof of Theorem 3.16. It is clear that

$$\mathcal{T}_0 \preceq \mathcal{F}_0 \preceq \mathcal{T}_1 \preceq \mathcal{F}_1 \preceq \cdots \preceq \mathcal{P} \preceq \mathcal{T} \preceq \mathcal{L}.$$

Lemmas 3.19 and 3.20 imply that none of the displayed $\preceq$ can be reversed. To prove that the hierarchy is comprehensive, let $\mathcal{G}$ be an arbitrary class of graphs. If $\mathcal{L} \preceq \mathcal{G}$, then clearly $\mathcal{G} \equiv \mathcal{L}$ and we are done. Otherwise $\mathcal{G} \preceq \mathcal{T}$ by Lemma 3.18 (1). If $\mathcal{T} \preceq \mathcal{G}$, then $\mathcal{G} \equiv \mathcal{T}$ and we are done. Otherwise, $\mathcal{G} \preceq \mathcal{P}$ by Lemma 3.18 (2). If $\mathcal{P} \preceq \mathcal{G}$, then $\mathcal{G} \equiv \mathcal{P}$ and we are done. Otherwise $\mathcal{G}$ has bounded tree depth by Lemma 3.18 (3). By Proposition 3.12 there is $d \in \mathbb{N}$ such that $\mathcal{G}$ has stack depth d. Then $\mathcal{G} \equiv \mathcal{T}_d$ or $\mathcal{G} \equiv \mathcal{F}_d$ by Lemma 3.23. ☐

4 Grohe's theorem

In this section, we use the notion of graph deconstruction to give a novel proof of Grohe's theorem, which establishes the hardness of the homomorphism problem on any class of structures whose cores have unbounded treewidth. We believe that our proof constitutes a modular, relatively transparent, and relatively simple alternative to the original proof [27]. Other than the definition of graph deconstruction, the only element needed from the previous section is the fact that, if a graph class $\mathcal{G}$ has unbounded treewidth, then $\mathcal{L} \preceq \mathcal{G}$ (this follows from Proposition 3.17 and Lemma 3.18.)

For the sake of brevity and because it is unnecessary for our purposes here, we do not introduce here a full framework for parameterized complexity. (We do carry this out in the next section, where we in particular introduce our notion of quantifier-free reduction.) We introduce the following definitions to be used in the scope of this section. For each class $\mathcal{A}$ of structures, define p-HOM($\mathcal{A}$) to be the problem whose instances are pairs $(\mathbf{A}, \mathbf{B})$ of similar structures where $\mathbf{A} \in \mathcal{A}$, and the question is to decide whether or not $\mathbf{A} \xrightarrow{h} \mathbf{B}$. We consider such a problem p-HOM($\mathcal{A}$) to be *tractable* if there exists a computable function f and a polynomial-time algorithm g such that, on each instance $(\mathbf{A}, \mathbf{B})$ of p-HOM($\mathcal{A}$), g run on input $(f(\mathbf{A}), \mathbf{B})$ decides if $\mathbf{A} \xrightarrow{h} \mathbf{B}$. It can be recognized that this definition is equivalent to that of fixed-parameter tractability, where the structure $\mathbf{A}$ is taken to be the parameter; see the characterization of fixed-parameter tractability in terms of precomputation on the parameter [24, Theorem 1.37]. It is well-known that the tractability of p-HOM($\mathcal{L}^*$) is equivalent to the complexity class collapse W[1] = FPT.

We prove the following formulation of Grohe's theorem [27].

Theorem 4.1. *Assume that $\mathcal{A}$ is a computably enumerable class of structures having bounded arity. If the graphs of the cores of $\mathcal{A}$ have unbounded treewidth, then the problem* p-HOM($\mathcal{A}$) *is not tractable, unless* p-HOM($\mathcal{L}^*$) *is as well.*

At the heart of our proof are three polynomial-time reductions, presented in the following three lemmas. In each case, we describe the output of the claimed polynomial-time algorithm; it is readily verified that the output can be produced in polynomial time. The second and third lemmas are based on results that appeared in previous work [11].

Lemma 4.2. *For each $k \geqslant 1$, there exists a polynomial-time algorithm that, given graphs $\mathbf{G}$ and $\mathbf{H}$, a $\mathbf{H}$-deconstruction $(B_h)_{h \in H}$ of $\mathbf{G}$ of width $\leqslant k$, and a structure $\mathbf{D}$ similar to $\mathbf{G}^*$, outputs a structure $\mathbf{D}'$ such that $\mathbf{G}^* \xrightarrow{h} \mathbf{D}$ iff $\mathbf{H}^* \xrightarrow{h} \mathbf{D}'$*

Proof. The structure $\mathbf{D}'$ is defined as follows. Its universe D' is the set of all partial homomorphisms f from $\mathbf{G}^*$ to $\mathbf{D}$ with $|\mathsf{dom}(f)| \leqslant k$. The relation $E^{\mathbf{D}'}$ is defined as the set of pairs $(f, f') \in D' \times D'$ such that $f \cup f'$ (as a set of ordered pairs) is a partial homomorphism from $\mathbf{G}^*$ to $\mathbf{D}$. Each relation $C_h^{\mathbf{D}'}$ is defined as $\{f \in D' \mid \mathsf{dom}(f) = B_h\}$.

Suppose that e is a homomorphism from $\mathbf{G}^*$ to $\mathbf{D}$. Then the mapping $e' : H \to D'$ defined by $e'(h) = e \upharpoonright B_h$ (the restriction of e to B_h) is a homomorphism from $\mathbf{H}^*$ to $\mathbf{D}'$.

Suppose that e' is a homomorphism from $\mathbf{H}^*$ to $\mathbf{D}'$. We define a map $e : G \to D$ as follows. For each $g \in G$, by the connectivity condition (Definition 3.1) and the definition of $E^{\mathbf{D}'}$, all maps of the form $e'(h)$ with $g \in \mathsf{dom}(e'(h))$ send g to the same value. Define $e(g)$ to be that value; we have that $e'(h) \subseteq e$. To verify that e is a homomorphism from $\mathbf{G}^*$ to $\mathbf{D}$, since each relation of $\mathbf{G}^*$ has arity 1 or is the relation $E^{\mathbf{G}}$, it suffices to argue that for any pair $(g, g') \in \mathsf{refl}(E^{\mathbf{G}})$, the map $e \upharpoonright \{g, g'\}$ is a partial homomorphism from $\mathbf{G}^*$ to $\mathbf{D}$. Let (g, g') be such a pair; by the coverage condition (Definition 3.1) there exists $(h, h') \in \mathsf{refl}(E^{\mathbf{H}})$ such that $\{g, g'\} \subseteq B_h \cup B_{h'}$. We have that $e'(h) \cup e'(h')$ is a partial homomorphism from $\mathbf{G}$ to $\mathbf{D}$ (this is clear if $h = h'$; if $h \neq h'$, this follows from from the definition of $E^{\mathbf{D}'}$). This concludes the proof, as $e'(h) \cup e'(h') \subseteq e$. $\square$

Lemma 4.3. *For each $r \geqslant 1$, there exists a polynomial-time algorithm that, given a structure $\mathbf{A}$ whose relations have arity $\leqslant r$ and a structure $\mathbf{D}$ similar to $\mathsf{graph}(\mathbf{A})^*$, outputs a structure $\mathbf{D}'$ such that $\mathsf{graph}(\mathbf{A})^* \xrightarrow{h} \mathbf{D}$ iff $\mathbf{A}^* \xrightarrow{h} \mathbf{D}'$.*

Proof. The structure $\mathbf{D}'$ has universe $D' = A \times D$. Each relation $C_a^{\mathbf{D}'}$ is defined as $\{a\} \times C_a^{\mathbf{D}}$, and for each relation symbol R of the structure of $\mathbf{A}$, define $R^{\mathbf{D}'}$ to be the set of all k-tuples $((a_1, d_1), \ldots, (a_k, d_k))$ on D' such that $k = ar(R)$ and for all $i, j \in [k]$, it holds that $(a_i, a_j) \in E^{\mathsf{graph}(\mathbf{A})^*}$ implies $(d_i, d_j) \in E^{\mathbf{D}}$.

Suppose that e is a homomorphism from $\mathsf{graph}(\mathbf{A}^*)$ to $\mathbf{D}$. Then it is straightforward to verify that $e' : A \to D'$ defined by $e'(a) = (a, e(a))$ is a homomorphism from $\mathbf{A}^*$ to $\mathbf{D}'$.

Suppose that e' is a homomorphism from $\mathbf{A}^*$ to $\mathbf{D}'$. By the definition of the relations $C_a^{\mathbf{D}'}$, each element a is mapped by e to an element of the form (a, d) with $d \in C_a^{\mathbf{D}}$. Define $e : A \to D$ so that, for each $a \in A$, it holds that $e'(a) = (a, e(a))$. We have that e is a homomorphism from $\mathsf{graph}(\mathbf{A})^*$ to $\mathbf{D}$: when $(a, a') \in E^{\mathsf{graph}(\mathbf{A})^*}$, there exists a tuple $(a_1, \ldots, a_k)$ in a relation $R^{\mathbf{A}^*}$ of $\mathbf{A}^*$ with a and a' among its entries, so $(e(a), e(a')) \in E^{\mathbf{D}}$ follows from the fact that e' is a homomorphism and the definition of $R^{\mathbf{D}'}$. ☐

Lemma 4.4. *There exists a polynomial-time algorithm that, given a core $\mathbf{A}$ and a structure $\mathbf{D}$ similar to $\mathbf{A}^*$, outputs a structure $\mathbf{D}'$ such that $\mathbf{A}^* \xrightarrow{h} \mathbf{D}$ iff $\mathbf{A} \xrightarrow{h} \mathbf{D}'$.*

Proof. Define $\mathbf{D}'$ as the structure $\langle\{(a, d) \in A \times D \mid d \in C_a^{\mathbf{D}}\}\rangle^{\mathbf{A} \times \mathbf{D}_\sigma}$, where $\mathbf{D}_\sigma$ denotes the restriction of $\mathbf{D}$ to the vocabulary σ of $\mathbf{A}$. (If the specified set of pairs is empty, the algorithm outputs a fixed *no* instance.)

Suppose that e is a homomorphism from $\mathbf{A}^*$ to $\mathbf{D}$; then, the map $e' : A \to D'$ defined by $e'(a) = (a, e(a))$ is straightforwardly verified to be a homomorphism from $\mathbf{A}$ to $\mathbf{D}'$.

Suppose that e' is a homomorphism from $\mathbf{A}$ to $\mathbf{D}'$. Here, for any homomorphism g' from $\mathbf{A}$ to $\mathbf{D}'$, we let $g'_1 : A \to A$ and $g'_2 : A \to D$ denote the maps such that $g'(a) = (g'_1(a), g'_2(a))$ for each $a \in A$. We have that e'_1 is a homomorphism from $\mathbf{A}$ to itself; since $\mathbf{A}$ is a core, e'_1 is a bijection. It follows that each finite power of e'_1, and hence e'^{-1}_1, is a homomorphism from $\mathbf{A}$ to itself. Composing e'^{-1}_1 with e', we obtain a homomorphism f' from $\mathbf{A}$ to $\mathbf{D}'$ such that f'_1 is the identity map on A. By definition of $\mathbf{D}'$, for each $a \in A$ it holds that $f'_2(a) \in C_a^{\mathbf{D}}$, and for each $R \in \sigma$ and each tuple $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$, it holds that $(f_2(a_1), \ldots, f_2(a_k)) \in R^{\mathbf{D}_\sigma}$ by definition of $\mathbf{D}'$. Hence f'_2 is a homomorphism from $\mathbf{A}^*$ to $\mathbf{D}$. ☐

Proof of Theorem 4.1. Assume that the problem p-Hom($\mathcal{A}$) is tractable via (f', g'); we show that the problem p-Hom($\mathcal{L}^*$) is tractable. Let $r \geqslant 1$ be a bound on the arity of $\mathcal{A}$. As noted at the beginning of the section, our hypothesis on $\mathcal{A}$ implies that there exists $k \geqslant 1$ such that, for each graph $\mathbf{G}$, there exists a structure $\mathbf{A} \in \mathcal{A}$ that *corresponds* to $\mathbf{G}$, by which we mean that the core $\mathbf{C}$ of $\mathbf{A}$ has a graph $\mathbf{H} = \mathsf{graph}(\mathbf{C})$ such that $\mathbf{G}$ has a $\mathbf{H}$-deconstruction $(B_h)_{h \in H}$ of width $\leqslant k$.

The following pair (f, g) establishes the tractability of p-Hom($\mathcal{L}^*$). Given an instance $(\mathbf{G}^*, \mathbf{B})$ thereof, the algorithm first performs a computation depending only on $\mathbf{G}^*$. In particular, it enumerates the structures in $\mathcal{A}$ until it finds a structure $\mathbf{A} \in \mathcal{A}$ that corresponds to $\mathbf{G}$; it outputs the core $\mathbf{C}$ of $\mathbf{A}$, the core's graph $\mathbf{H} = \mathsf{graph}(\mathbf{C})$, and the

$\mathbf{H}$-deconstruction $(B_h)_{h\in H}$ of $\mathbf{G}$ having width $\leqslant k$. All of this information plus the value of $f'(\mathbf{A})$ is the output of $f(\mathbf{G}^*)$. Then, g is defined to be the polynomial-time algorithm that invokes the algorithm of Lemma 4.2 on $(\mathbf{G}^*, \mathbf{D})$ to obtain an instance $(\mathbf{H}^*, \mathbf{D}')$; invokes the algorithm of Lemma 4.3 on $(\mathbf{H}^*, \mathbf{D}')$ to obtain an instance $(\mathbf{C}^*, \mathbf{D}'')$; and, then invokes the algorithm of Lemma 4.4 on $(\mathbf{C}^*, \mathbf{D}'')$ to obtain an instance $(\mathbf{C}, \mathbf{D}''')$. We have $\mathbf{C} \xrightarrow{h} \mathbf{D}'''$ if and only if $\mathbf{A} \xrightarrow{h} \mathbf{D}'''$. Thus, the algorithm invokes the algorithm g' on $(f'(\mathbf{A}), \mathbf{D}''')$ and outputs the answer of this invocation. □

5 Complexity classification

In this section we study the complexity of the parameterized homomorphism problems associated to classes of structures $\mathcal{A}$:

$$p\text{-}\textsc{Hom}(\mathcal{A}) := \big\{(\mathbf{B}, \ulcorner\mathbf{A}\urcorner) \mid \mathbf{A} \in \mathcal{A} \;\&\; \mathbf{A} \xrightarrow{h} \mathbf{B}\big\}.$$

Here, $\ulcorner\mathbf{A}\urcorner$ is a natural number coding the structure $\mathbf{A}$ in some natural way. The goal of this section is to show that the complexities of homomorphism problems are captured in a strong sense by the hierarchy from Section 3, namely with respect to a computationally very weak notion of reduction which we call quantifier-free after a pre-computation (*qfap*).

We recall some basic notions from parameterized complexity theory in the next subsection; define qfap-reductions in Section 5.2; and consider the homomorphism problem for graph classes and subsequently for general classes of structures in Sections 5.3 and 5.4.

5.1 Parameterized and descriptive complexity

A *parameterized problem* Q is a subset of $\{0,1\}^* \times \mathbb{N}$. By a *classical problem* we mean a subset of $\{0,1\}^*$. Given an instance (x,k) of Q we refer to k as its *parameter*. The *kth slice* of Q is the classical problem $\{x \in \{0,1\}^* \mid (x,k) \in Q\}$.

Following [23], we say Q is in L *after a pre-computation* if there is a computable function $a : \mathbb{N} \to \{0,1\}^*$ and a classical problem $P \subseteq \{0,1\}^*$ in L such that for all $(x,k) \in \{0,1\}^* \times \mathbb{N}$

$$(x,k) \in Q \iff \langle x, a(k)\rangle \in P,$$

where $\langle\cdot,\cdot\rangle$ is some standard pairing function for binary strings. Equivalently, this means that $(x,k) \overset{?}{\in} Q$ is decidable in space $O(f(k) + \log n)$ for some computable $f : \mathbb{N} \to \mathbb{N}$. The class of such problems is denoted *para-L*. This mode of speech makes sense not only for L but for any classical complexity class, and we refer to [23] for the corresponding theory. For example, FPT is the class of parameterized problems which are in P after a pre-computation. A *parameterized reduction* from a parameterized problem Q to another Q' is a function $r : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^* \times \mathbb{N}$ such that there is a computable $g : \mathbb{N} \to \mathbb{N}$ such that for all $(x,k) \in \{0,1\}^* \times \mathbb{N}$ we have for $(x',k') := r((x,k))$ that $k' \leqslant g(k)$ and: $(x,k) \in Q \iff (x',k') \in Q'$. If there is a computable f such that $r((x,k))$ is computable in space $O(f(k) + \log|x|)$ (on a Turing machine with write-only output tape), then we speak of a *pl-reduction*.

In descriptive complexity one considers classical problems as isomorphism closed classes of (finite) structures of some fixed vocabulary. In the parameterized setting we are led to consider the slices of parameterized problems as such classes of structures.

Definition 5.1. A *parameterized problem* is a subset $Q \subseteq \mathrm{STR} \times \mathbb{N}$ such that for every $k \in \mathbb{N}$ there is a vocabulary τ_k such that the *k-th slice of* Q, i.e. $\{\mathbf{A} \mid (\mathbf{A}, k) \in Q\}$, is an isomorphism closed class of τ_k-structures.[1] If there is $r \in \mathbb{N}$ such that $\mathrm{ar}(R) \leqslant r$ for all $R \in \bigcup_k \tau_k$, we say that Q *has bounded arity.*

This definition is not in conflict with the mode of speech above if one views binary strings as structures in the usual way. Flum and Grohe [22] transferred capturing results (cf. [20, Chapter 7]) of classical descriptive complexity to the parameterized setting via the concept of *slicewise definability.* Many parameterized classes could be characterized this way [23, 13]. For example, a parameterized problem Q is *slicewise FO-definable* if there exists a computable function d mapping every $k \in \mathbb{N}$ to a first-order sentence $d(k)$ defining the k-th slice of Q (cf. [22]).

5.2 Reductions that are quantifier-free after a pre-computation

Central to descriptive complexity are first-order reductions which take a structure $\mathbf{A}$ to the structure $I(\mathbf{A})$ where I is a first-order interpretation (see e.g. [20, Chapter 12.3]). We recall the definition (see e.g. [20, Chapter 11.2]).

Definition 5.2. Let σ, τ be (finite, relational) vocabularies and U be a unary relation symbol outside τ. An *interpretation (of τ in σ)* is a sequence $I = (\varphi_R)_{R \in \tau \dot\cup \{U,=\}}$ of σ-formulas such that there exists $w \in \mathbb{N}$ such that for all $R \in \tau \dot\cup \{U\}$ we have $\varphi_R = \varphi_R(\bar{x}_1, \ldots, \bar{x}_{\mathrm{ar}(R)})$ and $\varphi_= = \varphi_=(\bar{x}_1, \bar{x}_2)$ where every $\bar{x}_i$ is a tuple of w variables. The number w is the *dimension* of I. The vocabularies σ and τ are the *input* and *output vocabulary of* I, respectively. An interpretation is *quantifier-free* if all its formulas are. An interpretation I determines the partial function from $\mathrm{STR}[\sigma]$ into $\mathrm{STR}[\tau]$ which maps a σ-structure $\mathbf{A}$ to a τ-structure $\mathbf{B}$ if there exists a surjection $f : \varphi_U(\mathbf{A}) \to B$ such that for all $R \in \tau$ and all $\bar{a}_1, \bar{a}_2, \ldots \in \varphi_U(\mathbf{A})$:

$$\mathbf{A} \models \varphi_=(\bar{a}_1, \bar{a}_2) \iff f(\bar{a}_1) = f(\bar{a}_2);$$

$$\mathbf{A} \models \varphi_R(\bar{a}_1, \ldots, \bar{a}_{\mathrm{ar}(R)}) \iff f(\bar{a}_1) \cdots f(\bar{a}_{\mathrm{ar}(R)}) \in R^{\mathbf{B}};$$

such a $\mathbf{B}$, if it exists, is unique up to isomorphism; if no such $\mathbf{B}$ exists, the partial function determined by I is not defined on $\mathbf{A}$.

For technical reasons we extend this partial function to a partial function from $\mathrm{STR}[\sigma] \cup \{\varnothing\}$ to $\mathrm{STR}[\tau] \cup \{\varnothing\}$ by adding to its domain $\varnothing$ as well as those $\mathbf{A} \in \mathrm{STR}[\sigma]$ with

[1] This slightly deviates from [23], where the τ_k's are assumed to be pairwise equal and only *ordered* structures are considered.

$\varphi_U(\mathbf{A}) = \emptyset$; these additional arguments are all mapped to $\emptyset$. We denote the resulting partial function again by I.

We need to agree upon a way how to consider pairs of structures as a single structure:

Definition 5.3. Given a pair $(\mathbf{A}, \mathbf{B})$ of a σ-structure $\mathbf{A}$ and a τ-structure $\mathbf{B}$, define the structure $\langle \mathbf{A}, \mathbf{B} \rangle$ by taking the disjoint union of $\mathbf{A}$ and $\mathbf{B}$ and interpreting two new unary relation symbols P_1 and P_2 by the (copies of the) universes of $\mathbf{A}$ and $\mathbf{B}$ respectively. Naturally here, the disjoint union of $\mathbf{A}$ and $\mathbf{B}$ has universe $(\{1\} \times A) \dot\cup (\{2\} \times B)$ and interprets $R \in \sigma \cup \tau$ by $R_A \cup R_B$ where $R_A := \emptyset$ if $R \notin \sigma$ and else $R_A := \{((1, a_1), \ldots, (1, a_{\mathrm{ar}(R)})) \mid \bar{a} \in R^{\mathbf{A}}\}$; R_B is defined analogously. For $k \geqslant 3$ many structures $\mathbf{A}_1, \ldots, \mathbf{A}_k$ we inductively set $\langle \mathbf{A}_1, \ldots, \mathbf{A}_k \rangle := \langle \langle \mathbf{A}_1, \ldots, \mathbf{A}_{k-1} \rangle, \mathbf{A}_k \rangle$.

It is well-known that NP contains problems that are complete under quantifier-free reductions, i.e. reductions computed by a quantifier-free interpretation I as above. Dawar and He [17] transferred the notions to the parameterized setting and asked whether central completeness results for the classes of the W-hierarchy exhibit a similar robustness. More precisely, Dawar and He defined a parameterized reduction r from Q to Q' to be *slicewise quantifier-free definable* if there exists $w \in \mathbb{N}$ and a computable function d that maps every $k \in \mathbb{N}$ to some quantifier-free interpretation $d(k)$ of dimension w such that $r((\mathbf{A}, k)) = d(k)((\mathbf{A}, k))$; here, one views $(\mathbf{A}, k)$ in some suitable way as a single structure.

Definition 5.4. Let Q, Q' be parameterized problems (Definition 5.1). For $k \in \mathbb{N}$ let τ_k be the vocabulary of the k-th slice of Q. A parameterized reduction r from Q to Q' is *quantifier-free after a pre-computation* if there are $w \in \mathbb{N}$ and computable functions

- $p : \mathbb{N} \to \mathbb{N}$
- $a : \mathbb{N} \to \mathrm{STR}$
- d mapping $k \in \mathbb{N}$ to a quantifier-free interpretation $d(k)$ of dimension w,

such that for all $(\mathbf{A}, k) \in \mathrm{STR}[\tau_k] \times \mathbb{N}$:

- $d(k)$ is defined on $\langle a(k), \mathbf{A} \rangle$, and
- $r((\mathbf{A}, k)) = (\mathbf{A}', k')$ for $\mathbf{A}' := d(k)(\langle a(k), \mathbf{A} \rangle)$ and $k' := p(k)$.

We write $Q \leqslant_{qfap} Q'$ to indicate that such a reduction exists, and $Q \equiv_{qfap} Q'$ to indicate that both $Q \leqslant_{qfap} Q'$ and $Q' \leqslant_{qfap} Q$.

Remark 5.5. Note that the new *p*arameter $p(k)$ is computed by p from k alone, a is the pre-computation providing an *a*uxiliary structure, and d provides the *d*efinition of the new structure $\mathbf{A}'$.

Remark 5.6. We allow a reduction r to output $(\emptyset, k')$ for some $k' \in \mathbb{N}$. This is considered to be a "no" instance of any parameterized problem. For example, in the definition above we have $r(\mathbf{A}, k) = (\emptyset, p(k))$ if $\varphi_U(\mathbf{A}) = \emptyset$ where we write $d(k) = (\varphi_R)_{R \in \{U, =\} \cup \ldots}$.

Lemma 5.7. *Let Q, Q', Q'' be parameterized problems. If $Q \leqslant_{qfap} Q'$ and $Q' \leqslant_{qfap} Q''$, then $Q \leqslant_{qfap} Q''$.*

Proof. We need some folklore combinatorics concerning first-order interpretations. We give some details in order to be clear about our special treatment of $\varnothing$.

Claim 1: For $i \in \{1,2\}$ there is a dimension 1 quantifier-free interpretation Pr_i such that $Pr_i(\langle \mathbf{A}_1, \mathbf{A}_2 \rangle)$ is defined and isomorphic to $\mathbf{A}_i$ for all structures $\mathbf{A}_1, \mathbf{A}_2$.

We omit the easy proof.

Claim 2: Assume I, J are quantifier-free interpretations of dimensions w, w' respectively, and such that the output vocabulary of J contains the input vocabulary of I. Then there is a quantifier-free interpretation $(I \circ J)$ of dimension $w \cdot w'$ which is defined on a structure $\mathbf{A}$ whenever both J is defined on $\mathbf{A}$ and I is defined on $J(\mathbf{A})$, and then outputs $(I \circ J)(\mathbf{A}) \cong I(J(\mathbf{A}))$.

Notationally, we understand here that $\varnothing \cong \varnothing$.

Proof of Claim 2: Let $I = (\varphi_R)_R$ have dimension w and $J = (\psi_S)_S$ have dimension w'. Let σ be the input and τ be the output vocabulary of J. Associate with each variable x_j a w'-tuple $\bar{x}_j$ of variables. For every τ-formula $\psi = \psi(x_1, x_2, \ldots)$ there is a σ-formula $J(\psi) = J(\psi)(\bar{x}_1, \bar{x}_2, \ldots)$ such that for all $\mathbf{A} \in \mathrm{STR}[\sigma]$ with $J(\mathbf{A})$ defined and $\neq \varnothing$ we have for all $\bar{a}_1, \bar{a}_2, \ldots \in \varphi_U(\mathbf{A}) \subseteq A^{w'}$:

$$\mathbf{A} \models J(\psi)(\bar{a}_1, \bar{a}_2, \ldots) \iff J(\mathbf{A}) \models \psi(f(\bar{a}_1), f(\bar{a}_2), \ldots);$$

here, f is a surjection from $\varphi_U(\mathbf{A})$ onto the universe of $J(\mathbf{A})$ witnessing that $J(\mathbf{A})$ is defined and $\neq \varnothing$. The formula $J(\psi)$ is obtained from ψ by replacing atomic subformulas $Sx_{i_1} \cdots x_{i_{\mathrm{ar}(S)}}$ and $x_{i_1} = x_{i_2}$ of ψ by $\psi_S(\bar{x}_{i_1}, \ldots, \bar{x}_{i_{\mathrm{ar}(S)}})$ and $\varphi_=(\bar{x}_{i_1}, \bar{x}_{i_2})$, respectively. Then the interpretation $(J(\varphi_R))_R$ is as desired whenever $I(\mathbf{A}) \neq \varnothing$. To additionally ensure output $\varnothing$ whenever $J(\mathbf{A}) = \varnothing$, replace the formula $J(\varphi_U)(\bar{x}_1, \ldots, \bar{x}_w)$ by $J(\varphi_U)(\bar{x}_1, \ldots, \bar{x}_w) \wedge \bigwedge_{i \in [w]} \psi_U(\bar{x}_i)$. $\dashv$

Claim 3: Assume I, J are quantifier-free interpretations of dimensions w, w' respectively. Then there is a quantifier-free interpretation $\langle I, J \rangle$ of dimension $w + w' + 2$ which is defined on a structure $\mathbf{A}$ whenever both I and J are defined on $\mathbf{A}$, and then outputs $\langle I, J \rangle(\mathbf{A}) \cong \langle I(\mathbf{A}), J(\mathbf{A}) \rangle$.

Notationally, we understand here that $\langle \mathbf{A}, \mathbf{B} \rangle = \varnothing$ if $\mathbf{A} = \varnothing$ or $\mathbf{B} = \varnothing$.

Proof of Claim 3: Write $I = (\varphi_R)_{R \in \sigma \cup \{U,=\}}$ and $J = (\psi_R)_{R \in \tau \cup \{U,=\}}$. Then $\langle I, J \rangle$ is the interpretation $(\chi_R)_{R \in \sigma \cup \tau \cup \{P_1, P_2\} \cup \{U,=\}}$ defined as follows. Let $\bar{x}_i$ range over w-tuples and $\bar{y}_i$ range over w'-tuples. Set

$$\begin{aligned}
\chi_U(\bar{x}_1 \bar{y}_1 uv) &:= \varphi_U(\bar{x}_1) \wedge \psi_U(\bar{y}_1), \\
\chi_=(\bar{x}_1 \bar{y}_1 u_1 v_1, \bar{x}_2 \bar{y}_2 u_2 v_2) &:= (u_1 = v_1 \wedge u_2 = v_2 \wedge \varphi_=(\bar{x}_1, \bar{x}_2)) \\
&\qquad \vee (u_1 \neq v_1 \wedge u_2 \neq v_2 \wedge \psi_=(\bar{y}_1, \bar{y}_2)), \\
\chi_{P_1}(\bar{x}\bar{y}uv) &:= u = v, \\
\chi_{P_2}(\bar{x}\bar{y}uv) &:= u \neq v.
\end{aligned}$$

For $R \in \sigma \cup \tau$ define $\chi_R(\bar{x}_1\bar{y}_1u_1v_1, \ldots, \bar{x}_{\mathrm{ar}(R)}\bar{y}_{\mathrm{ar}(R)}u_{\mathrm{ar}(R)}v_{\mathrm{ar}(R)})$ by

$$\big(\varphi_R(\bar{x}_1, \ldots, \bar{x}_{\mathrm{ar}(R)}) \wedge \textstyle\bigwedge_{i\in[\mathrm{ar}(R)]} u_i = v_i\big) \vee \big(\psi_R(\bar{y}_1, \ldots, \bar{y}_{\mathrm{ar}(R)}) \wedge \textstyle\bigwedge_{i\in[\mathrm{ar}(R)]} u_i \neq v_i\big),$$

where φ_R and ψ_R are inconsistent formulas if $R \in \tau\backslash\sigma$ resp. $R \in \sigma\backslash\tau$. This interpretation is as desired. In particular, if $I(\mathbf{A}) = \varnothing$ or $J(\mathbf{A}) = \varnothing$, then $\varphi_U(\mathbf{A}) = \varnothing$ resp. $\psi_U(\mathbf{A}) = \varnothing$, and then $\chi_U(\mathbf{A}) = \varnothing$, and hence $\langle I, J\rangle(\mathbf{A}) = \varnothing$. $\dashv$

We now prove the lemma. Let (p, a, d) witness $Q \leqslant_{\mathit{qfap}} Q'$ and (p', a', d') witness $Q' \leqslant_{\mathit{qfap}} Q''$ respectively. Define

- $\tilde{p}(k) := p'(p(k))$;
- $\tilde{a}(k) := \langle a(k), a'(p(k))\rangle$;
- $\tilde{d}(k) := d'(p(k)) \circ \langle Pr_2 \circ Pr_1, d(k) \circ \langle Pr_1 \circ Pr_1, Pr_2\rangle\rangle$.

We claim $(\tilde{p}, \tilde{a}, \tilde{d})$ witnesses $Q \leqslant_{\mathit{qfap}} Q''$. By construction the interpretations $I \circ J$ and $\langle I, J\rangle$ are computable from (I, J), and hence $\tilde{d}$ is computable. The dimensions of $d(k)$ and $d'(p(k))$ are constant (independent of k), say, w and w' respectively. Then it is easily checked, that $\tilde{d}(k)$ has constant dimension $w' \cdot (4w + 3)$.

Write $\mathbf{A}' := d(k)(\langle a(k), \mathbf{A}\rangle), k' := p(k)$ and $\mathbf{A}'' := d'(k')\big(\langle a'(k'), \mathbf{A}'\rangle\big), k'' := p'(k')$. Then $(\mathbf{A}, k) \in Q$ if and only if $(\mathbf{A}'', k'') \in Q''$. It suffices to show that $\tilde{d}(\langle \tilde{a}(k), \mathbf{A}\rangle) \cong \mathbf{A}''$.

But $\langle Pr_1 \circ Pr_1, Pr_2\rangle(\langle \tilde{a}(k), \mathbf{A}\rangle) \cong \langle a(k), \mathbf{A}\rangle$, and $(Pr_2 \circ Pr_1)(\langle \tilde{a}(k), \mathbf{A}\rangle) \cong a'(p(k))$. Hence

$$\tilde{d}(k)(\langle \tilde{a}(k), \mathbf{A}\rangle) \cong d'(k')\big(\langle a'(k'), d(k)(\langle a(k), \mathbf{A}\rangle)\rangle\big) \cong d'(k')\big(\langle a'(k'), \mathbf{A}'\rangle\big) = \mathbf{A}'',$$

as was to be shown. $\square$

Lemma 5.8. *Let Q, Q' be parameterized problems and assume Q' has bounded arity. If $Q \leqslant_{\mathit{qfap}} Q'$, then $Q \leqslant_{\mathit{pl}} Q'$.*

Proof. Let (p, a, d) witness $Q \leqslant_{\mathit{qfap}} Q'$. We need to explain how to compute the output $(\mathbf{A}', k')$ of the reduction in parameterized logarithmic space; here $k' = p(k)$ and $\mathbf{A}' = d(k)(\langle a(k), \mathbf{A}\rangle)$. The computation of $p(k), a(k)$ and $d(k)$ requires an amount of space that depends on the parameter k only. We show how to compute an isomorphic copy of $\mathbf{A}'$ from (a binary encoding of) $\langle a(k), \mathbf{A}\rangle$ and $d(k)$ in parameterized logarithmic space.

First note the following: for every formula $\varphi = \varphi(\bar{x})$ from $d(k)$ and for every length $|\bar{x}|$ tuple $\bar{a}$ from the universe $\tilde{A}$ of $\langle a(k), \mathbf{A}\rangle$ one can decide in space $O(|\varphi| \log |\varphi| + \log |\tilde{A}|)$ whether $\langle a(k), \mathbf{A}\rangle \models \varphi(\bar{a})$. Indeed, if w is the dimension of $d(k)$ and r bounds the arity of Q', then φ has at most rw many variables $\bar{x}$, and this is an absolute constant.

We compute a copy of $\mathbf{A}'$ with universe $[m]$ for $m = |A'|$. The binary encoding of the structure $\langle a(k), \mathbf{A}\rangle$ determines a linear order on $\tilde{A}$, and this induces a lexicographic order on finite tuples over $\tilde{A}$. To compute m, cycle through all $\bar{a} \in \tilde{A}^w$ in lexicographic order and increase a counter whenever $\bar{a}$ passes the following check: check that $\langle a(k), \mathbf{A}\rangle \models \varphi_U(\bar{a})$, and check that there is no $\bar{a}' \in \tilde{A}^w$ lexicographically smaller $\bar{a}$ and such that $\langle a(k), \mathbf{A}\rangle \models (\varphi_=(\bar{a}, \bar{a}') \wedge \varphi_U(\bar{a}'))$. The latter check is done by cycling through all $\bar{a}' \in \tilde{A}^w$.

Using a similar loop, one can determine, given $i \in [m]$, the i-th tuple $\bar{a}$ passing the check; we denote this tuple by $\bar{a}_i$. Now, to determine the bits of the encoding of the copy of $\mathbf{A}'$, it is sufficient to determine, given a relation symbol R and a tuple $\bar{\imath} = (i_1, \ldots, i_{\mathrm{ar}(R)})$ from $[m]^{\mathrm{ar}(R)}$, whether $\bar{\imath}$ satisfies the interpretation of R over $[m]$. This is done by computing $\bar{a}_{i_1}, \ldots, \bar{a}_{i_{\mathrm{ar}(R)}}$ and checking whether $\langle a(k), \mathbf{A}\rangle \models \varphi_R(\bar{a}_{i_1}, \ldots, \bar{a}_{i_{\mathrm{ar}(R)}})$. $\square$

Convention For technical reasons we need to consider homomorphism problems p-HOM$(\mathcal{A})$ also for classes $\mathcal{A}$ which are not necessarily decidable. In such a case we slightly abuse notation and write $p\text{-Hom}(\mathcal{A}) \leqslant_{qfap} Q$ for a parameterized problem Q to mean that there are *partially* computable functions p, a, d whose domain contains $\{\ulcorner\mathbf{A}\urcorner \mid \mathbf{A} \in \mathcal{A}\}$ such that for all $\mathbf{A} \in \mathcal{A}$ and similar $\mathbf{B}$ we have that $d(\ulcorner\mathbf{A}\urcorner)(\langle a(\ulcorner\mathbf{A}\urcorner), \mathbf{B}\rangle) =: \mathbf{B}'$ is defined and:

$$\mathbf{A} \xrightarrow{h} \mathbf{B} \iff (\mathbf{B}', p(\ulcorner\mathbf{A}\urcorner)) \in Q.$$

5.3 Homomorphism problems for graph classes

Let $\mathcal{G}, \mathcal{H}$ be computably enumerable classes of graphs. In this subsection we show that the associated homomorphism problems p-HOM$(\mathcal{G}^*)$ and p-HOM$(\mathcal{H}^*)$ are $\equiv_{qfap}$-equivalent if the graph classes $\mathcal{G}$ and $\mathcal{H}$ are $\equiv$-equivalent:

Theorem 5.9. *If $\mathcal{G} \preceq \mathcal{H}$, then p-HOM$(\mathcal{G}^*) \leqslant_{qfap}$ p-HOM$(\mathcal{H}^*)$.*

Proof. Choose w such that graphs in $\mathcal{G}$ have $\mathcal{H}$-deconstructions of width at most w. Using that $\mathcal{H}$ is computably enumerable, it is not hard to see that there is an algorithm that computes given $\mathbf{G} \in \mathcal{G}$ a graph $\mathbf{H} \in \mathcal{H}$ and an $\mathbf{H}$-deconstruction $(B_h)_{h\in H}$ of $\mathbf{G}$ of width at most w. Given an instance $(\mathbf{B}, \ulcorner\mathbf{G}^*\urcorner)$ of p-HOM$(\mathcal{G}^*)$ with $\mathbf{G} \in \mathcal{G}$ and $\mathbf{B}$ similar to $\mathbf{G}^*$ the reduction outputs $(\mathbf{B}', \ulcorner\mathbf{H}^*\urcorner)$ where $\mathbf{H} \in \mathcal{H}$ is as above and $\mathbf{B}'$ is defined as follows. Assume first that all bags $B_h, h \in H$, are nonempty.

For $\ell \in \mathbb{N}$ let $PH(\mathbf{G}^*, \mathbf{B}, \ell)$ denote the set of pairs

$$(g_1 \cdots g_\ell, b_1 \cdots b_\ell) \in G^\ell \times B^\ell$$

such that $\{(g_1, b_1), \ldots, (g_\ell, b_\ell)\}$ is a partial homomorphism from $\mathbf{G}^*$ to $\mathbf{B}$. For each $h \in H$ choose a tuple $\bar{g}^h := g_1^h \cdots g_w^h \in G^w$ that lists the elements of B_h. Note that $PH(\mathbf{G}^*, \mathbf{B}, \ell)$ is empty only if $(\mathbf{B}, \ulcorner\mathbf{G}^*\urcorner)$ is a "no"-instance of p-HOM$(\mathcal{G}^*)$. If $PH(\mathbf{G}^*, \mathbf{B}, \ell)$ is non-empty it carries a structure $\mathbf{B}'$ defined as follows:

$$\begin{aligned}
B' &:= PH(\mathbf{G}^*, \mathbf{B}, w),\\
E^{\mathbf{B}'} &:= \big\{((\bar{g}, \bar{b}), (\bar{g}', \bar{b}')) \in PH(\mathbf{G}^*, \mathbf{B}, w)^2 \mid (\bar{g}\bar{g}', \bar{b}\bar{b}') \in PH(\mathbf{G}^*, \mathbf{B}, 2w)\big\},\\
C_h^{\mathbf{B}'} &:= \big\{(\bar{g}, \bar{b}) \in PH(\mathbf{G}^*, \mathbf{B}, w) \mid \bar{g} = \bar{g}^h\big\}, \quad \text{for } h \in H.
\end{aligned}$$

We claim that

$$\mathbf{G}^* \xrightarrow{h} \mathbf{B} \iff \mathbf{H}^* \xrightarrow{h} \mathbf{B}'.$$

If f is a homomorphism from $\mathbf{G}^*$ to $\mathbf{B}$, then $h \mapsto (\bar{g}^h, f(\bar{g}^h))$ is a homomorphism from $\mathbf{H}^*$ to $\mathbf{B}'$. Conversely, suppose that f' is a homomorphism from $\mathbf{H}^*$ to $\mathbf{B}'$. If f' maps $h \in H$ to $(\bar{g}, \bar{b}) \in B'$, let f^h be the map $\{(g_1, b_1), \ldots, (g_w, b_w)\}$. Note that $\mathsf{dom}(f^h) = B_h$ because f' preserves the colours C_h. Any two such maps f^h and $f^{h'}$ are compatible in the sense that they agree on arguments on which they are both defined: indeed, if $a \in \mathsf{dom}(f^h) \cap \mathsf{dom}(f^{h'})$ then $a \in B_h$ and $a \in B_{h'}$, so there is a path in $\mathbf{H}$ from h to h'; if $f^h(a) \neq f^{h'}(a)$ then there exists neighbors h_0, h_1 on this path such that $f^{h_0}(a) \neq f^{h_1}(a)$; then $f^{h_0} \cup f^{h_1}$ is not a function, and in particular $(f'(h_0), f'(h_1)) \notin E^{\mathbf{B}'}$; as $(h_0, h_1) \in E^{\mathbf{H}}$ this contradicts f' being a homomorphism. Therefore and since every $g \in G$ appears in some B_h, $f := \bigcup_{h \in H} f^h$ is a function from G to B. To verify it is a homomorphism we show that it preserves E; that is preserves the colours C_g can be seen similary. So, given an edge $(g, g') \in E^{\mathbf{G}}$ we have to show $(f(g), f(g')) \in E^{\mathbf{B}}$. Choose $(h, h') \in \mathsf{refl}(E^{\mathbf{H}})$ such that $\{g, g'\} \in B_h \cup B_{h'}$. Then $f^h \cup f^{h'}$ is a partial homomorphism from $\mathbf{G}^*$ to $\mathbf{B}$: this is clear if $h = h'$; otherwise $(h, h') \in E^{\mathbf{H}}$, so $(f'(h), f'(h')) \in E^{\mathbf{B}'}$ and it follows by definition of $E^{\mathbf{B}'}$ that $f^h \cup f^{h'}$ is a partial homomorphism. But $f^h \cup f^{h'}$ is defined on g, g', so $f^h \cup f^{h'}$ and hence f maps (g, g') to an edge in $E^{\mathbf{B}}$.

We are left to show that there is a quantifier-free interpretation producing $\mathbf{B}'$ from $\langle \tilde{\mathbf{G}}, \mathbf{B} \rangle$ for some structure $\tilde{\mathbf{G}}$ computable from $\mathbf{G}^*$. For $\tilde{\mathbf{G}}$ we take the expansion of $\mathbf{G}^*$ that interprets for every $h \in H$ a w-ary relation symbol B_h by $\{\bar{g}^h\}$. For w-tuples of variables $\bar{x} = x_1 \cdots x_w$ and $\bar{y} = y_1 \cdots y_w$ consider the formula

$$\begin{aligned} ph^w(\bar{x}, \bar{y}) \quad := \quad & \textstyle\bigwedge_{i \in [w]} P_1 x_i \wedge \bigwedge_{i \in [w]} P_2 y_i \wedge \bigwedge_{(i,i') \in [w]^2} (x_i = x_{i'} \to y_i = y_{i'}) \\ & \textstyle\wedge \bigwedge_{(i,i') \in [w]^2} (E x_i x_{i'} \to E y_i y_{i'}) \wedge \bigwedge_{i \in [w]} \bigwedge_{g \in G} (C_g x_i \to C_g y_i). \end{aligned}$$

Let ph^{2w} be similarly defined for $2w$-tuples. Then define

$$\begin{aligned} \varphi_U(\bar{x}\bar{y}) \quad &:= \quad ph^w(\bar{x}, \bar{y}) \\ \varphi_=(\bar{x}\bar{y}, \bar{x}'\bar{y}') \quad &:= \quad \textstyle\bigwedge_{i \in [w]} (x_i = x'_i \wedge y_i = y'_i) \\ \varphi_E(\bar{x}\bar{y}, \bar{x}'\bar{y}') \quad &:= \quad ph^{2w}(\bar{x}\bar{x}', \bar{y}\bar{y}') \\ \varphi_{C_h}(\bar{x}\bar{y}) \quad &:= \quad B_h \bar{x}, \end{aligned}$$

for $h \in H$. This is a quantifier-free interpretation I of dimension $2w$ such that $I(\langle \tilde{\mathbf{G}}, \mathbf{B} \rangle)$ is defined and equals $\emptyset$ if $PH(\mathbf{G}^*, \mathbf{B}, w) = \emptyset$, and otherwise equals $\mathbf{B}'$. This finishes the proof for the case that the bags $B_h, h \in H$, are nonempty.

In the general case we can assume that any B_h is nonempty whenever there exists $h' \in H$ in the connected component of h in $\mathbf{H}$ such that $B_{h'} \neq \emptyset$. Thus we can assume that $\mathbf{H}$ is the disjoint union of $\mathbf{H}_0$ and $\mathbf{H}_1$ such that $B_h \neq \emptyset$ for all $h \in H_0$ and $B_h = \emptyset$ for all $h \in H_1$. As seen above we get a $\mathbf{B}'$ such that either $\mathbf{B}' = \emptyset$ and $(\mathbf{B}, \ulcorner \mathbf{G}^* \urcorner)$ is a "no"-instance of p-HOM$(\mathcal{G}^*)$, or $\mathbf{B}'$ is a structure such that

$$\mathbf{G}^* \xrightarrow{h} \mathbf{B} \quad \Longleftrightarrow \quad \mathbf{H}_0^* \xrightarrow{h} \mathbf{B}'.$$

Define $\mathbf{B}''$ as follows using a new vertex $b'' \notin B'$:

$$\begin{array}{rcl} B'' & := & B' \cup \{b''\}, \\ E^{\mathbf{B}''} & := & E^{\mathbf{B}'} \cup \{(b'', b'')\}, \\ C_h^{\mathbf{B}''} & := & \left\{ \begin{array}{ll} C_h^{\mathbf{B}'} & , h \in H_0 \\ \{b''\} & , h \in H_1 \end{array} \right. ; \end{array}$$

in case $\mathbf{B}' = \emptyset$ we understand here that the sets $B', E^{\mathbf{B}'}, C_h^{\mathbf{B}'}$ are empty.

It is straightforward to check that $\mathbf{B}''$ can be defined by a quantifier-free interpretation: e.g. as formula $\varphi_U(\bar{x}\bar{y})$ one may now take

$$ph^w(\bar{x}, \bar{y}) \vee \bigwedge_{i \in [w]} (C_{g_0} x_i \wedge x_i = y_i \wedge P_1 x_i),$$

for some fixed vertex $g_0 \in G$. Furthermore, it is easy to see that

$$\mathbf{H}_0^* \xrightarrow{h} \mathbf{B}' \iff \mathbf{H}^* \xrightarrow{h} \mathbf{B}'',$$

if $\mathbf{B}' \neq \emptyset$; if $\mathbf{B}' = \emptyset$, then $\mathbf{H}^* \xrightarrow{h} \mathbf{B}''$ fails. Therefore $(\mathbf{G}^*, \mathbf{B}) \mapsto (\mathbf{H}^*, \mathbf{B}'')$ is a reduction from p-Hom$(\mathcal{G}^*)$ to p-Hom$(\mathcal{H}^*)$ as desired. □

5.4 Homomorphism problems for classes of arbitrary structures

Let $\mathcal{A}$ be a computably enumerable class of structures.

Theorem 5.10. *Suppose $\mathcal{A}$ has bounded arity. Let $\mathcal{G}$ denote the class of graphs from the hierarchy ($*$) having the property that $\mathcal{G} \equiv \mathsf{graph}(\mathsf{core}(\mathcal{A}))$. Then*

$$p\text{-Hom}(\mathcal{A}) \equiv_{qfap} p\text{-Hom}(\mathcal{G}^*).$$

Thus, the complexity of p-Hom$(\mathcal{A})$ is determined in a strong sense by the level which the graph class $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$ takes in our hierarchy. For example, because the reductions are weaker than fpt-reductions it is the level $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$ takes in our hierarchy what determines whether p-Hom$(\mathcal{A})$ is W[1]-complete or fixed-parameter tractable (cf. [27]), and because it is weaker than pl-reductions this level determines whether p-Hom$(\mathcal{A})$ is in para-L or PATH or TREE (cf. [11]).

We divide the proof into a sequence of lemmas. The first two of them are analogues of Lemmas 4.4 and 4.3. It is straightforward, and left to the reader, to verify that these reductions are quantifier-free after a pre-computation.

Lemma 5.11. *If $\mathcal{A}$ is a class of cores, then p-Hom$(\mathcal{A}^*) \leqslant_{qfap} p$-Hom$(\mathcal{A})$.*

Lemma 5.12. p-Hom$(\mathsf{graph}(\mathcal{A})^*) \leqslant_{qfap} p$-Hom$(\mathcal{A}^*)$.

Corollary 5.13. p-Hom$(\mathcal{A}) \equiv_{qfap} p$-Hom$(\mathsf{core}(\mathcal{A})^*)$.

Proof. By Lemma 5.7 it suffices to establish the following sequence of reductions:

$$p\text{-}\mathrm{HOM}(\mathcal{A}) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A})^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A})) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathcal{A}).$$

To see $p\text{-}\mathrm{HOM}(\mathcal{A}) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A})^*)$, map an instance $(\mathbf{B}, \ulcorner\mathbf{A}\urcorner)$ with $\mathbf{A} \in \mathcal{A}$ to $(\mathbf{B}', \ulcorner\mathsf{core}(\mathbf{A})^*\urcorner)$ where $\mathbf{B}'$ expands $\mathbf{B}$ interpeting $C_a, a \in A$, by $C_a^{\mathbf{B}'} := B$. It is easy to see that this is quantifier-free after a pre-computation. That $p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A})^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A}))$ follows from Lemma 5.11. Finally, to see $p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A})) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathcal{A})$ we use (here and only here) that $\mathcal{A}$ is computably enumerable: this ensures that there exists a partially computable function p such that for all $\mathbf{A} \in \mathsf{core}(\mathcal{A})$ we have that $p(\ulcorner\mathbf{A}\urcorner)$ is the code of a structure in $\mathcal{A}$ whose core is $\mathbf{A}$; the reduction then maps $(\mathbf{B}, \ulcorner\mathbf{A}\urcorner)$ to $(\mathbf{B}, p(\ulcorner\mathbf{A}\urcorner))$. Again it is clear that this is quantifier-free after a pre-computation. □

We show a partial converse to Lemma 5.12. This is the technically most involved step in the proof of Theorem 5.10.

Lemma 5.14. *If $\mathcal{A}$ has bounded arity, then $p\text{-}\mathrm{HOM}(\mathcal{A}^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathsf{graph}(\mathcal{A})^*)$.*

Proof. We consider two cases depending on whether or not $\mathsf{graph}(\mathcal{A})$ has bounded treewidth.

Case 1: Suppose $\mathsf{graph}(\mathcal{A})$ has unbounded treewidth. Then $\mathcal{L} \preceq \mathsf{graph}(\mathcal{A})$ by the Hierarchy Theorem. By Theorem 5.9 we get $p\text{-}\mathrm{HOM}(\mathcal{L}^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathsf{graph}(\mathcal{A})^*)$. Since trivially $p\text{-}\mathrm{HOM}(\mathcal{H}^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathcal{L}^*)$ for every class of graphs $\mathcal{H}$ it suffices to show $p\text{-}\mathrm{HOM}(\mathcal{A}^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathcal{H}^*)$ for some class of graphs $\mathcal{H}$. In fact, we show that for every $\mathcal{A}$ with bounded arity (not necessarily of the form $\mathcal{A}^*$) there exists a class of graphs $\mathcal{H}$ such that $p\text{-}\mathrm{HOM}(\mathcal{A}) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathcal{H}^*)$.

Let σ be a vocabulary and $\mathbf{A}$ be a σ-structure. We define the following graph $\mathrm{in}(\mathbf{A}) = (L(\mathbf{A}) \dot\cup R(\mathbf{A}), E^{\mathrm{in}(\mathbf{A})})$, reminiscent of the incidence graph. Its universe has "left" vertices

$$L(\mathbf{A}) := \big\{(R, \bar{a}, i) \mid R \in \sigma, \bar{a} \in R^A, i \in [\mathrm{ar}(R)]\big\}$$

together with "right" vertices $R(\mathbf{A}) := A$; for notational simplicity we assume that $L(\mathbf{A}) \cap R(\mathbf{A}) = \emptyset$. Its edges $E^{\mathrm{in}(\mathbf{A})}$ are also devided into two kinds, namely we have edges "on the left" between $(R, \bar{a}, i) \in L(\mathbf{A})$ and $(R, \bar{a}, i') \in L(\mathbf{A})$ for $i \neq i'$ together with "left to right" edges between $(R, a_1 \cdots a_{\mathrm{ar}(R)}, i) \in L(\mathbf{A})$ and $a_i \in R(\mathbf{A})$.

Given $\mathbf{A} \in \mathcal{A}$ and a structure $\mathbf{B}$, say both of vocabulary σ, define the following structure $\mathbf{B}'$. It expands $\mathrm{in}(\mathbf{B})$ to a structure interpreting the language of $\mathrm{in}(\mathbf{A})^*$. Namely, for $(R, \bar{a}, i) \in L(\mathbf{A})$ we set

$$C^{\mathbf{B}'}_{(R,\bar{a},i)} \quad := \quad \big\{(R, \bar{b}, i) \mid R \in \sigma, \bar{b} \in R^{\mathbf{B}}, i \in [\mathrm{ar}(R)]\big\},$$

a subset of $L(\mathbf{B})$, and for $a \in R(\mathbf{A})$ we set $C_a^{\mathbf{B}'} := R(\mathbf{B}) = B$.

We claim that $(\mathbf{A}, \mathbf{B}) \mapsto (\mathrm{in}(\mathbf{A})^*, \mathbf{B}')$ is a reduction as desired. We first show that

$$\mathbf{A} \xrightarrow{h} \mathbf{B} \quad \Longleftrightarrow \quad \mathrm{in}(\mathbf{A})^* \xrightarrow{h} \mathbf{B}'.$$

To see the forward direction, assume h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Define $h' : L(\mathbf{A}) \cup R(\mathbf{A}) \to B' = L(\mathbf{B}) \cup R(\mathbf{B})$ setting $h'(a) := h(a) \in R(\mathbf{B})$ for $a \in R(\mathbf{A})$ and $h'((R, \bar{a}, i)) := (R, h(\bar{a}), i) \in L(\mathbf{B})$ for $(R, \bar{a}, i) \in L(\mathbf{A})$. Note that indeed $(R, h(\bar{a}), i) \in L(\mathbf{B})$ because $h(\bar{a}) \in R^{\mathbf{B}}$ as h is a homomorphism. It also follows that h' preserves the colours $C_{(R,\bar{a},i)}$; that it preserves the colours C_a is clear. It is also clear that it preserves edges "on the left". Finally consider a "left to right" edge between $(R, a_1 \cdots a_{\mathrm{ar}(R)}, i) \in L(\mathbf{A})$ and $a_i \in R(\mathbf{A})$. Then $(h'((R, \bar{a}, i)), h'(a_i)) = ((R, h(\bar{a}), i), h(a_i))$ and this is in $E^{\mathbf{B}} = E^{\mathbf{B}'}$.

Conversely, let h' be a homomorphism from $\mathrm{in}(\mathbf{A})^*$ to $\mathbf{B}'$ and let h be its restriction to $R(\mathbf{A}) = A$. Since h' preserves the colours $C_a, a \in A$, the function h takes values in $R(\mathbf{B}) = B$. We claim h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Let $\bar{a} = a_1 \cdots a_{\mathrm{ar}(R)} \in R^{\mathbf{A}}$ for $R \in \sigma$. For each $i \in [\mathrm{ar}(R)]$ there exists a tuple $\bar{b}_i \in R^{\mathbf{B}}$ such that $h'((R, \bar{a}, i)) = (R, \bar{b}_i, i)$ because h' preserves the colour $C_{(R,\bar{a},i)}$. But, in fact, $\bar{b} := \bar{b}_i$ does not depend on i: if there would be $i, i' \in [\mathrm{ar}(R)]$ such that $\bar{b}_i \neq \bar{b}_{i'}$, then h' would not preserve the edge "on the left" between $(R, \bar{a}, i)$ and $(R, \bar{a}, i')$. It now suffices to show that $\bar{b} = h(\bar{a})$, that is, $b_i = h(a_i)$ for all $i \in [\mathrm{ar}(R)]$ (where we write $\bar{b} = b_1 \cdots b_{\mathrm{ar}(R)}$). If this would fail for $i \in [\mathrm{ar}(R)]$, then $((R, \bar{b}, i), h(a_i)) = (h'(R, \bar{a}, i), h'(a_i)) \notin E^{\mathrm{in}(\mathbf{B})} = E^{\mathbf{B}'}$ while $((R, \bar{a}, i), a_i) \in E^{\mathrm{in}(\mathbf{A})}$, contradicting that h' is a homomorphism.

Thus, $(\mathbf{A}, \mathbf{B}) \mapsto (\mathrm{in}(\mathbf{A})^*, \mathbf{B}')$ defines a parameterized reduction from p-HOM($\mathcal{A}$) to p-HOM($\mathrm{in}(\mathcal{A})^*$) where $\mathrm{in}(\mathcal{A}) := \{\mathrm{in}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{A}\}$ is a class of graphs. We are left to show that this reduction is quantifier-free after a pre-computation. It is here where we are going to use the assumption that $\mathcal{A}$ has bounded arity.

It suffices to show that some isomorphic copy of $\mathbf{B}'$ can be produced from $\langle \mathbf{A}', \mathbf{B} \rangle$ by a suitable interpretation I where $\mathbf{A}'$ is some auxiliary structure. It will be clear that I and $\mathbf{A}'$ are computable from $\ulcorner \mathbf{A} \urcorner$. The dimension of I is $r + 1$ where $r \geqslant 1$ bounds the arity of $\mathcal{A}$. If $\mathbf{A}$ has vocabulary σ, the auxiliary structure $\mathbf{A}'$ is $(\{0\} \cup (\sigma \times [r]))^*$, i.e. take the $\emptyset$-structure with universe $\{0\} \cup (\sigma \times [r])$ and add colours for the elements. The interpretation produces the isomorphic copy of $\mathbf{B}'$ where $(R, \bar{b}, i) \in L(\mathbf{B})$ is replaced by $((R, i), \bar{b}\bar{0})$ where $\bar{0}$ is a $(r - \mathrm{ar}(R))$-tuple of 0s. Similarly, $b \in R(\mathbf{B}) = B$ is replaced by $b\bar{0}$ with an r-tuple of 0s. Recall, $\mathbf{B}'$ has vocabulary

$$\{C_{(R,\bar{a},i)} \mid (R, \bar{a}, i) \in L(\mathbf{A})\} \cup \{C_a \mid a \in R(\mathbf{A})\} \cup \{E\}.$$

Letting $\bar{x} = x_1 \cdots x_r$ and $\bar{x}' = x'_1 \cdots x'_r$ be r-tuples of variables, the interpretation I reads as follows. For $R \in \sigma, i \in [\mathrm{ar}(R)]$ set

$$\begin{aligned} \mathit{left}_{R,i}(z\bar{x}) &:= P_1 z \wedge C_{(R,i)} z \wedge R x_1 \cdots x_{\mathrm{ar}(R)} \wedge \textstyle\bigwedge_{j \in [r] \setminus [\mathrm{ar}(R)]} (P_1 x_j \wedge C_0 x_j), \\ \mathit{right}(z\bar{x}) &:= P_2 z \wedge \textstyle\bigwedge_{j \in [r]} (P_1 x_j \wedge C_0 x_j), \end{aligned}$$

and define

$$\begin{aligned} \varphi_U(z\bar{x}) &:= \textstyle\bigvee_{R \in \sigma} \bigvee_{i \in [\mathrm{ar}(R)]} \mathit{left}_{R,i}(z\bar{x}) \vee \mathit{right}(z\bar{x}), \\ \varphi_{C_{(R,\bar{a},i)}}(z\bar{x}) &:= \mathit{left}_{R,i}(z\bar{x}), \\ \varphi_{C_a}(z\bar{x}) &:= \mathit{right}(z\bar{x}), \\ \varphi_E(z\bar{x}, z'\bar{x}') &:= \varphi'_E(z\bar{x}, z'\bar{x}') \vee \varphi'_E(z'\bar{x}', z\bar{x}), \end{aligned}$$

where $\varphi'_E(z\bar{x}, z'\bar{x}')$ is

$$\bigvee_{R\in\sigma} \bigvee_{i\in[\mathrm{ar}(R)]} \Big(\big(\mathit{left}_{R,i}(z\bar{x}) \wedge \bigvee_{i'\in[r]\setminus\{i\}} \mathit{left}_{R,i'}(z'\bar{x}') \wedge \bigwedge_{j\in[r]} x_j = x'_j \big) \\ \vee \big(\mathit{left}_{R,i}(z\bar{x}) \wedge \mathit{right}(z'\bar{x}') \wedge z' = x_i \big) \Big).$$

Case 2: Suppose $\mathsf{graph}(\mathcal{A})$ has bounded treewidth. Then $\mathsf{graph}(\mathcal{A}) \preceq \mathcal{T}$ and by the Hierarchy Theorem there exists a class of forests $\mathcal{H}$ such that $\mathsf{graph}(\mathcal{A}) \equiv \mathcal{H}$. Moreover, we can assume that $\mathcal{H}$ is computably enumerable and closed under deleting proper (connected) components since so are the classes of graphs listed in this theorem. Theorem 5.9 implies that p-Hom$(\mathsf{graph}(\mathcal{A})^*) \equiv_{qfap} p$-Hom$(\mathcal{H}^*)$, so it suffices to show

$$p\text{-}\mathrm{Hom}(\mathcal{A}^*) \leqslant_{qfap} p\text{-}\mathrm{Hom}(\mathcal{H}^*).$$

Arguing as in Proposition 3.10 (2), one sees that there is a w such that for every $\mathbf{A} \in \mathcal{A}$ there exists $\mathbf{H} \in \mathcal{H}$ such that $\mathsf{graph}(\mathbf{A})$ has an $\mathbf{H}$-decomposition. Given an instance $(\mathbf{B}, \ulcorner\mathbf{A}^*\urcorner)$ of p-Hom$(\mathcal{A}^*)$ with $\mathbf{A}^* \in \mathcal{A}^*$ compute $\mathbf{H} \in \mathcal{H}$ and a $\mathbf{H}$-decomposition $(B_h)_{h\in H}$ of $\mathsf{graph}(\mathbf{A})$ of width w. As in the proof of Theorem 5.9 we can assume that each component of $\mathbf{H}$ either has all its bags empty or all its bags nonempty. We can even assume that all bags are nonempty - otherwise use the above closure assumption on $\mathcal{H}$ and throw away all components with empty bags. Define $\mathbf{B}'$ as in the proof of Theorem 5.9, now with $\mathbf{A}^*$ replacing $\mathbf{G}^*$ there. We show

$$\mathbf{A}^* \xrightarrow{h} \mathbf{B} \quad \iff \quad \mathbf{H}^* \xrightarrow{h} \mathbf{B}'.$$

The direction from left to right is seen as in the proof of Theorem 5.9. Conversely, let f' be a homomorphism from $\mathbf{H}^*$ to $\mathbf{B}'$ and let $\mathbf{A}_0$ be a component of $\mathbf{A}$. Choose a component, i.e. a tree, $\mathbf{H}_0$ of $\mathbf{H}$ such that $(B_h \cap A_0)_{h\in H_0}$ is a tree decomposition of $\mathsf{graph}(\mathbf{A}_0)$. For $h \in H_0$ define f_0^h as in the proof of Theorem 5.9. Again we show that $f_0 := \bigcup_{h\in H_0} f_0^h$ is a homomorphism from $\mathbf{A}_0$ to $\mathbf{B}$. To see this, let R be a relation symbol in the vocabulary of $\mathbf{A}$ and $\bar{a} \in R^{\mathbf{A}_0}$. Then the components of $\bar{a}$ form a clique in $\mathsf{graph}(\mathbf{A}_0)$. It is well-known for tree decompositions, that cliques are contained in some bag (see e.g. [6, Lemma 4]), that is, there is $h \in H_0$ such that B_h contains all components of $\bar{a}$. As f_0^h is a partial homomorphism from $\mathbf{A}_0$ to $\mathbf{B}$ and contains $\bar{a}$ in its domain, we get $f_0^h(\bar{a}) = f_0(\bar{a}) \in R^{\mathbf{B}}$ as desired.

That $(\mathbf{A}^*, \mathbf{B}) \mapsto (\mathbf{H}^*, \mathbf{B}')$ is quantifier-free after a pre-computation is seen as in the proof of Theorem 5.9. □

The last two lemmas imply:

Corollary 5.15. *If $\mathcal{A}$ has bounded arity, then*

$$p\text{-}\mathrm{Hom}(\mathcal{A}^*) \equiv_{qfap} p\text{-}\mathrm{Hom}(\mathsf{graph}(\mathcal{A})^*).$$

Proof of Theorem 5.10. Note that with $\mathcal{A}$ also $\mathsf{core}(\mathcal{A})$ is computably enumerable. Thus

$$p\text{-}\mathrm{Hom}(\mathcal{A}) \equiv_{qfap} p\text{-}\mathrm{Hom}(\mathsf{core}(\mathcal{A})^*) \equiv_{qfap} p\text{-}\mathrm{Hom}(\mathsf{graph}(\mathsf{core}(\mathcal{A}))^*)$$

by Corollaries 5.13 and 5.15. Now apply Theorem 5.9. □

6 Pebble games

Following the motivation given in the introduction, this section presents pebble games which can and will be used to solve the homomorphism problems associated with the lower levels of the hierarchy (Theorem 3.16). We consider pebble games (cf. [20, Section 3.3]) played by two players, *Spoiler* and *Duplicator* on two similar structures $\mathbf{A}, \mathbf{B}$. Let us call a tuple $v = (p_1, \ldots, p_r) \in \mathbb{N}^r$ for $r \geqslant 1$ a *game vector* with r *rounds* and $\sum_{i \in [r]} p_i$ *pebbles.*

Informally speaking, the game has r rounds; in the ith round, Spoiler places $\leqslant p_i$ many pebbles on elements of A and Duplicator responds placing equally many pebbles on B; Duplicator wins if in the end the correspondence between pebbled elements is a partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Formally, we define a *Duplicator winning strategy for the v-game on* $(\mathbf{A}, \mathbf{B})$ to be a sequence $(W_1, \ldots, W_r)$ of sets of partial homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ such that:

- For all $S \subseteq A$ with $|S| \leqslant p_1$, the set W_1 contains a mapping with domain S.
- For all $i \in [r-1]$, $g \in W_i$ and supersets S of $\mathsf{dom}(g)$ with $|S \backslash \mathsf{dom}(g)| \leqslant p_{i+1}$, there is an extension $g' \in W_{i+1}$ of g with domain S.

We write $\mathbf{A} \xrightarrow{v} \mathbf{B}$ to indicate that such a strategy exists.

This section's first main theorem provides a decidable characterization of the homomorphism problems arising from a single structure that are solved by the v-game. We say that *the v-game solves p-*$\textsc{Hom}(\mathcal{A})$ if for any instance $(\mathbf{A}, \mathbf{B})$ thereof, the existence of a Duplicator winning strategy for the v-game (on the instance) implies that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Let $v = (p_1, \ldots, p_r)$ be a game vector. A *v-decomposition* of a graph $\mathbf{G}$ is an $\mathbf{H}$-decomposition $(B_h)_{h \in H}$ such that:

- $\mathbf{H}$ is a rooted forest of height $< r$,
- $|B_h| \leqslant p_1$ for all nodes h of $\mathbf{H}$ at level 1,
- $B_h \subseteq B_{h'}$ and $|B_{h'} \backslash B_h| \leqslant p_{i+1}$ for all nodes h, h' of $\mathbf{H}$ at levels i and $i+1$, respectively, with $(h, h') \in E^{\mathbf{H}}$.

The *level* of a node h in $\mathbf{H}$ is the number of vertices in the unique path from the root (of h's component) to h. So, roots are considered to be at level 1.

Theorem 6.1. *Let $\mathbf{A}$ be a structure, and v be a game vector. The following are equivalent.*

1. *The v-game solves p-*$\textsc{Hom}(\{\mathbf{A}\})$.
2. *The graph* $\mathsf{graph}(\mathsf{core}(\mathbf{A}))$ *has a v-decomposition.*

This theorem immediately implies that, given a structure $\mathbf{A}$ and a game vector v, it is decidable whether or not the v-game solves p-$\textsc{Hom}(\{\mathbf{A}\})$. Another immediate consequence of this theorem is that for any class of structures $\mathbf{A}$, the v-game solves p-$\textsc{Hom}(\mathcal{A})$ if and only if each graph of the form $\mathsf{graph}(\mathsf{core}(\mathbf{A}))$, with $\mathbf{A} \in \mathcal{A}$, has a v-decomposition.

We will provide the proof of this theorem after presenting two further theorems built on it. These two theorems analyze two natural measures associated with our pebble games: number of pebbles and number of rounds. We show that these two measures correspond exactly (± 1) to tree depth and stack depth; this is made precise as follows.

Theorem 6.2 (Correspondence between number of pebbles and tree depth)**.** *Let $\mathcal{A}$ be a class of structures, let $\mathcal{G}$ denote $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$, and let $n \geqslant 1$. The following are equivalent.*

1. *There exists a game vector v with n pebbles such that the v-game solves p-*Hom*$(\mathcal{A})$.*
2. *The $\underbrace{(1,\ldots,1)}_{n \text{ times}}$-game solves p-*Hom*$(\mathcal{A})$.*
3. *The class $\mathcal{G}$ has tree depth $< n$.*

Proof. ($1 \Leftrightarrow 2$): The implication from 2 to 1 is immediate. To prove that 1 implies 2, let $v = (p_1, \ldots, p_r)$ be a game vector with $\sum_{i \in [r]} p_i = n$; it suffices to show that if there is a Duplicator winning strategy $W_1, \ldots, W_n$ for the $(1, \ldots, 1)$-game, then there is a Duplicator winning strategy for the v-game. The sequence $W_{p_1}, W_{p_1+p_2}, \ldots, W_{p_1+\cdots+p_r}$ is straightforwardly verified to be a Duplicator winning strategy for the v-game.

($2 \Leftrightarrow 3$): By appeal to Theorem 6.1, it suffices to show that each graph in $\mathcal{G}$ has a $(1, \ldots, 1)$-decomposition if and only if condition 3 holds. For the forward direction, let $(B_h)_{h \in H}$ be a $(1, \ldots, 1)$-decomposition of a graph $\mathbf{G} \in \mathcal{G}$ with respect to the rooted forest $\mathbf{H}$. We process $\mathbf{H}$ by removing any vertex $h \in H$ with $B_h = \emptyset$ and by contracting together adjacent vertices $h, h' \in H$ with $B_h = B_{h'}$, We obtain that each root of $\mathbf{H}$ has $|B_h| = 1$ and that each node h' having a parent h satisfies $B_{h'} \supseteq B_h$ and $|B_{h'}| = |B_h| + 1$. Rename each root $h' \in H$ by the unique element in $B_{h'}$ and rename each $h' \in H$ with a parent h by the unique element that is in $B_{h'} \backslash B_h$; then, it holds that the new $\mathbf{H}$ has height $< n$, and witnesses that the tree depth of $\mathbf{G}$ is $< n$. For the backward direction, let $\mathbf{T}$ be a rooted tree of height $< n$ witnessing that a component C of a graph $\mathbf{G} \in \mathcal{G}$ has tree depth $< n$. Then C has a $(1, \ldots, 1)$-decomposition given by $(B_t)_{t \in T}$ defined by $B_t = \{a \mid a \text{ is an ancestor of } t\}$; this is straightforwardly verified. Combining the $(1, \ldots, 1)$-decompositions of the components of $\mathbf{G}$, we obtain the desired decomposition of $\mathbf{G}$. ☐

Theorem 6.3 (Correspondence between number of rounds and stack depth)**.** *Let $\mathcal{A}$ be a class of structures, let $\mathcal{G}$ denote $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$, and let $r \geqslant 0$. The following are equivalent.*

1. *There exists a game vector v with $r+1$ rounds such that the v-game solves p-*Hom*$(\mathcal{A})$.*
2. *$\mathcal{G} \preceq \mathcal{F}_r$.*
3. *The class $\mathcal{G}$ has bounded tree depth and has stack depth $\leqslant r$.*

Proof. ($1 \Leftrightarrow 2$): By appeal to Theorem 6.1, it suffices to show that each graph in $\mathcal{G}$ has a v-decomposition if and only if $\mathcal{G} \preceq \mathcal{F}_r$. For the forward direction, let $v = (p_1, \ldots, p_{r+1})$. By definition, a v-decomposition is a $\mathbf{H}$-decomposition for a rooted forest $\mathbf{H}$ of height $\leqslant r$. Also, it follows from the definition that each bag B_h has size $|B_h| \leqslant p_1 + \cdots + p_r$. Thus $\mathcal{G}$ has $\mathcal{F}_r$-decompositions of bounded width, and by Proposition 3.10, we obtain $\mathcal{G} \preceq \mathcal{F}_r$.

For the backward direction, by Proposition 3.10, there exists $w \geqslant 1$ such that $\mathcal{G}$ has $\mathcal{F}_r$-decompositions of width $< w$. When $\mathbf{H} \in \mathcal{F}_r$ and $(B_h)_{h \in H}$ is a $\mathbf{H}$-decomposition of width $< w$ for a graph $\mathbf{G}$, define $(B'_h)_{h \in H}$ by $B'_h = \bigcup_a B_a$ where the union is over all ancestors a of h. It is readily seen that $(B'_h)_{h \in H}$ is a $\underbrace{(w, \ldots, w)}_{r+1}$-decomposition of $\mathbf{G}$.

($2 \Leftrightarrow 3$): By Proposition 3.17, whenever condition 2 holds, $\mathcal{G}$ has bounded tree depth. Hence, the desired equivalence follows from Lemma 3.23. □

6.1 Proof of Theorem 6.1

To prove Theorem 6.1 we need to develop the theory of the introduced pebble games.

Proposition 6.4. *The $\xrightarrow{v}$ relation is transitive, that is, when $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are similar structures, if $\mathbf{A} \xrightarrow{v} \mathbf{B}$ and $\mathbf{B} \xrightarrow{v} \mathbf{C}$, then $\mathbf{A} \xrightarrow{v} \mathbf{C}$.*

Proof. Let $W_1, \ldots, W_r$ be a Duplicator winning strategy for the v-game on $(\mathbf{A}, \mathbf{B})$, and let $V_1, \ldots, V_r$ be a Duplicator winning strategy for the v-game on $(\mathbf{B}, \mathbf{C})$. For each $i \in [r]$, define U_i to be the set $\{g \circ f \mid f \in W_i, g \in V_i, \mathsf{dom}(g) = \mathsf{im}(f)\}$. It is straightforward to verify that $U_1, \ldots, U_r$ is a Duplicator winning strategy for the v-game on $(\mathbf{A}, \mathbf{C})$. □

Proposition 6.5. *If $\mathbf{A}$ and $\mathbf{B}$ are structures such that there exists a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$, then for any game vector v, there is a Duplicator winning strategy for the v-game on $(\mathbf{A}, \mathbf{B})$.*

Proof. Such a strategy is given by taking W_i to be the set containing each restriction of h to a subset $S \subseteq A$ with $|S| \leqslant p_1 + \cdots + p_i$. □

Theorem 6.6. *Let $v = (p_1, \ldots, p_r)$ be a game vector, let $\mathbf{T}, \mathbf{B}$ be similar structures, and assume that $\mathsf{graph}(\mathbf{T})$ has a v-decomposition. If there exists a Duplicator winning strategy for the v-game on $(\mathbf{T}, \mathbf{B})$, then there is a homomorphism from $\mathbf{T}$ to $\mathbf{B}$.*

Proof. Let $\mathbf{H}$ be a forest such that $(B_h)_{h \in H}$ gives a v-decomposition of $\mathsf{graph}(\mathbf{T})$. For each $h \in H$, we define a map f_h from B_h to $\mathbf{B}$ that is a partial homomorphism from $\mathbf{T}$ to $\mathbf{B}$, in the following inductive manner. Let $W_1, \ldots, W_r$ be a Duplicator winning strategy for the v-game. For each root h_0 of $\mathbf{H}$, define f_{h_0} to be a map in W_1 that is defined on B_{h_0}. When node h' is the child (in $\mathbf{H}$) of a node h at level i having f_h defined, we define $f_{h'}$ to be a map in W_{i+1} that is defined on $B_{h'}$ and that extends f_h; such a map exists by the definition of Duplicator winning strategy and by the definition of v-decomposition. For every element $t \in T$, by the definition of $\mathbf{H}$-decomposition, there exists an $h \in H$ such that $t \in \mathsf{dom}(f_h)$. If for an element $t \in T$ it holds that $t \in \mathsf{dom}(f_h) \cap \mathsf{dom}(f_{h'})$ where h is the parent of h' in $\mathbf{H}$, by the way in which we defined the mappings f_h, $f_{h'}$, it holds that $f_h(t) = f_{h'}(t)$. It follows by the connectivity condition of a decomposition that for any $h, h' \in H$ such that $t \in \mathsf{dom}(f_h) \cap \mathsf{dom}(f_{h'})$, one has $f_h(t) = f_{h'}(t)$. Thus $f := \bigcup_{h \in H} f_h$ is a map from T to B, and in fact a homomorphism from $\mathbf{T}$ to $\mathbf{B}$: for any tuple of a relation of $\mathbf{T}$, its entries are contained in a bag B_h, and f extends f_h, which is a partial homomorphism from $\mathbf{T}$ to $\mathbf{B}$ defined on B_h. □

For each structure $\mathbf{A}$ and each game vector $v = (p_1, \ldots, p_r)$, we define the structure $\mathbf{T}_v(\mathbf{A})$ as follows. Let us say that a tuple $(C_1, \ldots, C_m)$ is a *set vector* (of v in A) if $1 \leqslant m \leqslant r$, $C_1 \subseteq \cdots \subseteq C_m \subseteq A$, $|C_1| \leqslant p_1$, and for each $i \in [m-1]$, it holds that $|C_{i+1} \backslash C_i| \leqslant p_{i+1}$. Let $S = S(v, A)$ be the set of all set vectors of v in A. For a set vector $s = (C_1, \ldots, C_m)$, when $a \in C_m$, define $u(a, s)$ to be the smallest prefix of s with $a \in C_{|u(a,s)|}$ (equivalently, with $a \in C_{|u(a,s)|} \cap \cdots \cap C_m$), and define B_s to be the set $\{(a, u(a, s)) \mid a \in C_m\}$. The universe of $\mathbf{T}_v(\mathbf{A})$ is $\bigcup_{s \in S} B_s$, and every symbol R from the vocabulary of $\mathbf{A}$ is interpreted by

$$\bigcup_{s \in S} \Big\{ ((a_1, u_1), \ldots, (a_{\mathrm{ar}(R)}, u_{\mathrm{ar}(R)})) \in B_s^{\mathrm{ar}(R)} \mid (a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}} \Big\}.$$

Proposition 6.7. *For every game vector v and every structure $\mathbf{A}$, the graph $\mathsf{graph}(\mathbf{T}_v(\mathbf{A}))$ has a v-decomposition.*

Proof. We use the notation in the definition of $\mathbf{T}_v(\mathbf{A})$ above. One can naturally define a forest with vertices S (the set of all set vectors of v in A) by making two vectors $s, s' \in S$ adjacent if and only if s' extends s by one entry (or vice-versa). Taking as roots the length 1 set vectors, this gives a rooted forest $\mathcal{S}$. We claim that with respect to this rooted forest, $(B_s)_{s \in S}$ is a v-decomposition of $\mathsf{graph}(\mathbf{T}_v(\mathbf{A}))$.

Let $((a, u), (a', u')) \in \mathsf{refl}(E^{\mathsf{graph}(\mathbf{T}_v(\mathbf{A}))})$. If $(a, u) = (a', u')$, then $\{(a, u), (a', u')\} \in B_u$. If $((a, u), (a', u')) \in E^{\mathsf{graph}(\mathbf{T}_v(\mathbf{A}))}$, then there are $s \in S$, $(a_1, \ldots, a_{\mathrm{ar}(R)}) \in R^{\mathbf{A}}$ and $i, j \in [\mathrm{ar}(R)]$ such that $(a, u) = (a_i, u(a_i, s))$ and $(a', u') = (a_j, u(a_j, s))$. Assume $|u(a_i, s)| \leqslant |u(a_j, s)|$ (the case $|u(a_j, s)| \leqslant |u(a_i, s)|$ is symmetric). Then $a \in C_{|u|} \subseteq C_{|u'|} \ni a'$, so $\{a, a'\} \in C_{|u'|}$ and hence $(a, u), (a', u') \in B_{u'}$. To see connectivity of the decomposition, note that every (a, u) appears precisely in those B_s such that u is a prefix of s, and the set of these s is connected in $\mathcal{S}$. Thus $(B_s)_{s \in S}$ is an $\mathcal{S}$-decomposition of $\mathsf{graph}(\mathbf{T}_v(\mathbf{A}))$.

Further, every root of $\mathcal{S}$ is a length 1 set vector $s = (C)$, so $|C| = |B_s| \leqslant p_1$. If s ands s' are adjacent of levels i and $i + 1$ respectively, then s has length i and s' extends $s = (C_1, \ldots, C_i)$ by a set C. Then $|B_{s'} \backslash B_s| = |C \backslash C_i| \leqslant p_{i+1}$. Thus $(B_s)_{s \in S}$ is a v-decomposition of $\mathbf{T}_v(\mathbf{A})$. $\square$

Proposition 6.8. *For each game vector v and each structure $\mathbf{A}$, there is a homomorphism from $\mathbf{T}_v(\mathbf{A})$ to $\mathbf{A}$.*

Proof. The homomorphism is the projection onto the first coordinate, that is, the mapping that sends an element (a, u) of the universe of $\mathbf{T}_v(\mathbf{A})$ to a. $\square$

Theorem 6.9. *Let $\mathbf{A}, \mathbf{B}$ be similar structures with vocabulary σ, and let $v = (p_1, \ldots, p_r)$ be a game vector. The following are equivalent.*

1. *There exists a Duplicator winning strategy for the v-game on $(\mathbf{A}, \mathbf{B})$.*
2. *For every σ-structure $\mathbf{T}$ such that $\mathsf{graph}(\mathbf{T})$ has a v-decomposition: if there is a homomorphism from $\mathbf{T}$ to $\mathbf{A}$, then there is a homomorphism from $\mathbf{T}$ to $\mathbf{B}$.*
3. *There exists a homomorphism from $\mathbf{T}_v(\mathbf{A})$ to $\mathbf{B}$.*

Proof. ($1 \Rightarrow 2$): Suppose that $\mathbf{A} \xrightarrow{v} \mathbf{B}$ and $\mathbf{T} \xrightarrow{h} \mathbf{A}$ and that $\mathsf{graph}(\mathbf{T})$ has a v-decomposition. We have to show $\mathbf{T} \xrightarrow{h} \mathbf{B}$. By Proposition 6.5, we have $\mathbf{T} \xrightarrow{v} \mathbf{A}$. By Proposition 6.4, we have $\mathbf{T} \xrightarrow{v} \mathbf{B}$. We obtain $\mathbf{T} \xrightarrow{h} \mathbf{B}$ by Theorem 6.6.

($2 \Rightarrow 3$): This is immediate from Propositions 6.7 and 6.8.

($3 \Rightarrow 1$): Let h be a homomorphism from $\mathbf{T}_v(\mathbf{A})$ to $\mathbf{B}$. For each $i \in [r]$, define $W_i = \{h_s^+ \mid s \text{ is a set vector of length } i\}$ where h_s^+ is the map defined on $\pi_1(B_s)$ that takes $a \in \pi_1(B_s)$ to $h(a, u(a, s))$. By the definition of $\mathbf{T}_v(\mathbf{A})$, each W_i contains only partial homomorphisms.

We verify that the W_i give a Duplicator winning strategy, as follows. First, when $C \subseteq A$ and $|C| \leqslant p_1$, we have that W_1 contains a map defined on C, via the set vector $s = (C)$. Next, suppose that $i \in [r-1]$, that $g \in W_i$, and that C is a superset of $\mathsf{dom}(g)$ with $|C \backslash \mathsf{dom}(g)| \leqslant p_{i+1}$. By the definition of W_i, there exists a set vector s of length i such that $g = h_s^+$. Let s' be equal to the length $(i+1)$ set vector that extends s with C. The mapping $h_{s'}^+$ has domain C, and extends $h_s^+ = g$ because for each $a \in \mathsf{dom}(g)$, it holds that $u(a, s) = u(a, s')$. $\square$

Let $\mathbf{A}$ be a structure, and let v be a game vector. Let $\textsc{Hom}(\mathbf{A})$ denote the classical problem $\{\mathbf{B} \mid \mathbf{A} \xrightarrow{h} \mathbf{B}\}$. We say that *the v-game solves* $\textsc{Hom}(\mathbf{A})$ if for every structure $\mathbf{B}$ similar to $\mathbf{A}$, the existence of a Duplicator winning strategy for the v-game on $(\mathbf{A}, \mathbf{B})$ implies that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. For a class of structures $\mathcal{A}$, we say that *the v-game solves* p-$\textsc{Hom}(\mathcal{A})$ if, for each $\mathbf{A} \in \mathcal{A}$, the v-game solves $\textsc{Hom}(\mathbf{A})$.

Theorem 6.10 (Generalization of Theorem 6.1)**.** *Let $\mathbf{A}$ be a structure, and let $v = (p_1, \ldots, p_r)$ be a game vector. The following are equivalent.*

1. *The v-game solves* $\textsc{Hom}(\mathbf{A})$*.*
2. *There exists a homomorphism from $\mathbf{A}$ to $\mathbf{T}_v(\mathbf{A})$.*
3. *The graph* $\mathsf{graph}(\mathsf{core}(\mathbf{A}))$ *has a v-decomposition.*

Proof. ($1 \Rightarrow 2$): By the equivalence of 1 and 3 in Theorem 6.9, and the trivial fact that $\mathbf{T}_v(\mathbf{A}) \xrightarrow{h} \mathbf{T}_v(\mathbf{A})$ we have $\mathbf{A} \xrightarrow{v} \mathbf{T}_v(\mathbf{A})$. Then $\mathbf{A} \xrightarrow{h} \mathbf{T}_v(\mathbf{A})$ by the assumption that the v-game solves p-$\textsc{Hom}(\mathbf{A})$.

($2 \Rightarrow 3$): If $\mathbf{A} \xrightarrow{h} \mathbf{T}_v(\mathbf{A})$ then there is a homomorphism h from $\mathsf{core}(\mathbf{A})$ to $\mathbf{T}_v(\mathbf{A})$. By Proposition 6.8 $\mathbf{T}_v(\mathbf{A}) \xrightarrow{h} \mathbf{A}$, and clearly $\mathbf{A} \xrightarrow{h} \mathsf{core}(\mathbf{A})$, so there is a a homomorphism h' from $\mathbf{T}_v(\mathbf{A})$ to $\mathsf{core}(\mathbf{A})$. It follows that $h' \circ h$ is a homomorphism from $\mathsf{core}(\mathbf{A})$ to itself and thus has to be injective. Hence h is injective. By Proposition 6.7, we know that $\mathbf{T}_v(\mathbf{A})$ has a v-decomposition $(B_s)_{s \in S}$. It is straightforward to verify that $(B'_s)_{s \in S}$ defined by $B'_s = \{a \in \mathsf{core}(A) \mid h(a) \in B_s\}$ is a v-decomposition of $\mathsf{graph}(\mathsf{core}(\mathbf{A}))$; here, $\mathsf{core}(A)$ denotes the universe of $\mathsf{core}(\mathbf{A})$.

($3 \Rightarrow 1$): Suppose $\mathsf{graph}(\mathsf{core}(\mathbf{A}))$ has a v-decomposition and $\mathbf{A} \xrightarrow{v} \mathbf{B}$; we want to show that $\mathbf{A} \xrightarrow{h} \mathbf{B}$. Since $\mathsf{core}(\mathbf{A}) \xrightarrow{h} \mathbf{A}$, we have $\mathsf{core}(\mathbf{A}) \xrightarrow{v} \mathbf{A}$ (by Proposition 6.5). By the transitivity of $\xrightarrow{v}$ (Proposition 6.4), we have $\mathsf{core}(\mathbf{A}) \xrightarrow{v} \mathbf{B}$, and by Theorem 6.6, we obtain $\mathsf{core}(\mathbf{A}) \xrightarrow{h} \mathbf{B}$. As $\mathbf{A} \xrightarrow{h} \mathsf{core}(\mathbf{A})$, we conclude that $\mathbf{A} \xrightarrow{h} \mathbf{B}$. $\square$

7 The homomorphism problems in L

For a class of structures $\mathcal{A}$ consider the classical problem

$$\textsc{Hom}(\mathcal{A}) := \big\{(\mathbf{B}, \mathbf{A}) \mid \mathbf{A} \in \mathcal{A} \ \& \ \mathbf{A} \xrightarrow{h} \mathbf{B}\big\}.$$

In this section we prove the following characterization of the $\textsc{Hom}(\mathcal{A})$ problems solvable in classical logarithmic space.

Theorem 7.1. *Assume that* PATH $\neq$ para-L *and let* $\mathcal{A}$ *be a class of structures with bounded arity. The following are equivalent.*

1. $\mathcal{A} \in \mathrm{L}$ *and* $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$ *has bounded tree depth.*
2. $\textsc{Hom}(\mathcal{A}) \in \mathrm{L}$.

This characterization is conditional on the hypothesis PATH $\neq$ para-L, an hypothesis from parameterized complexity theory. The class PATH has been studied in previous works [21, 11] and is defined as follows.

Definition 7.2. A parameterized problem $Q \subseteq \{0,1\}^* \times \mathbb{N}$ is in PATH if and only if there are a computable $f : \mathbb{N} \to \mathbb{N}$ and a nondeterministic Turing machine $\mathbb{A}$ that accepts Q and for all inputs (x, k) and all runs on it uses space $O(f(k) + \log |x|)$ and performs at most $O(f(k) \cdot \log |x|)$ nondeterministic steps.

One can argue that PATH is a natural and important parameterized complexity class, e.g. some fundamental problems which turn out to be complete for PATH under $\leqslant_{pl}$. The above characterization further underlines its importance. We refer to [21, 11, 14] for more information. Here, let us mention the following result.

Proposition 7.3 ([11])**.** *p-$\textsc{Hom}(\mathcal{P}^*)$ is complete for* PATH *under* $\leqslant_{pl}$.

Let us also mention that, as made precise in [14], the collapse of PATH to para-L would imply that Savitch's theorem can be improved.

Lemma 7.4. *For every game vector v, there exists a logarithmic space algorithm that, given a pair $(\mathbf{A}, \mathbf{B})$ of similar structures, decides whether $\mathbf{A} \xrightarrow{v} \mathbf{B}$.*

Proof. Write $v = (p_1, \ldots, p_r)$ and $\ell := \sum_{i \in [r]} p_i$. For an instance $(\mathbf{A}, \mathbf{B})$ of $\textsc{Hom}(\mathcal{A})$ consider the following Boolean formula in the variables $X(\bar{a}, \bar{b})$ for $\bar{a} \in A^\ell, b \in B^\ell$:

$$\bigwedge\nolimits_{\bar{a}_1 \in A^{p_1}} \bigvee\nolimits_{\bar{b}_1 \in B^{p_1}} \bigwedge\nolimits_{\bar{a}_2 \in A^{p_2}} \bigvee\nolimits_{\bar{b}_2 \in B^{p_2}} \cdots \bigwedge\nolimits_{\bar{a}_r \in A^{p_r}} \bigvee\nolimits_{\bar{b}_r \in B^{p_r}} X(\bar{a}_1 \cdots \bar{a}_r, \bar{b}_1 \cdots \bar{b}_r).$$

Further, consider the assignment evaluating $X(\bar{a}_1 \cdots \bar{a}_r, \bar{b}_1 \cdots \bar{b}_r)$ by 1 or 0 depending on whether $\{(a_i, b_i) \mid i \in [\ell]\}$ is a partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$ or not. This assignment satisfies the formula if and only if $\mathbf{A} \xrightarrow{v} \mathbf{B}$. Both the formula and the assignment are computable from $(\mathbf{A}, \mathbf{B})$ in logarithmic space, and so is the truth value. $\square$

Proof of Theorem 7.1. ($1 \Rightarrow 2$) Let $d \geqslant 1$ bound the tree depth of $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$. By Theorem 6.2, the v-game solves $\textsc{Hom}(\mathcal{A})$ where $v := (\underbrace{1, \ldots, 1}_{d \text{ times}})$. It follows that $\textsc{Hom}(\mathcal{A}) = \{(\mathbf{B}, \mathbf{A}) \mid \mathbf{A} \in \mathcal{A} \;\&\; \mathbf{A} \xrightarrow{v} \mathbf{B}\}$. This is in L by Lemma 7.4 and the assumption $\mathcal{A} \in$ L.

($2 \Rightarrow 1$) Clearly, (2) implies $\mathcal{A} \in$ L. For contradiction, assume $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$ has unbounded tree depth. Then $\mathcal{P} \preceq \mathsf{graph}(\mathsf{core}(\mathcal{A}))$ by Lemma 3.18. By Theorems 5.9 and 5.10 (and Lemma 5.8) we get p-$\textsc{Hom}(\mathcal{P}^*) \leqslant_{pl} p$-$\textsc{Hom}(\mathcal{A})$. But (2) implies p-$\textsc{Hom}(\mathcal{P}^*) \in$ para-L, and this contradicts the assumption PATH $\neq$ para-L by Proposition 7.3. $\square$

8 Model checking existential sentences

In this section we study the complexity of the parameterized model-checking problems associated with sets of (first-order) sentences Φ, namely

$$p\text{-MC}(\Phi) := \big\{(\mathbf{A}, \ulcorner\varphi\urcorner) \mid \varphi \in \Phi \;\&\; \mathbf{A} \models \varphi\big\}.$$

Here, $\ulcorner\varphi\urcorner$ is a natural number coding in some straightforward sense the sentence φ. An *existential* sentence is one in which the quantifier $\forall$ does not occur and negation symbols $\neg$ appear only in front of atoms. A *primitive positive* sentence is one built from atoms by means of $\wedge$ and $\exists$. For $q, r \in \mathbb{N}$ let $\Sigma_1^q[r]$ and $\mathrm{PP}^q[r]$ denote the sets of existential and, respectively, primitive positive sentences of quantifier rank at most q where all appearing relation symbols have arity at most r.

The goal of this section is to prove the following.

Theorem 8.1. *Let $q, r \in \mathbb{N}, q \geqslant 1, r \geqslant 2$. Then*

$$p\text{-MC}(\Sigma_1^q[r]) \equiv_{qfap} p\text{-MC}(\mathrm{PP}^q[2]) \equiv_{qfap} p\text{-}\textsc{Hom}(\mathcal{F}_{q-1}^*).$$

We devide the proof into several lemmas.

Lemma 8.2. *Let $q \in \mathbb{N}, q \geqslant 1$. Then p-$\textsc{Hom}(\mathcal{F}_{q-1}^*) \leqslant_{qfap} p$-MC$(\mathrm{PP}^q[2])$.*

Proof. The proof proceeds by standard means defining a "canonical query" [8]. Details follow. Given a tree $\mathbf{T} \in \mathcal{T}$ and $r \in T$ we define a primitive positive formula $\varphi_{\mathbf{T},r}(x)$ such that for every structure $\mathbf{A}$ similar to $\mathbf{T}^*$ and every $a \in A$

$$\mathbf{A} \models \varphi_{\mathbf{T},r}(a) \iff \text{there exists a homomorphism } h \text{ from } \mathbf{T}^* \text{ to } \mathbf{B} \text{ with } h(r) = a;$$

moreover, $\mathsf{qr}(\varphi_{\mathbf{T},r}) = h$ for h the height of $\mathbf{T}$ when rooted at r. We give the definition by induction on h. For $h = 0$ the tree $\mathbf{T}$ consists of one node r, and we set $\varphi_{\mathbf{T},r}(x) = C_r x$. Otherwise, let $t_1 \ldots, t_\ell$ list the neighbors of r in $\mathbf{T}$. For $i \in [\ell]$ let $\mathbf{T}_i$ denote the connected component of $\langle T \backslash \{r\} \rangle^{\mathbf{T}}$ containing t_i. Then $\mathbf{T}_i$ rooted at t_i has height at most $h - 1$, so $\varphi_{\mathbf{T}_i,t_i}(y)$ is defined and we can set

$$\varphi_{\mathbf{T},r}(x) := C_r x \wedge \textstyle\bigwedge_{i \in [\ell]} \exists y (Exy \wedge \varphi_{\mathbf{T}_i,t_i}(y)).$$

Given an instance $(\mathbf{A}, \ulcorner\mathbf{F}^{*}\urcorner)$ of p-$\textsc{Hom}(\mathcal{F}^{*}_{q-1})$, the forest $\mathbf{F}$ is the disjoint union of, say, c many trees $\mathbf{T}_1, \ldots, \mathbf{T}_c \in \mathcal{T}_{q-1}$. For $i \in [c]$ choose a root $r_i \in T_i$ witnessing that $\mathbf{T}_i$ has height at most $q-1$ and set $\psi := \bigwedge_{i \in [c]} \exists x \varphi_{\mathbf{T}_i, r_i}(x)$. Then $(\mathbf{A}, \ulcorner\mathbf{F}^{*}\urcorner) \mapsto (\mathbf{A}, \ulcorner\psi\urcorner)$ is a reduction as desired. □

Let $\mathrm{DPP}^q[r]$ denote the set of disjunctions of sentences from $\mathrm{PP}^q[r]$.

Lemma 8.3. *Let $q, r \in \mathbb{N}$. Then p-$\mathrm{MC}(\Sigma_1^q[r]) \leqslant_{qfap} p$-$\mathrm{MC}(\mathrm{DPP}^q[r])$.*

Proof. Given an instance $(\mathbf{A}, \ulcorner\varphi\urcorner)$ of p-$\mathrm{MC}(\Sigma_1^q[r])$, let τ denote the vocabulary of φ. For each $R \in \tau$ choose a new relation symbol $\overline{R}$ of the same arity. The reduction maps $(\mathbf{A}, \ulcorner\varphi\urcorner)$ to $(\mathbf{A}', \ulcorner\psi\urcorner)$ where $\mathbf{A}'$ expands $\mathbf{A}$ interpreting every new symbol $\overline{R}$ by $A^{\mathrm{ar}(R)} \backslash R^{\mathbf{A}}$. The sentence ψ is obtained from φ by replacing subformulas of the form $\neg R\bar{x}$ by $\overline{R}\bar{x}$, and then moving disjunctions out using the equivalence $\exists x(\chi \vee \psi) \equiv (\exists x\chi \vee \exists x\psi)$ and de Morgan rules. This preserves the quantifier rank. It is easy to see that $(\mathbf{A}, \ulcorner\varphi\urcorner) \mapsto (\mathbf{A}', \ulcorner\psi\urcorner)$ is a reduction as desired. □

The following lemma comprises the key step in the proof of Theorem 8.1. It heavily relies on the results from Sections 3 and 5.

Lemma 8.4. *Let $q, r \in \mathbb{N}, q \geqslant 1$. Then*

$$p\text{-}\mathrm{MC}(\mathrm{PP}^q[r]) \leqslant_{qfap} p\text{-}\textsc{Hom}(\mathcal{F}^{*}_{q-1}).$$

Proof. Given an instance $(\mathbf{B}, \ulcorner\varphi\urcorner)$ of p-$\mathrm{MC}(\mathrm{PP}^q[r])$ we can assume that the existential quantifiers in φ quantify pairwise distinct variables. Following a known construction of [8] define a structure $\mathbf{A}_\varphi$ interpreting the vocabulary τ of φ as follows. Its universe is the set of variables of φ and a relation $R \in \tau$ is interpreted by those tuples $\bar{x}$ such that the atom $R\bar{x}$ appears in φ. Then

$$\mathbf{B} \models \varphi \quad \Longleftrightarrow \quad \mathbf{A}_\varphi \xrightarrow{h} \mathbf{B}.$$

This shows p-$\mathrm{MC}(\mathrm{PP}^q[r]) \leqslant_{qfap} p$-$\textsc{Hom}(\mathcal{A})$ where $\mathcal{A} := \{\mathbf{A}_\varphi \mid \varphi \in \mathrm{PP}^q[r]\}$. Note that $\mathcal{A}$ is decidable and of bounded arity. We claim that p-$\textsc{Hom}(\mathcal{A}) \leqslant_{qfap} p$-$\textsc{Hom}(\mathcal{F}^{*}_{q-1})$.

We have p-$\textsc{Hom}(\mathcal{A}) \leqslant_{qfap} p$-$\textsc{Hom}(\mathcal{A}^{*})$ trivially, and p-$\textsc{Hom}(\mathcal{A}^{*}) \leqslant_{qfap} p$-$\textsc{Hom}(\mathsf{graph}(\mathcal{A})^{*})$ by Lemma 5.14, so it suffices to show p-$\textsc{Hom}(\mathsf{graph}(\mathcal{A})^{*}) \leqslant_{qfap} p$-$\textsc{Hom}(\mathcal{F}^{*}_{q-1})$. Applying Theorem 5.9 it suffices to show $\mathsf{graph}(\mathcal{A}) \preceq \mathcal{F}_{q-1}$.

We argue similarly as in [11, Theorem 3.12]. Define a graph on the universe of $\mathbf{A}_\varphi$, i.e. the variables of φ, by putting an edge between x and y if an only if $\exists x$ and $\exists y$ are consecutive quantifiers in the formula tree of φ. This graph is a forest $\mathbf{F}_\varphi$ of height at most $q-1$ when we root each component by the first-quantified variable in it, i.e. the variable which is closest to the root in the formula tree of φ.

The closure of $\mathbf{F}_\varphi$ contains $\mathsf{graph}(\mathbf{A}_\varphi)$: if $(x, y) \in E^{\mathsf{graph}(\mathbf{A}_\varphi)}$ then there exists $R \in \tau$ and a tuple $\bar{x}$ of variables containing x and y such that $R\bar{x}$ appears in φ; since φ is a sentence every variable in $\bar{x}$ and especially x and y are quantified in φ, so $\exists x$ and $\exists y$ appear on some branch in the formula tree of φ; this means there is a path from x to y in $\mathbf{F}_\varphi$, so (x, y) is in the closure of $\mathbf{F}_\varphi$.

It follows from Propositions 2.2 and 3.10 that $\mathsf{graph}(\mathbf{A}_\varphi)$ has an $\mathbf{F}_\varphi$-deconstruction of width at most $q-1$. $\square$

Remark 8.5. Recall that in general we allow our reductions r to output $(\emptyset, k)$ for some k. Without loss of generality this does not happen when reducing to a problem of the form p-HOM($\mathcal{A}^*$) for some class of structures $\mathcal{A}$. Namely, assume r on $(\mathbf{B}, k)$ outputs $(\mathbf{B}', \ulcorner \mathbf{A}^* \urcorner)$ and $\mathbf{B}'$ is possibly $\emptyset$; change r to output instead of $\mathbf{B}'$ the disjoint union of $\mathbf{B}'$ and one point that does not have any of the colours $C_a, a \in A$, of $\mathbf{A}^*$.

Lemma 8.6. *Let $q \geqslant 1, r \geqslant 2$. Then p-MC(DPP$^q[r]$) $\leqslant_{qfap}$ p-HOM($\mathcal{F}^*_{q-1}$).*

Proof. Let an instance $(\mathbf{B}, \ulcorner \varphi_1 \vee \cdots \vee \varphi_k \urcorner)$ of p-MC(DPP$^q[r]$) be given, where $k \geqslant 1$ and $\varphi_i \in \mathrm{PP}^q[r]$ for all $i \in [k]$. For $i \in [k]$ let $(\mathbf{B}_i, \ulcorner \mathbf{F}_i^* \urcorner)$ be the output of the reduction of the previous lemma. Then $\mathbf{F}_i \in \mathcal{F}_{q-1}$, so $\mathbf{F}_i$ is a disjoint union of, say, c_i many trees $\mathbf{T}_{i1}, \ldots, \mathbf{T}_{ic_i} \in \mathcal{T}_{q-1}$. By the Remark 8.5 we can assume that all $\mathbf{B}_i$ are structures, i.e. different from $\emptyset$. Then

$$\begin{aligned}
\mathbf{B} \models \varphi_1 \vee \cdots \vee \varphi_k &\iff \exists i \in [k] : \mathbf{B} \models \varphi_i \\
&\iff \exists i \in [k] : \mathbf{F}_i^* \xrightarrow{h} \mathbf{B}_i \\
&\iff \exists i \in [k]\ \forall j \in [c_i] : \mathbf{T}^*_{ij} \xrightarrow{h} \mathbf{B}_i \\
&\iff \forall \bar{\jmath} \in \textstyle\prod_{i \in [k]} [c_i]\ \exists i \in [k] : \mathbf{T}^*_{ij_i} \xrightarrow{h} \mathbf{B}_i, \qquad (2)
\end{aligned}$$

where we write $\bar{\jmath} = j_1 \cdots j_k$. We can assume that the $\mathbf{T}_{ij}$'s are pairwise disjoint and for each of them a root is chosen witnessing that it has height at most $q-1$. Fix some $\bar{\jmath} \in \prod_{i \in [k]} [c_i]$. Define the tree $\mathbf{T}_{\bar{\jmath}}$ as follows: take the (disjoint) union of the trees $\mathbf{T}_{1j_1}, \ldots, \mathbf{T}_{kj_k}$ and then merge their roots to a new node r. Then r witnesses that $\mathbf{T}_{\bar{\jmath}} \in \mathcal{T}_{q-1}$ and all $\mathbf{T}_{ij_i}, i \in [k]$, are subtrees of $\mathbf{T}_{\bar{\jmath}}$ pairwise intersecting precisely in r. We define a structure $\mathbf{B}^{\bar{\jmath}}$ such that

$$\exists i \in [k] : \mathbf{T}^*_{ij_i} \xrightarrow{h} \mathbf{B}_i \iff \mathbf{T}^*_{\bar{\jmath}} \xrightarrow{h} \mathbf{B}^{\bar{\jmath}}. \qquad (3)$$

The structure $\mathbf{B}^{\bar{\jmath}}$ is the disjoint union of the structures $\mathbf{B}_i^{\bar{\jmath}}, i \in [k]$, defined next. To ensure (3) it suffices that $\mathbf{B}_i^{\bar{\jmath}}$ satisfies

$$\mathbf{T}^*_{\bar{\jmath}} \xrightarrow{h} \mathbf{B}_i^{\bar{\jmath}} \iff \mathbf{T}^*_{ij_i} \xrightarrow{h} \mathbf{B}_i. \qquad (4)$$

Indeed, assuming (4) for every $i \in [k]$ we derive (3) as follows. If $\mathbf{T}^*_{ij_i} \xrightarrow{h} \mathbf{B}_i$ for some i, then $\mathbf{T}^*_{\bar{\jmath}} \xrightarrow{h} \mathbf{B}_i^{\bar{\jmath}}$ for such an i by (4) and, clearly, this implies $\mathbf{T}^*_{\bar{\jmath}} \xrightarrow{h} \mathbf{B}^{\bar{\jmath}}$. Conversely, assume $\mathbf{T}^*_{\bar{\jmath}} \xrightarrow{h} \mathbf{B}^{\bar{\jmath}}$. Since $\mathbf{T}^*_{\bar{\jmath}}$ is connected, any homomorphism from $\mathbf{T}^*_{\bar{\jmath}}$ to $\mathbf{B}^{\bar{\jmath}}$ has image in some $\mathbf{B}_i^{\bar{\jmath}}$, i.e. $\mathbf{T}^*_{\bar{\jmath}} \xrightarrow{h} \mathbf{B}_i^{\bar{\jmath}}$ for some i; for such an i then (4) implies $\mathbf{T}^*_{ij_i} \xrightarrow{h} \mathbf{B}_i$.

Given $\mathbf{B}_i$ it is not hard to define $\mathbf{B}_i^{\bar{\jmath}}$ such that (4) is satisfied. First forget all interpretations of symbols outside the vocabulary of $\mathbf{T}^*_{\bar{\jmath}}$: these are all colours C_t for $t \in F_i \backslash T_{ij_i}$ as well as C_{r_i} where r_i is the root chosen for $\mathbf{T}_{ij_i}$. To get a structure in the vocabulary of $\mathbf{T}^*_{\bar{\jmath}}$, we add interpretations of C_t for $t \in T_{\bar{\jmath}} \backslash T_{ij_i}$: all these C_t are interpreted by $C_{r_i}^{\mathbf{B}_i}$. Finally, we add loops to the elements in $C_{r_i}^{\mathbf{B}_i}$, i.e. set

$$E^{\mathbf{B}_i^{\bar{\jmath}}} := E^{\mathbf{B}_i} \cup \{(b,b) \mid b \in C_{r_i}^{\mathbf{B}_i}\}.$$

Then (4) is straightforwardly verified, and thus we know (3).

Finally, let $\mathbf{F}$ be the disjoint union of the trees $\mathbf{T}_{\bar{\jmath}}, \bar{\jmath} \in \prod_{i\in[k]}[c_i]$, that is, make disjoint copies of the $\mathbf{T}_{\bar{\jmath}}$'s and then take the union. For every $\bar{\jmath}$ every node $t \in T_{\bar{\jmath}}$ has a copy $t(\bar{\jmath})$ in F. We adapt $\mathbf{B}^{\bar{\jmath}}$ by renaming C_t by $C_{t(\bar{\jmath})}$, more precisely, let $\tilde{\mathbf{B}}^{\bar{\jmath}}$ be the structure with universe $B^{\bar{\jmath}}$ interpreting E by $E^{\mathbf{B}^{\bar{\jmath}}}$ and $C_{t(\bar{\jmath})}$ by $C_t^{\mathbf{B}^{\bar{\jmath}}}$ for $t \in T_{\bar{\jmath}}$. This structure is the disjoint union of $\tilde{\mathbf{B}}_i^{\bar{\jmath}}, i \in [k]$, where $\tilde{\mathbf{B}}_i^{\bar{\jmath}}$ is obtained by analoguous renaming applied to $\mathbf{B}_i^{\bar{\jmath}}$.

Then the disjoint union $\mathbf{B}'$ of the $\tilde{\mathbf{B}}^{\bar{\jmath}}, \bar{\jmath} \in \prod_{i\in[k]}[c_i]$, is similar to $\mathbf{F}^*$, and

$$\begin{aligned} \mathbf{B} \models \varphi_1 \vee \cdots \vee \varphi_k &\iff \forall \bar{\jmath} \in \textstyle\prod_{i\in[k]}[c_i] : \mathbf{T}_{\bar{\jmath}}^* \xrightarrow{h} \mathbf{B}^{\bar{\jmath}} \\ &\iff \mathbf{F}^* \xrightarrow{h} \mathbf{B}'. \end{aligned}$$

The first equivalence follows from (3) and (2), and the second is trivial.

It is clear that $\ulcorner \mathbf{F}^* \urcorner$ can be computed from $\ulcorner \varphi_1 \vee \cdots \vee \varphi_k \urcorner$. We show how to get a structure isomorphic to $\mathbf{B}'$ by a suitable interpretation from $\langle \mathbf{A}, \mathbf{B} \rangle$ where $\mathbf{A}$ is a suitable auxiliary structure. Choose (p, a, d) for the reduction used above, i.e. the one producing $(\mathbf{B}_i, \ulcorner \mathbf{F}_i^* \urcorner)$ from $(\mathbf{B}, \ulcorner \varphi_i \urcorner)$ for every $i \in [k]$. Choose $w \in \mathbb{N}$ such that d maps numbers to interpretations of dimension w. For $i \in [k]$ write $I_i := d(\ulcorner \varphi_i \urcorner)$ and note $I_i(\langle a(\ulcorner \varphi_i \urcorner), \mathbf{B} \rangle) = \mathbf{B}_i$. We define our auxiliary structure to be

$$\mathbf{A} := \Big\langle\, a(\ulcorner \varphi_1 \urcorner), \ldots, a(\ulcorner \varphi_k \urcorner),\ ([k] \times \textstyle\prod_{i\in[k]}[c_i])^* \,\Big\rangle.$$

Note $([k] \times \prod_{i\in[k]}[c_i])$ is the structure interpreting the empty language on the universe $[k] \times \prod_{i\in[k]}[c_i]$. In the following we let i range over $[k]$ and $\bar{\jmath}$ over $\prod_{i\in[k]}[c_i]$.

There is a (quantifier-free) interpretation of dimension 1 which produces an isomorphic copy of $\langle a(\ulcorner \varphi_i \urcorner), \mathbf{B} \rangle$ from $\langle \mathbf{A}, \mathbf{B} \rangle$. Composing (see Claim 2 in the proof of Lemma 5.7) with I_i gives an interpretation I_i' of dimension w such that $I_i'(\langle \mathbf{A}, \mathbf{B} \rangle) \cong \mathbf{B}_i$. Further, there is an interpretation of dimension 1 producing $\mathbf{B}_i^{\bar{\jmath}}$ from $\mathbf{B}_i$. Composing with I_i' gives an interpretation $I_i^{\bar{\jmath}}$ of dimension w such that $I_i^{\bar{\jmath}}(\langle \mathbf{A}, \mathbf{B} \rangle) \cong \mathbf{B}_i^{\bar{\jmath}}$. As $\mathbf{B}_i^{\bar{\jmath}}$ is obtained by renaming colours we get $\tilde{I}_i^{\bar{\jmath}}$ with $\tilde{I}_i^{\bar{\jmath}}(\langle \mathbf{A}, \mathbf{B} \rangle) \cong \tilde{\mathbf{B}}_i^{\bar{\jmath}}$. The structure we want to produce is the disjoint union of $(\tilde{I}_i^{\bar{\jmath}}(\langle \mathbf{A}, \mathbf{B} \rangle))_{i\bar{\jmath}}$. We use the following general claim.

Claim: Let $J_1, \ldots, J_\ell$ be quantifier-free interpretations of dimension w. Then there is a quantifier-free interpretation J of dimension $w + 1$ such that $J(\langle ([\ell])^*, \mathbf{A} \rangle)$ is defined whenever all $J_i(\mathbf{A}), i \in [\ell]$, are defined and $\neq \emptyset$, and then $J(\langle ([\ell])^*, \mathbf{A} \rangle)$ is isomorphic to the disjoint union of $J_i(\mathbf{A}), i \in [\ell]$.

Proof of Claim: Note $J_i \circ Pr_2$ is quantifier-free, has dimension w and produces $J_i(\mathbf{A})$ from $\langle ([\ell])^*, \mathbf{A} \rangle$ (see Claim 1 in the proof of Lemma 5.7). Write $(\varphi_R^j)_R$ for this interpretation and let $\bar{x}_i$ range over w-tuples of variables.

Define $J := (\psi_R)_R$ as follows.

$$\begin{aligned}
\psi_U(y_1\bar{x}_1) &:= \textstyle\bigvee_{j\in[\ell]}(C_j y_1 \wedge P_1(y_1) \wedge \varphi_U^j(\bar{x}_1)),\\
\psi_=(y_1\bar{x}_1, y_2\bar{x}_2) &:= \textstyle\bigvee_{j\in[\ell]}(C_j(y_1) \wedge C_j(y_2) \wedge \varphi_=^j(\bar{x}_1, \bar{x}_2)),\\
\psi_R(y_1\bar{x}_1, \dots, y_{\mathrm{ar}(R)}\bar{x}_{\mathrm{ar}(R)}) &:= \textstyle\bigvee_{j\in[\ell]}(\varphi_R^j(\bar{x}_1, \dots, \bar{x}_{\mathrm{ar}(R)}) \wedge \bigwedge_{i\in[\mathrm{ar}(R)]} C_j(y_i));
\end{aligned}$$

where we understand that φ_R^j is some inconsistent formula if R is not in the output vocabulary of J_j. $\dashv$

To finish the proof, the disjoint union of $(\tilde{I}_i^{\bar{\jmath}}(\langle \mathbf{A}, \mathbf{B}\rangle))_{i\bar{\jmath}}$ is produced by first producing $\langle([k] \times \prod_{i\in[k]}[c_i])^*, \langle \mathbf{A}, \mathbf{B}\rangle\rangle$ from $\langle \mathbf{A}, \mathbf{B}\rangle$ and composing with an interpretation J chosen for the $\tilde{I}_i^{\bar{\jmath}}$'s according to the claim. $\square$

Proof of Theorem 8.1. Applying Lemmas 8.3, 8.6 and 8.2 in row, we get

$$p\text{-MC}(\Sigma_1^q[r]) \leqslant_{qfap} p\text{-MC}(\mathrm{DPP}^q[r]) \leqslant_{qfap} p\text{-}\textsc{Hom}(\mathcal{F}_{q-1}^*) \leqslant_{qfap} p\text{-MC}(\mathrm{PP}^q[2]).$$

Noting $p\text{-MC}(\mathrm{PP}^q[2]) \subseteq p\text{-MC}(\Sigma_1^q[r])$, Theorem 8.1 follows. $\square$

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] A. Atserias, A. Bulatov, and V. Dalmau. On the power of k-consistency. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming*, pp, 279–290, 2007.

[3] A. Atserias and M. Weyer. Decidable relationships between consistency notions for constraint satisfaction problems. In *Proceedings of the 18th Annual Conference of the European Association for Computer Science Logic*, pp. 102–116, 2009.

[4] L. Barto and M. Kozik. Constraint satisfaction problems of bounded width. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, 2009.

[5] A. Blumensath and B. Courcelle. On the monadic second-order transduction hierarchy. *Logical Methods in Computer Science*, 6(2), 2010.

[6] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.

[7] A. A. Bulatov, A. A. Krokhin, and B. Larose. Dualities for constraint satisfaction problems. In *Complexity of Constraints*, pp. 93–124, 2008.

[8] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the ACM Symposium on Theory of Computing*, pp. 77–90, 1977.

[9] H. Chen and V. Dalmau. Beyond hypertree width: Decomposition methods without decompositions. In *11th International Conference on Principles and Practice of Constraint Programming*, pp. 167–181, 2005.

[10] H. Chen, V. Dalmau, and B. Grußien. Arc consistency and friends. *Journal of Logic and Computation* 23 (1): 87–108, 2013.

[11] H. Chen and M. Müller. The fine classification of conjunctive queries and parameterized logarithmic space complexity. *ACM Transactions on Computation Theory 6 (2)*, Article No. 7, 2014.

[12] H. Chen and M. Müller. One hierarchy spawns another: graph deconstructions and the complexity classification of conjunctive queries. Proceedings of the joint 23rd EACSL Computer Science Logic and 29th ACM/IEEE Symposium Logic in Computer Science, pp. 32:1–32:10, 2014.

[13] Y. Chen, J. Flum, and M. Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. Proceedings of the 18th IEEE Conference on Computational Complexity, pp.13-29, 2003.

[14] Y. Chen and M. Müller. Bounded variable logic, parameterized logarithmic space, and Savitch's theorem. In *Proceedings 39th International Symposium on Mathematical Foundations of Computer Science*, pp. 183–195, 2014.

[15] V. Dalmau. Linear datalog and bounded path duality of relational structures. *Logical Methods in Computer Science* 1(1), paper 5, 2005.

[16] V. Dalmau, P. G. Kolaitis, and M. Y. Vardi. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *Proceedings of Principles and Practice of Constraint Programming*, LNCS 2470, pp. 310–326, 2002.

[17] A. Dawar and Y. He. Parameterized complexity classes under logical reductions. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science*, pp. 258–269, 2009.

[18] R. Diestel. *Graph Theory, 4th Edition*. Graduate Texts in Mathematics 173, Springer, 2012.

[19] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[20] H.-D. Ebbinghaus and J. Flum. *Finite model theory*. Perspectives in Mathematical Logic, Springer, 1995.

[21] M. Elberfeld, C. Stockhusen, and T. Tantau. On the space complexity of parameterized problems. In *Proceedings of the 7th International Symposium of Parameterized and Exact Computation*, Springer LNCS 7535, pp. 206–217, 2012.

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

[23] J. Flum and M. Grohe. Describing parameterized complexity classes. *Information and Compututation* 187 (2): 291–319, 2003.

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

[25] G. Gottlob, N. Leone, and F. Scarcello. The complexity of acyclic conjunctive queries. *Journal of the ACM* 48 (3):431–498, 2001.

[26] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences* 64 (3):579–627, 2002.

[27] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM* 54(1), 2007.

[28] M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, 2001.

[29] N. Immerman. *Descriptive Complexity*. Springer, 1998.

[30] P. Kolaitis and M. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences* 61:302–332, 2000.

[31] Ph. G. Kolaitis and M. Y. Vardi. A game-theoretic approach to constraint satisfaction. In *Proceedings of the 17th National Conference on AI*, pp. 175–181, 2000.

[32] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 735–744, 2010.

[33] J. Nešetřil and P. Ossona de Mendez. Tree depth, subgraph coloring, and homomorphism bounds. *European Journal of Combinatorics* 27 (6): 1022–1041, 2006.

[34] C. Papadimitriou and M. Yannakakis. On the Complexity of Database Queries. *Journal of Computer and System Sciences* 58 (3): 407–427, 1999.

[35] N. Robertson and P.D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B* 35 (1): 39–61, 1983.

[36] N. Robertson and P.D Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B* 41 (1): 92-114, 1986.

[37] N. Schweikardt, T. Schwentick, and L. Segoufin. Database theory: Query languages. In M. J. Atallah and M. Blanton (eds.), *Algorithms and Theory of Computation Handbook*, vol. 2: Special Topics and Techniques, chapter 19. CRC Press, 2nd ed., 2009.