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

Hubie Chen

Departamento LSI, Universidad del País Vasco
and IKERBASQUE, Basque Foundation for Science
E-20018 San Sebastián, Spain
hubie.chen@ehu.es

Moritz Müller

Kurt Gödel Research Center
Universität Wien
Währingerstraße, 1090 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 and completely describe the resulting hierarchy given by this relation. This binary 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 reductions. 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.

Categories and Subject Descriptors F.4.1 [*Mathematical logic and formal languages*]: Mathematical logic—Computational logic; H.2.3 [*Database management*]: Languages—Query languages

General Terms Theory

Keywords Conjunctive queries, Homomorphisms, Graph decompositions, Parameterized complexity

CSL-LICS 2014, July 14–18, 2014, Vienna, Austria.

http://dx.doi.org/10.1145/2603088.2603107

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. 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 problems, including the homomorphism problem on relational structures, the constraint satisfaction problem, and conjunctive query containment [8, 25]. 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 [1, 3, 4, 7, 9, 21–23, 25, 27–29].

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 $p\text{-}\textsc{Hom}(\mathcal{A})$, for classes of relational structures $\mathcal{A}$: the problem $p\text{-}\textsc{Hom}(\mathcal{A})$ 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 $p\text{-}\textsc{Hom}(\mathcal{A})$ was presented by Grohe [23]. In particular, a known sufficient condition for fixed-parameter tractability of $p\text{-}\textsc{Hom}(\mathcal{A})$ was that the *cores* of $\mathcal{A}$ have *bounded treewidth* [13]; Grohe completed the picture by showing that for any class $\mathcal{A}$ not satisfying this condition, the problem $p\text{-}\textsc{Hom}(\mathcal{A})$ 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 $p\text{-}\textsc{Hom}(\mathcal{A})$ problems was presented [10]. 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-HOM($\mathcal{A}$) problem is para-L equivalent to p-HOM($\mathcal{T}^*$), para-L equivalent to p-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-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 present and study Ehrenfeucht-Fraïssé-style pebble games for solving the problems coming from the lower end of our hierarchy, and then use our framework to study the complexity of model checking existential sentences.

A graph-theoretic hierarchy. To derive our classification, we first focus on graphs. Previous work [10] 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-HOM($\mathcal{M}^*$) reduces to p-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 [24] in the theory of conjunctive queries is as follows:

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

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

$$\text{If } \mathcal{G} \leq \mathcal{H}, \text{ then } p\text{-HOM}(\mathcal{G}^*) \text{ reduces to } p\text{-HOM}(\mathcal{H}^*). \quad (1)$$

This key property is shown to imply the three just-named results. The definition of the relation $\leq$ (Definition 3.5) is simple and is based on a notion which we call *graph deconstruction*, which is strongly related to and inspired by the notion of tree decomposition.

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 of graph classes (Theorem 3.14):

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

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. To understand the upper levels of the hierarchy, we make use of known *excluded minor theorems*. To determine the lower part of the hierarchy (below $\mathcal{P}$), we introduce a new complexity measure on graph classes which we call *stack depth*; we characterize the stack depth of a graph class $\mathcal{G}$ as equivalent both to the maximum d such that all depth d trees are present as minors of $\mathcal{G}$, as well as to the minimum depth of forests that allow for bounded width decompositions of the graphs in $\mathcal{G}$ (Lemma 3.21). (The first value is the official definition of stack depth.)

A complexity-theoretic hierarchy. Having understood the relation $\leq$ on graph classes, we then turn to study homomorphism problems. As mentioned, we prove that $\mathcal{G} \leq \mathcal{H}$ implies p-HOM($\mathcal{G}^*$) reduces to p-HOM($\mathcal{H}^*$) (Theorem 4.9); this property was indeed a primary motivation for the definition of $\leq$. 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 [16], and Dawar and He studied a type of quantifier-free interpretation in the parameterized setting [14]. 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 [20].

Let $\mathcal{A}$ be an arbitrary class of bounded-arity structures. We prove that p-HOM($\mathcal{A}$) is equivalent under qfap-reductions to p-HOM($\mathcal{G}^*$), where $\mathcal{G}$ is the class of graphs of the cores of the structures in $\mathcal{A}$ (Theorem 4.10). By this theorem, property (1) and our description of the graph hierarchy, we obtain:

Every problem of the form p-HOM($\mathcal{A}$) is equivalent,
under qfap-reduction, to a problem p-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-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-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 [10, 23, 24], which derivation we find to be clearer and simpler than those of the original works. Our treatment also strengthens these 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 [23] are pairwise equivalent under qfap-reductions and hence in a very strong sense.

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 [3, 4, 7, 9, 13, 26]).

Seminal work of Koalitits and Vardi [26] showed that certain natural consistency algorithms could be viewed as determining the winner in certain Ehrenfeucht-Fraïssé type pebble games [16]. Since this work, there has been sustained effort devoted to presenting pebble games that solve cases of the homomorphism problem. For example, there is study of pebble games that solve p-HOM($\mathcal{A}$) when the class $\mathcal{G}$ of graphs of structures from $\mathcal{A}$ has bounded treewidth [2, 13], and also when $\mathcal{G}$ has bounded pathwidth [12].

We complete the picture by presenting natural pebble games that are shown to solve p-HOM$(\mathcal{A})$ when $\mathcal{G}$ has bounded tree depth (Section 5). 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 5.1). We also show equivalences (± 1) between the number of pebbles needed to solve p-HOM$(\mathcal{A})$ and the tree depth of $\mathcal{G}$; and, between the number of rounds needed to solve p-HOM$(\mathcal{A})$ and the stack depth of $\mathcal{G}$ (Section 5). We believe that the latter result reinforces the suggestion that stack depth is a natural graph-theoretic measure.

As a fruit of our development of pebble games, we obtain a characterization of the classical homomorphism problems HOM$(\mathcal{A})$ decidable in classical logarithmic space: these are precisely those where the cores of structures from $\mathcal{A}$ have bounded tree depth.

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. As an indication favoring this perspective, we show, in the final section, that the hierarchy can be used to classify the complexity of model-checking existential sentences having bounded quantifier rank.

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 with vocabulary* σ, for short, a *σ-structure* $\mathbf{B}$ 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 STR$[\sigma]$, and the class of all structures by STR.

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 $\emptyset$ 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 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 so is the graph $\mathsf{graph}(\mathbf{A})$. 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* 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:

- 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 (see [6]) 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. $\mathbf{T}$ it 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

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 $(h, h') \in E^{\mathbf{H}}$.

EXAMPLE 3.2. Let $n \geqslant 1$. When $\mathbf{H}$ is the n-by-n grid (defined in Section 2.2), 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 3.3. 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 3.4. *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 $\mathbf{H}$.*

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

- We write 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 write 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 3.5. 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 $\mathcal{G} \precneq \mathcal{H}$ if $\mathcal{G} \preceq \mathcal{H}$ and $\mathcal{G} \not\equiv \mathcal{H}$.

PROPOSITION 3.6. *The relation $\preceq$ is reflexive and transitive.*

This implies that $\equiv$ is an equivalence relation.

PROPOSITION 3.7. *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 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 3.8. *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.

REMARK 3.9. 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 should be considered ∞ if the set is infinite.

PROPOSITION 3.10. *A class of graphs has bounded stack depth if and only if it has bounded pathwidth.*

THEOREM 3.11. *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 of $\mathbf{G}$ is *nice* if it is of the form $(\mu(m))_{m \in M}$ such that the following hold:

- μ is a minor map from M to 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, $g \in \mu(m)$, $g' \in \mu(m')$, and $(g, g') \in E^{\mathbf{G}}$, then g' is a child of g.

LEMMA 3.12. *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 $\bigcup_{m \in \nu(n)} \mu(m)$.*

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 $d = 0$ or if $d > 0$ 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 if a rooted tree has property $P(d, k)$, then the tree contains $\mathbf{T}_{d,k}$ as a minor.

LEMMA 3.13. *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 $\mathbf{G}$.*

We are ready to prove Theorem 3.11.

Proof. (Theorem 3.11) 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 3.6). 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, we have that 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 tree $\mathbf{G}^i$ is contained in $\mathcal{G}'$, and 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 3.13, by induction, $\mathcal{H}$ has nice $\mathcal{T}_d$-deconstructions of bounded width. As a consequence of Lemma 3.12, we obtain that $\mathcal{G}$ has nice $\mathcal{T}_d$-deconstructions of bounded width. □

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. (cf. Section 2.2).

THEOREM 3.14. *(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.

PROPOSITION 3.15. *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 $\mathcal{G} \preceq \mathcal{F}_d$.*

LEMMA 3.16. *Let $\mathcal{G}$ be a class of graphs.*

1. *If $\mathcal{L} \npreceq \mathcal{G}$, then $\mathcal{G} \preceq \mathcal{T}$.*
2. *If $\mathcal{T} \npreceq \mathcal{G}$, then $\mathcal{G} \preceq \mathcal{P}$.*
3. *If $\mathcal{P} \npreceq \mathcal{G}$, then $\mathcal{G}$ has bounded tree depth.*

Proof. We begin with the third claim. Suppose that $\mathcal{P} \npreceq \mathcal{G}$. It holds by Proposition 3.7 that $\mathcal{P} \npreceq \mathsf{minors}(\mathcal{G})$, from which it follows that $\mathcal{P} \nsubseteq \mathsf{minors}(\mathcal{G})$ and hence, by Theorem 2.1, $\mathcal{G}$ has bounded tree depth.

To establish the second claim, we reason in an analogous way. If $\mathcal{T} \npreceq \mathcal{G}$, it follows from Proposition 3.7 and Theorem 2.1 that $\mathcal{G}$ has bounded pathwidth, and it follows from Proposition 3.15 that $\mathcal{G} \preceq \mathcal{P}$.

We can also prove the first claim in an analogous way, 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; it follows from this that $\mathcal{L} \npreceq \mathcal{G}$ implies $\mathcal{R} \npreceq \mathcal{G}$. That $\mathcal{L} \preceq \mathcal{R}$ follows from Example 3.2: each graph has $\mathcal{R}$-deconstructions of width $\leqslant 2$. □

LEMMA 3.17. *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* $d \geqslant 0$

LEMMA 3.18. *For each* $d \geqslant 0$*, the following hold:*

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

The following builds on [5, Lemma 4.8].

LEMMA 3.19. *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 that are minors of* $\mathcal{G}$*.)*

LEMMA 3.20. *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. 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 $\mathcal{C}$.

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 $\mathcal{G} \equiv \mathcal{F}_d$. It follows directly from Theorem 3.11 that $\mathcal{G} \preceq \mathcal{F}_d$. For each $k \geqslant 1$, let us use $k \times \mathbf{T}_{d,k}$ to denote the graph consisting of k disjoint copies of $\mathbf{T}_{d,k}$. By assumption, we have $\{k \times \mathbf{T}_{d,k} \mid k \geqslant 1\} \subseteq \mathsf{minors}(\mathcal{G})$. We also have that (up to isomorphism) each graph in $\mathcal{F}_d$ is a subgraph of a graph of the form $k \times \mathbf{T}_{d,k}$; hence, by Proposition 3.7, 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 3.11, 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 $(B^i_t)_{t \in T}$ of width $\leqslant v$; then, the disjoint union of the $\mathbf{G}_i$ has a $\mathbf{T}$-deconstruction of width $\leqslant vL$, namely, $(B^1_t \cup \cdots \cup B^L_t)_{t \in T}$. 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 3.21. *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. (Theorem 3.14) 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 3.17 and 3.18 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 3.16 (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 3.16 (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 3.16 (3). By Proposition 3.10 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 3.21. □

4. Complexity classification

4.1 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 [20], we say Q is in L *after a pre-computation* if there is a computable $a : \mathbb{N} \to \{0,1\}^*$ and a classical problem $P \subseteq \{0,1\}^*$ in L such that for all $(x,k) \in \{0,1\}^* \times \mathbb{N}$

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

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 [20] for the corresponding theory. E.g. 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 4.1. A *parameterized problem* is a subset $Q \subseteq$ 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. 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.[1]

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 [19] transferred capturing results (cf. [16, Chapter 7]) of classical descriptive complexity to the parameterized setting via the concept of *slicewise definability*. Many parameterized classes could be characterized this way [11, 20]. 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. [19]).

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

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

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

[1] This slightly deviates from [20], where the τ_k's are assumed to be pairwise equal and only *ordered* structures are considered.

hierarchy from Section 3. We start by carefully defining a very weak notion of reduction.

4.2 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. [16, Chapter 12.3]). We recall the definition (see e.g. [16, Chapter 11.2]).

DEFINITION 4.2. 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 STR$[\sigma]$ into 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 STR$[\sigma] \cup \{\emptyset\}$ to STR$[\tau] \cup \{\emptyset\}$ by adding to its domain $\emptyset$ as well as those $\mathbf{A} \in$ STR$[\sigma]$ with $\varphi_U(\mathbf{A}) = \emptyset$; 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 4.3. 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 [14] 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 4.4. Let Q, Q' be parameterized problems (Definition 4.1). 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

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

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

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

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 4.5. Note 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 4.6. We allow a reduction r between parameterized problems to output $(\emptyset, k)$ for $k \in \mathbb{N}$. 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 4.7. *Let Q, Q', Q'' be parameterized problems. If it holds that $Q \leqslant_{qfap} Q'$ and $Q' \leqslant_{qfap} Q''$, then $Q \leqslant_{qfap} Q''$.*

LEMMA 4.8. *Let Q, Q' be parameterized problems and assume Q' has bounded arity. If $Q \leqslant_{qfap} Q'$, then $Q \leqslant_{pl} Q'$.*

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-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:

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

4.3 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 4.9. *p-HOM$(\mathcal{G}^*) \leqslant_{qfap} p$-HOM$(\mathcal{H}^*)$, if $\mathcal{G} \preceq \mathcal{H}$.*

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:

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

We claim that

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

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 $\{(g_1, b_1), \dots, (g_w, b_w)\}$. 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

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

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

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

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 $\mathbf{H}$ 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 $b'' \notin B'$:

$$\begin{aligned} B'' \quad &:= \quad B' \cup \{b''\}, \\ E^{\mathbf{B}''} \quad &:= \quad E^{\mathbf{B}'} \cup \{(b'', b'')\}, \\ C_h^{\mathbf{B}''} \quad &:= \quad \begin{cases} C_h^{\mathbf{B}'} & , h \in H_0 \\ \{b''\} & , h \in H_1 \end{cases} ; \end{aligned}$$

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 as above: e.g. as formula $\varphi_U(\bar{x}\bar{y})$ one may now take

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

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

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

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. □

4.4 Homomorphism problems for arbitrary classes of structures

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

THEOREM 4.10. *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-HOM$(\mathcal{A}) \equiv_{qfap} p$-HOM$(\mathcal{G}^*)$.*

Thus, the complexity of p-HOM$(\mathcal{A})$ is determined in a strong sense by the level $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$ 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. [23]), 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. [10]).

We devide the proof into a sequence of lemmas.

LEMMA 4.11. *p-HOM$(\mathcal{A}^*) \leqslant_{qfap} p$-HOM$(\mathcal{A})$, if $\mathcal{A}$ is a class of cores.*

LEMMA 4.12. *p-HOM$(\mathsf{graph}(\mathcal{A})^*) \leqslant_{qfap} p$-HOM$(\mathcal{A}^*)$.*

These two lemmas are based on reductions from [10].

COROLLARY 4.13. *p-HOM$(\mathcal{A}) \equiv_{qfap} p$-HOM$(\mathsf{core}(\mathcal{A})^*)$.*

We show a partial converse to Lemma 4.12. This is the technically most involved step in the proof of Theorem 4.10.

LEMMA 4.14. *Suppose $\mathcal{A}$ has bounded arity. Then*

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

Proof. (Sketch) *Case 1:* Suppose $\mathsf{graph}(\mathcal{A})$ has unbounded treewidth. Then $\mathcal{L} \leq \mathsf{graph}(\mathcal{A})$ by the Hierarchy Theorem. By Theorem 4.9 we get p-HOM$(\mathcal{L}^*) \leqslant_{qfap} p$-HOM$(\mathsf{graph}(\mathcal{A})^*)$. Since trivially p-HOM$(\mathcal{H}^*) \leqslant_{qfap} p$-HOM$(\mathcal{L}^*)$ for every class of graphs $\mathcal{H}$ it suffices to show p-HOM$(\mathcal{A}^*) \leqslant_{qfap} p$-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-HOM$(\mathcal{A}) \leqslant_{qfap} p$-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

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

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 $\text{in}(\mathbf{A})^*$. Namely, for $(R, \bar{a}, i) \in L(\mathbf{A})$ we set

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

a subset of $L(\mathbf{B})$, and for $a \in R(\mathbf{A})$ we set $C^{\mathbf{B}'}_a := R(\mathbf{B}) = B$. It can be shown that $(\mathbf{A}, \mathbf{B}) \mapsto (\text{in}(\mathbf{A})^*, \mathbf{B}')$ is a reduction as desired.

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 $\mathsf{graph}(\mathcal{A}) \equiv \mathcal{H}$. Theorem 4.9 implies that $p\text{-}\mathrm{HOM}(\mathsf{graph}(\mathcal{A})^*) \equiv_{qfap} p\text{-}\mathrm{HOM}(\mathcal{H}^*)$, so it suffices to show $p\text{-}\mathrm{HOM}(\mathcal{A}^*) \leqslant_{qfap} p\text{-}\mathrm{HOM}(\mathcal{H}^*)$. The proof of this is similar to the proof of Theorem 4.9. □

Proof. (Theorem 4.10) Note that with $\mathcal{A}$ also $\mathsf{core}(\mathcal{A})$ is computably enumerable. We thus get

$$\begin{aligned} p\text{-}\mathrm{HOM}(\mathcal{A}) &\equiv_{qfap} p\text{-}\mathrm{HOM}(\mathsf{core}(\mathcal{A})^*) \\ &\equiv_{qfap} p\text{-}\mathrm{HOM}(\mathsf{graph}(\mathsf{core}(\mathcal{A}))^*) \end{aligned}$$

by Corollary 4.13 and Lemmas 4.14, 4.12. Now apply Theorem 4.9. □

5. Pebble games

Following the motivation given in the introduction we turn to pebble games which solve the homomorphism problems associated with the lower levels of the hierarchy (Theorem 3.14). We consider pebble games (cf. [16, Section 3.3]) played by two players, *Spoiler* and *Duplicator* on two similar structures $\mathbf{A}, \mathbf{B}$. Let us call $v = (p_1, \ldots, p_r) \in \mathbb{N}^r$ 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.

In this extended abstract we confine ourselves to stating the main results.

Our first main theorem is a decidable characterization of the homomorphism problems (arising from a single structure) solved by the v-game. We say that *the v-game solves p-*$\mathrm{HOM}(\mathcal{A})$ 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 5.1. *Let $\mathbf{A}$ be a structure, and v be a game vector. The following are equivalent.*

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

We next 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, as made precise in the following.

THEOREM 5.2. *(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-$\mathrm{HOM}(\mathcal{A})$.*
2. *The $\underbrace{(1, \ldots, 1)}_{n \text{ times}}$-game solves p-$\mathrm{HOM}(\mathcal{A})$.*
3. *The class $\mathcal{G}$ has tree depth $< n$.*

THEOREM 5.3. *(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-$\mathrm{HOM}(\mathcal{A})$.*
2. *$\mathcal{G} \preceq \mathcal{F}_r$.*
3. *The class $\mathcal{G}$ has bounded tree depth and stack depth $\leqslant r$.*

6. The homomorphism problems in L

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

$$\mathrm{HOM}(\mathcal{A}) := \{(\mathbf{B}, \mathbf{A}) \mid \mathbf{A} \in \mathcal{A} \;\&\; \mathbf{A} \xrightarrow{h} \mathbf{B}\}.$$

In this section we prove the following.

THEOREM 6.1. *Assume that* PATH $\neq$ para-L *and let $\mathcal{A}$ be a class of structures with bounded arity. The following are equivalent.*

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

This result characterizes the classical homomorphism problems in logarithmic space. The characterization is conditional on the hypothesis PATH $\neq$ para-L, an hypothesis from parameterized complexity theory. The class PATH has been discovered by Elberfeld et al. [17] and is defined as follows.

DEFINITION 6.2. 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 class, e.g. there are some fundamental problems which turn out to be complete for PATH under $\leqslant_{pl}$. The above characterization further underlines its importance. We refer to [10, 17] for more information. Here, we just mention the following result.

PROPOSITION 6.3 ([10]). *p-$\mathrm{HOM}(\mathcal{P}^*)$ is complete for* PATH *under $\leqslant_{pl}$.*

LEMMA 6.4. *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. (Theorem 6.1) ($1 \Rightarrow 2$) Let $d \geqslant 1$ bound the tree depth of $\mathsf{graph}(\mathsf{core}(\mathcal{A}))$. By Theorem 5.2, the v-game solves $\textsc{Hom}(\mathcal{A})$ where $v := (\underbrace{1,\ldots,1}_{d \text{ times}})$. It follows that $\textsc{Hom}(\mathcal{A}) = \{(\mathbf{B},\mathbf{A}) \mid \mathbf{A} \in \mathcal{A} \;\&\; \mathbf{A} \overset{v}{\to} \mathbf{B}\}$. This problem is in L by Lemma 6.4 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 3.16. By Theorems 4.9 and 4.10 (and Lemma 4.8) 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 6.3. □

7. 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

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

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 $\text{PP}^q[r]$ 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.

We establish the following theorem.

THEOREM 7.1. *Let $q, r \in \mathbb{N}, q \geqslant 1, r \geqslant 2$. Then*

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

Acknowledgments

The first author was supported by the Spanish Project FORMALISM (TIN2007-66523), by the Basque Government Project S-PE12UN050(SAI12/219), and by the University of the Basque Country under grant UFI11/45. The second author was supported by FWF (Austrian Science Fund) Project P 24654 N25.

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. *Proc. ICALP'07*, pages 279–290, 2007.

[3] A. Atserias and M. Weyer. Decidable relationships between consistency notions for constraint satisfaction problems. *Proc. CSL'09*, pages 102–116, 2009.

[4] L. Barto and M. Kozik. Constraint satisfaction problems of bounded width. *Proc. FOCS'09*, 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. *Complexity of Constraints*, pp. 93–124, 2008.

[8] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proc. STOC'77*, pages 77–90, 1977.

[9] H. Chen and V. Dalmau. Beyond hypertree width: Decomposition methods without decompositions. *Proc. Constraint Programming'05*, pages 167–181, 2005.

[10] H. Chen and M. Müller. The fine classification of conjunctive queries and parameterized logarithmic space complexity. *Proc. PODS'13*, pages 309-320, 2013, full version *arXiv:1306.5424*.

[11] Y. Chen, J. Flum, and M. Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. *Proc. CCC'03*, pages 13–29, 2003.

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

[13] V. Dalmau, P. G. Kolaitis, and M. Y. Vardi. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. *Proc. Constraint Programming '02*, LNCS, 2002.

[14] A. Dawar and Y. He. Parameterized complexity classes under logical reductions. *Proc. MFCS'09*, pages 258–269, 2009.

[15] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

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

[17] M. Elberfeld, C. Stockhusen, and T. Tantau. On the space complexity of parameterized problems. *Proc. IPEC'12*, LNCS 7535:206–217, Springer, 2012.

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

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

[20] J. Flum and M. Grohe. Describing parameterized complexity classes. *Inf. Comput.*, 187(2):291–319, 2003.

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

[22] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.* 64(3):579–627, 2002.

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

[24] Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? *Proc. STOC'01*, 2001.

[25] P. Kolaitis and M. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *J. of Comp. and Syst. Sciences*, 61:302–332, 2000.

[26] P. G. Kolaitis and M. Y. Vardi. A game-theoretic approach to constraint satisfaction. *Proc. AI'00*, pages 175–181, 2000.

[27] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *Proc. STOC'10*, pages 735–744, 2010.

[28] C. Papadimitriou and M. Yannakakis. On the Complexity of Database Queries. *J. of Comp. and Syst. Sciences* 58(3):407–427, 1999.

[29] 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, chpt 19. CRC Press, 2nd edition, 2009.