

The complexity of homomorphism and constraint satisfaction problems seen from the other side

Martin Grohe*

September 8, 2006

Abstract

We give a complexity theoretic classification of homomorphism problems for graphs and, more generally, relational structures obtained by restricting the left hand side structure in a homomorphism. For every class \mathbf{C} of structures, let $\text{HOM}(\mathbf{C}, -)$ be the problem of deciding whether a given structure $\mathcal{A} \in \mathbf{C}$ has a homomorphism to a given (arbitrary) structure \mathcal{B} . We prove that, under some complexity theoretic assumption from parameterized complexity theory, $\text{HOM}(\mathbf{C}, -)$ is in polynomial time if and only if \mathbf{C} has bounded treewidth modulo homomorphic equivalence.

Translated into the language of constraint satisfaction problems, our result yields a characterization of the tractable structural restrictions of constraint satisfaction problems. Translated into the language of database theory, it implies a characterization of the tractable instances of the evaluation problem for conjunctive queries over relational databases.

1. Introduction

The homomorphism problem for relational structures is a fundamental algorithmic problem playing an important role in different areas of computer science. The observation that the homomorphism problem is equivalent to both the evaluation problem and containment problem for conjunctive database queries goes back to Chandra and Merlin [7] (also see [29, 36]). Feder and Vardi [17] noted that constraint satisfaction problems in artificial intelligence can also be phrased as homomorphism problems. It is therefore no surprise that considerable efforts have been made to classify various restrictions of the homomorphism problem according to their computational complexity.

Let us look at graph homomorphisms first. Recall that a homomorphism from a graph \mathcal{G} to a graph \mathcal{H} is a mapping from the vertex set of \mathcal{G} to the vertex set of \mathcal{H} that preserves adjacency. The general homomorphism problem asks, given two graphs \mathcal{G} and \mathcal{H} , if there is a homomorphism from \mathcal{G} to \mathcal{H} . Several well-known NP-complete problems can be viewed as restrictions of the homomorphism problem. For example, the clique problem is equivalent to the homomorphism problem where the left hand side input graph \mathcal{G} is a complete graph, and the 3-colourability problem is equivalent to the homomorphism problem where the right hand side input graph \mathcal{H} is a triangle. For classes \mathbf{C} and \mathbf{D} of graphs, let $\text{HOM}(\mathbf{C}, \mathbf{D})$ denote the restriction of the homomorphism problem to input graphs $\mathcal{G} \in \mathbf{C}$ and $\mathcal{H} \in \mathbf{D}$. To simplify the notation, if either \mathbf{C} or \mathbf{D} is the class of all structures, we just use the placeholder ‘ $-$ ’.

Hell and Nešetřil [26] proved the following beautiful classification theorem: For every class \mathbf{D} of (undirected simple) graphs, $\text{HOM}(-, \mathbf{D})$ is in polynomial time if all graphs in \mathbf{D} are bipartite.¹ Otherwise, $\text{HOM}(-, \mathbf{D})$ is NP-complete.² This completely settles the classification problem for restrictions on the right hand side graphs.

We look at the corresponding question for restrictions on the left hand side, that is, at problems of the form $\text{HOM}(\mathbf{C}, -)$. Of course for every fixed graph \mathcal{G} the problem $\text{HOM}(\{\mathcal{G}\}, -)$ is in polynomial time (as

*Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany.
Email: grohe@informatik.hu-berlin.de

¹We consider the 1 vertex graph without any edges as bipartite.

²As a matter of fact, Hell and Nešetřil [26] stated their result in the following equivalent *nonuniform* version: For every graph \mathcal{H} , the problem $\text{HOM}(-, \{\mathcal{H}\})$ is in polynomial time if \mathcal{H} is bipartite. Otherwise, $\text{HOM}(-, \{\mathcal{H}\})$ is NP-complete. See Section 6.1 for more on the issue of uniformity.

opposed to $\text{HOM}(-, \{\mathcal{H}\})$). Furthermore, it is well-known that the problem $\text{HOM}(\mathbf{C}, -)$ is in polynomial time if \mathbf{C} is a class of graphs of *bounded treewidth*. This can be seen by a straightforward dynamic programming algorithm, and has been noted independently by several researchers in different contexts [8, 20]. Dalmau, Kolaitis, and Vardi [13] were able to push this a step further by showing that $\text{HOM}(\mathbf{C}, -)$ is in polynomial time if the class \mathbf{C} has *bounded treewidth modulo homomorphic equivalence*, a property that is defined as follows: Two graphs \mathcal{G}, \mathcal{H} are homomorphically equivalent if there is a homomorphism from \mathcal{G} to \mathcal{H} and a homomorphism from \mathcal{H} to \mathcal{G} . A class \mathbf{C} has bounded treewidth modulo homomorphic equivalence if there is a k such that every graph in \mathbf{C} is homomorphically equivalent to a graph of treewidth at most k . For example, every bipartite graph with at least one edge is homomorphically equivalent to the graph with two vertices and one edge between them. Thus the class of all bipartite graphs has bounded treewidth modulo homomorphic equivalence. We show that essentially the result of Dalmau et al. [13] is optimal:

Assume that $\text{FPT} \neq \text{W}[1]$. Then for every recursively enumerable class \mathbf{C} of graphs, the problem $\text{HOM}(\mathbf{C}, -)$ is in polynomial time if and only if \mathbf{C} has bounded treewidth modulo homomorphic equivalence.

$\text{FPT} \neq \text{W}[1]$ is a standard assumption from parameterized complexity theory that is widely believed to be true. We will give a brief introduction into parameterized complexity theory in Section 2.4. Let us just remark here that $\text{FPT} = \text{W}[1]$ would imply that 3-SAT is in deterministic time $2^{o(n)}$ [2], and that $\text{PTIME} = \text{NP}$ would imply $\text{FPT} = \text{W}[1]$. The latter immediately prompts the question of whether we can weaken the assumption of the theorem to $\text{PTIME} \neq \text{NP}$. It turns out that we cannot — we prove that the statement of the theorem is equivalent to the assumption $\text{FPT} \neq \text{W}[1]$, that is, if $\text{FPT} = \text{W}[1]$ then there is a recursively enumerable (actually, polynomial time decidable) class \mathbf{C} of graphs of unbounded treewidth modulo homomorphic equivalence such that the problem $\text{HOM}(\mathbf{C}, -)$ is in polynomial time. This is quite remarkable, because the statement of the theorem has nothing to do with parameterized complexity theory.

Under the assumption $\text{FPT} \neq \text{W}[1]$, we can only prove the theorem for recursively enumerable classes \mathbf{C} of graphs. However, if we slightly strengthen the assumption to *nonuniform-FPT* \neq *nonuniform-W[1]*, we can prove our theorem for arbitrary classes \mathbf{C} . Again we can show that this statement is equivalent to the assumption, that is, if *nonuniform-FPT* = *nonuniform-W[1]* then there is a class \mathbf{C} of graphs of unbounded treewidth modulo homomorphic equivalence such that the problem $\text{HOM}(\mathbf{C}, -)$ is in polynomial time.

Hell and Nešetřil’s classification of problems $\text{HOM}(-, \mathbf{D})$ is not only a classification of all polynomial time computable problems of this form under the assumption $\text{PTIME} \neq \text{NP}$, but it actually also is a *dichotomy theorem* in the sense that problems $\text{HOM}(-, \mathbf{D})$ are either in polynomial time or NP-complete. We give some evidence that no such dichotomy holds for problems $\text{HOM}(\mathbf{C}, -)$ by constructing a polynomial time computable class \mathbf{C} of graphs for which $\text{HOM}(\mathbf{C}, -)$ is equivalent to the LOG-CLIQUE problem. It seems unlikely that LOG-CLIQUE, the problem of deciding whether a given graph with n vertices has a clique of size at least $\log n$, is either in polynomial time or NP-complete. However, we prove a dichotomy for the parameterized complexity of the homomorphism problem. Let $p\text{-HOM}(\mathbf{C}, -)$ denote the parameterization of $\text{HOM}(\mathbf{C}, -)$ by the size of the left hand side input graph $\mathcal{G} \in \mathbf{C}$ (see Section 2.4 for details). Then $p\text{-HOM}(\mathbf{C}, -)$ is fixed-parameter tractable if, and only if, $\text{HOM}(\mathbf{C}, -)$ is in polynomial time. Moreover, if $p\text{-HOM}(\mathbf{C}, -)$ is not fixed-parameter tractable then it is $\text{W}[1]$ -complete.

So far, we have only looked at homomorphisms of undirected graphs. However, our classification theorem holds in the much wider setting of relational structures. This includes, for example, homomorphisms of directed graphs and coloured graphs and also the important class of *constraint satisfaction problems*. Homomorphisms of relational structures are naturally defined as relation preserving mappings, and we can extend the problem $\text{HOM}(\mathbf{C}, \mathbf{D})$ to arbitrary classes \mathbf{C} and \mathbf{D} of relational structures.

Our characterization of the tractable problems $\text{HOM}(\mathbf{C}, -)$ extends from classes of undirected graphs to classes of relational structures in a straightforward manner. This may be slightly surprising, because Hell and Nešetřil’s [26] classification of the tractable right hand side restrictions (that is, problems $\text{HOM}(-, \mathbf{D})$) so far resisted all attempts to be extended even to classes \mathbf{D} of oriented trees. The only restriction we have to pose on the classes \mathbf{C} is that the *arity* of the relations of structures in \mathbf{C} is bounded. In particular, the arity is always bounded if all structure in \mathbf{C} have the same finite vocabulary; this is the most common special case. Putting everything together, our main theorem reads as follows:

Theorem 1. *Assume that $\text{FPT} \neq \text{W}[1]$. Then for every recursively enumerable class \mathbf{C} structures of bounded arity the following three statements are equivalent:*

- (1) $\text{HOM}(\mathbf{C}, -)$ is in polynomial time.
- (2) $p\text{-HOM}(\mathbf{C}, -)$ is fixed-parameter tractable.
- (3) \mathbf{C} has bounded treewidth modulo homomorphic equivalence.

If either statement is false, then $p\text{-HOM}(\mathbf{C}, -)$ is $\text{W}[1]$ -hard.

As for the graph version of the theorem, we can drop the hypothesis that \mathbf{C} be recursively enumerable if we strengthen the complexity theoretic assumption to $\text{nonuniform-FPT} \neq \text{nonuniform-W}[1]$.

Our theorem has an important pre-cursor [25] stating the same equivalence for certain special classes of structures.³ For every class \mathbf{G} of undirected graphs, let $\mathbf{C}(\mathbf{G})$ be the class of all relational structures whose underlying graph is in \mathbf{G} . For example, $\mathbf{C}(\mathbf{G})$ contains all directed graphs obtained by orienting the edges of a graph in \mathbf{G} in an arbitrary way. Then essentially, [25] proves our main result for classes of the form $\mathbf{C}(\mathbf{G})$, except that there is no need for homomorphic equivalence in (3), because it can easily be seen that a class $\mathbf{C}(\mathbf{G})$ has bounded treewidth if and only if it has bounded treewidth modulo homomorphic equivalence. In Section 5 we shall prove that a slight improvement of the main result of [25] follows from our Theorem 1.

Theorem 1 will be proved in Sections 3 and 4. The proof uses similar ideas as the proof of [25]. At its core, it builds on the deep Excluded Grid Theorem due to Robertson and Seymour [33]. In Section 6, we discuss applications of Theorem 1 in the contexts of constraint satisfaction problems and database query evaluation. Finally, in Section 7 we make a few complexity theoretic observations and, in particular, show the equivalence of the complexity theoretic assumption $\text{FPT} \neq \text{W}[1]$ with the statement of our theorem.

2. Preliminaries

\mathbb{N} denotes the set of positive integers, and for every $n \in \mathbb{N}$ we let $[n] = \{1, \dots, n\}$.

2.1. Relational structures and homomorphisms. A *vocabulary* τ is a finite set of relation symbols of specified *arities*. The *arity* of τ is the maximum of the arities of all relations symbols it contains. A τ -*structure* \mathcal{A} consists of a finite set A called the *universe* of \mathcal{A} and for each relation symbol $R \in \tau$, say, of arity r , an r -ary relation $R^{\mathcal{A}} \subseteq A^r$. Note that we require vocabularies and structures to be finite. If \mathbf{C} is a class of structures and τ a vocabulary, then we write $\mathbf{C}[\tau]$ for the class of all τ -structures in \mathbf{C} . We say that a class \mathbf{C} of structures is of *bounded arity* if there is an r such that arity of the vocabulary of every structure in \mathbf{C} is at most r . Note that every class $\mathbf{C}[\tau]$ for a (finite) vocabulary τ is of bounded arity.

For vocabularies $\sigma \subseteq \tau$, the σ -*reduct* of a τ -structure \mathcal{B} is the σ -structure \mathcal{A} with universe $A = B$ and $R^{\mathcal{A}} = R^{\mathcal{B}}$ for all $R \in \sigma$. We denote the σ -reduct of \mathcal{B} by $\mathcal{B}|_{\sigma}$. A τ -structure \mathcal{B} is an *expansion* of a σ -structure \mathcal{A} if $\mathcal{B}|_{\sigma} = \mathcal{A}$. A τ -structure \mathcal{A} is a *substructure* of a τ -structure \mathcal{B} if $A \subseteq B$ and $R^{\mathcal{A}} \subseteq R^{\mathcal{B}}$ for all $R \in \tau$. We write $\mathcal{A} \subseteq \mathcal{B}$ to denote that \mathcal{A} is a substructure of \mathcal{B} and $\mathcal{A} \subset \mathcal{B}$ to denote that \mathcal{A} is a *proper* substructure of \mathcal{B} , that is, $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{A} \neq \mathcal{B}$. A substructure $\mathcal{A} \subseteq \mathcal{B}$ is *induced* if for all $R \in \tau$, say, of arity r , we have $R^{\mathcal{A}} = R^{\mathcal{B}} \cap A^r$. For a subset $A \subseteq B$, we write $\mathcal{B}[A]$ to denote the induced substructure of \mathcal{B} with universe A .

We distinguish between the cardinality $|A|$ of the universe A of a τ -structure \mathcal{A} and the size $\|\mathcal{A}\|$ of \mathcal{A} , which we define as

$$\|\mathcal{A}\| = |\tau| + |A| + \sum_{R \in \tau} |R^{\mathcal{A}}| \cdot \text{arity}(R).$$

$\|\mathcal{A}\|$ is roughly the size of a reasonable encoding of \mathcal{A} as an input for a RAM (see [18] for details). When taking structures \mathcal{A} as inputs for algorithms, we measure the running time in terms of $\|\mathcal{A}\|$.

2.2. Homomorphisms. We often abbreviate tuples (a_1, \dots, a_k) by \bar{a} . If f is a mapping whose domain contains a_1, \dots, a_k we write $f(\bar{a})$ to abbreviate $(f(a_1), \dots, f(a_k))$.

A *homomorphism* from a τ -structure \mathcal{A} to a τ -structure \mathcal{B} is a mapping $h : A \rightarrow B$ such that for all $R \in \tau$ and all tuples $\bar{a} \in R^{\mathcal{A}}$ we have $h(\bar{a}) \in R^{\mathcal{B}}$. For two classes \mathbf{C} and \mathbf{D} of structures, $\text{HOM}(\mathbf{C}, \mathbf{D})$ is the following problem:

³The results in [25] are formulated in terms of the evaluation problem for conjunctive queries, but this is equivalent.

$\text{HOM}(\mathbf{C}, \mathbf{D})$

Instance: $\mathcal{A} \in \mathbf{C}, \mathcal{B} \in \mathbf{D}$.

Problem: Decide if there is a homomorphism from \mathcal{A} to \mathcal{B} .

Let us point out that the classes \mathbf{C} and \mathbf{D} may contain structures of different vocabularies. By definition there can never be a homomorphism between structures of different vocabularies, so instances $(\mathcal{A}, \mathcal{B})$ of $\text{HOM}(\mathbf{C}, \mathbf{D})$ where \mathcal{A} and \mathcal{B} have different vocabularies are always ‘no’-instances. If \mathbf{D} is the class of all finite structures, we usually write $\text{HOM}(\mathbf{C}, -)$ instead of $\text{HOM}(\mathbf{C}, \mathbf{D})$. (Similarly, we write $\text{HOM}(-, \mathbf{D})$ if \mathbf{C} is the class of all structures, but we are mainly interested in problems $\text{HOM}(\mathbf{C}, -)$ in this paper.)

Remark 2. A remark is in order as to what we actually mean when we say that for some class \mathbf{C} the problem $\text{HOM}(\mathbf{C}, -)$ is in polynomial time. When considered as a decision problem in the usual sense, $\text{HOM}(\mathbf{C}, -)$ is the language (over some finite alphabet) consisting of the encodings of all pairs $(\mathcal{A}, \mathcal{B})$ of structures such that $\mathcal{A} \in \mathbf{C}$ and there is a homomorphism from \mathcal{A} to \mathcal{B} . Then clearly, the membership problem for \mathbf{C} is reducible to $\text{HOM}(\mathbf{C}, -)$, and hence $\text{HOM}(\mathbf{C}, -)$ can only be in PTIME if the class \mathbf{C} itself is decidable in PTIME. However, we prefer to view $\text{HOM}(\mathbf{C}, -)$ as a *promise problem*; when we say that $\text{HOM}(\mathbf{C}, -)$ is in polynomial time for an arbitrary class \mathbf{C} of structures we mean the following: *There is a polynomial time algorithm that, if its input consists of (the encoding of) two structures $\mathcal{A} \in \mathbf{C}$ and \mathcal{B} , correctly decides whether there is a homomorphism from \mathcal{A} to \mathcal{B} . If the input is not of this form, then the answer of the algorithm may be arbitrary.* This liberal point of view only makes our results stronger.

The reader feeling uneasy about this should restrict our main theorem to polynomial time decidable classes \mathbf{C} of structures. If the polynomial time decidability of the problem $\text{HOM}(\mathbf{C}, -)$ is viewed in the strict sense, this is not a severe restriction because polynomial time decidable classes \mathbf{C} are the only ones for which $\text{HOM}(\mathbf{C}, -)$ may be in polynomial time anyway.

Two structures \mathcal{A} and \mathcal{B} are *homomorphically equivalent* if there is a homomorphism from \mathcal{A} to \mathcal{B} and also a homomorphism from \mathcal{B} to \mathcal{A} . Note that if structures \mathcal{A} and \mathcal{A}' are homomorphically equivalent, then for every structure \mathcal{B} there is a homomorphism from \mathcal{A} to \mathcal{B} if and only if there is a homomorphism from \mathcal{A}' to \mathcal{B} ; in other words: the instances $(\mathcal{A}, \mathcal{B})$ and $(\mathcal{A}', \mathcal{B})$ of the homomorphism problem are equivalent. However, the two instances may have vastly different sizes.

Homomorphic equivalence is closely related to the concept of the core of a structure: A structure \mathcal{A} is a *core* if there is no homomorphism from \mathcal{A} to a proper substructure of \mathcal{A} . A *core of* a structure \mathcal{A} is a substructure $\mathcal{A}' \subseteq \mathcal{A}$ such that there is a homomorphism from \mathcal{A} to \mathcal{A}' and \mathcal{A}' is a core. Obviously, every core of a structure is homomorphically equivalent to the structure. The following lemma states another basic fact about cores:

Lemma 3. *Let \mathcal{A} and \mathcal{B} be homomorphically equivalent structures, and let \mathcal{A}' and \mathcal{B}' be cores of \mathcal{A} and \mathcal{B} , respectively. Then \mathcal{A}' and \mathcal{B}' are isomorphic.*

In particular, all cores of a structure \mathcal{A} are isomorphic. Therefore, we often speak of *the* core of \mathcal{A} .

Remark 4. It is easy to see that it is NP-hard to decide, given structures $\mathcal{A} \subseteq \mathcal{B}$, whether \mathcal{A} is isomorphic to the core of \mathcal{B} . (For an arbitrary graph \mathcal{G} , let \mathcal{A} be a triangle and \mathcal{B} the disjoint union of \mathcal{G} with \mathcal{A} . Then \mathcal{A} is a core of \mathcal{B} if, and only if, \mathcal{G} is 3-colourable.) Hell and Nešetřil [27] proved that it is co-NP-complete to decide whether a graph is a core.

2.3. Graph minors and tree width. Let E be a binary relation symbol. We view graphs as $\{E\}$ -structures \mathcal{G} . We usually denote the vertex set of a graph \mathcal{G} (that is, the universe of the $\{E\}$ -structure \mathcal{G}) by $V^{\mathcal{G}}$ instead of G . Unless explicitly stated otherwise, we assume graphs to be undirected and loop free. In this case, we usually view edges $e = (v, w)$ as unordered pairs $\{v, w\}$ and use notations like $v \in e$ or $\{v, w\} \in E^{\mathcal{G}}$.

A graph \mathcal{H} is a *minor* of a graph \mathcal{G} if \mathcal{H} is isomorphic to a graph that can be obtained from a subgraph of \mathcal{G} by contracting edges. A *minor map* from \mathcal{H} to \mathcal{G} is a mapping $\mu : V^{\mathcal{H}} \rightarrow 2^{V^{\mathcal{G}}}$ with the following properties:

- For all $v \in V^{\mathcal{H}}$ the set $\mu(v)$ is non-empty and connected in \mathcal{G} .

- For all $v, w \in V^{\mathcal{H}}$ with $v \neq w$ the sets $\mu(v)$ and $\mu(w)$ are disjoint.
- For all edges $\{v, w\} \in E^{\mathcal{H}}$ there are $v' \in \mu(v), w' \in \mu(w)$ such that $\{v', w'\} \in E^{\mathcal{G}}$.

Slightly abusing terminology, we call a minor map μ from \mathcal{H} to \mathcal{G} *onto* if $\bigcup_{v \in V^{\mathcal{H}}} \mu(v) = V^{\mathcal{G}}$. It is easy to see that there is a minor map from \mathcal{H} to \mathcal{G} if, and only if, \mathcal{H} is a minor of \mathcal{G} . Moreover, if \mathcal{H} is a minor of a connected graph \mathcal{G} then we can always find a minor map from \mathcal{H} onto \mathcal{G} .

Trees are connected acyclic graphs. A *tree-decomposition* of a graph \mathcal{G} is a pair (\mathcal{T}, β) , where \mathcal{T} is a tree and $\beta : V^{\mathcal{T}} \rightarrow 2^{V^{\mathcal{G}}}$ such that the following conditions are satisfied:

- For every $v \in V^{\mathcal{G}}$ the set $\{t \in V^{\mathcal{T}} \mid v \in \beta(t)\}$ is non-empty and connected in \mathcal{T} .
- For every $e \in E^{\mathcal{G}}$ there is a $t \in V^{\mathcal{T}}$ such that $e \subseteq \beta(t)$.

The *width* of a tree-decomposition (\mathcal{T}, β) is $\max\{|\beta(t)| \mid t \in V^{\mathcal{T}}\} - 1$, and the *treewidth* of a graph \mathcal{G} , denoted by $\text{tw}(\mathcal{G})$, is the minimum w such that \mathcal{G} has a tree-decomposition of width w .

For $k, \ell \geq 1$, the $(k \times \ell)$ -grid is the graph with vertex set $[k] \times [\ell]$ and an edge between (i, j) and (i', j') if $|i - i'| + |j - j'| = 1$. It is not hard to see that the $(k \times k)$ -grid has treewidth k . Robertson and Seymour proved the following “converse”, which is known as the Excluded Grid Theorem.

Theorem 5 (Robertson and Seymour [33]). *For every k there exists a $w(k)$ such that the $(k \times k)$ -grid is a minor of every graph of treewidth at least $w(k)$.*

The best currently known upper bound for $w(k)$ is 20^{2k^5} [35].

We need to transfer some of the notions of graph minor theory to arbitrary relational structures. The *Gaifman graph* (also known as *primal graph*) of a τ -structure \mathcal{A} is the graph $\mathcal{G}(\mathcal{A})$ with vertex set A and an edge between a and b if $a \neq b$ and there is a relation symbol $R \in \tau$, say, of arity r , and a tuple $(a_1, \dots, a_r) \in R^{\mathcal{A}}$ such that $a, b \in \{a_1, \dots, a_r\}$. We can now transfer graph theoretic notions to relational structures. In particular, a subset $B \subseteq A$ is *connected* in a structure \mathcal{A} if it is connected in $\mathcal{G}(\mathcal{A})$. A *minor map* from a structure \mathcal{A} to a structure \mathcal{B} is a mapping $\mu : A \rightarrow 2^B$ that is a minor map from $\mathcal{G}(\mathcal{A})$ to $\mathcal{G}(\mathcal{B})$. A *tree decomposition* of a τ -structure \mathcal{A} can simply be defined to be a tree-decomposition of $\mathcal{G}(\mathcal{A})$. Equivalently, a tree decomposition of \mathcal{A} can be defined directly by replacing the second condition in the definition of tree decompositions of graphs by

- For every $R \in \tau$ and $(a_1, \dots, a_r) \in R^{\mathcal{A}}$ there is a $t \in V^{\mathcal{T}}$ such that $\{a_1, \dots, a_r\} \subseteq \beta(t)$.

A class \mathbf{C} of structures has *bounded treewidth* if there is a $w \in \mathbb{N}$ such that $\text{tw}(\mathcal{A}) \leq w$ for all $\mathcal{A} \in \mathbf{C}$. A class \mathbf{C} of structures has *bounded treewidth modulo homomorphic equivalence* if there is a $w \in \mathbb{N}$ such that every $\mathcal{A} \in \mathbf{C}$ is homomorphically equivalent to a structure of treewidth at most w . Slightly ambiguously, we say that a class \mathbf{C} of structures has *unbounded treewidth modulo homomorphic equivalence* if it does not have bounded treewidth modulo homomorphic equivalence. The following straightforward lemma provides a useful characterization of bounded treewidth modulo homomorphic equivalence in terms of cores.

Lemma 6. *A structure \mathcal{A} is homomorphically equivalent to a structure of treewidth at most w if and only if the core of \mathcal{A} has treewidth at most w .*

Proof: Use the fact that treewidth is monotone with respect to substructures, that is, if $\mathcal{A} \subseteq \mathcal{B}$ then $\text{tw}(\mathcal{A}) \leq \text{tw}(\mathcal{B})$, and Lemma 3, by which two homomorphically equivalent structures have isomorphic cores. \square

2.4. Parameterized complexity theory. Parameterized complexity theory provides a framework for a fine-grain complexity analysis of algorithmic problems that are intractable in general. It builds on a notion of tractability called *fixed-parameter tractability*, which relaxes the classical notion of tractability, polynomial time computability, by admitting algorithms whose running time is exponential, but only in terms of some *parameter* of the problem instance that can be expected to be small in the typical applications.

A *parameterization* of a decision problem $P \subseteq \Sigma^*$ is a polynomial time computable mapping $\kappa : \Sigma^* \rightarrow \mathbb{N}$, and a *parameterized problem* over some alphabet Σ is a pair (P, κ) consisting of a problem $P \subseteq \Sigma^*$

and a parameterization κ of P . For example, the *parameterized clique problem* p -CLIQUE is the problem (P, κ) , where P is the set of all pairs (\mathcal{G}, k) (suitably encoded over some finite alphabet) and κ is defined by $\kappa(\mathcal{G}, k) := k$. We usually present parameterized problems in the following form:

<p>p-CLIQUE <i>Instance:</i> Graph \mathcal{G}, $k \in \mathbb{N}$. <i>Parameter:</i> k. <i>Problem:</i> Decide if \mathcal{G} has a clique of size k.</p>

The problem we are mainly interested in here is the homomorphism problem parameterized by the size of the left hand side input structure. For two classes \mathbf{C} and \mathbf{D} of structures, we let:

<p>p-HOM(\mathbf{C}, \mathbf{D}) <i>Instance:</i> Structures $\mathcal{A} \in \mathbf{C}$, $\mathcal{B} \in \mathbf{D}$. <i>Parameter:</i> \mathcal{A}. <i>Problem:</i> Decide if there is a homomorphism from \mathcal{A} to \mathcal{B}.</p>
--

Again, we write p -HOM($\mathbf{C}, -$) to denote p -HOM(\mathbf{C}, \mathbf{D}) if \mathbf{D} is the class of all finite structures.

A parameterized problem (P, κ) over Σ is *fixed-parameter tractable* if there is a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that decides if a given instance $x \in \Sigma^*$ belongs to P in time

$$f(\kappa(x)) \cdot |x|^{O(1)}.$$

FPT denotes the class of all fixed-parameter tractable parameterized problems.

An *fpt-reduction* from a parameterized problem (P, κ) over Σ to a parameterized problem (P', κ') over Σ' is a mapping $R : \Sigma^* \rightarrow (\Sigma')^*$ such that:

- For all $x \in \Sigma^*$ we have $x \in P \iff R(x) \in P'$.
- There is a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that, given $x \in \Sigma^*$, computes $R(x)$ in time $f(\kappa(x)) \cdot |x|^{O(1)}$.
- There is a computable function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that for all instances $x \in \Sigma^*$ we have $\kappa'(R(x)) \leq g(\kappa(x))$.

Hardness and *completeness* of parameterized problems for a parameterized complexity class are defined in the usual way.

For example, for every class \mathbf{C} of structures that contains all complete graphs, the mapping that associates with every instance (\mathcal{G}, k) of p -CLIQUE the instance $(\mathcal{K}_k, \mathcal{G})$ of p -HOM($\mathbf{C}, -$), where \mathcal{K}_k denotes the complete graph with k vertices, is an fpt-reduction from p -CLIQUE to p -HOM($\mathbf{C}, -$).

Downey and Fellows [14] defined a hierarchy $W[1] \subseteq W[2] \subseteq \dots$ of parameterized complexity classes, and they conjecture that this hierarchy is strict and that FPT is strictly contained in $W[1]$. The classes of the W -hierarchy are defined in terms of a parameterized version of the satisfiability problem for bounded-depth Boolean circuits. We refer the reader to [16, 19] for the technical definitions. In this paper, we are only interested in the class $W[1]$, which can be seen as an analogue of NP in parameterized complexity theory. Our hardness proof uses the following theorem:

Theorem 7 (Downey and Fellows [15]). *p -CLIQUE is $W[1]$ -complete under fpt-reductions.*

By the reduction we have given above, this implies that p -HOM($\mathbf{C}, -$) is $W[1]$ -hard under fpt-reductions for every class \mathbf{C} of structures that contains all complete graphs.

The notions as we described them are based on what Downey and Fellows [16] call *strongly uniform* fixed parameter tractability. There also is a nonuniform version of the theory. A parameterized problem (P, κ) is *nonuniformly fixed-parameter tractable* if there is a constant $c \in \mathbb{N}$ such that for every $k \in \mathbb{N}$ there is an algorithm that, given an $x \in \Sigma^*$ with $\kappa(x) = k$, decides if $x \in P$ in time $O(|x|^c)$. The class of all nonuniformly fixed-parameter tractable problems is denoted by *nonuniform-FPT*. Similarly, one defines

a notion of *nonuniform fpt-reduction*. Then *nonuniform-W[1]* is the class of all parameterized problems that can be reduced to p -CLIQUE by a nonuniform fpt-reduction. We can slightly strengthen our results if we replace the assumption $\text{FPT} \neq \text{W}[1]$ by its nonuniform version. Note that $\text{FPT} = \text{W}[1]$ implies $\text{nonuniform-FPT} = \text{nonuniform-W}[1]$, because if p -CLIQUE is in FPT , then it is in nonuniform-FPT . Thus the assumption $\text{nonuniform-FPT} \neq \text{nonuniform-W}[1]$ is at least as strong as $\text{FPT} \neq \text{W}[1]$. The converse is not known.

3. The tractability result

Theorem 8 (Dalmau, Kolaitis, and Vardi [13]). *Let \mathbf{C} be a class of structures of bounded treewidth modulo homomorphic equivalence. Then $\text{HOM}(\mathbf{C}, -)$ is in polynomial time.*

Dalmau et al. state a slightly weaker theorem; they only admit classes \mathbf{C} of structures of a fixed vocabulary τ . However, their proof can easily be extended to arbitrary classes \mathbf{C} . For the reader's convenience, we sketch a proof of the theorem:

Proof: Let $w \in \mathbb{N}$ such that all structures in \mathbf{C} are homomorphically equivalent to a structure of treewidth at most w . Let $\mathcal{A} \in \mathbf{C}$, and let \mathcal{B} be an arbitrary structure. By Lemma 6, the core \mathcal{A}' of \mathcal{A} has treewidth at most w .

We consider the following 2 player game, which is known as the *existential $(w+1)$ -pebble game* [28]: The positions of the game are pairs (S, h) consisting of a subset $S \subseteq A$ of size at most $w+1$ and a mapping $h : S \rightarrow B$. The initial position is (\emptyset, \emptyset) . In each round of the game, player I chooses a new subset S of A of size at most $w+1$. Then player II chooses a new mapping $h : S \rightarrow B$ subject to the following *compatibility condition*: If (S', h') was the previous position, then $h(a) = h'(a)$ for all $a \in S \cap S'$. Player II wins the game if in each position (S, h) that occurs, h is a homomorphism from the induced substructure $\mathcal{A}[S]$ of \mathcal{A} to \mathcal{B} .

Claim: There is a homomorphism from \mathcal{A} to \mathcal{B} if and only if player II has a winning strategy for the game.

Proof: The forward direction is straightforward: Suppose that h is a homomorphism from \mathcal{A} to \mathcal{B} . The strategy of player II is to always choose the restriction of h to the current set S . Obviously, this is a winning strategy.

The backward direction is based on the fact that the core \mathcal{A}' of \mathcal{A} has treewidth at most w . Let (\mathcal{T}, β) be a tree decomposition of \mathcal{A}' of width at most w . We fix a root r for \mathcal{T} . For each node t of \mathcal{T} , let T_t be the vertex set of the subtree of \mathcal{T} rooted in t . Let \mathcal{A}'_t denote the induced substructure

$$\mathcal{A}' \left[\bigcup_{u \in T_t} \beta(u) \right]$$

of \mathcal{A}' , and let $H(t)$ be the set of all mappings $h : \beta(t) \rightarrow B$ that can be extended to a homomorphism from \mathcal{A}'_t to \mathcal{B} . Then there is a homomorphism from \mathcal{A}' to \mathcal{B} (and hence from \mathcal{A} to \mathcal{B}) if, and only if, $H(r) \neq \emptyset$ for the root r .

Now suppose that there exists no homomorphism from \mathcal{A} to \mathcal{B} . Then $H(r) = \emptyset$. We shall describe a winning strategy for player I in the game. He maintains the following invariant for each position (S, h) that occurs:

- (\star) There is some node t of \mathcal{T} such that $S = \beta(t)$, and $h \notin H(t)$.

Furthermore, in each move the node t in (\star) will be closer to a leaf, and if t is a leaf then player I will win the game. Let us emphasise that we play the game on the structures \mathcal{A} and \mathcal{B} and not on \mathcal{A}' and \mathcal{B} . But since \mathcal{A}' is a substructure of \mathcal{A} , player I can always pick subsets $S \subseteq A'$. In his first move, player I picks $\beta(r)$. Since $H(r) = \emptyset$, (\star) will be satisfied after the first round of the game no matter how player II answers. Now suppose we are in a position $(\beta(t), h)$, where $h \notin H(t)$. If t is a leaf, h cannot be a homomorphism from $\mathcal{A}[\beta(t)]$ to \mathcal{B} , and player I wins the game. If t is not a leaf and h is a homomorphism, then there must be some child t' of t such that for all $h' \in H(t')$, h is not compatible with h' , because otherwise h would be in $H(t)$. (Note that here we use the first property of tree decompositions in an essential way.) Player I chooses $\beta(t')$ for such a child and maintains the invariant (\star).

This completes the proof of the claim.

There is a straightforward dynamic programming algorithm deciding whether player I has a winning strategy for the game in time $O(|A|^{2(w+1)})$. This proves the theorem. \square

4. The main hardness result

In this section, we shall prove the following hardness theorem. Combined with the tractability theorem of the previous section, it yields Theorem 1 (details will be given after the proof of Theorem 9).

Theorem 9. *Let \mathbf{C} be a recursively enumerable class structures of bounded arity that does not have bounded treewidth modulo homomorphic equivalence. Then $p\text{-HOM}(\mathbf{C}, -)$ is $\text{W}[1]$ -hard under fpt-reductions.*

The proof requires some preparation. Let $k \geq 2$ and $K = \binom{k}{2}$, and let \mathcal{A} be a connected τ -structure whose Gaifman graph contains a $(k \times K)$ -grid as a minor. Let $\mu : [k] \times [K] \rightarrow 2^A$ a minor map from the $(k \times K)$ -grid onto \mathcal{A} . Recall that μ being *onto* means that the sets $\mu(h)$, for $h \in [k] \times [K]$, form a partition of the universe A of \mathcal{A} and that we can choose the minor map to be onto because \mathcal{A} is connected. We fix some bijection ρ between $[K]$ and the set of all unordered pairs of elements of $[k]$. For $p \in [K]$, we sloppily write $i \in p$ instead of $i \in \rho(p)$. It will be convenient to jump back and forth between viewing the columns of the $(k \times K)$ -grid as being indexed by elements of $[K]$ and unordered pairs of elements of $[k]$.

Let $\mathcal{G} = (V^{\mathcal{G}}, E^{\mathcal{G}})$ be a graph. We shall define a τ -structure $\mathcal{B} = \mathcal{B}(\mathcal{A}, \mu, \mathcal{G})$ such that there is a homomorphism from \mathcal{A} to \mathcal{B} if, and only if, \mathcal{G} contains a k -clique. The universe of \mathcal{B} is

$$B = \left\{ (v, e, i, p, a) \mid v \in V^{\mathcal{G}}, e \in E^{\mathcal{G}}, i \in [k], p \in [K] \text{ such that } (v \in e \iff i \in p), a \in \mu(i, p) \right\}.$$

We define the *projection* $\Pi : B \rightarrow A$ by letting

$$\Pi(v, e, i, p, a) = a$$

for all $(v, e, i, p, a) \in B$. Recall that the minor map μ is onto. Thus every $a \in A$ is contained in $\mu(i, p)$ for some $i \in [k]$ and $p \in [K]$. Note that, for an element $a \in \mu(i, p)$, $\Pi^{-1}(a)$ contains all tuples (v, e, i, p, a) with $v \in V^{\mathcal{G}}, e \in E^{\mathcal{G}}$ such that $(v \in e \iff i \in p)$. As usually, we extend Π and Π^{-1} to tuples of elements by defining it componentwise. We shall define the relations of \mathcal{B} in such a way that Π is a homomorphism from \mathcal{B} to \mathcal{A} . For every $R \in \tau$, say, of arity r , and for all tuples $\bar{a} = (a_1, \dots, a_r) \in R^{\mathcal{A}}$ we add to $R^{\mathcal{B}}$ all tuples $\bar{b} = (b_1, \dots, b_r) \in \Pi^{-1}(\bar{a})$ satisfying the following two constraints for all $b, b' \in \{b_1, \dots, b_r\}$:

(C1) If $b = (v, e, i, p, a)$ and $b' = (v', e', i', p', a')$
then $v = v'$.

(C2) If $b = (v, e, i, p, a)$ and $b' = (v', e', i', p, a')$
then $e = e'$.

Lemma 10. *The projection Π is a homomorphism from \mathcal{B} to \mathcal{A} .*

Proof: Follows immediately from the definition of \mathcal{B} . \square

Lemma 11. *If \mathcal{G} contains a k -clique, then there is a homomorphism from \mathcal{A} to \mathcal{B} .*

Proof: Let v_1, \dots, v_k be the vertex set of a k -clique in \mathcal{G} . Recall that ρ is a bijection between $[K]$ and the set of unordered pairs of elements of $[k]$. For $p \in [K]$ with $\rho(p) = \{i, j\}$, let $e_p \in E^{\mathcal{G}}$ be the edge between v_i and v_j .

We define $h : A \rightarrow B$ by letting

$$h(a) = (v_i, e_p, i, p, a)$$

for $i \in [k]$, $p \in [K]$, and $a \in \mu(i, p)$. We have to make sure that indeed $h(a) \in B$ for all $a \in A$, that is, that $(v_i \in e_p \iff i \in p)$, but this is immediate from the definition of e_p .

To prove that h is a homomorphism, let $R \in \tau$ be r -ary and $\bar{a} = (a_1, \dots, a_r) \in R^A$. Let i_1, \dots, i_r and p_1, \dots, p_r be such that $a_j \in \mu(i_j, p_j)$ for $j \in [r]$. Then

$$h(\bar{a}) = \left((v_{i_1}, e_{p_1}, i_1, p_1, a_1), \dots, (v_{i_r}, e_{p_r}, i_r, p_r, a_r) \right).$$

Conditions (C1) and (C2) are trivially satisfied, thus $h(\bar{a}) \in R^B$. \square

Lemma 12. *Suppose that \mathcal{A} is a core. If there is a homomorphism from \mathcal{A} to \mathcal{B} , then \mathcal{G} contains a k -clique.*

Proof: Let $h : A \rightarrow B$ be a homomorphism from \mathcal{A} to \mathcal{B} . Then $f = \Pi \circ h$ is a homomorphism from \mathcal{A} to \mathcal{A} . Thus by Lemma 3, f is an isomorphism, because \mathcal{A} is a core. Without loss of generality we assume that f is the identity. If this is not the case, we consider the homomorphism $h \circ f^{-1}$ instead of h .

By the definition of Π , this means that for $i \in [k]$, $p \in [K]$, and $a \in \mu(i, p)$ we have

$$h(a) = (v_a, e_a, i, p, a).$$

for a vertex $v_a \in V^{\mathcal{G}}$ and an edge $e_a \in E^{\mathcal{G}}$ such that $(v_a \in e_a \iff i \in p)$.

Claim 1: For $i \in [k]$, $p \in [K]$ and $a, a' \in \mu(i, p)$ we have $v_a = v_{a'}$ and $e_a = e_{a'}$.

Proof: Since $\mu(i, p)$ is connected in \mathcal{A} , it suffices to prove the claim for a, a' such that there is an edge between a and a' in the Gaifman graph of \mathcal{A} .

So let $R \in \tau$, say, of arity r , and $\bar{a} = (a_1, \dots, a_r) \in R^A$ such that $a, a' \in \{a_1, \dots, a_r\}$. Since h is a homomorphism we have $h(\bar{a}) \in R^B$. Thus by conditions (C1) and (C2), we must have $v_a = v_{a'}$ and $e_a = e_{a'}$. This proves Claim 1.

Claim 2: For $i, i' \in [k]$, $p \in [K]$ and $a \in \mu(i, p)$, $a' \in \mu(i', p)$ we have $e_a = e_{a'}$.

Proof: By a simple inductive argument in which Claim 1 is the base case, it suffices to prove Claim 2 for $i' = i + 1$.

Since μ is a minor map from the $(k \times K)$ -grid to \mathcal{A} and there is an edge between (i, p) and (i', p) in the grid, there must be some edge between $\mu(i, p)$ and $\mu(i', p)$ in the Gaifman graph $\mathcal{G}(\mathcal{A})$. Thus there must be some relation $R \in \tau$ and tuple $\bar{a} \in R^A$ such that both $\mu(i, p)$ and $\mu(i', p)$ contain an element of \bar{a} .

Let $R \in \tau$ be r -ary and $\bar{a} = (a_1, \dots, a_r) \in R^A$. Without loss of generality, suppose that $a_1 \in \mu(i, p)$ and $a_2 \in \mu(i', p)$. Since h is a homomorphism we have $h(\bar{a}) \in R^B$. Thus by condition (C2) we have $e_{a_1} = e_{a_2}$. By Claim 1, we have $e_a = e_{a_1}$ and $e_{a'} = e_{a_2}$. This completes the proof of Claim 2.

Claim 3: For $i \in [k]$, $p, p' \in [K]$ and $a \in \mu(i, p)$, $a' \in \mu(i, p')$ we have $v_a = v_{a'}$.

Proof: Analogously to the proof of Claim 2 using condition (C1) instead of (C2).

Together, the three claims imply that there are vertices $v_1, \dots, v_k \in V^{\mathcal{G}}$ and edges $e_1, \dots, e_K \in E^{\mathcal{G}}$ such that for all $i \in [k]$, $p \in [K]$, and $a \in \mu(i, p)$ we have $h(a) = (v_i, e_p, i, p, a)$. Since $h(a) \in B$ for all $a \in A$, this implies that

$$v_i \in e_p \iff i \in p.$$

Thus v_1, \dots, v_k form a k -clique. \square

Proof of Theorem 9: We shall give an fpt-reduction from p -CLIQUE to $\text{HOM}(\mathbf{C}, -)$. Let \mathcal{G} be a graph and $k \geq 1$. Let $K = \binom{k}{2}$. By the Excluded Grid Theorem 5, there is some structure $\mathcal{A} \in \mathbf{C}$ such that the $(k \times K)$ -grid is a minor of the Gaifman graph of the core of \mathcal{A} .

We enumerate the recursively enumerable class \mathbf{C} until we find such an $\mathcal{A} = \mathcal{A}(k)$. Then we compute the core \mathcal{A}' of \mathcal{A} and a minor map μ from the $(k \times K)$ -grid to \mathcal{A}' . We let \mathcal{A}'' be the connected component of \mathcal{A}' that contains the image of μ . \mathcal{A}'' is also a core. Without loss of generality we can assume that μ is a minor map from the $(k \times K)$ -grid onto \mathcal{A}'' . We let $\mathcal{B}' = \mathcal{B}(\mathcal{A}'', \mu, \mathcal{G})$. By Lemma 11 and Lemma 12, there is a homomorphism from \mathcal{A}'' to \mathcal{B}' if, and only if, \mathcal{G} contains a k -clique. Let \mathcal{B} be the disjoint union

of \mathcal{B}' with $\mathcal{A}' \setminus \mathcal{A}''$. Since \mathcal{A}' is a core, every homomorphism from \mathcal{A}' to \mathcal{B} maps \mathcal{A}'' to \mathcal{B}' . Thus there is a homomorphism from \mathcal{A}' to \mathcal{B} if, and only if, \mathcal{G} contains a k -clique. Since \mathcal{A}' is the core of \mathcal{A} , it follows that there is a homomorphism from \mathcal{A} to \mathcal{B} if, and only if, \mathcal{G} has a k -clique.

The construction of \mathcal{A} only depends on k and is effective because \mathbf{C} is recursively enumerable. Computing the core \mathcal{A}' and the minor map μ may require time exponential in the size of \mathcal{A} , but this is still bounded in terms of k . Observe that the size of an r -ary relation $R^{\mathcal{B}}$ is at most

$$|\Pi^{-1}(A^r)| \leq (|V^{\mathcal{G}}| \cdot |E^{\mathcal{G}}| \cdot |A|)^r.$$

Since the arity of \mathbf{C} is bounded, this is polynomial in $\|\mathcal{A}\|$ and $\|\mathcal{G}\|$. It follows that the size of \mathcal{B} is polynomially bounded in terms of $\|\mathcal{A}\|$ and $\|\mathcal{G}\|$, and it is easy to see that \mathcal{B} can actually be computed in polynomial time. This shows that the reduction $(\mathcal{G}, k) \mapsto (\mathcal{A}, \mathcal{B})$ is an fpt-reduction. \square

Proof of Theorem 1: Assume that $\text{FPT} \neq \text{W}[1]$ and let \mathbf{C} be a recursively enumerable class of structures of bounded arity. Trivially, if $\text{HOM}(\mathbf{C}, -)$ is in polynomial time then it is fixed-parameter tractable. By Theorem 9 and our assumption $\text{FPT} \neq \text{W}[1]$, if $p\text{-HOM}(\mathbf{C}, -)$ is fixed-parameter tractable then \mathbf{C} has bounded treewidth modulo homomorphic equivalence. Finally, if \mathbf{C} has bounded treewidth modulo homomorphic equivalence, then $\text{HOM}(\mathbf{C}, -)$ is in polynomial time by Theorem 8. \square

Remark 13. Inspection of our proof shows that we need the recursive enumerability of the class \mathbf{C} only to guarantee the uniformity of the fpt-reduction from $p\text{-CLIQUE}$ to $p\text{-HOM}(\mathbf{C}, -)$. If we strengthen the complexity theoretic assumption to $\text{nonuniform-FPT} \neq \text{nonuniform-W}[1]$, then we can prove the theorem for arbitrary classes \mathbf{C} of bounded arity.

Remark 14. It is not hard to show that for every decidable class \mathbf{C} of structures, the problem $p\text{-HOM}(\mathbf{C}, -)$ is in $\text{W}[1]$. Hence for decidable classes \mathbf{C} , the $\text{W}[1]$ -hardness stated in Theorems 9 and 1 becomes $\text{W}[1]$ -completeness.

5. A characterization of tractability in terms of the underlying graphs

For a class \mathbf{G} of graphs, let $\mathbf{C}(\mathbf{G})$ denote the class of all structures whose Gaifman graph is in \mathbf{G} . The following theorem is a slight improvement of a result due to Grohe, Schwentick, and Segoufin [25].

Theorem 15. *Assume that $\text{FPT} \neq \text{W}[1]$. Let \mathbf{G} be a recursively enumerable class of graphs. Let τ be a vocabulary that is at least binary. Then the following three statements are equivalent:*

- (1) $\text{HOM}(\mathbf{C}(\mathbf{G}), -)$ is in polynomial time.
- (2) $\text{HOM}(\mathbf{C}(\mathbf{G})[\tau], -)$ is in polynomial time.
- (3) \mathbf{G} has bounded treewidth.

The statement proved in [25] is weaker in that it requires the vocabulary τ to contain at least two at least binary relation symbols, one of which needs to be interpreted by equality. Note that (3) states that \mathbf{G} has bounded treewidth and not just bounded treewidth modulo homomorphic equivalence.

Essentially, to derive Theorem 15 from Theorem 1, we have to prove that if \mathbf{G} is a class of graphs of unbounded treewidth, then for every τ that is at least binary, $\mathbf{C}(\mathbf{G})[\tau]$ has unbounded treewidth modulo homomorphic equivalence. Of course it suffices to do this for the “minimal” binary vocabulary $\tau = \{E\}$, where E is a binary relation symbol. Then what we need to prove is that if we have a class of graphs of unbounded treewidth, then we can orient the edges of these graphs in such a way that the resulting class of directed graphs has unbounded treewidth modulo homomorphic equivalence.

It will be convenient to work with *walls* instead of grids. Figure 1 shows the walls of *height* 1, 2, 3, 4, which we denote by $\mathcal{W}_1, \dots, \mathcal{W}_4$. It is obvious from the figure how to define \mathcal{W}_k , the *wall of height* k for every $k \geq 1$. Observe that the $(k+1) \times (k+1)$ -grid is a minor of the wall \mathcal{W}_k . Therefore, the treewidth of \mathcal{W}_k is at least $(k+1)$. Furthermore, the wall \mathcal{W}_k is a subgraph of the $(k+1) \times 2k$ -grid. Thus, by the Excluded Grid Theorem 5, for every k there is a w such that every graph of treewidth at least w has the wall \mathcal{W}_k as a minor. An important difference between walls and grids is that walls have maximum degree 3. We

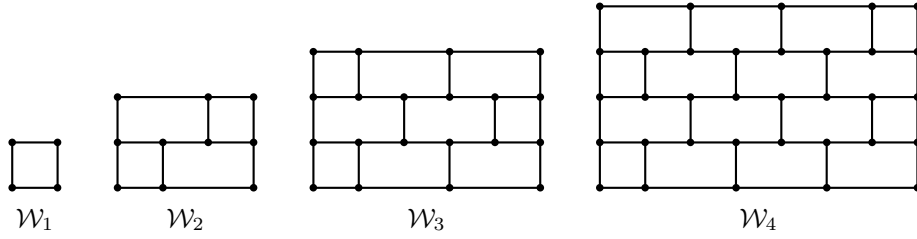


Figure 1. Walls of height 1–4

will exploit the well-known fact that if a graph of maximum degree 3 is a minor of some other graph, then it is actually a topological minor. Let us explain this: A *subdivision* of a graph \mathcal{G} is a graph obtained from \mathcal{G} by replacing the edges of \mathcal{G} by nonintersecting paths. \mathcal{G} is a *topological minor* of \mathcal{H} if some subdivision of \mathcal{G} is (isomorphic to) a subgraph of \mathcal{H} . Clearly, if \mathcal{G} is a topological minor of \mathcal{H} , then it is also a minor. In general, the converse does not hold, but it does if \mathcal{G} is a graph of maximum degree at most 3. To see this, note that contracting an edge e such that both endpoints of e have degree at least 3 yields a new vertex of degree at least 4. Call an edge contraction *topological* if at most one endpoint of the contracted edge has degree at least 3. It follows that a graph of maximum degree 3 is a minor of a graph \mathcal{H} if and only if \mathcal{G} can be obtained from a subgraph of \mathcal{H} by a series of topological edge contractions. But this is precisely the case if \mathcal{G} is a topological minor of \mathcal{H} .

The preceding discussion yields the following lemma:

Lemma 16 ([33]). *A class \mathbf{G} of graphs has unbounded treewidth if and only if for every $k \geq 1$ there is a graph $\mathcal{G} \in \mathbf{G}$ such that the wall \mathcal{W}_k is a topological minor of \mathcal{G} .*

With this lemma at hand, we can turn towards a proof of Theorem 15. An *orientation* of an undirected graph \mathcal{G} is a directed graph obtained from \mathcal{G} by giving each edge one direction.

Lemma 17. *Let \mathcal{G} be a graph such that the wall \mathcal{W}_k is a topological minor of \mathcal{G} . Then \mathcal{G} has an orientation \mathcal{D} such that every directed graph homomorphically equivalent to \mathcal{D} has treewidth at least $(k + 1)$.*

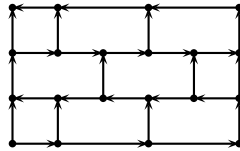


Figure 2. An acyclic orientation of \mathcal{W}_3 with a Hamilton path

Proof: We first observe that the wall \mathcal{W}_k has an acyclic orientation \mathcal{A}_k with a Hamilton path. Figure 2 illustrates this. If h is a homomorphism from \mathcal{A}_k to some acyclic directed graph \mathcal{B} , then h is one-to-one. Obviously, this can be extended to subdivisions of \mathcal{W}_k : If \mathcal{S} is a subdivision of \mathcal{W}_k , then \mathcal{S} has an acyclic orientation \mathcal{A} , which is a “directed subdivision” of \mathcal{A}_k , such that if h is a homomorphism from \mathcal{A} to some acyclic directed graph \mathcal{B} , then h is one-to-one.

Now let \mathcal{G} be a graph that contains \mathcal{W}_k as a topological minor, and let $\mathcal{S} \subseteq \mathcal{G}$ be a subdivision of \mathcal{W}_k . Let \mathcal{A} be an acyclic orientation of \mathcal{S} such that if h is a homomorphism from \mathcal{A} to some acyclic directed graph \mathcal{B} , then h is one-to-one. We extend \mathcal{A} in arbitrary way to an acyclic orientation \mathcal{D} of \mathcal{G} . Let \mathcal{D}' be homomorphically equivalent to \mathcal{D} . Then \mathcal{D}' is also acyclic. Let h be a homomorphism from \mathcal{D} to \mathcal{D}' . Then h is one-to-one. Thus \mathcal{A} is isomorphic to a subgraph of \mathcal{D}' , and therefore \mathcal{W}_k is a topological minor of the Gaifman graph of underlying \mathcal{D}' . As \mathcal{W}_k has treewidth at least $(k + 1)$, so has \mathcal{D}' . \square

Lemma 18. *Let \mathbf{G} be a class of graphs of unbounded treewidth, and τ a vocabulary that is at least binary. Then $\mathbf{C}(\mathbf{G})[\tau]$ has unbounded treewidth modulo homomorphic equivalence.*

Proof: Suppose first that contains τ some binary relation symbol E . By Lemma 16, for every $k \geq 1$ there is a graph $\mathcal{G}_k \in \mathbf{G}$ such that \mathcal{W}_k is a topological minor of \mathcal{G}_k . Let \mathcal{D}_k be the orientation of \mathcal{G}_k obtained from Lemma 17, and let \mathcal{A}_k be the τ -structure with $(A_k, E^{A_k}) = \mathcal{D}_k$ and $R^{A_k} = \emptyset$ for all $R \in \tau \setminus \{E\}$. Then $\mathcal{G}(\mathcal{A}_k) = \mathcal{G}_k$ and hence $\mathcal{A}_k \in \mathbf{C}(\mathbf{G})$. Furthermore, by Lemma 17, \mathcal{A}_k is not homomorphically equivalent to a structure of treewidth less than k . Hence $\mathbf{C}(\mathbf{G})$ does not have bounded treewidth modulo homomorphic equivalence.

If τ does not contain a binary relation symbol, we can simply use a relation symbol R of higher arity, say, r , instead of E and define R^{A_k} to contain all tuples (a_1, \dots, a_r) such that (a_1, a_2) is an edge of \mathcal{D}_k and $a_2 = a_3 = \dots = a_r$. \square

Proof of Theorem 15: The implication (1) \implies (2) is trivial. To prove that (2) \implies (3), suppose that \mathbf{G} has unbounded treewidth. Then by Lemma 18, $\mathbf{C}(\mathbf{G})$ has unbounded treewidth modulo homomorphic equivalence. Thus by Theorem 1, $\text{HOM}(\mathbf{C}(\mathbf{G})[\tau], -)$ is not in polynomial time. To prove that (3) \implies (1), observe that if \mathbf{G} has bounded treewidth, then $\mathbf{C}(\mathbf{G})$ has bounded treewidth and therefore bounded treewidth modulo homomorphic equivalence. Hence, by Theorem 1, $\text{HOM}(\mathbf{C}(\mathbf{G}), -)$ is in polynomial time. \square

6. Applications

6.1. Constraint satisfaction problems. An instance \mathcal{I} of a *constraint satisfaction problem* (CSP) is specified by a set V of *variables*, a *domain* D , and a set \mathcal{C} of constraints of the form $R^{\mathcal{I}}x_1 \dots x_r$, where $r \geq 1$, $x_1, \dots, x_r \in V$, and $R^{\mathcal{I}}$ an r -ary relation on the domain D . A *solution* for \mathcal{I} is an assignment $h : V \rightarrow D$ such that for all constraints $R^{\mathcal{I}}x_1 \dots x_r \in \mathcal{C}$ we have $(h(x_1), \dots, h(x_r)) \in R^{\mathcal{I}}$. The relations $R^{\mathcal{I}}$ appearing in \mathcal{C} form the *constraint language* of the instance \mathcal{I} .

Feder and Vardi [17] observed that constraint satisfaction problems can be described as homomorphism problems for relational structures. With every CSP-instance $\mathcal{I} = (V, D, \mathcal{C})$ we associate two structures $\mathcal{A}(\mathcal{I})$ and $\mathcal{B}(\mathcal{I})$ as follows: The vocabulary $\tau(\mathcal{I})$ of both structures contains an r -ary relation symbol R for every r -ary relation $R^{\mathcal{I}}$ in the constraint language of \mathcal{I} . The universe of $\mathcal{B}(\mathcal{I})$ is D , and the relations of \mathcal{B} are those appearing in the constraint language. More precisely, for every $R \in \tau(\mathcal{I})$ we let $R^{\mathcal{B}(\mathcal{I})} = R^{\mathcal{I}}$. The universe of $\mathcal{A}(\mathcal{I})$ is V , and for each r -ary relation symbol $R \in \tau(\mathcal{I})$ we let $R^{\mathcal{A}(\mathcal{I})} = \{(x_1, \dots, x_r) \mid R^{\mathcal{I}}x_1 \dots x_r \in \mathcal{C}\}$. Then a mapping $h : V \rightarrow D$ is a solution for \mathcal{I} if and only if it is a homomorphism from $\mathcal{A}(\mathcal{I})$ to $\mathcal{B}(\mathcal{I})$. Thus instance \mathcal{I} is satisfiable if and only if there is a homomorphism from $\mathcal{A}(\mathcal{I})$ to $\mathcal{B}(\mathcal{I})$. Conversely, for all pairs of structures \mathcal{A}, \mathcal{B} of the same vocabulary, we can easily construct a CSP-instance \mathcal{I} such that $\mathcal{A}(\mathcal{I}) = \mathcal{A}$ and $\mathcal{B}(\mathcal{I}) = \mathcal{B}$.

For classes \mathbf{C}, \mathbf{D} of relational structures we let $\text{CSP}(\mathbf{C}, \mathbf{D})$ be the restricted constraint satisfaction problem with instances \mathcal{I} satisfying $\mathcal{A}(\mathcal{I}) \in \mathbf{C}$ and $\mathcal{B}(\mathcal{I}) \in \mathbf{D}$. If \mathbf{C} or \mathbf{D} is the class of all structures, we use the notation $\text{CSP}(-, \mathbf{D})$ or $\text{CSP}(\mathbf{C}, -)$, respectively. By the preceding discussion, for all classes \mathbf{C}, \mathbf{D} of relational structures the problems $\text{CSP}(\mathbf{C}, \mathbf{D})$ and $\text{HOM}(\mathbf{C}, \mathbf{D})$ are polynomial time equivalent. In the literature on constraint satisfaction problems, restrictions on the class \mathbf{C} , that is, on the structure $\mathcal{A}(\mathcal{I})$ imposed by the constraints on the variables, are often referred to as *structural restrictions*. Restrictions on \mathbf{D} are referred to as *constraint language restrictions*. Another distinction made is between *nonuniform CSP*, where domain and constraint language (and hence the structure $\mathcal{B}(\mathcal{I})$) are fixed, and *uniform CSP*, where they are part of the input [29]. Considerable efforts have been made towards identifying restrictions that lead to tractable problems; recent results include [4, 5, 6, 9, 10, 11, 24]. Our main result implies an almost complete characterization of the tractable structural restrictions for uniform CSP:

Corollary 19. *Assume that $\text{FPT} \neq \text{W}[1]$. Then for every recursively enumerable class \mathbf{C} of structures of bounded arity, $\text{CSP}(\mathbf{C}, -)$ is in polynomial time if and only if \mathbf{C} has bounded treewidth modulo homomorphic equivalence.*

It is an interesting question what happens in the nonuniform case. Let us call a class \mathbf{C} a *nonuniformly tractable structural restriction* if $\text{CSP}(\mathbf{C}, \{\mathcal{B}\})$ is in polynomial time for all structures \mathcal{B} . Now we may ask

what the nonuniformly tractable restrictions are. Clearly, if $\text{CSP}(\mathbf{C}, -)$ is in polynomial time then \mathbf{C} is a nonuniformly tractable structural restriction. Thus all classes \mathbf{C} of bounded treewidth modulo homomorphic equivalence are. However, we can go beyond bounded treewidth modulo homomorphic equivalence: Let us say that a class \mathbf{C} of structures has *logarithmic treewidth* if there is a constant c such that for all $\mathcal{A} \in \mathbf{C}$ it holds that $\text{tw}(\mathcal{A}) \leq c \cdot \log \|\mathcal{A}\|$.

Proposition 20. *Every class \mathbf{C} of structures of logarithmic treewidth is a nonuniformly tractable structural restriction.*

Proof: If \mathbf{C} has logarithmic treewidth, then for every fixed structure \mathcal{B} with d elements we can solve $\text{HOM}(\mathbf{C}, \{\mathcal{B}\})$ in time $n^{O(\log d)}$ by a straightforward dynamic programming algorithm on a tree decomposition of logarithmic width of the input structure \mathcal{A} . Such a decomposition can be computed in polynomial time by an algorithm due to Robertson and Seymour [34] (also see [32, 19]) that, given an n -vertex graph \mathcal{G} of treewidth k , computes a tree-decomposition of \mathcal{G} of width at most $4k + 1$ in time $2^{O(k)} \cdot n^2$. \square

It is an open question whether this tractability result can be extended to classes \mathbf{C} of structures that have logarithmic treewidth modulo homomorphic equivalence. The algorithm underlying Theorem 8 is only quasi-polynomial in this case.

6.2. Conjunctive query evaluation. It is well known that relational databases can be modeled by finite relational structures and that first-order logic forms the core of the standard query language SQL (see, for example, [1]). A particularly important class of queries is the class of *conjunctive queries*, which can be defined by formulas of first-order logic of the following form:

$$\varphi(x_1, \dots, x_k) = \exists x_{k+1} \dots \exists x_\ell (\alpha_1 \wedge \dots \wedge \alpha_m), \quad (1)$$

where each *atom* α_i is of the form $Rx_{j_1} \dots x_{j_r}$ for an r -ary relation symbol R and indices $j_1, \dots, j_r \in [\ell]$. Let $\tau(\varphi)$ be the vocabulary that contains all relation symbols appearing in φ . For a $\tau(\varphi)$ -structure \mathcal{B} and an ℓ -tuple $\bar{b} = (b_1, \dots, b_\ell) \in B^\ell$, we say that \mathcal{B}, \bar{b} *satisfies* $\alpha_i = Rx_{j_1} \dots x_{j_r}$ if $(b_{j_1}, \dots, b_{j_r}) \in R^{\mathcal{B}}$. For a k -tuple $\bar{b} = (b_1, \dots, b_k) \in B^k$, we say that \mathcal{B}, \bar{b} *satisfies* $\varphi(x_1, \dots, x_k)$ (and write $\mathcal{B} \models \varphi(\bar{b})$) if there are b_{k+1}, \dots, b_ℓ such that $\mathcal{B}, (b_1, \dots, b_\ell)$ satisfies $\alpha_1, \dots, \alpha_m$. We let

$$\varphi(\mathcal{B}) = \{\bar{b} \in B^k \mid \mathcal{B} \models \varphi(\bar{b})\}.$$

Intuitively, $\varphi(\mathcal{B})$ is the answer of the query φ in the “database” \mathcal{B} . The *conjunctive query evaluation problem* is to compute $\varphi(\mathcal{B})$ for given φ and \mathcal{B} . For every class Φ of conjunctive queries and every class \mathbf{D} of structures, we let $\text{EVAL}(\Phi, \mathbf{D})$ be the problem of computing $\varphi(\mathcal{B})$ for a given $\varphi \in \Phi$ and $\mathcal{B} \in \mathbf{D}$. As usually, we write $\text{EVAL}(\Phi, -)$ if \mathbf{D} is the class of all structures.

Observe the close relationship between query evaluation and constraint satisfaction problems [29]: Let φ be a conjunctive query and \mathcal{B} a structure as above. Let $\mathcal{I} = (V, D, \mathcal{C})$ be the CSP-instance with $V = \{x_1, \dots, x_k\}$, $D = B$, and $\mathcal{C} = \{\alpha_1, \dots, \alpha_m\}$, where we identify the atom $\alpha_i = Rx_{j_1} \dots x_{j_r}$ with the constraint $R^{\mathcal{B}}x_{j_1} \dots x_{j_r}$. Then for every k -tuple $\bar{b} = (b_1, \dots, b_k) \in B^k$ it holds that $\mathcal{B} \models \varphi(\bar{b})$ if and only if there is a solution $h : V \rightarrow D$ for \mathcal{I} with $h(x_i) = b_i$ for $i \in [k]$. Thus “solutions” to the query evaluation problem, that is, tuples in the query answer, are projections of solutions to the constraint satisfaction problem. However, there is an important difference: In the query evaluation problem, we have to compute all tuples in the query answer, whereas in the constraints satisfaction problem we are usually happy to find just one solution or decide if a solution exists, if we view the problem as a decision problem, which we do in this paper.

To apply our main result to the conjunctive query evaluation problem, we have to define an analogue of structural restrictions for conjunctive queries. Let $\varphi(x_1, \dots, x_k)$ be a conjunctive query. Recall that the vocabulary $\tau(\varphi)$ consists of all relation symbols that occur in φ , and let F_φ be a new k -ary relation symbol not contained in $\tau(\varphi)$. We define a $\tau(\varphi) \cup \{F_\varphi\}$ -structure $\mathcal{A}(\varphi)$ as follows: The universe of $\mathcal{A}(\varphi)$ is the set of variables occurring in φ . For every relation symbol $R \in \tau(\varphi)$, we let $R^{\mathcal{A}(\varphi)} = \{(x_{j_1}, \dots, x_{j_r}) \mid Rx_{j_1} \dots x_{j_r} \text{ is an atom of } \varphi\}$. Moreover, we let $F_\varphi^{\mathcal{A}(\varphi)} = \{(x_1, \dots, x_k)\}$. Recall that $\mathcal{A}(\varphi)|_{\tau(\varphi)}$ denotes the $\tau(\varphi)$ -reduct of $\mathcal{A}(\varphi)$. It is easy to see that for every $\tau(\varphi)$ -structure \mathcal{B} ,

$$\varphi(\mathcal{B}) = \left\{ (h(x_1), \dots, h(x_k)) \mid h \text{ is a homomorphism from } \mathcal{A}(\varphi)|_{\tau(\varphi)} \text{ to } \mathcal{B} \right\}.$$

For every class Φ of conjunctive queries, we let $\mathbf{C}(\Phi) = \{\mathcal{A}(\varphi) \mid \varphi \in \Phi\}$. We say that Φ has *bounded arity* if there is a $k \in \mathbb{N}$ such that for all $\varphi \in \Phi$, the vocabulary $\tau(\varphi)$ is at most k -ary. Note that if $\mathbf{C}(\Phi)$ has bounded arity then so has Φ , but that the converse is not always true. Instead, $\mathbf{C}(\Phi)$ has bounded arity if and only if Φ has bounded arity and there is a $k \in \mathbb{N}$ such that every $\varphi \in \Phi$ has at most k free variables.

Corollary 21. *Assume that $\text{FPT} \neq \text{W}[1]$. Then for every recursively enumerable class Φ of conjunctive queries of bounded arity, $\text{EVAL}(\Phi, -)$ is in polynomial time if and only if $\mathbf{C}(\Phi)$ has bounded treewidth modulo homomorphic equivalence.*

Proof: For the forward direction, suppose that $\text{EVAL}(\Phi, -)$ is in polynomial time.

We first prove that $\mathbf{C}(\Phi)$ has bounded arity. As $\text{EVAL}(\Phi, -)$ is in polynomial time, the size of the query answers $\varphi(\mathcal{B})$ must be polynomially bounded. Hence there exists a $k_0 \in \mathbb{N}$ such that for every $\varphi \in \Phi$ and every sufficiently large \mathcal{B} it holds that $|\varphi(\mathcal{B})| \leq \|\mathcal{B}\|^{k_0}$. Let $k_1 \in \mathbb{N}$ such that $\tau(\varphi)$ is at most k_1 -ary for every $\varphi \in \Phi$ and suppose for contradiction that there is a $\varphi \in \Phi$ with more than $k_0 \cdot k_1$ free variables. Let us fix such a φ , and let $k > k_0 \cdot k_1$ be the number of free variables of φ . For every $n \geq 1$, let \mathcal{B}_n be the $\tau(\varphi)$ -structure with universe $[n]$ and $R^{\mathcal{B}} = [n]^r$ for every r -ary $R \in \tau(\varphi)$. Then $\varphi(\mathcal{B}_n) = [n]^k$, because every tuple of elements of \mathcal{B}_n satisfies all atoms of φ . However, $\|\mathcal{B}_n\| = O(n^{k_1})$ (where the big-Oh constant may depend on φ) and therefore

$$|\varphi(\mathcal{B}_n)| \leq \|\mathcal{B}_n\|^{k_0} = O(n^{k_0 \cdot k_1}) = o(n^k) = o(|\varphi(\mathcal{B}_n)|).$$

This is a contradiction, and thus $\mathbf{C}(\Phi)$ must have bounded arity.

Hence by Theorem 1, it suffices to prove that $\text{HOM}(\mathbf{C}(\Phi), -)$ is in polynomial time. So suppose we are given an instance \mathcal{A}, \mathcal{B} of $\text{HOM}(\mathbf{C}(\Phi), -)$, where $\mathcal{A} = \mathcal{A}(\varphi)$ for some $\varphi = \varphi(x_1, \dots, x_k) \in \Phi$ and \mathcal{B} is an arbitrary $\tau(\varphi) \cup \{F_\varphi\}$ -structure. Using the polynomial time algorithm for $\text{EVAL}(\Phi, -)$, we can enumerate the set

$$\varphi(\mathcal{B}|_{\tau(\varphi)}) = \left\{ (h(x_1), \dots, h(x_k)) \mid h \text{ is a homomorphism from } \mathcal{A}|_{\tau(\varphi)} \text{ to } \mathcal{B}|_{\tau(\varphi)} \right\}.$$

To find out if there is a homomorphism from \mathcal{A} to \mathcal{B} , we simply have to check whether the set $\varphi(\mathcal{B}|_{\tau(\varphi)})$ contains a tuple in $F_\varphi^{\mathcal{B}}$.

For the backward direction, suppose that $\mathbf{C}(\Phi)$ has bounded treewidth modulo homomorphic equivalence. Then by Theorem 8, $\text{HOM}(\mathbf{C}(\Phi), -)$ is in polynomial time. We shall prove that $\text{EVAL}(\Phi, -)$ is in polynomial time as well.

Again, we prove first that there is a bound on the number of free variables of the queries in Φ . Let $k_0 \in \mathbb{N}$ such that every $\mathcal{A} \in \mathbf{C}(\Phi)$ is homomorphically equivalent to a structure of treewidth at most k_0 . Let $\varphi(x_1, \dots, x_k) \in \Phi$, where we assume that the variables x_1, \dots, x_k are distinct. We shall prove that $k \leq k_0 + 1$. Consider the structure $\mathcal{A} = \mathcal{A}(\varphi)$. The relation $F_\varphi^{\mathcal{A}}$ contains only one tuple (x_1, \dots, x_k) . Hence $(x_1, \dots, x_k) \in F^{\mathcal{A}'}$ for the core \mathcal{A}' of \mathcal{A} . As x_1, \dots, x_k are pairwise distinct, this implies that the treewidth of \mathcal{A}' is at least $k - 1$, and thus $k \leq k_0 + 1$.

To prove that $\text{EVAL}(\Phi, -)$ is in polynomial time, consider an instance φ, \mathcal{B} , where $\varphi = \varphi(x_1, \dots, x_k) \in \Phi$ and \mathcal{B} is a $\tau(\varphi)$ -structure. Let $\mathcal{A} = \mathcal{A}(\varphi)$. For every k -tuple $\bar{b} \in B^k$, let (\mathcal{B}, \bar{b}) denote the $\tau(\varphi) \cup \{F_\varphi\}$ -expansion of \mathcal{B} with $F_\varphi^{(\mathcal{B}, \bar{b})} = \{(b_1, \dots, b_k)\}$. Then

$$\begin{aligned} \varphi(\mathcal{B}) &= \left\{ (h(x_1), \dots, h(x_k)) \mid h \text{ is a homomorphism from } \mathcal{A}(\varphi)|_{\tau(\varphi)} \text{ to } \mathcal{B} \right\} \\ &= \{ \bar{b} \in B^k \mid \text{there is a homomorphism from } \mathcal{A}(\varphi) \text{ to } (\mathcal{B}, \bar{b}) \}. \end{aligned}$$

Using the polynomial time algorithm for $\text{HOM}(\mathbf{C}(\Phi), -)$, this set can be computed in polynomial time, because k is bounded by the constant $k_0 + 1$. \square

It is an interesting open problem to characterize those classes of conjunctive queries for which the answer can be computed in *polynomial total time* (that is, in time polynomial in the size of the input and the size of the output, which can be exponential in the size of the input) or by a *polynomial delay algorithm* (that is, an algorithm that may run for an exponential number of steps, but produces at least one tuple in

the solution every polynomial number of steps). This question is even interesting for the special case of conjunctive queries without any existential quantifiers, that is, queries where all variables are free. In this case, the problem is equivalent to the problem of enumerating all solutions for a given CSP-instance. It is proved in [8] (also see [18]) that for classes Φ of conjunctive queries of bounded treewidth, the problem $\text{EVAL}(\Phi, -)$ is in polynomial total time, and it is not hard to show that it has a polynomial delay algorithm. It is not clear at all whether this can be extended to classes Φ of bounded treewidth modulo homomorphic equivalence.

7. Complexity theoretic remarks

The following result shows that the assumption $\text{FPT} \neq \text{W}[1]$ is not only sufficient to prove our main theorem, but also necessary. Recall that a function $f : \mathbb{N} \rightarrow \mathbb{N}$ is *time constructible* if there is a deterministic Turing machine that for all $n \in \mathbb{N}$ on every input of length n halts in exactly $f(n)$ steps. Clearly, for every computable function f there is time constructible function g such that $f(n) \leq g(n)$ for all $n \in \mathbb{N}$.

Proposition 22. *Suppose that $\text{FPT} = \text{W}[1]$. Then there is a polynomial time decidable class \mathbf{C} of graphs of unbounded treewidth modulo homomorphic equivalence such that $\text{HOM}(\mathbf{C}, -)$ is in polynomial time.*

Proof: $\text{FPT} = \text{W}[1]$ implies that the parameterized clique problem is fixed-parameter tractable. Suppose p -CLIQUE is solvable in time $f(k) \cdot n^c$ for some computable function f and constant c . Without loss of generality we may assume that f is time constructible and that $f(k) \geq k$ for all $k \geq 1$.

For $k \geq 1$, let \mathcal{G}_k be the graph with vertex set $[f(k)]$ and edges $\{i, j\}$ for $1 \leq i < j \leq k$. Thus \mathcal{G}_k is a clique of size k padded with $f(k) - k$ isolated vertices. The core of \mathcal{G}_k is a clique of size k and thus has treewidth $k - 1$. Let \mathbf{C} be the class of all \mathcal{G}_k , for $k \geq 1$. As the cores of the structures in \mathbf{C} have unbounded treewidth, \mathbf{C} has unbounded treewidth modulo homomorphic equivalence. Since f is time constructible, \mathbf{C} is decidable in polynomial time.

We shall now describe a polynomial time algorithm for $\text{HOM}(\mathbf{C}, -)$. Let $\mathcal{G} \in \mathbf{C}$, say, $\mathcal{G} = \mathcal{G}_k$, and let \mathcal{H} be an arbitrary graph. There is a homomorphism from \mathcal{G} to \mathcal{H} if, and only if, \mathcal{H} contains a k -clique. Deciding whether \mathcal{H} contains a k -clique requires time $f(k) \cdot \|\mathcal{H}\|^c \leq \|\mathcal{G}\| \cdot \|\mathcal{H}\|^c$ if we use the fpt -algorithm for p -CLIQUE. \square

A similar result holds for the nonuniform version of the assumption:

Proposition 23. *Suppose that $\text{nonuniform-FPT} = \text{nonuniform-W}[1]$. Then there is a class \mathbf{C} of graphs of unbounded treewidth modulo homomorphic equivalence such that $\text{HOM}(\mathbf{C}, -)$ is in polynomial time.*

Proof: $\text{nonuniform-FPT} = \text{nonuniform-W}[1]$ implies that p -CLIQUE is in nonuniform-FPT . This means that there exists a constant c and, for every $k \geq 1$, a Turing machine M_k and a constant d_k that decides if its input x is (the encoding of a) graph that contains a k -clique in at most $d_k \cdot |x|^c$ steps. Let us fix such a c and families $(M_k)_{k \geq 1}$, $(d_k)_{k \geq 1}$.

It is easy to find an enumeration $(N_1, e_1), (N_2, e_2), \dots$ of all pairs (N, e) consisting of a Turing machine N and a natural number e such that there is a polynomial time algorithm A that, given a natural number i in unary and an input x , decides if N_i accepts x in at most $e_i |x|^c$ steps.

Let $f : \mathbb{N} \rightarrow \mathbb{N}$ such that $N_{f(k)} = M_k$ and $e_{f(k)} = d_k$. Note that f is not necessarily computable. Without loss of generality we can assume that $f(k) \geq k$ for all $k \geq 1$. So given $f(k)$ in unary and a graph \mathcal{H} , the algorithm A decides in polynomial time whether \mathcal{H} contains a k -clique.

We define a class \mathbf{C} of graphs as follows: For every $k \geq 1$, again we let \mathcal{G}_k be the graph with vertex set $[f(k)]$ and edges $\{i, j\}$ for $1 \leq i < j \leq k$. Thus \mathcal{G}_k is a clique of size k padded with $f(k) - k$ isolated vertices. Let \mathbf{C} be the class of all \mathcal{G}_k , for $k \geq 1$. Clearly, \mathbf{C} has unbounded treewidth modulo homomorphic equivalence.

We shall now describe a polynomial time algorithm for $\text{HOM}(\mathbf{C}, -)$. Let $\mathcal{G} \in \mathbf{C}$, say, $\mathcal{G} = \mathcal{G}_k$, and let \mathcal{H} be an arbitrary graph. There is a homomorphism from \mathcal{G} to \mathcal{H} if, and only if, \mathcal{H} contains a k -clique. We can decide in polynomial time whether \mathcal{H} contains a k -clique by executing our algorithm A on input $f(k) = |V^{\mathcal{G}}|, \mathcal{H}$. \square

Our last result gives some evidence that there are problems of the form $\text{HOM}(\mathbf{C}, -)$ which are in NP, but neither in PTIME nor NP-complete. Consider the following problem:

LOG-CLIQUE

Instance: Graph \mathcal{G} .

Problem: Decide if \mathcal{G} has a clique of size at least $\log |V^{\mathcal{G}}|$.

LOG-CLIQUE has been considered by Papadimitriou and Yannakakis [30]. It is unlikely that LOG-CLIQUE is in PTIME because that would imply that the parameterized clique problem is fixed-parameter tractable and thus $\text{FPT} = \text{W}[1]$. It also seems unlikely that LOG-CLIQUE is NP-complete because that would imply that $\text{NP} \subseteq \text{DTIME}(n^{O(\log n)})$.

We call two problems P and Q *polynomial time equivalent* if there is a polynomial time reduction from P to Q and, vice versa, a polynomial time reduction from Q to P .

Proposition 24. *There is a polynomial time decidable class \mathbf{C} of graphs such that $\text{HOM}(\mathbf{C}, -)$ and LOG-CLIQUE are polynomial time equivalent.*

Proof: For every $n \geq 1$, let \mathcal{G}_n be the graph with vertex set $[n]$ and edges $\{i, j\}$ for $1 \leq i < j \leq \lceil \log n \rceil$, and let $\mathbf{C} = \{\mathcal{G}_n \mid n \geq 1\}$.

To see that LOG-CLIQUE is reducible to $\text{HOM}(\mathbf{C}, -)$, we just observe that a graph \mathcal{H} with n vertices has a clique of size at least $\log n$ if, and only if, there is a homomorphism from \mathcal{G}_n to \mathcal{H} .

To reduce $\text{HOM}(\mathbf{C}, -)$ to LOG-CLIQUE, we proceed as follows: Suppose we are given a graph $\mathcal{G} \in \mathbf{C}$ and a graph \mathcal{H} . Suppose that $\mathcal{G} = \mathcal{G}_n$ and thus $n = |V^{\mathcal{G}}|$. Let $n' = |V^{\mathcal{H}}|$, $\ell = \lceil \log n \rceil$, and $\ell' = \lceil \log n' \rceil$.

We shall define a graph \mathcal{H}' such that there is a homomorphism from \mathcal{G} to \mathcal{H} if, and only if, \mathcal{H}' has a clique of size at least $\log |V^{\mathcal{H}'}|$.

If $\ell = \ell'$ then we simply let $\mathcal{H}' = \mathcal{H}$.

If $\ell > \ell'$ then we let \mathcal{H}' be the graph obtained from \mathcal{H} by adding $n - |V^{\mathcal{H}}|$ isolated vertices.

If $\ell < \ell'$ then we define \mathcal{H}' as follows: We add n' new vertices $v_1, \dots, v_{n'}$ to \mathcal{H} . Let $k = \ell' + 1 - \ell$. We add edges between v_i and v_j for $1 \leq i < j \leq k$ and between v_i and all vertices of \mathcal{H} for $1 \leq i \leq k$. Observe that $\lceil \log |V^{\mathcal{H}'}| \rceil = \lceil \log |V^{\mathcal{H}}| \rceil + 1 = \ell' + 1$. Furthermore, \mathcal{H}' has an $\ell + k = \ell' + 1$ -clique if, and only if, \mathcal{H} has an ℓ -clique, which is the case if, and only if, there is a homomorphism from \mathcal{G} to \mathcal{H} . \square

8. Conclusions

We give a characterisation of all tractable problems of the form $\text{HOM}(\mathbf{C}, -)$, for classes \mathbf{C} of bounded arity, in terms of the treewidth modulo homomorphic equivalence of the structures in \mathbf{C} . Given the large variety of classes \mathbf{C} , it is quite surprising that such a clear cut combinatorial characterisation of tractability exists at all. Furthermore, this characterisation coincides for two natural notions of tractability, polynomial time decidability and fixed-parameter tractability.

It is easy to see that our characterization of tractable problems $\text{HOM}(\mathbf{C}, -)$ fails for classes \mathbf{C} of unbounded arity: For example, consider the class \mathbf{C} of all structures with universe $[n]$ and one n -ary relation that only contains the tuple $(1, \dots, n)$. Clearly, this class has unbounded treewidth modulo homomorphic equivalence, but it is easy to see that $\text{HOM}(\mathbf{C}, -)$ is nevertheless in polynomial time. For a while, it looked as if *bounded hypertree width modulo homomorphic equivalence* [21] (also see [3, 9, 10, 22]) could be the right tractability criterion for classes \mathbf{C} of unbounded arity, but recently it was proved that a bounded fractional edge cover number of the hypergraph associated with the structures in \mathbf{C} also implies tractability [24] and goes beyond bounded hypertree width modulo homomorphic equivalence. There are good reasons to believe that the right combinatorial characterization of tractable $\text{HOM}(\mathbf{C}, -)$ is bounded fractional hypertree width modulo homomorphic equivalence [24], which is a combination of bounded hypertree width and bounded fractional edge cover number. However, currently it is neither known whether $\text{HOM}(\mathbf{C}, -)$ is in polynomial time for all classes \mathbf{C} bounded fractional hypertree width (it is known that the parameterized problem $p\text{-HOM}(\mathbf{C}, -)$ is tractable for such classes) nor whether the tractability of $\text{HOM}(\mathbf{C}, -)$ implies that \mathbf{C} has bounded fractional hypertree width modulo homomorphic equivalence.

Shortly after the conference version of this article [23] was published, Dalmau and Jonsson [12] obtained a similar characterization for the problem of *counting homomorphisms* between two structures. The combinatorial criterion for tractability of the counting problem is just bounded treewidth (homomorphic equivalence is not needed). It is open to characterize the tractable instances of the corresponding *construction problem* (“Given \mathcal{A} , \mathcal{B} compute a homomorphism from \mathcal{A} to \mathcal{B} if there exists one.”) and the *listing problem* (“Given \mathcal{A} , \mathcal{B} compute all homomorphism from \mathcal{A} to \mathcal{B} .”), the latter with respect to polynomial total time or polynomial delay algorithms.

Related to the homomorphism problem is the *embedding problem*, where by embedding we simply mean one-to-one homomorphism. The embedding problem for graphs is also known as the *subgraph isomorphism problem*. Again we are interested in restrictions $\text{EMB}(\mathbf{C}, -)$ of the embedding problem. Note first that $\text{EMB}(\mathbf{C}, -)$ is NP-complete even if \mathbf{C} is the class of all paths, because the Hamiltonian path problem is reducible to the embedding problem for paths. On the other hand, the naturally parameterized problem $p\text{-EMB}(\mathbf{C}, -)$ is fixed-parameter tractable for all classes \mathbf{C} of bounded treewidth [31]. I conjecture that this is optimal, that is, that $p\text{-EMB}(\mathbf{C}, -)$ is fixed parameter tractable if, and only if, \mathbf{C} has bounded treewidth and that it is W[1]-complete otherwise. A proof of this conjecture would also solve another notorious open problem in parameterized complexity theory: Let $\mathcal{K}_{k,k}$ be the complete bipartite graph with k vertices on both sides of the bipartition. Is the problem of deciding whether a given graph contains $\mathcal{K}_{k,k}$ as a subgraph, parameterized by k , W[1]-complete?⁴ It looks as if it should be easy to prove that the problem is indeed W[1]-complete, but so far this eluded all efforts.

Acknowledgements. I would like to thank the anonymous referee for pointing out a mistake in an earlier version of Corollary 21.

References

- [1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [2] K.A. Abrahamson, R.G. Downey, and M.R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogs. *Annals of Pure and Applied Logic*, 73:235–276, 1995.
- [3] I. Adler, G. Gottlob, and M. Grohe. Hypertree-width and related hypergraph invariants. In S. Felsner, editor, *Proceedings of the 3rd European Conference on Combinatorics, Graph Theory, and Applications*, volume AE of *DMTCS Proceedings Series*, pages 5–10, 2005.
- [4] A. A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 649–658, 2002.
- [5] A. A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science*, pages 321–330, 2003.
- [6] A. A. Bulatov, A. A. Krokhnin, and P. Jeavons. The complexity of maximal constraint languages. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 667–674, 2001.
- [7] A.K. Chandra and P.M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th ACM Symposium on Theory of Computing*, pages 77–90, 1977.
- [8] Ch. Chekuri and A. Rajaraman. Conjunctive query containment revisited. In Ph. Kolaitis and F. Afrati, editors, *Proceedings of the 5th International Conference on Database Theory*, volume 1186 of *Lecture Notes in Computer Science*, pages 56–70. Springer-Verlag, 1997.
- [9] H. Chen and V. Dalmau. Beyond hypertree width: Decomposition methods without decompositions. In *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming*, 2005. To appear.

⁴I first heard about this problem from Markus Schaefer.

- [10] D. Cohen, P. Jeavons, and M. Gyssens. A unified theory of structural tractability for constraint satisfaction and spread cut decomposition. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005. To appear.
- [11] V. Dalmau. Generalized majority-minority operations are tractable. In *Proceedings of the 20th IEEE Symposium on Logic in Computer Science*, pages 438–447, 2005.
- [12] V. Dalmau and P. Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoretical Computer Science*, 329:315–323, 2004.
- [13] V. Dalmau, Ph. G. Kolaitis, and M. Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In P. Van Hentenryck, editor, *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*, volume 2470 of *Lecture Notes in Computer Science*, pages 310–326. Springer-Verlag, 2002.
- [14] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24:873–921, 1995.
- [15] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141:109–131, 1995.
- [16] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
- [17] T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1998.
- [18] J. Flum, M. Frick, and M. Grohe. Query evaluation via tree-decompositions. *Journal of the ACM*, 49(6):716–752, 2002.
- [19] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.
- [20] E.C. Freuder. Complexity of k -tree structured constraint satisfaction problems. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 4–9, 1990.
- [21] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64:579–627, 2002.
- [22] G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: Game theoretic and logical characterizations of hypertree width. *Journal of Computer and System Sciences*, 66:775–808, 2003.
- [23] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 552–561, 2003.
- [24] M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006.
- [25] M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 657–666, 2001.
- [26] P. Hell and J. Nešetřil. On the complexity of H -coloring. *Journal of Combinatorial Theory, Series B*, 48:92–110, 1990.
- [27] P. Hell and J. Nešetřil. The core of a graph. *Discrete Mathematics*, 109:117–126, 1992.
- [28] Ph.G. Kolaitis and M.Y. Vardi. On the expressive power of datalog: tools and a case study. *Journal of Computer and System Sciences*, 51(1):110–134, 1995.
- [29] Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, pages 205–213, 1998.

- [30] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the VC-dimension. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 12–18, 1993.
- [31] J. Plehn and B. Voigt. Finding minimally weighted subgraphs. In R. Möhring, editor, *Graph-Theoretic Concepts in Computer Science, WG '90*, volume 484 of *Lecture Notes in Computer Science*, pages 18–29. Springer-Verlag, 1990.
- [32] B. Reed. Tree width and tangles: A new connectivity measure and some applications. In R.A. Bailey, editor, *Surveys in Combinatorics*, volume 241 of *LMS Lecture Note Series*, pages 87–162. Cambridge University Press, 1997.
- [33] N. Robertson and P.D. Seymour. Graph minors V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41:92–114, 1986.
- [34] N. Robertson and P.D. Seymour. Graph minors XII. Distance on a surface. *Journal of Combinatorial Theory, Series B*, 64:240–272, 1995.
- [35] N. Robertson, P.D. Seymour, and R. Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62:323–348, 1994.
- [36] M.Y. Vardi. Constraint satisfaction and database theory: A tutorial. In *Proceedings of the 19th ACM Symposium on Principles of Database Systems*, pages 76–85, 2000.