Fixed-Parameter Tractability, Definability, and Model Checking

Jörg Flum [*] Martin Grohe [†]

February 12, 2001

Abstract

In this article, we study parameterized complexity theory from the perspective of logic, or more specifically, descriptive complexity theory.

We propose to consider parameterized *model-checking* problems for various fragments of first-order logic as generic parameterized problems and show how this approach can be useful in studying both fixed-parameter tractability and intractability. For example, we establish the equivalence between the model-checking for existential first-order logic, the homomorphism problem for relational structures, and the substructure isomorphism problem. Our main tractability result shows that model-checking for first-order formulas is fixed-parameter tractable when restricted to a class of input structures with an excluded minor. On the intractability side, for every $t \geq 0$ we prove an equivalence between model-checking for first-order formulas with t quantifier alternations and the parameterized halting problem for alternating Turing machines with t alternations. We discuss the close connection between this *alternation hierarchy* and Downey and Fellows' W-hierarchy.

On a more abstract level, we consider two forms of definability, called *Fagin definability* and *slicewise definability*, that are appropriate for describing parameterized problems. We give a characterization of the class FPT of all fixed-parameter tractable problems in terms of slicewise definability in finite variable least fixed-point logic, which is reminiscent of the Immerman-Vardi Theorem characterizing the class PTIME in terms of definability in least fixed-point logic.

1 Introduction

Parameterized complexity is a branch of complexity theory which has matured in the last 10 years, as witnessed in the culminating monograph [10]. It gives a framework for a refined complexity analysis of hard algorithmic problems. The basic idea can best be explained by an example: Consider the problem of evaluating a query in a relational database. This problem usually has a high complexity (depending on the query language, of course, but the problem is NP-complete even for the very basic conjunctive queries [5]). The main factor contributing to this complexity is the length of the query. In practice, however, queries are usually short, certainly much shorter than the size of the database. Thus when analyzing the complexity of the problem we should put much more emphasis on the size of the database than on the length of the query. An algorithm evaluating a query of length k in a database of size m in time $O(2^k \cdot m)$ is therefore much better than one performing the same task in time $O(m^{k/2})$, although both are exponential.

Parameterized complexity theory studies problems whose instances are *parameterized* by some function of the input, such as the length of the query in our example. The idea is to choose the parameterization in such a way that it can be assumed to take small values for the instances one is interested in. Then the complexity of an algorithm is measured not only in the size of the input, but also in terms of the parameter. A parameterized problem is *fixed-parameter tractable* if there is an algorithm solving it in time $f(k) \cdot n^c$, where n denotes the size of the input, k the parameter, and $f : \mathbb{N} \to \mathbb{N}$ is a computable function and $c > 0$ a constant.

Parameterized complexity theory provides methods for proving problems to be fixed-parameter tractable, but also gives a framework for dealing with apparently intractable problems in a similar way that the theory of NP-completeness does in classical complexity theory.

The purpose of this article is to establish a very fruitful connection between parameterized complexity theory and logic. Our approach is that of descriptive complexity theory. We study the definability of parameterized problems and

[*] Institut für Mathematische Logik, Eckerstr. 1, 79104 Freiburg, Germany. Email: flum@sun2.ruf.uni-freiburg.de

[†] Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 S. Morgan St. (M/C 249), Chicago, IL 60607-7045, USA. Email: grohe@uic.edu

try to obtain information about the parameterized complexity of the problems through the syntactical structure of the defining sentences. On the one hand, we use this approach to prove that certain problems are tractable because they can be defined by syntactically simple formulas. On the other hand, we characterize classes of intractable problems by syntactical means.

Central to our approach are *parameterized model-checking* problems of the following form. For a class Φ of formulas, we let $MC(\Phi)$ be the problem

$MC(\Phi)$		
	Input:	A finite structure $\mathcal{A}$.
	Parameter:	A sentence $\varphi \in \Phi$.
	Question:	Does $\mathcal{A}$ satisfy φ?

In most cases, Φ will be a fragment of first-order logic.

After a preliminary section, we discuss some basic facts about parameterized model-checking problems in Section 3. In Section 4 we introduce two notions of definability of parameterized problems, which we call slicewise definability and Fagin definability, and relate them to model-checking. We then show how Fagin definability can be used to establish the fixed-parameter tractability of various problems.

In Section 5 we study the parameterized complexity of the model-checking problem for Σ_1-formulas (that is, existential first-order formulas in prenex normal form). We associate a graph with each such formula and use it to establish a surprisingly close connection between this model-checking problem, the homomorphism problem, and the subgraph isomorphism problem. As an application of our result we show that for Σ_1-sentences whose graph has bounded tree-width the model-checking problem is fixed-parameter tractable, even if inequalities are disregarded in the graph of the formula. Model-checking for formulas with a tree-like graph or hypergraph has recently received much attention (see [6, 25, 18, 16]).

So far we have only looked for tractable cases of the model-checking problem $MC(\Phi)$ that are obtained by restricting the class of formulas Φ. A different approach is to restrict the class of structures where the input structure $\mathcal{A}$ is taken from (see, for example, [7, 29, 17]). We prove a far reaching result: For any class C of graphs with an excluded minor, the model-checking problem for first-order logic is fixed-parameter tractable if the inputs are taken from C. This implies, for example, that parameterized versions of the dominating set problem or the (induced) subgraph isomorphism problem are fixed-parameter tractable when restricted to such classes of graphs.

Our last result on fixed-parameter tractability is a descriptive characterization of the complexity class FPT of all fixed-parameter tractable problems in terms of slicewise definability in finite variable fragments of least-fixed point logic. This simple result can be seen as a parameterized analogue of the well-known Immerman-Vardi Theorem [21, 30] characterizing the class PTIME in terms of definability in least-fixed-point logic.

The final section is devoted to fixed-parameter intractability. We define a hierarchy $\mathrm{A}[t]$ of parameterized complexity classes in terms of alternating Turing machine acceptance (t is the number of alternations). This hierarchy can be seen as a parameterized analogue of the polynomial hierarchy. We prove that for all $t \geq 1$, the model-checking problem for Σ_t-formulas is complete for the tth level of this hierarchy. Then we study the relation between our A-hierarchy and Downey and Fellows' W-hierarchy. It is known that the first levels of the respective hierarchies, $\mathrm{A}[1]$ and $\mathrm{W}[1]$ coincide [4]. We slightly improve a result of Downey, Fellows, and Regan [11] relating $\mathrm{W}[t]$, the tth level of the W-hierarchy, to the model-checking problems for a certain fragment of Σ_t. However, the questions whether $\mathrm{A}[t]$ and $\mathrm{W}[t]$ coincide for $t \geq 2$ remains open.

2 Preliminaries

2.1 Logic We assume that the reader is familiar with first-order logic; we just recall a few basic notions to fix our notation (compare [13] for a more detailed introduction of these notions).

In this article, a vocabulary *is a finite set of relation symbols.* Associated with every relation symbol is a natural number, its *arity*. The *arity* of a vocabulary is the maximal arity of the relation symbols it contains. Usually, vocabularies are also permitted to contain function and constant symbols. All results of this article, with the single exception of Theorem 33, would remain true if function and constant symbols were allowed, but adding them would not give us any new insights. So, for convenience, we restrict our attention to relational vocabularies. In the following, τ always denotes a vocabulary.

A τ-*structure* $\mathcal{A}$ consists of a set A, called the *universe* of $\mathcal{A}$, and a relation $R^{\mathcal{A}} \subseteq A^r$ for each r-ary relation symbol $R \in \tau$. We synonymously write $\bar{a} \in R^{\mathcal{A}}$ or $R^{\mathcal{A}}\bar{a}$ to denote that the tuple $\bar{a} \in A^r$ belongs to the relation $R^{\mathcal{A}}$. For $\tau \subseteq \tau'$, a τ-structure $\mathcal{A}$ is the τ-*reduct* of a τ'-structure $\mathcal{A}'$ if $A = A'$ and $R^{\mathcal{A}} = R^{\mathcal{A}'}$ for all $R \in \tau$. A τ'-structure $\mathcal{A}'$ is a τ'-*expansion* of a τ-structure $\mathcal{A}$ if $\mathcal{A}$ is the τ-*reduct* of $\mathcal{A}'$.

We only consider finite structures. When we consider classes of structures, they are always assumed to be closed under isomorphism. *STR* denotes the class of all (finite) structures. If C is a class of structures, $C[\tau]$ denotes the subclass of all τ-structures in C. Furthermore, $C[s]$ denotes the class of all structures in C whose vocabulary is at most s-ary. We consider graphs as $\{E\}$-structures $\mathcal{G} = (G, E^{\mathcal{G}})$, where $E^{\mathcal{G}}$ is an irreflexive and symmetric binary relation (i.e. graphs are loop-free and undirected). *GRAPH* denotes the class of all graphs.

The class of all first-order formulas is denoted by FO. Recall that *atomic formulas* are formulas of the form $x = y$ or $Rx_1 \dots x_r$, where $x, y, x_1, \dots, x_r$ are variables and R is a r-ary relation symbol. *Literals* are atomic or negated atomic formulas. A first-order formula φ is in *negation normal form* if negation symbols only occur directly in front of atomic subformulas. φ is *existential* (*universal*) if it is in negation normal form and contains no universal quantifiers (no existential quantifiers, respectively). φ is in *prenex normal form* if it is of the form $Q_1 x_1 \dots Q_k x_k \theta$, where $Q_1, \dots, Q_k \in \{\exists, \forall\}$ and θ is quantifier-free.

EFO (AFO) denotes the class of all existential (universal, respectively) first-order formulas. For $t \geq 1$, Σ_t denotes the class of all FO-formulas of the form

$$\exists x_{11} \dots \exists x_{1k_1} \forall x_{21} \dots \forall x_{2k_2} \ \dots \ Q x_{t1} \dots Q x_{tk_t}\ \theta,$$

where $Q = \forall$ if t is even and $Q = \exists$ otherwise and θ is quantifier-free. Π_t-formulas are defined analogously starting with a block of universal quantifiers.

If Φ is a class of formulas of some logic, then $\Phi[\tau]$ denotes the class of all formulas of vocabulary τ in L, and $\Phi[s]$ denotes the class of all formulas in Φ whose vocabulary is at most s-ary. We write $\mathcal{A} \models \varphi$ if, for some τ, $\mathcal{A}$ is a τ-structure, φ is in $\mathrm{L}[\tau]$, and $\mathcal{A}$ is a model of φ.

2.2 Coding issues We use random access machines (RAMs) with the uniform cost measure as our underlying model of computation (cf. [1]).

Very often, the objects of our computations are structures. Therefore, we have to fix a way of representing structures on a RAM. The two most common ways of doing this are the *array representation* and the *list representation*. For both representations we assume that the universes of our structures are initial segments of the natural numbers; of course this is no real restriction because every structure is isomorphic to one with such a universe.

Both representations start with an encoding of the vocabulary and a natural number representing the size of the universe of the structure. The difference between the two representations is in how relations are stored. In the array representation, a k-ary relation is stored as a k-dimensional array with $0, 1$-entries. For graphs, this is just the adjacency matrix. The advantage of this representation is that for each tuple it can be checked in constant time whether it belongs to the relation or not. However, for sparse relations this representation wastes a lot of space.

In the more concise list representation, a relation is represented as a list of all tuples it contains. Clearly, the list representation of a structure can be computed from the array representation in linear time, but not vice versa. For graphs $\mathcal{G}$, it is easy to construct the common adjacency list representation from the list representation (in time linear in $|G| +$ (size of the representation), where $|G|$ denotes the number of elements in G). In this article, we always assume that structures are given in the list representation, but all results also hold for the array representation. The *size* of a structure $\mathcal{A}$, denoted by $||\mathcal{A}||$, is defined to be $|A| +$ (size of the list representation of $\mathcal{A}$). The complexity of algorithms on structures is measured in this size. Remark 10 shows that this can be relevant.

2.3 Parameterized problems We only recall those notions of the theory needed in this article. For a comprehensive treatment we refer the reader to Downey and Fellow's recent monograph [10]. A *parameterized problem* is a set $P \subseteq \Sigma^* \times \Pi^*$, where Σ and Π are finite alphabets. Following [10], we usually represent a parameterized problem P in the following form:

P		
	Input:	$x \in \Sigma^*$.
	Parameter:	$y \in \Pi^*$.
	Question:	Is $(x, y) \in P$?

In most cases, we have $\Pi = \{0,1\}$ and consider the parameters $y \in \Pi^*$ as natural numbers (in binary). A natural example is the parameterized version of the well-known *VERTEX COVER* problem:

VC		
	Input:	Graph $\mathcal{G}$.
	Parameter:	$k \in \mathbb{N}$.
	Question:	Does $\mathcal{G}$ have a vertex cover of size k?

Recall that a *vertex cover* of a graph is a set X of vertices such that every edge is incident to one of the vertices in X. Similarly, we can define parameterized versions of *DOMINATING SET (DS)* and *CLIQUE*. (A *dominating set* of a graph is a set X of vertices such that every vertex not contained in X is adjacent to a vertex in X. A *clique* is a set of pairwise adjacent vertices.)

An example where the set of parameters is not $\mathbb{N}$, but the class *GRAPH* of all finite graphs is the following parameterized *SUBGRAPH ISOMORPHISM* problem:

SI		
	Input:	Graph $\mathcal{G}$.
	Parameter:	Graph $\mathcal{H}$.
	Question:	Is $\mathcal{H}$ isomorphic to a subgraph of $\mathcal{G}$?

Similarly, we can define parameterized versions of the *INDUCED SUBGRAPH ISOMORPHISM* problem and the *GRAPH HOMOMORPHISM* problem.

Definition 1 A parameterized problem $P \subseteq \Sigma^* \times \Pi^*$ is *fixed-parameter tractable* if there is a computable function $f : \Pi^* \to \mathbb{N}$, a constant $c \in \mathbb{N}$, and an algorithm that, given a pair $(x, y) \in \Sigma^* \times \Pi^*$, decides if $(x, y) \in P$ in time $f(|y|) \cdot |x|^c$.[1]

We denote the class of all fixed-parameter tractable problems by FPT.

Of course we can always consider parameterized problems as classical problems and determine their complexity in the classical sense. Clearly, every parameterized problem in PTIME is also in FPT.

The best currently known algorithm for vertex cover *VC* has running time $O(k \cdot n + \max\{1.255^k \cdot k^2, 1.291^k \cdot k\})$ [15], where n denotes the size of the input graph. Thus *VC* $\in$ FPT.

2.4 Reductions between parameterized problems It is conjectured that none of the problems *DS*, *CLIQUE*, *SI* is in FPT. As it is often the case in complexity theory, we can not actually prove this, but only prove that the problems are hard for certain complexity classes that are conjectured to contain FPT strictly. To do this we need a suitable concept of reduction. We actually introduce three different types of reduction:

Definition 2 Let $P \subseteq \Sigma^* \times \Pi^*$ and $P' \subseteq (\Sigma')^* \times (\Pi')^*$ be parameterized problems.

(1) A *parameterized T-reduction* from P to P' is an algorithm with an oracle for P' that solves any instance (x, y) of P in time $f(|y|) \cdot |x|^c$ in such a way that for all questions $(x', y') \in P'$? to the oracle we have $|y'| \leq g(|y|)$ (for computable functions $f, g : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$).

P is *fixed-parameter T-reducible* to P' (we write $P \leq_{\mathrm{T}}^{\mathrm{fp}} P'$), if there is a parameterized T-reduction from P to P'.

(2) A *parameterized m-reduction* from P to P' is an algorithm that computes for every instance (x, y) of P an instance (x', y') of P' in time $f(|y|) \cdot |x|^c$ such that $|y'| \leq g(|y|)$ and

$$(x, y) \in P \iff (x', y') \in P'$$

(for computable functions $f, g : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$).

P is *fixed-parameter m-reducible* to P' (we write $P \leq_{\mathrm{m}}^{\mathrm{fp}} P'$), if there is a parameterized m-reduction from P to P'.

[1] This is what Downey and Fellows call *strongly uniformly fixed-parameter tractable*. For variants of this definition, and also of Definition 2 and the definition of the W-hierarchy, the reader should consult [10].

Whereas every parameterized problem that is in PTIME (when considered as a classical problem) is in FPT, it is not the case that every PTIME many-one reduction between two parameterized problems is also a parameterized m-reduction. To capture both concepts we occasionally use the following third kind of reduction:

Definition 3 Let $P \subseteq \Sigma^* \times \Pi^*$ and $P' \subseteq (\Sigma')^* \times (\Pi')^*$ be parameterized problems.

A *pp m-reduction* from P to P' is a parameterized m-reduction from P to P' that is also a polynomial time many-one reduction from P to P' in the classical sense, i.e. the function f in Definition 2(2) is a polynomial.

P is *pp m-reducible* to P' (we write $P \leq_{\mathrm{m}}^{\mathrm{fpp}} P'$), if there is a pp m-reduction from P to P'.

For example, *CLIQUE* $\leq_{\mathrm{m}}^{\mathrm{fp}}$ *SI* by the simple parameterized m-reduction that reduces the instance $(\mathcal{G}, k)$ of *CLIQUE* to the instance $(\mathcal{G}, \mathcal{K}_k)$ of *SI*. Here $\mathcal{K}_k$ denotes the complete graph with k vertices. Note that if we represent integers in binary, this reduction is not a pp m-reduction.

Observe that $\leq_{\mathrm{T}}^{\mathrm{fp}}$, $\leq_{\mathrm{m}}^{\mathrm{fp}}$, and $\leq_{\mathrm{m}}^{\mathrm{fpp}}$ are transitive and that for all P, P' we have

$$P \leq_{\mathrm{m}}^{\mathrm{fpp}} P' \implies P \leq_{\mathrm{m}}^{\mathrm{fp}} P' \quad \text{and} \quad P \leq_{\mathrm{m}}^{\mathrm{fp}} P' \implies P \leq_{\mathrm{T}}^{\mathrm{fp}} P'.$$

Furthermore, if $P \leq_{\mathrm{T}}^{\mathrm{fp}} P'$ and $P' \in$ FPT then $P \in$ FPT. For any of the reductions $\leq_{\mathrm{T}}^{\mathrm{fp}}, \leq_{\mathrm{m}}^{\mathrm{fp}}, \leq_{\mathrm{m}}^{\mathrm{fpp}}$ we let $\equiv_{\cdots}^{\cdots}$ denote the corresponding equivalence relation.

We define *hardness* and *completeness* of parameterized problems for a parameterized complexity class (under parameterized m- or T-reductions) in the usual way. For a parameterized problem P, we let $[P]_{\mathrm{m}}^{\mathrm{fp}} := \{P' \mid P' \leq_{\mathrm{m}}^{\mathrm{fp}} P\}$, and for a class P of parameterized problems $[\mathrm{P}]_{\mathrm{m}}^{\mathrm{fp}} := \bigcup_{P \in \mathrm{P}} [P]_{\mathrm{m}}^{\mathrm{fp}}$.

Remark 4 Very often, it is natural to think of a parameterized problem P as derived from a (classical) problem $L \subseteq \Sigma^*$ by a *parameterization* $p : \Sigma^* \to \mathbb{N}$ in such a way that $P = \{(x, k) \mid x \in L, k = p(x)\}$.

Slightly abusing notation, we represent such a P in the form

P		
	Input:	$x \in \Sigma^*$.
	Parameter:	$p(x)$.
	Question:	Decide if $x \in L$?

As an example, let us reconsider the subgraph isomorphism problem. Instead of taking graph $\mathcal{H}$ as the parameter, we may also consider pairs of graphs $(\mathcal{G}, \mathcal{H})$ as inputs and parameterize the problem by the size of $\mathcal{H}$. In our new notation, this would be the problem

SI'		
	Input:	Graphs $\mathcal{G}$, $\mathcal{H}$.
	Parameter:	$\|\|\mathcal{H}\|\|$.
	Question:	Is $\mathcal{H}$ isomorphic to a subgraph of $\mathcal{G}$?

It is easy to see, however, that *SI* $\equiv_{\mathrm{m}}^{\mathrm{fp}}$ *SI'*.

2.5 Parameterized intractability Some combinatorial problems are provably not fixed-parameter tractable, and others, such as *GRAPH COLORABILITY*, are not fixed-parameter tractable unless PTIME = NP. However, many interesting problems, such as the parameterized *CLIQUE* problem, do not seem to be fixed-parameter tractable, although there is no known way to prove this or reduce it to classical complexity theoretic questions such as PTIME $\stackrel{?}{=}$ NP. To classify such problems, Downey and Fellows (cf. [10]) introduced a hierarchy

$$\mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots$$

of classes above FPT. These classes can best be defined in terms of the satisfiability problem for formulas of propositional logic. Formulas of propositional logic are build up from *propositional variables* $X_1, X_2, \ldots$ by taking conjunctions, disjunctions, and negations. The negation of a formula φ is denoted by $\neg\varphi$. We distinguish between *small conjunctions*, denoted by $\wedge$, which are just conjunctions of two formulas, and *big conjunctions*, denoted by $\bigwedge$, which

are conjunctions over arbitrary finite sets of formulas. Analogously, we distinguish between *small disjunctions*, denoted by $\vee$, and *big disjunctions*, denoted by $\bigvee$.

Every formula φ corresponds to a labeled tree $\mathcal{T}_\varphi$ in a natural way. The *size* of φ is defined to be the number of vertices of $\mathcal{T}_\varphi$. The *depth* of φ is defined to be the maximum number of nodes labeled $\wedge, \bigwedge, \vee, \bigvee$ on a path from the root to a leaf of $\mathcal{T}_\varphi$. Thus when computing the depth, we do not count negations.

A formula is *small* if it only contains small conjunctions and small disjunction. We define $C_0 = D_0$ to be the class of all small formulas. For an $i \geq 1$, we define C_i to be the class of all big conjunctions of formulas in D_{i-1}, and we define D_i to be the class of all big disjunctions of formulas in C_{i-1}. Note that these definitions are purely syntactical; every formula formula in a C_i or D_i is equivalent to a formula in C_0. But of course the translation from a formula in C_i to an equivalent formula in C_0 usually increases the depth of a formula. For all $i, d \geq 0$ we let $C_{i,d}$ denote the class of all formulas in C_i whose small subformulas have depth at most d (equivalently, we may say that the whole formula has depth at most $d + i$). We define $D_{i,d}$ analogously.

The *weight* of an assignment α for the variables of a propositional formula is the number of variables set to TRUE by α. For any class P of propositional formulas, let *weighted satisfiability for* P be the following parameterized problem:

WSAT(P)		
	Input:	$\varphi \in P$.
	Parameter:	$k \in \mathbb{N}$.
	Question:	Does φ have a satisfying assignment of weight k?

Now we are ready to define the *W-hierarchy*: For every $t \geq 1$, we let

$$\mathrm{W}[t] := \bigcup_{d \geq 0} [\mathit{WSAT}(C_{t,d})]^{\mathrm{fp}}_{\mathrm{m}}.$$

In other words, a parameterized problem is in $\mathrm{W}[t]$ if there is a $d \geq 0$ such that the problem is fixed-parameter m-reducible to the weighted satisfiability problem for $C_{t,d}$. It is an immediate consequence of the definition of parameterized m-reductions that FPT $\subseteq \mathrm{W}[1]$. Actually, it is conjectured that this inclusion is strict and that $\mathrm{W}[t]$ is strictly contained in $\mathrm{W}[t+1]$ for every $t \geq 1$.

Example 5 The parameterized *CLIQUE*-problem is in $\mathrm{W}[1]$. To see this, for every graph $\mathcal{G}$ we describe a propositional formula $\varphi := \varphi(\mathcal{G}) \in C_{1,1}$ such that $\mathcal{G}$ has a clique of size k if, and only if, φ has a satisfying assignment of weight k. It will be obvious from the construction that φ can be computed from $\mathcal{G}$ in polynomial time.

So let $\mathcal{G}$ be a graph. For all $a \in G$ let X_a be a propositional variable. Let

$$\varphi := \bigwedge_{\substack{a,b \in G, a \neq b \\ ab \notin E^G}} (\neg X_a \vee \neg X_b)$$

Then every satisfying assignment of φ corresponds to a clique of $\mathcal{G}$.

Actually, Downey and Fellows proved the following non-trivial result:

Theorem 6 (Downey and Fellows [8, 9]) *(1) CLIQUE is* $\mathrm{W}[1]$*-complete under parameterized m-reductions.*

(2) DS is $\mathrm{W}[2]$*-complete under parameterized m-reductions.*

Remark 7 Downey and Fellows phrase their definition of the W-hierarchy in terms of Boolean circuits rather than propositional formulas. But since the classes of the hierarchy only involve circuits/formulas of bounded depth, this does not really make a difference (cf. [10]). In their definition of $\mathrm{W}[t]$, Downey and Fellows admit more complicated formulas than those in C_t. But they prove that our definition is equivalent. A surprising by-product of their results is that for every $t \geq 1$ and every $d \geq 0$, the problem *WSAT*$[D_{t+1,d}]$ is contained in $\mathrm{W}[t]$. It is not hard to prove this result directly, and even easier to prove that *WSAT*$[D_{1,d}]$ is in FPT. (This explains why we only defined a hierarchy using the C_ts).

There is another, more serious source of confusion in the various definitions of the W-hierarchy: Downey and Fellows are never really clear about what kind of reductions they are using to define the classes. We decided, more or less in accordance with [10], that parameterized m-reductions are most natural.

We will further discuss the W-hierarchy and other seemingly intractable classes in Section 8.

3 Model-checking

In this article we are mainly concerned with the complexity of various *parameterized model-checking problems*. For a set Φ of formulas, we let

$$MC(\Phi) := \big\{(\mathcal{A}, \varphi) \;\big|\; \mathcal{A} \in \mathit{STR}, \varphi \text{ sentence in } \Phi, \mathcal{A} \models \varphi\big\},$$

or more intuitively,

$MC(\Phi)$		
	Input:	$\mathcal{A} \in \mathit{STR}$.
	Parameter:	$\varphi \in \Phi$.
	Question:	Does $\mathcal{A} \models \varphi$?

In this section we collect a few basic facts about parameterized model-checking problems. For every Φ we consider, we assume that we have fixed an encoding $\gamma : \Phi \to \{0,1\}^*$, and we let $||\varphi||$ be the length of $\gamma(\varphi)$.

Taking sentences as parameters seems a little unusual. The following parameterization of the model checking problem looks more natural:

$MC'(\Phi)$		
	Input:	$\mathcal{A} \in \mathit{STR}$, $\varphi \in \Phi$.
	Parameter:	$\|\|\varphi\|\|$.
	Question:	Does $\mathcal{A} \models \varphi$?

However, it is easy to see that $MC(\Phi) \equiv^{\text{fp}}_{\text{m}} MC'(\Phi)$.

It is well-known that various problems of model theory or complexity theory can be reduced from structures to graphs. The following two lemmas contain such reductions. Although their proofs only use standard techniques, they are subtle and require some care. Therefore we decided to give the proofs in some detail. We will apply these lemmas several times later.

Lemma 8 *There are polynomial time transformations that associate with every structure $\mathcal{A} \in \mathit{STR}$ a graph $\mathcal{H}(\mathcal{A})$ and with every sentence $\varphi \in$ FO a sentence $\varphi_{\mathit{GRAPH}} \in$ FO, respectively, such that*

$$\mathcal{A} \models \varphi \iff \mathcal{H}(\mathcal{A}) \models \varphi_{\mathit{GRAPH}}.$$

Furthermore for every $t \geq 1$, if $\varphi \in \Sigma_t$ then $\varphi_{\mathit{GRAPH}} \in \Sigma_{t+1}$ and if $\varphi \in \Pi_t$ then $\varphi_{\mathit{GRAPH}} \in \Pi_{t+1}$.

Proof: Let $\mathcal{A} \in \mathit{STR}[\tau]$ and $\varphi \in \text{FO}[\tau]$. Without loss of generality we can assume that φ is in prenex and in negation normal form.

Step 1. In the first step we translate $\mathcal{A}$ to a structure $\mathcal{B}(\mathcal{A})$ of a vocabulary $\beta(\tau)$ that only consists of unary and binary relation symbols. φ is translated to a corresponding sentence φ_B of vocabulary $\beta(\tau)$.

$\beta(\tau)$ contains unary relation symbols U and U_R for each symbol $R \in \tau$ and binary relation symbols $E_1, \ldots, E_s$, where s is the arity of τ.

The universe of $\mathcal{B}(\mathcal{A})$ is

$$B(\mathcal{A}) := A \cup \{b(R, \bar{a}) \mid R \in \tau, \bar{a} \in R^{\mathcal{A}}\}.$$

We assume that the elements $b(R,\bar{a})$ are all pairwise distinct and distinct from those in A. Note that the cardinality of $B(\mathcal{A})$ is essentially $||\mathcal{A}||$, up to an additive term depending on τ. The unary relations are defined in the obvious way: We let $U^{\mathcal{B}(\mathcal{A})} := A$ and $U_R^{\mathcal{B}(\mathcal{A})} := \{b(R,\bar{a}) \mid \bar{a} \in R^{\mathcal{A}}\}$ for every $R \in \tau$. The binary relations $E_1, \dots, E_s$ are defined by

$$E_i^{\mathcal{B}(\mathcal{A})} := \{(a_i, b(R,\bar{a})), (b(R,\bar{a}), a_i) \mid R \in \tau, \bar{a} = (a_1, \dots, a_r) \in R^{\mathcal{A}}, 1 \leq i \leq r\}.$$

Note that $E_i^{\mathcal{B}(\mathcal{A})}$ is symmetric, this will be useful later.

To define φ_B, we first relativize all quantifiers to U, i.e. we inductively replace all subformulas $\exists x\psi$ by $\exists x(Ux \wedge \psi)$ and all subformulas $\forall x\psi$ by $\forall x(Ux \to \psi)$. We obtain a formula φ'.

φ_B is obtained from φ' by replacing every atomic subformula $R\bar{x}$, for r-ary $R \in \tau$, by

$$\exists z(U_R z \wedge \bigwedge_{i=1}^{r} E_i x_i z) \tag{1}$$

where z is a new variable. Then we have

$$\mathcal{A} \models \varphi \iff \mathcal{B}(\mathcal{A}) \models \varphi_B. \tag{2}$$

Furthermore, $\mathcal{B}(\mathcal{A})$ can be computed from $\mathcal{A}$ in time $O(||\tau|| \cdot ||\mathcal{A}||)$, where $||\tau||$ denotes the length of the encoding of τ, and φ_B can be computed from φ in linear time.

Step 2. In this step we replace the binary relations $E_1, \dots, E_s$ by a single new binary relation E. We let

$$\gamma(\tau) := (\beta(\tau) \setminus \{E_1, \dots, E_s\}) \cup \{E, P_1, \dots, P_s\},$$

where $P_1, \dots, P_s$ are new unary relation symbols.

We transform $\mathcal{B}(\mathcal{A})$ to a $\gamma(\tau)$-structure $\mathcal{C}(\mathcal{A})$ as follows: For $1 \leq i \leq s$ and for all $a, b \in B(\mathcal{A})$ such that $E_i^{\mathcal{B}(\mathcal{A})}ab$ we introduce a new element $c(i,a,b)$, add the pairs $(a, c(i,a,b))$, $(c(i,a,b), a)$, $(c(i,a,b), b)$, $(b, c(i,a,b))$ to $E^{\mathcal{C}(\mathcal{A})}$ and add $c(i,a,b)$ to $P_i^{\mathcal{C}(\mathcal{A})}$.

We define a sentence φ_C by replacing each subformula of φ_B of the form $E_i xy$ by $\exists z(Exz \wedge Ezy \wedge P_i z)$. Then we have the analogue of (2) and the subsequent remarks for $\mathcal{C}(\mathcal{A}), \varphi_C$ instead of $\mathcal{B}(\mathcal{A}), \varphi_B$.

Step 3. The restriction of $\mathcal{C}(\mathcal{A})$ to E is already a graph, i.e. $E^{\mathcal{C}(\mathcal{A})}$ is symmetric and irreflexive, so all we have to do is to eliminate the unary relations $U, (U_R)_{R\in\tau}, (P_i)_{1\leq i\leq s}$. Say, $Q_1, \dots, Q_l$ is an enumeration of all these relations. Note that every $a \in C(\mathcal{A})$ is either isolated or of valence two or adjacent to a vertex of valence two. We use this to define certain trees $\mathcal{T}_1, \dots, \mathcal{T}_l$ and corresponding existential first-order formulas $\xi_1(x), \dots, \xi_l(x)$ and attach a copy of $\mathcal{T}_i$ to each vertex in $Q_i^{\mathcal{C}(\mathcal{A})}$ in such a way that in the resulting graph $\mathcal{H}(\mathcal{A})$ we have for all vertices a:

$$\mathcal{H}(\mathcal{A}) \models \xi_i(a) \iff a \in Q_i^{\mathcal{C}(\mathcal{A})}.$$

Furthermore, the $\mathcal{T}_i$ and thus the ξ_i can be chosen of size polynomial in l. We omit the details.

Then we let φ_{GRAPH} be the formula obtained from φ_C by replacing every subformula of the form $Q_i x$ by $\xi_i(x)$, for $1 \leq i \leq l$, and transforming the resulting formula into prenex normal form in the usual manner.

Clearly the transformations $\mathcal{A} \mapsto \mathcal{H}(\mathcal{A})$ and $\varphi \mapsto \varphi_{GRAPH}$ are polynomial, and we have

$$\mathcal{A} \models \varphi \iff \mathcal{H}(\mathcal{A}) \models \varphi_{GRAPH}.$$

It remains to prove that if $\varphi \in \Sigma_t$ ($\varphi \in \Pi_t$) then $\varphi_{GRAPH} \in \Sigma_{t+1}$ ($\varphi_{GRAPH} \in \Pi_{t+1}$, respectively). This follows easily from the way we defined the sentences φ_B, φ_C, and φ_{GRAPH} in Steps 1–3, noting that each positive (negative) occurrence of a relation symbol $R \in \tau$ only gives rise to positive (negative, respectively) occurrences of U_R, the E_i, and the P_i. Thus for the positive occurrences of the $R \in \tau$ we get a new block of existential quantifiers and for the negative occurrences a new block of universal quantifiers. This increases the alternation depth by at most one. □

Recall that for a class C of structures and an $s \geq 0$, by $C[s]$ we denote the class of all structures in C whose vocabulary is at most s-ary. Similarly, for a class Φ of formulas, by $\Phi[s]$ we denote the class of all formulas in Φ whose vocabulary is at most s-ary.

Lemma 9 *Let $s \geq 1$. There is a polynomial time transformation that associates with every structure $\mathcal{A} \in \mathit{STR}[s]$ a graph $\mathcal{H}'(\mathcal{A})$ and a linear time function that associates with every sentence $\varphi \in \mathrm{FO}$ a sentence $\varphi'_{\mathit{GRAPH}} \in \mathrm{FO}$, such that*

$$\mathcal{A} \models \varphi \iff \mathcal{H}'(\mathcal{A}) \models \varphi'_{\mathit{GRAPH}}.$$

Furthermore, for all $t \geq 1$, if $\varphi \in \Sigma_t$ then $\varphi'_{\mathit{GRAPH}} \in \Sigma_t$ and if $\varphi \in \Pi_t$ then $\varphi'_{\mathit{GRAPH}} \in \Pi_t$.

Proof: Let us first look back at the proof of the last lemma and note that if all relation symbols only occur positively in φ then the transformation $\varphi \mapsto \varphi_{\mathit{GRAPH}}$ only generates a new block of existential quantifiers. If all relation symbols only occur negatively then we only get a new block of universal quantifiers. Thus if $\varphi \in \Sigma_{2t-1}$ ($\varphi \in \Pi_{2t}$) and all relation symbols only occur positively in φ then also $\varphi_{\mathit{GRAPH}} \in \Sigma_{2t-1}$ ($\varphi_{\mathit{GRAPH}} \in \Pi_{2t}$, respectively). Similarly, if $\varphi \in \Sigma_{2t}$ ($\varphi \in \Pi_{2t-1}$) and all relation symbols only occur negatively in φ then also $\varphi_{\mathit{GRAPH}} \in \Sigma_{2t}$ ($\varphi_{\mathit{GRAPH}} \in \Pi_{2t-1}$, respectively).

We first transform $\mathcal{A}$ to an $\mathcal{A}'$ and φ to a φ' in the same prefix class such that

$$\mathcal{A} \models \varphi \iff \mathcal{A}' \models \varphi',$$

and either all relation symbols in φ' occur positively or negatively, whichever we need to apply the previous remark. For this purpose, let τ be an at most s-ary vocabulary. We let $\tau' := \tau \cup \{\overline{R} \mid R \in \tau\}$, where $\overline{R}$ is a new relation symbol that has the same arity as R. For every $\mathcal{A} \in \mathit{STR}[\tau]$, we let $\mathcal{A}'$ be the τ'-expansion of $\mathcal{A}$ with $\overline{R}^{\mathcal{A}'} = A^r \setminus R^{\mathcal{A}}$ for r-ary $R \in \tau$. Note that $\mathcal{A}'$ can be computed from $\mathcal{A}$ in time $O(||\mathcal{A}||^s)$. To define φ' we either replace each negative literal $\neg R\bar{x}$ by $\overline{R}\bar{x}$ or each positive literal $R\bar{x}$ by $\neg\overline{R}\bar{x}$. □

Remark 10 Note that if we represent structures by the array representation, then the transformation of Lemma 9 is actually polynomial even if we do not fix the arity of the vocabulary in advance. This follows from the fact that the array representation of the structure $\mathcal{A}'$ (in the proof of the lemma) can be computed from the array representation of $\mathcal{A}$ in linear time, uniformly over all vocabularies.

For a parameterized problem $P \subseteq \mathit{STR} \times \Pi^*$ and a class C of structures we let $P|_C$ denote the *restriction* of P to C. In particular,

$$\mathit{MC}(\Phi)|_{\mathit{GRAPH}} = \{(\mathcal{G}, \varphi) \mid \mathcal{G} \in \mathit{GRAPH}, \varphi \in \Phi, \mathcal{G} \models \varphi\}.$$

Note that for every class Φ of formulas and vocabulary τ the two problems $\mathit{MC}(\Phi[\tau])$ and $\mathit{MC}(\Phi)|_{\mathit{STR}[\tau]}$, though formally different, are essentially the same. Therefore we do not distinguish between them.

Corollary 11 *(1)* $\mathit{MC}(\mathrm{FO}) \equiv^{\mathrm{fpp}}_{\mathrm{m}} \mathit{MC}(\mathrm{FO})|_{\mathit{GRAPH}}$.

(2) For all $t \geq 1$ we have

$$\mathit{MC}(\Sigma_t) \leq^{\mathrm{fpp}}_{\mathrm{m}} \mathit{MC}(\Sigma_{t+1})|_{\mathit{GRAPH}} \quad \textit{and} \quad \mathit{MC}(\Pi_t) \leq^{\mathrm{fpp}}_{\mathrm{m}} \mathit{MC}(\Pi_{t+1})|_{\mathit{GRAPH}}.$$

(3) For all $s \geq 2$, $t \geq 1$ we have

$$\mathit{MC}(\Sigma_t[s]) \equiv^{\mathrm{fpp}}_{\mathrm{m}} \mathit{MC}(\Sigma_t)|_{\mathit{GRAPH}} \quad \textit{and} \quad \mathit{MC}(\Pi_t[s]) \equiv^{\mathrm{fpp}}_{\mathrm{m}} \mathit{MC}(\Pi_t)|_{\mathit{GRAPH}}.$$

We do not know whether $\mathit{MC}(\Sigma_t) \leq^{\mathrm{fp}}_{\mathrm{m}} \mathit{MC}(\Sigma_t)|_{\mathit{GRAPH}}$ and $\mathit{MC}(\Pi_t) \leq^{\mathrm{fp}}_{\mathrm{m}} \mathit{MC}(\Pi_t)|_{\mathit{GRAPH}}$ for $t \geq 1$.

4 Defining parameterized problems

Definability is the connection between arbitrary parameterized problems and our logical analysis that focuses on model-checking problems. In [11], Downey, Fellows, and Regan consider two forms of definability: Their exposition motivates two general notions of definability, which we call *slicewise definability* and *Fagin definability*.

For a parameterized problem $P \subseteq \Sigma^* \times \Pi^*$ and $y \in \Pi^*$, we call $P \cap (\Sigma^* \times \{y\})$ the yth *slice* of P.

Definition 12 Let $P \subseteq \mathit{STR} \times \Pi^*$ be a parameterized problem and Φ a class of formulas. P is *slicewise* Φ*-definable* if there is a computable function $\delta : \Pi^* \to \Phi$ such that for all $\mathcal{A} \in \mathit{STR}$ and $y \in \Pi^*$ we have $(\mathcal{A}, y) \in P \iff \mathcal{A} \models \delta(y)$.

For example, the parameterized subgraph isomorphism problem *SI* is slicewise Σ_1-definable via the function $\delta : \mathit{GRAPH} \to \Sigma_1$ defined as follows: For a graph $\mathcal{H}$ with vertex set $H = \{h_1, \dots, h_n\}$ of cardinality n, $\delta(\mathcal{H})$ is the sentence

$$\exists x_1 \dots \exists x_n \Big(\bigwedge_{1 \le i < j \le n} x_i \neq x_j \wedge \bigwedge_{\substack{1 \le i,j \le n \\ E^{\mathcal{H}} h_i h_j}} E x_i x_j \Big).$$

Slicewise Φ-definability is closely related to the model-checking problem for Φ: If $P \subseteq \mathit{STR} \times \Pi^*$ is slicewise Φ-definable, then $P \le^{\mathrm{fp}}_{\mathrm{m}} \mathit{MC}(\Phi)$. On the other hand, if Φ is a decidable set of formulas, then the problem $\mathit{MC}(\Phi)$ is slicewise Φ-definable for trivial reasons.

Definition 13 Let τ be a vocabulary, $P \subseteq \mathit{STR}[\tau] \times \mathbb{N}$ a parameterized problem, and Φ a class of formulas. P is Φ*-Fagin-definable* if there is a relation symbol $X \notin \tau$ (say, r-ary) and a sentence $\varphi \in \Phi[\tau \cup \{X\}]$ such that for all $\mathcal{A} \in \mathit{STR}[\tau]$ and $k \in \mathbb{N}$ we have $(\mathcal{A}, k) \in P$ if and only if there is a $B \subseteq A^r$ such that $|B| = k$ and $(\mathcal{A}, B) \models \varphi$. (Here $(\mathcal{A}, B)$ denotes the $\tau \cup \{X\}$-expansion of $\mathcal{A}$ that interprets X by B.) Then φ *Fagin-defines* P.

We often consider X as a relation *variable* and thus write $\varphi(X) \in \Phi[\tau]$ instead of $\varphi \in \Phi[\tau \cup \{X\}]$ and $\mathcal{A} \models \varphi(B)$ instead of $(\mathcal{A}, B) \models \varphi$.

For example, parameterized vertex cover *VC* is Fagin-defined by the formula

$$\varphi_{\mathit{VC}} := \forall y \forall z \big(E y z \to (X y \vee X z) \big).$$

It is easy to see that every problem that is FO-Fagin-definable is also FO-slicewise definable. Indeed, if $P \subseteq \mathit{STR}[\tau] \times \mathbb{N}$ is Fagin-defined by a formula $\varphi(X) \in \mathrm{FO}[\tau]$, where X is r-ary, then it is slicewise FO-defined via the function $\delta : \mathbb{N} \to \mathrm{FO}[\tau]$ with $\delta(k) = \exists \bar{x}_1 \dots \exists \bar{x}_k \big(\bigwedge_{1 \le i < j \le k} \bar{x}_i \neq \bar{x}_j \wedge \varphi_k \big)$, where $\bar{x}_1, \dots, \bar{x}_k$ are r-tuples of distinct variables not occurring in φ and φ_k is the sentence obtained from φ by replacing each subformula of the form $X \bar{y}$ by $\bigvee_{i=1}^k \bar{x}_i = \bar{y}$. Fagin-definability implies slicewise definability also for other reasonable classes Φ of formulas, e.g., for the class Σ^1_1 of formulas of second-order logic of the form $\exists X_1 \dots \exists X_l \psi$, where $X_1, \dots X_l$ are relation variables and ψ is first-order.

The converse is certainly not true, not even for problems of the specific form $P \subseteq \mathit{STR}[\tau] \times \mathbb{N}$: It is obvious that there are slicewise FO-definable problems of arbitrarily high classical complexity (choose δ in Definition 12 arbitrarily complex). On the other hand we have the following characterization of Σ^1_1-Fagin-definable problems.

Proposition 14 *Let $P \subseteq \mathit{STR}[\tau] \times \mathbb{N}$. Then, (1) and (2) are equivalent, where*

(1) P is Σ^1_1-Fagin-definable.

(2) P is in NP *(when considered as a classical problem) and for some $r \ge 1$, $(\mathcal{A}, k) \in P$ implies $k \le |A|^r$.*

Proof: The implication of (1) $\Rightarrow$ (2) being clear, we turn to a proof of (2) $\Rightarrow$ (1). Choose r according to (2). Then,

$$\{(\mathcal{A}, B) \mid B \subseteq A^r \text{ and } (\mathcal{A}, |B|) \in P\}$$

is a class of $\tau \cup \{X\}$-structures in NP, where X is r-ary. By Fagin's Theorem [14], there is a Σ^1_1-formula $\varphi(X)$ of vocabulary $\tau \cup \{X\}$ axiomatizing this class. Then, $\varphi(X)$ Σ^1_1-Fagin-defines P. □

Thus, slicewise definability is the more general notion. Nevertheless, Fagin definability can be very useful. We illustrate this by the following generalization of a result due to Cai and Chen, namely Theorem 3.5 of [3], which is based on a result due to Kolaitis and Thakur [24] that syntactically characterizes certain minimization problems. It is motivated by comparing the formula φ_{VC} defining the fixed-parameter tractable problem *VC* with the following formulas φ_{DS} and $\varphi_{\mathit{CLIQUE}}$ defining the W[1]-hard problems *DS* and *CLIQUE*, respectively:

$$\varphi_{\mathit{DS}} := \forall y \exists x (X x \wedge (x = y \vee E x y)),$$
$$\varphi_{\mathit{CLIQUE}} := \forall y \forall z \big((X y \wedge X z) \to (y = z \vee E y z) \big).$$

Observe that in φ_{DS} the relation variable X is in the scope of an existential quantifier and in φ_{CLIQUE} it occurs negatively.

Theorem 15 *Let τ be a vocabulary and $P \subseteq STR[\tau] \times \mathbb{N}$ a parameterized problem that is Fagin-defined by a* FO$[\tau]$*-formula $\varphi(X)$ in which X does not occur in the scope of an existential quantifier or negation symbol. Then P is in* FPT.

Proof: For simplicity, let us assume that X is unary. Without loss of generality we can assume that

$$\varphi = \forall y_1 \ldots \forall y_l \bigvee_{i=1}^{m} \bigwedge_{j=1}^{p} \psi_{ij},$$

where each ψ_{ij} either is Xy_q for some $q \in \{1, \ldots, l\}$, or a first-order formula with free variables in $\{y_1, \ldots, y_l\}$ in which X does not occur. In a preprocessing phase we replace the latter ones by atomic formulas: For each such ψ_{ij} we introduce a new relation symbol R_{ij} whose arity matches the number of free variables of ψ_{ij} and let τ^* be the set of all these relation symbols. We let φ^* be the formula obtained from φ by replacing each subformula $\psi_{ij}(\bar{z})$ by $R_{ij}\bar{z}$. Then $\varphi^* = \forall y_1 \ldots \forall y_l \bigvee_{i=1}^{m} \chi_i$, where each χ_i is a conjunction of atomic formulas.

For a structure $\mathcal{A} \in STR[\tau]$ we let $\mathcal{A}^*$ be the τ^*-structure with universe A and with

$$R_{ij}^{\mathcal{A}^*} := \{\bar{a} \mid \mathcal{A} \models \psi_{ij}(\bar{a})\}.$$

Then we have for $B \subseteq A$,

$$\mathcal{A} \models \varphi(B) \iff \mathcal{A}^* \models \varphi^*(B).$$

Given $\mathcal{A}$, each $R_{ij}^{\mathcal{A}^*}$ can be computed in time $O(|A|^{||\psi_{ij}||})$, thus $\mathcal{A}^*$ can certainly be computed in time $O(|A|^{||\varphi||})$.

For $1 \leq i \leq m$, $\bar{a} = a_1 \ldots a_l \in A^l$, and $B \subseteq A$ we let

$$\beta(B, \bar{a}, i) := B \cup \{a_j \mid Xy_j \text{ is a conjunct of } \chi_i\}.$$

Since $\chi_i(X, \bar{y})$ is positive in X, the following two statements are equivalent for every B' with $B \subseteq B' \subseteq A$:

- $\mathcal{A}^* \models \chi_i(B', \bar{a})$.
- $\beta(B, \bar{a}, i) \subseteq B'$ and $\mathcal{A}^* \models \chi_i(\beta(B, \bar{a}, i), \bar{a})$.

This equivalence is used by Algorithm 1 to decide P.

Recall that, given a τ-structure $\mathcal{A}$ and a parameter $k \in \mathbb{N}$, the algorithm is supposed to decide whether there is a $B \subseteq A$ with $|B| = k$ such that for all $\bar{a} \in A^l$ there is an i such that $\mathcal{A}^* \models \chi_i(B, \bar{a})$.

The crucial observation to see that the algorithm is correct is that whenever the main loop in Lines 3–10 is entered, $\mathcal{S}$ is a set of subsets $B \subseteq A$ such that $|B| \leq k$ and for all $\bar{a}$ considered so far (in earlier runs through the loop) we have $\mathcal{A}^* \models \bigvee_{i=1}^{m} \chi_i(B, \bar{a})$.

To get a bound on the running time, we note that whenever a new set is added to $\mathcal{S}$ (in Line 10) then it is an extension of a strictly smaller set that has just been removed from $\mathcal{S}$ (in Line 6). Furthermore, for each set removed (in Line 6) at most m such extensions can be added. Thus an upper bound for the number of sets that can be in $\mathcal{S}$ at any time is m^k. The main loop (in Lines 3–10) is called n^l times, where $n := |A|$. This gives an overall bound on the running time of $O(m^k \cdot n^l)$ plus the time needed to compute $\mathcal{A}^*$.

Since l does not depend on the instance, but just on the formula φ, this yields the fixed-parameter tractability of P. □

Besides *VC*, many other parameterized problems can be shown to be fixed-parameter tractable by a simple application of this theorem. Let us consider one example in detail: The *valence* of a graph is the maximal number of neighbors a vertex in the graph has. For an $l \geq 1$, we consider the restriction of *DOMINATING SET* to graphs of valence at most l, i.e. the problem

CHECK-$\varphi(\mathcal{A} \in \mathit{STR}[\tau], k \in \mathbb{N})$

```
1  compute A*
2  initialize set S ⊆ Pow(A) by S := {∅}
3  for all ā ∈ A^l do
4      for all B ∈ S do
5          if A* ⊭ ⋁_{i=1}^{m} χ_i(B, ā) then
6              S := S \ {B}
7              for i = 1 to m do
8                  compute B' := β(B, ā, i)
9                  if |B'| ≤ k and A* ⊨ χ_i(B', ā)
10                     then S := S ∪ {B'}
11 if S ≠ ∅
12     then accept
13     else reject.
```

Algorithm 1

VC_l		
	Input:	Graph $\mathcal{G}$.
	Parameter:	$k \in \mathbb{N}$.
	Question:	Is the valence of $\mathcal{G}$ at most l and does $\mathcal{G}$ have a dominating set of size at most k?

This problem is Fagin-defined by the following first-order formula:

$$\forall x \exists^{\leq l} z\ Exz \wedge \forall y_0 \forall y_1 \ldots \forall y_l \big(\forall z(Ey_0 z \rightarrow \bigvee_{i=1}^{l} z = y_i) \rightarrow \bigvee_{i=0}^{l} Xy_i\big).$$

($\exists^{\leq m} x \psi(x)$ abbreviates $\exists y_1 \ldots \exists y_m \forall x(\psi(x) \rightarrow \bigvee_{i=1}^{m} x = y_i)$.)

Other examples of problems that can be shown to be in FPT by Theorem 14 are *HITTING SET FOR SIZE THREE SETS*, *MATRIX DOMINATION*, or *SHORT 3DIMENSIONAL MATCHING* of [10].

5 Homomorphisms, embeddings, and model-checking

In this section we analyze the close relationship between the homomorphism problem, the embedding problem, and model-checking problems for Σ_1-formulas from the point of view of parameterized complexity (compare [25] for a further analysis of this relationship).

A *homomorphism* from a τ-structure $\mathcal{B}$ into a τ-structure $\mathcal{A}$ is a mapping $h : B \rightarrow A$ such that for all $R \in \tau$ and tuples $\bar{b} \in R^{\mathcal{B}}$ we have $h(\bar{b}) \in R^{\mathcal{A}}$. The parameterized *HOMOMORPHISM PROBLEM (HOM)* is defined as follows:

HOM		
	Input:	$\mathcal{A} \in$ *STR*.
	Parameter:	$\mathcal{B} \in$ *STR*.
	Question:	Is there a homomorphism from $\mathcal{B}$ to $\mathcal{A}$?

A *(weak) embedding* of $\mathcal{B}$ into $\mathcal{A}$ is an injective homomorphism from $\mathcal{B}$ to $\mathcal{A}$. Note that a graph $\mathcal{H}$ is isomorphic to a subgraph of a graph $\mathcal{G}$ in the usual graph theoretic sense, if there is an embedding of $\mathcal{H}$ into $\mathcal{G}$. Thus the following parameterized *EMBEDDING PROBLEM (EMB)* is a generalization of the subgraph isomorphism problem *SI*:

EMB		
	Input:	$\mathcal{A} \in$ *STR*.
	Parameter:	$\mathcal{B} \in$ *STR*.
	Question:	Is there an embedding of $\mathcal{B}$ into $\mathcal{A}$?

The *Gaifman graph* of a τ-structure $\mathcal{A}$ is the graph $\mathcal{G}(\mathcal{A})$ with universe A in which two elements $a \neq b$ are adjacent if there is an $R \in \tau$ and a tuple $\bar{a} \in R^{\mathcal{A}}$ such that both a and b occur in the tuple $\bar{a}$. For a class C of graphs we let *STR*$[C]$ denote the class of all structures whose Gaifman graph is in C. Note that *STR*$[$*GRAPH*$] =$ *STR*. Furthermore, we let *STR*$[\tau, C] :=$ *STR*$[\tau] \cap$ *STR*$[C]$ for every vocabulary τ and *STR*$[s, C] :=$ *STR*$[s] \cap$ *STR*$[C]$ for every $s \geq 1$. We define restrictions *HOM*$[\dots]$ and *EMB*$[\dots]$ of the respective problems, where for every possible restriction '$\dots$' in the square brackets we require the *parameter* $\mathcal{B}$ to belong to the class *STR*$[\dots]$. For example, we let

$$\textit{EMB}[s, C] := \textit{EMB} \cap (\textit{STR} \times \textit{STR}[s, C]).$$

Lemma 16 *For all classes C of graphs and $s \geq 1$ we have* *HOM*$[C] \leq_{\mathrm{m}}^{\mathrm{fpp}}$ *EMB*$[C]$ *and* *HOM*$[s, C] \leq_{\mathrm{m}}^{\mathrm{fpp}}$ *EMB*$[s, C]$.

Proof: Suppose we are given an instance $(\mathcal{A}, \mathcal{B})$ of *HOM*$[C]$. Let τ be the vocabulary of $\mathcal{A}$. Let $\mathcal{A}_B$ be the τ-structure which for every element of $\mathcal{A}$ contains $|B|$ duplicates, i.e. $A_B := A \times B$ and, for every r-ary $R \in \tau$,

$$R^{\mathcal{A}_B} := \big\{((a_1, b_1), \dots, (a_r, b_r)) \;\big|\; R^{\mathcal{A}} a_1 \dots a_r\big\}.$$

Then, every homomorphism $h : \mathcal{B} \to \mathcal{A}$ gives rise to an embedding $f : \mathcal{B} \to \mathcal{A}_B$ defined by $f(b) = (h(b), b)$ and every embedding $f : \mathcal{B} \to \mathcal{A}_B$ induces a homomorphism $h : \mathcal{B} \to \mathcal{A}$ defined by letting $h(b)$ be the projection on the first component of $f(b)$. □

Note that, unless PTIME $=$ NP, there is no polynomial-time reduction from *EMB*$[C]$ to *HOM*$[C]$ for the class C of all paths, because *HOM*$[C]$ can easily be seen to be in PTIME by a dynamic programming algorithm, whereas *EMB*$[2, C]$ is NP-complete by a reduction from *HAMILTONIAN PATH*. Thus *EMB*$[s, C] \leq_{\mathrm{m}}^{\mathrm{fpp}}$ *HOM*$[s, C]$ does not hold for any $s \geq 2$. However, we will see that for all $s \geq 2$ and C we actually have *HOM*$[s, C] \equiv_{\mathrm{T}}^{\mathrm{fp}}$ *EMB*$[s, C]$.

Before we show this, we introduce two related model-checking problems. With each first-order formula φ we associate a graph $\mathcal{G}(\varphi)$. Its universe is $\mathrm{var}(\varphi)$, the set of all variables in φ, and there is an edge between distinct $x, y \in \mathrm{var}(\varphi)$ in $\mathcal{G}(\varphi)$ if φ has an atomic subformula in which both x, y occur.

Let $\varphi^{\neq}$ be the formula obtained from φ by deleting all inequalities, i.e. all atomic subformulas of the form $x = y$ that occur in the scope of an odd number of negation symbols. We are also interested in $\mathcal{G}(\varphi^{\neq})$. Let us see an example:

$$\varphi := \exists x_1 \dots \exists x_k \Big(\bigwedge_{1 \leq i < j \leq k} \neg x_i = x_j \wedge \bigwedge_{i=1}^{k-1} E x_i x_{i+1} \Big).$$

$\mathcal{G}(\varphi)$ is the complete graph with vertex set $\{x_1, \dots, x_k\}$, whereas $\mathcal{G}(\varphi^{\neq})$ is the path $x_1 \dots x_k$. Note that φ says that a graph has a subgraph isomorphic to a path of length k, whereas $\varphi^{\neq}$ says that a graph contains a homomorphic image of a path of length k. This generalizes to the following simple lemma, whose proof we omit.

Lemma 17 *For every structure $\mathcal{B} \in$ STR there is a Σ_1-sentence $\varphi_{\mathcal{B}}$ (whose quantifier-free part is a conjunction of literals) such that $\mathcal{G}(\varphi_{\mathcal{B}}^{\neq}) = \mathcal{G}(\mathcal{B})$, and for every structure $\mathcal{A}$ we have:*

$$\mathcal{A} \models \varphi_{\mathcal{B}} \iff \textit{There is an embedding of } \mathcal{B} \textit{ into } \mathcal{A}.$$
$$\mathcal{A} \models \varphi_{\mathcal{B}}^{\neq} \iff \textit{There is a homomorphism from } \mathcal{B} \textit{ to } \mathcal{A}.$$

Furthermore, the mapping $\mathcal{B} \mapsto \varphi_{\mathcal{B}}$ is computable in linear time.

For $s \in \mathbb{N}$ and a class C of graphs we let

$$\Sigma_1[s, C] := \big\{\varphi \in \Sigma_1[\tau] \;\big|\; \tau \text{ } s\text{-ary vocabulary}, \mathcal{G}(\varphi) \in C\big\},$$
$$\Sigma_1^{\neq}[s, C] := \big\{\varphi \in \Sigma_1[\tau] \;\big|\; \tau \text{ } s\text{-ary vocabulary}, \mathcal{G}(\varphi^{\neq}) \in C\big\}.$$

Furthermore, we let $\Sigma_1[C] := \bigcup_{s\geq 1} \Sigma_1[s,C]$ and $\Sigma_1^{\neq}[C] := \bigcup_{s\geq 1} \Sigma_1^{\neq}[s,C]$.

Lemma 17 implies that for every $s \geq 1$ and for every class C of graphs we have $\mathit{HOM}[s,C] \leq_{\mathrm{m}}^{\mathrm{fpp}} \mathit{MC}(\Sigma_1[s,C])$ and $\mathit{EMB}[s,C] \leq_{\mathrm{m}}^{\mathrm{fpp}} \mathit{MC}(\Sigma_1^{\neq}[s,C])$. Unless PTIME = NP, the converse of these statements is wrong. To see this, let C be the class of all graphs that only consist of isolated vertices, i.e. all graphs $\mathcal{G}$ with $E^{\mathcal{G}} = \emptyset$. Then clearly $\mathit{HOM}[C]$ and $\mathit{EMB}[C]$ are in PTIME, but we can reduce the satisfiability problem for propositional formulas to $\mathit{MC}(\Sigma_1[1,C])$ and $\mathit{MC}(\Sigma_1^{\neq}[1,C])$.

Theorem 18 *Let C be a class of graphs and $s \geq 2$. Then*

$$\mathit{HOM}[s,C] \equiv_{\mathrm{T}}^{\mathrm{fp}} \mathit{EMB}[s,C] \equiv_{\mathrm{T}}^{\mathrm{fp}} \mathit{MC}(\Sigma_1^{\neq}[s,C]) \equiv_{\mathrm{T}}^{\mathrm{fp}} \mathit{MC}(\Sigma_1[s,C]).$$

Proof: We have already seen that $\mathit{HOM}[s,C] \leq_{\mathrm{T}}^{\mathrm{fp}} \mathit{EMB}[s,C]$ (Lemma 16) and that $\mathit{EMB}[s,C] \leq_{\mathrm{T}}^{\mathrm{fp}} \mathit{MC}(\Sigma_1^{\neq}[s,C])$ (Lemma 17). To complete the cycle we shall prove that $\mathit{MC}(\Sigma_1^{\neq}[s,C]) \leq_{\mathrm{T}}^{\mathrm{fp}} \mathit{MC}(\Sigma_1[s,C])$ and that $\mathit{MC}(\Sigma_1[s,C]) \leq_{\mathrm{T}}^{\mathrm{fp}} \mathit{HOM}[s,C]$.

We first prove that $\mathit{MC}(\Sigma_1[s,C]) \leq_{\mathrm{T}}^{\mathrm{fp}} \mathit{HOM}[s,C]$. Let $\varphi \in \Sigma_1[s,C]$, say, of vocabulary τ, and $\mathcal{A}$ a τ-structure. We shall describe an algorithm that decides whether $\mathcal{A} \models \varphi$ using $\mathit{HOM}[s,C]$ as an oracle.

Let S be a binary relation symbol not contained in τ and $\tau' := \tau \cup \{S\}$. Furthermore, let $\mathcal{A}'$ be the τ'-expansion of $\mathcal{A}$ with $S^{\mathcal{A}'} = \emptyset$. Our algorithm first computes a sentence $\varphi' := \bigvee_{i=1}^{m} \exists \bar{x}_i \psi_i$ of vocabulary τ' such that

(1) $\mathcal{A} \models \varphi$ if, and only if, $\mathcal{A}' \models \varphi'$.

(2) For $1 \leq i \leq m$, the formula ψ_i is a conjunction of literals, and we have $\mathcal{G}(\psi_i) = \mathcal{G}(\varphi)$.

This can be achieved by first translating φ to a sentence whose quantifier-free part is in disjunctive normal form, then swapping existential quantifiers and the disjunction, and then adding dummy literals of the form $\neg Sxy$ until $\mathcal{G}(\psi_i) = \mathcal{G}(\varphi)$.

Let $\tau'' := \tau' \cup \{\overline{R} \mid R \in \tau'\} \cup \{E, \overline{E}\}$, where for all $R \in \tau'$ the symbol $\overline{R}$ is a new relation symbol of the same arity as R and $E, \overline{E}$ are new binary relation symbols. Let $\mathcal{A}''$ be the τ'' expansion of $\mathcal{A}'$ in which $\overline{R}$ is interpreted as the complement of $R^{\mathcal{A}''}$ and $E, \overline{E}$ are interpreted as equality and inequality, respectively. For $1 \leq i \leq m$, we define a τ''-structure $\mathcal{B}_i$ with $\mathcal{G}(\mathcal{B}_i) = \mathcal{G}(\varphi)$ such that $\mathcal{A}' \models \exists \bar{x}_i \psi_i$ if, and only if, there is a homomorphism from $\mathcal{B}_i$ into $\mathcal{A}''$. We let $\mathcal{B}_i$ be the τ''-structure with universe $\mathrm{var}(\psi_i)$ and

$$\begin{aligned}
R^{\mathcal{B}_i} &:= \{\overline{y} \mid R\overline{y} \text{ is a literal of } \psi_i\} \qquad (\text{for } R \in \tau'),\\
\overline{R}^{\mathcal{B}_i} &:= \{\overline{y} \mid \neg R\overline{y} \text{ is a literal of } \psi_i\} \qquad (\text{for } R \in \tau'),\\
E^{\mathcal{B}_i} &:= \{yz \mid y = z \text{ is a literal of } \psi_i\},\\
\overline{E}^{\mathcal{B}_i} &:= \{yz \mid \neg y = z \text{ is a literal of } \psi_i\}.
\end{aligned}$$

It is obvious that $\mathcal{B}_i$ does indeed have the desired property. Altogether, our construction yields a parameterized T-reduction.

It remains to prove that $\mathit{MC}(\Sigma_1^{\neq}[s,C]) \leq_{\mathrm{T}}^{\mathrm{fp}} \mathit{MC}(\Sigma_1[s,C])$. We use the so called *color coding* technique of Alon, Yuster, and Zwick [2].

Let $l \geq 1$ and X a set. An *l-perfect family of hash functions on X* is a family F of functions $f : X \to \{1,\ldots,l\}$ such that for all subsets $Y \subseteq X$ of size l there is an $f \in F$ such that $f(Y) = \{1,\ldots,l\}$ (i.e. on Y, f is one-to-one). Alon, Yuster, and Zwick [2] show that given $n, l \geq 1$, an l-perfect family of hash functions on $\{1,\ldots,n\}$ of size $2^{O(l)} \cdot \log n$ can be computed in time $2^{O(l)} \cdot n \cdot \log n$.

For a similar reason as outlined above, without loss of generality we can restrict our attention to sentences $\varphi \in \mathit{MC}(\Sigma_1^{\neq}[s,C])$ whose quantifier-free part is a conjunction of literals. Given such a sentence $\varphi = \exists x_1 \ldots \exists x_k \psi$, say of vocabulary τ, and a τ-structure $\mathcal{A}$, we define a family of sentences $\varphi_\gamma \in \mathit{MC}(\Sigma_1[s,C])$ and a family of structures $\mathcal{A}_f$ such that $\mathcal{A} \models \varphi$ if, and only if, there is a γ and f such that $\mathcal{A}_f \models \varphi_\gamma$.

A *coloring of* φ is a function $\gamma : \{1,\ldots,k\} \to \{1,\ldots,k\}$ such that $\gamma(i) \neq \gamma(j)$ if $\neg x_i = x_j$ occurs in φ. For a coloring γ we let ψ_γ be the formula obtained from ψ by replacing all literals $\neg x_i = x_j$ by $C_{\gamma(i)}x_i \wedge C_{\gamma(j)}x_j$ (here,

$C_1, \ldots, C_k$ are new unary "color" relation symbols) and let $\varphi_\gamma := \exists x_1 \ldots \exists x_k \psi_\gamma$. Note that $\mathcal{G}(\varphi_\gamma) = \mathcal{G}(\varphi^{\neq})$. Thus $\varphi_\gamma \in MC(\Sigma_1[s, C])$.

With every $f : A \to \{1, \ldots, k\}$, which we call a *coloring of* $\mathcal{A}$, we let $\mathcal{A}_f$ be the $\tau \cup \{C_1, \ldots, C_k\}$-expansion of $\mathcal{A}$ with $C_i^{\mathcal{A}_f} := f^{-1}(i)$ for $1 \leq i \leq k$.

Observe that

$$\mathcal{A} \models \varphi \iff \begin{array}{l}\text{there is a coloring } \gamma \text{ of } \varphi \text{ and a coloring } f \text{ of } \mathcal{A} \text{ such that} \\ \mathcal{A}_f \models \varphi_\gamma.\end{array} \tag{3}$$

The problem is that there are $k^{|A|}$ colorings of $\mathcal{A}$, so (3) does not yet give rise to a parameterized reduction. The crucial trick is that to achieve this equivalence we do not have to consider all possible colorings f of $\mathcal{A}$. For $1 \leq l \leq k$, let F_l be an l-perfect family of hash-function on A and $F := \bigcup_{l=1}^{k} F_l$. We claim that

$$\mathcal{A} \models \varphi \iff \text{there is a coloring } \gamma \text{ of } \varphi \text{ and an } f \in F \text{ such that } \mathcal{A}_f \models \varphi_\gamma. \tag{4}$$

The backward direction follows immediately from (3). For the forward direction, suppose that $\mathcal{A} \models \varphi$. Let $\bar{a} \in A^k$ such that $\mathcal{A} \models \psi(\bar{a})$. There is a function $f \in F$ whose restriction to $\{a_1, \ldots, a_k\}$ is one-to-one. Define γ by $\gamma(i) := f(a_i)$. Then, γ is a coloring of φ, $\mathcal{A}_f \models \psi_\gamma(\bar{a})$ and hence, $\mathcal{A}_f \models \varphi_\gamma$.

Since the family F can be chosen sufficiently small and computed sufficiently fast, the equivalence (4) gives rise to a parameterized reduction. □

Remark 19 We do not know if the parameterized T-reductions in Theorem 18 can be replaced by parameterized m-reductions. However, for many interesting classes C they can be replaced. One such example is the class of all graphs. Similar techniques work for all classes C of graphs for which there exists an algorithm that, given a graph $\mathcal{H} \in C$, computes a connected $\mathcal{H}' \in C$ such that $\mathcal{H}$ is a subgraph of $\mathcal{H}'$. For all such classes C we can show that

$$HOM[s, C] \equiv_{\mathrm{m}}^{\mathrm{fp}} EMB[s, C] \equiv_{\mathrm{m}}^{\mathrm{fp}} MC(\Sigma_1^{\neq}[s, C]) \equiv_{\mathrm{m}}^{\mathrm{fp}} MC(\Sigma_1[s, C]).$$

(for all $s \geq 1$).

5.1 Sentences of bounded tree-width Our main application of Theorem 18 is to sentences whose underlying graphs have bounded tree-width. In the time since we submitted this article, considerable progress has been made in this area. For an extensive discussion of model-checking algorithms based on tree-decompositions of the sentences, we refer the reader to [16].

We think of a *tree* $\mathcal{T}$ as directed from its root, which we denote by $r^{\mathcal{T}}$, to the leaves and thus can speak of a *child* and of the *parent* of a vertex.

A *tree-decomposition* of a τ-structure $\mathcal{A}$ is a pair $(\mathcal{T}, (A_t)_{t \in T})$, where $\mathcal{T}$ is a tree and $(A_t)_{t \in T}$ a family of subsets of A such that

(1) For every $a \in A$, the set $\{t \in T \mid a \in A_t\}$ is non-empty and induces a subtree of $\mathcal{T}$ (that is, is connected).

(2) For every k-ary relation symbol $R \in \tau$ and all $a_1, \ldots, a_k \in A$ such that $R^{\mathcal{A}} a_1, \ldots, a_k$ there exists a $t \in T$ such that $a_1, \ldots, a_k \in A_t$.

The *width* of a tree-decomposition $(\mathcal{T}, (A_t)_{t \in T})$ is $\max\{|A_t| \mid t \in T\} - 1$. The *tree-width* $\mathrm{tw}(\mathcal{A})$ of $\mathcal{A}$ is the minimal width of a tree-decomposition of $\mathcal{A}$.

For $s \geq 1$, let W_s denote the class of all structures of tree-width at most s and $GW_s := GRAPH \cap W_s$. Note that $W_s \subseteq STR[s + 1]$, because a graph of tree-width s has clique number at most $s + 1$.[2] (The clique number of a graph $\mathcal{G}$ is the maximal cardinality of a set of pairwise adjacent vertices of $\mathcal{G}$.)

Plehn and Voigt [27] were the first to realize that tree-width is a relevant parameter for the problems considered in the previous section. They proved that the parameterized embedding problem restricted to (parameter) graphs of bounded tree-width is fixed parameter tractable. Chekuri and Rajaraman [6] proved that for every $s \geq 1$ the problem $HOM[GW_s]$ is in PTIME (when considered as an unparameterized problem) and therefore fixed-parameter tractable. They phrased their result in terms of the equivalent *conjunctive query containment* problem (also see [25]).

Thus as a corollary of Theorem 18 we obtain:

[2] This is slightly imprecise, because a structure $\mathcal{A} \in W_s$ might have a vocabulary of arbitrarily high arity, as long as no tuple contained in a relation of $\mathcal{A}$ consists of more than $s + 1$ distinct elements. But since any structure with this property can easily be transformed to an $(s + 1)$-ary structure that is essentially the same, we decided to accept this imprecision in exchange for a simpler notation.

Corollary 20 *Let $s \geq 1$. Then the problems EMB$[GW_s]$, MC$(\Sigma_1[GW_s])$, and MC$(\Sigma_1^{\neq}[GW_s])$ are in* FPT.

Papadimitriou and Yannakakis [26] proved the model-checking results of this corollary for the related case of acyclic conjunctive queries.

Unfortunately, it turns out that Corollary 20 is the only real application of Theorem 18. Very recently, Schwentick, Segoufin, and the second author [20] have proved that for every class C of graphs of unbounded tree-width and every $s \geq 2$, the problem *HOM*$[C, s]$ is W[1]-complete under parameterized T-reductions.

6 FO-model-checking on graphs with excluded minors

The fixed-parameter tractability results of the previous section were obtained by putting syntactical restrictions on the sentences, i.e. the *parameter* of the model-checking problem. In this section we put restrictions on the structures, i.e. the *input* of the model-checking problem.

Recall the definition of the parameterized problem *MC*$(\Phi)|_D$, for a class Φ of formulas and a class D of structures:

MC$(\Phi)|_D$		
	Input:	$\mathcal{A} \in$ *STR*.
	Parameter:	$\varphi \in \Phi$.
	Question:	Is $\mathcal{A} \in D$ and $\mathcal{A} \models \varphi$?

Our starting point is the following theorem due to Courcelle. Remember that *monadic second-order logic* is the extension of first-order logic where one is allowed to quantify not only over individual elements of a structure but also over sets of elements. MSO denotes the class of all formulas of monadic second-order logic. Remember that W_s denotes the class of all structures of tree-width at most s (for $s \geq 0$).

Theorem 21 ([7]) *Let $s \geq 0$. Then MC*(MSO)$|_{W_s}$ *is in* FPT.

A graph $\mathcal{H}$ is a *minor* of a graph $\mathcal{G}$ (we write $\mathcal{H} \preceq \mathcal{G}$) if $\mathcal{H}$ is can be obtained from a subgraph of $\mathcal{G}$ by contracting edges. $\mathcal{H}$ is an *excluded minor* for a class D if $\mathcal{H}$ is not a minor of any graph in D. Note that a class D of graphs has an excluded minor if, and only if, there is an $n \in \mathbb{N}$ such that $\mathcal{K}_n$ is an excluded minor for D.

Examples of classes of graphs with an excluded minor are classes of graphs of bounded tree-width or classes of graphs embeddable in a fixed surface.

Recall that for a class D of graphs, *STR*$[D]$ denotes the class of all structures whose Gaifman graph is in C.

Theorem 22 *Let D be a* PTIME*-decidable class of graphs with an excluded minor. Then MC*(FO)$|_{STR[D]}$ *is in* FPT.

The rest of this section is devoted to the proof of this theorem, which needs some preparation.

A class D of graphs is called *minor closed* if for all $\mathcal{G} \in D$ and $\mathcal{H} \preceq \mathcal{G}$ we have $\mathcal{H} \in D$. Robertson and Seymour proved:

Theorem 23 ([28]) *Every minor-closed class of graphs is* PTIME*-decidable.*

This, together with Theorem 22, immediately yields:

Corollary 24 *Let $D \subsetneq GRAPH$ be minor closed. Then MC*(FO)$|_{STR[D]}$ *is in* FPT.

Recall the definition of the Gaifman graph $\mathcal{G}(\mathcal{A})$ of a structure $\mathcal{A}$ (cf. Page 13). The distance $d^{\mathcal{A}}(a, b)$ between $a, b \in A$ is the length of the shortest path from a to b in $\mathcal{G}(\mathcal{A})$. For $r \in \mathbb{N}$ and $a \in A$, the *r-ball* around a is the set $B_r^{\mathcal{A}}(a) := \{b \in A \mid d^{\mathcal{A}}(a, b) \leq r\}$. For an $X \subseteq A$, $\langle X \rangle^{\mathcal{A}}$ denotes the substructure induced by $\mathcal{A}$ on X, i.e. the structure with universe X and $R^{\langle X \rangle^{\mathcal{A}}} = R^{\mathcal{A}} \cap X^r$ for all r-ary relation symbols R in the vocabulary of $\mathcal{A}$. Furthermore, we let $\mathcal{A} \setminus X := \langle A \setminus X \rangle^{\mathcal{A}}$.

The *local tree-width* of $\mathcal{A}$ is the function ltw$(\mathcal{A}) : \mathbb{N} \to \mathbb{N}$ defined by

$$\text{ltw}(\mathcal{A})(r) := \max\big\{\text{tw}(\langle B_r^{\mathcal{A}}(a) \rangle^{\mathcal{A}}) \;\big|\; a \in A\big\}.$$

For functions $f, g : \mathbb{N} \to \mathbb{N}$ we write $f \leq g$ if $f(n) \leq g(n)$ for all $n \in \mathbb{N}$. A class D of structures has *bounded local tree-width* if there is a function $\lambda : \mathbb{N} \to \mathbb{N}$ such that for all $\mathcal{A} \in D$ we have $\mathrm{ltw}(\mathcal{A}) \leq \lambda$.

The "local" character of first-order formulas allows to generalize Theorem 21 for first-order logic from classes of structures of bounded tree-width to classes of structures of bounded local tree-width:

Theorem 25 ([17]) *Let D be a* PTIME*-decidable class of structures of bounded local tree-width. Then MC*$(\mathrm{FO})|_D$ *is in* FPT.

For $\lambda : \mathbb{N} \to \mathbb{N}$ we let

$$GL(\lambda) := \big\{\mathcal{G} \in \mathit{GRAPH} \;\big|\; \forall \mathcal{H} \preceq \mathcal{G} : \ \mathrm{ltw}(\mathcal{H}) \leq \lambda\big\},$$

and, for $\mu \in \mathbb{N}$,

$$B(\lambda, \mu) := \big\{\mathcal{A} \in \mathit{STR} \;\big|\; \exists X \subseteq A \ (|X| \leq \mu \wedge \mathcal{G}(\mathcal{A} \setminus X) \in GL(\lambda))\big\}.$$

Note that the clique-number of a graph in $GL(\lambda)$ is at most $\lambda(1) + 1$, thus the clique-number of a graph in $B(\lambda, \mu)$ is at most $\mu + \lambda(1) + 1$. This implies that $B(\lambda, \mu) \subseteq \mathit{STR}[\mu + \lambda(1) + 1]$.[3]

Lemma 26 *Let $\lambda : \mathbb{N} \to \mathbb{N}$ and $\mu \in \mathbb{N}$. Then MC*$(\mathrm{FO})|_{B(\lambda,\mu)}$ *is in* FPT.

Proof: The class $GL(\lambda)$ of graphs is minor closed and hence PTIME-decidable by Theorem 23. This implies that $B(\lambda, \mu)$ is PTIME-decidable.

Then for $\mu = 0$ the statement follows from Theorem 25. The case $\mu > 0$ can be reduced to the case $\mu = 0$ as follows: For every $\mathcal{A} \in \mathcal{B}(\lambda, \mu)$ and sentence $\varphi \in \mathrm{FO}$, we define a structure $\mathcal{A}^* \in B(\lambda, 0)$ and a sentence $\varphi^* \in \mathrm{FO}$ in such a way that $A \models \varphi$ if, and only if, $\mathcal{A}^* \models \varphi^*$, and the mappings $\mathcal{A} \mapsto \mathcal{A}^*$ and $\varphi \mapsto \varphi^*$ are computable in polynomial time.

So suppose we are given $\mathcal{A} \in \mathcal{B}(\lambda, \mu)$ and $\varphi \in \mathrm{FO}$. For simplicity, we assume that their vocabulary consists of a single binary relation symbol E.

Let $X \subseteq A$ such that $|X| \leq \mu$ and $\mathcal{A} \setminus X \in B(\lambda, 0)$. Such an X can be computed in polynomial time because the class $B(\lambda, 0)$ is PTIME-decidable. Say, $X = \{a_1, \ldots, a_\mu\}$.

Let $\tau := \{E, P_1, \ldots, P_\mu, Q_1, \ldots, Q_\mu, R_1, \ldots, R_\mu\}$, where the P_i, Q_i, and R_i are unary. We let $A^* := A$, $E^{\mathcal{A}^*} := E^{\mathcal{A}} \cap (A \setminus X)^2$, and, for $1 \leq i \leq \mu$

$$\begin{aligned} P_i^{\mathcal{A}^*} &:= \{a_i\}, \\ Q_i^{\mathcal{A}^*} &:= \{b \in A \mid E^{\mathcal{A}} a_i b\}, \\ R_i^{\mathcal{A}^*} &:= \{b \in A \mid E^{\mathcal{A}} b a_i\}. \end{aligned}$$

Furthermore, we let φ^* be the sentence obtained from φ by replacing each subformula Exy by

$$Exy \vee \bigvee_{i=1}^{\mu} ((P_i x \wedge Q_i y) \vee (P_i y \wedge R_i x)).$$

Clearly, these definitions lead to the desired result. □

To complete the proof of Theorem 22 we use a decomposition theorem for non-trivial minor-closed classes of graphs that roughly says that all graphs in such a class are built up in a tree-like manner from graphs in a $B(\lambda, \mu)$. It is based on Robertson and Seymour's deep structure theory for graphs without $\mathcal{K}_n$-minors. The precise statement requires some new notation. Let $(\mathcal{T}, (A_t)_{t \in T})$ be a tree-decomposition of a structure $\mathcal{A}$. The *torso* of this decomposition at $t \in \mathcal{T}$, denoted by $[A_t]$, is the graph with universe A_t and an edge between two distinct vertices $a, b \in A_t$ if either there is an edge between a and b in the Gaifman graph $\mathcal{G}(\mathcal{A})$ or there exists an $s \in T \setminus \{t\}$ such that $a, b \in A_s$. $(\mathcal{T}, (A_t)_{t \in T})$ is a tree-decomposition *over* a class D of graphs if all its torsos belong to D.

[3] Cf. Footnote [2] on Page 15.

Theorem 27 ([19]) *Let D be a class of graphs with an excluded minor. Then there exist $\lambda : \mathbb{N} \to \mathbb{N}$ and $\mu \in \mathbb{N}$ such that every $\mathcal{G} \in D$ has a tree-decomposition over $B(\lambda, \mu)$.*

Furthermore, given $\mathcal{G}$ such a decomposition can be computed in PTIME.

Clearly, this theorem implies the analogous statement for all structures in *STR*$[D]$.

The *adhesion* of a tree-decomposition $(\mathcal{T}, (A_t)_{t\in T})$ is $\max\{|A_s \cap A_t| \mid E^{\mathcal{T}}st\}$. The *clique number* of a class of structures is the maximum of the clique numbers of the Gaifman graphs of structures in D, if this maximum exists, or ∞ otherwise. Note that if $(\mathcal{T}, (A_t)_{t\in T})$ is a decomposition over a class D then the clique-number of D is an upper bound for the adhesion of $(\mathcal{T}, (A_t)_{t\in T})$. Remembering that the clique-number of $B(\lambda, \mu)$ is $\lambda(1) + \mu + 1$, we see that the adhesion of a tree-decomposition over $B(\lambda, \mu)$ is at most $\lambda(1) + \mu + 1$.

Proof (of Theorem 22): Let D be a PTIME-decidable class of graphs with an excluded minor and λ, μ such that every $\mathcal{G} \in D$ has a tree-decomposition over $B(\lambda, \mu)$. Let $\nu := \lambda(1) + \mu + 1$.

We shall describe an algorithm that, given $\mathcal{A} \in$ *STR*$[D]$ and $\varphi \in$ FO, decides if $A \models \varphi$.

So let $\mathcal{A} \in$ *STR*$[D]$, say, of vocabulary τ and $\varphi \in$ FO$[\tau]$. Our algorithm starts by computing a tree-decomposition $(\mathcal{T}, (A_t)_{t\in T})$ of $\mathcal{A}$ over $B(\lambda, \mu)$. For $t \in T$ we let $A_{\geq t} := \bigcup_{u\geq t} A_u$ ($u \geq t$ if there is a path from t to u in the (directed) tree $\mathcal{T}$). In particular, $A_{\geq r} = A$ for the root $r := r^{\mathcal{T}}$ of $\mathcal{T}$.

Furthermore, we let $B_r := \emptyset$ and $B_t := A_t \cap A_s$ for $t \in T \setminus \{r\}$ with parent s. Recall that the adhesion of $(\mathcal{T}, (A_t)_{t\in T})$ is at most ν. Thus $|B_t| \leq \nu$ for $t \in T$.

The *quantifier rank* of a first-order formula is the maximal depth of nested quantifiers in this formula. Let q be the quantifier rank of φ. Simple techniques from logic show that there is an algorithm that, given a vocabulary τ and $q, m \in \mathbb{N}$, computes a finite set $\Phi_{\tau,q,m}$ of first-order formulas of vocabulary τ of quantifier rank $\leq q$ with free variables among $v_1, \ldots, v_m$ such that every such formula is equivalent to a formula in $\Phi_{\tau,q,m}$. Without loss of generality we can assume that $\varphi \in \Phi_{\tau,q,m}$ (otherwise we can compute a $\varphi' \in \Phi_{\tau,q,m}$ equivalent to φ and work with φ').

A (τ, q, m)-*type* is a subset of $\Phi_{\tau,q,m}$. Given a τ-structure $\mathcal{A}'$ and $B = \{b_1, \ldots, b_m\} \subseteq A'$ let $\mathrm{tp}_q^{\mathcal{A}'}(B)$ – more precisely, $\mathrm{tp}_q^{\mathcal{A}'}(b_1, \ldots, b_m)$ – be the (τ, q, m)-type

$$\mathrm{tp}_q^{\mathcal{A}'}(B) := \{\psi(v_1, \ldots, v_m) \in \Phi_{\tau,q,m} \mid \mathcal{A}' \models \psi(b_1, \ldots, b_m)\}.$$

We come back to our structure $\mathcal{A}$ and the tree-decomposition $(\mathcal{T}, (A_t)_{t\in T})$. By induction from the leaves to the root, for every $t \in T$ we compute $\mathrm{tp}_q^{\langle A_{\geq t}\rangle^{\mathcal{A}}}(B_t)$, which for brevity we denote by $\mathrm{tp}_q^{\geq t}(B_t)$. Since $B_r = \emptyset$, $\mathrm{tp}_q^{\geq r}(B_r)$ is a set of sentences, and we have $\mathcal{A} \models \varphi$ if, and only if, $\varphi \in \mathrm{tp}_q^{\geq r}(B_r)$.

So let t be a vertex of $\mathcal{T}$ and assume that we have already computed $\mathrm{tp}_q^{\geq u}(B_u)$ for all children u of t (if there are any). (Actually, the case that t has no children is much simpler than the following general case, because it is a direct application of Lemma 26.)

For every (τ, q, m)-type Φ we introduce a new $(m+1)$-ary relation symbol R_Φ. Furthermore, we let $P_1, \ldots, P_\nu$ be new unary relation symbols and $\tau' := \tau \cup \{R_\Phi \mid m \leq \nu, \Phi \text{ a } (\tau, q, m)\text{-type}\} \cup \{P_1, \ldots, P_\nu\}$. In four steps, we define a τ'-structure $\widetilde{\mathcal{A}}_t$ that contains all the relevant information to compute $\mathrm{tp}_q^{\geq t}(B_t)$. In the first three steps we define "intermediate" structures $\mathcal{A}_t^1, \mathcal{A}_t^2, \mathcal{A}_t^3$.

(1) $\mathcal{A}_t^1$ is the induced substructure of $\mathcal{A}$ with universe A_t.

(2) Suppose that $B_t = \{b_1, \ldots, b_m\}$ for an $m \leq \nu$. Then $\mathcal{A}_t^2$ is the $\tau \cup \{P_1, \ldots, P_\nu\}$-expansion of $\mathcal{A}_t^1$ with $P_i^{\mathcal{A}_t^2} := \{b_i\}$ for $1 \leq i \leq m$ and $P_i^{\mathcal{A}_t^2} := \emptyset$ for $m+1 \leq i \leq \nu$.

(3) $\mathcal{A}_t^3$ is obtained from $\mathcal{A}_t^2$ by adding a new vertex c_u for every child u of t and edges from c_u to all vertices of B_u.

(4) $\widetilde{\mathcal{A}}_t$ is the τ'-expansion of $\widetilde{\mathcal{A}}_t^3$ with

$$R_\Phi^{\widetilde{\mathcal{A}}_t} := \big\{(d_1, \ldots, d_{m_u}, c_u) \;\big|\; u \text{ child of } t, \; B_u = \{d_1, \ldots, d_{m_u}\}, \mathrm{tp}_q^{\geq u}(B_u) = \Phi\big\}.$$

MODELCHECK$_D$($\mathcal{A} \in$ *STR*, $\varphi \in$ FO)

1 **if** $\mathcal{G}(\mathcal{A}) \notin D$ **then** reject
2 compute tree-decomposition $(\mathcal{T}, (A_t)_{t\in T})$ of $\mathcal{A}$ over $B(\lambda, \mu)$
3 $q :=$ qr(φ), $\tau :=$ vocabulary of φ
4 **for** $m = 0$ **to** ν
5 compute $\Phi_{\tau,q,m}$
6 **for all** $t \in T$ (from the leaves to the root)
7 compute $\widetilde{\mathcal{A}}_t$
8 $\mathrm{tp}_q^{\geq t}(B_t) := \emptyset$
9 $m := |B_t|$
10 **for all** $\psi \in \Phi_{\tau,q,m}$
11 compute $\widetilde{\psi}$
12 **if** $\widetilde{\mathcal{A}}_t \models \widetilde{\psi}$
13 **then** $\mathrm{tp}_q^{\geq t}(B_t) := \mathrm{tp}_q^{\geq t}(B_t) \cup \{\psi\}$
14 **if** $\varphi \in \mathrm{tp}_q^{\geq r}(B_r)$
15 **then** accept
16 **else** reject.

Algorithm 2

Standard Ehrenfeucht-Fraïssé type methods show that there is a computable function that associates with every formula $\psi \in \Phi_{\tau,q,m}$ a sentence $\widetilde{\psi} \in \mathrm{FO}[\tau']$ such that $\psi \in \mathrm{tp}_q^{\geq t}(B_t)$ if, and only if, $\widetilde{\mathcal{A}}_t \models \widetilde{\psi}$.

We claim that $\widetilde{\mathcal{A}}_t \in B(\lambda + 1, \mu)$. To see this, observe that the Gaifman graph $\mathcal{G}(\widetilde{\mathcal{A}}_t)$ is the graph obtained from the torso $[A_t]$ by adding the vertices c_u and edges between c_u and every element of B_u. Recall that, by the definition of the torso, each B_u is a clique in $[A_t]$. It is easy to see that adding vertices and connecting them with cliques can increase the tree-width of a graph by at most one. This implies the claim.

Now we can put everything together and obtain an algorithm deciding $MC(\mathrm{FO})_{STR[D]}$, a high-level description of which is given as Algorithm 2. Its correctness is straightforward. Let us just have a look at the running time: Let n be the size of the input structure. Then Lines 1 and 2 require time polynomial in n (indendently of φ). The time required in Lines 3–5 only depends on φ. The main loop in Lines 6–13 is called $|T|$ times, which is polynomial in n. Computing $\widetilde{\mathcal{A}}_t$ is polynomial in $|A_t|$ and the number of children of t since we have already computed $\mathrm{tp}_q^{\geq u}(B_u)$ for all children u of t (with constants heavily depending on $||\varphi||$). The main task is to decide whether $\widetilde{\mathcal{A}}_t \models \widetilde{\psi}$ in Line 12; by Lemma 26 this is fixed-parameter tractable because $\widetilde{\mathcal{A}}_t \in B(\lambda + 1, \mu)$. The time required in Lines 14–16 again only depends on φ. □

A consequence of our results is that slicewise first-order definable parameterized problems are fixed-parameter tractable when restricted to classes of structures whose underlying class of graphs has an excluded minor.

Corollary 28 *Let D be a* PTIME*-decidable class of graphs with an excluded minor and $P \subseteq STR \times \Pi^*$ a parameterized problem that is slicewise* FO*-definable. Then $P|_{STR[D]}$ is in* FPT.

7 A logical characterization of fixed-parameter tractability

In this section we give a characterization of FPT in the spirit of descriptive complexity theory.

We briefly review some facts from this area (see [13, 22] for details). It is common in descriptive complexity theory to identify decision problems, usually modeled by languages $L \subseteq \Sigma^*$ for a finite alphabet Σ, with classes of finite structures. More precisely, one identifies problems with classes of *ordered finite structures*. An *ordered structure* is a structure whose vocabulary contains the binary relation symbol $\leq$, and this symbol is interpreted as a linear order

of the universe. *ORD* denotes the class of all ordered structures. In this section, τ always denotes a vocabulary that contains $\leq$.

One of the most important results in descriptive complexity theory is the Immerman-Vardi Theorem [21, 30] saying that a class of ordered structures is in PTIME if, and only if, it is definable in least-fixed point logic FO(LFP). More concisely,

$$\text{PTIME} = \text{FO}(\text{LFP}).$$

We prove a similar result characterizing the class FPT in terms of the finite variable least fixed-point logics LFP^s, for $s \geq 1$, which were introduced by Kolaitis and Vardi [23]. Analogously to the classical setting we model parameterized problems by subsets of $\mathit{ORD}[\tau] \times \mathbb{N}$, for some τ.

Theorem 29 *A parameterized problem $P \subseteq \mathit{ORD}[\tau] \times \mathbb{N}$ is in* FPT *if, and only if, there is an $s \geq 1$ such that P is slicewise* LFP^s*-definable. More concisely, we may write*

$$\text{FPT} = \bigcup_{s \geq 1} \text{slicewise-LFP}^s.$$

In the proof of this result we assume that the reader is familiar with descriptive complexity theory, in particular with least fixed-point logic and the proof of the Immerman-Vardi Theorem. Those who are not may safely skip the rest of this section.

We first recall the definition of LFP^s: In the terminology of [13] (p. 174), LFP^s-sentences are FO(LFP)-sentences in the form

$$\exists \bar{y} [\text{S–LFP}_{\bar{x}_1, X_1, \ldots, \bar{x}_m, X_m} \varphi_1, \ldots, \varphi_m] \bar{y},$$

where $\varphi_1, \ldots, \varphi_m$ are first-order formulas with at most s individual variables. (That is, LFP^s-sentences are existential closures of simultaneous fixed-points over FO^s-formulas.)

We use the following two facts, the first implicit in [31] and the second in the proof of the Immerman-Vardi Theorem (cf. [13]). Fix τ and $s \in \mathbb{N}$ and let n always denote the size of the input structure.

(1) There is a computable function that associates an $O(n^{2s})$-algorithm $\mathbb{A}_\varphi$ with each $\varphi \in \text{LFP}^s[\tau]$ such that $\mathbb{A}_\varphi$ accepts a structure $\mathcal{A} \in \mathit{ORD}[\tau]$ if, and only if, $\mathcal{A}$ satisfies φ.

(2) There is a $t \in \mathbb{N}$ and a computable function that associates with every $O(n^s)$-algorithm $\mathbb{A}$ accepting a class $C \subseteq \mathit{ORD}[\tau]$ a sentence $\varphi_{\mathbb{A}} \in \text{LFP}^t[\tau]$ such that a τ-structure $\mathcal{A}$ satisfies $\varphi_{\mathbb{A}}$ if, and only if, $\mathcal{A} \in C$.

There is a slight twist in (2). When proving it one usually assumes that all structures are sufficiently large, in particular larger than the constant hidden in $O(n^s)$. This way it can be assumed that the algorithm is actually an n^{s+1}-algorithm (without any hidden constants.) Then one argues that small structures are no problem because they can be described up to isomorphism in first-order logic. When restricting the number of variables, one has to be careful with such an argument. Luckily, we are safe here because we only consider ordered structures, and there is a $t \in \mathbb{N}$ (depending on τ) such that every structure $\mathcal{A} \in \mathit{ORD}[\tau]$ can be characterized up to isomorphism by an FO^t-sentence.

Proof (of Theorem 29): For the backward direction, suppose that $P \subseteq \mathit{ORD}[\tau] \times \mathbb{N}$ is slicewise LFP^s-definable via $\delta : \mathbb{N} \to \text{LFP}^s$. Then Algorithm 3 shows that P is in FPT. The crucial fact is that Lines 1 and 2 do not depend on the input structure $\mathcal{A}$ and Line 3 requires time $O(n^{2s})$.

For the forward direction, suppose that $P \subseteq \mathit{ORD}[\tau] \times \mathbb{N}$ is in FPT. Choose $f : \mathbb{N} \to \mathbb{N}$, $c \in \mathbb{N}$ and an algorithm $\mathbb{A}$ deciding P in time $f(k) \cdot n^c$. The algorithm $\mathbb{A}$ gives rise to a sequence $\mathbb{A}_k$ ($k \geq 1$) of algorithms, where $\mathbb{A}_k$ decides the class $\{\mathcal{A} \mid (\mathcal{A}, k) \in P\} \subseteq \mathit{ORD}[\tau]$ in time $O(n^c)$. Then (2) yields the desired slicewise definition of P. □

8 Beyond FPT

In this last section we discuss how logical definability is related to the classes of the W-hierarchy. We introduce another hierarchy of parameterized problems, which we call the A-hierarchy, in terms of model-checking problems

DECIDE-P($\mathcal{A} \in$ *ORD*, $k \in \mathbb{N}$)

1 compute $\delta(k) \in \mathrm{LFP}^s$
2 compute $\mathbb{A}_{\delta(k)}$ (cf. (1))
3 simulate $\mathbb{A}_{\delta(k)}$ on input $\mathcal{A}$
4 **if** $\mathbb{A}_{\delta(k)}$ accepts $\mathcal{A}$
5 **then** accept
6 **else** reject.

Algorithm 3

for first-order logic and show that the A-hierarchy can be seen as a parametric analogue of the polynomial hierarchy. We then discuss the relation between the A-hierarchy and the W-hierarchy.

Our treatment is motivated by the following two results. They relate the class $\mathrm{W}[1]$ to model-checking and computations of non-deterministic Turing machines, respectively. Recall that $MC(\Sigma_1[s])$ denotes the parameterized model-checking problem for existential formulas in prenex normal form whose vocabulary contains at most s-ary relation symbols.

Theorem 30 (Downey, Fellows, Regan [11]) *$MC(\Sigma_1)|_{GRAPH}$ is $\mathrm{W}[1]$-complete under parameterized m-reductions.*

Proof: We prove that *CLIQUE* $\equiv^{\mathrm{fp}}_{\mathrm{m}}$ $MC(\Sigma_1)|_{GRAPH}$.

CLIQUE $\leq^{\mathrm{fp}}_{\mathrm{m}}$ $MC(\Sigma_1)|_{GRAPH}$ follows from the fact that *CLIQUE* is slicewise Σ_1-definable, so we only have to prove the converse.

An *atomic k-type* (in the theory of graphs) is a sentence $\theta(x_1, \ldots, x_k)$ of the form $\bigwedge_{1 \leq i < j \leq k} \alpha_{ij}(x_i, x_j)$, where $\alpha_{ij}(x_i, x_j)$ is either $x_i = x_j$ or Ex_ix_j or $(\neg Ex_ix_j \wedge \neg x_i = x_j)$ (for $1 \leq i < j \leq k$).

It is easy to see that there is a computable mapping f that associates with every EFO-sentence φ a sentence $\tilde{\varphi}$ of the form

$$\bigvee_{i=1}^{l} \exists x_1 \ldots \exists x_k \theta_i(x_1, \ldots, x_k), \tag{5}$$

where each θ_i is an atomic k-type, such that for all graphs $\mathcal{G}$ we have $\mathcal{G} \models \varphi \iff \mathcal{G} \models \tilde{\varphi}$.

For each graph $\mathcal{G}$ and each atomic k-type $\theta(\bar{x}) = \bigwedge_{1 \leq i < j \leq k} \alpha_{ij}(x_i, x_j)$ we define a graph $h(\mathcal{G}, \theta)$ as follows:

- The universe of $h(\mathcal{G}, \theta)$ is $\{1, \ldots, k\} \times G$.
- There is an edge between (i, v) and (j, w), for $1 \leq i < j \leq k$ and $v, w \in G$, if $\mathcal{G} \models \alpha_{ij}(v, w)$.

Then $h(\mathcal{G}, \theta)$ contains a k-clique if, and only if, $\mathcal{G} \models \exists \bar{x} \theta(\bar{x})$. Now we are ready to define the reduction from $MC(\Sigma_1)|_{GRAPH}$ to *CLIQUE*. Given an instance $(\mathcal{G}, \varphi)$ of $MC(\Sigma_1)|_{GRAPH}$, we first compute the sentence

$$\tilde{\varphi} = \bigvee_{i=1}^{l} \exists x_1 \ldots \exists x_k \theta_i.$$

We let $\mathcal{G}'$ be the disjoint union of the graphs $h(\mathcal{G}, \theta_i)$ for $1 \leq i \leq l$. Then $\mathcal{G}'$ has a k-clique if, and only if, $\mathcal{G} \models \varphi$. $\square$

Theorem 31 (Cai, Chen, Downey, and Fellows [4]) *The parameterized problem SHORT TURING MACHINE ACCEPTANCE (NM)*[4] *is $\mathrm{W}[1]$-complete, where*

[4] *NM* stands for nondeterministic Turing machine. This notation should be seen in connection with the AM_t below, which refers to alternating Turing machines.

NM		
	Input:	A non-deterministic Turing machine M.
	Parameter:	$k \in \mathbb{N}$.
	Question:	Does M accept the empty word in at most k steps?

Downey and Fellows call Theorem 31 a parameterized "analog of Cook's Theorem". In our notation, we may write $[NM]^{\text{fp}}_{\text{m}} = \text{W}[1]$. It is now very natural to define a "parameterized analogue of the polynomial hierarchy", which we call the A-*hierarchy*, by letting $\text{A}[t] := [AM_t]^{\text{fp}}_{\text{m}}$ for all $t \geq 1$, where

AM_t		
	Input:	An alternating Turing machine M whose initial state is existential.
	Parameter:	$k \in \mathbb{N}$.
	Question:	Does M accept the empty word in at most k steps with at most t alternations?

Note that $NM = AM_1$, thus $\text{W}[1] = \text{A}[1]$. Our following theorem can be seen as a natural generalization of Theorem 30.

Theorem 32 *For all $t \geq 1$, the problem $MC(\Sigma_t)|_{GRAPH}$ is $\text{A}[t]$-complete under parameterized m-reductions. Thus*

$$\text{A}[t] = [MC(\Sigma_t)|_{GRAPH}]^{\text{fp}}_{\text{m}} = \bigcup_{s \geq 1} [MC(\Sigma_t[s])]^{\text{fp}}_{\text{m}}.$$

Proof: The second equality follows from Corollary 11(3).

To prove that $MC(\Sigma_t)|_{GRAPH} \leq^{\text{fp}}_{\text{m}} AM_t$, we first observe that, for every graph $\mathcal{G}$ and every quantifier-free formula $\theta(x_1, \ldots, x_m)$, in time $p(||\theta||) \cdot |G|^2$, for a suitable polynomial p, we can construct a deterministic Turing machine $M(\mathcal{G}, \theta)$ with input alphabet G that accepts an input word $a_1 \ldots a_m$ over G if, and only if, $\mathcal{G} \models \theta(a_1, \ldots, a_m)$ and that performs at most $f(||\theta||)$ steps for some computable function $f : \mathbb{N} \to \mathbb{N}$. Just to give an example, to check whether $Ex_k x_l$ holds, say with $k < l$, we need states $s(i)$ for $1 \leq i \leq k$ and $s(i, a)$ for $k < i \leq l, a \in G$. The machine starts in state $s(1)$ with head in position 1 in state $s(1)$. It moves its head right until it reaches position k in state $s(k)$, reads a_k and goes to position $(k+1)$ in state $s(k+1, a_k)$. Then it moves right again until it reaches position l in state $s(l, a_k)$. From this state it can reach an accepting state if, and only if, $E^{\mathcal{G}} a_k a_l$.

Now suppose we are given an instance of $MC(\Sigma_t)|_{GRAPH}$, i.e. a graph $\mathcal{G}$ and a sentence

$$\varphi = \exists x_{11} \ldots \exists x_{1k_1} \forall x_{21} \ldots \forall x_{2k_2} \ \ldots \ Q x_{t1} \ldots Q x_{tk_t} \ \theta,$$

where θ is quantifier-free. Then the following alternating Turing machine $M(\mathcal{G}, \varphi)$ accepts the empty word if, and only if, $\mathcal{G} \models \varphi$: It first writes a sequence of elements of $\mathcal{G}$ on the tape using existential and universal states appropriately and then simulates $M(\mathcal{G}, \theta)$ on this input. Again $g(||\theta||)$, for some computable function g, is an upper bound for the number of steps $M(\mathcal{G}, \varphi)$ has to perform.

To finish the proof, by Corollary 11(3) it suffices to show that $\text{AM}_t \leq^{\text{fp}}_{\text{m}} MC(\Sigma_t[\tau])$ for a suitable vocabulary τ. To illustrate the idea, we first consider the case $t = 1$. Suppose we are given a nondeterministic Turing machine M with alphabet Σ, set Q of states, initial state q_0, accepting state q_{acc} and transition relation δ. Let $\Sigma_H := \{a_H \mid a \in \Sigma\}$, a_H coding the information that the head of M scans a cell containing a. Let $\tau := \{ST, AL, H, IN, ACC, R, L, S\}$ with unary ST, AL, H, IN, ACC and 4-ary R, L, S and let $\mathcal{A}_M$ be the τ-structure given by

$$\begin{array}{l}
A_M := Q \dot\cup \Sigma \dot\cup \Sigma_H; \\
ST^{\mathcal{A}_M} := Q; \quad AL^{\mathcal{A}_M} := \Sigma; \quad H^{\mathcal{A}_M} := \Sigma_H; \\
IN^{\mathcal{A}_M} := \{q_0\}; \quad ACC^{\mathcal{A}_M} := \{q_{\text{acc}}\}; \\
R^{\mathcal{A}_M} := \{(q, a_H, b, q') \mid (q, a, 1, b, q') \in \delta\}; \\
L^{\mathcal{A}_M} := \{(q, a_H, b, q') \mid (q, a, -1, b, q') \in \delta\}; \\
S^{\mathcal{A}_M} := \{(q, a_H, b_H, q') \mid (q, a, 0, b, q') \in \delta\} \cup \{(q_{\text{acc}}, a_H, a_H, q_{\text{acc}}) \mid a \in \Sigma\},
\end{array}$$

where $(q, a, h, b, q') \in \delta$ means: if M is in state q and its head scans $a \in \Sigma$, then M replaces a by b, moves its head one cell to the right ($h = 1$), to the left ($h = -1$), or does not move its head ($h = 0$); finally, it changes to state q'.

Let k be given as parameter for AM_1. In k steps, M scans at most the first k cells. The quantifier-free formula (note that the following formulas only depend on k and not on M)

$$\varphi_{\text{config}}(x, y_1, \ldots, y_k) := STx \wedge \bigvee_{i=1}^{k} (Hy_i \wedge \bigwedge_{j \neq i} ALy_j)$$

states that $(x, y_1, \ldots, y_k)$ is a configuration with state x, with the head facing y_i, and with y_j being the content of the jth cell ($j \neq i$). Let $\varphi_{\text{start}}(x, y_1, \ldots, y_k)$ be a quantifier-free formula stating that $(x, y_1, \ldots, y_k)$ is the starting configuration. Similarly, we define a quantifier-free formula $\varphi_{\text{step}}(x, y_1, \ldots, y_k, x', y'_1, \ldots, y'_k)$ stating that the configuration $(x', y'_1, \ldots, y'_k)$ is the successor configuration of $(x, y_1, \ldots, y_k)$ (we agree that each accepting configuration is its own successor).

Now, the equivalence

$$M \text{ stops in} \leq k \text{ steps} \quad \Longleftrightarrow \quad \mathcal{A}_M \models \varphi_k$$

holds for the existential sentence φ_k:

$$\begin{aligned}&\exists x_1 \exists y_{11} \ldots \exists y_{1k} \ \ldots \ \exists x_k \exists y_{k1} \ldots \exists y_{kk} (\varphi_{\text{start}}(x, y_{11}, \ldots, y_{1k}) \\ &\wedge \textstyle\bigwedge_{i=1}^{k-1} \varphi_{\text{step}}(x_i, y_{i1}, \ldots, y_{ik}, x_{i+1}, y_{i+11}, \ldots, y_{i+1k}) \wedge ACCx_k).\end{aligned}$$

For $t \geq 1$, we can proceed similarly, with the addition that universal quantifiers are needed to take care of universal states of the input machine. Now, τ also contains two further unary symbols F (for the existential states) and U (for the universal states) which get the corresponding interpretations in $\mathcal{A}_M$. For example, for $t = 2$, we can take a Σ_t-sentence7 equivalent to:

$$\begin{aligned}&\bigvee_{l \leq k} \exists x_1 \exists y_{11} \ldots \exists y_{1k} \ \ldots \ \exists x_k \exists y_{l1} \ldots \exists y_{lk} \Big(\\ &\quad \varphi_{\text{start}}(x, y_{11}, \ldots, y_{1k}) \\ &\wedge \textstyle\bigwedge_{i=1}^{l-1} \varphi_{\text{step}}(x_i, y_{i1}, \ldots, y_{ik}, x_{i+1}, y_{i+1\,1}, \ldots, y_{i+1\,k}) \\ &\wedge Fx_1 \wedge \ldots \wedge Fx_{l-1} \wedge Ux_l \\ &\wedge \forall x_{l+1} \forall y_{l+1\,1} \ldots \forall y_{l+1\,k} \ \ldots \ \forall x_k \forall y_{k1} \ldots \forall y_{kk} \big(\\ &\qquad \big(Ux_{l+1} \wedge \ldots \wedge Ux_{k-1} \\ &\qquad \wedge \textstyle\bigwedge_{i=l}^{k-1} \varphi_{\text{step}}(x_i, y_{i1}, \ldots, y_{ik}, x_{i+1}, y_{i+1\,1}, \ldots, y_{i+1\,k}) \big) \rightarrow ACCx_k) \big) \Big).\end{aligned}$$

(Without loss of generality we assume that the accepting state is both existential and universal and that at least one transition is always possible in a universal state). □

The following result due to Downey, Fellows, and Regan allows to compare the W- and the A-hierarchy. Let $t, u \geq 1$. A formula φ is $\Sigma_{t,u}$, if it is Σ_t and all quantifier blocks after the leading existential block have length $\leq u$.

Theorem 33 ([11]) *For all $t \geq 1$,*

$$\mathrm{W}[t] = \bigcup_{\substack{\tau \text{ vocabulary} \\ u \geq 1}} [MC(\Sigma_{t,u}[\tau])]^{\text{fp}}_{\text{m}} = \bigcup_{u \geq 1} [MC(\Sigma_{t,u})|_{GRAPH}]^{\text{fp}}_{\text{m}}.$$

Note that for $t = 1$, this is just Theorem 30. The crucial step in proving the theorem for $t \geq 2$ is to establish the W[t]-completeness of the problems *WEIGHTED MONOTONE t-NORMALIZED SATISFIABILITY* (for even t) and *WEIGHTED ANTIMONOTONE t-NORMALIZED SATISFIABILITY* (for odd $t \geq 3$). We refer the reader to [10] for the (difficult) proofs of these results. Once these basic completeness results are established, it is relatively easy to derive Theorem 33 (also cf. our proof of Theorem 36). We encourage the reader to give a purely "logical" proof of the second equality in the theorem.

Since $\Sigma_{1,u} = \Sigma_1$, we get

$$\mathrm{W}[1] = \bigcup_{\tau \text{ vocabulary}} [MC(\Sigma_1[\tau])]^{\mathrm{fp}}_{\mathrm{m}} = [MC(\Sigma_1)|_{GRAPH}]^{\mathrm{fp}}_{\mathrm{m}} = \mathrm{A}[1].$$

By Theorem 32, $\mathrm{W}[t] \subseteq \mathrm{A}[t]$ for all $t \geq 2$. The question whether $\mathrm{W}[t] = \mathrm{A}[t]$ for all t remains open; in view of Theorem 32 this question is equivalent to $\mathrm{W}[t] = [MC(\Sigma_t)|_{GRAPH}]^{\mathrm{fp}}_{\mathrm{m}}$ for all $t \geq 1$. In this form it is stated as an open problem in [11]. Consider, for example, the following parameterized problem

P_0		
	Input:	Graph $\mathcal{G}$.
	Parameter:	$(k, l) \in \mathbb{N}^2$.
	Question:	Are there $a_1, \ldots, a_k \in G$ such that every clique of size l contains an a_i?

Since P_0 is slicewise Σ_2-definable, we have $P_0 \in \mathrm{A}[2]$. But is P_0 in $\mathrm{W}[2]$?

In the definition of the W-hierarchy we can restrict the length of the non-leading quantifier-blocks to one:

Proposition 34 *For all $t \geq 1$,*

$$\mathrm{W}[t] = \bigcup_{\tau \text{ vocabulary}} [MC(\Sigma_{t,1}[\tau])]^{\mathrm{fp}}_{\mathrm{m}}.$$

Proof: The inclusion $\supseteq$ being trivial we turn to a proof of $\subseteq$: Fix τ and $u \geq 1$. We show that $MC(\Sigma_{t,u}[\tau]) \leq^{\mathrm{fp}}_{\mathrm{m}} MC(\Sigma_{t,1}[\tau'])$ for suitable τ'. The idea is to replace the blocks of at most u quantifiers by a single quantifier ranging over the set of u-tuples of a structure. To explain this idea we first use a vocabulary containing function symbols (and sketch afterwards how one can do without). Let $\tau' := \tau \cup \{T, p_1, \ldots, p_u\}$, where T is a unary relation symbol (for ordered u-tuples) and $p_1, \ldots, p_u$ are unary function symbols (the projection functions). Given a τ-structure $\mathcal{A}$ let $\mathcal{A}'$ be a τ'-structure with

$$A' := A \dot{\cup} A^u, \qquad T^{\mathcal{A}'} := A^u,$$

where for $(a_1, \ldots, a_u) \in A^u$, $p_i(a_1, \ldots, a_u) = a_i$ and where the relation symbols of τ are interpreted as in $\mathcal{A}$. Now, e.g. for

$$\varphi = \exists x_1 \ldots \exists x_k \forall y_1 \ldots \forall y_u \psi(\bar{x}, \bar{y})$$

with quantifier-free ψ, let

$$\varphi' = \exists x_1 \ldots \exists x_k \forall y (\neg T x_1 \wedge \ldots \wedge \neg T x_k \wedge (Ty \to \psi(\bar{x}, p_1(y) \ldots p_u(y)))).$$

Then,

$$\mathcal{A} \models \varphi \iff \mathcal{A}' \models \varphi',$$

which gives the desired parameterized m-reduction.

Let us explain, for the case $t = 2$, how to proceed to avoid function symbols. One has to add to τ, besides T as above, for every relation symbol $R \in \tau$, say r-ary, every subset $M \subseteq \{1, \ldots, r\}$, and every function $\rho : M \to \{1, \ldots, u\}$ a new relation symbol $R_{M,\rho}$; e.g., if $M = \{s, \ldots, r\}$ then $R^{\mathcal{A}'}_{M,\rho} a_1 \ldots a_{s-1} b$ if, and only if, $a_1 \ldots a_{s-1} \in A$, $b = (b_1, \ldots, b_u) \in A^u$, and $R^{\mathcal{A}} a_1 \ldots a_{s-1} b_{\rho(s)} \ldots b_{\rho(r)}$; and a subformula $R x_{i_1} \ldots x_{i_{s-1}} y_{\rho(s)} \ldots y_{\rho(r)}$ of φ is replaced by $R_{M,\rho} x_{i_1} \ldots x_{i_{s-1}} y$. □

Downey, Fellows, and Regan [11] also gave a (much simpler) characterization of the W-hierarchy in terms of Fagin-definability. We find it worthwhile to sketch a short proof of this result. For a class Φ of formulas we let $\mathrm{FD}(\Phi)$ be the class of all problems that are Φ-Fagin-definable. Let $\Pi_t[s]$ denote the class of all Π_t-formulas whose vocabulary is at most s-ary.

Theorem 35 ([11]) *For all $t \geq 1$ we have* $\mathrm{W}[t] = [\mathrm{FD}(\Pi_t[2])]^{\mathrm{fp}}_{\mathrm{m}} = [\mathrm{FD}(\Pi_t)]^{\mathrm{fp}}_{\mathrm{m}}$.

Proof: Recall that $\mathrm{W}[t] = \bigcup_{d\geq 1}[\mathit{WSAT}(C_{t,d})]^{\mathrm{fp}}_{\mathrm{m}}$, where $C_{t,d}$ is the class of all propositional formulas of the form

$$\bigwedge_{i_1}\bigvee_{i_2}\cdots\Big(\bigwedge_{i_t}/\bigvee_{i_t}\Big)\,\varphi_{i_1\ldots i_t} \tag{6}$$

where the $\varphi_{i_1\ldots i_t}$ are small formulas of depth at most d.

To prove that $\mathrm{W}[t] \subseteq [\mathrm{FD}(\Pi_t[2])]^{\mathrm{fp}}_{\mathrm{m}}$, we first transform a propositional formula φ of the form (6) into a propositional formula φ' of essentially the same form, but with all the $\varphi_{i_1\ldots i_t}$ being disjunctions (if t is even) or conjunctions (if t is odd) of exactly d' literals, for some constant d' only depending on d. This can be done by first transforming the small formulas into equivalent formulas in conjunctive normal form or disjunctive normal form, respectively, and then repeatedly replacing disjunctions (conjunctions, respectively) γ with less than the maximum number of literals by the two clauses $\gamma \vee X$ and $\gamma \vee \neg X$ ($\gamma \wedge X$ and $\gamma \wedge \neg X$, respectively), for some variable X not appearing in γ.

We associate with φ' an $\{E, P, N, T, L\}$-structure $\mathcal{C}$ which is obtained from the tree corresponding to φ' as follows: We first remove the root. Then we identify all leaves corresponding to the same propositional variable. To indicate whether a variable occurs positively or negatively in a clause, we use the binary relations P and N. The unary relation T contains all the top level nodes, and the unary relation L contains all the (former) leaves. It is easy to write a Π_t-formula $\psi'(X)$ such that φ' has a satisfying assignment of weight k if, and only if, there exists a k-element subset $B \subseteq C$ such that $\mathcal{C} \models \psi'(B)$.

This can best be illustrated with a simple example: Let

$$\varphi' := (X \vee Y \vee Z) \wedge (X \vee \neg Y \vee Z) \wedge (X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Y \vee \neg Z).$$

The corresponding structure $\mathcal{C}$ is displayed in Figure 1.

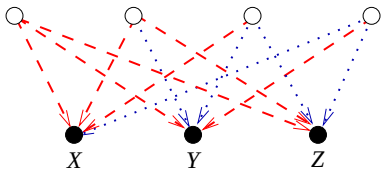

Figure 1.

We let

$$\psi'(X) := \forall x(Xx \to Lx) \wedge \forall x\forall y_1\forall y_2\forall y_3\Big(\big(Tx \wedge \bigwedge_{i=1}^{3} Exy_i \wedge \bigwedge_{1\leq i<j\leq 3} y_i \neq y_j\big) \longrightarrow \bigvee_{i=1}^{3}\big((Pxy_i \wedge Xy_i) \vee (Nxy_i \wedge \neg Xy_i)\big)\Big).$$

To prove $[\mathrm{FD}(\Pi_t)]^{\mathrm{fp}}_{\mathrm{m}} \subseteq \mathrm{W}[t]$, we just note that for every formula $\psi(X) \in \Pi_t$ and every structure $\mathcal{A}$, there is a $C_{t,d}$-formula φ with the property that every assignment α for φ corresponds to a set B, whose size is the weight of the assignment, such that α satisfies φ if, and only if $\mathcal{A} \models \psi(B)$. Here d is a constant that just depends on ψ. Furthermore, the transformation $(\mathcal{A}, \psi) \mapsto \varphi$ is computable in time polynomial in $\mathcal{A}$. □

We do not know of any simple proof of the equivalence between the two characterizations of the W-hierarchy in terms of slicewise $\Sigma_{t,u}$-definability (cf. Theorem 33) and Π_t-Fagin definability. While the proof of the previous theorem shows that Π_t-Fagin definability is actually quite close to the definition of $\mathrm{W}[t]$ in terms of the weighted satisfiability problem for C_t-formulas, it seems that it is a significant step to get from there to slicewise $\Sigma_{t,u}$-definability.

Our last result is another characterization of the W-hierarchy in terms of Fagin definability that is much closer to the slicewise characterization. A first-order formula $\varphi(X)$ is *bounded to* the r-ary relation variable X, if in $\varphi(X)$ quantifiers appear only in the form $\exists x_1 \ldots \exists x_r(Xx_1 \ldots x_r \wedge \psi)$ or $\forall x_1 \ldots \forall x_r(Xx_1 \ldots x_r \rightarrow \psi)$, which we abbreviate by $\exists \bar{x} \in X\ \psi$ and $\forall \bar{x} \in X\ \psi$, respectively. For $t \geq 1$ we let Π_t^b be the class of all formulas $\varphi(X)$ of the form

$$\forall \bar{x}_1 \exists \bar{x}_2 \ldots Q \bar{x}_t \theta$$

where $Q = \forall$ if t is odd and $Q = \exists$ otherwise and where θ is bounded to X.

For example, *CLIQUE* is Fagin-defined by the Π_0^b-formula

$$\forall x \in X \forall y \in X(x \neq y \rightarrow Exy)$$

and *DOMINATING SET* by the Π_1^b-formula

$$\forall x \exists y \in X(x = y \vee Exy).$$

Theorem 36 *For $t \geq 1$, $\mathrm{W}[t] = [\mathrm{FD}(\Pi_{t-1}^b)]_{\mathrm{m}}^{\mathrm{fp}}$.*

Proof: First, assume that the problem $P \subseteq \mathit{STR}[\tau] \times \mathbb{N}$ is Fagin-defined by $\varphi(X) \in \Pi_{t-1}^b$, say

$$\varphi(X) := \forall \bar{y}_1 \exists \bar{y}_2 \forall \bar{y}_3 \ldots Q \bar{y}_t \psi,$$

where ψ only contains bounded quantifiers. Let l be the maximum of the lengths of the tuples $\bar{y}_i$, for $1 \leq i \leq t$. For simplicity, let us assume that X is unary. Since Xy is equivalent to $\exists z \in X\, z = y$, we can assume that in ψ, the variable X only occurs in quantifier bounds. We show that $P \in \mathrm{W}[t]$. Given a parameter k set (with new variables $x_1, \ldots, x_k$)

$$\varphi^k := \exists x_1 \ldots \exists x_k \forall \bar{y}_1 \exists \bar{y}_2 \forall \bar{y}_3 \ldots Q \bar{y}_t (\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \psi^*),$$

where ψ^* is obtained from ψ by inductively replacing $\forall u \in X \chi(u)$ and $\exists u \in X\ \chi(u)$ by $\bigwedge_{i=1}^k \chi(x_i)$ and $\bigvee_{i=1}^k \chi(x_i)$, respectively. Note that φ^k is a $\Sigma_{t,l}$-formula and that for every structure $\mathcal{A}$,

$$(\mathcal{A}, k) \in P \iff \mathcal{A} \models \varphi^k.$$

Thus, P is slicewise $\Sigma_{t,l}$-definable and hence in $\mathrm{W}[t]$.

For the converse direction, we prove that for all $t, u \geq 1$ the problem $\mathit{MC}(\Sigma_{t,u})|_{\mathit{GRAPH}}$ is in $\mathrm{FD}(\Pi_{t-1}^b)$. The idea of this reduction is to associate with every graph $\mathcal{G}$ and $\Sigma_{t,u}$-formula $\exists x_1 \ldots \exists x_k \varphi$ a structure $\mathcal{C}$ which essentially is a Boolean circuit whose satisfying assignments of size k correspond to assignments to the variables $x_1, \ldots, x_k$ such that $\mathcal{G}$ satisfies φ. We can Fagin-define the weighted satisfiability problem for this circuit by a Π_{t-1}^b-formula. We leave the details to the reader.

For readers familiar with [10] (Theorem 12.6 on page 299 is the relevant result), we state another proof. For $t = 1$, the result follows from the fact the $\mathrm{W}[1]$-complete problem *CLIQUE* is Π_0^b-Fagin definable. For odd $t \geq 2$, the problem *WEIGHTED ANTIMONOTONE t-NORMALIZED SATISFIABILITY* is $\mathrm{W}[t]$-complete. It is parameterized m-reducible to the problem Fagin-defined by the Π_{t-1}^b-formula

$$\begin{aligned}\varphi(X) :=& \forall y_0 \forall y_1 \exists y_2 \forall y_3 \ldots \exists y_{t-1} \\ & ((Ey_0y_1 \wedge Ey_1y_2 \wedge \ldots \wedge Ey_{t-2}y_{t-1}) \rightarrow \forall x \in X \neg Ey_{t-1}x).\end{aligned}$$

For even t we use the completeness of *WEIGHTED MONOTONE t-NORMALIZED SATISFIABILITY* and argue similarly. □

Remark 37 Downey, Fellows and Taylor [12] proved that the parameterized model-checking problem for full first-order logic is complete for the class $\mathrm{AW}[*]$, a parameterized complexity class above the W-hierarchy that is defined in terms of the satisfiability problem for quantified Boolean formulas.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42:844–856, 1995.

[3] L. Cai and J. Chen. On fixed-parameter tractability and approximability of NP optimization problems. *Journal of Computer and System Sciences*, 54:465–474, 1997.

[4] L. Cai, J. Chen, R.G. Downey, and M.R. Fellows. On the parameterized complexity of short computation and factorization. *Archive for Mathematical Logic*, 36:321–337, 1997.

[5] 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.

[6] 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.

[7] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, pages 194–242. Elsevier Science Publishers, 1990.

[8] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24:873–921, 1995.

[9] 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.

[10] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[11] R.G. Downey, M.R. Fellows, and K. Regan. Descriptive complexity and the W-hierarchy. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetic*, volume 39 of *AMS-DIMACS Volume Series*, pages 119–134. AMS, 1998.

[12] R.G. Downey, M.R. Fellows, and U. Taylor. The parameterized complexity of relational database queries and an improved characterization of $W[1]$. In Bridges, Calude, Gibbons, Reeves, and Witten, editors, *Combinatorics, Complexity, and Logic – Proceedings of DMTCS '96*, pages 194–213. Springer-Verlag, 1996.

[13] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[14] R. Fagin. Generalized first–order spectra and polynomial–time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings, Vol. 7*, pages 43–73, 1974.

[15] M.R. Fellows and U. Stege. An improved fixed-parameter-tractable algorithm for vertex cover. Technical Report 318, Department of Computer Science, ETH Zurich, 1999.

[16] J. Flum, M. Frick, and M. Grohe. Query evaluation via tree-decompositions. In J. van den Bussche and V. Vianu, editors, *Proceedings of the 8th International Conference on Database Theory*, volume 1973 of *Lecture Notes in Computer Science*, pages 22–38. Springer Verlag, 2001.

[17] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable graphs. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Proceedings of the 26th International Colloquium on Automata, Languages and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 331–340. Springer-Verlag, 1999.

[18] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. In *Proceedings of the 18th ACM Symposium on Principles of Database Systems*, pages 21–32, 1999.

[19] M. Grohe. Local tree-width, excluded minors, and approximation algorithms. To appear in *Combinatorica*.

[20] 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*, 2001. To appear.

[21] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[22] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[23] Ph. G. Kolaitis and M. Y. Vardi. On the expressive power of variable-confined logics. In *Proceedings of the 11th IEEE Symposium on Logic in Computer Science*, 1996.

[24] Ph.G. Kolaitis and M.N. Thakur. Approximation properties of NP minimization classes. *Journal of Computer and System Sciences*, 50:391–411, 1995.

[25] 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.

[26] C.H. Papadimitriou and M. Yannakakis. On the complexity of database queries. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, pages 12–19, 1997.

[27] 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.

[28] N. Robertson and P.D. Seymour. Graph minors XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.

[29] D. Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6:505–526, 1996.

[30] M.Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 137–146, 1982.

[31] M.Y. Vardi. On the complexity of bounded-variable queries. In *Proceedings of the 14th ACM Symposium on Principles of Database Systems*, pages 266–276, 1995.