Electronic Colloquium on Computational Complexity, Report No. 131 (2009)

Parameterized Complexity of First-Order Logic

Anuj Dawar
Cambridge University
Computer Laboratory
anuj.dawar@cl.cam.ac.uk

Stephan Kreutzer
Oxford University
Computing Laboratory
kreutzer@comlab.ox.ac.uk

2nd December 2009

Abstract

We show that if $\mathcal{C}$ is a class of graphs which is nowhere dense then first-order model-checking is fixed-parameter tractable on $\mathcal{C}$. As all graph classes which exclude a fixed minor, or are of bounded local tree-width or locally exclude a minor are nowhere dense, this generalises algorithmic meta-theorems obtained for these classes in the past (see [11, 13, 4]).

Conversely, if $\mathcal{C}$ is not nowhere dense and in addition is closed under taking sub-graphs and satisfies some effectivity conditions then FO-model checking is not FPT on $\mathcal{C}$ unless FPT = AW[$*$].

Hence, for classes of graphs closed under sub-graphs, this essentially gives a precise characterisation of classes for which FO model-checking is tractable.

However, our result generalises to much more general classes of graphs. In particular we show that every class which can efficiently be coloured *over a class with the type representation property* allows tractable first-order model-checking. Such classes include all classes which are nowhere dense and also all classes of bounded clique-width. This result therefore unifies all known meta-theorems for first-order logic.

1 Introduction

In 1990, Courcelle [1] proved his celebrated theorem stating that every graph property definable in *Monadic Second-Order Logic* (MSO_2) can be decided in linear time on all graph classes of bounded tree width. Results of this form are commonly referred to as *algorithmic meta-theorems*. In their most general form they are results of the form "every computational problem which can be defined in a given logic $\mathcal{L}$ can be solved efficiently on every class $\mathcal{C}$ of graphs satisfying certain conditions". This formulation highlights the motivation for these results from an algorithmic point of view which we will describe in more detail below. Alternatively we can formulate algorithmic meta-theorems in the language of logic and parameterized complexity as "the model-checking problem for the logic $\mathcal{L}$ is fixed-parameter tractable on every class $\mathcal{C}$ satisfying certain conditions". See Section 2 for details on parameterized complexity. In the following we will use this logical formulation of the results we are after.

ISSN 1433-8092

Following Courcelle's theorem, much work has gone into establishing further meta-theorems for variants of monadic second-order logic and for first-order logic. Courcelle, Makowski and Rotics [2] showed that MSO_1, a variant of monadic second-order logic without quantification over sets of edges, is fixed-parameter tractable by linear time parameterized algorithms. For first-order logic, Seese [22] proved that first-order model-checking is fixed-parameter tractable on graph classes of bounded maximum degree. This was later generalised by Frick and Grohe [13] to graph classes of *bounded local tree-width*, which includes the class of planar graphs, and by Flum and Grohe [11] to graph classes excluding a fixed minor. Graph classes excluding a minor and graph classes of bounded local tree-width are incomparable concepts. In [4], therefore, Dawar, Grohe and Kreutzer introduced a new concept of graph classes *locally excluding a minor*, which strictly generalises both excluded minors and bounded local tree-width, and showed that first-order model-checking is fixed-parameter tractable on all graph classes locally excluding a minor. With the exception of bounded local clique-width, this is the most general meta-theorem for first-order logic known so far. See [14, 17] for recent surveys on algorithmic meta-theorems.

The study of algorithmic meta-theorems is of interest to both logic and algorithmic graph theory. An important task in the theory of graph algorithms is to find feasible instances of otherwise intractable algorithmic problems. For this purpose, concepts originating in graph structure theory such as bounded tree-width or excluding a minor have proved to be extremely useful and many NP-complete problems become tractable on graph classes whose tree-width is bounded by a fixed constant or which exclude a fixed minor. Studying, for instance, the methods used to prove that many problems become tractable on graph classes of bounded tree width shows that many algorithms are based on a similar technique and it is therefore a natural question to ask how far these algorithmic techniques range. On the other hand, it is interesting to investigate which types of problems become tractable when the tree width is bounded. Algorithmic meta-theorems provide elegant answers to these questions in that they establish tractability results for a very large and natural class of problems. For instance, the aforementioned result by Flum and Grohe shows that all first-order problems are tractable on all graph classes excluding a fixed minor and it also shows that all these problems can be solved by similar algorithmic techniques.

From a logical perspective, tractability results on specific classes of graphs yield interesting new insights into the complexity of commonly used logics such as first-order or monadic second-order logic with potential applications in the design and analysis of query- or specification languages based on these logics. For instance, it has long been realised that monadic second-order logic is particularly well-behaved on trees, witnessed by the closed connection between MSO and automata on trees. This realisation has been extremely fruitful in the study of query languages for XML data, databases whose skeletons are trees. It is likely that in a similar way a better understanding of the structure and type of classes of graphs on which first-order model-checking is tractable would have interesting implications for the design and analysis of languages based on first-order concepts.

Much effort therefore has gone into establishing more and more general meta-theorems, i.e. to find more and more general classes of graphs and graph structural concepts for which meta-theorems can be established. Ideally, we aim for a precise characterisation of classes of graphs for which first-order model-checking becomes tractable. That is, we aim at identifying a property $\mathcal{P}$ of graph classes such that first-order model-checking is fixed-parameter tractable on a class $\mathcal{C}$ of graphs if, and only if, $\mathcal{C}$ has property $\mathcal{P}$. Clearly, with today's technology this can only be achieved with respect to common assumptions in complexity theory.

In this paper we introduce a new technique for obtaining algorithmic meta-theorems for first-order logic on specific classes of graphs which is based on ideas of low tree-width colourings as studied in [6, 20]. Previous meta-theorems for first-order logic have, in one way or another, mostly been based on the idea of decomposition a graph recursively into almost disjoint sub-graphs of simpler structure than the original graph. Our technique instead is based on the following idea. Given a graph G and a formula φ of quantifier-rank q we partition G into a number of disjoint sets, i.e. we colour it by a number of colours, such that any q colours together induce a sub-graph which is structurally much simpler than the original graph. If all these sub-graphs have a property that we call the *type representation property*, then this will allow us to efficiently compute for each of these sub-graphs a small piece of information which, combining it for all such sub-graphs, will allow us to determine whether the formula φ is true in G. Hence, to show that first-order model-checking is tractable for a class $\mathcal{C}$ of graphs we need to show that every graph $G \in \mathcal{C}$ can be coloured in a way that any small number of colours induce a graph with the *type representation property*. This is the main technical result of this paper.

Theorem. (Theorem 5.4) For each $r \geq 0$ let $\mathcal{C}_r$ be a class of graphs with the type representation property and let $\mathcal{C} := (\mathcal{C}_r)_{r\geq 0}$. If $\mathcal{D}$ is efficiently colourable over $\mathcal{C}$ then $\mathrm{MC}(\mathrm{FO}, \mathcal{D}) \in \mathrm{FPT}$.

As the most important application of our technique we show that the class of graphs of *tree depth* of most k (see below for details) has the type representation property. In [20], Nešetřil and Ossona de Mendez introduce the concept of graph classes which are *nowhere dense* and show that any such class $\mathcal{C}$ allows small tree depth colouring, i.e. for each $G \in \mathcal{C}$ we can compute efficiently a colouring of G with not too many colours such that any q colours induce a sub-graph of tree depth at most q. The low tree-depth colouring of nowhere dense classes of graphs has been used in [20] to obtain several algorithmic applications, for instance in relation for finding homomorphisms. A different techniques for establish parameterized algorithms for problems such as variants of the dominating set, the independent set of clique problem has been established in [5].

Low tree-depth colouring of nowhere dense classes will allow us to apply our method above to show that first-order logic is fixed-parameter tractable on any class of graphs which is nowhere dense. The concept of nowhere dense classes properly extends classes locally excluding a minor and in this sense our result implies the most general meta-theorem for first-order logic (with the exception of clique-width). But we can show even more. For classes of graphs closed under sub-graphs we can prove a corresponding hardness result for graph classes which are not nowhere dense and thereby give a precise characterisation of the sub-graph closed graph classes on which first-order logic is tractable. To the best of our knowledge this is the first time that such an exact characterisation of tractability has been established within a natural class of graph classes such as those closed under sub-graphs. One of the main results of this paper, therefore, is the following (see below for details).

Theorem. (Corollary 5.7 and Theorem 6.1) Let $\mathcal{C}$ be a class of graphs.

1. If $\mathcal{C}$ is nowhere dense, then $\text{MC}(\text{FO}, \mathcal{C})$ is fixed-parameter tractable. For every $\varepsilon > 0$, the running time of the algorithm for deciding whether a formula φ is true in a graph $G \in \mathcal{C}$ can be bounded by $f(|\varphi|) \cdot n^{1+\varepsilon}$, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function.

2. Otherwise, if $\mathcal{C}$ is effectively not nowhere dense and closed under sub-graphs, then $\text{MC}(\text{FO}, \mathcal{C})$ is not fixed-parameter tractable unless FPT = AW[$*$].

The running time stated in the previous theorem matches the running time achieved by Frick and Grohe for first-order model-checking on graph classes of bounded local tree width (which this result generalises) and improves significantly on the running time achieved in [11, 4] for graph classes excluding a minor or locally excluding a minor.

In [19], Nešetřil and Ossona de Mendez introduce the concept of graph classes of bounded expansion. Every class exluding a minor has bounded expansion and every class of bounded expansion is nowhere dense. But for graph classes of bounded expansion we can show the following

Corollary. (Corollary 5.8) First-order model-checking is fixed-parameter tractable by linear time parameterized algorithms on any class of graphs of bounded expansion (and hence on classes which exclude a fixed minor).

The previous result gives a strong tractability result for first-order logic on very large classes of graphs. Furthermore, for classes closed under sub-graphs it characterises (essentially) exactly the classes of graphs on which first-order model-checking is tractable. However, our method is more general and also applies to further classes of graphs which are no longer sparse. In particular, we show that the class of graphs of *clique-width* $\leq k$ has the type representation property and therefore every class that can be coloured over classes of bounded clique-width allow tractable first-order model-checking.

Theorem. (Corollary 7.2) For each $r \geq 0$ let $\mathcal{C}_r$ be the class of graphs of clique-width at most r and let $\mathcal{C} := (\mathcal{C}_r)_{r \geq 0}$. If $\mathcal{D}$ is efficiently colourable over $\mathcal{C}$ then $\text{MC}(\text{FO}, \mathcal{D}) \in \text{FPT}$.

This includes all classes of bounded clique-width and all classes which are nowhere dense and hence unifies all known meta-theorems for first-order logic. But as explained below, this includes graph classes which have unbounded clique-width and are not nowhere dense and thereby establishes new tractability results going beyond bounded clique-width and nowhere denseness.

Organisation. We find it illustrative to present our method for obtaining meta-theorems together with the application of this method to graph classes which are nowhere dense. The paper is therefore organised as follows. In Section 2 we present notation and concepts used throughout the paper. In Section 3 we present the concept of tree-depth and nowhere dense classes of graphs. In Section 4 we introduce the concept of *type representation schemes* and illustrate it by showing that classes of graphs of bounded tree-depth allow such schemes.

In Section 5 we introduce the concept of $\mathcal{C}$-colourings and the general method for obtaining meta-theorems and illustrate it by showing that first-order model-checking is fixed-parameter tractable on nowhere dense classes of graphs.

In Section 6, we show that first-order model-checking is not fixed-parameter tractable on every class of graphs closed under sub-structures which is not nowhere dense, under some further assumptions. Finally, in Section 7 we show our results on graph classes with low clique-width colouring.

2 Preliminaries

Our graph theoretical notation follows [7]. In particular, if G is a graph we refer to its set of vertices by $V(G)$ and to its set of edges by $E(G)$. All graphs in this paper are undirected and simple, i.e. without self-loops. A *colouring* of a graph G is an assignment of colours to the vertices of G. A colouring is *proper* if whenever $\{u, v\} \in E(G)$, then u and v are assigned different colours.

We refer to [10, 9] for background on logic. The complexity theoretical framework we use in this paper is *parameterized complexity*. See [8, 12] for details. Let $\mathcal{C}$ be a class of coloured graphs. The *parameterized model-checking problem* $\text{MC}(\text{FO}, \mathcal{C})$ for first-order logic (FO) on $\mathcal{C}$ is defined as the problem to decide, given $G \in \mathcal{C}$ and $\varphi \in \text{FO}$, if $G \models \varphi$. The *parameter* is $|\varphi|$. $\text{MC}(\text{FO}, \mathcal{C})$ is *fixed-parameter tractable* (fpt), if for all $G \in \mathcal{C}$ and $\varphi \in \text{FO}$, $G \models \varphi$ can be decided in time $f(|\varphi|) \cdot |G|^c$, for some computable function $f : \mathbb{N} \to \mathbb{N}$ and $c \in \mathbb{N}$. The class FPT is the class of all problems which are fixed-parameter tractable. In parameterized complexity theory it plays a similar role to polynomial time in classical complexity theory. The role of NP as a witness for intractability is played by a class called W[1] and it is a standard assumption in parameterized complexity theory that FPT $\neq$ W[1], similar to P $\neq$ NP in classical complexity. It has been shown that $\text{MC}(\text{FO}, \mathcal{G})$, where $\mathcal{G}$ is the class of all finite graphs, is complete for a parameterized complexity class called AW[$*$] which is much larger than W[1]. Hence, unless FPT $=$ AW[$*$], an assumtion widely disbelieved in the community, first-order model-checking is not fixed-parameter tractable on the class of all graphs.

Let G be a structure and $v_1, \ldots, v_k$ be elements in $V(G)$. For $q \geq 0$, the *first-order q-type* $\text{tp}_q^G(\overline{v})$ of $\overline{v}$ is the class of all FO-formulas $\varphi(\overline{x})$ of quantifier-rank $\leq q$ such that $G \models \varphi(\overline{v})$. The first-order 0-type is referred to as the *atomic type* of $\overline{v}$ and denoted by $\text{atp}^G(\overline{v})$. We will usually omit the superscript G if its is clear from the context. A *first-order q-type* $\tau(\overline{x})$ is a maximally consistent class of formulas $\varphi(\overline{x})$.

By definition, types are infinite. However, it is well known that there are only finitely many FO-formulas of quantifier rank $\leq q$ which are pairwise not equivalent. Furthermore, we can effectively *normalise* formulas in such a way that equivalent formulas are normalised syntactically to the same formula. Hence, we can represent types by their finite set of normalised formulas and we can also check whether a formula belongs to a type. Note, though, that it is undecidable whether a set of formulas is a type as by definition, types are satisfiable.

We refer to [9] for a definition of Ehrenfeucht-Fraïssé games.

Let $\mathcal{C}$ be a class of graphs. The *first-order theory* $\text{Th}_{\text{FO}}(\mathcal{C})$ is defined as the class of first-order formulas true in all graphs $G \in \mathcal{C}$.

3 Tree-Depth and Nowhere Dense Classes of Graphs

In this section we present the concepts of *tree depth* and *nowhere dense* classes of graphs introduced in [21, 20].

The tree depth of graphs. We first need some further notation. A *rooted tree* (T, r) is a connected acyclic graph T with a distinguished vertex r, the root of the tree. A *rooted forest* is the disjoint union of rooted trees. The height of a vertex v in a rooted tree (T, r) is the length of the path (number of edges) from the root r to v. The height of a tree is the maximal height of its vertices. The height of a rooted forest is the maximal height of the trees it contains. A vertex $u \in V(T)$ is an *ancestor* of $v \in V(T)$, and v is a *descendant* of u, if u lies on the path from the root r to v.

Let T be a rooted tree. The *closure* of T $\text{clos}(T)$ is defined as the graph obtained from T by adding an edge from every vertex $v \in V(T)$ to each of its descendants in T. The closure of a rooted forest is defined analogously.

3.1 Definition ([21])**.** *A graph G has* tree-depth h *if it is a sub-graph of the closure of a rooted forest F of height at most h. We call F a* tree-depth decomposition *of G.*

It is an easy exercise to show that every graph G of tree-depth at most h also has path-width and hence tree-width at most h. To simplify presentation, we will always assume in the sequel that the tree-depth decomposition is actually a tree, rather than a forest. At no point will the extension to forests cause any difficulties whatsoever.

The following was proved in [21].

3.2 Theorem. *There is an algorithm which, given a graph G of tree-depth at most h, computes a tree-depth decomposition in time $f(h) \cdot |G|$, for some computable function h.*

Nowhere dense classes of graphs. We now recall the definition of nowhere dense classes of graphs. A graph H is a *minor* of G (written $H \preccurlyeq G$) if H can be obtained from a sub-graph of G by contracting edges. An equivalent characterisation (see [7]) states that H is a minor of G if there is a map that associates to each vertex v of H a non-empty tree $G_v \subseteq G$ such that G_u and G_v are disjoint for $u \neq v$ and whenever there is an edge between u and v in H there is an edge in G between some node in G_u and some node in G_v. The sub-graphs G_v are called *branch sets*.

We say that H is a *minor at depth* r of G (and write $H \preccurlyeq_r G$) if H is a minor of G and this is witnessed by a collection of branch sets $\{G_v \mid v \in V(H)\}$, each of which induces a graph G_v of radius at most r. That is, for each $v \in V(H)$, there is a $w \in V(G)$ such that $G_v \subseteq N_r^{G_v}(w)$.

The following definition is due to Nešetřil and Ossona de Mendez [20].

3.3 Definition (nowhere dense classes)**.** *A class of graphs $\mathcal{C}$ is said to be* nowhere dense *if for every r there is a graph H such that $H \not\preccurlyeq_r G$ for all $G \in \mathcal{C}$.*

$\mathcal{C}$ is called somewhere dense *if it is not nowhere dense.*

It follows immediately from the definitions that if a class $\mathcal{C}$ of graphs which is not nowhere dense then there is a radius r such that every graph H is a depth r minor of some graph $G_H \in \mathcal{C}$. If, furthermore, $\mathcal{C}$ is closed under taking sub-graphs, then the depth-d image I_H of H in G_H is itself a graph in $\mathcal{C}$. Note that the size of I_H is polynomially bounded in H (for fixed r). Classes which are not nowhere dense are called *somewhere dense* in [20]. Let us call a class *effectively somewhere dense* if, given a graph H, a depth-d image $I_H \in \mathcal{C}$ of H in a graph $G_H \in \mathcal{C}$ can be computed in polynomial time.

Low Tree Depth Colourings. We will also need the following results from [20]. For each $p \in \mathbb{N}$ and each graph G let $\chi_p(G)$ be the least number of colours needed for a proper vertex colouring of G such that any $i < p$ colours induce a sub-graph of G of tree-depth at most i. Clearly, this is well-defined, as the colouring which assigns a different colour to each vertex has this property. However, for special classes $\mathcal{C}$ of graphs we can do with far fewer colours.

It was shown in [19] that if $\mathcal{C}$ is a class of graphs of *bounded expansion*, then for each $p > 0$ there is an $N(p) > 0$ such that $\chi_p(G) \leq N(p)$ for all $G \in \mathcal{C}$. Furthermore, such a colouring can be computed in linear time, for each p. If $\mathcal{C}$ does not have bounded expansion then this fails as bounded expansion is actually equivalent to the existence of such an $N(p)$ for all p.

However, it was shown in [20] that if $\mathcal{C}$ is nowhere dense then

$$\lim_{p\to\infty} \limsup_{G\in\mathcal{C}} \frac{\log \chi_p(G)}{\log |G|} = 0.$$

Hence, for every $\delta > 0$ there is a p_0, n_0 such that if $G \in \mathcal{C}$ and $|G| > n_0$ then G can be coloured by at most $|G|^\delta$ colours such that any $i < p$ parts of this colouring induce a sub-graph of G of tree depth at most i. Furthermore, for every $\varepsilon > 0$ there is an algorithm for computing such a colouring in time $|G|^{1+\varepsilon}$.

4 The Type Representation Property and Graph Classes of Bounded Tree-Depth

In this section we introduce the concept of *type representation schemes* and the *type representation property*. Moreover, we will show that for each k, the class of all graphs of tree depth at most k has the type representation property. Graph classes with the type representation property form the cornerstone of the method for establishing tractability results for first-order logic presented in the next section.

4.1 Definition. *Let $\mathcal{C}$ be a class of graphs. A* type representation scheme $\mathfrak{L}$ *for $\mathcal{C}$ consists, for each $r \geq 0$, of*

1. *a finite set $\mathfrak{L}_r$ of labels*
2. *for each $G \in \mathcal{C}$ and each $\overline{v} \in V(G)^i$ a labelling* $\mathrm{lab}_{\mathfrak{L}}(\overline{v}) \in \mathfrak{L}_i$ *and*
3. *an algorithm A which, given $G \in \mathcal{C}$ and $r \geq 0$, computes for each $L \in \mathfrak{L}_i$, where $i \leq r$, a tuple* $\mathrm{wit}^G(L) \in V(G)^i$ *such that* $\mathrm{lab}_{\mathfrak{L}}(\mathrm{wit}^G(L)) = L$*, if such a tuple exists, or otherwise marks L as not realised in G*

such that for all $G \in \mathcal{C}$ and $\overline{u} := (u_1, \dots, u_i) \in V(G)^i$, $\overline{v} := (v_1, \dots, v_i) \in V(G)^i$ with $\mathrm{lab}_{\mathfrak{L}}(\overline{u}) = \mathrm{lab}_{\mathfrak{L}}(\overline{v})$ *the following properties hold:*

- *(equivalence) For all $\varphi(x_1, \dots, x_i) \in$ FO of quantifier-rank at most $r - i$*

$$G \models \varphi(\overline{u}) \textit{ if, and only if, } G \models \varphi(\overline{v}).$$

- *(consistency) If $i > 1$, then* $\mathrm{lab}_{\mathfrak{L}}(u_1, \dots, u_{i-1}) = \mathrm{lab}_{\mathfrak{L}}(v_1, \dots, v_{i-1})$.

We say that $L' \in \mathfrak{L}_{i+1}$ extends *$L \in \mathfrak{L}_i$ if there is a tuple $(v_1, \dots, v_{i+1}) \in V(G)^{i+1}$ with* $\mathrm{lab}_{\mathfrak{L}}(\overline{v}) = L'$ *and* $\mathrm{lab}_{\mathfrak{L}}(v_1, \dots, v_i) = L$.

$\mathcal{C}$ has the type representation property *if it has a type representation scheme where the algorithm A runs in time $f(r) \cdot |G|^c$, for some computable function $f : \mathbb{N} \to \mathbb{N}$ and constant $c \in \mathbb{N}$.*

We will refer to the pair (f, c) as the time bound *of the representation scheme.*

Note that every class of graphs has a type representation scheme: simply let $\mathfrak{L}_r$ be the (finite) set of all first-order types of r tuples up to quantifier-rank r. Clearly, this is a type representation scheme. The problem is that computing the types and finding representatives for the labels $L \in \mathfrak{L}_r$ cannot be done efficiently in general and hence not every class of graphs has the type representation property.

We will show next that for each $k \geq 0$, the class of graphs of tree depth at most k has the type representation property.

It will be convenient for us to encode graphs of tree-depth $\leq h$ as labelled trees of height h. Let T be a tree of height h with root r. Let $\Sigma_h := \{c, g_0, g_1\} \cup \{(e_0, \dots, e_h) : e_i \in \{0, 1, -\}\}$. We encode a graph $G \subseteq \mathrm{clos}(T)$ as Σ_h-labelled tree (T, σ), where $\sigma(r) := \{c, g_r\}$, where $g_r = g_1$ if $r \in V(G)$ and $g_r = g_0$ otherwise, and for each $v \neq r$ of height i in the tree we let $\sigma(v) := \{g_v, (e_0, \dots, e_h)\}$ where

- $g_v = g_1$ if $v \in V(G)$ and $g_v = g_0$ otherwise,
- $e_j := -$ for all $j \geq i$ and
- for $j < i$, if u_j is the ancestor of v at height j, then $e_j := 1$ if $\{v, u_j\} \in E(G)$ and $e_j := 0$ otherwise.

Hence, c marks the root of the tree and vertices labelled g_1 represent the vertices in G. The tuples $(e_0, \dots, e_h)$ encode the edges of G. It is easily seen that every formula $\varphi(\overline{x}) \in$ FO of quantifier-rank q can effectively be translated into a formula $\varphi^*(\overline{x}) \in$ FO of quantifier-rank at most $q + h$ such that for all $\overline{u} \in V(G)$, $G \models \varphi(\overline{u})$ if, and only if, $(T, \sigma) \models \varphi^*(\overline{u})$. We fix this translation for the rest of the paper.

4.2 Definition. *Let T be a tree of height h and let $x, y \in V(T)$. The* least common ancestor $\mathrm{lca}_T(x, y)$ *of x and y in T is the element of T of maximal height that is an ancestor of both x and y. We define* $\mathrm{lch}_T(x, y)$ *to be the height of* $\mathrm{lca}_T(x, y)$.

If T is clear from the context we will omit the subscript in $\mathrm{lch}_T(x, y)$ and simply write $\mathrm{lch}(x, y)$.

4.3 Lemma (Equivalence Lemma). *Let (T, σ) be a Σ_h-labelled tree of height at most h encoding a graph G and let $\varphi(x_1, \ldots, x_r) \in \text{FO}$ be a formula of quantifier-rank at most q. If $u_1, \ldots, u_r, v_1, \ldots, v_r \in V(T)$ are such that $g_1 \in \sigma(u_i), g_1 \in \sigma(v_i)$, for all $1 \leq i \leq r$, and for all $1 \leq i \leq j \leq h$,*

$$\text{tp}^{(T,\sigma)}_{(r+1+q-j)\cdot h}(v_j) = \text{tp}^{(T,\sigma)}_{(r+1+q-j)\cdot h}(u_j) \text{ and } \text{lch}(v_i, v_j) = \text{lch}(u_i, u_j),$$

then $G \models \varphi(v_1, \ldots, v_r)$ if, and only if, $G \models \varphi(v_1, \ldots, v_r)$.

Proof. First note that if $\text{tp}^{(T,\sigma)}_h(u) = \text{tp}^{(T,\sigma)}_h(v)$ for some vertices $u, v \in V(T)$, then u and v are of the the same height in T as the height of a vertex is definable by a first-order formula of quantifier-rank at most h. It follows that the height of u_i equals the height of v_i, for all $1 \leq i \leq r$. By the same argument, v_i is an ancestor of v_j if, and only if, u_i is an ancestor of u_j and the distances between v_i, v_j and u_i, u_j are the same.

By induction on q we show that Duplicator has a winning strategy in the q-round Ehrenfeucht-Fraïssé game $\mathfrak{G}_q(A, v_1, \ldots, v_r; B, u_1, \ldots, u_r)$, where $A = B = G$. The distinction between A and B is simply to easy notation.

For $q = 0$ it suffices to show that $\text{atp}(v_1, \ldots, v_r) = \text{atp}(u_1, \ldots, u_r)$. Clearly, $\sigma(u_i) = \sigma(v_i)$, for all i, as they have the same atomic type. We show next that $\{v_i, v_j\} \in E(G)$ if, and only if, $\{u_i, u_j\} \in E(G)$. If v_j, v_i are incomparable by the ancestor relation, then u_j, u_i are incomparable and there is no edge between them by the definition of a tree-depth decomposition. Conversely, if v_i is an ancestor of v_j and the height of v_i is s then, by the remark above, u_i is an ancestor of u_j and the height of u_i is also s. But then the edge between v_i, v_j is encoded in the label of v_j and as u_j has the same label there is an edge between u_i and u_j. The converse is analogous.

Now let $q > 0$. By assumption, $\text{tp}^{(T,\sigma)}_{(r+q-j)\cdot h}(v_j) = \text{tp}^{(T,\sigma)}_{(r+q-j)\cdot h}(u_j)$, for all $1 \leq j \leq r$. Suppose first that Spoiler chooses $v \in A$.

- If v is an ancestor of some v_j, then Duplicator chooses the corresponding ancestor of u_j. More precisely, let $P := x_0 \ldots x_s$ be the path from the root $r = x_0$ of T to $x_s = v_j$ and let $S := y_0 \ldots y_s$ be the corresponding path from the root to u_j. If Spoiler chooses x_i, $i < j$, then Duplicator chooses y_j. Now, $A \models \vartheta(v_j)$ where $\vartheta(z) := \exists z_0 \ldots \exists z_s \big(z_0 = c \wedge z_s = z \wedge \bigwedge_{i=0}^{j-1} E(z_i, z_{i+1}) \wedge \text{tp}^{(T,\sigma)}_{(r+q-j-1)\cdot h}(x_i)$. Here, $\text{tp}^{(T,\sigma)}_{(r+q-j-1)\cdot h}(x_i)$ is the conjunction of all formulas $\chi(x)$ of quantifier-rank $\leq (r+q-j-1)\cdot h$ true at x_i.

  By assumption, $B \models \vartheta(u_j)$ and therefore $\text{tp}^{(T,\sigma)}_{(r+q-j-1)\cdot h}(y_i) = \text{tp}^{(T,\sigma)}_{(r+q-j-1)\cdot h}(x_i)$. Furthermore, by the choice of x_i and y_i it is clear that $\text{lch}(x_i, v_s) = \text{lch}(y_i, u_s)$ for all $1 \leq s \leq r$. Hence, we can apply the induction hypothesis to conclude that Duplicator has a winning strategy on the remaining $q-1$ round game $\mathfrak{G}_{q-1}(A, v_1, \ldots, v_r, x_i; B, u_1, \ldots, u_r, y_i)$.
- The other cases, where Spoiler chooses a descendant v of some v_j or an element not related to any $v_1, \ldots, v_r$ can be argued similarly as distances up to h steps in T can be defined as in the previous case.

The case where Spoiler chooses $v \in B$ is symmetric. □

The following lemma is a simple consequence of Courcelle's theorem [1].

4.4 Lemma. *1. There is an algorithm which, given $q \in \mathbb{N}$ and a Σ_h-labelled tree (T, σ) of height at most h computes for each $v \in V(T)$ the type $\mathrm{tp}_{q \cdot h}(v)$ in time $f(q, h) \cdot |T|$, where $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is a computable function.*

2. There is an algorithm which, given $(t_1, \dots, t_r, (a_{i,j})_{1 \le i < j \le r})$ where the t_i are $q \cdot h$-types and $1 \le a_{i,j} \le h$, for $i < j$, computes a tuple $\overline{v} \in V(T)^r$ such that $\mathrm{lch}(v_i, v_j) = a_{i,j}$ and $\mathrm{tp}_{q \cdot h}^{(T,\sigma)}(v_i) = t_i$, if such a tuple exists, in time $f(r) \cdot |T|$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

The previous lemmas together imply that for each $k \ge 0$ the class $\mathcal{C}_k$ of graphs of tree depth at most k has the type respresentation property: for $r \ge 0$ we let $\mathfrak{L}_r := \big(t_1, \dots, t_r, (a_{i,j})_{1 \le i < j \le r}\big)$, where the t_i are $(3r+1)$-types $\mathrm{tp}_{3r+1}(x)$ and $a_{i,j} \in \{0, \dots, k\}$. For $G \in \mathcal{C}_k$ and $\overline{v} \in V(G)^r$ we let $\mathrm{lab}_{\mathfrak{L}}(\overline{v}) := \big(t_1, \dots, t_r, (a_{i,j})_{1 \le i < j \le r}\big)$, where $t_i := \mathrm{tp}_{3r+1}^{(T,\sigma)}(x)$ and $a_{i,j} := \mathrm{lch}^{(T,\sigma)}(u_i, u_j)$. Here, (T, σ) is the tree encoding of a tree-depth decomposition of G[1].

4.5 Theorem. *For each $k \ge 0$, the class $\mathcal{C}_k$ of graphs of tree depth at most k has the type representation property.*

Another consequence of the previous lemmas is the following observation.

4.6 Theorem. *There is an algorithm which, given a graph G of tree-depth at most h and a formula $\varphi(x_1, \dots, x_r)$ of quantifier-rank at most q, computes in time $f(h, r+q) \cdot \mathcal{O}(|G|)$ a data structure $\mathcal{A}$ such that given $v_1, \dots, v_r \in V(G)$ we can decide in time $\mathcal{O}(r)$ whether $G \models \varphi(v_1, \dots, v_r)$.*

5 A General Meta-Theorem and Applications

The aim of this section is to prove a general meta-theorem for first-order logic and derive some consequences. The basis of the meta-theorem are graph classes with the type representation property as introduced in the previous section. We first need some notation.

5.1 Definition. *For each $r \ge 0$ let $\mathcal{C}_r$ be a class of graphs and let $\mathcal{C} := (\mathcal{C}_r)_{r \ge 0}$. A $\mathcal{C}$-*colouring of width *r of a graph G is a colouring of G such that for all colours $C_1, \dots, C_r$, the sub-graph $G[C_1, \dots, C_r]$ of G induced by the colours $C_1, \dots, C_r$ is in $\mathcal{C}_r$.*

A class $\mathcal{D}$ of graphs is colourable over *$\mathcal{C}$, or $\mathcal{C}$-*colourable*, if for every $r \ge 0$ and every $G \in \mathcal{D}$ there is a $\mathcal{C}$-colouring of G of width r.*

Note that we do not require the colouring in the previous definition to be proper in the graph theoretical sense, i.e. both endpoints of an edge can be coloured by the same colour. For our purposes we need that $\mathcal{C}$-colourings can be computed efficiently.

5.2 Definition. *For each $r \ge 0$ let $\mathcal{C}_r$ be a class of graphs and let $\mathcal{C} := (\mathcal{C}_r)_{r \ge 0}$. A class $\mathcal{D}$ of graphs is* efficiently colourable over *$\mathcal{C}$ if*

[1]Note that tree-depth decompositions of a graph are not necessarily unique. However, as they can be computed by a deterministic algorithm there is a canonical one which we choose. This will, however, never be a problem in the sequel and we could even work without a canonical representation.

- *for each* $r \geq 0$ *and* $\varepsilon > 0$ *there is an* $n_0 \geq 0$ *such that for all* $G \in \mathcal{C}$, $\chi_p(G) \leq |G|^\varepsilon$, *where* $\chi_p(G)$ *denotes the minimal number of colours for a* $\mathcal{C}$*-colouring of* G *of width* r *and*

- *there is an algorithm which, given* $G \in \mathcal{D}$ *and* $r \geq 0$, $\varepsilon > 0$ *computes such a colouring of* G *in time* $f(r,\varepsilon) \cdot |G|^c$, *for some computable function* $f : \mathbb{N} \times \mathbb{Q} \to \mathbb{N}$ *and* $c \in \mathbb{N}$.

We will refer to the pair (f, c) *as the* time bound *of the colouring.*

As an example, let $\mathcal{C}_r$ be the class of all graphs of tree depth at most r and let $\mathcal{C} := (\mathcal{C}_r)_{r\geq 0}$. As we have seen in Section 3 above, every nowhere dense class $\mathcal{D}$ of graphs is efficiently colourable over $\mathcal{C}$. Furthermore, for each $r, \varepsilon > 0$ a colouring can be computed in time $f(r,\varepsilon) \cdot |G|^{1+\varepsilon}$.

5.3 Lemma. *For* $\varepsilon > 0$, *every nowhere dense class* $\mathcal{D}$ *is efficiently colourable over* $\mathcal{C}$ *with time bound* $(f, 1+\varepsilon)$, *for some computable function* $f : \mathbb{N} \times \mathbb{Q} \to \mathbb{N}$.

We now prove the main result of this section.

5.4 Theorem. *For each* $r \geq 0$ *let* $\mathcal{C}_r$ *be a class of graphs with the type representation property and let* $\mathcal{C} := (\mathcal{C}_r)_{r\geq 0}$. *If* $\mathcal{D}$ *is efficiently colourable over* $\mathcal{C}$ *then* $\text{MC}(\text{FO}, \mathcal{D}) \in \text{FPT}$.

Furthermore, if (f_1, c_1) *and* (f_2, c_2) *are the time bounds of the type representation scheme and the colouring, respectively, then for every* $\delta > 0$ *there is a computable function* $f : \mathbb{N} \to \mathbb{N}$ *such that for each* $\varphi \in \text{FO}$ *and* $G \in \mathcal{D}$, $G \models \varphi$ *can be decided in time* $f(|\varphi|) \cdot |G|^{\max\{c_1+\delta, c_2\}}$.

For the rest of this section let us fix $\mathcal{C}$ and $\mathcal{D}$ as in the statement of the theorem. Suppose we are given a graph $G \in \mathcal{D}$ and a formula $\varphi' \in \text{FO}$ of which we want to decide $G \models \varphi$. We first translate φ' into an equivalent formula φ in prenex normal form where in addition we assume that the first quantifier is existential, i.e. φ is of the form $\exists x_1 Q x_2 \dots Q_r x_r \psi(x_1, \dots, x_r)$. Let r be the number of variables in φ and let $\mathfrak{L}$ be the type representation scheme of $\mathcal{C}_r$. Fix some $\varepsilon < \frac{1}{3r}$.

Step 1. We first compute a colouring of G by $c := |G|^\varepsilon$ colours $\mathfrak{C} := \{C_1, \dots, C_c\}$ such that any $i \leq r$ colours $\overline{C} := (C_1, \dots, C_i)$ induce a sub-graph $G_{\overline{C}} := G[C_1, \dots, C_i] \in \mathcal{C}_r$. We will denote the colour of $v \in V(G)$ by $\mathfrak{C}(v)$.

5.5 Definition. *1. For* $\overline{v} := v_1, \dots, v_s \in V(G)$, *where* $s \leq r$, *we define*

$$\text{col}(\overline{v}) := (\text{lab}_{\mathfrak{L}}(\overline{v}), \mathfrak{C}(v_1), \dots, \mathfrak{C}(v_s)).$$

2. *We call a tuple* $\overline{c} := (L, C_1, \dots, C_s)$ realisable at width s *if* L *is realisable in* $G_{(C_1,\dots,C_s)}$ *and denote the set of realisable tuples at width* s *by* $\mathcal{R}_s$. *We let* $\mathcal{R} := \bigcup_{i=1}^{r} \mathcal{R}_s$ *and call the tuples in* $\mathcal{R}$ realisable.

3. *For* $\overline{c} := (L, \overline{C}) \in \mathcal{R}_s$ *we define* $\text{wit}(\overline{c}) := \text{wit}^{G_{\overline{C}}}(L)$. *Hence,* $\text{wit}(\overline{c})$ *is a tuple* $\overline{v} \in V(G_{\overline{C}})$ *witnessing that* L *is realised in* $G_{\overline{C}}$.

4. *If* $\overline{c} := (L, C_1, \dots, C_s)$ *and* $\overline{c}' := (L', C'_1, \dots, C'_{s+1})$ *are realisable tuples we say that* $\overline{c}'$ extends $\overline{c}$ *if for all* $1 \leq i < j \leq s$, $C_i = C'_i$ *and* L' *extends* L *in* $G_{(C'_1,\dots,C'_{s+1})}$.

After the initial preprocessing in Step 1, the algorithm for checking whether $G \models \varphi$ now proceeds in three further steps.

Step 2. For all $\overline{c} \in \mathcal{R}_r$ let $\overline{v} := \text{wit}(\overline{c})$. If $G \models \psi(\overline{v})$ mark $\overline{c}$ as *good*, otherwise mark it as *bad*.

Step 3. For $i := r - 1...1$ (counting downwards) and for all $\overline{c} \in \mathcal{R}_i$ do the following:

- if $Q_i = \forall$ and all $\overline{c}' \in \mathcal{R}_{i+1}$ which extend $\overline{c}$ are good then mark $\overline{c}$ as good. Otherwise mark it as bad.
- if $Q_i = \exists$ and there exists a good $\overline{c}' \in \mathcal{R}_{i+1}$ extending $\overline{c}$, then mark $\overline{c}$ as good. Otherwise mark it as bad.

Step 4. If there is a good tuple in $\mathcal{R}_1$ then accept, otherwise reject. (Recall that φ was assumed to be in prenex normal form starting with an existential quantifier.)

We next show correctness of the algorithm.

5.6 Lemma. *On input $G, \varphi := \exists x_1 Q x_2 \dots Q_r x_r \psi(x_1, \dots, x_r)$, where ψ is atomic, the algorithm returns true if, and only if, $G \models \varphi$.*

Proof. To simplify notation let us define

$$\varphi_i(x_1, \dots, x_i) := Q_{i+1} x_i \dots Q_r x_r \psi(x_1, \dots, x_r),$$

for $0 \leq i \leq r$. Hence, $\varphi_0 = \varphi$ and $\varphi_r = \psi$. By induction on $0 \leq q \leq r$ we will show that for all $v_1, \dots, v_s \in V(G)$, where $s := r - q$,

$$G \models \varphi_s(v_1, \dots, v_s) \text{ if, and only if, } \text{col}(v_1, \dots, v_s) \text{ is good after Step 3.}$$

Clearly, for $q = r - 1$ this implies that $G \models \varphi_1(v)$ if, and only if, $\text{col}(v)$ is good. Hence, in Step 4, the algorithm accepts if, and only if, there is a good tuple $\overline{c} \in \mathcal{R}_1$ if, and only if, there $G \models \varphi_1(\text{wit}(\overline{c}))$ if, and only if, $G \models \exists x_1 \varphi_1$, i.e. $G \models \varphi$.

Assume first that $q = 0$ and let $\overline{v} := v_1, \dots, v_r \in V(G)$. Let $\overline{c} := \text{col}(\overline{v}) = (L, \overline{C})$. Suppose that $G \models \varphi_r(\overline{v})$. By the equivalence property of type representation schemes, if $\text{col}(\overline{v}) = \text{col}(\overline{u})$ for some $\overline{u} := u_1, \dots, u_r \in V(G)$ then $G_{\overline{C}} \models \varphi_r(\overline{v})$ if, and only if, $G_{\overline{C}} \models \varphi_r(\overline{u})$. Further, as φ_r is atomic, $G_{\overline{C}} \models \varphi_r(\overline{u})$ if, and only if, $G \models \varphi_r(\overline{u})$. Hence, as $G \models \varphi_r(\overline{v})$, this implies that $G_{\overline{C}} \models \varphi_r(\text{wit}(\overline{c}))$ and therefore $\overline{c}$ is marked good in Step 2.

Conversely, assume that $\overline{c} := \text{col}(v_1, \dots, v_r) \in \mathcal{R}_r$ is good after Step 2. By construction of the algorithm, $G \models \varphi_r(\text{wit}(\overline{c}))$ and therefore, by the equivalence property of type representation schemes as before, $G \models \varphi_r(v_1, \dots, v_r)$.

Now assume that $q > 0$ and set $s := r - q$. Let $v_1, \dots, v_s \in V(G)$ and let $\overline{c} := \text{col}(v_1, \dots, v_s)$. Suppose first that $G \models \varphi_s(v_1, \dots, v_s)$. If $Q_s = \forall$, then this implies that $G \models \varphi_{s+1}(v_1, \dots, v_s, v)$ for all $v \in V(G)$. By induction hypothesis, $\text{col}(v_1, \dots, v_s, v)$ is good for all $v \in V(G)$. As this spans all tuples $\overline{c} \in \mathcal{R}_{s+1}$ which extend $\overline{c}$ this implies that $\overline{c}$ is marked as good in Step 3.

If $Q_s = \exists$, then there must be a vertex $v \in V(G)$ such that $G \models \varphi_{s+1}(v_1, \dots, v_s, v)$ and therefore, by induction hypothesis, $\text{col}(v_1, \dots, v_s, v)$ is marked as good. As clearly $\text{col}(v_1, \dots, v_s, v)$ extends $\overline{c}$, $\overline{c}$ is marked as good in Step 3.

Finally, assume that $\overline{c} \in \mathcal{R}_s$ is marked as good in Step 3. We again distinguish between $Q_s = \forall$ and $Q_s = \exists$. If $Q_s = \forall$, then $\overline{c}$ being good implies that all $\overline{c}'$ which extend $\overline{c}$ are good. By induction hypothesis and the equivalence property of type representation schemes, this implies that for all $\overline{u} := (u_1, \ldots, u_{s+1})$ such that $\mathrm{col}(\overline{u})$ extends $\overline{c}$, $G \models \varphi_{s+1}(\overline{u})$. However, for every $v \in V(G)$, $\mathrm{col}(v_1, \ldots, v_s, v)$ extends $\mathrm{col}(v_1, \ldots, v_s)$ and therefore for all $v \in V(G)$, $G \models \varphi_{s+1}(v_1, \ldots, v_s, v)$. Thus, $G \models \varphi_s(v_1, \ldots, v_s)$. The case $Q_s = \exists$ can be argued analogously. This concludes the proof of the lemma. □

The previous lemma established the correctness of the algorithm. We now analyse its running time. In what follows, let $r := |\varphi|$. Let $f_1 : \mathbb{N} \to \mathbb{N}$ and $c_1 \in \mathbb{N}$ be the time bound for the type representation scheme of $\mathcal{C}$ as defined in Definition 4.1 and let $f_2 : \mathbb{N} \to \mathbb{N}$ and $c_2 \in \mathbb{N}$ be the time bound for computing the $\mathcal{C}$-colouring as defined in Definition 5.2.

1. The algorithm first computes the colouring of G. By definition of efficient $\mathcal{C}$-colourings, for every fixed r and $\varepsilon > 0$ there exists the required colouring of G using $c := |G|^{\varepsilon}$ colours which can be computed in time $f_2(r, \varepsilon)|G|^{c_2}$.

2. The algorithm then computes in each $G_{\overline{C}} := G[\overline{C}]$, where $\overline{C} := (C_1, \ldots, C_r)$ is a tuple of colours, the witnesses $\mathrm{wit}(L, \overline{C})$ for all $L \in \mathcal{R}_r$. This takes time $c^r \cdot f_1(r) \cdot |G|^{c_1}$.

3. After this preparation the algorithm proceeds to the two main steps. In Step 2 we need time $c^r \cdot \mathcal{O}(r)$ to check which tuples $\overline{c}$ are good and Step 3 requires a total of $r \cdot c^r \cdot c^r = r \cdot c^{2r}$ time.

Hence, in total the algorithm needs

$$f_2(|\varphi|, \varepsilon)|G|^{c_2} + c^{3|\varphi|} \cdot g(|\varphi|) \cdot |G|^{c_1},$$

where $g : \mathbb{N} \to \mathbb{N}$ is a computable function, depending on f_2. However, as $c := |G|^{\varepsilon}$ and $\varepsilon < 3r$, the running time is bounded by $f_2(|\varphi|, \varepsilon)|G|^{c_2} + g(|\varphi|) \cdot |G|^{c_1+\delta}$, where $\delta := \varepsilon \cdot 3|\varphi| < 1$, and this is enough to show that the algorithm runs in parameterized polynomial time. This completes the proof of Theorem 5.4.

Recall that for nowhere dense classes of graphs the time bounds for the type representation scheme over graphs of bounded tree depth as linear in the size of $|G|$ and that the colouring can be computed in time $|G|^{1+\varepsilon}$, for every $\varepsilon > 0$. Hence, for nowhere dense classes of graphs we get the following result.

5.7 Corollary. *Let $\mathcal{C}$ be a nowhere dense class of graphs. For every $\varepsilon > 0$ there is a computable function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm which, given $G \in \mathcal{C}$ and $\varphi \in$ FO, decides $G \models \varphi$ in time $f(|\varphi|) \cdot |G|^{1+\varepsilon}$.*

For graph classes $\mathcal{C}$ of *bounded expansion* we can do even better. It was shown in [19] that if $\mathcal{C}$ is a class of graphs of bounded expansion then for each $p \geq 0$ there exists an $N(p) \geq 0$ such that every graph $G \in \mathcal{C}$ can be coloured by $N(p)$ colours in a way that any $i \leq p$ colours induce a sub-graph of tree depth at most i and such a colouring can be computed in linear time. Hence, following the analysis of the running time above, for such classes we obtain a linear time parameterized algorithm.

5.8 Corollary. *Let $\mathcal{C}$ be a class of graphs of bounded expansion. There is linear time parameterized algorithm solving the first-order model-checking problem* $\text{MC}(\text{FO}, \mathcal{C})$ *on* $\mathcal{C}$.

Recall that graph classes of bounded expansion strictly generalise graph classes excluding a fixed minor. Hence the previous result improves significantly over the time bounds achieved in [11].

6 Graph Classes which are Somewhere Dense

As a consequence of the main theorem in the previous section we obtained that first-order model-checking is fixed-parameter tractable on all classes of graphs which are nowhere dense. In this section we will show that, if we consider classes of graphs closed under sub-graphs, then essentially tractable first-order model-checking cannot be extended beyond classes that are nowhere dense. In this way, for classes closed under sub-graphs, we essentially obtain an precise characterisation of the classes of graphs for which first-order model-checking is tractable.

Recall the definition of effectively somewhere dense classes of graphs in Section 2. If a class $\mathcal{C}$ of graphs is not nowhere dense then there is a radius r such that every graph H is a depth r minor of some graph $G_H \in \mathcal{C}$. If, furthermore, $\mathcal{C}$ is closed under taking sub-graphs, then the depth-d image I_H of H in G_H is itself a graph in $\mathcal{C}$. Note that the size of I_H is polynomially bounded in H (for fixed r). Classes which are not nowhere dense are called *somewhere dense* in [20]. Let us call a class *effectively somewhere dense* if, given a graph H, a depth-d image $I_H \in \mathcal{C}$ of H in a graph $G_H \in \mathcal{C}$ can be computed in polynomial time.

6.1 Theorem. *If $\mathcal{C}$ is closed under sub-graphs and effectively somewhere dense then* $\text{MC}(\text{FO}, \mathcal{C}) \notin \text{FPT}$ *unless* $\text{FPT} = \text{AW}[*]$.

To prove the theorem we will show that first-order model-checking on the class of all graphs, which is AW[*] complete, is parameterized reducible to first-order model-checking on any effectively somewhere dense class closed under sub-graphs. We find it convenient to state this in terms of a first-order interpretations. See e.g. [16].

6.2 Definition. *Let $\sigma := \{E\}$ be the signature of graphs, where E is a binary relation symbol. A (one-dimensional)* interpretation *from σ-structures to σ-structures is a triple $\Gamma := (\varphi_{univ}(x), \varphi_{valid}, \varphi_E(x,y))$ of* $\text{FO}[\sigma]$*-formulas.*

For every σ-structure T with $T \models \varphi_{valid}$ we define a graph $G := \Gamma(T)$ as the graph with vertex set $V(G) := \{u \in V(T) : T \models \varphi_{univ}(v)\}$ and edge set $E(G) := \{\{u,v\} \in V(G) : T \models \varphi_E(u,v)\}$.

If $\mathcal{C}$ is a class of σ-structures we define $\Gamma(\mathcal{C}) := \{\Gamma(T) : T \in \mathcal{C}, T \models \varphi_{valid}\}$.

Every interpretation naturally defines a mapping from $\text{FO}[\sigma]$-formulas φ to $\text{FO}[\sigma]$-formulas $\varphi^* := \Gamma(\varphi)$. Here, φ^* is obtained from φ by recursively replacing

- first-order quantifiers $\exists x\varphi$ and $\forall x\varphi$ by $\exists x(\varphi_{univ}(x) \wedge \varphi^*)$ and $\forall x(\varphi_{univ}(x) \rightarrow \varphi^*)$ respectively, and
- atoms $E(x,y)$ by $\varphi_E(x,y)$.

The following lemma is easily proved (see [16]).

6.3 Lemma (interpretation lemma)**.** *Let* Γ *be an* FO*-interpretation from* σ*-structures to* σ*-structures. Then for all* FO*-formulas and all* σ*-structures* $G \models \varphi_{valid}$

$$G \models \Gamma(\varphi) \quad \iff \quad \Gamma(G) \models \varphi.$$

6.4 Definition. *Let* $\mathcal{C}, \mathcal{D}$ *be classes of* σ*-structures. A* first-order reduction (Γ, f) *from* $\mathcal{C}$ *to* $\mathcal{D}$ *consists of a first-order interpretation* Γ *of* $\mathcal{C}$ *in* $\mathcal{D}$ *together with a polynomial-time computable function* $f : \mathcal{C} \to \mathcal{D}$ *such that for all* $G \in \mathcal{C}$ *and all* $\varphi \in \text{FO}[\sigma]$*,*

$$G \models \varphi \textit{ if, and only if, } f(G) \models \Gamma(\varphi).$$

The following lemma follows immediately from the definitions.

6.5 Lemma. *Let* $\mathcal{C}, \mathcal{D}$ *be two classes of graphs and let* (Γ, f) *be a first-order reduction from* $\mathcal{C}$ *to* $\mathcal{D}$*. Then* (Γ, f) *is a parameterized reduction from* $\text{MC}(\text{FO}, \mathcal{C})$ *to* $\text{MC}(\text{FO}, \mathcal{D})$*. In particular, if* $\text{MC}(\text{FO}, \mathcal{D}) \in \text{FPT}$ *then* $\text{MC}(\text{FO}, \mathcal{C}) \in \text{FPT}$*.*

Let $\mathcal{G}$ be the class of all graphs and let $\mathcal{C}$ be an effectively somewhere dense class of graphs closed under sub-graphs. Let r be the radius as above such that every graph occurs as a depth r minor of some graph in $\mathcal{G}$. We first define the function $f : \mathcal{G} \to \mathcal{C}$.

Let $H \in \mathcal{G}$ be a graph. We construct a graph H' as follows. Let $I \subseteq V(H)$ be the set of isolated vertices in H and let $V := V(H) \setminus I$.

For every vertex $v \in V$ we add the following gadget $\rho(v) := (V_v, E_v)$ to H': $V_v := \{v, v_1, v_2\}$ and $E_v := \big\{\{v, v_1\}, \{v, v_2\}\big\}$. Hence, essentially, we take v and add two new neighbours of degree 1. For every edge $\{u, v\} \in E(H)$ we add a path of length $2r$ linking v and u in H'. Formally, we fix an ordering $\leq_H$ on $V(H)$ and let

$$\begin{array}{rl} V(H') := & V(H) \dot\cup \{v_1, v_2 : v \in V(H) \setminus I\} \ \dot\cup \\ & \{e^i_{(v,w)} : 1 \leq i \leq 2r, \{u, v\} \in E(H), u \leq_H v\} \end{array}$$

and

$$\begin{array}{rl} E(H') := & \Big\{\{v, e^1_{(v,w)}\}, \{w, e^{2r}_{(v,w)}\}, \{e^i_{(v,w)}, e^{i+1}_{(v,w)}\} : \begin{array}{l} 1 \leq i < 2r, v \leq_H w, \\ \{v, w\} \in E(H) \end{array} \Big\} \cup \\ & \big\{\{v, v1\}, \{v, v_2\} : v \in V(H) \setminus I\big\} \end{array}$$

Now, let $G_{H'}$ be a depth d image of H' in a graph $G \in \mathcal{C}$. As $\mathcal{C}$ is closed under sub-graphs, $G_{H'} \in \mathcal{C}$ and, as $\mathcal{C}$ is effectively somewhere dense, given H, we can compute $G_{H'}$ in polynomial time. We define $f(H) := G_{H'}$.

To complete the reduction we define a first-order interpretation of $\mathcal{G}$ in $\mathcal{C}$. For this, we let $\varphi_{\text{univ}}(x)$ be the formula that says x *is an isolated vertex* or x *has degree at least* 3 and *there are two disjoint paths of length at most* r *from* x *to vertices of degree* 1. Now let H be a graph and let $G := f(H)$ be the image of H' in $\mathcal{C}$. Then $\varphi_{\text{univ}}(x)$ will be true at all vertices in G which are copies of vertices $v \in V(H)$. Now to define the edges we take the formula $\varphi_E(x, y)$ which says that x, y satisfy φ_{univ} and there is a path between x and y of length at most $2r^2$. Finally, we let φ_{valid} be the formula that says *every vertex either satisfies*

φ_{univ} *or lies on a path of length at most* $4r^2$ *between two vertices satisfying* φ_{univ} *and has degree* 2.

Now clearly, for all graphs $G \in \mathcal{G}$, $\Gamma(f(G)) \cong G$ and hence, by the interpretation lemma, $G \models \varphi$ if, and only if, $f(G) \models \Gamma(\varphi)$.

Theorem 6.1 now follows immediately from the fact that $\text{MC}(\text{FO}, \mathcal{G})$ is AW[∗]-complete (see e.g.[12]).

A further consequence of this construction is the following

6.6 Corollary. *If* $\mathcal{C}$ *is a somewhere dense class of graphs closed under subgraphs then* $\text{Th}_{\text{FO}}(\mathcal{C})$ *is undecidable.*

7 Graph Classes of Low Clique-Width Colouring

In this section we apply our method developed in Section 5 for establishing even more general meta-theorems for first-order logic. For $p \geq 0$, let $\mathcal{C}_p$ be the class of graphs of *clique-width* at most p and let $\mathcal{C} := (\mathcal{C}_p)_{p \geq 0}$. See [3] for a definition of clique-width. In this section we will show that $\mathcal{C}$ has the type representation property. As a consequence, every class $\mathcal{D}$ of graphs which is efficiently colourable over $\mathcal{C}$ has tractable first-order model-checking, see Corollary 7.2 below. As clique-width generalises tree-depth and every class of graphs of bounded clique-width is trivially colourable over $\mathcal{C}$, this result strictly generalises all known meta-theorems for first-order logic and provides a unifying link between classes of bounded clique-width and classes which exclude a minor or have bounded local tree-width or are nowhere dense.

We will show first that $\mathcal{C}$ has the type representation property.

7.1 Lemma. *For any* $p \geq 0$*, the class* $\mathcal{C}$ *of graphs of clique-width at most* p *has the type representation property.*

Proof. It is well known that, similarly to graphs of small tree-with, graphs G of clique-width at most p can be encoded as labelled binary trees $T(G)$ over a signature Σ_p. Essentially, the tree $T(G)$ corresponds to a clique-width expression generating G. The vertices of G are in one-to-one correspondence to the leaves of $T(G)$ and the inner vertices of $T(G)$ are labelled by the operations in the clique-width expression. See e.g. [2]. We will follow the presentation in [18].

Given a graph $G \in \mathcal{C}$ we can compute a clique-width expression generating G. It has been shown in [15] that computing approximate clique-width expressions generating G is fixed-parameter tractable. More precisely, one first computes a rank-decomposition of G of optimal width $s \leq p$ from which a clique-width expression of width at most 2^{s+1} can be computed efficiently. Hence, given $G \in \mathcal{C}$ we can compute a tree $T(G)$ encoding G, where $T(G)$ is a tree over Σ_k for $k := 2^{p+1}$, in time $f(p) \cdot |G|^c$, for some $c \in \mathbb{N}$ and computable function $f : \mathbb{N} \to \mathbb{N}$.

Furthermore, every first-order formula φ of quantifier-rank q can be translated effectively into an MSO_1-formula φ^* on Σ_k-labelled binary trees such that $G \models \varphi$ if, and only if, $T(G) \models \varphi^*$, where $T(G)$ is the encoding of G as a Σ_k-labelled tree. The quantifier-rank of φ^* is $q + k'$, where k' only depends on k and hence on the clique-width p of G.

We define a type representation scheme for $\mathcal{C}$ as follows. Let $r \geq 0$ be given. Let $\bar{c} := \{c_1, \ldots, c_r\}$ be constant symbols disjoint from any symbol in Σ_k. We let $\mathfrak{L}_r$ be the class of MSO-types of quantifier-rank r of $\Sigma_k\dot\cup\overline{c}$-structures, up to equivalence. Clearly, up to equivalence, there are only finitely many pairwise non-equivalent such types and hence $\mathfrak{L}_r$ is finite.

Now, given a graph G and a tuple $\overline{v} := (v_1, \dots, v_r) \in V(G)^r$ we first compute the Σ_k-labelled tree $T(G)$ as above. To compute the label $\text{lab}_{\mathfrak{L}}(\overline{v})$ we have to compute the $\text{tp}_r^{\text{MSO}}(\overline{v})$ in G. For this, let $\overline{u}$ be the tuple of leaves in $T(G)$ corresponding to $\overline{v}$. As above, we can compute the quantifier-$(r+k')$-MSO type of the structure $\big(T(G), u_1, \dots, u_r\big)$, the $\Sigma_k\dot\cup\overline{c}$-structure obtained from $T(G)$ by interpreting c_i by u_i, in time $f(r+k') \cdot |T|$ for some computable function $f : \mathbb{N} \to \mathbb{N}$. From this the type $\text{tp}_r^{\text{MSO}}(\overline{v})$ can be computed easily.

Finally, using exactly the same method, for each type $\tau \in \mathfrak{L}_r$ we can compute a tuple $\text{wit}^G(\tau) \in V(G)^r$ such that $\text{lab}_{\mathfrak{L}}(\text{wit}^G(\tau)) = \tau$ in linear time, if such a witness exists.

It is easily verified that $(\mathfrak{L}_r)_{r\geq 0}$ forms a type representation scheme. □

As a consequence, every class $\mathcal{D}$ which is efficiently colourable over $\mathcal{C} := (\mathcal{C}_p)_{p\geq 0}$ has tractable first-order model-checking. This is the main result of this section.

7.2 Corollary. *For $p \geq 0$ let $\mathcal{C}_p$ be the class of graphs of clique-width at most p and let $\mathcal{C} := (\mathcal{C}_p)_{p\geq 0}$. Let $\mathcal{D}$ be a class of graphs which is efficiently colourable over $\mathcal{C}$. Then* $\text{MC}(\text{FO}, \mathcal{D}) \in \text{FPT}$.

As explained above, this theorem strictly generalises the existing meta-theorem for first-order logic on graph classes of bounded clique-width and also Corollary 5.7 and thereby provides the most general meta-theorem known so far. In particular, it also applies to classes of graphs of unbounded clique-width and which are not nowhere dense.

References

[1] 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, 1990.

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

[3] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 1–3:77–114, 2000.

[4] A. Dawar, M. Grohe, and S. Kreutzer. Locally excluding a minor. In *Logic in Computer Science (LICS)*, pages 270–279, 2007.

[5] A. Dawar and S. Kreutzer. Domination problems in nowhere-dense classes of graphs. In *Foundations of software technology and theoretical computer science (FSTTCS)*, 2009.

[6] M. DeVos, G. Ding, B. Oporowski, D. Sanders, B. Reed, P. Seymour, and D. Vertigan. Excluding any graph as a minor allows a low tree-width 2-coloring. *Journal of Combinatorial Theory, Series B*, 91:25 – 41, 2004.

[7] R. Diestel. *Graph Theory.* Springer-Verlag, 3rd edition, 2005.

[8] R. Downey and M. Fellows. *Parameterized Complexity.* Springer, 1998.

[9] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer, 2nd edition, 1999.

[10] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic.* Springer, 2nd edition, 1994.

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

[12] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006. ISBN 3-54-029952-1.

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

[14] M. Grohe. Logic, graphs, and algorithms. In T. J.Flum, E.Grädel, editor, *Logic and Automata – History and Perspectives.* Amsterdam University Press, 2007.

[15] P. Hlinený and S. il Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*, 38(3):1012–1032, 2008.

[16] W. Hodges. *A shorter model theory.* Cambridge University Press, 1997.

[17] S. Kreutzer. Algorithmic meta-theorems. to appear. See http://web.comlab.ox.ac.uk/people/Stephan.Kreutzer/Publications/amt-survey.pdf, 2008.

[18] S. Kreutzer. On the parameterised intractability of monadic second-order logic. In *Proc. of Computer Science Logic (CSL)*, 2009.

[19] J. Nesetril and P. O. de Mendez. Grad and classes with bounded expansion i. decompositions. *Eur. J. Comb.*, 29(3):760–776, 2008.

[20] J. Nešetřil and P. Ossona de Mendez. On nowhere dense graphs. Technical report, Charles University, 2008.

[21] J. Nešetřil and P. O. de Mendez. Tree depth, subgraph coloring and homomorphisms. *European Journal of Combinatorics*, 2005.

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

ECCC ISSN 1433-8092

http://eccc.hpi-web.de