Jan van den Heuvel, Stephan Kreutzer, Micha l Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, Sebastian Siebertz

Model-checking for successor-invariant first-order formulas on graph classes of bounded expansion

Conference Item

Original citation: Originally presented at 2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, 20-23 June 2017, ACM/IEEE
DOI: 10.1109/LICS.2017.8005115

StKr and RoRa are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC consolidator grant DISTRUCT, agreement No. 648527). MiPi and SeSi are supported by the National Science Centre of Poland via POLONEZ grant agreement UMO-2015/19/P/ST6/03998. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 665778. MiPi is supported by Foundation for Polish Science (FNP) via the START stipend programme. Part of the research for this paper was carried out during a stay of JvdH and DAQu at the Logic and Semantics Research Group of TU Berlin. JvdH and DAQu would like to thank the group for its hospitality

Model-Checking for Successor-Invariant First-Order Formulas on Graph Classes of Bounded Expansion *

Jan van den Heuvel † Stephan Kreutzer ‡ Michał Pilipczuk §
Daniel A. Quiroz † Roman Rabinovich ‡ Sebastian Siebertz §

May 4, 2017

Abstract

A successor-invariant first-order formula is a formula that has access to an auxiliary successor relation on a structure's universe, but the model relation is independent of the particular interpretation of this relation. It is well known that successor-invariant formulas are more expressive on finite structures than plain first-order formulas without a successor relation. This naturally raises the question whether this increase in expressive power comes at an extra cost to solve the model-checking problem, that is, the problem to decide whether a given structure together with some (and hence every) successor relation is a model of a given formula.

It was shown earlier that adding successor-invariance to first-order logic essentially comes at no extra cost for the model-checking problem on classes of finite structures whose underlying Gaifman graph is planar [13], excludes a fixed minor [11] or a fixed topological minor [10, 25]. In this work we show that the model-checking problem for successor-invariant formulas is fixed-parameter tractable on any class of finite structures whose underlying Gaifman graphs form a class of bounded expansion. Our result generalises all earlier results and comes close to the best tractability results on nowhere dense classes of graphs currently known for plain first-order logic.

* StKr and RoRa are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC consolidator grant DISTRUCT, agreement No. 648527). MiPi and SeSi are supported by the National Science Centre of Poland via POLONEZ grant agreement UMO-2015/19/P/ST6/03998. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 665778. MiPi is supported by Foundation for Polish Science (FNP) via the START stipend programme.

Part of the research for this paper was carried out during a stay of JvdH and DAQu at the Logic and Semantics Research Group of TU Berlin. JvdH and DAQu would like to thank the group for its hospitality.

† Department of Mathematics, London School of Economics and Political Science, United Kingdom; {j.van-den-heuvel, d.quiroz}@lse.ac.uk.

‡ Technische Universität Berlin, Germany; {stephan.kreutzer, roman.rabinovich}@tu-berlin.de.

§ Institute of Informatics, University of Warsaw, Poland; {michal.pilipczuk, siebertz}@mimuw.edu.pl.

1 Introduction

Pinpointing the exact complexity of first-order model-checking has been the object of a large body of research. The model-checking problem for first-order logic, denoted MC(FO), is the problem of deciding for a given finite structure $\mathfrak{A}$ and a formula $\varphi \in \mathrm{FO}$ whether $\mathfrak{A} \models \varphi$. Initially, Vardi proposed to distinguish the complexity of MC(FO) into *data*, *formula*, and *combined* complexity. As shown by Vardi [37], for any fixed formula φ the model checking problem is solvable in polynomial time, i.e. the data complexity of MC(FO) is in PTIME. On the other hand, it is also well known that the general model-checking problem for first-order logic, that is, with both the formula and the structure regarded as parts of the input, is PSPACE-complete already on a fixed 2-element structure [3].

A more fine-grained analysis of model-checking complexity can be achieved through the lens of parametrised complexity. In this framework, the model-checking problem $\mathrm{MC}(\mathcal{L})$ for a logic $\mathcal{L}$ is said to be *fixed-parameter tractable* if it can be solved in time $f(|\varphi|) \cdot |\mathfrak{A}|^c$, for some function f (usually required to be computable) and a constant c independent of φ and $\mathfrak{A}$. The complexity class FPT of all fixed-parameter tractable problems is the parametrised analogue to PTIME as model of efficient solvability. Hence, parametrised complexity lies somewhere between data and combined complexity, in that the formula is not taken to be fixed and yet has a different influence on the complexity than the structure. In particular, in the framework of parametrised complexity the complexity of first-order model checking on a specific class $\mathcal{C}$ of structures can be studied in a meaningful way. We will denote MC(FO) restricted to a class $\mathcal{C}$ as $\mathrm{MC}(\mathrm{FO}, \mathcal{C})$.

In general, a parametrised algorithmic problem takes as input a pair (x, k), where x is an instance and k is an integer parameter. The problem is said to be *fixed-parameter tractable* (*FPT* for short) if it can be solved in time $f(k) \cdot |x|^c$, where f and c are as before.

Perhaps the most famous result on the parametrised complexity of model-checking is Courcelle's theorem [4], which states that every algorithmic property on graphs definable in *monadic second-order logic* can be evaluated in linear time on any class of graphs of bounded treewidth. An equivalent statement is that $\mathrm{MC}(\mathrm{MSO}, \mathcal{C})$ is FPT via a linear-time algorithm for any class $\mathcal{C}$ of bounded treewidth. Starting with this foundational result, much work has gone into understanding the complexity of first-order and monadic second-order model-checking with respect to specific classes of graphs or structures. In particular, much of this effort has concentrated on sparse classes of graphs such as planar graphs, graphs of bounded treewidth, graphs of bounded maximum degree, or classes that exclude a fixed (topological) minor. Recently, more abstract notions of sparsity have been considered, namely classes of bounded expansion and nowhere dense classes. Sparse classes of graphs in this sense have in common that if a class $\mathcal{C}$ is sparse and we close it under taking subgraphs, then it is still sparse, i.e. sparse graphs have no dense subgraphs.

It was shown in [26, 27] that Courcelle's theorem cannot be extended in full generality much beyond bounded treewidth. For first-order logic, however, Seese [35] proved that first-order model checking is fixed-parameter tractable on any class of graphs of bounded maximum degree. This result was the starting point of a long series of papers establishing tractability results for first-order model-checking on sparse classes of graphs, see e.g. [5, 8, 14, 16, 21], and see [20] for a survey. This line of research culminated in the theorem of Grohe et al. [21] stating that for any class $\mathcal{C}$ of graphs that is closed under taking subgraphs, $\mathrm{MC}(\mathrm{FO}, \mathcal{C}) \in \mathrm{FPT}$ if and only if $\mathcal{C}$ is nowhere dense. For sparse classes of graphs that are closed under taking subgraphs, this yields a precise characterisation of tractability for first-order model-checking.

The immediate follow-up question is whether this result can be extended in various ways. One line of research tries to extend tractability to even more general or different classes of graphs and structures, see e.g. [17, 18, 19].

In this paper we follow a different route and investigate whether tractability on sparse classes of graphs can be achieved for more expressive logics than FO. It is well-known and easy to see that following the common approach in finite model theory to add fixed-point or reachability operators to FO very quickly results in logics that are not fixed-parameter tractable (under a standard complexity theoretic assumption from parametrised complexity theory), even on planar graphs. Therefore, in this paper we study another classical type of extensions of first-order logic, namely *successor-* and *order-invariant* first-order logic.

Over graphs, an *order-invariant* first-order formula is a formula that, in addition to the edge relation, has access to a linear ordering on the vertex set of the input graph. However, the formula is required to be *invariant* under the precise linear ordering chosen. That is, if the formula is true in a graph G with a linear order $<$, then it must be true in G for all choices of linear orderings on $V(G)$. See Section 3 for details. A formula is *successor-invariant* if the same condition is true for a successor relation instead of a linear ordering.

Successor- and order-invariant first-order logic have both been studied intensively in the literature, see e.g. [1, 12, 29, 30, 32, 33]. However, the difference between the expressive powers of order-invariant, successor-invariant, and plain FO on various classes of structures remains largely unexplored. An unpublished result of Gurevich states that the expressive power of order-invariant FO is stronger than that of plain FO. Rossman [33] proved that successor-invariant FO is more expressive than plain first-order logic. The construction of [33] creates dense instances though, and no separation between successor-invariant FO and plain FO is known on sparse classes, say of bounded expansion. On the other hand, collapse results in this context are known only for very restricted settings. It is known that order-invariant FO collapses to plain FO on trees [1, 28] and on graphs of bounded treedepth [9]. Moreover, order-invariant FO is a subset of MSO on graphs of bounded degree and on graphs of bounded treewidth [1], and more generally, on decomposable graphs in the sense of [12].

In [13], Engelmann et al. study the evaluation complexity of successor and order-invariant first-order logic. They showed that successor-invariant FO is fixed-parameter tractable on planar graphs. This was later generalised in [11] to classes of graphs excluding a fixed minor, and then again to classes of graphs excluding a fixed topological minor in [10]. See also [25] for an independent and different proof of this latter result.

Our contribution. In this paper we narrow the gap between the known tractability results for plain first-order logic and successor-invariant first-order logic. In particular, we show that model-checking successor-invariant FO is fixed-parameter tractable on any class of graphs of bounded expansion. Classes of bounded expansion generalise classes with excluded topological minors, and form a natural meta-class one step below nowhere dense classes of graphs. Thus our result generalises the previous model-checking results of [10, 11, 13, 25].

More precisely, we show that if $\mathcal{C}$ is a class of structures of bounded expansion, then model-checking for successor-invariant first-order formulas on $\mathcal{C}$ can be solved in time $f(|\varphi|) \cdot n \cdot \alpha(n)$, where n is the size of the universe of the given structure, f is some function, and $\alpha(\cdot)$ is the inverse Ackermann function. Note that model-checking for plain first-order logic can be done in linear time on classes of bounded expansion [8], thus the running time of our algorithm is very close to the best known results for plain FO. See Theorem 4 in Section 3 for a precise statement of our main result.

The natural way of proving tractability for successor-invariant FO on a specific class $\mathcal{C}$ of graphs is to show that given any graph $G \in \mathcal{C}$, it can be augmented by a new set F of coloured edges such that a) in (G, F) a successor-relation is first-order definable and b) $G + F$ falls within a class $\mathcal{D}$ of graphs on which plain first-order logic is tractable. In this way, model-checking for successor-invariant FO on the class $\mathcal{C}$ is reduced to the model-checking problem for FO on $\mathcal{D}$. This technique was employed in [10, 11, 13, 25]. The main problem is how to construct the set of augmentation edges F. In [10, 11, 13, 25] the authors used topological arguments, based on Robertson and Seymour's structure theorem for classes with excluded minors [31] or its generalisation by Grohe and Marx to classes with excluded topological minors [22].

For classes of bounded expansion, the object of study in this paper, no such topological methods exist. Instead we rely on a characterisation of bounded expansion classes by *generalised colouring numbers*. The definition of these graph parameters is roughly based on measuring reachability properties in a linear vertex ordering of the input graph. Any such ordering yields a very weak form of decomposition of a graph in terms of an elimination tree. The main technical contribution of this paper is that we find a way to control these elimination trees so that we can use them to define a set F of new edges with the following properties: a) F forms a spanning tree of the input graph G, b) F has maximum degree at most 3, and c) after adding all the edges of F to the graph, the colouring numbers are still bounded. See Theorem 6 in Section 3 for a formal statement of this main technical contribution.

This construction, besides its use in this paper, yields a new insight into the elimination trees generated by colouring numbers. We believe it may prove useful for future research as well.

Organisation. In Section 2 we fix the terminology and notation used throughout the paper and recall the notions from the theory of sparse graphs, in particular the generalised colouring numbers. In Section 3 we show how having access to a low degree spanning tree in a graph can be used to reduce model-checking for successor-invariant FO to model-checking for plain FO, thus proving our main result. Finally, in Section 4 we present out main technical contribution: the construction of a low degree spanning tree that can be added to a graph without increasing the colouring numbers too much.

2 Preliminaries

Notation. By $\mathbb{N}$ we denote the set of nonnegative integers. For a set X, by $\binom{X}{2}$ we denote the set of unordered pairs of elements of X, that is, 2-element subsets of X. By $\alpha(\cdot)$ we denote the inverse Ackermann function.

We use standard graph-theoretical notation; see e.g. [6] for reference. All graphs considered in this paper are finite, simple, and undirected. For a graph G, by $V(G)$ and $E(G) \subseteq \binom{V(G)}{2}$ we denote the vertex and edge sets of G, respectively. For a vertex v and an edge e, we write $v \in G$ and $e \in G$ meaning $v \in V(G)$ and $e \in E(G)$, respectively. A graph H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For a vertex subset $X \subseteq V(G)$, the *subgraph induced by* X is equal to $G[X] = \left(X, E(G) \cap \binom{X}{2}\right)$. For a vertex v, we write $G - v$ for $G[V(G) \setminus \{v\}]$. For a set of unordered pairs $F \subseteq \binom{V(G)}{2}$, by $G + F$ we denote the graph $(V(G), E(G) \cup F)$.

For a nonnegative integer ℓ, a *walk of length* ℓ in G is a sequence $P = (v_1, \ldots, v_{\ell+1})$ of vertices such that $v_i v_{i+1} \in E(G)$ for all $1 \leq i \leq \ell$. Vertices v_i and edges $v_i v_{i+1}$ are *traversed* by the walk P, v_1 and $v_{\ell+1}$ are the *endpoints* of P, while $v_2, \ldots, v_\ell$ are the *internal* vertices of P. A walk P is a *path* if the vertices traversed by it are pairwise different. A walk or path *connects* its endpoints.

A graph G is *connected* if any pair of its vertices can be connected by a path. The *distance* between vertices $u, v \in V(G)$, denoted $\mathrm{dist}_G(u, v)$, is the minimum length of a path between u and v in G. The *radius* of a connected graph G is defined as $\min_{u \in V(G)} \max_{v \in V(G)} \mathrm{dist}_G(u, v)$.

A graph T is a *tree* if it is connected and has no cycles; equivalently, it is connected and has exactly $|V(T)| - 1$ edges. *Rooting* a tree T in some vertex $w \in V(T)$ imposes *child-parent* and *ancestor-descendant* relations in T. More precisely, a vertex v is an *ancestor* of a vertex u if it lies on the unique path from u to the root w, and it is the *parent* of u if it is the immediate successor of u on this path. Thus, each vertex is both an ancestor and a descendant of itself. We say *strict ancestor or descendant* to express that the considered vertices are different.

Shallow minors and bounded expansion. A graph H is a *minor* of G, written $H \preccurlyeq G$, if there are pairwise disjoint connected subgraphs $(I_u)_{u \in V(H)}$ of G, called *branch sets*, such that whenever $uv \in E(H)$, then there are $x_u \in I_u$ and $x_v \in I_v$ with $x_u x_v \in E(G)$. We call the family $(I_u)_{u \in V(H)}$ a *minor model* of H in G. A graph H is a *depth-r minor* of G, denoted $H \preccurlyeq_r G$, if there is a minor model $(I_u)_{u \in V(H)}$ of H in G such that each subgraph I_u has radius at most r.

For a graph H, we write $d(H)$ for the *average degree* of H, that is, for the number $2|E(H)|/|V(H)|$. A class of graphs $\mathcal{C}$ has *bounded expansion* if there is a function $f \colon \mathbb{N} \to \mathbb{N}$ such that for all nonnegative integers r, we have $d(H) \leq f(r)$ for every $H \preccurlyeq_r G$ with $G \in \mathcal{C}$.

Generalised colouring numbers. In this paper we will not rely on the above, original, definition of classes of bounded expansion, but on their alternative characterisation via *generalised colouring numbers*. Let us fix a graph G. By $\Pi(G)$ we denote the set of all linear orderings of $V(G)$. For $L \in \Pi(G)$, we write $u <_L v$ if u is smaller than v in L, and $u \leq_L v$ if $u <_L v$ or $u = v$.

For a nonnegative integer r, we say that a vertex u is *strongly r-reachable* from a vertex v with respect to L, if $u \leq_L v$ and there is a path P of length at most r that starts in v, ends in u, and all its internal vertices are larger than v in L. By $\mathrm{SReach}_r[G, L, v]$ we denote the set of vertices that are strongly r-reachable from v with respect to L. Note that $v \in \mathrm{SReach}_r[G, L, v]$ for any vertex v.

We define the *r-colouring number* of G (with respect to L) as follows:

$$\mathrm{col}_r(G, L) = \max_{v \in V(G)} |\mathrm{SReach}_r[G, L, v]| \qquad \text{and} \qquad \mathrm{col}_r(G) = \min_{L \in \Pi(G)} \mathrm{col}_r(G, L).$$

For a nonnegative integer r and ordering $L \in \Pi(G)$, the *r-admissibility* $\mathrm{adm}_r[G, L, u]$ of a vertex v with respect to L is defined as the maximum size of a family $\mathcal{P}$ of paths that satisfies the following two properties:

- each path $P \in \mathcal{P}$ has length at most r, starts in v, ends in a vertex that is smaller than v in L, and all its internal vertices are larger than v in L;
- the paths in $\mathcal{P}$ are pairwise vertex-disjoint, apart from sharing the start vertex v.

The *r-admissibility* of G (with respect to L) is defined similarly to the r-colouring number:

$$\mathrm{adm}_r(G, L) = \max_{v \in V(G)} \mathrm{adm}_r[G, L, v] \qquad \text{and} \qquad \mathrm{adm}_r(G) = \min_{L \in \Pi(G)} \mathrm{adm}_r(G, L).$$

The r-colouring numbers were introduced by Kierstead and Yang [24], while r-admissibility was first studied by Dvořák [7]. It was shown that those parameters are related as follows.

Lemma 1 (Dvořák [7])
For each graph G, nonnegative integer r, and vertex ordering $L \in \Pi(G)$, we have

$$\mathrm{adm}_r(G, L) \leq \mathrm{col}_r(G, L) \leq (\mathrm{adm}_r(G, L))^r.$$

We remark that in Dvořák's work, the reachability sets never include the starting vertex, hence the above inequality is stated slightly different in [7].

As proved by Zhu [38], the generalised colouring numbers are tightly related to densities of low-depth minors, and hence they can be used to characterise classes of bounded expansion.

Theorem 2 (Zhu [38])
A class $\mathcal{C}$ of graphs has bounded expansion if and only if there is a function $f \colon \mathbb{N} \to \mathbb{N}$ such that $\mathrm{col}_r(G) \leq f(r)$ for all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$.

By Lemma 1, we may equivalently demand that there is a function $f \colon \mathbb{N} \to \mathbb{N}$ such that $\mathrm{adm}_r(G) \leq f(r)$ for all nonnegative integers r and all $G \in \mathcal{C}$.

As shown by Dvořák [7], on classes of bounded expansion one can compute $\mathrm{adm}_r(G)$ in linear fixed-parameter time, parametrised by r. More precisely, we have the following.

Theorem 3 (Dvořák [7])
Let $\mathcal{C}$ be a class of bounded expansion. Then there is an algorithm that, given a graph $G \in \mathcal{C}$ and a nonnegative integer r, computes a vertex ordering $L \in \Pi(G)$ with $\mathrm{adm}_r(G, L) = \mathrm{adm}_r(G)$ in time $f(r) \cdot |V(G)|$, for some computable function f.

We remark that Dvořák states the result in [7] as the existence of a linear-time algorithm for each fixed value of r. However, an inspection of the proof reveals that it is actually a single fixed-parameter algorithm that can take r as input. To the best of our knowledge, a similar result for computing $\mathrm{col}_r(G)$ is not known, but by Lemma 1 we can use admissibility to obtain an approximation of the r-colouring number of a given graph from a class of bounded expansion.

3 Model-checking

We start by introducing successor-invariant first-order formulas and stating the main theorem (Theorem 4) formally. Next, we show how to prove this theorem assuming our main technical result, Theorem 6. The proof follows by a combination of several tools borrowed from the literature.

Successor-invariant first-order formulas. A *finite and purely relational signature* τ is a finite set $\{R_1, \ldots, R_k\}$ of relation symbols, where each relation symbol R_i has an associated arity a_i. A *finite* τ*-structure* $\mathfrak{A}$ consists of a finite set A (the *universe* of $\mathfrak{A}$) and a relation $R_i(\mathfrak{A}) \subseteq A^{a_i}$ for each relation symbol $R_i \in \tau$. If $\mathfrak{A}$ is a finite τ-structure, then the *Gaifman graph* of $\mathfrak{A}$, denoted $G(\mathfrak{A})$, is the graph on the vertex set A in which two elements $u, v \in A$ are adjacent if and only if $u \neq v$ and u and v appear together in some relation $R_i(\mathfrak{A})$ of $\mathfrak{A}$. We say that a class $\mathcal{C}$ of finite τ-structures has *bounded expansion* if the graph class $G(\mathcal{C}) = \{G(\mathfrak{A}) \;:\; \mathfrak{A} \in \mathcal{C}\}$ has bounded expansion. Similarly, for $r \in \mathbb{N}$, we write $\mathrm{adm}_r(\mathfrak{A})$ for $\mathrm{adm}_r(G(\mathfrak{A}))$, etc.

Let V be a set. A *successor relation* on V is a binary relation $S \subseteq V \times V$ such that (V, S) is a directed path of length $|V| - 1$. Let τ be a finite relational signature. A formula $\varphi \in \mathrm{FO}[\tau \cup \{S\}]$ is *successor-invariant* if for all τ-structures $\mathfrak{A}$ and for all successor relations S_1, S_2 on $V(\mathfrak{A})$ it holds that $(\mathfrak{A}, S_1) \models \varphi \iff (\mathfrak{A}, S_2) \models \varphi$. We denote the set of all such successor-invariant first-order formulas by $\mathrm{FO}[\tau_{\mathrm{succ}}]$. Note that the set $\mathrm{FO}[\tau_{\mathrm{succ}}]$ is not decidable, and hence one usually does not speak of successor-invariant first-order logic, as for a logic one usually requires a decidable syntax [23]. For any $\varphi \in \mathrm{FO}[\tau_{\mathrm{succ}}]$ and any τ-structure $\mathfrak{A}$, we denote $\mathfrak{A} \models_{\mathrm{succ-inv}} \varphi$ if for any (equivalently, every) successor relation S on the universe of $\mathfrak{A}$ it holds that $(\mathfrak{A}, S) \models \varphi$.

With these definitions in mind, we can finally state our main result formally.

Theorem 4
Let τ be a finite and purely relational signature and let $\mathcal{C}$ be a class of τ-structures of bounded expansion. Then there exists an algorithm that, given a finite τ-structure $\mathfrak{A} \in \mathcal{C}$ and a formula $\varphi \in \mathrm{FO}[\tau_{\mathrm{succ}}]$, verifies whether $\mathfrak{A} \models_{\mathrm{succ-inv}} \varphi$ in time $f(|\varphi|) \cdot n \cdot \alpha(n)$, where f is a function and n is the size of the universe of $\mathfrak{A}$.

In the language of parametrised complexity, Theorem 4 essentially states that the model-checking problem for successor-invariant first-order formulas is fixed-parameter tractable on classes of finite structures whose underlying Gaifman graph belongs to a fixed class of bounded expansion. There is a minor caveat, though. The formal definition of fixed-parameter tractability, see e.g. [15], requires the function f to be computable, which is not asserted by Theorem 4. In order to have this property, it suffices to assume that the class $\mathcal{C}$ is *effectively of bounded expansion*. In the characterisation of Theorem 2, this means that there exist a *computable* function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{col}_r(\mathfrak{A}) \leq f(r)$ for each $\mathfrak{A} \in \mathcal{C}$. See [21] for a similar discussion regarding model-checking first-order logic on (effectively) nowhere dense classes of graphs.

As we mentioned in Section 1, fixed-parameter tractability of model-checking successor-invariant FO has been shown earlier for planar graphs [13], graphs excluding a fixed minor [11], and graphs excluding a fixed topological minor [10, 25]. As all the above-mentioned classes have (effectively) bounded expansion, Theorem 4 thus generalises all the previously known results in this area. Let us remark that on general relational structures, the model-checking problem for plain first-order logic is complete for the parametrised complexity class AW[$\star$], and hence unlikely to be fixed-parameter tractable [15].

From a spanning tree to a successor relation. In principle, our approach follows that of all earlier results on successor-invariant model-checking. As we would like to check whether $\mathfrak{A} \models_{\mathrm{succ-inv}} \varphi$, we may compute an arbitrary successor relation S on the universe of $\mathfrak{A}$, and verify whether $(\mathfrak{A}, S) \models \varphi$. Of course, we will try to compute a successor relation S so that adding it to $\mathfrak{A}$ preserves the structural properties as much as possible, so that model-checking on $(\mathfrak{A}, S)$ can be done efficiently. Ideally, if $G(\mathfrak{A})$ contained a Hamiltonian path, we could add a successor relation without introducing any new edges to $G(\mathfrak{A})$. However, in general this might be impossible.

The other helpful ingredient is that we do not actually have to add a successor relation, but it suffices to add some structural information so that a first-order formula can interpret a successor relation. This approach is known as the interpretation method [20] and can be used to reduce successor-invariant model-checking to the plain first-order case. In our concrete case, Eickmeyer et al. [11] have shown that adding a spanning tree of constant maximum degree is enough to be able to interpret some successor relation. Here, for relational signatures $\tau' \supseteq \tau$, a τ'-structure $\mathfrak{A}'$ is a τ'-*expansion* of a τ-structure $\mathfrak{A}$ if after dropping relations $R(\mathfrak{A}')$ for relation symbols $R \in \tau' \setminus \tau$, $\mathfrak{A}'$ becomes equal to $\mathfrak{A}$.

Lemma 5 (Lemma 4.4 of Eickmeyer et al. [11], adjusted)
Let τ be a finite and purely relational signature, and let k be a positive integer. Suppose we are given a finite τ-structure $\mathfrak{A}$, together with a spanning tree T of $G(\mathfrak{A})$ with maximum degree at most k. Then there is a finite and purely relational signature τ_k and a first-order formula $\psi^{(k)}_{\mathrm{succ}}(x, y)$, both depending only on k, and a $(\tau \cup \tau_k)$-expansion $\mathfrak{A}'$ of $\mathfrak{A}$, such that

- *the Gaifman graphs of $\mathfrak{A}'$ and $\mathfrak{A}$ are equal;*
- *$\psi^{(k)}_{\mathrm{succ}}(x, y)$ defines a successor relation on $\mathfrak{A}'$.*

Moreover, for a fixed k and given $\mathfrak{A}$ and T, one can compute $\mathfrak{A}'$ in time linear in the size of the input.

Informally speaking, the structure τ' contains the edges that form the spanning tree, using a new colour. Important for us is that the construction guarantees that the Gaifman graphs of $\mathfrak{A}$ and $\mathfrak{A}'$ are the same, and thus so are their structural properties.

Lemma 5 differs from the original statement of Eickmeyer et al. [11] in two ways. First, Eickmeyer et al. [11] state the running time only as polynomial, however a verification of the proof yields that the straightforward implementation runs in linear time. Second, Eickmeyer et al. [11] require the existence of a k-walk instead of a spanning tree of maximum degree k. Here, a k-*walk* is a walk in the graph that visits every vertex at least once and at most k times. Observe that if a graph has a spanning tree of maximum degree at most k, then in particular it contains a k-walk that can be computed from the given spanning tree in polynomial time by performing a depth-first search on the tree. Thus, the assumption of having a spanning tree of maximum degree k is sufficient for Lemma 5 to work. We find working with spanning trees more natural than with k-walks, however both approaches are essentially equivalent: if a graph admits a k-walk, then it has a spanning tree of maximum degree at most $2k$.

As discussed above, Lemma 5 essentially reduces model-checking successor-invariant first-order formulas to plain first-order logic, provided we can expose some spanning tree with maximum degree bounded by a constant in the Gaifman graph of the given structure. In general this might be not possible; e.g. the Gaifman graph could be a star. Therefore, the idea is to add a carefully constructed binary relation to the structure so that such a low-degree spanning tree can be found, while maintaining the property that the structure still belongs to a class where model-checking first-order logic is fixed-parameter tractable. This approach was used in all the previous works [10, 11, 25], and, with a small twist, we will use it also here. More precisely, in the next section we will prove the following theorem which gives a construction of a low-degree spanning tree that can be added to a graph without increasing its colouring numbers too much.

Theorem 6
There exists an algorithm that, given a graph G and an ordering L of $V(G)$, computes a set of unordered pairs $F \subseteq \binom{V(G)}{2}$ such that the graph $T = (V(G), F)$ is a tree of maximum degree at most 3 and

$$\mathrm{adm}_r(G + F, L) \leq 2 + 3 \cdot \mathrm{col}_{2r}(G, L).$$

The running time of the algorithm is $\mathcal{O}((m + n) \cdot \alpha(m))$, where $m = |E(G)|$ and $n = |V(G)|$.

We remark that in the earlier work [10, 11, 25], the bound on the maximum degree of the constructed spanning tree was a constant depending on the size of the excluded (topological) minor, while Theorem 6 always bounds the maximum degree of the spanning tree by 3.

Model-checking plain FO. Before showing how Theorem 4 follows from Theorem 6, we first need to draw upon the literature on model-checking first-order logic on classes of bounded expansion. The following statement encapsulates the model-checking results of Dvořák et al. [8] and of Grohe and Kreutzer [20]. However, it is slightly stronger than the statements claimed in [8, 20]. We will later argue how this statement follows from the approach presented in these works.

Theorem 7
Let τ be a finite and purely relational signature. Then for every formula $\varphi \in \mathrm{FO}[\tau]$ there exists a nonnegative integer $r(\varphi)$, computable from φ, such that the following holds. Given a τ-structure $\mathfrak{A}$, it can be verified whether $\mathfrak{A} \models \varphi$ in time $f(|\varphi|, \mathrm{col}_{r(\varphi)}(\mathfrak{A})) \cdot n$, where n is the size of the universe of $\mathfrak{A}$ and f is a computable function.

Observe that if $\mathfrak{A}$ is drawn from a fixed class of bounded expansion $\mathcal{C}$, then $\mathrm{adm}_{r(\varphi)}(\mathfrak{A})$ is a parameter depending only on φ, hence we recover fixed-parameter tractability of model-checking FO on any class of bounded expansion, parametrised by the length of the formula. Theorem 7 is stronger than this latter statement in that it says that the input structure does not need to be drawn from a fixed class of bounded expansion, where the colouring number is bounded in terms of the radius r for all values of r, but it suffices to have a bound on the colouring numbers up to some radius $r(\varphi)$, which depends only on the formula φ. We need this strengthening in our algorithm for the following reason: When adding a low-degree spanning tree to the Gaifman graph, we are not able to control all the colouring numbers at once, but only for some particular value of the radius. Theorem 7 ensures that this is sufficient for the model-checking problem to remain tractable.

We now sketch how Theorem 7 may be derived from the works of Dvořák et al. [8] and of Grohe and Kreutzer [20]. We prefer to work with the algorithm of Grohe and Kreutzer [20], because we find it conceptually simpler. For a given quantifier rank q and an nonnegative integer $i \le q$, the algorithm computes the set of all *types* $\mathfrak{R}_i^q$ realised by i-tuples in the input structure $\mathfrak{A}$: for a given i-tuple of elements $\overline{a}$, its *type* is the set of all FO formulas $\varphi(\overline{x})$ with i free variables and quantifier rank at most $q - i$ for which $\varphi(\overline{a})$ holds. Note that for $i = 0$, this corresponds to the set of sentences of quantifier rank at most q that hold in the structure, from which the answer to the model-checking problem can be directly read; whereas for $i = q$, we consider quantifier-free formulas with q free variables. Essentially, $\mathfrak{R}_q^q$ is computed explicitly, and then one inductively computes $\mathfrak{R}_i^q$ based on $\mathfrak{R}_{i+1}^q$. The above description is, however, a bit too simplified, as each step of the inductive computation introduces new relations to the structure, but does not change its Gaifman graph; we will explain this later.

When implementing the above strategy, the assumption that the structure is drawn from a class of bounded expansion is used via *treedepth-p colourings*, a colouring notion functionally equivalent to the generalised colouring numbers. More precisely, a treedepth-p colouring of a graph G is a colouring $\gamma : V(G) \to \Gamma$, where Γ is a set of colours, such that for any subset $C \subseteq \Gamma$ of i colours, $i \le p$, the vertices with colours from C induce a subgraph of treedepth at most i. The *treedepth-p chromatic number* of a graph G, denoted $\chi_p(G)$, is the smallest number of colours $|\Gamma|$ needed for a treedepth-p colouring of G. As proved by Zhu [38], the treedepth-p chromatic numbers are bounded in terms of r-colouring numbers as follows:

Theorem 8 (Zhu [38])
For any graph G and $p \in \mathbb{N}$ we have

$$\chi_p(G) \le \left(\mathrm{col}_{2^{p-2}}(G)\right)^{2^{p-2}}.$$

Moreover, an appropriate treedepth-p colouring can be constructed in polynomial time from an ordering $L \in \Pi(G)$, certifying an upper bound on $\mathrm{col}_{2^{p-2}}(G)$.

The computation of both $\mathfrak{R}_q^q$ and $\mathfrak{R}_i^q$ from $\mathfrak{R}_{i+1}^q$ is done by rewriting every possible type as a purely existential formula. Each rewriting step, however, enriches the signature by unary relations corresponding to colours of some treedepth-p colouring γ, as well as binary relations representing edges of appropriate treedepth decompositions certifying that γ is correct. However, the binary relations are added in a way that the Gaifman graph of the structure remains intact. For us it is important that in all the steps, the parameter p used for the definition of γ depends only on q and i in a computable manner. Thus, by Theorem 8, to ensure that γ uses a bounded number of colours, we only need to ensure the boundedness of $\mathrm{col}_{r(q)}(\mathfrak{A})$ for some computable function $r(q)$. By taking q to be the quantifier rank of the input formula, the statement of Theorem 7 follows.

From a spanning tree to model-checking. We can now combine all the ingredients and show how our main result follows from Theorem 6.

Proof (of Theorem 4, assuming Theorem 6) Consider the signature τ_3 and the first-order formula $\psi^{(3)}_{\mathrm{succ}}(x, y)$ given by Lemma 5 for $k = 3$. Let $\tau' = \tau \cup \tau_3 \cup \{T\}$, where T is a binary relation symbol not used in $\tau \cup \tau_3$. Let $\varphi' \in \mathrm{FO}[\tau']$ be constructed from the input formula φ by replacing each usage of the successor relation $S(x, y)$ with the formula $\psi^{(3)}_{\mathrm{succ}}(x, y)$.

Using Theorem 7 for the signature τ', compute the value of $r = r(\varphi')$. Next, using the algorithm of Theorem 3, compute a vertex ordering $L \in \Pi(\mathfrak{A})$ such that $\mathrm{adm}_{2r}(\mathfrak{A}, L) = \mathrm{adm}_{2r}(\mathfrak{A})$. Apply Theorem 6 to $G(\mathfrak{A})$ and L, thus obtaining a tree T with the universe of $\mathfrak{A}$ as the vertex set, such that the maximum degree in T is at most 3 and

$$\mathrm{adm}_r(\mathfrak{A}_T) \le \mathrm{adm}_r(\mathfrak{A}_T, L) \le 2 + 2 \cdot \mathrm{col}_{2r}(\mathfrak{A}, L) \le 2 + 2(\mathrm{adm}_{2r}(\mathfrak{A}, L))^{2r} = 2 + 2(\mathrm{adm}_{2r}(\mathfrak{A}))^{2r}.$$

Here $\mathfrak{A}_T$ is the $\tau \cup \{T\}$-extension of $\mathfrak{A}$ obtained by adding the edges of T as a binary relation. Next, apply the algorithm of Lemma 5 to $\mathfrak{A}_T$ and $k = 4$, thus computing a τ'-structure $\mathfrak{A}'$ with the same Gaifman graph as $\mathfrak{A}_T$, in which $\psi^{(3)}_{\mathrm{succ}}(x, y)$ defines a successor relation. It is then clear that

$$\mathfrak{A} \models_{\mathrm{succ-inv}} \varphi \quad \iff \quad \mathfrak{A}' \models \varphi'.$$

It remains to apply the algorithm of Theorem 7 to $\mathfrak{A}'$ and φ', which runs in time $f(|\varphi|) \cdot n$ due to the bound on $\mathrm{adm}_r(\mathfrak{A}_T) = \mathrm{adm}_r(\mathfrak{A}')$. Observe that all the other steps also work in time $f(|\varphi|) \cdot n$, apart from the application of the algorithm of Theorem 6, which takes time $\mathcal{O}((m + n) \cdot \alpha(m))$, where $m = |E(G(\mathfrak{A}))|$. However, in classes of bounded expansion the number of edges is bounded linearly in the number of vertices, hence the time complexity analysis follows. $\square$

4 Constructing a low-degree spanning tree

In this section we prove Theorem 6. The main step towards this goal is the corresponding statement for connected graphs, as expressed in the following lemma.

Lemma 9
There exists an algorithm that, given a connected graph G and a vertex ordering L of G, computes a set of unordered pairs $F \subseteq \binom{V(G)}{2}$ such that the graph $T = (V(G), F)$ is a tree of maximum degree at most 3 and

$$\mathrm{adm}_r(G + F, L) \le 3 \cdot \mathrm{col}_{2r}(G, L).$$

The running time of the algorithm is $\mathcal{O}(m \cdot \alpha(m))$, where $m = |E(G)|$.

We first show that Theorem 6 follows easily from Lemma 9.

Proof (of Theorem 6, assuming Lemma 9) Let G be a (possibly disconnected) graph, and let $G_1, \ldots, G_p$ be the connected components of G. For each $i \in \{1, \ldots, p\}$, let L_i be the ordering obtained by restricting L to $V(G_i)$. Obviously $\mathrm{col}_{2r}(G_i, L_i) \leq \mathrm{col}_{2r}(G, L)$.

Apply the algorithm of Lemma 9 to G_i and L_i, obtaining a subset of unordered pairs F_i such that $T_i = (V(G_i), F_i)$ is a tree of maximum degree at most 3 and

$$\mathrm{adm}_r(G_i + F_i, L_i) \leq 3 \cdot \mathrm{col}_{2r}(G_i, L_i) \leq 3 \cdot \mathrm{col}_{2r}(G, L).$$

For each $i \in \{1, \ldots, p\}$, select a vertex v_i of G_i with degree at most 1 in T_i; since T_i is a tree, such a vertex exists. Define

$$F = \{v_1 v_2, v_2 v_3, \ldots, v_{p-1} v_p\} \cup \bigcup_{i=1}^{p} F_i.$$

Obviously we have that $T = (V(G), F)$ is a tree. Observe that it has maximum degree at most 3. This is because each vertex v_i had degree at most 1 in its corresponding tree T_i, and hence its degree can grow to at most 3 after adding edges $v_{i-1}v_i$ and $v_i v_{i+1}$. By Lemma 9, the construction of T takes time $\mathcal{O}(m + n\alpha(n))$.

It remains to show that $\mathrm{adm}_r(G + F, L) \leq 2 + 3 \cdot \mathrm{col}_{2r}(G, L)$. Take any vertex u of G, say $u \in V(G_i)$, and let $\mathcal{P}$ be a set of paths of length at most r that start in u, are pairwise vertex-disjoint (apart from u), and end in vertices smaller than u in L while internally traversing only vertices larger than u in L. Observe that at most two of the paths from $\mathcal{P}$ can use any of the edges from the set $\{v_1 v_2, v_2 v_3, \ldots, v_{p-1} v_p\}$, since any such path has to use either $v_{i-1} v_i$ or $v_i v_{i+1}$. The remaining paths are entirely contained in $G_i + F_i$, and hence their number is bounded by $\mathrm{adm}_r(G_i + F_i, L_i) \leq 3\mathrm{col}_{2r}(G, L)$. The theorem follows. $\square$

In the remainder of this section we focus on Lemma 9.

Proof (of Lemma 9) We begin our proof by showing how to compute the set F. This will be a two step process, starting with an *elimination tree*. For a connected graph G and an ordering L of $V(G)$, we define the *(rooted) elimination tree* $S(G, L)$ of G imposed by L (cf. [2, 34]) as follows. If $V(G) = \{v\}$, then the rooted elimination tree $S(G, L)$ is just the tree on the single vertex v. Otherwise, the root of $S(G, L)$ is the vertex w that is the smallest with respect to the ordering L in G. For each connected component C of $G - w$ we construct a rooted elimination tree $S(C, L|_{V(C)})$, where $L|_{V(C)}$ denotes the restriction of L to the vertex set of C. These rooted elimination trees are attached below w as subtrees by making their roots into children of w. Thus, the vertex set of the elimination tree $S(G, L)$ is always equal to the vertex set of G. See Figure 1 for an illustration. The solid black lines are the edges of G; the dashed blue lines are the edges of S. The ordering L is given by the numbers written in the vertices.

Let $S = S(G, L)$ be the rooted elimination tree of G imposed by L. For a vertex u, by G_u we denote the subgraph of G induced by all descendants of u in S, including u. The following properties follow easily from the construction of a rooted elimination tree.

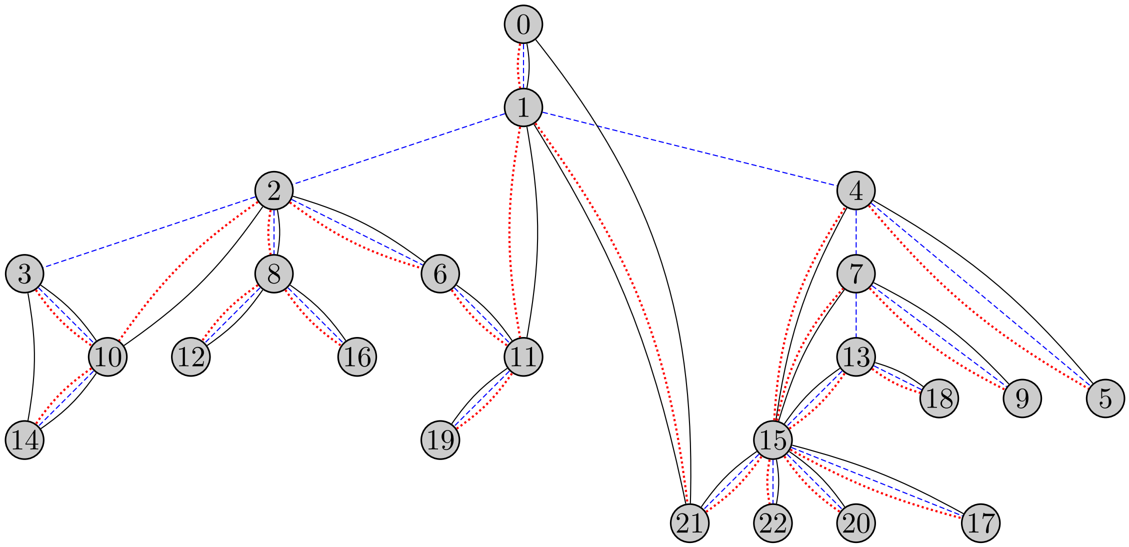

Figure 1: A graph G (solid black lines), the elimination tree S (dashed blue lines), and the tree U (dotted red lines). Numbering of nodes reflects the ordering L.

Claim 1 *The following assertions hold.*

1. *For each $u \in V(G)$, the subgraph G_u is connected.*
2. *Whenever a vertex u is an ancestor of a vertex v in S, we have $u \leq_L v$.*
3. *For each $uv \in E(G)$ with $u <_L v$, u is an ancestor of v in S.*
4. *For each $u \in V(G)$ and each child v of u in S, u has at least one neighbour in $V(G_v)$.*

Proof Assertions 1 and 2 follow immediately from the construction of S. For assertion 3, suppose that u and v are not bound by the ancestor-descendant relation in S, and let w be their lowest common ancestor in S. Then u and v would be in different connected components of $G_w - w$, hence uv could not be an edge; a contradiction. It follows that u and v are bound by the ancestor-descendant relation, implying that u is an ancestor of v, due to $u <_L v$ and assertion 2. Finally, for assertion 4, recall that by assertion 1 we have that G_u is connected, whereas by construction G_v is one of the connected components of $G_u - u$. Hence, in G there is no edge between $V(G_v)$ and any of the other connected components of $G_u - u$. If there was no edge between $V(G_v)$ and u as well, then there would be no edge between $V(G_v)$ and $V(G_u) \setminus V(G_v)$, contradicting the connectivity of G_u. ⌟

We now define a set of edges $B \subseteq E(G)$ as follows. For every vertex u of G and every child v of u in S, select an arbitrary neighbour $w_{u,v}$ of u in G_v; such a neighbour exists by Claim 1.4. Then let B_u be the set of all edges $uw_{u,v}$, for v ranging over the children of u in S. Define

$$B = \bigcup_{u \in V(G)} B_u.$$

Let U be the graph spanned by all the edges in B, that is, $U = (V(G), B)$. In Figure 1, the edges of U are represented by the dotted red lines.

Claim 2 *The graph U is a tree.*

Proof Observe that for each $u \in V(G)$, the number of edges in B_u is equal to the number of children of u in S. Since every vertex of G has exactly one parent in S, apart from the root of S, we infer that

$$|B| \leq \sum_{u \in V(G)} |B_u| = |V(G)| - 1.$$

Therefore, since B is the edge set of U, to prove that U is a tree it suffices to prove that U is connected. To this end, we prove by a bottom-up induction on S that for each $u \in V(G)$, the subgraph $U_u = \left(V(G_u), B \cap \binom{V(G_u)}{2}\right)$ is connected. Note that for the root w of S this claim is equivalent to $U_w = U$ being connected.

Take any $u \in V(G)$, and suppose by induction that for each child v of u in S, the subgraph U_v is connected. Observe that U_u can be constructed by taking the vertex u and, for each child v of u in S, adding the connected subgraph U_v and connecting it to u via edge $uw_{u,v} \in B_u$. Thus, U_u constructed in this manner is also connected, as claimed. ⌟

Next, we verify that U can be computed within the claimed running time. Note that we do not need to compute S, as we will use it only in the analysis. We remark that this is the only place in the algorithm where the running time is not linear.

Claim 3 *The tree U can be computed in time $\mathcal{O}(m \cdot \alpha(m))$.*

Proof We use the classic Find & Union data structure on the set $V(G)$. Recall that in this data structure, at each moment we maintain a partition of $V(G)$ into a number of equivalence classes, each with a prescribed representative, where initially each vertex is in its own class. The operations are a) for a given $u \in V(G)$, find a representative of the class to which u belongs, and b) merge two equivalence classes into one. Tarjan [36] gave an implementation of this data structure where both operations run in amortised time $\alpha(k)$, where k is the total number of operations performed.

Having initialised the data structure, we process the vertex ordering L from the smallest end, starting with an empty suffix. For an already processed suffix X of L, the maintained classes within X will represent the partition of $G[X]$ into connected components, while every vertex outside X will still be in its own equivalence class. Let us consider one step, when we process a vertex u, thus moving from a suffix X to the suffix $X' = X \cup \{u\}$. Iterate through all the neighbours of u, and for each neighbour v of u such that $u <_L v$, verify whether the equivalence classes of u and v are different. If this is the case, merge these classes and add the edge uv to B. A straightforward induction shows that the claimed invariant holds. Moreover, when processing u we add exactly the edges of B_u to B, hence at the end we obtain the set B and the tree $U = (V(G), B)$.

For the running time analysis, observe that in total we perform $\mathcal{O}(m)$ operations on the data structure, thus the running time is $\mathcal{O}(m\alpha(m))$. We remark that we assume that the ordering L is given as a bijection between $V(G)$ and numbers $\{1, 2, \ldots, |V(G)|\}$, thus for two vertices u, v we can check in constant time whether $u <_L v$. ⌟

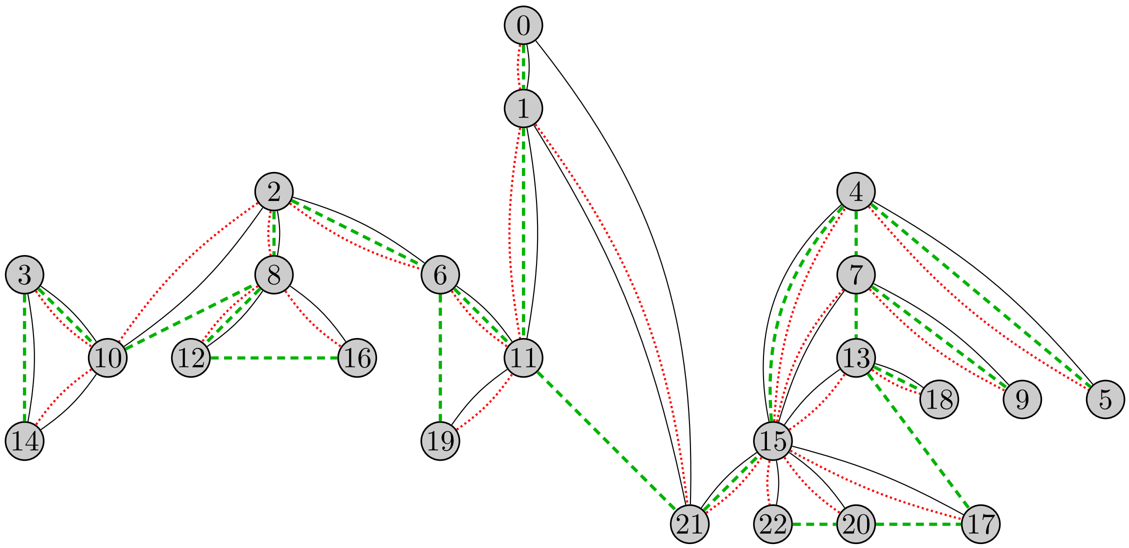

Figure 2: A graph G (solid black lines), the tree U (dotted red lines), and the tree T (thick dashed green lines).

By Claim 2 we have that U is a spanning tree of G, however its maximum degree may be (too) large. The idea is to use U to construct a new tree T with maximum degree at most 3 (on the same vertex set $V(G)$). The way we constructed U will enable us to argue that adding the edges of T to the graph G does not change the generalised colouring numbers too much.

Give U the same root as the elimination tree S. From now on we treat U as a rooted tree, which imposes parent-child and ancestor-descendant relations in U as well. Note that the parent-child and ancestor-descendant relations in S and in U may be completely different. For instance, consider vertices 4 and 15 in the example from Figure 1: 4 is a child of 15 in U, and an ancestor of 15 in S.

For every $u \in V(G)$, let $(x_1, \ldots, x_p)$ be an enumeration of the children of u in U, such that $x_i <_L x_j$ if $i < j$. Let $F_u = \{ux_1, x_1x_2, x_2x_3, \ldots, x_{p-1}x_p\}$, and define

$$F = \bigcup_{u \in V(G)} F_u \quad \text{and} \quad T = (V(G), F).$$

See Figure 2 for an illustration.

Claim 4 *The graph T is a tree with maximum degree at most* 3.

Proof Observe that for each $u \in V(G)$, we have that $|F_u|$ is equal to the number of children of u in U. Every vertex of G apart from the root of U has exactly one parent in U, hence

$$|F| \leq \sum_{u \in V(G)} |F_u| = V(G)| - 1.$$

Therefore, to prove that T is a tree, it suffices to argue that it is connected. This, however, follows immediately from the fact that U is connected, since for each edge in U there is a path in T that connects the same pair of vertices.

Finally, it is easy to see that each vertex u is incident to at most 3 edges of F: at most one leading to a child of u in U, and at most 2 belonging to F_v, where v is the parent of u in U. $\lrcorner$

Observe that once the tree U is constructed, it is straightforward to construct T in time $\mathcal{O}(n)$. Thus, it remains to check that adding F to G does not change the generalised colouring numbers too much.

Take any vertex $u \in V(G)$ and examine its children in U. We partition them as follows. Let $Z_u^{\uparrow}$ be the set of those children of u in U that are its ancestors in S, and let $Z_u^{\downarrow}$ be the set of those children of u in U that are its descendants in S. By the construction of U and by Claim 1.3, each child of u in U is either its ancestor or descendant in S. By Claim 1.2, this is equivalent to saying that $Z_u^{\uparrow}$, respectively $Z_u^{\downarrow}$, comprise the children of u in U that are smaller, respectively larger, than u in L. Note that by the construction of U, the vertices of $Z_u^{\downarrow}$ lie in pairwise different subtrees rooted at the children of u in S, thus u is the lowest common ancestor in S of every pair of vertices from $Z_u^{\downarrow}$. On the other hand, all vertices of $Z_u^{\uparrow}$ are ancestors of u in S, thus every pair of them is bound by the ancestor-descendant relation in S.

Claim 5 *The graph $G + F$ satisfies the following inequality:* $\mathrm{adm}_r(G + F, L) \leq 3 \cdot \mathrm{col}_{2r}(G, L)$.

Proof Write $H = G + F$. Let $F_{\mathrm{new}} = F \setminus E(G)$ be the set of edges from F that were not already present in G. If an edge $e \in F_{\mathrm{new}}$ belongs also to F_u for some $u \in V(G)$, then we know that u cannot be an endpoint of e. This is because each edge joining a vertex u with one of its children in U is already present in G. We say that the vertex u is the *origin* of an edge $e \in F_{\mathrm{new}} \cap F_u$, and denote it by $a(e)$. Observe that then both endpoints of e are the children of $a(e)$ in the tree U, and hence $a(e)$ is adjacent to both the endpoints of e in G.

To give an upper bound on $\mathrm{adm}_r(H, L)$, let us fix a vertex $u \in V(G)$ and a family of paths $\mathcal{P}$ in H such that

- each path in $\mathcal{P}$ has length at most r, starts in u, ends in a vertex smaller than u in L, and all its internal vertices are larger than u in L;
- the paths in $\mathcal{P}$ are pairwise vertex-disjoint, apart from the starting vertex u.

For each path $P \in \mathcal{P}$, we define a walk P' in G as follows. For every edge $e = xy$ from F_{new} traversed on P, replace the usage of this edge on P by the following detour of length 2: $x{-}a(e){-}y$. Observe that P' is a walk in the graph G, it starts in u, ends in the same vertex as P, and has length at most $2r$. Next, we define $v(P)$ to be the first vertex on P' (that is, the closest to u on P') that does not belong to G_u. Since the endpoint of P' that is not u does not belong to G_u, such a vertex exists. Finally, let P'' be the prefix of P' from u to the first visit of $v(P)$ on P' (from the side of u). Observe that the predecessor of $v(P)$ on P'' belongs to G_u and is a neighbour of $v(P)$ in G, hence $v(P)$ has to be a strict ancestor of u in S. We find that P'' is a walk of length at most $2r$ in G, it starts in u, ends in $v(P)$, and all its internal vertices belong to G_u, so in particular they are not smaller than u in L. This means that P'' certifies that $v(P) \in \mathrm{SReach}_{2r}[G, L, u]$.

Since $|\mathrm{SReach}_{2r}[G, L, u]| \leq \mathrm{col}_{2r}(G, L)$, in order to prove the bound on $\mathrm{adm}_r(H, L)$, it suffices to prove the following claim: For each vertex v that is a strict ancestor of u in S, there can be at most three paths $P \in \mathcal{P}$ for which $v = v(P)$. To this end, we fix a vertex v that is a strict ancestor of u in S and proceed by a case distinction on how a path P with $v = v(P)$ may behave.

Suppose first that v is the endpoint of P other than u, equivalently the endpoint of P' other than u. (For example, $u = 1$, $P = 1, 11, 21, 0$, $P' = 1, 11, 1, 21, 0$ and $v = 0$, in Figures 1 and 2.) However, the paths of $\mathcal{P}$ are pairwise vertex-disjoint, apart from the starting vertex u, hence there can be at most one path P from $\mathcal{P}$ for which v is an endpoint. Thus, this case contributes at most one path P for which $v = v(P)$.

Next suppose that v is an internal vertex of the walk P'; in particular, it is not the endpoint of P other than u. (For example, $u = 6$, $P = 6, 11, 21, 0$, $P' = 6, 11, 1, 21, 0$ and $v = 1$, in Figures 1 and 2.) Since the only vertex traversed by P that is smaller than u in L is this other endpoint of P, and v is smaller than u in L due to being its strict ancestor in S, it follows that each visit of v on P' is due to having $v = a(e)$ for some edge $e \in F_{\mathrm{new}}$ traversed on P. Select e to be such an edge corresponding to the first visit of v on P'. Let $e = xy$, where x lies closer to u on P than y. (That is, in our figures, $x = 11$ and $y = 21$.) Since v was chosen as the first vertex on P' that does not belong to G_u, we have $x \in G_u$.

Since $v = a(e) = a(xy)$, either $x \in Z_v^{\downarrow}$ or $x \in Z_v^{\uparrow}$. Note that the second possibility cannot happen, because then v would be a descendant of x in S, hence v would belong to G_u, due to $x \in G_u$; a contradiction. This means $x \in Z_v^{\downarrow}$.

Recall that, by construction, $Z_v^{\downarrow}$ contains at most one vertex from each subtree of S rooted at a child of v. Since v is a strict ancestor of u in S, we infer that x has to be the unique vertex of $Z_v^{\downarrow}$ that belongs to G_u. In the construction of F_v, however, we added only at most two edges of F_v incident to this unique vertex: at most one to its predecessor on the enumeration of the children of v, and at most one to its successor. Since paths from $\mathcal{P}$ are pairwise vertex-disjoint in H, apart from the starting vertex u, only at most two paths from $\mathcal{P}$ can use any of these two edges (actually, only at most one unless $x = u$). Only for these two paths we can have $v = a(e)$. Thus, this case contributes at most two paths P for which $v = v(P)$, completing the proof of the claim. ⌟

We conclude the proof by summarising the algorithm: first construct the tree U, and then construct the tree T. As argued, these steps take time $\mathcal{O}(m \cdot \alpha(m))$ and $\mathcal{O}(n)$, respectively. By Claims 4 and 5, T satisfies the required properties. □

5 Conclusion

In this paper we show that model-checking for successor-invariant first-order formulas is fixed-parameter tractable on any class of structures of bounded expansion. This significantly reduces the existing gap for sparse classes between the known tractability results for plain and for successor-invariant first-order logic.

The obvious open question is whether this gap can be closed completely on sparse classes of graphs, i.e. whether successor-invariant FO is fixed-parameter tractable on any nowhere dense class of structures. As nowhere dense classes can also be characterised by colouring numbers, it is conceivable that our techniques can be extended. However, for nowhere dense classes of graphs, the colouring numbers are no longer bounded by a constant (for any fixed value of r) but only by n^{ε}, where n is the number of vertices. This poses several technical problems meaning that our techniques do not readily extend. We leave this as an open problem for future research.

Another open problem is the exact time complexity of our model-checking algorithm on classes of bounded expansion. For plain FO it is known that MC(FO, $\mathcal{C}$) is parametrised linear time for any class $\mathcal{C}$ of bounded expansion.

The simple analysis of our algorithm provided in this paper yields a parametrised running time of $\mathcal{O}(n \cdot \alpha(n))$. The only step that requires more than linear time is the construction of a specific spanning tree in Theorem 6. At the moment we do not see how to avoid this non-linear step, and we leave it for future research whether the tree T in the theorem (or a similar suitable tree) can be constructed more efficiently.

Finally, it would be very interesting to extend our results to order-invariant FO. One approach would be to show that model-checking for FO augmented by a reachability operator that can define reachability along the (definable) successor-relation constructed above is tractable on bounded expansion classes. However, as explained in the introduction, first-order logic with even very restricted forms of reachability very quickly becomes intractable even on planar graphs. Again, this is an area where further research seems appropriate.

References

[1] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symbolic Logic*, 74:168–186, 2009.

[2] Hans L. Bodlaender, Jitender S. Deogun, Klaus Jansen, Ton Kloks, Dieter Kratsch, Haiko Müller, and Zsolt Tuza. Rankings of graphs. *SIAM J. Discrete Math.*, 11:168–181, 1998.

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

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

[5] Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *LICS 2007*, pages 270–279. IEEE Computer Soc., 2007.

[6] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 4th edition, 2010.

[7] Zdeněk Dvořák. Constant-factor approximation of the domination number in sparse graphs. *European J. Combin.*, 34:833–840, 2013.

[8] Zdeněk Dvořák, Daniel Král', and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *J. ACM*, 60:36, 2013.

[9] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *MFCS 2014*, volume 8634 of *Lecture Notes in Comput. Sci.*, pages 256–266. Springer, 2014.

[10] Kord Eickmeyer and Ken-ichi Kawarabayashi. Successor-invariant first-order logic on graphs with excluded topological subgraphs. In *CSL 2016*, volume 62 of *LIPIcs*, pages 18:1–18:15. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2016.

[11] Kord Eickmeyer, Ken-ichi Kawarabayashi, and Stephan Kreutzer. Model checking for successor-invariant first-order logic on minor-closed graph classes. In *LICS 2013*, pages 134–142. IEEE Computer Soc., 2013.

[12] Michael Elberfeld, Marlin Frickenschmidt, and Martin Grohe. Order invariance on decomposable structures. In *LICS 2016*, pages 397–406. ACM, 2016.

[13] Viktor Engelmann, Stephan Kreutzer, and Sebastian Siebertz. First-order and monadic second-order model-checking on ordered structures. In *LICS 2012*, pages 275–284. IEEE Computer Soc., 2012.

[14] Jörg Flum, Markus Frick, and Martin Grohe. Query evaluation via tree-decompositions. *J. ACM*, 49:716–752, 2002.

[15] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

[16] Marcus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48:1184–1206, 2001.

[17] Jakub Gajarský, Petr Hliněný, Daniel Lokshtanov, Jan Obdržálek, Sebastian Ordyniak, M. S. Ramanujan, and Saket Saurabh. FO model checking on posets of bounded width. In *FOCS 2015*, pages 963–974. IEEE Computer Soc., 2015.

[18] Jakub Gajarský, Petr Hliněný, Jan Obdržálek, Daniel Lokshtanov, and M. S. Ramanujan. A new perspective on FO model checking of dense graph classes. In *LICS 2016*, pages 176–184. ACM, 2016.

[19] Robert Ganian, Petr Hliněný, Daniel Král', Jan Obdržálek, Jarett Schwartz, and Jakub Teska. FO model checking of interval graphs. *Log. Methods Comput. Sci.*, 11:4:11, 2015.

[20] Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics*, volume 558 of *Contemp. Math.*, pages 181–205. AMS, 2011.

[21] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. In *STOC 2014*, pages 89–98. ACM, 2014.

[22] Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM J. Comput.*, 44:114–159, 2015.

[23] Yuri Gurevich. Logic and the challenge of computer science. In *Trends in Theoretical Computer Science*, volume 12 of *Principles Comput. Sci. Ser.*, pages 1–57. Computer Sci. Press, 1988.

[24] H.A. Kierstead and Daqing Yang. Orderings on graphs and game coloring number. *Order*, 20:255–264, 2003.

[25] Stephan Kreutzer, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. The generalised colouring numbers on classes of bounded expansion. In *MFCS 2016*, volume 58 of *LIPIcs*, pages 85:1–85:13. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2016.

[26] Stephan Kreutzer and Siamak Tazari. Lower bounds for the complexity of monadic second-order logic. In *LICS 2010*, pages 189–198. IEEE Computer Soc., 2010.

[27] Stephan Kreutzer and Siamak Tazari. On brambles, grid-like minors, and parameterized intractability of monadic second-order logic. In *SODA 2010*, pages 354–364. SIAM, 2010.

[28] Hannu Niemistö. On locality and uniform reduction. In *LICS 2005*, pages 41–50. IEEE Computer Soc., 2005.

[29] Martin Otto. Epsilon-logic is more expressive than first-order logic over finite structures. *J. Symbolic Logic*, 65:1749–1757, 2000.

[30] Andreas Potthoff. *Logische Klassifizierung Regulärer Baumsprachen.* PhD thesis, Universität Kiel, 1994. In German.

[31] Neil Roberton and P.D. Seymour. Graph minors XVII. Taming a vortex. *J. Combin. Theory Ser. B*, 77:162–210, 1999.

[32] Benjamin Rossman. Successor-invariance in the finite. In *LICS 2003*, pages 148–157. IEEE Computer Soc., 2003.

[33] Benjamin Rossman. Successor-invariant first-order logic on finite structures. *J. Symbolic Logic*, 72:601–618, 2007.

[34] Alejandro A. Schäffer. Optimal node ranking of trees in linear time. *Inform. Process. Lett.*, 33:91–96, 1989.

[35] Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Structures Comput. Sci.*, 6:505–526, 1996.

[36] Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22:215–225, 1975.

[37] Moshe Y. Vardi. The complexity of relational query languages. In *STOC 1982*, pages 137–146. ACM, 1982.

[38] Xuding Zhu. Colouring graphs with bounded generalized colouring number. *Discrete Math.*, 309:5562–5568, 2009.