# Model-Checking on Ordered Structures

KORD EICKMEYER, Technische Universität Darmstadt, Germany
JAN VAN DEN HEUVEL, London School of Economics and Political Science, United Kingdom
KEN-ICHI KAWARABAYASHI, National Institute of Informatics, Japan
STEPHAN KREUTZER, Technische Universität Berlin, Germany
PATRICE OSSONA DE MENDEZ, Centre d'Analyse et de Mathématiques Sociales (CNRS, UMR 8557), France and Charles University, Czech Republic
MICHAŁ PILIPCZUK, University of Warsaw, Poland
DANIEL A. QUIROZ, Universidad de Chile, Chile
ROMAN RABINOVICH, Technische Universität Berlin, Germany
SEBASTIAN SIEBERTZ, Universität Bremen, Germany

We study the model-checking problem for first- and monadic second-order logic on finite relational structures. The problem of verifying whether a formula of these logics is true on a given structure is considered intractable in general, but it does become tractable on interesting classes of structures, such as on classes whose Gaifman graphs have bounded treewidth. In this article, we continue this line of research and study model-checking for first- and monadic second-order logic in the presence of an ordering on the input structure. We do so in two settings: the general ordered case, where the input structures are equipped with a fixed

11

This article subsumes the results of References [17, 18, 20, 33, 38].

S. Kreutzer and R. Rabinovich were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Consolidator Grant DISTRUCT, grant agreement No. 648527). P. Ossona de Mendez was supported by grants ERCCZ LL-1201 and CE-ITI and by the European Associated Laboratory "Structures in Combinatorics" (LEA STRUCO) P202/12/G061. M. Pilipczuk and S. Siebertz were supported by the National Science Centre of Poland via POLONEZ grant agreement UMO-2015/19/P/ST6/03998, which has received funding from the European Union's Horizon 2020 research and innovation programme (Marie Skłodowska-Curie grant agreement No. 665778). D.A. Quiroz was supported by CONICYT, PIA/Concurso Apoyo a Centros Científicos y Tecnológicos de Excelencia con Financiamiento Basal AFB170001.

Authors' addresses: K. Eickmeyer, Department of Mathematics, Technische Universität Darmstadt, Schlossgartenstraße 7, Darmstadt, 64289, Germany; email: eickmeyer@mathematik.tu-darmstadt.de; J. van den Heuvel, Department of Mathematics, London School of Economics and Political Science, Houghton Street, London, WC2A 2AE, United Kingdom; email: j.van-denheuvel@lse.ac.uk; K.-I. Kawarabayashi, National Institute of Informatics, Hitotsubashi 2-1-2, Tokyo, 101-8430, Japan; email: k_keniti@nii.ac.jp; S. Kreutzer, Logic and Semantics, Technische Universität Berlin, Ernst-Reuter-Platz 7, Berlin, 10587, Germany; email: stephan.kreutzer@tu-berlin.de; P. O. de Mendez, Centre d'Analyse et de Mathématiques Sociales (CNRS, UMR 8557), 54, boulevard Raspail, Paris, 75006, France and Computer Science Institute (IUUK), Charles University, Malostranske nam. 25, Prague, 118 00, Czech Republic; email: pom@ehess.fr; M. Pilipczuk, Institute of Informatics, University of Warsaw, Stefana Banacha 2, Warsaw, 02-097, Poland; email: michal.pilipczuk@mimuw.edu.pl; D. A. Quiroz, Departamento de Ingeniería Matemática and Centro de Modelamiento Matemática, Universidad de Chile, Av. Beauchef 851, Santiago, Chile; email: dquiroz@cmm.uchile.cl; R. Rabinovich, Logic and Semantics, Technische Universität Berlin, Ernst-Reuter-Platz 7, Berlin, 10587, Germany; email: roman.rabinovich@tu-berlin.de; S. Siebertz, Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, Berlin, 10099, Germany; email: siebertz@informatik.hu-berlin.de.



1529-3785/2020/03-ART11 $15.00
https://doi.org/10.1145/3360011

order or successor relation, and the order-invariant case, where the formulas may resort to an ordering, but their truth must be independent of the particular choice of order. In the first setting we show very strong intractability results for most interesting classes of structures. In contrast, in the order-invariant case we obtain tractability results for order-invariant monadic second-order formulas on the same classes of graphs as in the unordered case. For first-order logic, we obtain tractability of successor-invariant formulas on classes whose Gaifman graphs have bounded expansion. Furthermore, we show that model-checking for order-invariant first-order formulas is tractable on coloured posets of bounded width.

CCS Concepts: • **Theory of computation → Finite Model Theory**; **Fixed parameter tractability**; *Complexity theory and logic*; • **Mathematics of computing** → *Graph algorithms*; Graphs and surfaces;

Additional Key Words and Phrases: Model checking, order-invariance, successor-invariance, algorithmic meta-theorems

**ACM Reference format:**
Kord Eickmeyer, Jan van den Heuvel, Ken-ichi Kawarabayashi, Stephan Kreutzer, Patrice Ossona de Mendez, Michał Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. 2020. Model-Checking on Ordered Structures. *ACM Trans. Comput. Logic* 21, 2, Article 11 (March 2020), 28 pages.
https://doi.org/10.1145/3360011

## 1 INTRODUCTION

Pinpointing the exact complexity of the model-checking problem for first-order and monadic second-order logic has been the object of a large body of research. The model-checking problem for a logic $\mathcal{L}$, denoted $\mathrm{MC}(\mathcal{L})$, is the problem of deciding for a given finite structure $\mathbb{A}$ and a formula $\varphi \in \mathcal{L}$ where $\mathbb{A}$ is a model of $\varphi$; in symbols $\mathbb{A} \models \varphi$. We will denote $\mathrm{MC}(\mathcal{L})$ restricted to a class $\mathscr{C}$ of input structures as $\mathrm{MC}(\mathcal{L}, \mathscr{C})$.

Vardi [59] proposed to distinguish the complexity of the model-checking problem into *data*, *formula*, and *combined* complexity, depending on whether we treat the structure $\mathbb{A}$ (the data) as input while considering $\varphi$ as fixed, the formula as input while considering $\mathbb{A}$ as fixed, or considering both $\mathbb{A}$ and $\varphi$ as part of the input. As shown by Vardi, for any fixed formula $\varphi \in \mathrm{FO}$ of size $|\varphi|$ the model-checking problem is solvable in polynomial time $n^{O(|\varphi|)}$, i.e., the data complexity of MC(FO) is in Ptime. However, the formula complexity and combined complexity of first-order logic is Pspace-complete already on a fixed 2-element structure [5]. Evaluating a fixed formula of monadic second-order logic belongs to the polynomial time hierarchy, and for each level $\Sigma_i^p$ and $\Pi_i^p$ there exists an MSO-formula whose model-checking problem is complete for that level [57]. Again, the formula complexity and combined complexity of monadic second-order logic is PSpace-complete.

A more fine-grained analysis of model-checking complexity can be achieved through the lens of parameterised 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 $f(|\varphi|) \cdot |\mathbb{A}|^c$, for some function $f$ (usually required to be computable) and a constant $c$ independent of $\varphi$ and $\mathbb{A}$. The complexity class FPT of all fixed-parameter tractable problems is the parameterised analogue to Ptime as a model of efficient solvability. Hence, parameterised 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. Already the model-checking problem for first-order logic is complete for the parameterised complexity class AW[∗], which is conjectured to strictly contain the class FPT. Thus it is widely believed that model-checking for first-order logic in general (and thus also for monadic second-order logic) is not fixed-parameter tractable.

Perhaps the most famous result on the parameterised complexity of model-checking is Courcelle's theorem [7], which states that every algorithmic property on graphs definable in monadic second-order logic (with quantification over edge sets) can be evaluated in linear time on any class

of graphs of bounded treewidth. An equivalent statement is that MC(MSO, $\mathscr{C}$) is fixed-parameter tractable via a linear-time algorithm for any class $\mathscr{C}$ of bounded treewidth. This result was followed by a similar result for monadic second-order logic with only quantification over vertex sets on graph classes of bounded clique-width [9]. It was shown in References [39, 40] that Courcelle's theorem cannot be extended in full generality much beyond bounded treewidth.

For first-order logic, Seese [55] proved that first-order model-checking is fixed-parameter tractable on any class of graphs of bounded 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., References [11, 15, 22, 24, 31], and see Reference [29] for a survey. This line of research culminated in the theorem of Grohe et al. [31], stating that for any nowhere dense class $\mathscr{C}$ of graphs we have MC(FO, $\mathscr{C}$) $\in$ FPT. Moreover, for classes of graphs that are closed under taking subgraphs, this yields a precise characterisation of tractability for first-order model-checking [15].

So far, most of the work on algorithmic meta-theorems has focused on unordered structures. Many of the results mentioned above rely on locality theorems for first-order logic, such as Gaifman's locality theorem [25], and the applied techniques do not readily extend to ordered structures. In this article, we study the complexity of first-order model-checking on structures where an ordering is available to be used in formulas. We do so in two different settings. The first is that the input structures are equipped with a fixed order or with a fixed successor relation. (A successor relation is a directed Hamiltonian path on the universe of the structure.) We show that first-order logic on ordered structures as well as on structures with a successor relation is essentially intractable on nearly all interesting classes.

The other cases we consider are order-*invariant* or successor-*invariant* formulas. In an order-invariant formula, we are allowed to use an order relation, but whether the formula is true in a given structure must not depend on the particular choice of order. In the following, we will speak about successor- and order-invariant formulas, and not about successor- and order-invariant logics, as usually it is required that a logic has a decidable syntax. However, it is undecidable whether a first-order sentence is successor or order invariant, see, e.g., Reference [41]. In particular, in case we deal with the model-checking problem for successor- or order-invariant formulas we assume that the input formula is indeed successor or order invariant, and we do not have to verify this property.

It is easily seen that the expressive power of order-invariant MSO formulas is greater than that of plain MSO formulas, as, e.g., with an order we can formalise in MSO that a structure has an even number of elements, a property not definable without an order. In fact, the expressive power of order-invariant MSO formulas is even greater than the expressive power of the extension of MSO formulas with counting quantifiers CMSO [28]. Over-restricted classes of structures, order-invariant MSO formulas, and CMSO formulas have the same expressive power (see, e.g., Reference [8]). This holds true for successor-invariant MSO formulas as well, as an order is definable from a successor relation via MSO.

We are able to show that the model-checking problem for order-invariant MSO formulas is tractable on essentially the same classes of graphs as for plain MSO formulas, i.e., we can increase the expressive power without restricting the tractable cases. To be precise, we show that the model-checking problem for order-invariant MSO formulas on classes of graphs of bounded clique-width is fixed-parameter tractable. Furthermore, combining the result of Courcelle [7] and results that one can add the edges of a successor relation to a graph of bounded treewidth without increasing its treewidth too much, we get that model-checking for order-invariant MSO formulas (with quantification over edge sets) on classes of graphs of bounded treewidth is fixed-parameter tractable.

Also successor- and order-invariant first-order formulas have both been studied intensively in the literature, see, e.g., References [1, 19, 47, 50, 52, 53]. However, the difference between the

expressive powers of order-invariant, successor-invariant, and plain FO formulas on various classes of structures remains largely unexplored. An unpublished result of Gurevich states that the expressive power of order-invariant FO formulas is stronger than that of plain FO formulas (see, e.g., Theorem 5.3 of Reference [41] for a presentation of the result). Rossman [53] proved the stronger result that successor-invariant FO formulas are more expressive than plain FO formulas. The construction of Reference [53] creates dense instances though, and no separation between successor-invariant FO formulas and plain FO formulas is known on sparse classes, say, on classes of bounded expansion. However, 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, 46] and on graphs of bounded treedepth [16]. 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 Reference [19].

We show that, up to a narrow gap, the model-checking results for plain FO formulas carry over to successor-invariant FO formulas. In particular, we show that model-checking successor-invariant FO formulas 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. More precisely, we show that if $\mathscr{C}$ is a class of structures of bounded expansion, then model-checking for successor-invariant first-order formulas on $\mathscr{C}$ 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 FO can be done in linear time on classes of bounded expansion [15], thus the running time of our algorithm is very close to the best known results for plain FO.

The natural way of proving tractability for successor-invariant FO formulas on a specific class $\mathscr{C}$ of graphs is to show that any given graph $G \in \mathscr{C}$ can be augmented by a new set $S$ of coloured edges that form a successor relation on $V(G)$ such that $G + S$ falls within a class $\mathscr{D}$ of graphs on which plain FO is tractable. In this way, model-checking for successor-invariant FO on the class $\mathscr{C}$ is reduced to the model-checking problem for FO on $\mathscr{D}$. The main problem is how to construct the set of augmentation edges $S$. For classes of bounded expansion, to construct such an edge set, 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 article is that we find a way to control these elimination trees so that we can use them to define, in a first step, 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 increase in the colouring numbers is bounded. In a second step, from the bounded degree spanning tree we will construct a successor relation $S$ as desired.

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

As mentioned before, the tractability of model-checking for FO on sparse graphs is well understood, while only few results are available for classes of dense graphs. We review some known results for FO model-checking on dense graph classes in Section 6 and show that a result by Gajarský et al. [26] carries over to order-invariant FO formulas.

**Organisation of the article.** In Section 2, we fix the terminology and notation used throughout the article. In Section 3, we study the case of ordered structures, i.e., structures equipped with a fixed order or successor relation. Order-invariant MSO formulas are considered in Section 4. We recall the notions from the theory of sparse graphs, in particular the generalised colouring numbers,

and prove tractability of the model-checking problem for successor-invariant FO formulas on bounded expansion classes in Section 5. Finally, in Section 6 we consider order-invariant FO formulas on posets of bounded width and other dense classes of structures.

## 2 PRELIMINARIES

**General 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, i.e., the inverse of the function $n \mapsto A(n,n)$ with

$$A(m,n) := \begin{cases} n+1, & \text{if } m = 0; \\ A(m-1,1), & \text{if } m > 0 \text{ and } n = 0; \\ A(m-1, A(m,n-1)), & \text{if } m,n > 0. \end{cases}$$

**Relational structures.** We consider finite structures over finite signatures that contain only relation symbols and constant symbols. Hence a signature $\tau = \{R_1, \ldots, R_k, c_1, \ldots, c_s\}$ is a finite set of relation symbols $R_i$ and constant symbols $c_i$, where each relation symbol $R \in \tau$ is assigned an *arity* $\mathrm{ar}(R)$ (arities are part of the signature). A $\tau$-structure $\mathbb{A} = (V(\mathbb{A}), R_1(\mathbb{A}), \ldots, R_k(\mathbb{A}), c_1(\mathbb{A}), \ldots, c_s(\mathbb{A}))$ consists of a set $V(\mathbb{A})$, the *universe* of $\mathbb{A}$, for each $R_i \in \tau$ a relation $R_i(\mathbb{A}) \subseteq V(\mathbb{A})^{\mathrm{ar}(R_i)}$, and for each $c_i \in \tau$ a constant $c_i(\mathbb{A}) \in V(\mathbb{A})$. If $\mathbb{A}$ is a $\tau$-structure and $R$ is a relation symbol not in $\tau$ with associated arity $r$ and $R(\mathbb{A}) \subseteq V(\mathbb{A})^r$ is an $r$-ary relation over $V(\mathbb{A})$, then we write $(\mathbb{A}, R(\mathbb{A}))$ for the $\tau \cup \{R\}$-structure obtained by extending $\mathbb{A}$ with the relation $R(\mathbb{A})$. The *order* $|\mathbb{A}|$ of a $\tau$-structure $\mathbb{A}$ is $|V(\mathbb{A})|$, and its *size* $\|\mathbb{A}\|$ is $|\tau| + |V(\mathbb{A})| + \sum_{R\in\tau} |R(\mathbb{A})|$, which corresponds to the size of a representation of $\mathbb{A}$ in an appropriate model of computation. We call a structure $G$ of signature $\{E\}$, where $E$ is a binary relation symbol, a *digraph*, and if $E(G)$ is symmetric and irreflexive, then we call $G$ a *graph*. We denote an undirected edge between vertices $u$ and $v$ by $uv$ and a directed edge by $(u,v)$. A directed path is sometimes denoted by a sequence $v_1, \ldots, v_n$ of vertices such that $(v_i, v_{i+1}) \in E(G)$ for all $1 \leqslant i < n$ and sometimes by a binary relation containing the pairs $(v_i, v_{i+1})$ for $1 \leqslant i < n$. 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 $|V|-1$. We write $\bar{a}$ for a finite sequence $(a_1, \ldots, a_k)$ and usually leave it to the context to determine the length of a sequence. The *Gaifman-graph* $G(\mathbb{A})$ of a $\tau$-structure $\mathbb{A}$ is the graph with vertex set $V(\mathbb{A})$ and edge set $\{(u,v) : u \neq v$, and there is an $R \in \tau$ and a tuple $\bar{a} \in R(\mathbb{A})$ such that $u, v \in \bar{a}\}$.

**First-order logic.** We assume familiarity with first-order logic FO and monadic second-order logic MSO. We write $\mathrm{FO}(\tau)$ and $\mathrm{MSO}(\tau)$ for the set of all FO and MSO formulas over signature $\tau$, respectively. If $\varphi$ is a formula of FO or MSO, then we write $|\varphi|$ for the length (of an encoding) of $\varphi$. If $\varphi$ is a sentence of $\mathrm{FO}(\tau)$ or $\mathrm{MSO}(\tau)$ and $\mathbb{A}$ a $\tau$-structure, then we write $\mathbb{A} \models \varphi$ if $\varphi$ is true in $\mathbb{A}$. If $\varphi(\bar{x})$ has free variables $\bar{x}$ and $\bar{a} \in V(\mathbb{A})^k$ is a tuple of the same length as $\bar{x}$, then we write $\mathbb{A} \models \varphi(\bar{a})$ if $\varphi$ is true in $\mathbb{A}$, where the free variables $\bar{x}$ are interpreted by the elements of $\bar{a}$ in the obvious way. We write $\varphi(\mathbb{A})$ for the relation $R := \{\bar{a} : \mathbb{A} \models \varphi(\bar{a})\}$. We call a formula $\varphi(\bar{x})$ over signature $\tau = \sigma \cup \{<\}$*order invariant* if for every $\sigma$-structure $\mathbb{A}$ and all linear orders $<_1, <_2$ over $V(\mathbb{A})$ we have $\varphi(\mathbb{A}, <_1) = \varphi(\mathbb{A}, <_2)$. Analogously, we call a formula $\varphi(\bar{x})$ over signature $\tau = \sigma \cup \{S\}$ (where $S$ is a binary relation symbol) *successor invariant* if for every $\sigma$-structure $\mathbb{A}$ and all successor relations $S_1, S_2$ over $V(\mathbb{A})$ we have $\varphi(\mathbb{A}, S_1) = \varphi(\mathbb{A}, S_2)$. We write FO[<-inv] and MSO[<-inv] for the set of all order-invariant FO and MSO formulas, respectively, and FO[+1-inv] and MSO[+1-inv] for the set of all successor-invariant FO and MSO formulas, respectively. We write FO[<] and MSO[<] for the set of all FO and MSO formulas, respectively, over a signature that contains at least the binary relation symbol <, and similarly for FO[+1] and MSO[+1]. For any order-invariant formula $\varphi$ and any $\tau$-structure $\mathbb{A}$, we write $\mathbb{A} \models_{\text{ord-inv}} \varphi$ if for some (equivalently, every) order

relation $<$ on the universe of $\mathbb{A}$ we have $(\mathbb{A}, <) \models \varphi$. Similarly, for a successor-invariant formula $\varphi$ we write $\mathbb{A} \models_{\text{succ-inv}} \varphi$ if for some (equivalently, every) successor relation $S$ on the universe of $\mathbb{A}$ we have $(\mathbb{A}, S) \models \varphi$.

Throughout the article, we study the complexity of order- and successor-invariant logics on restricted classes of structures. As usual in this type of research we focus on classes of graphs. More general structures can be reduced to this case using their Gaifman-graphs. In our analysis we will use the framework of *parameterised complexity*, see, e.g., References [10, 13, 23]. A *parameterised problem* is a subset of $\Sigma^* \times \mathbb{N}$, where $\Sigma$ is a fixed finite alphabet. For an instance $(w, k) \in \Sigma^* \times \mathbb{N}$, we call $k$ the parameter.

Let $\mathscr{C}$ be a class of graphs and $\mathcal{L}$ be one of first-order or monadic second-order logic. The *model-checking problem* $\mathrm{MC}(\mathcal{L}[<\text{-inv}], \mathscr{C})$ for order-invariant sentences of $\mathcal{L}$ on the class $\mathscr{C}$ of graphs is defined as the problem

| $\mathrm{MC}(\mathcal{L}[<\text{-inv}], \mathscr{C})$ | |
|---|---|
| *Input:* | $G \in \mathscr{C}$, order-invariant $\varphi \in \mathcal{L}(\{E, <\})$ |
| *Parameter:* | $\lvert\varphi\rvert$ |
| *Problem:* | $G \models_{\text{ord-inv}} \varphi$ ? |

As mentioned before, we assume that the given formula is order invariant and the algorithm does not have to verify this property. Analogously, we define the model-checking problem $\mathrm{MC}(\mathcal{L}[+1\text{-inv}], \mathscr{C})$ successor-invariant sentences of $\mathcal{L}$ on $\mathscr{C}$, where instead the formula $\varphi \in \mathcal{L}(\{E, S\})$ is required to be successor invariant. Finally, we define the *ordered model-checking problem* $\mathrm{MC}(\mathcal{L}[<], \mathscr{C})$ for $\mathcal{L}$ on $\mathscr{C}$ as

| $\mathrm{MC}(\mathcal{L}[<], \mathscr{C})$ | |
|---|---|
| *Input:* | $G \in \mathscr{C}$, $<$ a linear order of $V(G)$ and $\varphi \in \mathcal{L}(\{E, <\})$ |
| *Parameter:* | $\lvert\varphi\rvert$ |
| *Problem:* | $(G, <) \models \varphi$? |

Likewise, we define the model-checking problem $\mathrm{MC}(\mathcal{L}[+1], \mathscr{C})$ with successor for $\mathcal{L}$ on $\mathscr{C}$, which gets as input a graph $G \in \mathscr{C}$, a successor relation $S$ on $V(G)$, and a formula $\varphi \in \mathcal{L}(\{E, S\})$.

The order- and successor-invariant model-checking problems are *fixed-parameter tractable*, or in the complexity class FPT, if there are algorithms that correctly decide on input $(G, \varphi)$ whether $(G, <) \models \varphi$ for some linear order $<$, or $(G, S) \models \varphi$ for some successor relation $S$, respectively, in time $f(|\varphi|) \cdot \|G\|^{O(1)}$, for some function $f : \mathbb{N} \to \mathbb{N}$ in the case where $\varphi$ is order invariant, or successor invariant, respectively. We use an analogous definition of FTP for $\mathrm{MC}(\mathcal{L}[<])$ and $\mathrm{MC}(\mathcal{L}[+1])$. The model-checking problem for first-order logic is complete for the parameterised complexity class AW[$*$], which is conjectured to strictly contain the class FPT. Thus, it is widely believed that model-checking for first-order logic (and thus also for monadic second-order logic) is not fixed-parameter tractable.

## 3 MODEL-CHECKING ON ORDERED STRUCTURES

In this section, we investigate the tractability of model-checking on classes of ordered structures and on classes of structures with a successor relation. Of course, it is easy to come up with classes of ordered structures on which model-checking is fixed-parameter tractable, e.g., by taking the class of all cliques with a linear order on the vertex set. Thus we seek restrictions as weak as possible while still allowing us to show that model-checking is AW[$*$]-hard.

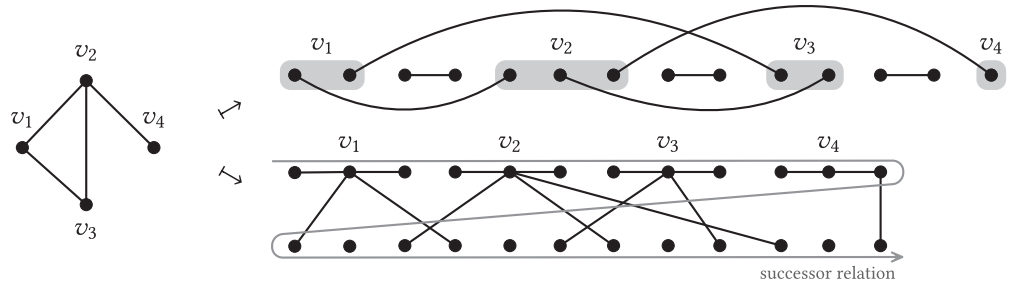


Fig. 1. A sample graph (left) encoded in a linear order plus partial matching (upper right) and a star forest plus successor relation (lower right).

### 3.1 Coloured Sets

We start by observing that on the class of ordered coloured sets (and, *a forteriori*, on the class of coloured sets with a successor relation), model-checking is tractable even for monadic second-order logic. This is Büchi-Elgot-Trakhtenbrot's Theorem (cf. Reference [23]), since coloured ordered sets are just strings. Thus model-checking for MSO is fixed-parameter tractable on structures whose signature contains only unary relation symbols, apart from the order relation.

### 3.2 Vertex-Ordered Graphs

The simplest case not covered by the preceding paragraph is that of ordered graphs, i.e., $\{E, <\}$-structures where both $E$ and $<$ are binary relation symbols. We show that model-checking for first-order logic is AW[$*$]-hard even for very simple graphs.

Theorem 3.1. *Let $\mathscr{C}$ be a class of graphs. If $\mathscr{C}$ contains all partial matchings, then* MC(FO[$<$], $\mathscr{C}$) *is AW[$*$]-hard. If $\mathscr{C}$ contains all star forests, then* MC(FO[+1], $\mathscr{C}$) *is AW[$*$]-hard.*

Here, a *partial matching* is a disjoint union of edges and isolated vertices (a graph of maximum degree 1), while a *star forest* is a disjoint union of stars (complete bipartite graphs $K_{1,n}$ with $n \geqslant 0$). Note that on both these graph classes, the model-checking problem for plain FO is fixed-parameter tractable.

Proof. For the first part we show how to construct in polynomial time for every graph $G$ an $\{E, <\}$-structure $\mathbb{A}$ such that $G$ can be FO-interpreted in $\mathbb{A}$. For this, let $\{v_1, \ldots, v_n\}$ be the vertex set of $G$ ordered in an arbitrary way, and assume that $v_i$ has degree $d_i$ in $G$. To each vertex in $G$ we associate an interval of length $\hat{d}_i := \max\{d_i, 1\}$ in $\mathbb{A}$ and separate the intervals by gaps of length 2. Thus with

$$D_k := 2k - 1 + \sum_{i=1}^{k-1} \hat{d}_i,$$

we associate with $v_i$ the interval $\{D_i, \ldots, D_i + \hat{d}_i - 1\}$. The edge set $E(\mathbb{A})$ consists of the edges $\{D_i - 2, D_i - 1\}$ for $i \geqslant 2$, together with the edges $\{D_i + k, D_j + \ell\}$ if $v_i v_j$ is an edge of $G$, $v_j$ is the $k$th neighbour of $v_i$ in the ordering, and $v_i$ is the $\ell$-th neighbour of $v_i$. The resulting construction is sketched in Figure 1. Notice that the edges $\{D_i - 2, D_i - 1\}$ are the only edges between consecutive elements, so they can be used to determine the intervals used in this construction. We leave the slightly technical but simple details of the FO-interpration to the reader.

For the second part we construct a structure $\mathbb{A}'$ consisting of a disjoint union of stars and a successor relation that can be used to recover the original graph using an FO interpretation. Again, we assume the vertex set of $G$ to be $\{v_1, \ldots, v_n\}$. A vertex $v$ is encoded by a path $v^{-1}, v, v^{+1}$. The

vertices of these paths are placed at the beginning of the successor relation in an arbitrary order. An edge $e$ is encoded by three vertices $e^{-1}, e, e^{+1}$ such that $e$ is a direct successor of $e^{-1}$ and $e^{+1}$ is a direct successor of $e$. All these vertices are placed at the end of the successor relation. For every edge $e = vw$, assume that $v$ is smaller than $w$ in the successor relation. We connect, in $\mathbb{A}'$, $v$ to $e^{-1}$ and $w$ to $e^{+1}$. Again, it is easy to see that $G$ may be recovered from $\mathbb{A}'$ using an FO interpretation. □

As a corollary of the previous theorem we get that MC(FO[+1], $\mathscr{P}$) and MC(FO[<], $\mathscr{T}$) are AW[∗]-hard for the class $\mathscr{P}$ of planar graphs and the class $\mathscr{T}$ of graphs of treewidth 1 (forests). However, for MC(FO[+1], $\mathscr{C}$) to be AW[∗]-hard it is essential that the graphs in the class $\mathscr{T}$ have unbounded degree. Indeed, on graph classes of bounded degree, successor-invariant FO model-checking is tractable.

Theorem 3.2. *For every $d \geqslant 0$ let $\mathscr{D}_d$ be the class of graphs of maximum degree at most $d$. Then for all $d \geqslant 0$,* MC(FO[+1], $\mathscr{D}_d$) *is fixed-parameter tractable. In fact, we can allow any (fixed) number of successor relations on top of $\mathscr{D}_d$ and still have tractable first-order model-checking.*

Proof. By a result of Seese [55], FO model-checking on graphs of bounded degree and also on all structures with Gaifman graph of bounded degree is fixed-parameter tractable. Adding a successor relation increases the degree of the Gaifman graph of a structure by at most two. □

## 4 ORDER-INVARIANT MSO

In this section, we consider order-invariant logics. The most expressive logic studied in the context of algorithmic meta-theorems is monadic second-order logic, the extension of first-order logic by quantification over sets of elements. With respect to graphs, there are two variants of MSO usually considered, one, called $\mathrm{MSO}_1$, where we can quantify over sets of vertices, and the other, called $\mathrm{MSO}_2$, where we can additionally quantify over sets of edges. It was shown by Courcelle [7] that $\mathrm{MSO}_2$ is fixed-parameter tractable on every class of graphs of bounded treewidth. Later, Courcelle et al. [9] showed that $\mathrm{MSO}_1$ is fixed-parameter tractable on every class of graphs of bounded clique-width, a concept more general than bounded treewidth. In this section, we show that for both logics we can allow order-invariance without increase in complexity.

We first consider the case of $\mathrm{MSO}_2$. As shown in Theorem 5.1.1. of Reference [51], for every graph $G$ of treewidth $k$ there is a successor relation $S$ on $V(G)$ such that the graph obtained from $G$ by adding the edges in $S$ has treewidth at most $k + 2$. From the proof one can easily derive an algorithm that takes as input a graph $G$ and a tree decomposition of $G$ of width $k$ and outputs a successor relation as desired in polynomial time. We also refer to the earlier results [6, 43] for proofs that for every graph $G$ of treewidth $k$ there is a successor relation $S$ on $V(G)$ such that the graph obtained from $G$ by adding the edges in $S$ has treewidth at most $k + 5$. We can use the algorithm of Bodlaender [2] to compute an optimal tree decomposition in time $2^{O(k^3)} \cdot n$ first, and then compute the desired successor relation. In combination with Courcelle's theorem, this implies the following.

Theorem 4.1. *For any class $\mathscr{C}$ of bounded treewidth,* MC($\mathrm{MSO}_2$[<-inv], $\mathscr{C}$) *is fixed-parameter tractable.*

In fact, MC($\mathrm{MSO}_2$[<-inv]) is fixed-parameter tractable with parameter $|\varphi| + \mathrm{tw}(G)$, where $\mathrm{tw}(G)$ is the treewidth of a graph $G$. We prove next that also for $\mathrm{MSO}_1$ and clique-width we can allow order-invariance without loss of tractability.

Theorem 4.2. *For any class $\mathscr{C}$ of graphs of bounded clique-width,* MC($\mathrm{MSO}_1$[<-inv], $\mathscr{C}$) *is fixed-parameter tractable.*

We first review the definition of clique-width. For the rest of this section, we fix a relational signature $\sigma$ in which every relation symbol has arity at most 2.

*Definition 4.3 ($\sigma$-clique-expression of width $k$).* Let $k \in \mathbb{N}$ be fixed. A *$\sigma$-clique-expression of width $k$* is a pair $(T, \lambda)$, where $T$ is a directed rooted tree in which all edges are directed away from the root and

$$\lambda : V(T) \to \{\mathbf{1}, \ldots, \mathbf{k}\} \cup \{\oplus\} \cup \{\text{edge}_{R,i\to j} : R \in \sigma,\ i,j = 1,\ldots,k\} \cup \{\text{rename}_{i\to j} : i,j = 1,\ldots,k\},$$

such that for all $t \in V(T)$ we have the following: If $\lambda(t) \in \{\mathbf{1}, \ldots, \mathbf{k}\}$, then $t$ is a leaf of $T$; if $\lambda(t) = \oplus$, then $t$ has exactly two successors; and in all other cases $t$ has exactly one successor.

*Definition 4.4.* Let $(T, \lambda)$ be a $\sigma$-clique-expression of width $k$. With every $t \in V(T)$ we associate a $\sigma$-structure $G(t)$ in which vertices are coloured by colours $\mathbf{1}, \ldots, \mathbf{k}$ as follows.

- If $t$ is a leaf, then $G(t)$ consists of one element coloured by $\lambda(t)$.
- If $\lambda(t) = \oplus$ and $t$ has successors $t_1, t_2$, then $G(t)$ is the disjoint union $G(t_1) \dot\cup G(t_2)$.
- If $\lambda(t) = \text{edge}_{R,i\to j}$ and $t_1$ is the successor of $t$, then $G(t)$ is the structure obtained from $G(t_1)$ by adding to the relation $R(G(t))$ all pairs $(u, v)$ such that $u$ has colour $i$ and $v$ has colour $j$.
- If $\lambda(t) = \text{rename}_{i\to j}$ and $t_1$ is the successor of $t$, then $G(t)$ is the structure obtained from $G(t_1)$ by changing the colour of all vertices $v$ that have colour $i$ in $G(t_1)$ to colour $j$ in $G(t)$.

The *$\sigma$-structure generated by* $(T, \lambda)$ is the structure $G(r)$, where $r$ is the root of $T$, from which we remove all colours $\{\mathbf{1}, \ldots, \mathbf{k}\}$. Finally, the *clique-width* of a $\sigma$-structure $G$ is the minimal width of a clique-expression generating $G$.

Combining results from Reference [34] and Reference [48] yields the following well-known result.

Fact 4.5. *There are a computable function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm that, given a graph $G$ of clique-width at most $k$ as input, computes a clique-expression of width at most $2^{k+1}$ in time $f(k) \cdot |G|^3$.*

Combining this theorem with results of Courcelle et al. [9] yields the following result.

Fact 4.6. *For any class $\mathscr{C}$ of graphs of bounded clique-width, $\mathrm{MC}(\mathrm{MSO}_1, \mathscr{C})$ is fixed-parameter tractable.*

In fact, the result applies to any $\sigma$-structure of bounded clique-width, provided that a clique-expression generating the structure (whose width is bounded by a computable function of the clique-width of the structure) is given or computable in polynomial time.

The next lemma is the main technical ingredient for the proof of Theorem 4.2 above.

Lemma 4.7. *There is an algorithm that, on input a graph $G$ of clique-width at most $k$, computes a linear order $<$ on $V(G)$ and a clique-expression of width at most $2^{k+2}$ generating the structure $(G, <)$.*

Proof. Let $G$ and $k$ be given. Using Theorem 4.5 we first compute an $\{E\}$-clique-expression $(T, \lambda)$ of width at most $2^{k+1}$ generating $G$. Let $r$ be the root of $T$. For every node $t \in V(T)$ we fix an ordering of its successors. Let $\prec$ be the partial order on $V(T)$ induced by this ordering.

Let $t \in V(T)$ be a node and let $s \neq t$ be the first node on the path $P$ from $t$ to $r$ with $\lambda(s) = \oplus$, if it exists. Let $s_1, s_2$ be the successors of $s$ with $s_1 \prec s_2$. We call $t$ a *left node* if $s_1 \in V(P)$, and a *right node* otherwise. If there is no node labelled $\oplus$ strictly above $t$, then we call $t$ a left node as well.

For every $t \in V(T)$ let $T_t$ be the subtree of $T$ with root $t$, and let $\lambda_{|T_t}$ be the restriction of $\lambda$ to the subtree $T_t$. We recursively define a transformation $\rho(T_t, \lambda_{|T_t})$ on the subtrees of $T$ defined as

follows. Intuitively, we produce a new clique-expression $(T', \lambda')$ over the signature $\{E, <\}$ using colours $\{(i, \mathit{left}), (i, \mathit{right}) : 1 \leqslant i \leqslant k\}$. Essentially, the new clique-expression will generate the same graph as $(T, \lambda)$, but so that if $t$ is a node in $T$ and $T_t$ generates the graph $G_t$, then $T'$ contains a node $t'$ generating an ordered version $G'_t := (G_t, <)$ of $G_t$ so that if $v \in V(G_t)$ has colour $i$, then, in $G'_t$, $v$ has colour $(i, \mathit{left})$ if $t$ is a left node and $(i, \mathit{right})$ if $t$ is a right node. Hence, whenever in $T$ we take the disjoint union of $G_s$ and $G_t$ and $s \prec t$, then we can define the ordering on $G'_s \dot\cup G'_t$ by adding all edges from nodes in $G'_s$ to $G'_t$, i.e., all edges from vertices coloured $(i, \mathit{left})$ to $(j, \mathit{right})$ for all pairs $i, j$. Formally, the transformation is defined as follows.

- If $t \in V(T)$ is a leaf, then $\rho(t) := (T', \lambda')$, where $T'$ consists only of $t$ and $\lambda'(t) := (\lambda(t), \mathit{left})$ if $t$ is a left node, and $\lambda'(t) := (\lambda(t), \mathit{right})$ if $t$ is a right node.
- Suppose $\lambda(t) = \mathrm{rename}_{i\to j}$ and let $s$ be the successor of $t$. Then $\rho(T_t, \lambda_{|T_t}) := (T', \lambda')$, where $T'$ is a tree defined as follows. Let $(T'', \lambda'') := \rho(T_s, \lambda_{|T_s})$ and let $r''$ be the root of $T''$. Then $T'$ is obtained from $T''$ by adding a new root $r'$, a new node $v$, and new edges $r'v$ and $vr''$. We define $\lambda'(r') := \mathrm{rename}_{(i,\mathit{left})\to(j,\mathit{left})}$, $\lambda'(v) := \mathrm{rename}_{(i,\mathit{right})\to(j,\mathit{right})}$, and $\lambda'(u) := \lambda''(u)$ for $u \in V(T'')$.
- Suppose $\lambda(t) = \mathrm{edge}_{E,i\to j}$ and let $s$ be the successor of $t$. Then $\rho(T_t, \lambda_{|T_t}) := (T', \lambda')$, where $T'$ is a tree defined as follows. Let $(T'', \lambda'') := \rho(T_s, \lambda_{|T_s})$ and let $r''$ be the root of $T''$. Then $T'$ is obtained from $T''$ by adding a path $v_1, v_2, v_3, v_4$ of length 3 and making $r''$ a successor of $v_4$. We define $\lambda'(v_1) := \mathrm{edge}_{E,(i,\mathit{left})\to(j,\mathit{left})}$, $\lambda'(v_2) := \mathrm{edge}_{E,(i,\mathit{right})\to(j,\mathit{left})}$, $\lambda'(v_3) := \mathrm{edge}_{E,(i,\mathit{left})\to(j,\mathit{right})}$, $\lambda'(v_4) := \mathrm{edge}_{E,(i,\mathit{right})\to(j,\mathit{right})}$, and $\lambda'(u) := \lambda''(u)$ for $u \in V(T'')$.
- Finally, suppose $\lambda(t) = \oplus$ and let $t_1, t_2$ be the successors of $t$ such that $t_1 \prec t_2$. Then $\rho(T_t, \lambda_{|T_t}) := (T', \lambda')$, where $T'$ is a tree defined as follows. For $i = 1, 2$, let $(T_i, \lambda_i) := \rho(T_{t_i}, \lambda_{|T_{t_i}})$ and let $r_i$ be the root of $T_i$. $T'$ consists of the union of $T_1$ and $T_2$, and the additional vertices $v_i$, $1 \leqslant i \leqslant k$, $w_{i,j}$, $1 \leqslant i, j \leqslant k$, and $v_\oplus$. We add the edges $v_i v_{i+1}$ for $1 \leqslant i < k$, the edges $w_{i,j} w_{i,j+1}$ for $1 \leqslant i \leqslant k$, $1 \leqslant j < k$, and the edges $w_{i,k} w_{i+1,1}$ for $1 \leqslant i < k$, and finally the edges $v_k w_{1,1}$, $w_{k,k} v_\oplus$, and $v_\oplus r_i$ for $i = 1, 2$. For every node $s \in V(T_i)$ we define $\lambda'(s) := \lambda_i(s)$, $i = 1, 2$. Furthermore, we define $\lambda(v_\oplus) := \oplus$ and $\lambda'(w_{i,j}) := \mathrm{edge}_{<,(i,\mathit{left})\to(j,\mathit{right})}$ for $1 \leqslant i, j \leqslant k$. Finally, if $t$ is a left node, then we define $\lambda(v_i) := \mathrm{rename}_{(i,\mathit{right})\to(i,\mathit{left})}$ for $i \leqslant k$, and if $t$ is a right node, then we define $\lambda(v_i) := \mathrm{rename}_{(i,\mathit{left})\to(i,\mathit{right})}$ for $i \leqslant k$.

Now, it is easily seen that $(T', \lambda')$ generates an $\{E, <\}$-structure $(V, E, <)$ where $(V, E)$ is the graph generated by $(T, \lambda)$ and $<$ is a linear order on $V$. The width of $(T', \lambda')$ is twice the width of $(T, \lambda)$, and hence at most $2^{k+2}$. □

We are now ready to prove Theorem 4.2.

Proof of Theorem 4.2. Let $\mathscr{C}$ be a class of graphs of clique-width at most $k$. On input $G \in \mathscr{C}$ and $\varphi \in \mathrm{MSO}_1[<\text{-inv}]$, we apply Lemma 4.7 to obtain a clique-expression $(T, \lambda)$ of width $2^{k+2}$ generating an ordered copy $(G, <)$ of $G$ in time $f(k) \cdot |G|^3$, for some computable function $f$. We can now apply Theorem 4.6 to decide whether $(G, <) \models \varphi$ in time $g(2^{k+2}) \cdot p(|G|)$, where $g$ is a computable function and $p$ a polynomial. As $\varphi$ is order invariant, if $(G, <) \models \varphi$, then $(G, <') \models \varphi$ for any linear order $<'$ on $G$. Hence, if $(G, <) \models \varphi$ we can accept and otherwise reject the input. □

It is worth pointing out the following feature of the model-checking algorithms established in Theorems 4.1 and 4.2. Instead of designing new model-checking algorithms, we reduce the verification of order-invariant MSO on classes of small treewidth or clique-width to the standard model-checking algorithms for MSO on classes of (slightly larger) treewidth and clique-width, respectively. The advantage of this approach is that we can reuse existing results on MSO on such classes of graphs. For instance, in Reference [37] the authors report on a practical implementation

of Courcelle's theorem, i.e., on the implementation of a model-checker for $\mathrm{MSO}_2$ on graph classes of bounded treewidth, and obtain astonishing performance results in practical tests. Our technique allows us to reuse this implementation so that with minimal effort it is possible to implement our algorithm on top of the work in Reference [37]. Furthermore, in Reference [22] it is shown that on graph classes $\mathscr{C}$ of bounded treewidth, the set of all satisfying assignments of a given MSO formula $\varphi(X)$ with free variables in a graph $G \in \mathscr{C}$ can be computed in time linear in the size of the output and the size of $G$. Again we can use the same algorithm to obtain the same result for order-invariant MSO.

## 5 SUCCESSOR-INVARIANT FO ON CLASSES OF BOUNDED EXPANSION

Classes of bounded expansion are classes of uniformly sparse graphs that have very good structural and algorithmic properties. Most notably, these classes admit efficient first-order model-checking, as shown by Dvořák et al. [15]. In this section we are going to lift this result to successor-invariant formulas. Let us give the required definitions first.

**Shallow minors and bounded expansion.** A graph $H$ is a *minor of* $G$, written $H \preccurlyeq G$, if there are pairwise vertex 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 $G$ and $r \in \mathbb{N}$, we write $\nabla_r(G)$ for the maximum edge density $|E(H)|/|V(H)|$ of a graph $H \preccurlyeq_r G$.

*Definition 5.1.* A class of graphs $\mathscr{C}$ has *bounded expansion* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\nabla_r(G) \leqslant f(r)$ for all $r \in \mathbb{N}$ and all $G \in \mathscr{C}$.

Let $\tau$ be a finite and purely relational signature and let $\mathscr{C}$ be a class of $\tau$-structures. We say that $\mathscr{C}$ has *bounded expansion* if the class $\{G(\mathbb{A}) : \mathbb{A} \in \mathscr{C}\}$ of the Gaifman-graphs of the structures from $\mathscr{C}$ has bounded expansion.

**Generalised colouring numbers.** We will mainly rely on an alternative characterisation of bounded expansion classes 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 $u \leqslant_L v$ if $u <_L v$ or $u = v$. For $r \in \mathbb{N}$, we say that a vertex $u$ is *strongly $r$-reachable* from a vertex $v$ with respect to $L$ if $u \leqslant_L v$ and there is a path $P$ of length at most $r$ that starts in $v$, ends in $u$, and all whose 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

$$\mathrm{col}_r(G, L) = \max_{v \in V(G)} \left|\mathrm{SReach}_r[G, L, v]\right|,$$

and the *$r$-colouring number* of $G$ (sometimes called *strong $r$-colouring number*) as

$$\mathrm{col}_r(G) = \min_{L \in \Pi(G)} \mathrm{col}_r(G, L).$$

For $r \in \mathbb{N}$ and ordering $L \in \Pi(G)$, the *$r$-admissibility* $\mathrm{adm}_r[G, L, v]$ 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 $v$, and is either the trivial length-zero path or ends in a vertex $u <_L v$ 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:

$$\mathrm{adm}_r(G, L) = \max_{v \in V(G)} \mathrm{adm}_r[G, L, v],$$

and the *r-admissibility of G* is given by

$$\mathrm{adm}_r(G) = \min_{L \in \Pi(G)} \mathrm{adm}_r(G, L).$$

The $r$-colouring numbers were introduced by Kierstead and Yang [36], while $r$-admissibility was first studied by Dvořák [14]. It was shown that those parameters are related as follows.

Fact 5.2 (Dvořák [14]). *For any graph $G$, $r \in \mathbb{N}$ and vertex ordering $L \in \Pi(G)$, we have*

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

(Note that in Dvořák's work, the reachability sets do not include the starting vertex, hence the above inequality is stated slightly differently in Reference [14].)

As proved by Zhu [60], 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.

Fact 5.3 (Zhu [60]). *A class $\mathscr{C}$ of graphs has bounded expansion if and only if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{col}_r(G) \leqslant f(r)$ for all $r \in \mathbb{N}$ and all $G \in \mathscr{C}$.*

We need to be a bit more precise and use the following lemma.

Fact 5.4 (Grohe et al. [30]). *For any graph $G$ and $r \in \mathbb{N}$ we have $\mathrm{adm}_r(G) \leqslant 6r \cdot (\nabla_r(G))^3$.*

As shown by Dvořák [14], on classes of bounded expansion one can compute $\mathrm{adm}_r(G)$ in linear fixed-parameter time, parameterised by $r$. More precisely, we have the following.

Fact 5.5 (Dvořák [14]). *Let $\mathscr{C}$ be a class of bounded expansion. Then there is an algorithm that, given a graph $G \in \mathscr{C}$ and $r \in \mathbb{N}$, 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 Reference [14] 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 5.2 we can use admissibility to obtain an approximation of the $r$-colouring number of a given graph from a class of bounded expansion.

Bounded expansion classes are very robust under local changes, e.g., under taking lexicographic products, as defined below.

*Definition 5.6.* Let $G$ and $H$ be graphs. The *lexicographic product* $G \bullet H$ of $G$ and $H$ is the graph with vertex set $V := V(G) \times V(H)$ and edge set

$$E := \Big\{\{(u, u'), (v, v')\} \,:\, uv \in E(G), \text{ or } v = u \text{ and } u'v' \in E(H)\Big\}.$$

In the above definition we use the notation $\{(u, u'), (v, v')\}$ for the edge $(u, u')(v, v')$ so that no confusion can arise. The following lemma shows that taking lexicographic products preserves the edge density of shallow minors. This was first proved in Reference [44]; the following improved bounds are given in Reference [32].

Fact 5.7 (Har-Peled and Quanrud [32]). *For any graph $G$ and $r, t \in \mathbb{N}$ we have $\nabla_r(G \bullet K_t) \leqslant 5t^2(r+1)^2\nabla_r(G)$.*

Bounded expansion classes are also stable under taking shallow minors, as expressed in the following fact.

Fact 5.8 (see Nešetřil and Ossona de Mendez [45, Proposition 4.1]). *If $J, H$, and $G$ are graphs and $r, s \in \mathbb{N}$ such that $J$ is a depth-$r$ minor of $H$ and $H$ is a depth-$s$ minor of $G$, then $J$ is a depth-$(2rs + r + s)$ minor of $G$.*

The following lemma is folklore, we provide a proof for completeness.

Lemma 5.9. *For any graph $G$ and $r \in \mathbb{N}$ we have $\nabla_r(G) \leqslant \mathrm{col}_{4r+1}(G)$.*

Proof. Set $c = \mathrm{col}_{4r+1}(G)$ and let $L$ be a linear order of $V(G)$ for which $\mathrm{col}_{4r+1}(G, L) = c$. Next let $H \preccurlyeq_r G$, say with a minor model $(I_u)_{u \in V(H)}$. We will show that $|E(H)| \leqslant c \cdot |V(H)|$.

For each $u \in V(H)$ let $m_u$ be the $<_L$-minimal vertex in $I_u$. We define a linear order on $V(H)$ by setting $u < v$ if $m_u <_L m_v$. Observe that since each branch set has radius at most $r$ and $m_u$ and $m_v$ are minimum in their respective branch sets, if $u < v$, there exists a vertex in $I_u$ that is strongly $(4r + 1)$-reachable from $m_v$. Hence, as $\mathrm{col}_{4r+1}(G) \leqslant c$, for each vertex $v$ there can be at most $c$ vertices $u$ with $u < v$. This implies that $H$ is $c$-degenerate and can have at most $c \cdot |V(H)|$ edges. □

We are going to prove the following theorem.

Theorem 5.10. *Let $\tau$ be a finite and purely relational signature and let $\mathscr{C}$ be a class of $\tau$-structures of bounded expansion. Then there exists an algorithm that, given a finite $\tau$-structure $\mathbb{A} \in \mathscr{C}$ and a successor-invariant formula $\varphi \in \mathrm{FO}[+1]$, verifies whether $\mathbb{A} \models_{\text{succ-inv}} \varphi$ in time $f(|\varphi|) \cdot n \cdot \alpha(n)$, where $f$ is a function and $n$ is the size of the universe of $\mathbb{A}$.*

In the language of parameterised complexity, Theorem 5.10 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. Note that Theorem 5.10 does not assert that $f$ is computable and, hence, the algorithm in the theorem is not necessarily strongly uniform. To have this property, it suffices to assume that the class $\mathscr{C}$ is *effectively of bounded expansion*. In the characterisation of Theorem 5.3, this means that there exist a *computable* function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{col}_r(G(\mathbb{A})) \leqslant f(r)$ for each $\mathbb{A} \in \mathscr{C}$. See Reference [31] for a similar discussion regarding model-checking first-order logic on (effectively) nowhere dense classes of graphs.

In principle, our approach follows that of the earlier results on successor-invariant model-checking. As $\varphi$ is successor invariant, to verify whether $\mathbb{A} \models_{\text{succ-inv}} \varphi$, we may compute an arbitrary successor relation $S$ on $V(\mathbb{A})$, and verify whether $(\mathbb{A}, S) \models \varphi$. Of course, we will try to compute a successor relation $S$ so that adding it to $\mathbb{A}$ preserves the structural properties as much as possible, so that model-checking on $(\mathbb{A}, S)$ can be done efficiently.

Our construction of such a structure preserving successor relation is based on the above described characterisation of bounded expansion classes by the generalised colouring numbers. As a first step, we show how to define a set $F$ of new edges with the following properties:

- $F$ forms a tree on the vertex set $V(G)$ of the input graph $G$,
- $F$ has maximum degree at most 3, and
- after adding all the edges of $F$ to $G$, the colouring numbers are still bounded.

In a second step, we construct from the bounded degree spanning tree a successor relation on $V(G)$, again ensuring that the relevant parameters remain bounded.

### 5.1 Constructing a Low-degree Spanning Tree

In this section, we prove the following theorem.

Theorem 5.11. *There exists an algorithm that, given a graph $G$, $r \in \mathbb{N}$, and 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*

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

*where $G + F$ is the graph $(V(G), E(G) \cup F)$. The running time of the algorithm is $O((n + m) \cdot \alpha(n))$, where $m = |E(G)|$ and $n = |V(G)|$.*

The main step toward this goal is the corresponding statement for connected graphs, as expressed in the following lemma.

Lemma 5.12. *For a connected graph $G$, the statement of Theorem 5.11 holds with the improved inequality*

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

We first show that Theorem 5.11 follows easily from Lemma 5.12.

Proof of Theorem 5.11, assuming Lemma 5.12. Let $G$ be a (possibly disconnected) graph, and let $G_1, \ldots, G_p$ be its connected components. For $i = 1, \ldots, p$, let $L_i$ be the ordering obtained by restricting $L$ to $V(G_i)$. Obviously $\mathrm{col}_{2r}(G_i, L_i) \leqslant \mathrm{col}_{2r}(G, L)$.

Apply the algorithm of Lemma 5.12 to $G_i$ and $L_i$, obtaining a set of unordered pairs $F_i$ such that $T_i = (V(G_i), F_i)$ is a tree of maximum degree at most 3 and

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

For $i = 1, \ldots, p$, 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

$$F = \{v_1v_2, v_2v_3, \ldots, v_{p-1}v_p\} \cup \bigcup_{i=1}^{p} F_i.$$

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 $v_iv_{i+1}$. By Lemma 5.12, the construction of each $T_i$ takes time $O((n_i + m_i) \cdot \alpha(n_i))$, where $m_i = |E(G_i)|$. It follows that the construction of $T$ takes time $O((n + m) \cdot \alpha(n))$.

It remains to argue that $\mathrm{adm}_r(G + F, L) \leqslant 2 + 2\mathrm{col}_{2r}(G, L)$. 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 $\{v_1v_2, v_2v_3, \ldots, v_{p-1}v_p\}$, since any such path has to use either $v_{i-1}v_i$ or $v_iv_{i+1}$. The remaining paths are entirely contained in $G_i + F_i$, and hence their number is bounded by $\mathrm{adm}_r(G_i + F_i, L_i) \leqslant 2\mathrm{col}_{2r}(G, L)$. The theorem follows. □

In the remainder of this section we focus on Lemma 5.12.

Proof of Lemma 5.12. 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. References [3, 54]) 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

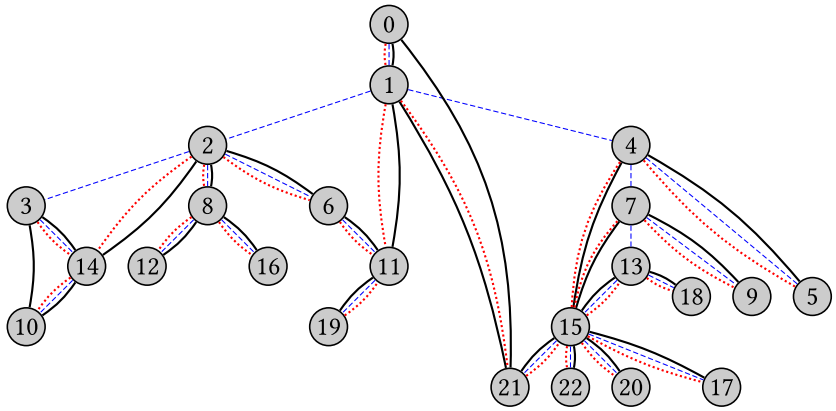


Fig. 2. A graph $G$ (solid black lines), the elimination tree $S$ (dashed blue lines), and a tree $U$ (dotted red lines).

elimination tree $S(G, L)$ is always equal to the vertex set of $G$. See Figure 2 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.

Claim 5.13. *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 $u \leqslant_L v$.*
*(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 5.13 (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 2, the edges of $U$ are represented by the dotted red lines.

Claim 5.14. *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

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

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 $U_u = (V(G_u), B \cap \binom{V(G_u)}{2})$ 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 $U_v$ to $u$ via the 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 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 5.15. *The tree $U$ can be computed in time $\mathcal{O}(m \cdot \alpha(n))$.*

Proof. We use the classic Union-Find 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 the representative of the class to which $u$ belongs, and (b) merge two equivalence classes into one. Tarjan [58] gave an implementation of this data structure where both operations run in amortised time $\alpha(n)$, where $n$ is the total number of elements covered by the data structure. Recall that $\alpha(\cdot)$ denotes the inverse Ackermann function.

Having initialised the data structure, we process the vertex ordering $L$ from the largest 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 $X' = X \cup \{u\}$. We iterate through all the neighbours of $u$, and for each neighbour $v$ of $u$ with $u <_L v$, verify whether the equivalence classes of $u$ and $v$ are different. If this is the case, then 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 a valid choice for 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, and thus the running time is $\mathcal{O}(m \cdot \alpha(n))$. 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 $u <_L v$. □

By Lemma 5.14 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 2: Vertex 4 is a child of 15 in $U$, and an ancestor of 15 in $S$.

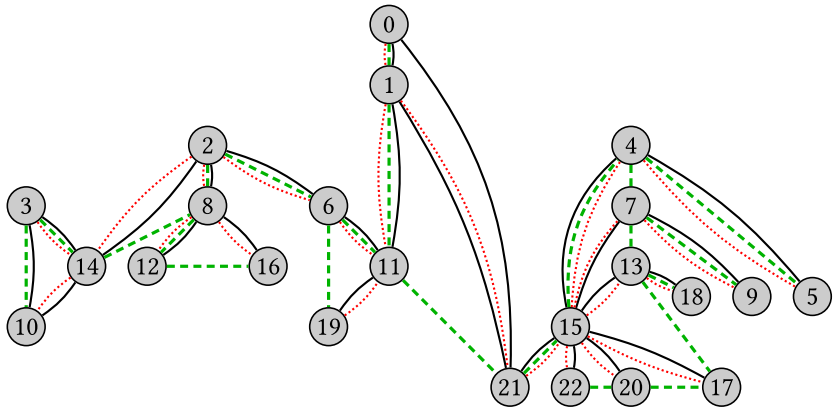


Fig. 3. A graph $G$ (solid black lines), a tree $U$ (dotted red lines), and the tree $T$ (thick dashed green lines).

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 3 for an illustration.

Claim 5.16. *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

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

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 $U$. □

Observe that once the tree $U$ is constructed, it is straightforward to construct $T$ in time $O(n)$. 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 5.13(3), each child of $u$ in $U$ is either its ancestor or descendant in $S$. By Claim 5.13(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 $Z_u^{\downarrow}$. However, 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 5.17. *The graph union $G + F$ satisfies* $\mathrm{adm}_r(G + F, L) \leqslant 2 \cdot \mathrm{col}_{2r}(G, L)$.

Proof. Write $H = G + F$. Let $F_{\text{new}} = F \setminus E(G)$ be the set of edges from $F$ that were not already present in $G$. If an edge $e \in F_{\text{new}}$ 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 edges joining a vertex $u$ with its children in $U$ were already present in $G$. We say that the vertex $u$ is the *origin* of an edge $e \in F_{\text{new}} \cap F_u$ and denote it by $a(e)$. Observe that $a(e)$ is adjacent to both endpoints of $e$ in $G$ by construction. For instance, in Figure 3 vertex 15 is the origin of the new edge between vertices 13 and 17, since this edge is contained in the path $F_{15}$. Also observe that if the endpoints of $e$ belong to $Z^{\uparrow}_{a(e)}$, then they are both ancestors of $a(e)$ in $S$, and thus are both smaller than $a(e)$ in $L$. Otherwise, if the endpoints of $e$ belong to $Z^{\downarrow}_{a(e)}$, then they are not bound by the ancestor-descendant relation in $S$ and $a(e)$ is their lowest common ancestor in $S$.

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 $F_{\text{new}}$ traversed on $P$, replace the usage of this edge on $P$ by the detour from $x$ to $a(e)$ to $y$ of length 2. 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 $|\mathrm{SReach}_{2r}[G, L, u]| \leqslant \mathrm{col}_{2r}(G, L)$, 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 two 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 2 and 3.) 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 2 and 3.) 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 $e \in F_{\text{new}}$ 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 $x \in Z^{\downarrow}_v$ or $x \in Z^{\uparrow}_v$. 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. We infer that $x \in Z^{\downarrow}_v$.

Recall that, by construction, $Z^{\downarrow}_v$ 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 $Z^{\downarrow}_v$ 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 one path from $\mathcal{P}$ can use these two edges. We can have $v = a(e)$ for this path only. Thus, this case contributes at most one path $P$ 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 $O(m \cdot \alpha(n))$ and $O(n)$, respectively. By Claims 5.16 and 5.17, $T$ satisfies the required properties. □

## 5.2 Constructing a Successor Relation

The preceding section provides us with a spanning tree of maximum degree at most 3. We now show how this can be used to obtain a successor relation from this spanning tree.

We give two constructions: One that constructs an actual successor relation, at the cost of possibly adding further edges. The added edges may increase the admissibility, but in a way that preserves bounded expansion. We also give a second construction that does not add additional edges and hence preserves also other structural properties. Such a construction may thus be potentially used for model-checking on other graph classes. This construction shows how a successor relation may be interpreted by a first-order formula in a graph with bounded-degree spanning tree, without adding any edges.

**Adding a successor relation.**

Theorem 5.18. *There exists an algorithm that, given a graph $G$, $r \in \mathbb{N}$, the edge set $F$ of a tree of maximum degree at most 3, computes a set of ordered pairs $S \subseteq V(G)^2$ such that $S$ is a successor relation on $V(G)$ and*

$$\operatorname{adm}_r(G + \bar{S}) \leqslant h\Big(r, \operatorname{adm}_{28r+13}(G + F)\Big),$$

*for an appropriately defined function $h$, where $\bar{S} = \{ab : (a, b) \in S\}$ is the set of undirected edges obtained from forgetting the directions of pairs in $S$. The running time of the algorithm is $O(m + n)$, where $m = |E(G)|$ and $n = |V(G)|$.*

As observed, e.g., in References [35, 56], the cube of every connected graph contains a Hamiltonian path. (The *cube* of a graph $G$ is the graph on the same vertex set as $G$ and in which two vertices are connected if their distance in $G$ is at most 3.) Furthermore, such a Hamiltonian path can be computed in linear time in the size of the original graph [42]. The set $S$ of edges whose existence is stated in Theorem 5.18 will simply be the Hamiltonian path computed in the cube of the spanning tree $F$ that we constructed above. It remains to prove the claimed bound on the $r$-admissibility of the new graph.

Proof of Theorem 5.18. Observe that we can find $G + S$, where $S$ is as described above, as a depth-3 minor of $(G + F) \bullet K_9$. This is a simple consequence of the fact that $F$ has maximum degree 3. Let $H \preccurlyeq_3 (G + F) \bullet K_9$ such that the edge density of $H$ is equal to $\nabla_3((G + F) \bullet K_9)$. Now we have

$$\begin{aligned}
\nabla_r(G + S) &\leqslant \nabla_r(H) \\
&\leqslant \nabla_{7r+3}((G + F) \bullet K_9) && \text{(by Fact 5.8)} \\
&\leqslant 5 \cdot 9^2 \cdot (7r + 4)^2 \cdot \nabla_{7r+3}(G + F) && \text{(by Fact 5.7)} \\
&\leqslant 5 \cdot 9^2 \cdot (7r + 4)^2 \cdot \operatorname{col}_{28r+13}(G + F) && \text{(by Lemma 5.9).}
\end{aligned}$$

Finally, by Lemma 5.4 we have $\mathrm{adm}_r(G+S) \leqslant 6r(\nabla_r(G+S))^3$, which gives us

$$\mathrm{adm}_r(G+S) \leqslant g(r, \mathrm{col}_{28r+13}(G+F)),$$

for an appropriately defined function $g$. Now the inequality of Lemma 5.2 leads to the stated result. □

**Interpreting a successor relation.** We show how in a graph with a spanning tree of degree 3, a successor relation can be interpreted after suitably colouring vertices and edges, but without adding further edges. We first notice that the existence of such a spanning tree guarantees the existence of a 3-walk, i.e., a walk through the graph that visits each vertex at least once and at most three times. The following lemma allows us to interpret a successor relation from a $k$-walk in first-order logic, for arbitrary $k$. For a natural number $\ell$, let $[\ell]$ be the set $\{1, \dots, \ell\}$.

Lemma 5.19. *Let $\sigma$ be a finite relational signature, $\mathbb{A}$ a finite $\sigma$-structure, and $w : [\ell] \to V(\mathbb{A})$ a $k$-walk through the Gaifman graph of $\mathbb{A}$, where $\ell \leqslant k \cdot |V(\mathbb{A})|$.*

*Then there is a finite relational signature $\sigma_k$ and a first-order formula $\varphi^{(k)}_{\mathrm{succ}}(x,y)$, both depending only on $k$, and a $(\sigma \cup \sigma_k)$-expansion $\mathbb{A}'$ of $\mathbb{A}$, which can be computed from $\mathbb{A}$ and $w$ in polynomial time, such that*

- *the Gaifman graphs of $\mathbb{A}'$ and $\mathbb{A}$ are the same;*
- *$\varphi^{(k)}_{\mathrm{succ}}$ defines a successor relation on $\mathbb{A}'$.*

Proof. We define a function $f : [\ell] \to [k]$, which counts how many times we have visited a vertex on the walk before, by

$$f(i) := |\{j \leqslant i \,:\, w(i) = w(j)\}|.$$

Furthermore, let $F : V(\mathbb{A}) \to [k]$ count how many times we visit a vertex:

$$F(v) := |\{i \in [\ell] \,:\, w(i) = v\}|.$$

To simplify notation, if $i \in [\ell]$, then we write $F(i)$ for $F(w(i))$.

We encode the $k$-walk $w$ by binary relations $E_{ab}$ with $a, b = 1, \dots, k$, in such a way that $(u,v) \in E_{ab}$ if and only if there is some $i \in [\ell - 1]$ such that

- $w(i) = u$ and $f(i) = a$, and
- $w(i+1) = v$ and $f(i+1) = b$.

That is, after visiting $u$ for the $a$th time, the walk $w$ proceeds to $v$, visiting it for the $b$th time. Note that if $k = 1$, we can immediately define a successor relation by

$$\varphi^{(1)}_{\mathrm{succ}}(x,y) := E_{11}xy.$$

If $k > 1$, then we show how to interpret a $(k-1)$-walk $w'$ in first-order logic, given a $k$-walk encoded by $\{E_{ab} \,:\, 1 \leqslant a, b \leqslant k\}$ as above. By daisy-chaining these interpretations we end up with a 1-walk (i.e., a Hamiltonian path). Plugging in the interpretation of this Hamiltonian path into $\varphi^{(1)}_{\mathrm{succ}}$ defined above gives the formulas $\varphi^{(k)}_{\mathrm{succ}}$.

To get from a $k$-walk to a $(k-1)$-walk, we look at all vertices that are visited $k$ times, and "jump" over these vertices, either when they are visited for the $(k-1)$th or for the $k$th time. Jumping over a vertex can be done in first-order logic, but we must be careful to choose the vertices for jumping in such a way that we never jump over an unbounded number of vertices in a row, as this is not possible in first-order logic. We encode the information on whether to jump when visiting for the $(k-1)$th or the $k$th time in a new unary predicate $P_k$.

To be precise, let $\varphi_{k\text{-times}}(x)$ be a formula that states that $x$ is visited $k$ times:

$$\varphi_{k\text{-times}}(x) \coloneqq \bigvee_{a=1}^{k} \exists y\, E_{ak} yx \vee \bigvee_{a=1}^{k} \exists y\, E_{ka} xy.$$

For those $u \in V(\mathbb{A})$ that are visited $k$-times, we agree to jump over them when they are visited for the $k$th time if $u \in P_k$, and when they are visited for the $(k-1)$th time otherwise. Thus, if $w(i) = u$, $f(i) = k$ and $u \in P_k$, we want to remove the $i$th step. However, it may be the case that $w(i+1)$ is also visited $k$ times and needs to be jumped over. We define first-order formulas that carry out a bounded number of such jumps as follows.

- For $a \in [k]$, the formula $\varphi_{\text{jump},a}(x)$ holds if we jump over $x$ when visiting it for the $a$th time:

$$\begin{aligned} \varphi_{\text{jump},1}(x), \ldots, \varphi_{\text{jump},k-2}(x) &\coloneqq \bot, \\ \varphi_{\text{jump},k-1}(x) &\coloneqq \varphi_{k\text{-times}}(x) \wedge \neg P_k x, \\ \varphi_{\text{jump},k}(x) &\coloneqq \varphi_{k\text{-times}}(x) \wedge P_k x. \end{aligned}$$

- For $r \geqslant 0$ and $a, b \in [k]$, the formula $\varphi^{(r)}_{\text{next},a,b}(x,y)$ holds if, when applying at most $r$ consecutive jumps on entering $x$ for the $a$th time, we end up in node $y$, which is visited for the $b$th time in the (original) walk. Specifically:

$$\begin{aligned} \varphi^{(0)}_{\text{next},a,b}(x,y) &\coloneqq x \doteq y \wedge \delta_{ab}, \\ \varphi^{(r+1)}_{\text{next},a,b}(x,y) &\coloneqq \Big(\neg\varphi_{\text{jump},a}(x) \to (x \doteq y \wedge \delta_{ab})\Big) \\ &\quad \wedge \Big(\varphi_{\text{jump},a}(x) \to \exists z \bigvee_{c=1}^{k} \Big(E_{ac} xz \wedge \varphi^{(r)}_{\text{next},c,b}(z,y)\Big)\Big). \end{aligned}$$

  Here, $\delta_{ab}$ is true if the indices $a$ and $b$ are the same:

$$\delta_{ab} \coloneqq \begin{cases} \top, & \text{if } a = b; \\ \bot, & \text{otherwise.} \end{cases}$$

- We will show below how to choose the predicate $P_k$ so that we never need to take more than two consecutive jumps. Thus, we can interpret a $(k-1)$-walk $w'$ using, for $a, b \in [k-2]$, the formulas

$$\varphi_{E,a,b}(x,y) \coloneqq \exists z \bigvee_{c=1}^{k} \Big(E_{ac} xz \wedge \varphi^{(2)}_{\text{next},c,b}(z,y)\Big).$$

  For $a \in [k-2]$ we set

$$\varphi_{E,a,k-1}(x,y) \coloneqq \exists z \bigvee_{c=1}^{k} \Big(E_{ac} xz \wedge \Big(\varphi^{(2)}_{\text{next},c,k-1}(z,y) \vee \varphi^{(2)}_{\text{next},c,k}(z,y)\Big)\Big).$$

  Next, for $b \in [k-2]$ we set

$$\begin{aligned} \varphi_{E,k-1,b}(x,y) &\coloneqq \Big(\neg\varphi_{\text{jump},k-1}(x) \to \exists z \bigvee_{c=1}^{k} \Big(E_{k-1,c} xz \wedge \varphi^{(2)}_{\text{next},c,b}(z,y)\Big)\Big) \\ &\quad \wedge \Big(\varphi_{\text{jump},k-1}(x) \to \exists z \bigvee_{c=1}^{k} \Big(E_{k,c} xz \wedge \varphi^{(2)}_{\text{next},c,b}(z,y)\Big)\Big), \end{aligned}$$

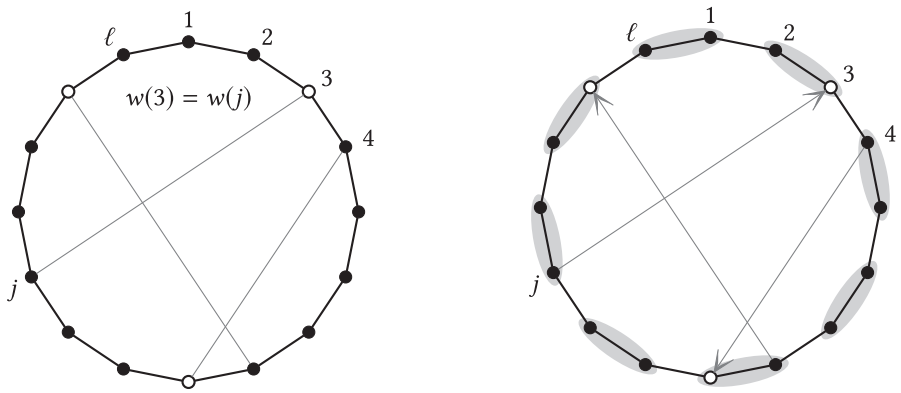


Fig. 4. Deciding when to jump over vertices in a $k$-walk.

and, finally, we define

$$\varphi_{E,k-1,k-1}(x,y) \coloneqq \left(\neg\varphi_{\text{jump},k-1}(x) \to \exists z \bigvee_{c=1}^{k}\left(E_{k-1,c}xz \wedge \left(\varphi^{(2)}_{\text{next},c,k-1}(z,y) \vee \varphi^{(2)}_{\text{next},c,k}(z,y)\right)\right)\right)$$
$$\wedge \left(\varphi_{\text{jump},k-1}(x) \to \exists z \bigvee_{c=1}^{k}\left(E_{k,c}xz \wedge \left(\varphi^{(2)}_{\text{next},c,k-1}(z,y) \vee \varphi^{(2)}_{\text{next},c,k}(z,y)\right)\right)\right).$$

To define the predicate $P_k$, let $T \subseteq [\ell]$ be the set of indices $i \in [\ell]$ for which $F(i) = k$ and $f(i) \in \{k-1, k\}$. We obtain a perfect matching $M$ on $T$ by matching $i$ and $j$ if and only if $w(i) = w(j)$ (cf. Figure 4 (a)). We define a subset $J \subset [\ell]$ with the intended meaning that if $i \in J$, we jump over the $i$th step of $w$. The set $J$ will satisfy the following two conditions:

- every vertex $v$ with $F(v) = k$ is jumped over exactly once, i.e.,
$$\left|\{i \in [\ell] \,:\, w(i) = v\} \cap J\right| = 1, \quad \text{and}$$
- we never jump more than twice in a row, i.e., if $i, i+1 \in J$, then $i+2 \notin J$.

We partition the set $[\ell]$ into intervals of size 2, setting

$$U \coloneqq \Big\{\{1,2\},\{3,4\},\dots\Big\},$$

with the last set $\{\ell\}$ being a singleton if $\ell$ is odd. Then the matching $M$ defines a multigraph without loops on $U$, and the degree of $I \in U$ is at most 2. We direct the edges of $M$, viewed as edges in the multigraph $(U, M)$, in such a way that every $I \in U$ has at most one incoming edge. This is possible because the multigraph $(U, M)$ can be decomposed into vertex-disjoint cycles and paths; we then orient the edges cyclically on each cycle and from one end to the other along each path.

The edges incident with $I$ correspond to the elements of $I \cap T$, and we put $i \in I$ into $J$ if and only if the edge corresponding to $i$ is directed toward $I$ (cf. Figure 4 (b)). For every $k = 1, \dots, \lfloor \frac{1}{2}(\ell - 1) \rfloor$ at most one of $2k-1$ and $2k$ is in $J$, and therefore $J$ satisfies the above requirements.

The definition of $P_k \subseteq V(G)$ is now straightforward:

$$P_k \coloneqq \{v \in V(G) \,:\, F(v) = k \text{ and } f(i) = k \text{ for the } i \in J \text{ with } w(i) = v\}.$$

In summary, we end up with

$$\sigma_k := \{E_{ab} : a, b \in [k]\} \cup \{P_a : a = 2, \ldots, k\},$$

and it is clear that our construction can be carried out in polynomial time. □

### 5.3 Proof of Theorem 5.10

Let us finally derive the main theorem, Theorem 5.10. 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. [15] and of Grohe and Kreutzer [29]. We also refer to the new expositions given in References [27, 49].

Theorem 5.20. *Let $\tau$ 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 $\varphi$, such that the following holds. Given a $\tau$-structure $\mathbb{A}$, it can be verified whether $\mathbb{A} \models \varphi$ in time $f(|\varphi|, \mathrm{adm}_{r(\varphi)}(G(\mathbb{A}))) \cdot n$, where $n$ is the size of the universe of $\mathbb{A}$ and $f$ is a computable function.*

Observe that if $\mathbb{A}$ is drawn from a fixed class of bounded expansion $\mathscr{C}$, then $\mathrm{adm}_{r(\varphi)}(G(\mathbb{A}))$ is a parameter depending only on $\varphi$, hence we recover fixed-parameter tractability of model-checking for FO on any class of bounded expansion, parameterised by the length of the formula. Theorem 5.20 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 $\varphi$. 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 5.20 ensures that this is sufficient for the model-checking problem to remain tractable.

We now sketch how Theorem 5.20 may be derived from the works of Dvořák et al. [15] and of Grohe and Kreutzer [29]. We prefer to work with the algorithm of Grohe and Kreutzer [29], because we find it conceptually simpler. For a given quantifier rank $q$ and an nonnegative integer $i \leqslant q$, the algorithm computes the set of all *types* $\mathfrak{R}_i^q$ realised by $i$-tuples in the input structure $\mathbb{A}$: 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 $\Gamma$ is a set of colours, such that for any subset $C \subseteq \Gamma$ of $i$ colours, $i \leqslant p$, 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 [60], the treedepth-$p$ chromatic numbers are bounded in terms of $r$-colouring numbers as follows.

Theorem 5.21 (Zhu [60]). *For any graph $G$ and $p \in \mathbb{N}$ we have*

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

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

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 $\gamma$, as well as binary relations representing edges of appropriate treedepth decompositions certifying that $\gamma$ 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 $\gamma$ depends only on $q$ and $i$ in a computable manner. Thus, by Theorem 5.21, to ensure that $\gamma$ uses a bounded number of colours, we only need to ensure the boundedness of $\mathrm{col}_{r(q)}(G(\mathbb{A}))$ for some computable function $r(q)$. By taking $q$ to be the quantifier rank of the input formula, the statement of Theorem 5.20 follows.

We can now combine all the ingredients and show how our main result follows from Theorem 5.18.

Proof of Theorem 5.10. Given a successor-invariant formula $\varphi \in \mathrm{FO}[\tau \cup \{S\}]$, we first compute the integer $r := r(\varphi)$ whose existence and computability is stated in Theorem 5.20. Given the structure $\mathbb{A}$, we now use the algorithm of Theorem 5.5 to compute an order $L$ of the vertex set $V(\mathbb{A})$ of the Gaifman graph of $\mathbb{A}$, which satisfies $\mathrm{adm}_{56r+26}(G(\mathbb{A})) \leqslant c(r)$ for some constant $c(r)$. Such constant exists by the assumption that $\mathbb{A}$ is from a class of bounded expansion. We use the algorithm of Theorem 5.11 to compute a set of unordered pairs $F \subseteq \binom{V(\mathbb{A})}{2}$ such that the graph $T = (V(\mathbb{A}), F)$ is a tree of maximum degree at most 3 and

$$\mathrm{adm}_{28r+13}(G(\mathbb{A}) + F, L) \leqslant 2 + 2 \cdot \mathrm{col}_{56r+26}(G(\mathbb{A}), L).$$

By Lemma 5.2 we find $\mathrm{col}_{56r+26}(G(\mathbb{A}), L) \leqslant \mathrm{adm}_{56r+26}(G(\mathbb{A}), L)^{56r+26} \leqslant c(r)^{56r+26}$. This means that $\mathrm{adm}_{28r+13}(G(\mathbb{A}) + F, L) \leqslant g(r)$ for $g(r) = 2 + 2c(r)^{56r+26}$. Now, using the algorithm of Theorem 5.18 we compute a successor relation $S$ such that

$$\mathrm{adm}_r(G(\mathbb{A}) + S) \leqslant h(r, \mathrm{adm}_{28r+13}(G(\mathbb{A}) + F)),$$

where $h$ is the function from Theorem 5.18. Finally, we apply the algorithm of Theorem 5.20 to decide whether $(\mathbb{A}, S) \models \varphi$ in time $f(|\varphi|, \mathrm{adm}_r(G(\mathbb{A}) + S)) \cdot n$. Since $\mathbb{A}$ is drawn from a fixed class of bounded expansion $\mathscr{C}$, $\mathrm{adm}_r(G(\mathbb{A}) + S)$ is a parameter depending only on $\varphi$. This finishes the proof of the theorem. □

## 6 DENSE GRAPHS

While model-checking for first-order logic has been studied rather thoroughly for sparse graph classes, few results are known for dense graphs.

- On classes of graphs with bounded clique-width (or, equivalently, bounded rank-width; cf. [48]), model-checking even for monadic second-order logic has been shown to be fixed-parameter tractable by Courcelle et al. [9].
- More recently, model-checking on coloured posets of bounded width has been shown to be fixed-parameter tractable for existential FO by Bova et al. [4] and for all of FO by Gajarský et al. [26].

Both of these results extend to order-invariant FO and therefore also to successor-invariant FO. For bounded clique-width, this has already been shown in Section 4 . For posets of bounded width we give a proof here. We first review the necessary definitions.

*Definition 6.1.* A *partially ordered set (poset)* $(P, \leqslant^P)$ is a set $P$ with a reflexive, transitive and antisymmetric binary relation $\leqslant^P$. A *chain* $C \subseteq P$ is a totally ordered subset, i.e., for all $x, y \in C$ one of $x \leqslant^P y$ and $y \leqslant^P x$ holds. An *antichain* is a set $A \subseteq P$ such that if $x \leqslant^P y$ for $x, y \in A$, then $x = y$. The *width* of $(P, \leqslant^P)$ is the maximal size $|A|$ of an antichain $A \subseteq P$.

A *coloured* poset is a poset $(P, \leqslant^P)$ together with a function $\lambda : P \to \Lambda$ mapping $P$ to some set $\Lambda$ of *colours*.

By $|P|$ we denote the length of a suitable encoding of $(P, \leqslant^P)$.

We will need Dilworth's Theorem, which relates the width of a poset to the minimum number of chains needed to cover the poset.

Theorem 6.2 (Dilworth's Theorem). *Let $(P, \leqslant^P)$ be a poset. Then the width of $(P, \leqslant^P)$ is equal to the minimum number $k$ of disjoint chains $C_1, \dots, C_k \subseteq P$ needed to cover $P$, i.e., such that $\bigcup_i C_i = P$.*

A proof can be found, e.g., in Reference [12, Sec. 2.5]. Moreover, by a result of Felsner et al. [21], both the width $w$ and a set of chains $C_1, \dots, C_w$ covering $P$ can be computed from $(P, \leqslant_P)$ in time $O(w \cdot |P|)$.

With this, we are ready to prove the following.

Theorem 6.3. *There is an algorithm that, on input a coloured poset $(P, \leqslant^P)$ with colouring $\lambda : P \to \Lambda$ and an order-invariant first-order formula $\varphi$, checks whether $P \models \varphi$ in time $f(w, |\varphi|) \cdot |P|^2$ where $w$ is the width of $(P, \leqslant^P)$.*

Proof. Using the algorithm of Reference [21], we compute a chain cover $C_1, \dots, C_w$ of $(P, \leqslant^P)$. To obtain a linear order on $P$, we just need to arrange the chains in a suitable order, which can be done by colouring the vertices with colours $\Lambda \times [w]$ via

$$\lambda'(v) = (\lambda(v), j), \quad \text{for } v \in C_j.$$

Then

$$\varphi_{\leqslant}(x, y) := \Big( \bigvee_{\substack{\lambda_x, \lambda_y \in \Lambda, \\ i < j}} (\lambda'(x) = (\lambda_x, i) \wedge \lambda'(y) = (\lambda_y, j)) \Big) \vee \Big( \bigvee_{\substack{\lambda_x, \lambda_y \in \Lambda, \\ i \in [w]}} (\lambda'(x) = (\lambda_x, i) \wedge \lambda'(y) = (\lambda_y, i) \wedge x \leqslant y) \Big)$$

defines a linear order on $(P, \leqslant_P)$ with colouring $\lambda'$. After substituting $\varphi_{\leqslant}$ for $\leqslant$ in $\varphi$, we may apply the algorithm of Gajarský et al. [26] to check whether $P \models \varphi$. □

## 7 CONCLUSION

We analysed the parameterized complexity of FO and MSO model-checking on graphs in the presence of a linear order or a successor relation. We showed that if the linear order or successor relation is part of the input, then FO (and hence in particular MSO) model-checking is AW[∗]-hard even on partial matchings and star forests, respectively. As FO model-checking is fixed-parameter tractable on nowhere dense graph classes, the classes obtained in the reduction by combining a successor relation with star forests cannot be nowhere dense. In fact, it is easily observed that one can find all graphs as depth-1 minors in the class we construct in the reduction.

However, in the successor-invariant case we are allowed to choose a successor relation that is compatible with the input structures, in the sense that good structural properties are preserved. We showed that this is possible in the case that the input structures come from a class of bounded expansion, and conclude that model-checking for successor-invariant first-order formulas is fixed-parameter tractable on classes of bounded expansion.

It remains an open problem whether a similar construction is possible in nowhere dense graph classes, which constitute the currently known limit of tractability for plain first-order model-checking. Our approach does not generalise to nowhere dense graphs for the following reason. The analog of Fact 5.3 for nowhere dense graphs is that a class $\mathscr{C}$ of graphs is nowhere dense if and only if there is a function $f : \mathbb{N} \times \mathbb{R} \to \mathbb{N}$ such that for all $r \in \mathbb{N}$ and all real $\varepsilon > 0$ we have $\mathrm{col}_r(H) \leqslant f(r, \varepsilon) \cdot n^\varepsilon$ for all $n$-vertex subgraphs $H$ of graphs $G \in \mathscr{C}$. Hence, if we follow our approach and construct for every graph $G \in \mathscr{C}$ a successor relation, or for our purpose equivalently a low degree spanning tree $F_G$, it is not sufficient to prove the analog of Theorem 5.11. While our proofs show that for all $n$-vertex graphs $G \in \mathscr{C}$ we have $\mathrm{col}_r(G + F_G) \leqslant f(r, \varepsilon) \cdot n^\varepsilon$, this is not sufficient to prove that the class $\mathscr{D} = \{G + F_G : G \in \mathscr{C}\}$ is nowhere dense. We need to establish bounds for the $r$-colouring numbers for all subgraphs of graphs from $\mathscr{D}$. This statement may in fact not be true, as in the characterisation of nowhere dense graphs via the colouring numbers we may have to construct a different order $L_H$ for each $n$-vertex subgraph $H$ such that $\mathrm{col}_r(H, L_H) \leqslant f(r, \varepsilon) \cdot n^\varepsilon$ is satisfied. Our approach does also not generalise to solve the order-invariant case. The reason is that the locality based arguments that are applied to test first-order properties of bounded expansion or nowhere dense graph classes can no longer be applied in the presence of an order relation.

For order-invariant MSO, we proved that the complexity of its model-checking is the same as for plain MSO: for $\mathrm{MSO}_2$ it is fixed-parameter tractable on classes of bounded treewidth and for $\mathrm{MSO}_1$ on classes of bounded clique-width.

## ACKNOWLEDGMENTS

We thank Christoph Dittmann and Viktor Engelmann for many hours of fruitful discussions. We thank the anonymous reviewers for their careful reading of the manuscript and their many insightful comments and suggestions.

## REFERENCES

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

[2] Hans L. Bodlaender. 1996. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* 25, 6 (1996), 1305–1317.

[3] Hans L. Bodlaender, Jitender S. Deogun, Klaus Jansen, Ton Kloks, Dieter Kratsch, Haiko Müller, and Zsolt Tuza. 1998. Rankings of graphs. *SIAM J. Discr. Math.* 11, 1 (1998), 168–181.

[4] Simone Bova, Robert Ganian, and Stefan Szeider. 2015. Model checking existential logic on partially ordered sets. *ACM Trans. Comput. Log.* 17, 2 (2015), 10:1–10:35.

[5] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC'77)*. ACM, 77–90.

[6] Yijia Chen and Jörg Flum. 2012. On the ordered conjecture. In *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'12)*. IEEE Computer Soc., 225–234.

[7] Bruno Courcelle. 1990. Graph rewriting: An algebraic and logic approach. In *Handbook of Theoretical Computer Science, Vol. B*, Jan van Leeuwen (Ed.). Elsevier, 194–242.

[8] Bruno Courcelle. 1996. The monadic second-order logic of graphs. X. Linear orderings. *Theor. Comput. Sci.* 160, 1 (1996), 87–143.

[9] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.* 33, 2 (2000), 125–150.

[10] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

[11] Anuj Dawar, Martin Grohe, and Stephan Kreutzer. 2007. Locally excluding a minor. In *Proceeedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07)*. IEEE Computer Soc., 270–279.
[12] Reinhard Diestel. 2010. *Graph Theory* (4th ed.). Graduate Texts in Mathematics, Vol. 173. Springer.
[13] Rodney G. Downey and Michael R. Fellows. 1999. *Parameterized Complexity*. Springer.
[14] Zdeněk Dvořák. 2013. Constant-factor approximation of the domination number in sparse graphs. *Eur. J. Combin.* 34, 5 (2013), 833–840.
[15] Zdeněk Dvořák, Daniel Král', and Robin Thomas. 2013. Testing first-order properties for subclasses of sparse graphs. *J. ACM* 60, 5 (2013), Art. 36.
[16] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. 2014. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *39th International Symposium on Mathematical Foundations of Computer Science (MFCS 2014)*. Lecture Notes in Computer Science, Vol. 8634. Springer, 256–266.
[17] Kord Eickmeyer and Ken-ichi Kawarabayashi. 2016. Successor-invariant first-order logic on graphs with excluded topological subgraphs. In *Proceedings of the Annual Conference on Computer Science Logic (CSL'16)*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 62. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 18:1–18:15.
[18] Kord Eickmeyer, Ken-ichi Kawarabayashi, and Stephan Kreutzer. 2013. Model checking for successor-invariant first-order logic on minor-closed graph classes. In *Proceedings of the 28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'13)*. IEEE Computer Soc., 134–142.
[19] Michael Elberfeld, Marlin Frickenschmidt, and Martin Grohe. 2016. Order invariance on decomposable structures. In *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS'16)*. ACM, 397–406.
[20] Viktor Engelmann, Stephan Kreutzer, and Sebastian Siebertz. 2012. First-order and monadic second-order model-checking on ordered structures. In *Proceedings of the 27th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS'12)*. IEEE Computer Soc., 275–284.
[21] Stefan Felsner, Vijay Raghavan, and Jeremy Spinrad. 2003. Recognition algorithms for orders of small width and graphs of small Dilworth number. *Order* 20, 4 (2003), 351–364.
[22] Jörg Flum, Markus Frick, and Martin Grohe. 2002. Query evaluation via tree-decompositions. *J. ACM* 49, 6 (2002), 716–752.
[23] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer.
[24] Marcus Frick and Martin Grohe. 2001. Deciding first-order properties of locally tree-decomposable structures. *J. ACM* 48, 6 (2001), 1184–1206.
[25] Haim Gaifman. 1982. On local and nonlocal properties. In *Proceedings of the Herbrand Symposium*, J. Stern (Ed.). Stud. Logic Found. Math., Vol. 107. North-Holland, 105–135.
[26] Jakub Gajarský, Petr Hliněný, Daniel Lokshtanov, Jan Obdržálek, Sebastian Ordyniak, M. S. Ramanujan, and Saket Saurabh. 2015. FO model checking on posets of bounded width. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS'15)*. IEEE Computer Soc., 963–974.
[27] Jakub Gajarský, Stephan Kreutzer, Jaroslav Nešetřil, Patrice Ossona de Mendez, Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. 2018. First-order interpretations of bounded expansion classes. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP'18)*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 107. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 126:1–126:14.
[28] Tobias Ganzow and Sasha Rubin. 2008. Order-invariant MSO is stronger than counting MSO in the finite. In *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science (STACS'08)*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 313–324.
[29] Martin Grohe and Stephan Kreutzer. 2011. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics*, Martin Grohe and Johann A. Makowsky (Eds.). Contemp. Math., Vol. 558. AMS, 181–205.
[30] Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos Stavropoulos. 2015. Colouring and covering nowhere dense graphs. In *Proceedings of the 41st International Workshop on Graph-Theoretic Concepts in Computer Science (WG'15)*. Lecture Notes in Comput. Sci., Vol. 9224. Springer, 325–338.
[31] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. 2017. Deciding first-order properties of nowhere dense graphs. *J. ACM* 64, 3 (2017), 17:1–17:32.
[32] Sariel Har-Peled and Kent Quanrud. 2017. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM J. Comput.* 46, 6 (2017), 1712–1744.
[33] Jan van den Heuvel, Stephan Kreutzer, Michał Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. 2017. Model-checking for successor-invariant first-order formulas on graph classes of bounded expansion. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*. IEEE Computer Soc., 1–11.
[34] Petr Hliněný and Sang-il Oum. 2008. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.* 38, 3 (2008), 1012–1032.
[35] Jerome J. Karaganis. 1968. On the cube of a graph. *Can. Math. Bull.* 11, 2 (1968), 295–296.
[36] H. A. Kierstead and Daqing Yang. 2003. Orderings on graphs and game coloring number. *Order* 20, 3 (2003), 255–264.

[37] Joachim Kneis, Alexander Langer, and Peter Rossmanith. 2011. Courcelle's theorem—A game-theoretic approach. *Discr. Optim.* 8, 4 (2011), 568–594.
[38] Stephan Kreutzer, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. 2016. The generalised colouring numbers on classes of bounded expansion. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS'16)*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 58. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 85:1–85:13.
[39] Stephan Kreutzer and Siamak Tazari. 2010. Lower bounds for the complexity of monadic second-order logic. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS 2010)*. IEEE Computer Soc., 189–198.
[40] Stephan Kreutzer and Siamak Tazari. 2010. On brambles, grid-like minors, and parameterized intractability of monadic second-order logic. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*. SIAM, 354–364.
[41] Leonid Libkin. 2004. *Elements of Finite Model Theory*. Springer.
[42] Yaw-Ling Lin and Steven S. Skiena. 1995. Algorithms for square roots of graphs. *SIAM J. Discr. Math.* 8 (1995), 99–118.
[43] Johann A. Makowsky. 2005. Coloured Tutte polynomials and Kauffman brackets for graphs of bounded tree width. *Discr. Appl. Math.* 145 (2005), 276–290.
[44] Jaroslav Nešetřil and Patrice Ossona de Mendez. 2008. Grad and classes with bounded expansion. I. Decompositions. *Eur. J. Combin.* 29, 3 (2008), 760–776.
[45] Jaroslav Nešetřil and Patrice Ossona de Mendez. 2012. *Sparsity—Graphs, Structures, and Algorithms*. Algorithms and Combinatorics, Vol. 28. Springer.
[46] Hannu Niemistö. 2005. On locality and uniform reduction. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*. IEEE Computer Soc., 41–50.
[47] Martin Otto. 2000. Epsilon-logic is more expressive than first-order logic over finite structures. *J. Symbol. Logic* 65 (2000), 1749–1757.
[48] Sang-il Oum and Paul Seymour. 2006. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B* 96, 4 (2006), 514–528.
[49] Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. 2018. Parameterized circuit complexity of model-checking on sparse structures. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'18)*. ACM, 789–798.
[50] Andreas Potthoff. 1994. *Logische Klassifizierung Regulärer Baumsprachen*. Ph.D. Dissertation. Universität Kiel. [in German].
[51] Daniel A Quiroz. 2017. *Chromatic and Structural Properties of Sparse Graph Classes*. Ph.D. Dissertation. The London School of Economics and Political Science.
[52] Benjamin Rossman. 2003. Successor-invariance in the finite. In *Proceedings of the 18th Annual IEEE Symposium of Logic in Computer Science (LICS'03)*. IEEE Computer Soc., 148–157.
[53] Benjamin Rossman. 2007. Successor-invariant first-order logic on finite structures. *J. Symbol. Logic* 72, 2 (2007), 601–618.
[54] Alejandro A. Schäffer. 1989. Optimal node ranking of trees in linear time. *Inform. Process. Lett.* 33, 2 (1989), 91–96.
[55] Detlef Seese. 1996. Linear time computable problems and first-order descriptions. *Math. Struct. Comput. Sci.* 6, 6 (1996), 505–526.
[56] Milan Sekanina. 1963. On an ordering of the vertices of a graph. *Časopis Pěst. Mat.* 88, 3 (1963), 265–282.
[57] Larry J. Stockmeyer. 1976. The polynomial-time hierarchy. *Theoret. Comput. Sci.* 3, 1 (1976), 1–22.
[58] Robert Endre Tarjan. 1975. Efficiency of a good but not linear set union algorithm. *J. ACM* 22, 2 (1975), 215–225.
[59] Moshe Y. Vardi. 1982. The complexity of relational query languages. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC'82)*. ACM, 137–146.
[60] Xuding Zhu. 2009. Colouring graphs with bounded generalized colouring number. *Discr. Math.* 309, 18 (2009), 5562–5568.

Received December 2018; revised June 2019; accepted August 2019