Successor-Invariant First-Order Logic on Graphs with Excluded Topological Subgraphs

Kord Eickmeyer[1] **and Ken-ichi Kawarabayashi**[2]

1 **Technical University Darmstadt, Department of Mathematics, Schlossgartenstr. 7, 64289 Darmstadt, Germany**
`eickmeyer@mathematik.tu-darmstadt.de`

2 **National Inst. of Informatics, Tokyo, Japan; and JST, ERATO, Kawarabayashi Large Graph Project, Chiyoda-ku, Hitotsubashi 2-1-2, Tokyo 101-8430, Japan**
`k_keniti@nii.ac.jp`

Abstract

We show that the model-checking problem for successor-invariant first-order logic is fixed-parameter tractable on graphs with excluded topological subgraphs when parameterised by both the size of the input formula and the size of the exluded topological subgraph. Furthermore, we show that model-checking for order-invariant first-order logic is tractable on coloured posets of bounded width, parameterised by both the size of the input formula and the width of the poset.

Results of this form, i.e. showing that model-checking for a certain logic is tractable on a certain class of structures, are often referred to as algorithmic meta-theorems since they give a unified proof for the tractability of a whole range of problems. First-order logic is arguably one of the most important logics in this context since it is powerful enough to express many computational problems (e.g. the existence of cliques, dominating sets etc.) and yet its model-checking problem is tractable on rich classes of graphs. In fact, Grohe et al. [21] have shown that model-checking for FO is tractable on all nowhere dense classes of graphs.

Successor-invariant FO is a semantic extension of FO by allowing the use of an additional binary relation which is interpreted as a directed Hamiltonian cycle, restricted to formulae whose truth value does not depend on the specific choice of a Hamiltonian cycle. While this is very natural in the context of model-checking (after all, storing a structure in computer memory usually brings with it a linear order on the structure), the question of how the computational complexity of the model-checking problem for this richer logic compares to that of plain FO is still open.

Our result for successor-invariant FO extends previous results for this logic on planar graphs [14] and graphs with excluded minors [13], further narrowing the gap between what is known for FO and what is known for successor-invariant FO. The proof uses Grohe and Marx's structure theorem for graphs with excluded topological subgraphs [22]. For order-invariant FO we show that Gajarský et al.'s recent result [19] for FO carries over to order-invariant FO.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases model-checking, algorithmic meta-theorem, successor-invariant first-order logic, topological subgraphs, parameterised complexity

Digital Object Identifier 10.4230/LIPIcs.CSL.2016.18

1 Introduction

Model-checking is one of the core algorithmic problems in finite model theory: Given a sentence φ in some logic L and a finite structure A, decide whether $A \models \varphi$. The problem

25th EACSL Annual Conference on Computer Science Logic (CSL 2016).
Editors: Jean-Marc Talbot and Laurent Regnier; Article No. 18; pp. 18:1–18:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

can be generalised by allowing φ to have free variables, in which case we would like to find instances $\bar{a}$ for which $A \models \varphi[\bar{a}]$, or count the number of such instances. One important application of this is the case where φ is a database query and A the database to be queried. The logic L from which φ is drawn then serves as an abstract model of the database query language.

Commonly studied logics L include first-order logic (FO) and monadic second-order logic (MSO). Even for first-order logic the model-checking problem is PSPACE complete already when restricted to structures A with two elements. On the other hand, for every fixed FO-formula φ, checking whether $A \models \varphi$ can be done in time polynomial in the size of A. This discrepancy between the *query complexity*, i.e. the complexity depending on the size of the query φ on the one hand and the *data complexity*, i.e. the complexity depending on the size of the structure A, on the other hand suggests that the complexity of model-checking problems is best studied in the framework of *parameterised complexity* [8, 17].

In parameterised complexity, apart from the size n of the input problem (commonly the length of an appropriate binary representation of φ and A) a *parameter* k is introduced. For model-checking problems the size of the input formula is a common choice of parameter. The role of PTIME as the class of problems commonly considered to be tractable is played by the parameterised complexity class of *fixed-parameter tractable (fpt)* problems, i.e. problems which can be solved in time

$$f(k) \cdot n^c$$

for some computable function f and a constant c. Note that the constant c must not depend on k, and indeed the model-checking problem for first-order logic is unlikely to be fixed-parameter tractable.

In order to obtain tractable instances of model-checking problems, one can restrict the space of admissible input structures A, e.g. by requiring the Gaifman graph of A to possess certain graph theoretic properties such as bounded degree or planarity. A long list of results have been obtained, starting with Courcelle's famous result that model-checking for monadic second-order logic is fixed-parameter tractable on structures A with bounded tree-width [4].

Results of this form are often referred to as *algorithmic meta-theorems* because many classical problems can be rephrased as model-checking problems by formalising them as a sentence φ in a suitable logic. For example, since the existence of a Hamiltonian cycle in a graph G of bounded tree-width can be expressed by a sentence φ of monadic second-order logic, Courcelle's Theorem immediately implies that hamiltonicity can be checked in polynomial time on such graphs. Besides giving a mere proof of tractability, algorithmic meta-theorems provide a unified treatment of how structural properties can be used in algorithm design. Cf. [20] and [24] for excellent surveys of the field of algorithmic meta-theorems.

The model-checking problem for first-order logic is particularly well studied and has been shown to be fixed-parameter tractable on a large number of graph classes: Starting with Seese's result [28] for graphs of bounded degree, Frick and Grohe showed tractability on classes of graphs with bounded tree-width and, more generally, locally bounded tree-width [18], which in particular includes planar graphs. This has been generalised to graph classes with excluded minors [16] and locally excluded minors [6]. Using rather different techniques, Dvořák et al. gave a linear fpt model-checking algorithm for first-order logic on graphs of bounded expansion [9]. As a generalisation of all the graph classes mentioned so far, Grohe et al. have shown in [21] that model-checking for first-order logic is possible in near-linear fpt on all nowhere dense graph classes.

While the tractability of model-checking for first-order logic on sparse graphs is well understood, few results are available for classes of dense graphs. Recently, Gajarský et al.

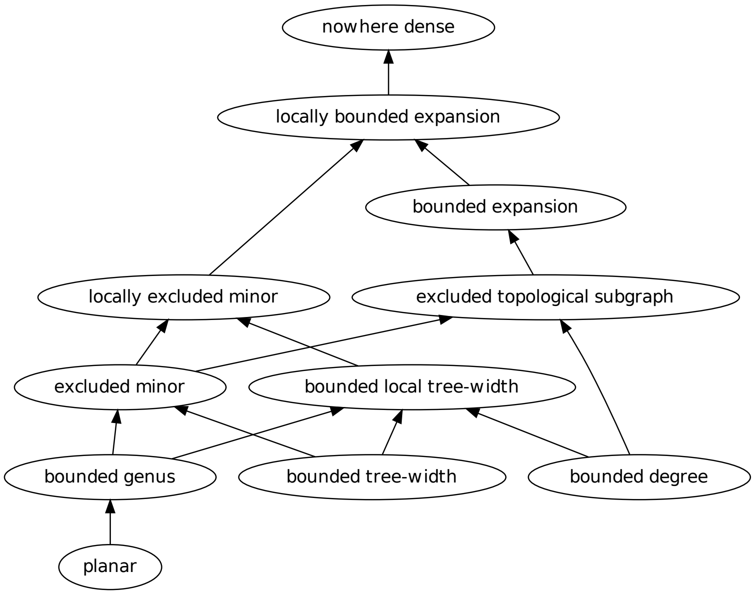

Figure 1 Sparse classes of graphs on which model-checking for first-order logic is tractable.

gave an fpt algorithm for FO model-checking on posets of bounded width, which we extend to order-invariant FO in Section 5.

Excluded Topological Subgraphs

A more general concept than that of a class of graphs excluding some graph H as a minor is that of graphs which exclude H as a *topological subgraph*. This is the concept originally used by Kuratowski in his famous result that a graph is planar if, and only if, it does not contain K_5 nor $K_{3,3}$ as a topological subgraph (cf. Section 4.4 in [7]). Recently, Grohe and Marx have extended Robertson and Seymour's graph structure theorem to classes of graphs excluding a fixed graph H as a topological subgraph [22]: These graphs can be decomposed along small separators into parts which exclude H as a minor and parts in which all but a bounded number of vertices have small degree.

Since every topological subgraph of a graph G is also a minor of G, if a class $\mathcal{C}$ of graphs excludes some graph H as a topological subgraph then it also excludes H as a minor. The converse is not true, however, since every 3-regular graph excludes K_5 as a topological subgraph, but for every $r \in \mathbb{N}$ there is a 3-regular graph containing K_r as a minor. On the other hand, graph classes with excluded topological subgraphs have bounded expansion, so model-checking for first-order logic is tractable on these classes by Dvořák et al.'s result.

Figure 1 shows an overview of sparse graph classes on which model-checking for first-order logic is tractable. Note that a class $\mathcal{C}$ of graphs excludes some finite graph H as a topological subgraph if, and only if, there is an $r \in \mathbb{N}$ such that $\mathcal{C}$ excludes the clique K_r as a topological subgraph.

Successor-Invariant Logic

We investigate the question in how far tractability results for first-order model-checking carry over to *successor-invariant first-order logic*, i.e. first-order logic enriched by a binary successor relation, restricted to formulae whose truth value does not depend on the specific choice of successor relation. Linear representations of an input structure A to a model-checking algorithm usually induce some linear order on the elements of $V(A)$, and it seems natural to make this linear order (or at least its successor relation) accessible to the query formula. This may, however, break the structural properties of the Gaifman graph of A needed by the model-checking algorithm.

Having access, even invariantly, to a successor relation provably increases the expressive power of FO on finite structures, as shown in [27]. However, all known classes of structures separating FO from order-invariant or successor-invariant FO contain large cliques, and in fact on trees [2] and on structures of bounded tree-depth [12] even order-invariant FO has the same expressive power as plain FO. On all the classes depicted in Figure 1, this question is still open, prompting for tractability results for successor-invariant or even order-invariant FO on these classes.

Previous work investigating the complexity of model-checking for successor-invariant first-order logic to that of plain first-order logic has been carried out by [14], who showed tractibility on planar graphs, and [13], who showed tractability on graph classes with excluded minors. Here we extend these results further by generalising from excluded minors to excluded topological subgraphs, further narrowing the gap between what is known for first-order logic and succesor-invariant first-order logic.

Note that for first-order logic, the result of [21] is optimal if one restricts attention to classes of graphs which are closed under taking subgraphs. In fact, Kreutzer has shown in [24] that under the complexity theoretic assumption that FPT $\neq$ W[1], if model-checking for FO on some subgraph-closed class $\mathcal{C}$ of graphs is fixed-parameter tractable, then $\mathcal{C}$ is nowhere dense (see also Section 1.4 of [9]). Examples of classes of graphs on which model-checking is fpt even for monadic second-order logic but which are not nowhere dense are graphs of bounded clique-width [5].

2 Preliminaries and Notation

For a natural number n we let $[n]$ denote the interval $\{1, \dots, n\}$.

2.1 Graphs

We will be dealing with finite simple (i.e. loop-free and without multiple edges) undirected graphs, cf. [7, 29] for an in-depth introduction. Thus a *graph* $G = (V, E)$ consists of some finite set V of *vertices* and a set $E \subseteq \binom{V}{2}$ of *edges*. We write $uv \in E$ for $\{u, v\} \in E$. For a set $U \subseteq V$ we denote the *induced subgraph* on U by $G[U]$, i.e. the graph (U, E') with

$$E' := \{uv \mid u, v \in U \text{ and } uv \in E\}.$$

For ease of notation we occasionally blur the distinction between a set U of vertices and the subgraph induced on this set. The union $G \cup H$ of two graphs $G = (V, E)$ and $H = (U, F)$ is defined as the graph $(U \cup V, E \cup F)$. For a set U of vertices, $K[U]$ denotes the complete graph (or *clique*) with vertex set U. For $k \in \mathbb{N}$, we denote the k-clique $K[[k]]$ by K_k.

A *walk* is a sequence of vertices $v_1, \dots, v_\ell \in V$, alternatively written as a function $v : [\ell] \to V$, such that $v_i v_{i+1} \in E$ for all $i = 1, \dots, \ell - 1$. A *path* is a walk in which

$v_i \neq v_j$ for $i \neq j$, except possibly $v_1 = v_\ell$, in which case the path is called a *cycle*. The vertices $v_2, \dots, v_{\ell-1}$ are called *inner vertices*. Two paths $v_1, \dots, v_\ell$ and $w_1, \dots, w_m$ are called *independent* if neither of them contains an inner vertex of the other, i.e. if $v_i = w_j$ implies $i \in \{1, \ell\}$ and $j \in \{1, m\}$.

For $k \geq 1$, a *k-walk* through a graph $G = (V, E)$ is a surjective walk $w : [\ell] \to V$ such that

$$1 \leq |\{i \in [\ell] \mid w(i) = v\}| \leq k$$

for all $v \in V$. A 1-walk is also called a *Hamiltonian path*.

Tree-Decompositions

A *tree* is a connected acyclic graph. A *tree-decomposition* of a graph $G = (V, E)$ is a pair $(\mathcal{T}, \mathcal{V})$ consisting of a tree $\mathcal{T} = (T, F)$ and a mapping $\mathcal{V} : T \to 2^V, t \mapsto \mathcal{V}_t$ such that

- $\bigcup_{t \in T} \mathcal{V}_t = V$,
- for every edge $uv \in E$ there is a $t \in T$ with $u, v \in \mathcal{V}_t$, and
- for every $v \in V$ the set $\{t \in T \mid v \in \mathcal{V}_t\}$ is a subtree of $\mathcal{T}$ (i.e. it is connected).

The sets $\mathcal{V}_t$ are called the *bags* of the tree-decomposition. Let $t \in \mathcal{T}$ have neighbours $\mathcal{N}(t) \subseteq \mathcal{T}$. The *torso* $\bar{\mathcal{V}}_t$ of $\mathcal{V}_t$ is the graph

$$G[\mathcal{V}_t] \cup \bigcup_{u \in \mathcal{N}(t)} K[\mathcal{V}_t \cap \mathcal{V}_u].$$

The graphs we will be dealing with do not in general allow tree-decompositions into bags of small size, but they do have decompositions $(\mathcal{T}, \mathcal{V})$ for which (the torsos of) all bags $\mathcal{V}_t$ have nice structural properties and for which

$$|\mathcal{V}_s \cap \mathcal{V}_t|$$

is small for all $s \neq t \in \mathcal{T}$. The *(maximal) adhesion* of $(\mathcal{T}, \mathcal{V})$ is the maximum of $|\mathcal{V}_s \cap \mathcal{V}_t|$ for all $s \neq t \in \mathcal{T}$.

Subgraphs, Minors, Topological Subgraphs

Let $G = (V, E)$ and $H = (W, F)$ be graphs. If $W \subseteq V$ and $F \subseteq E$ then we call H a *subgraph* of G and write $H \leq G$. In other words, H can be obtained from G be removing vertices and edges.

We say that H is a *minor* of G, written $H \preceq G$, if there are disjoint connected nonempty subgraphs $(B_w)_{w \in W}$ in G such that for every edge $xy \in F$ there is an edge $ab \in E$ for some $a \in B_x$ and $b \in B_y$. The sets $(B_w)_{w \in W}$ are called *branch sets* of the minor H. Equivalently, $H \preceq G$ if H can be obtained by repeatedly contracting edges in a subgraph of G.

A graph H' is a *subdivision* of a graph H if it can be obtained from H be replacing edges with paths. If $H' \leq G$ for some subdivision H' of H we say that H is a *topological subgraph* of G and write $H \preceq_{\text{top}} G$. In this case there is an injective mapping $\iota : W \to V$ and independent paths $P_{\iota(u)\iota(v)}$ connecting $\iota(u)$ to $\iota(v)$ in G for $uv \in F$. The vertices in the image of ι are called *branch vertices*. Obviously $H \preceq_{\text{top}} G$ implies $H \preceq G$, but the converse is not in general true.

2.2 Logics

We will be dealing with finite structures over finite, relational vocabularies. Thus a *vocabulary* σ is a finite set of relation symbols R, each with an associated *arity* $a(R)$, and a *σ-structure*

A consists of a finite set $V(A)$ (the *universe*) and relations $R(A) \subseteq A^{a(R)}$ for all $R \in \sigma$. For vocabularies $\sigma \subseteq \tau$ and a σ-structure A, a τ-*expansion* B is a τ-structure with $V(A) = V(B)$ and $R(B) = R(A)$ for all $R \in \sigma$.

The *Gaifman graph* of a structure A is the graph with vertex set $V(A)$ and edge set

$$\{xy \mid x \text{ and } y \text{ appear together in some relation } R(A)\}.$$

When applying graph-theoretic notions such as planarity to relational structures, we mean that the corresponding Gaifman graph has the said property.

We use standard definitions for first-order logic (FO), cf. [11, 10, 25]. In particular, $\perp$ and $\top$ denote false and true, respectively. Let σ be a vocabulary and $\mathrm{succ} \notin \sigma$ a new binary relation symbol. We set $\sigma_{\mathrm{succ}} := \sigma \cup \{\mathrm{succ}\}$ and say that succ is interpreted by a *successor relation* in a σ_{succ}-structure B if $\mathrm{succ}(B)$ is the graph of a cyclic permutation on $V(B)$. An $\mathrm{FO}[\sigma_{\mathrm{succ}}]$-formula φ is called *successor-invariant* if for all σ-structures A and all σ_{succ}-expansions B, B' of A in which succ is interpreted by a successor relation we have

$$B \models \varphi \quad \Leftrightarrow \quad B' \models \varphi,$$

when all free variables of φ are interpreted identically in B and B'. In this case we say that $A \models \varphi$ if $B \models \varphi$ for one such expansion B (equivalently for all such expansions).

Note that another common definition of successor relation is to require $\mathrm{succ}(A)$ to be of the form

$$\{(a_1, a_2), (a_2, a_3), \dots, (a_{n-1}, a_n)\}$$

for some enumeration $V(A) := \{a_1, \dots, a_n\}$ of the elements of $V(A)$. This differs from our definition in that we require $(a_n, a_1) \in \mathrm{succ}(A)$ as well, eliminating the somewhat artificial status of the first and last element. This does not affect the expressive power of successor-invariant FO, because a cyclic successor relation can be obtained from a linear one using a simple FO interpretation and vice versa. Note that the quantifier rank of formulas is slightly increased by this interpretation.

Order-invariant first-order logic is defined analogously to successor-invariant FO, by allowing the use of a binary relation $\leq$ which is interpreted as a linear order and demanding the truth value of a formula to be independent of the chosen linear order.

3 Model-Checking for Successor-Invariant First-Order Logic

The main result of this paper is the following:

▶ **Theorem 1.** *There is an algorithm $\mathbb{A}$ which takes as input*

- *a finite graph H,*
- *a finite σ-structure A over some relational vocabulary σ, such that the Gaifman graph of A does not contain H as a topological subgraph, and*
- *a successor-invariant formula $\varphi \in FO[\sigma_{\mathrm{succ}}]$*

and checks whether

$$A \models \varphi$$

in time $f(|V(H)| + |\varphi|) \cdot |V(A)|^c$ for some computable function f and $c \in \mathbb{N}$, both depending only on $\mathbb{A}$.

Note that model-checking for first-order logic on nowhere dense classes of graphs is possible in time $f(|\varphi|) \cdot |V(A)|^{1+\epsilon}$ for arbitrarily small $\epsilon > 0$ by a result of Grohe et al. [21]. Even though a representation of a structure A in computer memory is likely to induce a linear order on the elements of $V(A)$, making this linear order or its successor relation accessible to the formula φ potentially complicates the model-checking problem. In particular, adding the cycle corresponding to this linear order (or any other cycle through the whole graph) to A may introduce new shallow minors.

The proof of Theorem 1 is based on the following two lemmas:

▶ **Lemma 2.** *For every finite graph H there are constants $k \in \mathbb{N}$ and $c \in \mathbb{N}$ such that for every graph G which does not contain H as a topological subgraph there is a graph G' and a k-walk $w : [\ell] \to V(G')$ through G' such that G' is obtained from G by only adding edges and G' does not contain K_c as a topological subgraph. Furthermore, k, c, G' and w can be computed, given G and H, in time $f(|V(H)|) \cdot |V(G)|^d$ for some computable function f and $d \in \mathbb{N}$.*

▶ **Lemma 3.** *Let σ be a finite relational vocabulary, A a finite σ-structure, and $w : [\ell] \to V(A)$ a k-walk through the Gaifman graph of A.*

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

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

Lemma 3 is taken from [13, Lemma 4.4] and has been proved there. We will prove Lemma 2 in Section 4. The proof of Theorem 1 then is a combination of the above lemmas:

Proof of Theorem 1. Given a σ-structure A, a successor-invariant σ_{succ}-formula φ and a graph H which is not a topological subgraph of the Gaifman graph of A, we first compute the Gaifman graph G of A. Using the algorithm of Lemma 2 we then compute a k-walk $w : [\ell] \to V(A)$ through a supergraph G' of G which excludes some clique $K_{c'}$ as a topological subgraph.

Let E be a binary relation symbol. We expand A to a $(\sigma \cup \{E\})$-structure A' by setting

$$E(A') := \{(w(i), w(i+1)) \mid i \in [\ell - 1]\} \cup \{(w(\ell), w(1))\}.$$

Then G' is the Gaifman graph of A', which by Lemma 2 excludes K_c as a topological subgraph.

Using Lemma 3 we compute, for a suitable $\tau \supseteq \sigma$, a τ-expansion A'' of A' and an FO$[\tau]$-formula $\varphi^{(k)}_{\text{succ}}(x, y)$ which defines a successor relation on A''. We replace all atomic subformulae $\text{succ}\, xy$ in φ by $\varphi^{(k)}_{\text{succ}}(x, y)$, obtaining an FO$[\tau]$-formula $\tilde{\varphi}$ such that

$$A'' \models \tilde{\varphi} \quad \Leftrightarrow \quad (A, S) \models \varphi$$

where S the successor relation defined by $\varphi^{(k)}_{\text{succ}}$. Note $\varphi^{(k)}_{\text{succ}}$ and τ depend only on k, which in turn only depends on H.

Since the Gaifman graph G'' of A'' excludes H as a topological subgraph, there is a class $\mathcal{C}$ of graphs of bounded expansion such that $G'' \in \mathcal{C}$. We can therefore use Dvořák et al.'s model-checking algorithm [9] for FO on $\mathcal{C}$ to check whether

$$A'' \models \tilde{\varphi}$$

in time linear in $|A|$. ◀

4 k-walks in Graphs with Excluded Topological Subgraphs

In this section we will prove Lemma 2. Given a graph G which excludes a graph H as a topological subgraph, as a first step towards constructing a supergraph G' with a k-walk we compute a tree-decomposition of G into graphs which exclude H as a minor and graphs of almost bounded degree:

▶ **Theorem 4** (Theorem 4.1 in [22]). *For every $k \in \mathbb{N}$ there exists a constant $c = c(k) \in \mathbb{N}$ such that the following holds: If H is a graph on k vertices and G a graph which does not contain H as a topological subgraph, then there is a tree-decomposition $(\mathcal{T}, \mathcal{V})$ of G of adhesion at most c such that for all $t \in \mathcal{T}$*

- *$\bar{\mathcal{V}}_t$ has at most c vertices of degree larger than c,* or
- *$\bar{\mathcal{V}}_t$ excludes K_c as a minor.*

Furthermore, there is an algorithm that, given graphs G of size n and H of size k computes such a decomposition in time $f(k) \cdot n^{O(1)}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

For the rest of this section we assume a graph $G = (V, E)$ together with a tree-decomposition $(\mathcal{T}, \mathcal{V})$ satisfying the properties of Theorem 4 as given. We will construct k-walks through each of the bags of this decomposition, for a suitable k depending only on H, suitably adding edges within the bags in a way that will not create large topological subgraphs. We will then connect these k-walks to obtain a k'-walk through all of G, carefully adding further edges where necessary.

If $s, t \in \mathcal{T}$ are neighbours in $\mathcal{T}$ we will connect the k-walk through $\mathcal{V}_s$ and the k-walk through $\mathcal{V}_t$ by joining them along a suitably chosen vertex $v \in \mathcal{V}_s \cap \mathcal{V}_t$. Since the resulting walk may visit v a total of $k + 1$ times, we must be careful not to select the same vertex v more than a bounded number of times.

We first pick an arbitrary tree node $r \in \mathcal{T}$ as the root of the tree-decomposition. Notions such as parent and sibling nodes are meant with respect to this root node r. For a node $t \in \mathcal{T}$ we define its adhesion $\alpha_t \subseteq \mathcal{V}_t$ as

$$\alpha_t := \begin{cases} \emptyset & \text{if } t = r \\ \mathcal{V}_s \cap \mathcal{V}_t & \text{if } s \text{ is the parent of } t. \end{cases}$$

By adding the necessary edges within the bags we may assume that each $\mathcal{V}_t$ is identical to its torso, in other words we may assume that $G[\alpha_t]$ is a clique for each $t \in \mathcal{T}$.

4.1 Computing the k-walks w_t

Let $s, t \in \mathcal{T}$ be nodes such that s is the parent of t. It may happen that $\alpha_s \cap \alpha_t \neq \emptyset$, and in fact we can not bound

$$|\{s \in \mathcal{T} \mid v \in \mathcal{V}_s\}|$$

for all G excluding a fixed topological subgraph and all $v \in V(G)$. Since we are only allowed to visit each vertex a bounded (for a fixed excluded topological subgraph) number of times, we first compute, for $t \in \mathcal{T}$, a k-walk w_t through a suitable supergraph of $\mathcal{V}_t \setminus \alpha_t$.

If $\bar{\mathcal{V}}_t$ contains only c vertices of degree larger than c we choose an arbitrary enumeration $v_1, \dots, v_\ell$ of $\mathcal{V}_t \setminus \alpha_t$ and add edges

$$v_1v_2, v_2v_3, \dots, v_{\ell-1}v_\ell, v_\ell v_1$$

to G as far as they are not already present. This will increase the degree of each vertex by at most 2, so there are still at most c vertices of degree larger than $c+2$. We set

$$\begin{aligned} w_t : [\ell] &\to \mathcal{V}_t \\ i &\mapsto v_i \end{aligned}$$

for these bags.

If, on the other hand, $\bar{\mathcal{V}}_t$ exludes a clique K_c as a minor, we invoke the following lemma on the graph $\mathcal{V}_t \setminus \alpha_t$:

▶ **Lemma 5** (Lemma 3.3 in [13]). *For every natural number c there are $k, c' \in \mathbb{N}$ such that: If $G = (V, E)$ is a graph which does not contain a K_c-minor, then there is a supergraph $G' = (V, E')$ obtained from G by possibly adding edges such that G' does not contain a $K_{c'}$-minor and there is a k-walk w through G'. Moreover, G' and w can be found in polynomial time for fixed c.*

Since we ignore the vertices in α_t when computing the k-walk w_t, it may happen that the resulting supergraph of $\bar{\mathcal{V}}_t$ *does* contain a $K_{c'}$-minor. However, the largest possible clique minor is still of bounded size, because $|\alpha_t| \leq c$:

▶ **Lemma 6.** *Let $G = (V, E)$ be a graph such that $K_{c'} \not\preceq G$, and let $G \oplus K_c$ be the graph with vertex set $V' = V \cup [c]$ and edge set*

$$E' = E \cup \binom{[c]}{2} \cup \{va \mid v \in V, a \in [c]\}.$$

In other words, $G \oplus K_c$ is the disjoint sum of G and K_c plus edges between all vertices of G and all vertices of K_c. Then $K_{c+c'} \not\preceq G \oplus K_c$.

Proof. Otherwise let $X_1, \ldots, X_{c+c'}$ be the branch sets of a $K_{c+c'}$-minor in $G \oplus K_c$. At most c of the sets contain vertices of the added K_c-clique. The remaining sets form the branch sets of a K_c-minor in G, contradicting the assumption that $K_c \not\preceq G$. ◀

4.2 Connecting the k-walks

We still need to connect the k-walks through the individual bags of $(\mathcal{T}, \mathcal{V})$ to obtain a single k'-walk through the whole graph, for some k' to be determined below. This is the most complicated part of our construction, since we must guarantee that no vertex is visited more than k' times by the resulting walk, and that no large topological clique subgraphs are created.

In the case of graphs excluding some fixed minor, the Graph Structure Theorem guarantees the existence of a tree-decomposition into nearly embeddable graphs such that neighbouring bags intersect only in apices and vertices lying on some face or vortex of their near embeddings, and this was used in [13] to select vertices from the adhesion sets of bags in a suitable way. Since the decomposition theorem for graphs excluding a topological minor does not provide this kind of information, we need a different approach here. Instead, our method for selecting vertices along which to connect the k-walks relies on the fact that sparse graphs are *degenerate*, i.e. every subgraph of a sparse graph contains some vertex of small degree.

In connecting the walks w_t, we will proceed down the tree $\mathcal{T}$. At any point in the process we keep a set $D \subseteq \mathcal{T}$ and a walk w such that

- D is a connected subset of $\mathcal{T}$,
- the k'-walk has been constructed in $\bigcup_{t \in D} \mathcal{V}_t$,

- if $s \in D$ and s' is a sibling of s then also $s' \in D$,
- w is a k'-walk through $\bigcup_{t \in D} \mathcal{V}_t$, and if $s \in D$ has a child $t \notin D$, then the vertices in $\mathcal{V}_s \setminus \alpha_s$ are visited at most $k+1$ times by w.

We start with $D = \{r\}$ and $w = w_r$, where r is the root of $\mathcal{T}$. This is easily seen to satisfy all of the above conditions.

Now let $s \in D$ be a node whose children $t_1, \ldots, t_n$ are not in D. We let

$$C_i := \alpha_{t_i} \setminus \alpha_s$$

be the adhesion set of t_i with all vertices of the adhesion set of s removed. If $C_i = \emptyset$ then t_i can be made a sibling of s (rather than a child), so we assume that all C_i are nonempty. Since the properties of $(\mathcal{T}, \mathcal{V})$ are guaranteed for the torsos of the bags we may assume that $G[C_i]$ is a clique for each i and that w visits the vertices of $\bigcup C_i$ at most $k+1$ times.

It may happen that $C_i = C_j$ for some $i \neq j$. To deal with this, assume that

$$C_1 = C_2 = \cdots = C_m \neq C_i \text{ for i > m.}$$

For each $i = 1, \ldots, m$ we choose an edge $u_i v_i \in E(\mathcal{V}_{t_i})$ which is traversed by the walk w_{t_i} in the direction from u_i to v_i at some point. We add edges

$$u_i v_{i+1} \text{ for } i = 1, \ldots, m-1 \text{ and } u_m v_1$$

and connect the walks $w_{t_1}, \ldots, w_{t_m}$ along these edges. Because w_{t_i} is a walk through $\mathcal{V}_{t_i} \setminus \alpha_{t_i}$, we have

$$u_i, v_i \in \mathcal{V}_{t_j} \quad \Leftrightarrow \quad i = j$$

for $i, j = 1, \ldots, m$. To accomodate for the extra edges, we add the vertices u_i and v_i to $\mathcal{V}_s$, and therefore to α_{t_i} and C_i. Since these vertices together with the added edges form an isolated cycle

$$u_1 v_1 u_2 v_2 \ldots u_m v_m u_1$$

in $\mathcal{V}_s$, no new topological subgraphs are created by this. The maximal adhesion of $(\mathcal{T}, \mathcal{V})$ is still bounded by $c+2$.

Therefore we now assume that the cliques $C_1, \ldots, C_n$ are all distinct. It remains to find a function

$$f : [n] \to V$$

such that

- $f(i) \in C_i$ for all i, and
- $\left|f^{-1}(v)\right| \leq M$ for all $v \in V$ and some constant M depending only on H.

We define the function f iteratively on larger subsets of $[n]$ as follows: Let $\tilde{G}$ be the subgraph of G induced on the union of all C_i:

$$\tilde{G} = G\left[\bigcup_i C_i\right].$$

We show that $\tilde{G}$ contains a vertex of degree (in $\tilde{G}$) at most d, for some constant d depending only on the constant c from Theorem 4 (and therefore only on the excluded topological subgraph H we started with). If $\mathcal{V}_s$ contains only c vertices of degree larger than c then

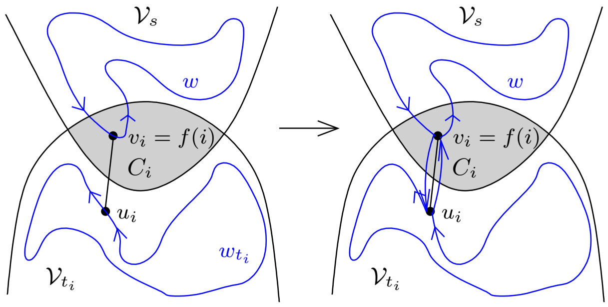

Figure 2 Connecting the individual k-walks.

this is true with $d = c$. If $\mathcal{V}_t$ excludes some clique K_c as a minor we use the fact that these graphs are d-degenerate for some d depending only on c. In fact, by Theorem 7.2.1 in [7] there is a constant d such that if the average degree of $\tilde{G}$ is at least d, then $K_c \preceq_{\text{top}} \tilde{G}$ and therefore $K_c \preceq \tilde{G}$.

In both cases there is a $v \in \bigcup_i C_i$ which has degree at most d in $\tilde{G}$. We want to bound the number of $i \in [n]$ for which $v \in C_i$. Since every clique C_i has size at most $c + 2$, and if $v \in C_i$ then all elements of $C \setminus \{v\}$ are neighbours of v, there can be at most

$$M := \binom{d}{0} + \binom{d}{1} + \cdots + \binom{d}{c+1}$$

many such C_i, and this bound only depends on c. It is therefore safe to define

$$f(i) := v \text{ for all } i \in [n] \text{ such that } v \in C_i.$$

We remove these cliques and iterate until no cliques remain.

Once the function f has been found we connect the walk w through $\bigcup_{t \in D} \mathcal{V}_t$ with the walks w_{t_i} through the bags V_{t_i}. Let $w : [\ell] \to V$ be the walk constructed so far. For each $i \in [n]$ let $v_i = f(i) \in C_i$ be the vertex chosen by f, and let $u_i \in \mathcal{V}_{t_i} \setminus \alpha_{t_i}$ be a neighbour of v_i. If no such neighbour exists it is safe to create one by adding an edge between v_i and an arbitrary vertex of $\mathcal{V}_{t_i} \setminus \alpha_{t_i}$. We now extend the walk w by inserting the k-walk w_{t_i} along the edge $v_i u_i$ when v_i is first visited by w. This increases the number of times v_i and u_i are visited by one each (cf. Figure 2).

After inserting all walks $w_{t_1}, \ldots, w_{t_n}$ we set

$$D := D \cup \{t_1, \ldots, t_n\}$$

and repeat the process until $D = \mathcal{T}$. Note that the resulting walk is a $(k + M + 1)$-walk through the supergraph G' of G obtained by adding edges to G.

4.3 Topological Subgraphs in G'

By now we have a supergraph G' of G, obtained by only adding edges, and a $k' = (k+M+1)$-walk $w : [\ell] \to V(G')$ through this supergraph. Furthermore, there is a $c' = c'(H)$ depending only on (the size of) H and a tree-decomposition $(\mathcal{T}, \mathcal{V})$ of G' such that if $s, t \in \mathcal{T}$ then $|\mathcal{V}_s \cap \mathcal{V}_t| \le c'$ and for all $t \in \mathcal{T}$

- $\bar{\mathcal{V}}_t$ has at most c' vertices of degree larger than c' or
- $\bar{\mathcal{V}}_t$ excludes $K_{c'}$ as a minor.

We show that this implies $K_{c'+2} \not\preceq_{\text{top}} G'$: Assume for a contradiction that $K_{c'+2} \preceq_{\text{top}} G$, and let $v_1, \dots, v_{c'+2}$ be the branch vertices of a $K_{c'+2}$-subdivision in G. Then there is a $t \in \mathcal{T}$ such that $\{v_1, \dots, v_{c'+2}\} \subseteq \mathcal{V}_t$: Otherwise choose $i < j$ and $t \neq t'$ so that

$$v_i \in \mathcal{V}_t \setminus \mathcal{V}_{t'} \quad \text{and} \quad v_j \in \mathcal{V}_{t'} \setminus \mathcal{V}_t.$$

Then, since the adhesion of $(\mathcal{T}, \mathcal{V})$ is at most c', there is a set $S \subseteq V$ of size at most c' separating two branch vertices, which is not possible in a $(c'+2)$-clique.

Now let $t \in \mathcal{T}$ be a tree node for which $\mathcal{V}_t$ contains all branch vertices. For $i < j$, let P_{ij} be the path in G connecting v_i and v_j. If all vertices on this path are in $\mathcal{V}_t$ we are done. Otherwise we may shorten this path to get a path P'_{ij} connecting v_i and v_j in the torso of $\mathcal{V}_t$. Thus

$$K_{c'+2} \preceq_{\text{top}} \mathcal{V}_t.$$

But none of the bags $\mathcal{V}_t$ can contain $K_{c'+2}$ as a topological subgraph: Since $K_{c'+2} \preceq_{\text{top}} \mathcal{V}_t$ implies $K_{c'+2} \preceq \mathcal{V}_t$ which in turn implies $K_{c'} \preceq \mathcal{V}_t$, none of the bags excluding $K_{c'}$ as a minor can contain $K_{c'+2}$ as a topological subgraph. But if $K_{c'+2} \preceq_{\text{top}} \mathcal{V}_t$ then there must be at least $c'+2$ vertices of degree at least $c'+1$, namely the branch vertices of the image of a subdivision of $K_{c'+2}$. We conclude that $K_{c'+2} \not\preceq_{\text{top}} G'$.

5 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. [26]), model-checking even for monadic second-order logic has been shown to be fpt by Courcelle et al. [5].
- More recently, model-checking on coloured posets of bounded width has been shown to be in fpt for existential FO by Bova et al. [3] and for all of FO by Gajarský et al. [19].

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 by Engelmann et al. in [14, Thm. 4.2]. For posets of bounded width we give a proof here. We first review the necessary definitions:

▶ **Definition 7.** A *partially ordered set (poset)* $(P, \leq^P)$ is a set P with a reflexive, transitive and antisymmetric binary relation $\leq^P$. A *chain* $C \subseteq P$ is a totally ordered subset, i.e. for all $x, y \in C$ one of $x \leq^P y$ and $y \leq^P x$ holds. An *antichain* is a set $A \subseteq P$ such that if $x \leq^P y$ for $x, y \in A$ then $x = y$. The *width* of $(P, \leq^P)$ is the maximal size $|A|$ of an antichain $A \subseteq P$. A *coloured* poset is a poset $(P, \leq^P)$ together with a function $\lambda : P \to \Lambda$ mapping P to some set Λ of *colours*. By $\|P\|$ we denote the length of a suitable encoding of $(P, \leq^P)$.

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

A proof can be found in [7, Sec. 2.5]. Moreover, by a result of Felsner et al. [15], both the width w and a set of chains $C_1, \dots, C_w$ covering P can be computed from $(P, \leq_P)$ in time $O(w \cdot \|P\|)$.

With this, we are ready to prove the following:

▶ **Theorem 9.** *There is an algorithm which, on input a coloured poset* $(P, \leq^P)$ *with colouring* $\lambda : P \to \Lambda$ *and an order-invariant first-order formula* φ*, checks whether* $P \models \varphi$ *in time* $f(w, |\varphi|) \cdot \|P\|^2$ *where* w *is the width of* $(P, \leq^P)$*.*

Proof. Using the algorithm of [15], we compute a chain cover $C_1, \ldots, C_w$ of $(P, \leq^P)$. 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

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

Then

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

defines a linear order on $(P, \leq_P)$ with colouring λ'. After substituting $\varphi_{\leq}$ for $\leq$ in φ we may apply Gajarský et al.'s algorithm [19] to check whether $P \models \varphi$. ◀

6 Conclusion and Further Research

We have shown that model-checking for successor-invariant first-order logic is fixed-parameter tractable on classes of graphs excluding some fixed graph H as a topological subgraph. This extends previous results showing tractibility on planar graphs [14] and graphs with excluded minors [13]. For dense graphs, we showed how the recent model-checking algorithm by Gajarský et al. [19] can be adapted to order-invariant FO.

This prompts for further generalisation in two ways: First, can we close the gap between plain first-order logic and its successor-invariant counterpart? Next steps could be graph classes with bounded expansion or with locally excluded minors. However, no structure theorem comparable to those of Robertson and Seymour and of Grohe and Marx are known for these graph classes.

Another interesting open question is whether model-checking for order-invariant first-order logic is tractable on any of the classes depicted in Figure 1. Since the Gaifman graph of a linearly ordered structure is a clique, there is no hope of finding a "good" linear order which can be added to the input structure without destroying the desirable properties of its Gaifman graph. As shown in [23], order-invariant first-order logic has a Gaifman-style locality property (see also [1]). It is, however, not at all clear how this could be turned into an efficient model-checking algorithm. In particular, no variant of Gaifman normal form is known for this logic.

References

1 Matthew Anderson, Dieter van Melkebeek, Nicole Schweikardt, and Luc Segoufin. Locality from circuit lower bounds. *SIAM J. Comput.*, 41(6):1481–1523, 2012.

2 Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symb. Log.*, 74(1):168–186, 2009.

3 Simone Bova, Robert Ganian, and Stefan Szeider. Model checking existential logic on partially ordered sets. *ACM Transactions on Computational Logic (TOCL)*, 17(2):10, 2015.

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

5 Bruno Courcelle, Johann Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

6 Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *Logic in Computer Science (LICS)*, pages 270–279, 2007.

7 Reinhard Diestel. *Graph Theory.* Number 173 in GTM. Springer, 4th edition, 2012.

8 Rod Downey and Michael R. Fellows. *Parameterized Complexity.* Springer, 1998.

9 Zdeněk Dvořák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *Journal of the ACM (JACM)*, 60(5):36, 2013.

10 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory.* Perspectives in Mathematical Logic. Springer, 2nd edition, 1999.

11 Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logic.* Springer, 2nd edition, 1994.

12 Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *Mathematical Foundations of Computer Science 2014 – 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, pages 256–266, 2014.

13 Kord Eickmeyer, Ken-Ichi Kawarabayashi, and Stephan Kreutzer. Model checking for successor-invariant first-order logic on minor-closed graph classes. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '13, pages 134–142. IEEE Computer Society, 2013.

14 Viktor Engelmann, Stephan Kreutzer, and Sebastian Siebertz. First-order and monadic second-order model-checking on ordered structures. In *Logics in Computer Science*, pages 275–284, 2012.

15 Stefan Felsner, Vijay Raghavan, and Jeremy Spinrad. Recognition algorithms for orders of small width and graphs of small dilworth number. *Order*, 20(4):351–364, 2003.

16 Jörg Flum and Martin Grohe. Fixed-parameter tractability, definability, and model-checking. *SIAM J. Comput.*, 31(1):113–145, 2001.

17 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Springer, 2006. ISBN 3-54-029952-1.

18 Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM*, 48:1148–1206, 2001.

19 Jakub Gajarský, Petr Hliněný, Daniel Lokshtanov, Jan Obdržálek, Sebastian Ordyniak, MS Ramanujan, and Saket Saurabh. Fo model checking on posets of bounded width. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 963–974. IEEE, 2015.

20 Martin Grohe. Logic, graphs, and algorithms. In E. Grädel, T. Wilke, and J. Flum, editors, *Logic and Automata – History and Perspectives.* Amsterdam University Press, 2007.

21 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 89–98. ACM, 2014.

22 Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM Journal on Computing*, 44(1):114–159, 2015.

23 Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Trans. Comput. Logic*, 1(1):112–130, July 2000.

24 Stephan Kreutzer. Algorithmic meta-theorems. In Javier Esparza, Christian Michaux, and Charles Steinhorn, editors, *Finite and Algorithmic Model Theory*, London Mathematical Society Lecture Note Series, chapter 5, pages 177–270. Cambridge University Press, 2011.

a preliminary version is available at Electronic Colloquium on Computational Complexity (ECCC), TR09-147, `http://www.eccc.uni-trier.de/report/2009/147`.

25 Leonid Libkin. *Elements of Finite Model Theory.* Texts in Theoretical Computer Science. Spinger-Verlag, 2004.

26 Sang-Il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96:514–528, 2006.

27 Benjamin Rossman. Successor-invariant first-order logic on finite structures. *J. Symb. Log.*, 72(2):601–618, 2007.

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

29 William T. Tutte. *Graph Theory*, volume 21 of *Encyclopedia of Mathematics and Its Applications.* Cambridge University Press, 2001.