Fine-grained Meta-Theorems for Vertex Integrity

Michael Lampis
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France
michail.lampis@lamsade.dauphine.fr

Valia Mitsou
Université de Paris, IRIF, CNRS, 75205, Paris, France
vmitsou@irif.fr

Abstract

Vertex Integrity is a graph measure which sits squarely between two more well-studied notions, namely vertex cover and tree-depth, and that has recently gained attention as a structural graph parameter. In this paper we investigate the algorithmic trade-offs involved with this parameter from the point of view of algorithmic meta-theorems for First-Order (FO) and Monadic Second Order (MSO) logic. Our positive results are the following: (i) given a graph G of vertex integrity k and an FO formula ϕ with q quantifiers, deciding if G satisfies ϕ can be done in time $2^{O(k^2q+q\log q)}+n^{O(1)}$; (ii) for MSO formulas with q quantifiers, the same can be done in time $2^{2^{O(k^2+kq)}}+n^{O(1)}$. Both results are obtained using kernelization arguments, which pre-process the input to sizes $2^{O(k^2)}q$ and $2^{O(k^2+kq)}$ respectively.

The complexities of our meta-theorems are significantly better than the corresponding meta-theorems for tree-depth, which involve towers of exponentials. However, they are worse than the roughly $2^{O(kq)}$ and $2^{2^{O(k+q)}}$ complexities known for corresponding meta-theorems for vertex cover. To explain this deterioration we present two formula constructions which lead to fine-grained complexity lower bounds and establish that the dependence of our meta-theorems on k is best possible. More precisely, we show that it is not possible to decide FO formulas with q quantifiers in time $2^{o(k^2q)}$, and that there exists a constant-size MSO formula which cannot be decided in time $2^{2^{o(k^2)}}$, both under the ETH. Hence, the quadratic blow-up in the dependence on k is unavoidable and vertex integrity has a complexity for FO and MSO logic which is truly intermediate between vertex cover and tree-depth.

2012 ACM Subject Classification Theory of Computation → Design and Analysis of Algorithms → Parameterized Complexity and Exact Algorithms

Keywords and phrases Model-Checking, Fine-grained complexity, Vertex Integrity

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding *Michael Lampis*: Partially supported by ANR JCJC project "ASSK" (ANR-18-CE40-0025-01)

1 Introduction

An algorithmic meta-theorem is a general statement proving that a large class of problems is tractable. Such results are of great importance because they allow one to quickly classify the complexity of a new problem, before endeavoring to design a fine-tuned algorithm. In the domain of parameterized complexity theory for graph problems, possibly the most well-studied type of meta-theorems are those where the class of problems in question is defined using a language of formal logic, typically a variant of First-Order (FO) or Monadic Second-Order (MSO) logic, which are the logics that allow quantification over vertices or sets of vertices respectively[1]. In this area, the most celebrated result is Courcelle's theorem

[1] Note that the version of MSO logic we use in this paper is sometimes also referred to as MSO_1 to distinguish from the version that also allows quantification over sets of edges.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[6], which states that all properties expressible in MSO logic are solvable in linear time, parameterized by treewidth and the size of the MSO formula. In the thirty years since the appearance of this fundamental result, numerous other meta-theorems in this spirit have followed (we give an overview of some such results below).

Despite its great success, Courcelle's theorem suffers from one significant weakness: the algorithm it guarantees for deciding an MSO formula ϕ on a graph G with n vertices and treewidth k has running time $f(k, \phi) \cdot n$, where f is, in the worst case, a tower of exponentials whose height can only be bounded as a function of ϕ. Unfortunately, it has been known since the work of Frick and Grohe [20] that this terrible parameter dependence cannot be avoided, even if one only considers FO logic on trees (or MSO logic on paths [40]). This has motivated the study of the complexity of FO and MSO logic with parameters which are more restrictive than treewidth. In the context of such parameters, fixed-parameter tractability for all MSO-expressible problems is already given by Courcelle's theorem, so the goal is to obtain more "fine-grained" meta-theorems which achieve a better dependence on ϕ and k.

The two results from this line of research which are most relevant to our paper are the meta-theorems for vertex cover given in [39], and the meta-theorem for tree-depth given by Gajarský and Hliněný [21]. Regarding vertex cover, it was shown in [39] that FO and MSO formulas with q quantifiers can be decided on graphs with vertex cover k in time roughly $2^{O(kq+q\log q)}$ and $2^{2^{O(k+q)}}$ respectively. Both of these results were shown to be tight, in the sense that improving their dependence on k would violate the Exponential Time Hypothesis (ETH). For tree-depth, it was shown in [21] that FO and MSO formulas with q quantifiers can be decided on graphs with tree-depth k with a complexity that is roughly k-fold exponential. Hence, for fixed k, the complexity we obtain is elementary, but the height of the tower of exponentials increases with k, and this cannot be avoided under the ETH [40].

Vertex cover and tree-depth are among the most well-studied measures in parameterized complexity. In all graphs G we have $\mathrm{vc}(G)+1 \geq \mathrm{td}(G) \geq \mathrm{pw}(G) \geq \mathrm{tw}(G)$, so these parameters form a natural hierarchy with pathwidth and treewidth, with vertex cover being the most restrictive. As explained above, the distance between the performance of meta-theorems for vertex cover (which are double-exponential for MSO) and for tree-depth (which give a tower of exponentials of height td) is huge, but conceptually this is perhaps not surprising. Indeed, one could argue that the structural distance between graphs of vertex cover k from the class of graphs of tree-depth k is also huge. As a reminder, a graph has vertex cover k if we can delete k vertices to obtain an independent set; while a graph has tree-depth k if there exists $k' \leq k$ such that we can delete k' vertices to obtain a disjoint union of graphs of tree-depth $k - k'$. Clearly, the latter (inductive) definition is more powerful and covers vastly more graphs, so it is natural that model-checking should be significantly harder for tree-depth.

The landscape of parameters described above indicates that there should be space to investigate interesting structural parameters *between* vertex cover and tree-depth, exactly because the distance between these two is large in terms of generality and complexity. One notion that has recently attracted attention in this area is *Vertex Integrity* [11], denoted as $\iota(G)$. A graph has vertex integrity k if there exists $k' \leq k$ such that we can delete k' vertices and obtain a disjoint union of graphs of *size* at most $k - k'$. Hence, the definition of vertex integrity is the same as for tree-depth, except that we replace the inductive step by simply bounding the size of the components that result after deleting a separator of the graph. This produces a notion that is more restrictive than tree-depth, but still significantly more general than vertex cover (where the resulting components must be singletons). In all graphs G, we have $\mathrm{vc}(G) + 1 \geq \iota(G) \geq \mathrm{td}(G)$, so it becomes an interesting question to investigate the complexity trade-off associated with these parameters, that is, how the complexity of various problems deteriorates as we move from vertex cover, to vertex integrity, to tree-depth. This type of study was recently undertaken systematically for many problems by Gima et al. [29]. In this paper we make an investigation in the same direction from the lens of algorithmic meta-theorems.

Our results We consider the problem of verifying whether a graph G satisfies a property given by an FO or MSO formula with q quantifiers, assuming $\iota(G) \leq k$. Our goal is to give a fine-grained determination of the complexity of this problem as a function of k. We obtain the following two positive results:

1. FO formulas with q quantifiers can be decided in time $2^{O(k^2 q + q\log q)} + n^{O(1)}$.
2. MSO formulas with q vertex and set quantifiers can be decided in time $2^{2^{O(k^2+kq)}} + n^{O(1)}$.

Hence, we obtain meta-theorems stating that any problem that can be expressed in FO or MSO logic can be solved in the aforementioned times. Both of these results are obtained through a kernelization argument, similar in spirit to the arguments used in the meta-theorems of [21, 39]. To describe the main idea, recall that if $\iota(G) \leq k$, then there exists a separator S of size at most k, such that removing it will disconnect the graph into components of size at most k. The key now is that these components can be partitioned into 2^{k^2} equivalence *types*, where components of the same type are isomorphic. We then argue that if we have a large number of isomorphic components, it is always safe to delete any one of them from the graph, as this does not change whether the given formula holds (Lemmas 12 and 14). We then complete the argument by applying the standard brute-force algorithms for FO and MSO logic on the kernels.

We complement the results above by showing that the approach of kernelizing and then executing the brute-force algorithm is best possible. More precisely, we show that, under the ETH, it is not possible to obtain a model-checking algorithm for FO logic running in time $2^{o(k^2 q)} n^{O(1)}$; while for MSO we construct a constant-sized formula which cannot be model-checked in time $2^{2^{o(k^2)}}$. Hence, the quadratic dependence on k, which distinguishes our meta-theorems from the corresponding meta-theorems for vertex cover, cannot be avoided.

Related work The study of structural parameters which trade off the generality of treewidth for improved algorithmic properties is by now a standard topic in parameterized complexity. The most common type of work here is to consider a problem that is intractable parameterized by treewidth and see whether it becomes tractable parameterized by vertex cover or tree-depth [2, 10, 13, 16, 17, 31, 32, 35, 34, 36, 42, 41]. See [1] for a survey of results of this type. In this context, vertex integrity has only recently started being studied as an intermediate parameter between vertex cover and tree-depth, and it has been discovered that fixed-parameter tractability for several problems which are W-hard by tree-depth can be extended from vertex cover to vertex integrity [4, 12, 25, 27, 29]. Note that some works use a measure called *core fracture* number, which is an equivalent notion to vertex integrity.

Algorithmic meta-theorems are a well-studied topic in parameterized complexity (see [30] for a survey). Courcelle's theorem has been extended to the more general notion of clique-width [7], and more efficient versions of these meta-theorems have been given for the more restricted parameters twin-cover [22], shrub-depth [24, 23], neighborhood diversity and max-leaf number [39]. Meta-theorems have also been given for even more general graph parameters, such as [5, 14, 19, 18], and for logics other than FO and MSO, with the goal of either targeting a wider class of problems [26, 37, 38, 44], or achieving better complexity [43]. Meta-theorems have also been given in the context of kernelization [3, 15, 28] and approximation [9]. To the best of our knowledge, the complexity of FO and MSO model checking parameterized by vertex integrity has not been explicitly studied before, but since vertex integrity is a restriction of tree-depth and a generalization of vertex cover, the algorithms of [21] and the lower bounds of [39] apply in this case.

2 Definitions and Preliminaries

First, let us formally define the notion of vertex integrity of a graph.

▶ **Definition 1.** *A graph G is said to have vertex integrity $\iota(G)$ when there exists a set $S \subset V(G)$ such that, if $S' \subset V(G)$ is the set of vertices of the largest connected component of $G \setminus S$ then $|S| + |S'| \leq \iota(G)$.*

We recall that Drange et al. [11] have shown that deciding if a graph has $\iota(G) \leq k$ admits a kernel of order $O(k^3)$. Hence, given a graph G that is promised to have vertex integrity k, we can execute this kernelization algorithm and then look for the optimal separator S in the kernel. As a result, finding a separator S proving that $\iota(G) \leq k$ can be done in $k^{O(k)} + n^{O(1)}$. Since this running time is dominated by the running times of our meta-theorems, we will always silently assume that the separator S is given in the input when the input graph has vertex integrity k.

A main question that will interest us is whether a graph satisfies a property expressible in First-Order (FO) or Monadic Second-Order (MSO) logic. Let us briefly recall the definitions of these logics. We use $x_i, i \in \mathbb{N}$ to denote vertex (FO) variables and $X_i, i \in \mathbb{N}$ to denote set (MSO) variables. Vertex variables take values from a set of vertex constants $U = \{u_i, i \in \mathbb{N}\}$, whereas vertex set variables take values from a set of vertex set constants $D = \{D_i, i \in \mathbb{N}\}$.

Now, given a graph G, in order to say that the assignment of a vertex variable x_i or a vertex set variable X_i to a constant corresponds to a particular vertex or vertex set of G, we make use of a *labeling* function ℓ that maps vertex constants to vertices of $V(G)$ and of a *coloring* function $\mathcal{C}$ that maps vertex set constants to vertex sets of $V(G)$. More formally, $\ell, \mathcal{C}$ are partial functions $\ell : U \to V(G)$ and $\mathcal{C} : D \to 2^{V(G)}$. The functions may be undefined for some constants, for example, if ℓ is not defined for the constant u_i we write $\ell(u_i) \uparrow$.

▶ **Definition 2.** *Given a triplet $G, \ell, \mathcal{C}$, a vertex $v \in V(G)$ is said to be* unlabeled *if $\nexists u_i \in U$ such that $\ell(u_i) = v$. A set of vertices $C_1 \subseteq V(G)$ is unlabeled if all the vertices of C_1 are unlabeled.*

▶ **Definition 3.** *We say that two labeling functions ℓ, ℓ'* agree *on a constant u_i if either they are both undefined on u_i or $\ell(u_i) = \ell'(u_i)$. Similarly, two coloring functions $\mathcal{C}, \mathcal{C}'$ agree on D_i if they are both undefined or $\mathcal{C}(D_i) = \mathcal{C}'(D_i)$.*

▶ **Definition 4.** *Given two triplets $G_1, \ell_1, \mathcal{C}_1$ and $G_2, \ell_2, \mathcal{C}_2$ and a bijective function $f : V(G_1) \to V(G_2)$. For $C_1 \subseteq V(G_1)$, we define $f(C_1) = \bigcup_{v \in C_1} \{f(v)\}$. We say that $V(G_1)$ and $V(G_2)$* have the same labelings for *f if $\forall u_i \in U$, either both $\ell_1(u_i), \ell_2(u_i)$ are undefined or $f(\ell_1(u_i)) = \ell_2(u_i)$; we say that $V(G_1)$ and $V(G_2)$* have the same colorings for *f if $\forall D_i \in D$, either both $\mathcal{C}_1(D_i), \mathcal{C}_2(D_i)$ are undefined or $f(\mathcal{C}_1(D_i)) = \mathcal{C}_2(D_i)$.*

▶ **Definition 5.** *An* isomorphism *between two triplets $G_1, \ell_1, \mathcal{C}_1$ and $G_2, \ell_2, \mathcal{C}_2$ is a bijective function $f : V(G_1) \to V(G_2)$ such that (i) for all $v, w \in V(G_1)$ we have $(v, w) \in E(G_1)$ if and only if $(f(v), f(w)) \in E(G_2)$, (ii) $V(G_1)$ and $V(G_2)$ have the same labelings and colorings for f. Two triplets $G_1, \ell_1, \mathcal{C}_1$ and $G_2, \ell_2, \mathcal{C}_2$ are* isomorphic *if there exists an isomorphism between them.*

▶ **Definition 6.** *Given a triplet $G, \ell, \mathcal{C}$. We say that two sets $C_1 \subseteq V(G)$ and $C_2 \subseteq V(G)$* have the same type *if there exist $\ell', \mathcal{C}'$ and an isomorphism $f : V(G) \to V(G)$ between the triplets $G, \ell, \mathcal{C}$ and itself such that f maps elements of C_1 to C_2 and vice versa and elements from $V(G) \setminus (C_1 \cup C_2)$ to themselves.*

Notice that only for vertices that don't belong in the sets C_1 and C_2 (which f maps to themselves) we can have that $f(\ell(u_i)) = \ell(u_i)$. This leads to the following observation:

▷ Observation 7. In order for two disjoint sets C_1 and C_2 to have the same type, they should necessarily be unlabeled (that is, $\forall u_i, \ell(u_i) \notin C_1 \cup C_2$).

▶ **Definition 8.** *Given a triplet $G, \ell, \mathcal{C}$ and a set $C_1 \subset V(G)$. The* restriction *of $\mathcal{C}$ to $G \setminus C_1$ is a function $\mathcal{C}' : D \to V(G) \setminus C_1$ such that $\mathcal{C}'(D_i) = \mathcal{C}(D_i) \setminus C_1$ for all $D_i \in D$ for which $\mathcal{C}(D_i) \cap C_1 \neq \emptyset$ and $\mathcal{C}, \mathcal{C}'$ agree on the rest of D_i.*

An MSO formula is a formula produced by the following grammar, where X represents a set variable, x a vertex variable, y a vertex variable or vertex constant, and Y a set variable or constant:

$$\phi \quad \to \quad \exists X.\phi \mid \exists x.\phi \mid \phi \vee \phi \mid \neg\phi \mid y \sim y \mid y = y \mid y \in Y$$

The operations above are vertex set quantification, vertex quantification, disjunction, negation, edge relation, vertex equality, and set inclusion respectively. Their semantics are defined inductively in the usual way: given a triplet $G, \ell, \mathcal{C}$ and an MSO formula ϕ, we say that the graph satisfies the property described by ϕ, or simply that $G, \ell, \mathcal{C}$ models ϕ, and write $G, \ell, \mathcal{C} \models \phi$ according to the following rules:

- $G, \ell, \mathcal{C} \models u_i \in D_j$ if $\ell(u_i)$ is defined and $\ell(u_i) \in \mathcal{C}(D_j)$.
- $G, \ell, \mathcal{C} \models u_i = u_j$ if $\ell(u_i), \ell(u_j)$ are defined and $\ell(u_i) = \ell(u_j)$.
- $G, \ell, \mathcal{C} \models u_i \sim u_j$ if $\ell(u_i), \ell(u_j)$ are defined and $(\ell(u_i), \ell(u_j)) \in E(G)$.
- $G, \ell, \mathcal{C} \models \phi \vee \psi$ if $G, \ell, \mathcal{C} \models \phi$ or $G, \ell, \mathcal{C} \models \psi$.
- $G, \ell, \mathcal{C} \models \neg\phi$ if it is not the case that $G, \ell, \mathcal{C} \models \phi$.
- $G, \ell, \mathcal{C} \models \exists x_i.\phi$ if there exists $v \in V(G)$ such that $G, \ell', \mathcal{C} \models \phi[x_i \setminus u_i]$, where $\ell(u_i) \uparrow$, $\phi[x_i \setminus u_i]$ is the formula obtained from ϕ if we replace every occurence of x_i with the (new) constant u_i and $\ell' : U \to V(G)$ is a partial function for which $\ell'(u_i) = v$, and ℓ', ℓ agree on all other values $u_j \neq u_i$.
- $G, \ell, \mathcal{C} \models \exists X_i.\phi$ if there exists $S \subseteq V(G)$ such that $G, \ell, \mathcal{C}' \models \phi[X_i \setminus D_i]$, where $\mathcal{C}(D_i) \uparrow$, $\phi[X_i \setminus D_i]$ is the formula obtained from ϕ if we replace every occurence of X_i with the (new) constant D_i and $\mathcal{C}' : D \to 2^{V(G)}$ is a partial function for which $\mathcal{C}'(D_i) = S$ and $\mathcal{C}', \mathcal{C}$ agree on all other values $D_j \neq D_i$.

If none of the above applies then $G, \ell, \mathcal{C}$ does not model ϕ and we write $G, \ell, \mathcal{C} \not\models \phi$. Observe that, from the syntactic rules presented above, a formula can have free (non-quantified) variables. However, we will only define model-checking for formulas without free variables (also called sentences). Slightly abusing notation, we will write $G \models \phi$ to mean $G, \ell, \mathcal{C} \models \phi$ for the nowhere defined functions $\ell, \mathcal{C}$. Note that our definition does not contain conjunctions or universal quantifiers, but these can be obtained from disjunctions and existential quantifiers using negations in the usual way, so we will use them freely when constructing formulas.

An FO formula is defined as an MSO formula that uses no set variables X_i. In the remainder, we will assume that all formulas are given to us in prenex form, that is, all quantifiers appear in the beginning of the formula. We call the problem of deciding whether $G, \ell, \mathcal{C} \models \phi$ the model-checking problem.

We recall the following basic fact:

▶ **Lemma 9.** *Let $G_1, \ell_1, \mathcal{C}_1$ and $G_2, \ell_2, \mathcal{C}_2$ be two isomorphic triplets. Then, for all MSO formulas ϕ we have $G_1, \ell_1, \mathcal{C}_1 \models \phi$ if and only if $G_2, \ell_2, \mathcal{C}_2 \models \phi$.*

Proof. $G_1, \ell_1, \mathcal{C}_1$ and $G_2, \ell_2, \mathcal{C}_2$ are isomorphic. Thus there exists a bijective function $f : V(G_1) \to V(G_2)$ such i) f preserves in G_2 the (non-)edges between the pairs of images of vertices in G_1 and ii) $V(G_1)$ and $V(G_2)$ have the same labelings and colorings for f.

We proceed by induction on the structure of ϕ.

- For $\phi := u_i \in D_j$. $G_1, \ell_1, \mathcal{C}_1 \models \phi$ iff $\ell_1(u_i) \in \mathcal{C}_1(D_j)$ iff $f(\ell_1(u_i)) \in f(\mathcal{C}_1(D_j))$ iff $\ell_2(u_i) \in \mathcal{C}_2(D_j)$ iff $G_2, \ell_2, \mathcal{C}_2 \models \phi$
- For $\phi := u_i = u_j$. $G_1, \ell_1, \mathcal{C}_1 \models \phi$ iff $\ell_1(u_i) = \ell_1(u_j)$ iff $f(\ell_1(u_i)) = f(\ell_1(u_j))$ iff $\ell_2(u_i) = \ell_2(u_j)$ iff $G_2, \ell_2, \mathcal{C}_2 \models \phi$
- For $\phi := u_i \sim u_j$. $G_1, \ell_1, \mathcal{C}_1 \models \phi$ iff $(\ell_1(u_i), \ell_1(u_j)) \in E(G_1)$ iff $(f(\ell_1(u_i)), f(\ell_1(u_j))) \in E(G_2)$ iff $(\ell_2(u_i), \ell_2(u_j)) \in E(G_2)$ iff $G_2, \ell_2, \mathcal{C}_2 \models \phi$
- For $\phi := \phi' \vee \phi''$, or $\phi := \neg\phi'$ By the inductive hypothesis, $G_1, \ell_1, \mathcal{C}_1 \models \phi'$ iff $G_2, \ell_2, \mathcal{C}_2 \models \phi'$ and $G_1, \ell_1, \mathcal{C}_1 \models \phi''$ iff $G_2, \ell_2, \mathcal{C}_2 \models \phi''$. Thus the statement also holds for ϕ.
- For $\phi := \exists x_i.\phi'$. We prove the one direction, the other is identical if we use f^{-1} instead of f in our arguments.
  $G_1, \ell_1, \mathcal{C}_1 \models \exists x_i.\phi'$ if there exists $v \in V(G_1)$ such that $G_1, \ell_1', \mathcal{C}_1 \models \phi[x_i \setminus u_i]$, where $\ell_1(u_i) \uparrow$, $\ell_1'(u_i) = v$, and ℓ_1', ℓ_1 agree on all other values $u_j \neq u_i$. We define a partial labeling function $\ell_2' : U \to V(G_2)$, such that $\ell_2'(u_i) = f(\ell_1'(u_i)) = f(v)$ and ℓ_2', ℓ_2 agree on all other values. It is easy to see that $G_1, \ell_1', \mathcal{C}_1$ and $G_2, \ell_2', \mathcal{C}_2$ are isomorphic, thus by the inductive hypothesis $G_2, \ell_2', \mathcal{C}_2 \models \phi[x_i \setminus u_i]$. Since $\exists f(v) \in V(G_2)$ such that $G_2, \ell_2', \mathcal{C}_2 \models \phi[x_i \setminus u_i]$ and $\ell_2(u_i) \uparrow$ (since $\ell_1(u_i) \uparrow$ and $V(G_1)$ and $V(G_2)$ have the same labelings for f), therefore $G_2, \ell_2, \mathcal{C}_2 \models \exists x_i.\phi'$.
- For $\phi := \exists X_i.\phi'$. The proof is similar with the above case. Once again we will only show the one direction.
  $G_1, \ell_1, \mathcal{C}_1 \models \exists X_i.\phi'$ if there exists $S \subseteq V(G_1)$ such that $G_1, \ell_1, \mathcal{C}_1' \models \phi[X_i \setminus D_i]$, where $\mathcal{C}_1(D_i) \uparrow$, $\mathcal{C}_1'(D_i) = S$ and $\mathcal{C}_1', \mathcal{C}_1$ agree on all other values $D_j \neq D_i$.
  We define a partial coloring function $\mathcal{C}_2' : D \to 2^{V(G_2)}$ such that $\mathcal{C}_2'(D_i) = f(\mathcal{C}_1'(D_i)) = f(S)$ and $\mathcal{C}_2', \mathcal{C}_2$ agree on all other values. Once again, $G_1, \ell_1, \mathcal{C}_1'$ and $G_2, \ell_2, \mathcal{C}_2'$ are isomorphic, thus by the inductive hypothesis $G_2, \ell_2, \mathcal{C}_2' \models \phi[X_i \setminus D_i]$. Since $\exists f(S) \subseteq V(G_2)$ such that $G_2, \ell_2, \mathcal{C}_2' \models \phi[X_i \setminus D_i]$ and we have that $\mathcal{C}_2(D_i) \uparrow$, therefore $G_2, \ell_2, \mathcal{C}_2 \models \exists X_i.\phi'$.

◀

3 FPT algorithms for FO and MSO Model-Checking parameterized by vertex integrity

In this section we prove Theorems 10 and 11. The statements appear right below.

▶ **Theorem 10.** *Given a graph G with $\iota(G) \leq k$ and an FO formula ϕ in prenex form having at most q quantifiers. Then deciding if $G \models \phi$ can be solved in time $(2^{O(k^2)} \cdot q)^q + poly(|G|)$.*

▶ **Theorem 11.** *Given a graph G with $\iota(G) \leq k$ and an MSO formula ϕ in prenex form having at most q_1 vertex variable quantifiers and at most q_2 vertex set variable quantifiers. Then deciding if $G \models \phi$ can be solved in time $\left(2^{2^{O(k^2+kq_2)}} \cdot q_1\right)^{q_1} + poly(|G|)$.*

The proofs are heavily based on Lemmata 12 and 14. The first, which is about FO Model-Checking, says that if we have at least $q+1$ components of the same type then we can erase one such component from the graph. The reason essentially is that, if $G, \ell, \mathcal{C}$ models ϕ by labeling a vertex v that belongs to the component to be removed, we can replace that vertex by a corresponding vertex in another component having the same type. Notice that the formula has q quantifiers and thus the graph will have q labels after the assignment. Since we have $q+1$ components of the same type, for one of these components the vertex that corresponds to v will be unlabeled.

The second, which is about MSO Model-Checking, says that since we can quantify over sets of vertices, unlike the case for FO, each set quantification can potentially affect a large number of components that originally had the same type (by coloring its intersection with each of them). However, since each component has size at most k, we have 2^k ways that the quantified set can overlap with the components. Thus, if we originally had a sufficiently large number of same type components, even after the coloring, we will still have a sufficient number of components that are of the same type, such that even if we remove one such component the answer of the problem won't change.

Lemmata 12 and 14, together with the fact that there exist a bounded number of types of components, give the kernels (Lemma 13 for FO and Lemma 15 for MSO).

▶ **Lemma 12.** *Given a triplet $G, \ell, \mathcal{C}$ having $q+1$ vertex sets $C_1, C_2, \dots, C_{q+1}$ of the same type and ϕ an FO formula in prenex form having q quantifiers. Then $G, \ell, \mathcal{C} \models \phi$ if and only if $G \setminus C_1, \ell, \mathcal{C}' \models \phi$, where $\mathcal{C}'$ is the restriction of $\mathcal{C}$ to $V(G) \setminus C_1$.*

Proof. We proceed by induction on the structure of the formula ϕ.

1. For $\phi := u_i \in D_j$, $\phi := u_1 = u_2$, or $\phi := u_1 \sim u_2$. From Observation 7 the sets are unlabeled. Thus $\nexists v \in C_1$ for which $\ell(u_1) = v$ or $\ell(u_2) = v$. Thus the statement of the lemma holds for the base case.
2. For $\phi := \phi_1 \vee \phi_2$ or $\phi := \neg\phi_1$. From the inductive hypothesis, we have that $G, \ell, \mathcal{C} \models \phi_1$ if and only if $G \setminus C_1, \ell, \mathcal{C}' \models \phi_1$ and that $G, \ell, \mathcal{C} \models \phi_2$ if and only if $G \setminus C_1, \ell, \mathcal{C}' \models \phi_2$. It is easy to see that the statement of the lemma holds also for ϕ.
3. The most interesting case is for $\phi := \exists x_i.\phi'$. If $G, \ell, \mathcal{C} \models \phi$ then from the definition of the semantics of ϕ there exists $v \in V(G)$ such that $G, \ell', \mathcal{C} \models \phi[x_i \setminus u_i]$ with $\ell(u_i) \uparrow$ and $\ell' : U \to V(G)$ being a partial function for which $\ell'(u_i) = v$, and ℓ' agrees with ℓ on all other values $u_j \neq u_i$.
   First we prove that without loss of generality $v \notin C_1$. Suppose that $v \in C_1$. Since C_1 and C_2 have the same type on $G, \ell, \mathcal{C}$, by Definition 6 there exists an isomorphism $f : C_1 \to C_2$. Consider now a labeling function $\ell'' : U \to V(G)$ where $\ell''(u_i) = f(\ell'(u_i)) = f(v)$, otherwise ℓ', ℓ'' agree on $u_j \neq u_i$. Observe that $G, \ell', \mathcal{C}$ and $G, \ell'', \mathcal{C}$ are isomorphic, thus from Lemma 9 we have that $G, \ell', \mathcal{C} \models \phi$ iff $G, \ell'', \mathcal{C} \models \phi$. In that case, instead of $v \in C_1$ we shall consider $f(v) \in C_2$. Thus, from now on we can assume that $v \notin C_1$
   For the triplet $G, \ell', \mathcal{C}$ q of the sets $C_1, C_2, \dots, C_{q+1}$ are still unlabeled and have the same type (C_1 is among them). Also ϕ' has $q-1$ quantifiers. Thus, by the inductive step, $G, \ell', \mathcal{C} \models \phi'$ if and only if $G \setminus C_1, \ell', \mathcal{C}' \models \phi'$. Since $v \in V(G) \setminus C_1$, we have that $G \setminus C_1, \ell, \mathcal{C}' \models \phi$.
   For the other direction, observe that $v \in V(G) \setminus C_1$ implies that $v \in V(G)$. Thus the statement holds with similar reasoning as above.

◀

▶ **Lemma 13.** *For a triplet $G, \ell, \mathcal{C}$ with vertex integrity $\iota(G) \leq k$ and with $\ell, \mathcal{C}$ everywhere undefined and for a formula ϕ with q quantifiers,* FO Model Checking *has a kernel of size*

$O(2^{k^2} \cdot q \cdot k)$, assuming we are given in the input $S \subseteq V(G)$ such that the largest component of $G \setminus S$ has size at most $k - |S|$.

Proof. We give a polynomial-time algorithm to calculate an upper bound on the number of components of $G \setminus S$ having the same type. Observe that types are only specified by the neighborhoods of the vertices of the components (ℓ and $\mathcal{C}$ are everywhere undefined thus there are no labels or colors on G).

First, we arbitrarily number the vertices of S and of each component. In order to classify the components into types, we map each component C_i to a vector $[N_1, N_2, \dots, N_{|C_i|}]$, where N_j is an ordered set containing the (numbered) neighbors of the j^{th} vertex of C_i (starting from the neighbors in S). Clearly, two components having the same vectors also have the same type, using the isomorphism that maps the i-th vertex of one to the i-th vertex of the other.

Since each component has at most k vertices and each vertex has at most 2^k different types of neighborhoods N_j, we can have at most 2^{k^2} vectors, thus at most 2^{k^2} types of components. Furthermore, since we are given S, we can test in polynomial time if two components have the same type under the arbitrary numbering we used. From Lemma 12, if more than q components have the same type we can remove one such component without changing the answer of the problem, thus we can in polynomial time either reduce the graph or conclude that each component type appears at most q times. In the end we will have at most $2^{k^2} \cdot q$ components, each having at most k vertices, thus the result. ◀

By applying the straightforward algorithm which runs in time $|V(G)|^q \cdot poly(|G|)$ for FO Model Checking, together with Lemma 13 we get the complexity promised by Theorem 10.

In order to prove Theorem 11 we need a stronger version of Lemma 12.

▶ **Lemma 14.** *Given a triplet $G, \ell, \mathcal{C}$ with at least $q' = 2^{k \cdot q_2} \cdot q_1 + 1$ vertex sets $C_1, C_2, \dots, C_{q'}$ having the same type and sizes at most k and an MSO formula ϕ in prenex form with q_1 FO quantifiers and q_2 MSO quantifiers. Then $G, \ell, \mathcal{C} \models \phi$ if and only if $G \setminus C_1, \ell, \mathcal{C}_1 \models \phi$, where $\mathcal{C}_1$ is the restriction of $\mathcal{C}$ to $V(G) \setminus C_1$.*

Proof. We proceed by induction on the structure of ϕ. We can reuse the arguments of Lemma 12, except for the case where $\phi := \exists X_i.\phi'$, so we focus on this case.

For the one direction, if $G, \ell, \mathcal{C} \models \phi$, from the definition of the semantics of ϕ, then there exists $S \subseteq V(G)$ such that $G, \ell, \mathcal{C}' \models \phi[X_i \setminus D_i]$ with $\mathcal{C}(D_i) \uparrow$ and $\mathcal{C}' : D \to 2^{V(G)}$ being a partial function for which $\mathcal{C}'(D_i) = S$, and $\mathcal{C}'$ agrees with $\mathcal{C}$ on all other values $D_j \neq D_i$.

Since each of the vertex sets $C_1, C_2, \dots, C_{q'}$ has size at most k, there are at most 2^k possible ways for S to intersect with each of them. Therefore, by pigeonhole principle, one such intersection appears in at least $\lceil \frac{q'}{2^k} \rceil = 2^{k(q_2-1)} \cdot q_1 + 1$ sets, call that group M. In order to be able to apply the inductive hypothesis, we need to prove that, without loss of generality, $C_1 \in M$.

Suppose that $C_1 \notin M$. We will do a "swapping" of C_1 with a vertex set (say C_2 without loss of generality) that does belong in the group M. Since C_1 and C_2 have the same type, that means that there exists an isomorphism $f : C_1 \to C_2$.

We consider a new coloring function $\mathcal{C}''$ that agrees with $\mathcal{C}'$ everywhere but on the constant D_i. This new coloring function will map D_i to the set of vertices S' (instead of S), where we have replaced every $v \in S \cap C_1$ with $f(v)$ and every $v \in S \cap C_2$ with $f^{-1}(v)$ (see Figure 1). More formally, $\mathcal{C}''(D_i) = S'$ where $S' = (S \setminus (C_1 \cup C_2)) \cup f(C_1 \cap S) \cup f^{-1}(C_2 \cap S)$. Then the triplets $G, \ell, \mathcal{C}'$ and $G, \ell, \mathcal{C}''$ are isomorphic and from Lemma 9 we have that $G, \ell, \mathcal{C}' \models \phi$ iff $G, \ell, \mathcal{C}'' \models \phi$. From now on we assume that C_1 belongs in M.

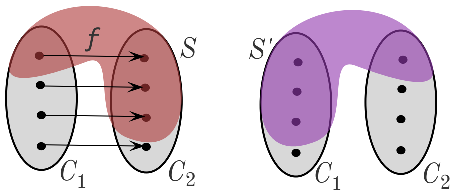

Figure 1 The way the vertex set S' intersects the vertex sets C_1 and C_2.

For the triplet $G, \ell, \mathcal{C}'$, the sets in M have all the same type and $|M| \geq 2^{k(q_2-1)} \cdot q_1 + 1$. Furthermore, the function ϕ' has q_1 FO and $q_2 - 1$ MSO quantifiers. Therefore, by the inductive hypothesis we can remove a set from M and the answer of the problem won't change, in other words we have that $G, \ell, \mathcal{C}' \models \phi'$ iff $G \setminus C_1, \ell, \mathcal{C}'_1 \models \phi'$, where $\mathcal{C}'_1$ is the restriction of $\mathcal{C}'$ on $V(G) \setminus C_1$. From the semantics of ϕ we have that $G \setminus C_1, \ell, \mathcal{C}_1 \models \phi$.

For the other direction, if $G \setminus C_1, \ell, \mathcal{C}_1 \models \phi$ then there exists $S_1 \subseteq V(G) \setminus C_1$ such that $G \setminus C_1, \ell, \mathcal{C}'_1 \models \phi[X_i \setminus D_i]$ with $\mathcal{C}_1(D_i) \uparrow$ and $\mathcal{C}_1$ being a partial coloring function for which $\mathcal{C}'_1(D_i) = S_1$, and $\mathcal{C}'_1$ agrees with $\mathcal{C}_1$ on all other values $D_j \neq D_i$.

As previously, S_1 partitions $C_2, \dots, C_{q'}$ into 2^k equivalence classes, depending on the intersection of each set with S_1, such that sets placed in the same class (i.e. having isomorphic intersection with S_1) have the same type in $G \setminus C_1, \ell, \mathcal{C}'_1$. Hence, one of these classes has size at least $\frac{q'-1}{2^k} = 2^{k(q_2-1)} \cdot q_1$, call this class M'. We construct a triplet $G, \ell, \mathcal{C}^*$ as follows: let $C_j \in M'$ and f' be the isomorphism from C_j to C_1; We set that $\mathcal{C}^*$ agrees with $\mathcal{C}$ on all sets except D_i; and for D_i we have $\mathcal{C}^*(D_i) = \mathcal{C}'_1(D_i) \cup f'(S_1 \cap C_j)$. In other words, we define $\mathcal{C}^*$ in such a way that the set C_1 has the same type as all sets of the class M'. But then we have $|M' \cup \{C_1\}| \geq 2^{k(q_2-1)} \cdot q_1 + 1$ sets of the same type and by inductive hypothesis we have $G, \ell, \mathcal{C}^* \models \phi[X_i \setminus D_i]$. Therefore, by the semantics of MSO we have $G, \ell, \mathcal{C} \models \phi$. ◀

▶ **Lemma 15.** *For a triplet $G, \ell, \mathcal{C}$ with vertex integrity $\iota(G) \leq k$ and with $\ell, \mathcal{C}$ everywhere undefined and for a formula ϕ with q_1 FO quantifiers and q_2 MSO quantifiers,* MSO Model Checking *has a kernel of size $O(2^{(k^2+kq_2)} \cdot q_1 \cdot k)$, assuming we are given in the input $S \subseteq V(G)$ such that the largest component of $G \setminus S$ has size at most $k - |S|$.*

Proof. The proof is the same as for Lemma 13. The only thing that changes is the number of same-type components required to have before removing one such component (q' required by Lemma 14 versus $q + 1$ required by Lemma 12). ◀

Applying the straightforward algorithm for MSO Model-Checking that runs in $2^{q_2 \cdot V(G)} \cdot V(G)^{q_1} \cdot poly|G|$ and Lemma 15 gives the complexity promised by Theorem 11.

4 Lower Bounds

In this section we show that the dependence of our meta-theorems on vertex integrity cannot be significantly improved, unless the ETH is false. Our strategy will be to present a unified construction which, starting from an arbitrary graph G with n vertices, produces a new graph $H(G)$, with small vertex integrity, such that we can deduce if two vertices of G are connected using appropriate constant-sized FO formulas of H. This will, in principle, allow us to express an FO or MSO-expressible property of G as a corresponding property of $H(G)$, and hence, if the original property is hard, to obtain a lower bound on model-checking on H. Let us describe this construction in more details.

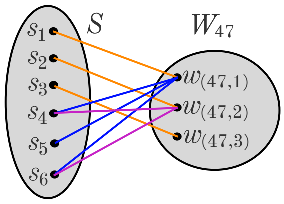

Figure 2 The connection between S and the set W_{47}. For this example $k = 3$, we can represent up to 2^9 numbers in binary. In order to represent $47_{10} = 000101111_2$, we shall connect $w_{(47,1)}$ with s_4, s_5 and s_6 in order to represent the three least significant bits (which are all 1), and $w_{(47,2)}$ with s_4 and s_6 to represent the next triad of bits. The three most significant bits are all 0.

Construction We are given a graph G on n vertices, say $V(G) = \{v_1, \ldots, v_n\}$, and m edges. Let $k = \lceil \sqrt{\log n} \rceil$. We construct a graph H as follows:

1. We begin constructing $V(H)$ by forming $n + m + 1$ sets of vertices, called S, $W_1, \ldots, W_n$, and $Y_1, \ldots, Y_m$. We have $|S| = 2k$, $|W_i| = k$ for all $i \in [n]$, and $|Y_j| = 2k + 1$ for all $j \in [m]$. The vertices of S are numbered arbitrarily as $s_1, s_2, \ldots, s_{2k}$.
2. Internally, S induces an independent set, each W_i, for $i \in [n]$ induces a clique, and each Y_j, for $j \in [m]$ induces a graph made up of two disjoint cliques of size k, denoted Y_j^1, Y_j^2, and a vertex connected to all $2k$ vertices of the cliques Y_j^1, Y_j^2.
3. For each $i \in [n]$, we attach a leaf to each vertex of W_i. For each $j \in [m]$, we attach two leaves to each vertex of Y_j^1, three leaves to each vertex of Y_j^2, and four leaves to the remaining vertex of Y_j.
4. For each $i \in [n]$, number the vertices of W_i arbitrarily as $w_{(i,1)}, w_{(i,2)}, \ldots, w_{(i,k)}$. For each $\beta \in [k]$ we connect $w_{(i,\beta)}$ to s_β. Furthermore, let $b_1 b_2 \ldots b_{k^2}$ be the binary representation of $i-1$ with the least significant digit first, that is, a sequence of bits such that $\sum_\beta b_\beta 2^{\beta-1} = i - 1$. Note that $k^2 \geq \log n$, therefore k^2 bits are sufficient to represent all numbers from 0 to $n - 1$. We partition this binary representation into k blocks of k bits. For $\beta \in [k]$ we consider the bits $b_{(\beta-1)k+1} \ldots b_{\beta k}$ and we use these bits to determine the connections between $w_{(i,\beta)}$ and the vertices $s_{k+1}, \ldots, s_{2k}$. More precisely, for $\beta, \gamma \in [k]$, we set that $w_{(i,\beta)}$ is connected to $s_{k+\gamma}$ if and only if $b_{(\beta-1)k+\gamma}$ is equal to 1.
5. For each $j \in [m]$ we do the following. Suppose the j-th edge of G has endpoints v_{i_1}, v_{i_2}. We number the vertices of Y_j^1 as $y^1_{(j,1)}, \ldots, y^1_{(j,k)}$, and the vertices of Y_j^2 as $y^2_{(j,1)}, \ldots, y^2_{(j,k)}$ in some arbitrary way. Now for all $\beta \in [k]$ we set that $y^1_{(j,\beta)}$ has the same neighbors in S as $w_{(i_1,\beta)}$ and $y^2_{(j,\beta)}$ has the same neighbors in S as $w_{(i_2,\beta)}$.

The construction of our graph is now complete. The intuition behind this construction is that each clique W_i represents a vertex $v_i \in V(G)$. In order to distinguish the vertices, we use the $k^2 \geq \log n$ possible edges between vertices in W_i and the second part of S, that is $\{s_{k+1}, \ldots, s_{2k}\}$. These edges should represent the binary representation of i. See Figure 2 for an example.

Vertices of H may be (arbitrarily) labeled for the purpose of the construction but for the purpose of Model-Checking the graph H is unlabeled. In order to give a numbering to the vertices of W_i, we use the matching between W_i and the first k vertices of the set S (the first vertex of W_i connects to the first vertex of S, etc).

The sets Y_j represent edges in G. If the j^{th} edge in $E(G)$ is the edge $(v_{i_1} v_{i_2})$, then Y_j^1 should have the same connections with S as the set W_{i_1} (similarly Y_j^2, W_{i_2}). In order to check in H whether (v_{i_1}, v_{i_2}) is an edge, we shall check if there exists a set Y_j such that each vertex of Y_j^1 has the same neighborhood in S as a vertex of W_{i_1} and each vertex of Y_j^2 has the same neighborhood in S as a vertex of W_{i_2}.

It is crucial here that the construction is such that $W_i, W_{i'}$ are distinguishable for $i \neq i'$ in terms of their neighborhoods in S, that is, there always exists $w \in W_i$ for which no $w' \in W_{i'}$ has $N(w) \cap S = N(w') \cap S$. We will show that it is not hard to express this property in FO logic. Furthermore, the leaves we have attached to various vertices will allow us to distinguish in FO logic whether a vertex belongs in a set W_i, Y_j^1, or Y_j^2.

We now establish some basic properties about H and what can be expressed about its vertices in FO logic:

▶ **Lemma 16.** *The graph H satisfies the following properties, for any coloring function $\mathcal{C}$.*

1. *We have $\iota(H) = O(\sqrt{\log n})$ and $|V(H)| = O(n^2\sqrt{\log n})$.*
2. *For each $i, i' \in [n]$ with $i \neq i'$, there exists a vertex $w \in W_i$ such that for all $w' \in W_{i'}$ we have $N(w) \cap S \neq N(w') \cap S$.*
3. *There exist constant-sized FO formulas $\phi_W(x_1), \phi_{Y1}(x_1), \phi_{Y2}(x_1), \phi_S(x_1)$ using one free variable x_1, such that $H, \ell, \mathcal{C} \models \phi_W[x_1 \setminus u_1]$ (respectively $H, \ell, \mathcal{C} \models \phi_{Y1}[x_1 \setminus u_1]$, $H, \ell, \mathcal{C} \models \phi_{Y2}[x_1 \setminus u_1]$, $H, \ell, \mathcal{C} \models \phi_S[x_1 \setminus u_1]$) if and only if $\ell(u_1) \in W_i$ for some $i \in [n]$ (respectively $\ell(u_1) \in Y_j^1$, $\ell(u_1) \in Y_j^2$, for some $j \in [m]$, $\ell(u_1) \in S$).*
4. *There exists a constant-sized FO formula ϕ_{WY} using only two free variables x_1, x_2 such that $H, \ell, \mathcal{C} \models \phi_{WY}[x_1 \setminus u_1][x_2 \setminus u_2]$ if and only if $\ell(u_1) \in W_i$ for some $i \in [n]$, $\ell(u_2) \in Y_j^\alpha$ for some $j \in [m], \alpha \in \{1, 2\}$, and for all $\beta \in [k]$ we have $N(w_{(i,\beta)}) \cap S = N(y^\alpha_{(j,\beta)}) \cap S$.*
5. *There exists a constant-sized FO formula ϕ_{adj} using only two free variables x_1, x_2 such that $H, \ell, \mathcal{C} \models \phi_{adj}[x_1 \setminus u_1][x_2 \setminus u_2]$ if and only if $\ell(u_1) \in W_i$ and $\ell(u_2) \in W_{i'}$ for some $i, i' \in [n]$ such that $(v_i, v_{i'}) \in E(G)$.*

Proof. For the first property, we observe that the largest component of $H \setminus S$ has size at most $10\sqrt{\log n} + 2$, while $|S| \leq 2\sqrt{\log n} + 2$. Furthermore, we have at most $m + n = O(n^2)$ components after removing S.

For the second property, since $i \neq i'$, their binary representations differ in some bit. Let $\beta, \gamma \in [k]$ be such that if $b_1 \dots b_{k^2}$ is the binary representation of $i - 1$ and $b'_1 \dots b'_{k^2}$ is the binary representation of $i' - 1$, we have $b_{(\beta-1)k+\gamma} \neq b'_{(\beta-1)k+\gamma}$. But then, exactly one of $w_{(i,\beta)}, w_{(i',\beta)}$ is connected to $s_{k+\gamma}$. Furthermore, $w_{(i,\beta)}$ is connected to s_β, but the only neighbor of s_β in $W_{i'}$ is $w_{(i',\beta)}$. Hence, $w_{(i,\beta)}$ is the claimed vertex.

For the third property, observe that, in H, vertices of S have no leaves attached, vertices of each X_i have one leaf attached, vertices of Y_j^1 have two leaves attached, vertices of Y_j^2 have three leaves attached, and the remaining vertices have four leaves attached. Hence, it suffices to be able to express in FO, with a constant-sized formula, the property "x_1 has exactly c leaves attached", where $c \in \{0, 1, 2, 3\}$. This is not hard to do. For example, the formula $\phi_2(x_1) := \exists x_2 \exists x_3 \forall x_4 \left((x_2 \sim x_1) \wedge (x_3 \sim x_1) \wedge (x_2 \neq x_3) \wedge ((x_4 = x_1) \vee (\neg(x_4 \sim x_2) \wedge \neg(x_4 \sim x_3)))\right)$ expresses the property that x_1 has at least two leaves attached to it. Using the same ideas we can construct $\phi_c(x_1)$, for $c \in \{1, 2, 3, 4\}$ and then $\phi_S(x_1) := \neg\phi_1(x_1)$, $\phi_W(x_1) := \phi_1(x_1) \wedge \neg\phi_2(x_1)$, $\phi_{Y1} := \phi_2(x_1) \wedge \neg\phi_3(x_1)$, $\phi_{Y2}(x_1) := \phi_3(x_1) \wedge \neg\phi_4(x_1)$.

For the fourth property, we set $\phi_{WY}(x_1, x_2) := \phi_{WY1}(x_1, x_2) \vee \phi_{WY2}(x_1, x_2)$, where we define two formulas $\phi_{WY\alpha}$ depending on whether $\alpha = 1$ or $\alpha = 2$. We have

$$\begin{aligned}\phi_{WY\alpha}(x_1, x_2) \quad := \quad & \phi_W(x_1) \wedge \phi_{Y\alpha}(x_2) \wedge \forall x_3 \big((\neg\phi_W(x_3)) \vee (\neg(x_3 \sim x_1) \wedge \neg(x_3 = x_1)) \vee \\ & \exists x_4 \left(\phi_{Y1}(x_4) \wedge (x_4 \sim x_2 \vee x_4 = x_2) \wedge \forall x_5 \left(\phi_S(x_5) \to (x_5 \sim x_3 \leftrightarrow x_5 \sim x_4)\right)\right)\big)\end{aligned}$$

What we are saying here is that $\phi_{WY1}[x_1 \setminus u_1][x_2 \setminus u_2]$ is satisfied if $\ell(u_1) \in W_i$, $\ell(u_2) \in Y_j^1$, for some $i \in [n], j \in [m]$, and for every $x_3 \in W_i$ there exists $x_4 \in Y_j^1$ such that $N(x_3) \cap S =$

$N(x_4) \cap S$. Therefore, if this property holds, then W_i and Y_j^1 represent the same vertex of V (similarly for ϕ_{WY2}).

For the last property, we set

$$\begin{aligned}\phi_{adj}(x_1, x_2) \quad := \quad & \phi_W(x_1) \wedge \phi_W(x_2) \wedge \exists x_3 \exists x_4 \big((\phi_{Y1}(x_3) \wedge \phi_{Y_2}(x_4)) \vee (\phi_{Y1}(x_4) \wedge \phi_{Y_2}(x_3))\big) \wedge \\ & \phi_{WY}(x_1, x_3) \wedge \phi_{WY}(x_2, x_4) \wedge \exists x_5(\neg\phi_S(x_5) \wedge x_3 \sim x_5 \wedge x_4 \sim x_5)\end{aligned}$$

In other words, $H, \ell, \mathcal{C} \models \phi_{adj}[x_1 \setminus u_1][x_2 \setminus u_2]$ if (i) $\ell(u_1) \in W_i$ and $\ell(u_2) \in W_{i'}$, for some $i, i' \in [n]$ (ii) there exist x_3, x_4 such that $x_3 \in Y_j^1$ and $x_4 \in Y_j^2$ for the same j; this is verified because x_3, x_4 have a common neighbor x_5 that does not belong in S (iii) $W_i, W_{i'}$ correspond to the same pair of vertices as the set $Y_j = Y_j^1 \cup Y_j^2$, which means that $(v_i, v_{i'}) \in E(G)$. ◀

We are now ready to prove our lower bounds.

▶ **Theorem 17.** *If there exists an algorithm which, given a graph G with n vertices and $\iota(G) = k$ and an FO formula ϕ with q quantifiers, decides whether $G \models \phi$ in time $2^{o(k^2 q)} n^{O(1)}$, then the ETH is false.*

Proof. We perform a reduction from q-Clique. It is well-known that, given a graph G on n vertices it is not possible to decide if G contains a clique of size q in time $n^{o(q)}$, unless the ETH is false [8]. We claim that we will construct the graph $H(G)$, as previously described, and an FO formula ϕ_C such that ϕ_C will contain $O(q)$ quantifiers and $H, \ell, \mathcal{C} \models \phi_C$ for the nowhere defined functions $\ell, \mathcal{C}$ if and only if G has a q-clique. If we achieve this, then, since by Lemma 16 we have $k = O(\sqrt{\log n})$, and the size of H is polynomially related to the size of G, the stated running time would become $2^{o(q(\sqrt{\log n})^2)} n^{O(1)} = n^{o(q)}$ and we refute the ETH. Our goal is then to define such an FO formula ϕ_C. We define

$$\begin{aligned}\phi_C \quad := \quad & \exists x_1 \exists x_2 \ldots \exists x_q \bigwedge_{i\in[q]} \phi_W(x_i) \wedge \bigwedge_{i,i'\in[q], i\neq i'} (x_i \neq x_{i'}) \\ & \forall x_{q+1} \forall x_{q+2} \bigwedge_{i\in[q]} \big(\neg(x_{q+1} = x_i)\big) \vee \bigwedge_{i\in[q]} \big(\neg(x_{q+2} = x_i)\big) \vee (x_{q+1} = x_{q+2}) \vee \\ & \phi_{adj}(x_{q+1}, x_{q+2})\end{aligned}$$

We now claim that by the construction of H, we have that $H, \ell, \mathcal{C} \models \phi_C$ if and only if G has a clique. If G has a clique $\{v_{i_1}, v_{i_2}, \ldots, v_{i_q}\}$, we map $x_1, x_2, \ldots, x_q$ to arbitrary vertices of $W_{i_1}, \ldots, W_{i_q}$. For the next part of the formula, either x_{q+1}, x_{q+2} correspond to some (different) $x_i, x_{i'}$ or the formula is true. Last, we claim that $H, \ell', \mathcal{C} \models \phi_{adj}[x_{q+1} \backslash u_i][x_{q+2}] \backslash u_{i'}]$, where $x_i, x_{i'}$ are substituted by $u_i, u_{i'}$ and $\ell'(u_i) \in W_i, \ell'(u_{i'}) \in W_{i'}$. Indeed, because we have a clique in G, by construction there exists a Y_j such that each vertex of Y_j^1 has the same neighborhood in S as W_i and each vertex of Y_j^2 has the same neighborhood in S as $W_{i'}$ (or the same with the roles of Y_j^1, Y_j^2 reversed). Hence, ϕ_{adj} is satisfied.

For the converse direction, suppose that $H, \ell, \mathcal{C} \models \phi_C$ for the nowhere defined labeling function ℓ. Then there exists a labeling function ℓ' that assigns $\ell'(u_1), \ell'(u_2), \ldots, \ell'(u_q)$ to some vertices of $\bigcup_{i\in[n]} W_i$ and is undefined everywhere else such that $\ell'(u_i) \neq \ell'(u_{i'})$ for $i \neq i'$ and $H, \ell', \mathcal{C} \models \phi_{C'}$ where

$$\phi_{C'} := \forall x_{q+1} \forall x_{q+2} \bigwedge_{i\in[q]} \big(\neg(x_{q+1} = u_i)\big) \vee \bigwedge_{i\in[q]} \big(\neg(x_{q+2} = u_i)\big) \vee (x_{q+1} = x_{q+2}) \vee \phi_{adj}(x_{q+1}, x_{q+2})$$

We extract a multi-set S of q vertices of G as follows: for $\beta \in [q]$, if $\ell'(u_\beta) \in W_i$, then we add v_i to S. We claim that for any two elements $v_i, v_{i'}$ of S we have $(v_i, v_{i'}) \in E$. If we prove this, then the vertices of S are distinct and form a q-clique in G.

Since we have universal quantifications for x_{q+1}, x_{q+2}, we can define a new labeling function ℓ'', with $\ell''(u_{q+1}) = \ell'(u_i)$ and $\ell''(u_{q+2}) = \ell'(u_{i'})$, for any $i, i' \in [q], i \neq i'$, with ℓ'', ℓ' agreeing everywhere else. Observe that this selection imposes that $H, \ell'', \mathcal{C} \models \phi_{adj}[x_{q+1} \setminus u_i][x_{q+2} \setminus u_{i'}]$ and from property 5 of Lemma 16 we get that $\ell'(u_i), \ell'(u_{i'})$ belong to two different $W_j, W_{j'}$ that correspond to the endpoints of an edge of G. ◀

▶ **Theorem 18.** *If there exists an algorithm which, given a graph G with n vertices and $\iota(G) = k$ and an MSO formula ϕ with constant size, decides whether $G \models \phi$ in time $2^{2^{o(k^2)}} n^{O(1)}$, then the ETH is false.*

Proof. Our strategy is similar to that of Theorem 17, except that we will now reduce from 3-Coloring, which is known not to be solvable in $2^{o(n)}$ on graphs on n vertices, under the ETH [33]. We will produce a constant-sized formula ϕ_{Col} with the property that $H, \ell, \mathcal{C} \models \phi_{Col}$ for the nowhere defined functions $\ell, \mathcal{C}$ if and only if G is 3-colorable. Since $k = O(\sqrt{\log n})$ an algorithm running in $2^{2^{o(k^2)}}$ would imply a $2^{o(n)}$ algorithm for 3-coloring G, contradicting the ETH. We define

$$\phi_{Col} := \exists X_1 \exists X_2 \exists X_3 \forall x_1 \forall x_2 (x_1 \in X_1 \lor x_1 \in X_2 \lor x_1 \in X_3) \land \bigwedge_{i=1,2,3} \phi_{adj}(x_1, x_2) \to \big(x_1 \in X_i \to \neg(x_2 \in X_i)\big)$$

Assume that G has a proper 3-coloring $c : V \to [3]$. Then we define, for $\alpha \in [2]$ $S_\alpha = \bigcup_{i:c(v_i)=\alpha} W_i$ and $S_3 = V(H) \setminus (S_1 \cup S_2)$. Let $\mathcal{C}'$ be a coloring function such that $\mathcal{C}'(D_\alpha) = S_\alpha$ for $\alpha = 1, 2, 3$ and $\mathcal{C}'(D_{\alpha'}) \uparrow$ for $\alpha' \notin [3]$. We claim that $H, \ell, \mathcal{C}' \models \phi_{Col}[X_1 \setminus D_1][X_2 \setminus D_2][X_3 \setminus D_3]$. Indeed, for any labeling function ℓ' that defines only $\ell'(u_1)$ and $\ell'(u_2)$ we have (i) $H, \ell', \mathcal{C}' \models u_1 \in D_1 \lor u_1 \in D_2 \lor u_1 \in D_3$ (since $\mathcal{C}'(D_1), \mathcal{C}'(D_2), \mathcal{C}'(D_3)$ is a partition of $V(H)$); (ii) If $H, \ell', \mathcal{C}' \models \phi_{adj}[x_1 \setminus u_1][x_2 \setminus u_2]$ then $\ell'(u_1) \in W_i, \ell'(u_2) \in W_{i'}$ for some $i, i' \in [n], i \neq i'$ with $(v_i, v_{i'}) \in E(G)$ (from property 5 of Lemma 16). Therefore $c(v_i) \neq c(v_{i'})$ so for $\alpha \in [3]$ $H, \ell', \mathcal{C}' \models u_1 \in D_\alpha \to \neg u_2 \in D_\alpha$.

For the converse direction, suppose that $H, \ell, \mathcal{C} \models \phi_{Col}$ for the nowhere defined $\ell, \mathcal{C}$. Then there exists a coloring function $\mathcal{C}'$ such that $\mathcal{C}'(D_\alpha) = S_\alpha$, for $\alpha \in [3]$ and $H, \ell, \mathcal{C}' \models \phi_{Col}[X_1 \setminus D_1][X_2 \setminus D_2][X_3 \setminus D_3]$. We extract a coloring of $V(G)$ as follows: for $i \in [n]$ we set $c(v_i)$ to be the minimum α such that $W_i \cap S_\alpha \neq \emptyset$. We show that the coloring $c : V(G) \to [3]$ defined in this way is proper. Consider $i, i' \in [n]$ such that $(v_i, v_{i'}) \in E(G)$. Let ℓ' be a labeling function such that $\ell'(u_1) \in W_i \cap S_{c(v_i)}$ and $\ell'(u_2) \in W_{i'} \cap S_{c(v_{i'})}$. Observe that $W_i \cap S_{c(v_i)} \neq \emptyset$ by the definition of $c(v_i)$. Then $H, \ell', \mathcal{C}' \models \phi_{adj}[x_1 \setminus u_1][x_2 \setminus u_2]$. Therefore we have that for $\alpha \in [3]$, $H, \ell', \mathcal{C}' \models u_1 \in D_\alpha \to \neg(u_2 \in D_\alpha)$. Therefore $S_{c(v_i)} \neq S_{c(v_{i'})}$, which means that $c(v_i) \neq c(v_{i'})$. ◀

References

1 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. In *ESA*, volume 173 of *LIPIcs*, pages 14:1–14:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

2 Rémy Belmonte, Michael Lampis, and Valia Mitsou. Parameterized (approximate) defective coloring. *SIAM J. Discret. Math.*, 34(2):1084–1106, 2020.

3 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016.

4 Hans L. Bodlaender, Tesshu Hanaka, Yasuaki Kobayashi, Yusuke Kobayashi, Yoshio Okamoto, Yota Otachi, and Tom C. van der Zanden. Subgraph isomorphism on graph classes that exclude a substructure. *Algorithmica*, 82(12):3566–3587, 2020.

5 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. In *FOCS*, pages 601–612. IEEE, 2020.

6 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

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

8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

9 Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Approximation schemes for first-order definable optimisation problems. In *LICS*, pages 411–420. IEEE Computer Society, 2006.

10 Holger Dell, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Tobias Mömke. Complexity and approximability of parameterized max-csps. *Algorithmica*, 79(1):230–250, 2017.

11 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016.

12 Pavel Dvořák, Eduard Eiben, Robert Ganian, Dusan Knop, and Sebastian Ordyniak. Solving integer linear programs with a small number of global variables and constraints. In *IJCAI*, pages 607–613. ijcai.org, 2017.

13 Pavel Dvořák and Dusan Knop. Parameterized complexity of length-bounded cuts and multicuts. *Algorithmica*, 80(12):3597–3617, 2018.

14 Zdenek Dvořák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *J. ACM*, 60(5):36:1–36:24, 2013.

15 Eduard Eiben, Robert Ganian, and Stefan Szeider. Meta-kernelization using well-structured modulators. *Discret. Appl. Math.*, 248:153–167, 2018.

16 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Inf. Comput.*, 209(2):143–153, 2011. URL: `https://doi.org/10.1016/j.ic.2010.11.026`, `doi:10.1016/j.ic.2010.11.026`.

17 Jirí Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theor. Comput. Sci.*, 412(23):2513–2523, 2011. URL: `https://doi.org/10.1016/j.tcs.2010.10.043`, `doi:10.1016/j.tcs.2010.10.043`.

18 Markus Frick. Generalized model-checking over locally tree-decomposable classes. *Theory Comput. Syst.*, 37(1):157–191, 2004.

19 Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48(6):1184–1206, 2001.

20 Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Ann. Pure Appl. Log.*, 130(1-3):3–31, 2004.

21 Jakub Gajarský and Petr Hliněný. Kernelizing MSO properties of trees of fixed height, and some consequences. *Log. Methods Comput. Sci.*, 11(1), 2015.

22 Robert Ganian. Improving vertex cover as a graph parameter. *Discret. Math. Theor. Comput. Sci.*, 17(2):77–100, 2015.

23 Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, and Patrice Ossona de Mendez. Shrub-depth: Capturing height of dense graphs. *Log. Methods Comput. Sci.*, 15(1), 2019.

24 Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, Patrice Ossona de Mendez, and Reshma Ramadurai. When trees grow low: Shrubs and fast MSO1. In *MFCS*, volume 7464 of *Lecture Notes in Computer Science*, pages 419–430. Springer, 2012.

25 Robert Ganian, Fabian Klute, and Sebastian Ordyniak. On structural parameterizations of the bounded-degree vertex deletion problem. *Algorithmica*, 83(1):297–336, 2021.

26 Robert Ganian and Jan Obdržálek. Expanding the expressive power of monadic second-order logic on restricted graph classes. In *IWOCA*, volume 8288 of *Lecture Notes in Computer Science*, pages 164–177. Springer, 2013.

27 Robert Ganian, Sebastian Ordyniak, and M. S. Ramanujan. On structural parameterizations of the edge disjoint paths problem. *Algorithmica*, 83(6):1605–1637, 2021. URL: `https://doi.org/10.1007/s00453-020-00795-3`, `doi:10.1007/s00453-020-00795-3`.

28 Robert Ganian, Friedrich Slivovsky, and Stefan Szeider. Meta-kernelization with structural parameters. *J. Comput. Syst. Sci.*, 82(2):333–346, 2016.

29 Tatsuya Gima, Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, and Yota Otachi. Exploring the gap between treedepth and vertex cover through vertex integrity. In *CIAC*, volume 12701 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 2021.

30 Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. *Model Theoretic Methods in Finite Combinatorics*, 558:181–206, 2011.

31 Gregory Z. Gutin, Mark Jones, and Magnus Wahlström. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM J. Discrete Math.*, 30(4):2177–2205, 2016. URL: `https://doi.org/10.1137/15M1034337`, `doi:10.1137/15M1034337`.

32 Ararat Harutyunyan, Michael Lampis, and Nikolaos Melissinos. Digraph coloring and distance to acyclicity. In *STACS*, volume 187 of *LIPIcs*, pages 41:1–41:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

33 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. URL: `https://doi.org/10.1006/jcss.2001.1774`, `doi:10.1006/jcss.2001.1774`.

34 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discret. Appl. Math.*, 264:90–117, 2019.

35 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structurally parameterized d-scattered set. *Discrete Applied Mathematics*, 2020. `doi:https://doi.org/10.1016/j.dam.2020.03.052`.

36 Leon Kellerhals and Tomohiro Koana. Parameterized complexity of geodetic set. In *IPEC*, volume 180 of *LIPIcs*, pages 20:1–20:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

37 Dusan Knop, Martin Koutecký, Tomás Masarík, and Tomás Toufar. Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity. *Log. Methods Comput. Sci.*, 15(4), 2019.

38 Dusan Knop, Tomás Masarík, and Tomás Toufar. Parameterized complexity of fair vertex evaluation problems. In *MFCS*, volume 138 of *LIPIcs*, pages 33:1–33:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

39 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. URL: `https://doi.org/10.1007/s00453-011-9554-x`, `doi:10.1007/s00453-011-9554-x`.

40 Michael Lampis. Model checking lower bounds for simple graphs. *Log. Methods Comput. Sci.*, 10(1), 2014.

41 Michael Lampis. Minimum stable cut and treewidth. In *ICALP*, volume 198 of *LIPIcs*, pages 92:1–92:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

42 Michael Lampis and Valia Mitsou. Treewidth with a quantifier alternation revisited. In *IPEC*, volume 89 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

43 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *MFCS*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011.

44 Stefan Szeider. Monadic second order logic on graphs with local cardinality constraints. *ACM Trans. Comput. Log.*, 12(2):12:1–12:21, 2011.