Decomposing Quantified Conjunctive (or Disjunctive) Formulas

Hubie Chen and Víctor Dalmau

Departament de Tecnologies de la Informació i les Comunicacions
Universitat Pompeu Fabra
Barcelona, Spain
{hubie.chen,victor.dalmau}@upf.edu

***Abstract*—Model checking–deciding if a logical sentence holds on a structure–is a basic computational task that is well-known to be intractable in general. For first-order logic on finite structures, it is PSPACE-complete, and the natural evaluation algorithm exhibits exponential dependence on the formula. We study model checking on the quantified conjunctive fragment of first-order logic, namely, prenex sentences having a purely conjunctive quantifier-free part. Following a number of works, we associate a graph to the quantifier-free part; each sentence then induces a prefixed graph, a quantifier prefix paired with a graph on its variables. We give a comprehensive classification of the sets of prefixed graphs on which model checking is tractable, based on a novel generalization of treewidth, that generalizes and places into a unified framework a number of existing results.**

I. Introduction

A. *Overview of result*

Model checking, the problem of deciding if a logical sentence holds on a structure, is a fundamental computational task that appears in many guises throughout computer science. Witness its appearance in areas such as logic, artificial intelligence, database theory, constraint satisfaction, and computational complexity, where versions thereof are often taken as canonical complete problems for complexity classes. It is well-known to be intractable in general: for first-order logic on finite structures it is PSPACE-complete, and indeed the natural algorithm for evaluating a first-order sentence ϕ on a finite structure $\mathbf{B}$ can require time $|B|^{m(\phi)}$, where $|B|$ is the size of the universe of $\mathbf{B}$, and $m(\phi)$ denotes the maximum number of free variables over subformulas of ϕ. This general intractability, coupled with the exponential dependence on the sentence, naturally prompts the search for restricted classes of sentences enjoying tractable model checking.

One fragment of first-order logic that has been heavily studied in this light is the fragment of *primitive positive* sentences, which are prenex sentences built from atomic formulas, conjunction, and existential quantification, that is, sentences having the form $\exists x_1 \ldots \exists x_m(\alpha_1 \wedge \ldots \wedge \alpha_n)$, where the x_i are variables and where the α_i are atomic formulas. These sentences have been approached from a variety of motivations and perspectives. In the database literature, they are known as *conjunctive queries* and are of central interest; the problem of model checking such sentences is also a formulation of the *constraint satisfaction problem* [17]. One approach to restricting such sentences is to restrict the *primal graph* of a sentence, which is the graph whose vertex set is the set of variables of the sentence, and where two variables are linked by an edge if they occur together in a common atomic formula. (This graph is known by a number of names, including *constraint graph* and *Gaifman graph*.) A classical result in this vein is that when the primal graphs of a set of primitive positive sentences have *bounded treewidth*, the model checking problem is polynomial-time decidable; see for instance the paper of Freuder [11]. Treewidth is a complexity measure on graphs that assigns a non-negative integer value to each finite graph; a set of graphs is said to have bounded treewidth if there exists a constant k that upper bounds the treewidth of all graphs in the set. For a set of primitive positive sentences having bounded treewidth, each sentence can be decomposed into a tree-like shape that admits efficient evaluation.

After bounded treewidth on the primal graphs (of primitive positive sentences) was identified as a sufficient condition for tractability, a natural consideration was whether or not there were other graph-based conditions that guaranteed tractability. This consideration can be formulated as follows.

Research Question 1: On which sets of primal graphs is primitive positive model checking tractable?

One can naturally ask this research question for two notions of tractability. The first is polynomial-time tractability, and the second is fixed-parameter tractability, where the formula is taken as the parameter of an instance; note that bounded treewidth implies tractability in both senses. Research Question 1 was completely resolved by Grohe, Schwentick, and Segoufin [16], who proved that bounded treewidth is the only explanation for tractability in this setting. Namely, they showed a perfect complement to the bounded treewidth tractability result: if a set of primal graphs is tractable–under either of the tractability notions–then the set has bounded treewidth. (As one would expect, this result is proved relative to a complexity-theoretic assumption, in particular, an established and widely believed assumption from parameterized complexity.) These authors make use of the *excluded grid theorem* of graph minor theory to help achieve an understanding of graph sets having unbounded treewidth. In their paper, they point to the research direction of considering larger fragments of first-order logic.

A fragment of first-order logic under current scrutiny is the class of *quantified conjunctive* sentences, which is the generalization of primitive positive sentences where both quantifiers are admitted, that is, sentences of the form $Q_1 v_1 \ldots Q_m v_m (\alpha_1 \wedge \ldots \wedge \alpha_n)$, where each $Q_i \in \{\forall, \exists\}$ is a quantifier, each v_i is a variable, and each α_j is an atomic formula. The classical quantified boolean formula (QBF) problem is a special case of model checking on such sentences, where the structures have boolean (two-element) universes. Model checking quantified conjunctive sentences is PSPACE-complete and thus in a certain sense captures the full complexity of first-order logic; this model checking problem also has the feature that many PSPACE problems can be naturally formulated within it.

Researchers have pursued the graph-based approach to identifying tractable restrictions of this fragment. One basic result, proved by Gottlob, Greco, and Scarcello [12] is that, in contrast to the primitive positive case, bounded treewidth of the primal graph is not sufficient to guarantee tractability of model checking quantified conjunctive sentences. Indeed, they show that even when the primal graph is a tree, this model checking problem is coNP-hard for Π_2 prefixes, harder for the respective higher levels of the polynomial hierarchy when further alternations are added, and PSPACE-hard for arbitrary prefixes.

The natural object pointed to by these results for further complexity studies is the pair consisting of the primal graph and the quantifier prefix of a quantified conjunctive sentence. We call such a pair a *prefixed graph*. Indeed, via this object, we have the following.

- The bounded treewidth tractability result on primitive positive sentences can be captured by considering sets of prefixed graphs having bounded treewidth and purely existential prefixes.
- The intractability results of Gottlob, Greco and Scarcello [12] can be described by considering sets of prefixed graphs having bounded treewidth and prefixes of various alternation forms.

A research issue prompted by this view of these results is to attempt to give tractability results on prefixed graphs (having arbitrary prefixes) that both generalize the given tractability result and make use of the prefix in a non-trivial way. Such tractability results were presented, for example, by Flum, Frick, and Grohe [10] and Adler and Weyer [2]. (These works in fact describe tractable fragments of general first-order model checking.) In analogy to and as a generalization of Research Question 1, one can ask for a complete description of the tractable sets of prefixed graphs.

Research Question 2: On which sets of prefixed graphs is quantified conjunctive model checking tractable?

Observe that all of the complexity results described thus far contributed towards the understanding of this research question, in particular by providing tractability or intractability results on particular sets of prefixed graphs.

In this article, we completely resolve Research Question 2, for both polynomial-time tractability and fixed-parameter tractability; we thus generalize and place into a unified framework all of the described complexity results. In particular, we introduce a new notion of width on prefixed graphs. We then prove that if a set of prefixed graphs has bounded width, then model checking is polynomial-time tractable, and hence also fixed-parameter tractable; otherwise, model checking is not fixed-parameter tractable and hence not polynomial-time tractable. In the case of bounded width, we show that sentences can be efficiently transformed so as to fall in a slight relaxation of bounded-variable first-order logic that allows for efficient evaluation. Note that model checking for bounded-variable first-order logic is well-known to be tractable (see for example Vardi [22]). As we discuss within the paper, our result also implies a classification result for model checking quantified *disjunctive* sentences.

Our width measure has a simple definition that takes into account the ordering given by the quantifier prefix and treats the two quantifiers asymmetrically. This measure is equal to treewidth (plus one!) on prefixed graphs having purely existential prefixes, and constitutes a natural generalization of treewidth in its own right. The novelty of this width measure is evidenced by an example set of formulas (described by prefixed graphs) to which our tractability result applies, but which provably do not fall into the tractable classes presented in the works of Flum, Frick, and Grohe [10] and Adler and Weyer [2].

It is worth pointing out and emphasizing that *both* our tractability results and our intractability results are novel, and are being presented for the first time in this paper. This is in contrast to many complexity dichotomy and classification theorems: oftentimes, when such theorems are established, they confirm that a known condition for intractability is the unique source of intractability, or analogously, that certain known conditions or techniques for tractability in fact are the only explanations for tractability.

B. Related work

We give a review of related work that discusses the aspects of previous articles that we see to be most highly related to the present work.

Dalmau, Kolaitis and Vardi [8] generalized the bounded treewidth tractability result on primitive positive sentences; they proved that for any set of such sentences logically equivalent to a sentence set having bounded treewidth, the set is tractable. Note that bounded treewidth implies bounded arity of relations, since a relation of arity k induces a clique of size k in the primal graph. Grohe [15] proved a complement to this tractability result by showing that, under the assumption of bounded arity, tractability of primitive positive sentences implies inclusion in the tractable class identified by Dalmau, Kolaitis and Vardi; Grohe's result also generalizes the discussed result of Grohe, Schwentick and Segoufin.

Researchers have also given complexity results for primitive positive sentences based on the hypergraph containing, for each atomic formula, an edge with the variables of the atomic formula. We describe a sampling of results; see the respective papers and the discussion therein for more information. Gottlob, Leone and Scarcello [13], [14] introduced and studied the hypergraph complexity measures of hypertree width and generalized hypertree width, and showed that bounded hypertree width constitutes a tractable class having various desirable properties. Later, the tractability of bounded generalized hypertree width was proved independently by Adler, Gottlob and Grohe [1] and Chen and Dalmau [6]. Comprehensive classification results on hypergraphs have been given by Marx under the truth-table representation of relations [19] and under the heavily-studied representation of relations via an explicit listing of tuples [18]; see also related work by Chen and Grohe [7].

We now turn to discuss results on quantified conjunctive sentences. Chen [5] presented an algorithm showing tractability of such sentences under bounded alternation, bounded treewidth, and bounded universe size on the structure. Gottlob, Greco and Scarcello [12] presented a number of complexity results, including a result showing hardness under bounded alternation and bounded treewidth, in essence showing that the bounded universe size assumption was crucial for Chen's algorithm. Pan and Vardi [20] performed a close study of the time complexity of Chen's algorithm, showing that the non-elementary growth rate with respect to the number of alternations and the treewidth is necessary.

Flum, Frick, and Grohe [10] described tractable classes for general first-order logic based on non-recursive stratified datalog programs. Chen and Dalmau [6] described a notion of treewidth for quantified conjunctive sentences, showing that bounded treewidth sentences are tractable via a consistency/pebble game type algorithm. This work is a point of contact with the empirical work on solving such sentences: Pulina and Tacchella [21] gave evidence suggesting that the Chen/Dalmau treewidth notion is a good estimator of empirical hardness. Adler and Weyer [2] generalized a tractability result of Flum, Frick, and Grohe as well as the tractability result of Chen and Dalmau by giving a notion of treewidth for first-order logic and showing that it has a number of desirable mathematical and computational properties.

Note: due to the space restriction, some of the proofs are placed in the appendix.

II. Preliminaries

a) **Graphs and prefixes:** All graphs that we will consider are undirected, finite, and simple. A graph G consists of a vertex set, denoted by $V(G)$, and an edge set, denoted by $E(G)$, which is a set of size-two subsets of $V(G)$. For two graphs G, G' with $V(G) = V(G')$ and $E(G) \subseteq E(G')$, we say that G is a *subgraph* of G', and also that G' is a *supergraph* of G. The union $G \cup G'$ of two graphs G, G' is defined to be the graph with vertex set $V(G) \cup V(G')$ and edge set $E(G) \cup E(G')$. When S is a set, we use $K(S)$ to denote the clique on S, that is, the graph with vertex set S and edge set $\{\{s, s'\} \mid s, s' \in S, s \neq s'\}$.

Let G be a graph and let $U \subseteq V(G)$ be a subset of the vertex set. The graph $G[U]$ is defined to be the graph with vertex set U and edge set $E(G) \cap E(K(U))$. The graph $G \setminus U$ is defined to be the graph $G[V(G) \setminus U]$. The set of *neighbors* of U, denoted by $N(U)$, is defined to be the set $\{v \in V(G) \setminus U \mid \exists u \in U$ such that $\{u, v\} \in E\}$.

A *quantifier prefix* is a sequence of the form $Q_1 v_1 \dots Q_n v_n$ where each $Q_i \in \{\forall, \exists\}$ is a quantifier, and the v_i are pairwise distinct variables. Relative to a quantifier prefix, a variable v_i for which $Q_i = \exists$ is called an *existentially quantified variable* or an *existential variable*; similarly, a variable v_i for which $Q_i = \forall$ is called a *universal variable* or a *universally quantified variable*. A quantifier prefix $Q_1 v_1 \dots Q_n v_n$ naturally induces an equivalence relation $\equiv_B$ on the variables $\{v_1, \dots, v_n\}$ where $v_i \equiv v_j$ if either (1) $i \leq j$ and $Q_i = Q_{i+1} = \cdots = Q_j$, or (2) $j \leq i$ and $Q_j = Q_{j+1} = \cdots = Q_i$. Each equivalence class of $\equiv_B$ is called a *block*. We say that a block is *existential* if its variables are existentially quantified, and that a block is *universal* if its variables are universally quantified. Relative to a quantifier prefix $P = Q_1 v_1 \dots Q_n v_n$, we define a preorder $\leq_P$ on the set of variables $\{v_1, \dots, v_n\}$ where $v_i \leq_P v_j$ if and only if $v_i \equiv_B v_j$ or $i \leq j$. We write $v_i <_P v_j$ if and only if $v_i \leq_P v_j$ and $v_i \not\equiv_B v_j$. We drop the subscript in $\leq_P$ and $<_P$ if the quantifier prefix is clear from the context.

A *prefixed graph* is an undirected graph G that has associated with it a quantifier prefix $P(G) = Q_1 v_1 \dots Q_n v_n$ where $v_1, \dots, v_n$ is a list of the vertices of $V(G)$, with each vertex appearing exactly once. Let G be a prefixed graph and let $U \subseteq V(G)$. We use $G[U]$ to denote the prefixed graph whose graph is $(V(G), E(G))[U]$ and whose quantifier prefix is the subsequence of $P(G)$ containing the elements of U. We use $G \setminus U$ to denote the prefixed graph $G[V(G) \setminus U]$.

b) **Parameterized complexity:** We present the elements of parameterized complexity that will be used in the paper, and refer the reader to the book by Flum and Grohe [9] for more information.

Let Σ be an alphabet used to encode decision problems. A *parameterization* is a polynomial-time computable mapping κ that maps each string $x \in \Sigma^*$ to a *parameter* $\kappa(x)$. A *parameterized problem* is a pair (Q, κ) consisting of a decision problem $Q \subseteq \Sigma^*$ and a parameterization κ.

A mapping g defined on Σ^* is said to be non-uniformly fixed-parameter tractable (nuFPT) with respect to a parameterization κ if there exist a function f and a polynomial p (both over the natural numbers) such that for every k, there exists an algorithm A_k that computes g on $\{x \in \Sigma^* \mid \kappa(x) = k\}$ in time bounded above by $f(\kappa(x))p(|x|)$. A mapping g defined on Σ^* is said to be fixed-parameter tractable (FPT) with respect to a parameterization κ if there exists a single algorithm A that can, for every k, play the role of A_k in the definition of nuFPT. A decision problem (Q, κ) is in nuFPT if the characteristic function of Q is nuFPT with respect to κ, and is in FPT if the characteristic function of Q is FPT with respect to κ.

Let (Q, κ), (Q', κ') be parameterized problems. An nuFPT (respectively, FPT) reduction from (Q, κ) to (Q', κ') is an nuFPT (respectively, FPT) mapping g such that (1) for all $x \in \Sigma^*$, it holds that $x \in Q$ if and only if $g(x) \in Q'$, and (2) for each k, the set $\kappa'(g(\{x \mid \kappa(x) = k\}))$ is finite. We will make use of the following facts.

Proposition 2.1: The composition of an nuFPT reduction from (Q, κ) to (Q', κ') and an nuFPT reduction from (Q', κ') to (Q'', κ'') is an nuFPT reduction from (Q, κ) to (Q'', κ'').

Proposition 2.2: The class of decision problems in nuFPT is closed under nuFPT reductions.

We will exhibit reductions from the k-clique problem, which we view as the parameterized problem of deciding, given a pair (G, k) consisting of a graph and a positive integer k, whether or not the graph contains a clique of size k; the parameterization is given by $\kappa(G, k) = k$. We also make use of the fact that the k-clique problem is complete for the parameterized complexity class known as $W[1]$.

c) **Problem framework:** By a *signature*, we mean a set consisting of relation symbols, each of which has a finite arity associated with it. Let σ be a signature. A *quantified conjunctive* sentence over σ is a first-order sentence of the form $P\phi$ where P is a quantifier prefix and ϕ is the conjunction of σ-atomic formulas; by a σ-atomic formula, we mean a predicate application $R(v_1, \ldots, v_k)$ where $R \in \sigma$, the v_i are variables, and k is the arity of R. We remark that in defining these sentences, we do not assume that equality "comes for free", but rather, assume that equality, if used, is explicitly represented in the signature σ. We permit arity 0 relation symbols, and say that a signature is *binary* if each relation symbol has arity less than or equal to 2.

A *structure* $\mathbf{B}$ over a signature σ consists of a *universe* B, which is a set, and a relation $R^{\mathbf{B}} \subseteq B^k$ for each $R \in \sigma$; here, k denotes the arity of R. Our results are robust across many natural representations of structures; two representations for which our results hold are (1) the representation of a relation by an explicit listing of included tuples, and (2) the representation of a relation by a truth table that contains a bit for every element of B^k, where B is the universe and k is the arity of the relation.

For a quantified conjunctive sentence $\Phi = P\phi$ with quantifier prefix $P = Q_1 v_1 \ldots Q_n v_n$, we define the prefixed graph G_Φ of Φ to be the graph with $V(G_\Phi) = \{v_1, \ldots, v_n\}$, $E(G_\Phi)$ equal to the set of all pairs $\{v_i, v_j\}$ such that v_i, v_j are different and occur together in a ϕ-atomic formula, and $P(G_\Phi) = P$.

Let $\mathcal{G}$ be a set of prefixed graphs. We define quantified conjunctive model checking over $\mathcal{G}$, denoted by $\mathsf{QC\text{-}MC}(\mathcal{G})$, to be the problem of deciding, given a pair $(\Phi, \mathbf{B})$ consisting of

- a quantified conjunctive sentence Φ having $G_\Phi \in \mathcal{G}$, and
- a structure $\mathbf{B}$,

both over the same signature, whether or not $\mathbf{B} \models \Phi$. We will generally view $\mathsf{QC\text{-}MC}(\mathcal{G})$ as a parameterized problem, and take its parameterization κ to be the mapping defined by $\kappa(\Phi, \mathbf{B}) = \Phi$.

III. Width definition and main theorem statement

We now present our width notion. An *elimination ordering* of a prefixed graph G is a pair $(G', u_1, \ldots, u_n)$ consisting of a supergraph G' of $(V(G), E(G))$ and an ordering $u_1, \ldots, u_n$ of the vertices $V(G)$ such that for all distinct variables u_i, u_j in the ordering, the following conditions hold:

(1) If u_k is an existential variable in the ordering such that $i < k$, $j < k$, $\{u_i, u_k\} \in E(G')$ and $\{u_j, u_k\} \in E(G')$, then $\{u_i, u_j\} \in E(G')$.
(2) If $\{u_i, u_j\} \in E(G')$, u_i is universal, u_j is existential and $u_i <_{P(G)} u_j$, then $i < j$.
(3) If u_i is existential, u_j is universal and $u_i <_{P(G)} u_j$, then $i < j$.

The *width* of an elimination ordering $(G', u_1, \ldots, u_n)$ is the maximum over all existential vertices u_k of the quantity $w(u_k) = 1 + |\{u_i \mid i < k, \{u_i, u_k\} \in E(G')\}|$. Note that $w(u_k)$ can be viewed as the size of the set containing u_k along with all G'-neighbors of u_k that come before it in the ordering. The *width* of a prefixed graph G is the minimum width over all of its elimination orderings. In the case that the prefixed graph G contains only existential quantification, it is readily seen that our definition specializes to the definition of treewidth based on elimination orderings, and that the width of G is equal to the treewidth of G plus one. We refer the reader to Bodlaender [4] for characterizations of treewidth based on elimination orderings.

To achieve our positive algorithmic results, we perform a translation from quantified conjunctive sentences to a certain fragment of first-order logic, defined as follows. We use $\mathrm{FO}^k_\forall$ to denote the set containing each first-order formula ϕ such that every subformula of ϕ either has k or fewer free variables or is a *universally quantified atomic formula*, by which we mean a formula of the form $\forall y_1 \ldots \forall y_j \psi$ for an atomic formula ψ. This is a relaxation of FO^k, the set of first-order formulas with at most k variables, primarily due to our allowing universally quantified atomic formulas: note that it is known and straightforward to verify that for any formula ϕ whose subformulas have at most k free variables, the formula ϕ can be rewritten to a logically equivalent FO^k-formula by renaming variables. We use $\mathrm{QCFO}^k_\forall$ to denote the set of all $\mathrm{FO}^k_\forall$ formulas built from atomic formulas, conjunction ($\wedge$), existential quantification ($\exists$), and universal quantification ($\forall$).

The following is the statement of our main theorem.

Theorem 3.1: Let $\mathcal{G}$ be a set of prefixed graphs.

- If there exists a constant $k \geq 1$ such that every prefixed graph in $\mathcal{G}$ has width less than or equal to k, then the problem $\mathsf{QC\text{-}MC}(\mathcal{G})$ is polynomial-time decidable (and hence fixed-parameter tractable). In particular, in this case, there exists a polynomial-time algorithm that, given a quantified conjunctive sentence Φ whose prefixed graph is in $\mathcal{G}$, computes a logically equivalent sentence $\Phi' \in \mathrm{QCFO}^k_\forall$.
- Otherwise, the problem $\mathsf{QC\text{-}MC}(\mathcal{G})$ is not fixed-parameter tractable, even when restricted to binary signatures, unless $\mathrm{W}[1] \subseteq$ nuFPT.

In the first case, that is, when there exists a constant $k \geq 1$ upper bounding the width of all prefixed graphs in $\mathcal{G}$, we say that the set $\mathcal{G}$ has *bounded width*; otherwise, we say that it has *unbounded width*. This theorem follows directly from Theorems 5.1, 5.2, and 6.5, proved below.

Remark 3.2: This theorem also gives a complexity classification on quantified disjunctive sentences. The definition of a quantified disjunctive sentence is that of a quantified conjunctive sentence, with the change that the quantifier-free part is a disjunction (rather than conjunction) of atomic formulas. The prefixed graph G_Φ of a quantified disjunctive sentence is defined identically. In the case that $\mathcal{G}$ is a set of graphs having bounded width, the tractability result applies to quantified disjunctive sentences; in particular, by taking the negations of quantified disjunctive sentences to obtain quantified conjunctive sentences, translating to the logic $\mathrm{QCFO}^k_\forall$, and then negating again, one obtains a translation into the logic $\mathrm{QCFO}^k_\exists$ that is dual to $\mathrm{QCFO}^k_\forall$. The intractability result also transfers to quantified disjunctive sentences: the key point is that model checking a set of quantified conjunctive sentences over binary signatures can be reduced to model checking a set of quantified disjunctive sentences in polynomial time, since computing the complements of the relations of structures can be performed in polynomial time under the assumption of bounded arity.

Remark 3.3: The non-uniformity of the complexity-theoretic assumption originates from the lack of any computability condition on the set $\mathcal{G}$. If the set $\mathcal{G}$ is assumed to be recursively enumerable, then the second part of the theorem can be proved under the (a priori) weaker assumption that W[1] $\subseteq$ FPT does not hold. (The situation is the same, for example, in the papers by Grohe, Schwentick and Segoufin [16] and Grohe [15].)

Remark 3.4: By the results of Bodirsky and Grohe [3], there exists a family $\mathcal{G}$ of prefixed graphs such that QC-MC($\mathcal{G}$) is in NP, but not NP-complete nor in P, unless P equals NP. This justifies the use of a complexity-theoretic assumption that is more refined than P $\neq$ NP.

Example 3.5: We define a set of prefixed graphs $\mathcal{G} = \{G_n\}_{n\geq 1}$ as follows. For each $n \geq 1$, define $P(G_n) = \exists x_1 \ldots \exists x_n \forall y$ and $E(G_n) = \{\{x_i, y\} \mid i \in \{1, \ldots, n\}\}$. Each prefixed graph G_n has width 1 via the elimination ordering $((V(G), E(G)), x_1, \ldots, x_n, y)$, and our main theorem (Theorem 3.1) thus implies the tractability of QC-MC($\mathcal{G}$). From [2, Proposition 1] and [2, Lemma 4], it follows directly that there is a sequence of quantified conjunctive formulas, whose prefixed graphs are those in $\mathcal{G}$, such that the sequence provably does not fall into the tractable classes presented by Flum, Frick, and Grohe [10] and Adler and Weyer [2]. $\square$

Example 3.6: Consider the set of prefixed graphs $\mathcal{G} = \{G_n\}_{n\geq 1}$ defined as follows. For each $n \geq 1$, define $P(G_n) = \exists x_1 \ldots \exists x_n \forall y \exists x$ and $E(G_n) = \{\{y, x\}\} \cup \{\{x_i, x\} \mid i \in \{1, \ldots, n\}\}$. Observe that each prefixed graph G_n is a star graph (and a tree) where if the variable y were to be removed, the result would be a star of existential variables. In an ordering satisfying the conditions of elimination ordering of G_n, by (3) each variable x_i must appear before y, and by (2) the variable y must appear before x. Hence, the only possible ordering, up to permutation of the variables x_i, is $x_1, \ldots, x_n, y, x$. For each such ordering, since x is connected to all other vertices, by condition (1) a supergraph giving an elimination ordering must connect all of the vertices, that is, must be a clique. We thus have that the width of G_n is $n+2$, and that the set $\mathcal{G}$ has unbounded width. $\square$

IV. Development

For a prefixed graph G and an arbitrary ordering $\overline{u} = u_1, \ldots, u_n$ of $V(G)$, define $G^{\overline{u}}$ to be the minimum (with respect to inclusion of the set of edges) supergraph of G satisfying condition (1) in the definition of elimination ordering; the graph $G^{\overline{u}}$ can be computed by starting from G and then iteratively adding edges wherever condition (1) is not satisfied, until a fixed point is reached. A straightforwardly verified fact that we will use is that, for any prefixed graph G and any ordering $\overline{u}$ of $V(G)$, if G' is a graph such that $(G', \overline{u})$ is an elimination ordering for G, then $(G^{\overline{u}}, \overline{u})$ is an elimination ordering which has width less than or equal to that of $(G', \overline{u})$. We shall abuse notation and denote $(G^{\overline{u}}, \overline{u})$ by $\overline{u}$, and for instance will say that $\overline{u}$ is an elimination ordering to mean that $(G^{\overline{u}}, \overline{u})$ is an elimination ordering.

In this section, we establish a number of results concerning our width notion that will be used to understand prefixed graphs from a computational standpoint. The first lemma, which follows, shows that any "projection" of an elimination ordering $\overline{u}$ for a graph G is an elimination ordering for the corresponding induced subgraph of G. The second lemma gives, for an ordering $\overline{u}$, a description of $G^{\overline{u}}$ in terms of G itself.

Lemma 4.1: If $\overline{u} = u_1, \ldots, u_n$ is an elimination ordering of a prefixed graph G then for every selection of indices $i_1, \ldots, i_m$ with $1 \leq i_1 < i_2 < \cdots < i_m \leq n$, the subsequence $\overline{u'} = u_{i_1}, \ldots, u_{i_m}$ is an elimination ordering of $G[\{u_{i_1}, \ldots, u_{i_m}\}]$ having width that is less than or equal to that of $\overline{u}$.

Proof. By assumption, we have that $(G^{\overline{u}}, \overline{u})$ is an elimination ordering of G. It is straightforward to verify (using the definition of elimination ordering) that $(G^{\overline{u}}[\{u_{i_1}, \ldots, u_{i_m}\}], \overline{u'})$, the restriction of $(G^{\overline{u}}, \overline{u})$ to $\overline{u'}$, is an elimination ordering of $G[\{u_{i_1}, \ldots, u_{i_m}\}]$ having width that is less than or equal to that of $\overline{u}$. It follows that $\overline{u'} = (G^{\overline{u'}}, \overline{u'})$ is an elimination ordering having width that is less than or equal to that of $\overline{u}$. $\square$

Lemma 4.2: Let G be a prefixed graph and $\overline{u} = u_1, \ldots, u_n$ be an ordering of $V(G)$. For every pair of indices i, j with $1 \leq i < j \leq n$, the pair $\{u_i, u_j\}$ is an element of $E(G^{\overline{u}})$ if and only if u_i and u_j are connected in $G[\{u_i, u_j\} \cup \{u_l \mid l > j, u_l \text{ is existential }\}]$.

We now introduce the notion of a *final universal variable*; intuitively, it is a universal variable y that can be eliminated from a prefixed graph G, that is, can be placed in the final position of an elimination ordering. After this, we relate the width and elimination orderings of G to those of $G \setminus \{y\}$.

Following this, we define an analogous notion for the existential variables, that of *final existential component*, and establish analogous results.

Definition 4.3: Let G be a prefixed graph. A universal variable $y \in V(G)$ that does not have any existential neighbor x with $y < x$ is called a *final* universal variable of G.

For two orderings $\overline{u_1}, \overline{u_2}$, we use the notation $(\overline{u_1}, \overline{u_2})$ to denote the concatenation of the orderings $\overline{u_1}$ and $\overline{u_2}$. With respect to a prefixed graph G, we say that an elimination ordering $(G', \overline{u})$ is *minimal* if its width is equal to that of G.

Lemma 4.4: If y is a final universal variable of a prefixed graph G then $\mathsf{width}(G) = \mathsf{width}(G \setminus \{y\})$. In particular, if $\overline{u}$ is a minimal elimination ordering for $G \setminus \{y\}$, then $(\overline{u}, y)$ is a minimal elimination ordering for G.

Proof. ($\leq$) If $\overline{u}$ is an elimination ordering of $G \setminus \{y\}$ then $(\overline{u}, y)$ is straightforwardly verified to be an elimination ordering of G of the same width.

($\geq$) This follows directly from Lemma 4.1. □

Definition 4.5: Let G be a prefixed graph and let $B_1, \ldots, B_r$ be the blocks of its quantifier prefix $P(G)$, in order. A nonempty set $C \subseteq B_r$ is a *final existential component* if

- B_r is existential,
- C is a connected component of $G[B_r]$, and
- $r = 1$ or $N(C) \cap B_{r-1} \neq \emptyset$.

For a final existential component C of a prefixed graph G, define G^{-C} to be the prefixed graph with $V(G^{-C}) = V(G) \setminus C$, $E(G^{-C}) = E(G \setminus C) \cup E(K(N(C)))$, and $P(G^{-C})$ equal to $P(G)$ but with the variables in C (and their accompanying quantifiers) removed.

Lemma 4.6: Let G be a prefixed graph and C be a final existential component of G. There is a minimal elimination ordering $(G', \overline{u})$ of G such that (1) $\overline{u} = (\overline{u_1}, \overline{u_2})$ where $\overline{u_2}$ is an ordering of C, and (2) G' contains the edges of $K(N(C))$.

Proof. Let $\overline{u}$ be a minimal elimination ordering of G. If v is a universal variable in $N(C)$ then v preceeds (in $\overline{u}$) every element of C. Indeed, let $D \subseteq C$ be maximal with the property that $G[D]$ is connected and every variable in D is preceeded by v (in $\overline{u}$). If there is some variable $w \in C \setminus D$ then, by Lemma 4.2, $G^{\overline{u}}$ contains $\{v, w\}$ contradicting condition (2) of elimination ordering.

Let $B_1, \ldots, B_r$ be the quantifier blocks of $P(G)$, in order. Write $\overline{u}$ as $u_1, \ldots, u_n$, and let $u_i \in C$.

We claim that if $j > i$ and u_j is existential then $u_j \in B_r$. Why? By definition of final existential component, $N(C) \cap B_{r-1}$ contains some element v which, by the previous claim, should preceed u_i (in $\overline{u}$). Hence, v preceeds u_j (in $\overline{u}$). If u_j was not in the block B_r, then it would be in a block strictly preceeding (in $P(G)$) the block B_{r-1}, and this would violate condition (3) of elimination ordering.

Assume now that $u_{i+1} \notin C$. We claim that $\{u_i, u_{i+1}\} \notin E(G^{\overline{u}})$. If u_{i+1} is univeral, then this follows immediately from condition (2) of elimination ordering. If u_{i+1} is existential, then this follows from the previous claim and Lemma 4.2. As a consequence of this claim, the ordering obtained by switching the positions of u_i and u_{i+1} is an elimination ordering of the same width. Iterative application of this argument shows that there is a minimal elimination ordering satisfying (1). Finally, it follows from Lemma 4.2 that any ordering satisfying (1) also satisfies (2). □

Lemma 4.7: Let G be a prefixed graph and C a final existential component. Then

$$\mathsf{width}(G) = \max(\mathsf{width}(G^{-C}), \mathsf{width}(G[C \cup N(C)]).$$

Furthermore, there is a polynomial-time algorithm that computes a minimal elimination ordering for G given a minimal elimination ordering for G^{-C} and $G[C \cup N(C)]$.

Proof. ($\geq$) Let $(G', \overline{u})$ be the minimal elimination ordering of G obtained by applying Lemma 4.6 to G and C. By applying Lemma 4.1 to the subsequences of $\overline{u}$ containing the vertices of $V(G) \setminus C$ and $C \cup N(C)$, respectively, one obtains elimination orderings for G^{-C} and $G[C \cup N(C)]$ of width less than or equal to $\mathsf{width}(G)$.

($\leq$) Let $\overline{u_{-C}} = (G^{\overline{u_{-C}}}, \overline{u_{-C}})$ be a minimal elimination ordering for G^{-C}. By Lemma 4.6, there exists a minimal elimination ordering of $G[C \cup N(C)]$ having the form $(G', (\overline{u_{N(C)}}, \overline{u_C}))$ where $\overline{u_{N(C)}}$ is an ordering of $N(C)$, $\overline{u_C}$ is an ordering of C, and G' contains the edges of $K(N(C))$. Observe that the two elimination orderings $(G^{\overline{u_{-C}}}, \overline{u_{-C}})$, $(G', (\overline{u_{N(C)}}, \overline{u_C}))$ overlap in exactly the variables $N(C)$, and each have graphs that contain the edges of $K(N(C))$.

We claim that $(G^{\overline{u_{-C}}} \cup G', (\overline{u_{-C}}, \overline{u_C}))$ is a suitable elimination ordering of G. This is verified in the appendix. □

Definition 4.8: When G is a prefixed graph containing a final existential component C such that $V(G) = C \cup N(C)$, we refer to G as a *simple* prefixed graph.

We remark that for any prefixed graph G and any final existential component C thereof, the prefixed graph $G[C \cup N(C)]$ is always simple.

We now turn to present a result on the width of simple prefixed graphs, but before doing so, present the following lemma that will be of help. We say that a prefixed graph G is *existential* if all of the variables are existentially quantified in $P(G)$. For an existential prefixed graph G, we have that the width of G is equal to the treewidth of $(V(G), E(G))$ plus one.

Lemma 4.9: Let G be an existential prefixed graph of width k and let $u_1, \ldots, u_i$ be a clique of G. Then there exists an elimination ordering for G of width k that starts with $u_1, \ldots, u_i$.

Lemma 4.9 is a well-known property of ordinary treewidth.

Lemma 4.10: Let G be a simple prefixed graph with final existential component C and let H be an existential prefixed graph with $V(H) = V(G)$ and $E(H) = E(G) \cup E(K(V(G) \setminus C))$. Then $\mathsf{width}(G) = \mathsf{width}(H) \leq |(V(G) \setminus C)| + \mathsf{width}(G[C])$.

Proof. ($\mathsf{width}(G) \leq \mathsf{width}(H)$): Let $\overline{u_1}$ be an ordering of $V(G) \setminus C$ such that for all $u, v \in V(G) \setminus C$, if $u <_{P(G)} v$ then u precedes v in $\overline{u_1}$. By Lemma 4.9 there is a minimal elimination ordering for H of the form $(\overline{u_1}, \overline{u_2})$, where $\overline{u_2}$ is an ordering of the variables in C. It is readily verified that $(\overline{u_1}, \overline{u_2})$ is an elimination ordering for G that has width less than or equal to the width of $(\overline{u_1}, \overline{u_2})$ viewed as an elimination ordering for H.

($\mathsf{width}(G) \geq \mathsf{width}(H)$): By Lemma 4.6 there exists a minimal elimination ordering $(G^{\overline{u}}, \overline{u})$ for G of the form $(\overline{u_1}, \overline{u_2})$ where $\overline{u_2}$ is an ordering of C and where $G^{\overline{u}}$ contains all edges of $K(N(C))$. Let $(H', \overline{u})$ be the elimination ordering for H induced by $\overline{u}$. It is straightforward to verify that $(H', \overline{u})$ is equal to $(G^{\overline{u}}, \overline{u})$. We claim that, with respect to this common elimination ordering, the width of H is less than or equal to the width of G.

Let $x_0 \in C$ be the first variable that occurs in $\overline{u_2}$, and let k be $|V(G) \setminus C|$. By Lemma 4.2, for each variable v in $V(G) \setminus C$, it holds that $\{v, x_0\}$ is an edge in $G^{\overline{u}}$. Hence $w(x_0)$ in $(G^{\overline{u}}, \overline{u})$ is equal to $k+1$. To establish the claim, consider any variable $v \in V(G)$. If v is existentially quantified in G, then $w(v)$ is taken into account when computing the width of $(G^{\overline{u}}, \overline{u})$ for each of G and H. If v is universally quantified in G, then $w(v)$ is taken into account when computing the width of $(G^{\overline{u}}, \overline{u})$ for H, but not G; however, $w(v)$ is less than or equal to k and hence less than or equal to $w(x_0)$.

($\mathsf{width}(H) \leq |(V(G) \setminus C)| + \mathsf{width}(G[C])$): Let $\overline{u_1}$ be an arbitrary ordering of the variables in $V(H)$ and $\overline{u_2}$ be a minimal elimination ordering for $G[C] = H[C]$. Then $(\overline{u_1}, \overline{u_2})$ is an elmination ordering for H of width at most $|(V(G) \setminus C)| + \mathsf{width}(G[C])$. □

We can now establish a basic computational property of our width notion: for any fixed k, an elimination ordering for an input prefixed graph G of width at most k can be efficiently computed, if one exists at all.

Theorem 4.11: For every $k \geq 1$ there is is a polynomial-time algorithm that, given as input a prefixed graph G, computes an elimination ordering for G of width less than or equal to k if $\mathsf{width}(G) \leq k$, and otherwise, correctly reports "$\mathsf{width}(G) > k$".

Proof. This is a direct consequence of Lemmas 4.4, 4.7, and 4.10, and the fact that one can compute treewidth decompositions in polynomial time for fixed k. □

V. Tractability

In this section, we establish the first part of Theorem 3.1. We first show that, when k upper bounds the width of a quantified conjunctive sentence, the sentence can be efficiently translated into an equivalent sentence in $\mathrm{QCFO}^k_\forall$ (Theorem 5.1). We then show that for any fixed k, sentences in $\mathrm{QCFO}^k_\forall$ can be efficiently model-checked (Theorem 5.2).

Theorem 5.1: For each constant $k \geq 1$, there exists a polynomial-time algorithm that, given any quantified conjunctive sentence Φ whose prefixed graph has width less than k computes a logically equivalent sentence $\Phi' \in \mathrm{QCFO}^k_\forall$.

The idea of the proof of Theorem 5.1 is as follows. The algorithm is defined by induction on the length of an elimination ordering $u_1, \ldots, u_n$ for Φ. Two cases are considered depending on how the last variable u_n in the elimination ordering is quantified. In each of the two cases, a formula Ψ is constructed; this formula Ψ is structurally similar to Φ, and has $u_1, \ldots, u_{n-1}$ as an elimination ordering. The algorithm can then be applied inductively to Ψ, and then from the resulting $\mathrm{QCFO}^k_\forall$ formula, the desired formula $\Phi' \in \mathrm{QCFO}^k_\forall$ can be constructed.

Theorem 5.2: If $\mathcal{G}$ is a set of prefixed graphs having bounded width, then the problem $\mathsf{QC\text{-}MC}(\mathcal{G})$ is polynomial-time decidable.

Proof. There exists a constant k upper bounding the width of all prefixed graphs in $\mathcal{G}$. The algorithm behaves as follows. Given an instance $(\Phi, \mathbf{B})$ of $\mathsf{QC\text{-}MC}(\mathcal{G})$, the algorithm invokes the algorithm of Theorem 5.1, with respect to k, to obtain a logically equivalent sentence $\Phi' \in \mathrm{QCFO}^k_\forall$. The algorithm then evaluates Φ' on $\mathbf{B}$ in the natural fashion, computing the satisfying assignments for each subformula of Φ' recursively. The set of satisfying assignments for a subformula ϕ^* of Φ is less than or equal to $|B|^k$ in the case that ϕ^* has k or fewer variables, and is less than or equal to $|R^{\mathbf{B}}|$ in the case that ϕ^* is a universally quantified atomic formula where the atomic formula is on relation symbol R. Thus, the set of satisfying assignments for each subformula is bounded above by a polynomial in the input representation, and the algorithm can be carried out in polynomial time. □

VI. Intractability

In this section, we establish hardness of the problem $\mathsf{QC\text{-}MC}(\mathcal{G})$ for graph sets $\mathcal{G}$ having unbounded width. It will be convenient to work with relatively quantified formulas. For a unary relation symbol S, we use $(\forall y \in S)\phi$ as syntactic shorthand for $\forall y(S(y) \to \phi)$. Let G be a prefixed graph. Let $R(G)$ be defined to be equal to $P(G)$, but with each universally quantified variable $\forall y$ replaced with $\forall y \in S_y$. Define ϕ_G to be the quantifier-free formula $\bigwedge_{\{v,v'\} \in E(G)} R_{(v,v')}(v, v')$. Define Φ_G to be the sentence $R(G)\phi_G$. Let σ_G be the signature $\{R_{(v,v')} \mid \{v, v'\} \in E(G)\} \cup \{S_y \mid \forall y \text{ appears in } P(G)\}$ where the $R_{(v,v')}$ are binary relation symbols and the S_y are unary relation symbols. We have that Φ_G is a sentence over σ_G. We will interpret Φ_G over structures $\mathbf{B}$ having the property that for every $\{v, v'\} \in E(G)$, it holds that $\{(a, b) \mid (a, b) \in R^{\mathbf{B}}_{(v,v')}\} = \{(a, b) \mid (b, a) \in R^{\mathbf{B}}_{(v',v)}\}$. When working with structures, for an edge $\{v, v'\} \in E(G)$, we may discuss only one of $R_{(v,v')}$, $R_{(v',v)}$, which will be justified by this property; for instance, in defining structures, we may define the interpretation of just one of $R_{(v,v')}$, $R_{(v',v)}$.

Let $\mathcal{G}$ be a set of prefixed graphs. We define $\mathsf{RQC\text{-}MC}(\mathcal{G})$ to be the parameterized problem of deciding, given a pair $(\Phi_G, \mathbf{B})$ where $G \in \mathcal{G}$ and where $\mathbf{B}$ is a structure over σ_G, whether or not $\mathbf{B} \models \Phi_G$; the prefixed graph G is taken as the parameter of an instance. We will concentrate on proving hardness of $\mathsf{RQC\text{-}MC}(\mathcal{G})$, which is justified by the following lemma.

Lemma 6.1: For any set $\mathcal{G}$ of prefixed graphs, there exists a polynomial-time reduction from the problem $\mathsf{RQC\text{-}MC}(\mathcal{G})$ to the problem $\mathsf{QC\text{-}MC}(\mathcal{G})$.

Proof. Let $(\Phi_G, \mathbf{B})$ be an instance of RQC-MC$(\mathcal{G})$. We assume that for each universally quantified variable y of G, it holds that $S_y^{\mathbf{B}} \neq \emptyset$. For each universally quantified variable y of G, we fix a mapping $f_y : B \to S_y^{\mathbf{B}}$ where f_y acts as the identity on $S_y^{\mathbf{B}}$. Let Φ be the formula $P(G)\phi_G$. We describe a structure $\mathbf{B}'$ such that $(\Phi, \mathbf{B}')$ is an equivalent instance of QC-MC$(\mathcal{G})$, as follows. We have $B' = B$.

If y, y' are both universally quantified, we define $R_{(y,y')}^{\mathbf{B}'} = B \times B$ if $R_{(y,y')}^{\mathbf{B}} \supseteq S_y^{\mathbf{B}} \times S_{y'}^{\mathbf{B}}$, and we define $R_{(y,y')}^{\mathbf{B}'} = \emptyset$ otherwise. If y is universally quantified and x is existentially quantified, we define $R_{(y,x)}^{\mathbf{B}'} = (R_{(y,x)}^{\mathbf{B}} \cap (S_y^{\mathbf{B}} \times B)) \cup \{(b_1, b_2) \mid (f_y(b_1), b_2) \in R_{(y,x)}^{\mathbf{B}}\}$. An existential winning strategy for $(\Phi, \mathbf{B}')$ is straightforwardly verified to also be an existential winning strategy for $(\Phi_G, \mathbf{B})$. In the other direction, suppose that there is an existential winning strategy for $(\Phi_G, \mathbf{B})$. Simulating this strategy and mapping the value assigned to each universally quantified variable y under f_y is straightforwardly verified to give an existential winning strategy for $(\Phi, \mathbf{B})$. $\square$

If G and H are prefixed graphs, we say that H is a *simplification* of G, denoted $H \prec G$, if:

1) $H = G^{-C}$ where C is a existential final component of G, or
2) $H = G[U]$ where $U \subsetneq V(G)$.

We will now show, in the next lemma, that when $H \prec G$, there is a quite desirable type of reduction from RQC-MC$(\{H\})$ to RQC-MC$(\{G\})$, in particular, a reduction that increases the universe size of the structure by at most a multiplicative constant (Lemma 6.2). The following lemma will then show (essentially) that, with respect to the problem RQC-MC$(\cdot)$, there is a nu-fpt reduction from any graph G' derived by taking simplifications of $\mathcal{G}$-graphs to $\mathcal{G}$ itself (Lemma 6.3); the reduction iteratively applies the reduction in Lemma 6.2.

Lemma 6.2: There exists a polynomial-time mapping that, given

- a pair H, G of non-empty prefixed graphs such that $H \prec G$ and
- an instance $(\Phi_H, \mathbf{B})$ of RQC-MC$(\{H\})$,

computes an instance $(\Phi_G, \mathbf{B}')$ of RQC-MC$(\{G\})$ that is equivalent in the sense that $\mathbf{B} \models \Phi_H$ if and only if $\mathbf{B}' \models \Phi_G$; in addition, for each such pair $H \prec G$, there exists a constant $L \geq 1$ such that any pair of instances $(\Phi_H, \mathbf{B})$, $(\Phi_G, \mathbf{B}')$ related by the mapping has $|B'| \leq L|B|$.

Proof. Suppose that $H = G[U]$ for a subset $U \subsetneq V(G)$, and let $(\Phi_H, \mathbf{B})$ be an instance of RQC-MC$(\{H\})$. Let W denote the variables $V(G) \setminus U$. We define $\mathbf{B}'$ as follows. For each $w \in W$, let b_w be a fresh value, and let $B' = B \cup \{b_w \mid w \in W\}$. For every universally quantified variable $w \in W$, we define $S_w^{\mathbf{B}'} = \{b_w\}$. For every pair of distinct variables $w, w' \in W$, we define $R_{(w,w')}^{\mathbf{B}'} = \{(b_w, b_{w'})\}$. For every pair of variables $w \in W$, $u \in U$, we define $R_{(w,u)}^{\mathbf{B}'} = \{b_w\} \times B$. It is straightforward to verify that $\mathbf{B} \models \Phi_H$ if and only if $\mathbf{B}' \models \Phi_G$.

Suppose now that $H = G^{-C}$ for a final existential component C of G with $B_1, \ldots, B_r$ the blocks of $P(G)$. If $r = 1$, we can use the previous argument, so we argue the case where $r > 1$ and where $N(C) \cap B_{r-1} \neq \emptyset$.

We define a structure $\mathbf{B}'$ in the following way.

Fix a variable $y_0 \in B_{r-1} \cap N(C)$. We define

$$S_{y_0}^{\mathbf{B}'} = \{(b, n) \mid b \in S_{y_0}^{\mathbf{B}}, n \in N(C)\}.$$

For all other universally quantified variables $y \in V(G)$, we define $S_y^{\mathbf{B}'} = S_y^{\mathbf{B}}$.

Define $D_C = \{(n \approx b) \mid n \in N(C), b \in B\}$. The universe of $\mathbf{B}'$ is $B' = B \cup D_C \cup S_{y_0}^{\mathbf{B}'}$. Clearly, we have $|B'| \leq |B|(2|N(C)| + 1)$.

We define the binary relations as follows.

1) For each edge $\{v, v'\} \in E(G)$ not containing y_0 that is in $E(H)$, define
$$R_{(v,v')}^{\mathbf{B}'} = R_{(v,v')}^{\mathbf{B}}$$
2) For each edge of the form $\{v, y_0\} \in E(G)$ that is in $E(H)$, define
$$R_{(v,y_0)}^{\mathbf{B}'} = \{(a, (b, n)) \mid (a, b) \in R_{(v,y_0)}^{\mathbf{B}}, n \in N(C)\}$$
3) For each edge $\{c, c'\} \in E(G)$ with $c, c' \in C$, define $R_{(c,c')}^{\mathbf{B}}$ to be $\{((n \approx b), (n \approx b)) \mid b \in S_{y_0}^{\mathbf{B}}, n \in N(C)\}$, that is, the equality relation on D_C.
4) For each edge $\{n, c\} \in E(G)$ with $n \in N(C) \setminus \{y_0\}$, $c \in C$, define $R_{(n,c)}^{\mathbf{B}'}$ to be the set of elements $(b, (n_c \approx b_c))$ such that
$$[n = n_c \Rightarrow b = b_c] \wedge [n \neq n_c \Rightarrow (b, b_c) \in R_{(n,n_c)}^{\mathbf{B}}]$$
5) For each edge $\{y_0, c\} \in E(G)$ with $c \in C$, define $R_{(y_0,c)}^{\mathbf{B}'}$ to be the set of elements $\{((b_0, n_0), (n_c \approx b_c))$ such that $[y_0 = n_c \Rightarrow b_0 = b_c]$ and
$$[y_0 \neq n_c \Rightarrow (b_0, b_c) \in R_{(y_0,n_c)}^{\mathbf{B}}] \wedge [n_0 = n_c]$$

We now prove that $\mathbf{B} \models \Phi_H$ if and only if $\mathbf{B}' \models \Phi_G$. We view the problem of deciding whether or not a quantified conjunctive sentence $P\phi$ holds on a structure $\mathbf{B}$ as game between an existential player and a universal player: the variables are set to values in B by the respective players according to the order P, and the existential player wins if and only if all ϕ-atomic formulas are satisfied. We use π_i to denote the operator which projects a relation onto the ith coordinate.

Suppose that existential player wins the Φ_H-game on $\mathbf{B}$. We show that the existential player can win the Φ_G-game on $\mathbf{B}$. Consider the following existential strategy for the Φ_G-game. The strategy plays according to the winning strategy for Φ_H on $B_1, \ldots, B_{r-2}$. The result is an assignment $g_{r_2} = h_{r_2}$ defined on $B_1 \cup \ldots \cup B_{r-2}$. Then, for an assignment g_{r-1} defined on B_{r-1}, we define h_{r-1} to be equal to g_{r-1} but with $h_{r-1}(y_0) = \pi_1(g_{r-1}(y_0))$. The Φ_H-winning strategy, in response to h_{r-1}, provides a response assignment h_r defined on $B_r \setminus C$ such that $h = h_{r-2} \cup h_{r-1} \cup h_r$ is a satisfying assignment of ϕ_H.

Define $n_0 = \pi_2(g_{r-1}(y_0))$. Define g_r to be the extension of h_r defined on B_r where for all $c \in C$, it holds that

$g_r(c) = (n_0 \approx h(n_0))$. Let g denote $g_{r-2} \cup g_{r-1} \cup g_r$. We verify that g satisfies all atomic formulas of ϕ_G, by considering the different cases above. For atomic formulas of type (1) and (2), it suffices to check the definition of $h(y_0) = h_{r-1}(y_0)$ in terms of $g(y_0)$ and to observe that on all other variables where h is defined, g is defined and equal to h. Atomic formulas of type (3) are clearly satisfied since g gives the same value to all $c \in C$. Consider an atomic formula $R_{(n,c)}(n, c)$ of type (4). We have $(g(n), g(c)) = (h(n), (n_0 \approx h(n_0)))$ which is straightforwardly verified to be in $R^{\mathbf{B}'}_{(n,c)}$. Consider an atomic formula $R^{\mathbf{B}'}_{(y_0,c)}(y_0, c)$ of type (5). We have $(g(y_0), g(c)) = ((h_{r-1}(y_0), n_0), (n_0 \approx h(n_0)))$, which is straightforwardly verified to be in $R^{\mathbf{B}'}_{(y_0,c)}$.

Suppose that the universal player wins the Φ_H-game on $\mathbf{B}$. We show that the universal player can win the Φ_G-game on $\mathbf{B}'$; we describe a universal strategy. First, the strategy simulates the winning universal strategy for the Φ_H-game on the blocks $B_1, \ldots, B_{r-2}$, obtaining an assignment $g_{r-2} = h_{r-2}$ to these blocks. There is an assignment h_{r-1} to B_{r-1} such that no extension of $h_{r-2} \cup h_{r-1}$ satisfies ϕ_H. We now consider two cases.

If $h_{r-2} \cup h_{r-1}$ already violates an atomic formula in ϕ_H of the form $R_{(n,n')}(n, n')$, with $n, n' \in N(C)$, then the strategy being defined sets $g_{r-1}(y_0) = (h_{r-1}(y_0), n)$, and sets g_{r-1} equal to h_{r-1} on the variables in $B_{r-1} \setminus \{y_0\}$. We claim that there is no assignment g_r defined on B_r such that $g = g_{r-2} \cup g_{r-1} \cup g_r$ satisfies ϕ_G. Consider any assignment g_r defined on B_r. Let $h' = h_{r-2} \cup h_{r-1}$, and fix an element $c \in C$. The value $g_r(c)$ has the form $(n \approx b)$. By considering the definition of $R^{\mathbf{B}'}_{(n,c)}$, it can be seen that in fact $g_r(c)$ has the form $(n \approx h'(n))$. By using the fact that $(h'(n), h'(n')) \notin R^{\mathbf{B}}_{(n,n')}$, inspection of the definition of $R^{\mathbf{B}'}_{(n,c)}$ shows that g does not satisfy ϕ_G.

If $h_{r-2} \cup h_{r-1}$ does not violate an atomic formula in ϕ_H of the form $R_{(n,n')}(n, n')$ with $n, n' \in N(C)$, then the strategy being defined sets $g_{r-1}(y_0) = (h_{r-1}(y_0), a)$ for some arbitrary $a \in N(C)$ and sets g_{r-1} equal to h_{r-1} on the variables in $B_{r-1} \setminus \{y_0\}$. We claim that there is no assignment g_r defined on B_r such that $g = g_{r-2} \cup g_{r-1} \cup g_r$ satisfies ϕ_G. Consider any assignment g_r defined on B_r, and set $h_r = g_r$. The assignment $h = h_{r-2} \cup h_{r-1} \cup h_r$ violates an atomic formula in ϕ_H. But by assumption, this atomic formula must have the form $R_{(v,v')}(v, v')$ where $\{v, v'\} \in E(G)$, and hence this atomic formula appears also in ϕ_G. The assignment g does not satisfy ϕ_G by the definition of (1) and (2). □

In this rest of this section, when $\mathcal{G}$ is a set of prefixed graphs $\mathcal{G}$, we will use $\mathcal{G}'$ to denote the closure of $\mathcal{G}$ under taking simplifications.

Lemma 6.3: For any set of prefixed graphs $\mathcal{G}$, there exists a nu-fpt reduction from RQC-MC($\mathcal{G}'$) to RQC-MC($\mathcal{G}$).

Proof. We describe a reduction. Let G' be an arbitrary prefixed graph in $\mathcal{G}'$. By definition of $\mathcal{G}'$, there exists a sequence of prefixed graphs $G' = G_1 \prec G_2 \prec \cdots \prec G_m$ with $G_m \in \mathcal{G}$. The reduction, given an instance $(\Phi_1, \mathbf{B}_1)$ of RQC-MC($\{G_1\}$), repeatedly applies the mapping of Lemma 6.2 to obtain a sequence of instances $(\Phi_1, \mathbf{B}_1) \ldots, (\Phi_m, \mathbf{B}_m)$ where $(\Phi_i, \mathbf{B}_i)$ is an instance of RQC-MC($\{G_i\}$); we use σ_i to denote its signature. The output of the reduction is $(\Phi_m, \mathbf{B}_m)$.

We bound the running time of this reduction as follows. We use the fact that there exists a polynomial s such that any binary structure $\mathbf{B}$ over signature σ (having at least one non-empty relation) has a representation of size $s(|B||\sigma|)$. We may and do assume that s has only positive coefficients. Let t be a polynomial that bounds the running time of the mapping of Lemma 6.2; we also assume that t has only positive coefficients.

Let L_i be the constant given by Lemma 6.2 for the pair $G_i \prec G_{i+1}$, for each $i = 1, \ldots, m-1$. For each i, we have $|B_i| \leq L_1 \cdots L_{i-1}|B_1|$. For each i, we thus have $|B_i| \leq L_1 \cdots L_{m-1}|B_1|$. The reduction described here invokes $m - 1$ times an algorithm of running time t on an input of size at most $s(L_1 \cdots L_{m-1}|B_1|)$. The total running time is thus bounded above by $(m-1)t(s(L_1 \cdots L_{m-1}|B_1|)) \leq (m-1)(L_1 \cdots L_{m-1})^D t(s(|B_1|))$ where D denotes the degree of the polynomial $t \cdot s$. □

Putting together the results given in this section so far, we have a nu-fpt reduction from RQC-MC($\mathcal{G}'$) to QC-MC($\mathcal{G}$). We thus need to show hardness of the problem RQC-MC($\mathcal{G}'$). The following lemma is key.

Lemma 6.4: Supose that $\mathcal{G}$ is a set of simple prefixed graphs of unbounded width. Then, there exists a nu-fpt reduction from either k-clique or co-k-clique to RQC-MC($\mathcal{G}$).

Proof. Let us consider the quantity width($G[C]$) over simple prefixed graphs G in $\mathcal{G}$ with $C = C_G$ denoting the final existential component of a graph G.

If this quantity is unbounded, then by the result of [16], there is a nu-fpt reduction from k-clique to RQC-MC($\{G[C] \mid G \in \mathcal{G}\}$), which is a particular case of RQC-MC($\mathcal{G}'$). The result then follows from Lemma 6.3.

If this quantity is bounded, we argue as follows. By Lemma 4.10, for each $k \geq 1$, there exists a graph $G \in \mathcal{G}$ such that $|V(G) \setminus C| = |N(C)| \geq k$. We consider the number of existentially quantified variables in $N(C)$ over all graphs $G \in \mathcal{G}$. If this is unbounded, then the graphs G^{-C} contain (as subgraphs) cliques of existential variables of all sizes. It follows that the set $\mathcal{G}'$ contains, as prefixed graphs, cliques of existential variables of all sizes, and the k-clique problem can be reduced directly to RQC-MC($\mathcal{G}'$); the result then follows from Lemma 6.3.

It remains to argue the case where the number of universally quantified variables in $N(C)$ is unbounded over all graphs $G \in \mathcal{G}$. By appeal to Lemma 6.3, it suffices to argue in the case where for each $k \geq 1$, there exists a graph $G \in \mathcal{G}$ with $|N(C)| = k$ and $N(C)$ contains only universal variables. We reduce from the problem co-k-clique. Let $((V(H), E(H)), k)$ be an instance of co-k-clique. Let $G \in \mathcal{G}$ be a graph with $|N(C)| = k$. We show how to encode the instance of co-k-clique as an instance of RQC-MC($\{G\}$). We define a structure $\mathbf{B}$ as follows. Define $S^{\mathbf{B}}_y = V(H)$ for each $y \in N(C)$ and define $D_C = \{(y', y'', v', v'') \in N(C)^2 \times V(H)^2 \mid y' \neq$

$y'', \{v', v''\} \notin E(H)\}$. For each pair $c, c' \in C$, define $R^{\mathbf{B}}_{(c,c')}$ to be the equality relation on D_C. For each $y \in N(C)$ and $c \in C$, define $R^{\mathbf{B}}_{(y,c)}$ to be the relation containing all pairs $(v, (y', y'', v', v'')) \in V(H) \times D_C$ such that

$$[y = y' \Rightarrow v = v'] \wedge [y = y'' \Rightarrow v = v''].$$

We claim that there is no k-clique in $(V(H), E(H))$ if and only if $\mathbf{B} \models \Phi_G$. Let $y_1, \ldots, y_k$ denote the elements of $N(C)$.

Suppose that there is no k-clique in $(V(H), E(H))$. Consider any mapping $f : N(C) \to V(H)$. There exist distinct indices i, j such that $\{f(y_i), f(y_j)\}$ is not contained in $E(H)$. It is straightforward to verify that the extension of f sending all variables $c \in C$ to $(y_i, y_j, f(y_i), f(y_j))$ satisfies ϕ_G. We conclude that $\mathbf{B} \models \Phi_G$.

Suppose that there is a k-clique $\{v_1, \ldots, v_k\}$ in $(V(H), E(H))$. Let $f : N(C) \to V(H)$ be the mapping that sends each y_i to v_i. We claim that there is no extension of f that satisfies ϕ_G. We prove this by contradiction. Assume that there is such an extension f'. By the definition of the relations $R^{\mathbf{B}}_{(c,c')}$, we have that f' sends all variables $c \in C$ to the same value. Let c be any variable in C. By the definition of the relations $R^{\mathbf{B}}_{(y,c)}$, we have that $f'(c)$ has the form $(y_i, y_j, f(y_i), f(y_j))$. But this cannot be an element of D_C, since $\{f(y_i), f(y_j)\} \in E(H)$, and we have the contradiction. We conclude that $\mathbf{B} \not\models \Phi_G$. $\square$

We can now give the main theorem of this section.

Theorem 6.5: Let $\mathcal{G}$ be a set of prefixed graphs. If $\mathcal{G}$ has unbounded width, then $\mathsf{QC\text{-}MC}(\mathcal{G})$ is not fixed-parameter tractable on binary signatures, unless W[1] $\subseteq$ nuFPT.

Proof. From Lemmas 4.4 and 4.7 it follows that if $\mathcal{G}$ has unbounded width then $\mathcal{G}'$ contains simple prefixed graphs of unbounded width. By Lemma 6.4, the problem $\mathsf{RQC\text{-}MC}(\mathcal{G}')$ admits a nu-fpt reduction from k-clique or co-k-clique. By Lemmas 6.3 and 6.1, the problem $\mathsf{QC\text{-}MC}(\mathcal{G})$, on binary signatures, does as well. $\square$

REFERENCES

[1] Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *Eur. J. Comb.*, 28(8):2167–2181, 2007.

[2] Isolde Adler and Mark Weyer. Tree-width for first order formulae. In *CSL 2009*, pages 71–85, 2009.

[3] Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of ICALP'08*, pages 184–196, 2008.

[4] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.

[5] Hubie Chen. Quantified constraint satisfaction and bounded treewidth. In *16th European Conference on Artificial Intelligence (ECAI)*, 2004.

[6] Hubie Chen and Victor Dalmau. From Pebble Games to Tractability: An Ambidextrous Consistency Algorithm for Quantified Constraint Satisfaction. In *Computer Science Logic 2005*, 2005.

[7] Hubie Chen and Martin Grohe. Constraint satisfaction with succinctly specified relations. *Journal of Computer and System Sciences*, 76(8):847–860, 2010.

[8] Victor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *Constraint Programming '02*, LNCS, 2002.

[9] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

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

[11] Eugene C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *AAAI 1990*, pages 4–9, 1990.

[12] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. The complexity of quantified constraint satisfaction problems under structural restrictions. In *IJCAI 2005*, 2005.

[13] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002.

[14] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. *J. Comput. Syst. Sci.*, 66(4):775–808, 2003.

[15] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1), 2007.

[16] Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In *STOC 2001*, 2001.

[17] P. Kolaitis and M. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.

[18] Dániel Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 735–744, 2010.

[19] Dániel Marx. Tractable structures for constraint satisfaction with truth tables. *Theory of Computing Systems*, 48:444–464, 2011.

[20] G. Pan and M. Vardi. Fixed-parameter hierarchies inside pspace. In *21st Annual IEEE Symposium on Logic in Computer Science*, pages 27–36, 2006.

[21] Luca Pulina and Armando Tacchella. Treewidth: A useful marker of empirical hardness in quantified boolean logic encodings. In *15th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, pages 528–542, 2008.

[22] Moshe Y. Vardi. On the complexity of bounded-variable queries. In *PODS'95*, pages 266–276, 1995.

APPENDIX

PROOF OF LEMMA 4.2

Proof. Consider the sequence of supergraphs $G_i^{\overline{u}}$ of $(V(G), E(G))$, defined for $i = n, \ldots, 1$ inductively as follows:

- $G_n^{\overline{u}} = (V(G), E(G))$
- $G_{i-1}^{\overline{u}} = G_i$ if u_i is universal
- $E(G_{i-1}^{\overline{u}}) = E(G_i^{\overline{u}}) \cup K(\{u_j \mid j < i, u_j \in N(u_i)\})$ if u_i is existential

For every k with $1 \leq k \leq n$ define V_k as $\{u_l \mid l > k, u_l \text{ is existential }\}$. Observe that $G_0^{\overline{u}} = G^{\overline{u}}$ and that $\{u_i, u_j\} \in G^{\overline{u}}$ if and only if $\{u_i, u_j\} \in G_j^{\overline{u}}$. The result follows by combining this observation with the following claim: for every $1 \leq i < j \leq k \leq n$ such that $\{u_i, u_j\} \notin E(G)$, $\{u_i, u_j\} \in G_k^{\overline{u}}$ if and only if u_i and u_j are connected in $G[\{u_i, u_j\} \cup V_k]$.

We shall finish the proof by proving the claim. Let i, j, k be a counterexample to the claim with $k - j$ minimum. Since $G_n^{\overline{u}} = G$ it follows that $k < n$. By the minimality of $k - j$ we can assume that $\{u_i, u_j\} \notin G_{k+1}^{\overline{u}}$, that u_i, u_j is not connected in $G[\{u_i, u_j\} \cup V_{k+1}]$, and that u_{k+1} is existential. Hence, we have that $\{u_i, u_j\} \in G_k^{\overline{u}}$ if and only if for every $l \in \{i, j\}$, $\{u_l, u_{k+1}\}$ is an edge of $G_{k+1}^{\overline{u}}$, which by induction hypothesis, is equivalent to the fact that for every $l \in \{i, j\}$, u_l and u_{k+1} are connected in $G[\{u_l, u_{k+1}\} \cup V_{k+1}]$. This is equivalent, since, u_i and u_j are not connected in $G[V_{k+1}]$), to the fact u_i and u_j are connected in $G[\{u_i, u_j\} \cup V_k]$. □

COMPLETION OF PROOF OF LEMMA 4.7

Proof. We show that $(G^{\overline{u_{-C}}} \cup G', (\overline{u_{-C}}, \overline{u_C}))$ is a suitable elimination ordering of G. First, observe that any G-edge including a vertex in C is contained in $G[C \cup N(C)]$ and hence G', and any G-edge including a vertex in $V(G) \setminus C$ is contained in G^{-C} and hence $G^{\overline{u_{-C}}}$; thus, the graph $G^{\overline{u_{-C}}} \cup G'$ is a supergraph of G. We now verify each of the conditions (1)-(3).

(1) Consider first an existential variable u_k in C. Any edges including u_k must be contained in $E(G')$, and hence condition (1) is satisfied for u_k, since $(G', (\overline{u_{N(C)}}, \overline{u_C}))$ is an elimination ordering. Consider next an existential variable u_k in $V(G) \setminus C$, and assume that $i < k$ and $j < k$. The variables u_i, u_j, u_k are all contained in $\overline{u_{-C}}$, and any edges between the variables u_i, u_j, u_k that are contained in $G^{\overline{u_{-C}}} \cup G'$ must be contained in $G^{\overline{u_{-C}}}$. That condition (1) holds for u_k follows from the fact that $(G^{\overline{u_{-C}}}, \overline{u_{-C}})$ is an elimination ordering.

(2) Suppose that $\{u_j, u_j\}$ is an edge, u_i is universal, u_j is existential, and $u_i <_{P(G)} u_j$. The variable u_i must be contained in $V(G) \setminus C$, since it is universal. If u_j is contained in $V(G) \setminus C$, then $i < j$ follows from the fact that $(G^{\overline{u_{-C}}}, \overline{u_{-C}})$ is an elimination ordering (and itself obeys (2)). If u_j is contained in C, then $i < j$ follows directly from the definition of the given ordering.

(3) Suppose that u_i is existential, u_j is universal, and $u_i <_{P(G)} u_j$. The variable u_j must be contained in $V(G) \setminus C$, since it is universal. The variable u_i must also be contained in $V(G) \setminus C$, since it does not occur in the last block and hence cannot be an element of a final existential component. Thus, $i < j$ follows from the fact that $(G^{\overline{u_{-C}}}, \overline{u_{-C}})$ is an elimination ordering (and itself obeys (3)).

Having established that $(G^{\overline{u_{-C}}} \cup G', (\overline{u_{-C}}, \overline{u_C}))$ is an elimination ordering of G, it remains only to show that this elimination ordering has width less than or equal to $\max(\mathsf{width}(G^{-C}), \mathsf{width}(G[C \cup N(C)]))$. To demonstrate this, the following observation suffices. For any existential variable x, the value of $w(x)$ in our elimination ordering $(G^{\overline{u_{-C}}} \cup G', (\overline{u_{-C}}, \overline{u_C}))$ is equal to the value of $w(x)$ in $(G^{\overline{u_{-C}}}, \overline{u_{-C}})$ when $x \in V(G) \setminus C$, and is equal to the value of $w(x)$ in $(G', (\overline{u_{N(C)}}, \overline{u_C}))$ when $x \in C$. □

PROOF OF THEOREM 5.1

Proof. The algorithm first invokes the algorithm of Theorem 4.11 to obtain an elimination ordering $\overline{u} = u_1, \ldots, u_n$ of the prefixed graph G of Φ. Following this, we define the behavior of the algorithm inductively on the length of the elimination ordering $\overline{u}$. In the case that $n = 0$, the sentence Φ does not contain any variables, and the algorithm outputs Φ.

Now suppose that the sequence has length $n > 0$. Assume that Φ has the form $P\phi$ where P is the quantifier prefix and ϕ is the quantifier-free part. We describe how the sentence Φ' is computed in two cases, depending on whether u_n is existential or universal.

Assume first that u_n is universal. We write ϕ as a conjunction of atomic formulas $\phi_1 \wedge \cdots \wedge \phi_m$. Variable u_n is a final universal variable by condition (2) of elimination ordering, and thus we have that Φ is logically equivalent to $P'\forall u_n \phi$ where P' is obtained by removing $\forall u_n$ from P. Also we have

$$P'\forall u_n(\phi_1 \wedge \cdots \wedge \phi_m) \equiv P'((\forall u_n \phi_1) \wedge \cdots \wedge (\forall u_n \phi_m))$$

where $\equiv$ denotes logical equivalence. Define $\psi_i = (\forall u_n \phi_i)$ for all $i \in \{1, \ldots, m\}$. Define Ψ to be the formula $P'(R_1(\overline{t_1}) \wedge \cdots \wedge R_m(\overline{t_m}))$ where the R_i are new relation symbols and $\overline{t_i}$ is a tuple containing the free variables of ψ_i.

The ordering $u_1, \ldots, u_{n-1}$ is an elimination ordering of the prefixed graph of Ψ, and so the algorithm can be invoked recursively to obtain a sentence $\Psi' \in \mathrm{QCFO}_{\forall}^k$ that is logically equivalent to Ψ. Define Φ' to be the formula obtained from Ψ' by replacing each instance of $R_i(\overline{t_i})$ in Ψ' with ψ_i. The formula Φ' is clearly logically equivalent to Φ. We now argue that $\Phi' \in \mathrm{QCFO}_{\forall}^k$.

Consider a subformula ϕ^* of Φ'. If ϕ^* is a subformula of one of the ψ_i, then by definition ϕ^* is a universally quantified atomic formula. Otherwise, we have that ϕ^* is obtained from a subformula ψ^* of Ψ' by replacing each instance of $R_i(\overline{t_i})$ in ψ^* with ψ_i. By induction, it holds that $\Psi' \in \mathrm{QCFO}_{\forall}^k$, implying that ψ^* either has k or fewer free variables or is a universally quantified atomic formula. It follows that ϕ^* must also have k or fewer variables or be a universally quantified atomic formula.

Assume now that u_n is existential. By condition (3) of elimination ordering u_n belongs to the last quantifier block of P implying that Φ is logically equivalent to $P'\exists u_n\phi$, where P' is obtained by removing $\exists u_n$ from P. Write ϕ as $\phi_1 \wedge \phi_2$ where ϕ_1 is the conjunction of all atomic formulas from ϕ that do not contain u_n, and ϕ_2 is the conjuntion of all atomic formulas from ϕ that do contain u_n. We have $P'\exists u_n(\phi_1 \wedge \phi_2) \equiv P'(\phi_1 \wedge (\exists u_n \phi_2))$. Define Ψ to be the formula $P'(\phi_1 \wedge R(\bar{t}))$, where R is a new relation symbol and $\bar{t}$ is a tuple containing the free variables of $\exists u_n\phi_2$. The prefixed graph of Ψ has as elimination ordering $(G^{\overline{u}}[\{u_1, \ldots, u_{n-1}\}], (u_1, \ldots, u_{n-1}))$, and so the algorithm can be invoked recursively to obtain a sentence $\Psi' \in \mathrm{QCFO}^k_\forall$ that is logically equivalent to Ψ. Define Φ' to be the formula obtained from Ψ' by replacing $R(\bar{t})$ in Ψ' with $\exists u_n\phi_2$. The formula Φ' is clearly logically equivalent to Φ.

We now argue that $\Phi' \in \mathrm{QCFO}^k_\forall$. Consider a subformula ϕ^* of Φ'. If ϕ^* is a subformula of one of the $\exists u_n\phi_2$, then the claim follows because ϕ_2 has at most k variables. Otherwise ϕ^* is obtained from a subformula ψ^* of Ψ' by replacing each instance of $R(\bar{t})$ in ψ^* with $\exists u_n\phi_2$. By induction, we have that $\Psi' \in \mathrm{QCFO}^k_\forall$. We consider the form of ψ^*. If ψ^* has k or fewer free variables, then ϕ^* also has k or fewer free variables. If ψ^* is a universally quantified atomic formula, we consider the atomic formula underlying ψ^*. If this atomic formula came from ϕ_1, then ϕ^* is equal to ψ^*. If this atomic formula is equal to $R(\bar{t})$, then it must have $(k-1)$ or fewer variables, since $\bar{t}$ has $(k-1)$ or fewer variables; it follows that ϕ^* must also have $(k-1)$ or fewer variables. $\square$