Counting Answers to Unions of Conjunctive Queries: Natural Tractability Criteria and Meta-Complexity

JACOB FOCKE*, CISPA, Helmholtz Center for Information Security, Germany
LESLIE ANN GOLDBERG*, Department of Computer Science, University of Oxford, U.K.
MARC ROTH*, Department of Computer Science, University of Oxford, U.K. and School of Electronic Engineering and Computer Science, Queen Mary University of London, U.K.
STANISLAV ŽIVNÝ*, Department of Computer Science, University of Oxford, U.K.

We study the problem of counting answers to unions of conjunctive queries (UCQs) under structural restrictions on the input query. Concretely, given a class C of UCQs, the problem #UCQ(C) provides as input a UCQ $\Psi \in C$ and a database $\mathcal{D}$ and the problem is to compute the number of answers of Ψ in $\mathcal{D}$.

Chen and Mengel [PODS'16] have shown that for any recursively enumerable class C, the problem #UCQ(C) is either fixed-parameter tractable or hard for one of the parameterised complexity classes W[1] or #W[1]. However, their tractability criterion is unwieldy in the sense that, given any concrete class C of UCQs, it is not easy to determine how hard it is to count answers to queries in C. Moreover, given a single specific UCQ Ψ, it is not easy to determine how hard it is to count answers to Ψ.

In this work, we address the question of finding a *natural* tractability criterion: The combined conjunctive query of a UCQ $\Psi = \varphi_1 \vee \cdots \vee \varphi_\ell$ is the conjunctive query $\bigwedge(\Psi) = \varphi_1 \wedge \cdots \wedge \varphi_\ell$. We show that under natural closure properties of C, the problem #UCQ(C) is fixed-parameter tractable if and only if the combined conjunctive queries of UCQs in C, and their contracts, have bounded treewidth. A contract of a conjunctive query is an augmented structure, taking into account how the quantified variables are connected to the free variables — if all variables are free, then a conjunctive query is equal to its contract; in this special case the criterion for fixed-parameter tractability of #UCQ(C) thus simplifies to the combined queries having bounded treewidth.

Finally, we give evidence that a closure property on C is necessary for obtaining a natural tractability criterion: We show that even for a single UCQ Ψ, the meta problem of deciding whether #UCQ($\{\Psi\}$) can be solved in time $O(|\mathcal{D}|^d)$ is NP-hard for any fixed $d \geq 1$. Moreover, we prove that a known exponential-time algorithm for solving the meta problem is optimal under assumptions from fine-grained complexity theory. As a corollary of our reduction, we also establish that approximating the Weisfeiler-Leman-Dimension of a UCQ is NP-hard.

CCS Concepts: • **Theory of computation → Design and analysis of algorithms**; • **Information systems → Relational database query languages**.

*For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper. Stanislav Živný was supported by UKRI EP/X024431/1. A full version of this work can be found at [31].

Authors' addresses: Jacob Focke, CISPA, Helmholtz Center for Information Security, Saarbrücken, Germany; Leslie Ann Goldberg, Department of Computer Science, University of Oxford, Oxford, U.K.; Marc Roth, Department of Computer Science, University of Oxford, Oxford, U.K. and School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K.; Stanislav Živný, Department of Computer Science, University of Oxford, Oxford, U.K..

ACM 2836-6573/2024/5-ART113
https://doi.org/10.1145/3651614

Additional Key Words and Phrases: counting problems, conjunctive queries, unions of conjunctive queries, simplicial complexes, Weisfeiler-Leman algorithm

ACM Reference Format:
Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Živný. 2024. Counting Answers to Unions of Conjunctive Queries: Natural Tractability Criteria and Meta-Complexity. *Proc. ACM Manag. Data* 2, 2 (PODS), Article 113 (May 2024), 17 pages. https://doi.org/10.1145/3651614

1 INTRODUCTION

Conjunctive queries are among the most fundamental and well-studied objects in database theory [2, 20, 42, 43, 54, 55]. A conjunctive query (CQ) φ with free variables $X = \{x_1, \ldots, x_k\}$ and quantified variables $Y = \{y_1, \ldots, y_d\}$ is of the form

$$\varphi(X) = \exists Y\, R_1(\mathbf{t}_1) \wedge \ldots \wedge R_n(\mathbf{t}_n),$$

where $R_1, \ldots, R_n$ are relational symbols and each $\mathbf{t}_i$ is a tuple of variables from $X \cup Y$. A database $\mathcal{D}$ consists of a set of elements $U(\mathcal{D})$, denoted the *universe* of $\mathcal{D}$, and a set of relations over this universe. The corresponding relation symbols are the *signature* of $\mathcal{D}$. If $R_1, \ldots, R_n$ are in the signature of $\mathcal{D}$ then an *answer* of φ in $\mathcal{D}$ is an assignment $a\colon X \to U(\mathcal{D})$ that has an extension to the existentially quantified variables Y that agrees with all the relations $R_1, \ldots, R_n$. Even more expressive is a union of conjunctive queries (UCQ). Such a union is of the form $\Psi(X) = \varphi_1(X) \vee \ldots \vee \varphi_\ell(X)$, where each $\varphi_i(X)$ is a CQ with free variables X. An answer to Ψ is then any assignment that is answer to at least one of the CQs in the union.

Since evaluating a given CQ on a given database is NP-complete [20] a lot of research focused on finding tractable classes of CQs. A fundamental result by Grohe, Schwentick, and Segoufin [38] established that the tractability of evaluating all CQs of bounded arity whose Gaifman graph is in some class of graphs C depends on whether or not the treewidth in C is bounded.

More generally, finding an answer to a conjunctive query can be cast as finding a (partial) homomorphism between relational structures, and therefore is closely related to the framework of constraint satisfaction problems. In this setting, Grohe [36] showed that treewidth modulo homomorphic equivalence is the right criterion for tractability. There is also an important line of work [34, 37, 45] culminating in the fundamental work by Marx [46] that investigates the parameterised complexity for classes of queries with unbounded arity. In general, tractability of conjunctive queries is closely related to how "tree-like" or close to acyclic they are.

Counting answers to CQs has also received significant attention in the past [3, 22, 25, 27, 28, 35, 51]. Chen and Mengel [21] gave a complete classification for the counting problem on classes of CQs (with bounded arity) in terms of a natural criterion loosely based on treewidth. They present a trichotomy into fixed-parameter tractable, W[1]-complete, and #W[1]-complete cases. In subsequent work [22], this classification was extended to unions of conjunctive queries (and to even more general queries in [27]). However, for UCQs, the established criteria for tractability and intractability are implicit (see [22, Theorems 3.1 and 3.2]) in the sense that, given a specific UCQ Ψ, it is not at all clear how hard it is to count answers to Ψ based on the criteria in [22]. To make this more precise: It is not even clear whether we can, in polynomial time in the size of Ψ, determine whether answers to Ψ can be counted in linear time in the input database.

1.1 Our contributions

With the goal of establishing a more practical tractability criterion for counting answers to UCQs, we explore the following two main questions in this work:

Q1) Is there a natural criterion that captures the fixed-parameter tractability of counting answers to a class of UCQs, parameterised by the size of the query?

Q2) Is there a natural criterion that captures whether counting answers to a single fixed UCQ is linear-time solvable (in the size of a given database)?

Question Q1): Fixed-Parameter Tractability. For a class C of UCQs, we consider the problem #UCQ(C) that takes as input a UCQ Ψ from C and a database $\mathcal{D}$, and asks for the number $\mathsf{ans}(\Psi \to \mathcal{D})$ of answers of Ψ in $\mathcal{D}$. We assume that the arity of the UCQs in C is bounded, that is, there is constant c such that each relation that appears in some query in C has arity at most c. As explained earlier, due to a result of Chen and Mengel [22], there is a known but rather unwieldy tractability criterion for #UCQ(C), when the problem is parameterised by the size of the query. On a high level, the number of answers of a UCQ Ψ in a given database can be expressed as a finite linear combination of CQ answer counts, using the principle of inclusion-exclusion. This means that $\mathsf{ans}(\Psi \to \mathcal{D})$ is equal to $\sum_i c_i \cdot \mathsf{ans}(\varphi_i \to \mathcal{D})$, where each φ_i is simply a conjunctive query (and not a union thereof). We refer to this linear combination as the *CQ expansion* of Ψ. Chen and Mengel showed that the parameterised complexity of computing $\mathsf{ans}(\Psi \to \mathcal{D})$ is guided by the hardest term in the respective CQ expansion. The complexity of computing these terms is simply the complexity of counting the answers of a conjunctive query, and this is well understood [21]. Hence, the main challenge for this approach is to understand the linear combination, i.e., to understand for which CQs the corresponding coefficients are non-zero. The problem is that the coefficients c_i of these linear combinations are alternating sums, which in similar settings have been observed to encode algebraic and even topological invariants [52]. This makes it highly non-trivial to determine which CQs actually contribute to the linear combination. We introduce the concepts required to state this classification informally, the corresponding definitions are given in Section 2.

We first give more details about the result of [21]. Let $\Gamma(C)$ be the class of those conjunctive queries that contribute to the CQ expansion of at least one UCQ in C, and that additionally are what we call #minimal. Intuitively, a conjunctive query φ is #minimal if there is no proper subquery φ' of φ that has the same number of answers as φ in every given database. Then the tractability criterion depends on the treewidth of the CQs in $\Gamma(C)$. It also depends on the treewidth of the corresponding class $\mathsf{contract}(\Gamma(C))$ of contracts (formally defined in Definition 2.9), which is an upper bound of what is called the "star size" in [28] and the "dominating star size" in [27]. Here is the formal statement of the known dichotomy for #UCQ(C).

Theorem 1.1 ([22]). *Let C be a recursively enumerable class of UCQs of bounded arity. If the treewidth of $\Gamma(C)$ and of $\mathsf{contract}(\Gamma(C))$ is bounded, then #UCQ(C) is fixed-parameter tractable. Otherwise, #UCQ(C) is $\mathrm{W}[1]$-hard.*

We investigate under which conditions this dichotomy can be simplified. We show that for large classes of UCQs there is actually a much more natural tractability criterion that does not rely on $\Gamma(C)$, i.e., here the computation of the coefficients of the linear combinations as well as the concept of #minimality do not play a role. We first show a simpler classification for UCQs without existential quantifiers. To state the results we require some additional definitions: The *combined query* $\bigwedge(\Psi)$ of a UCQ $\Psi(X) = \varphi_1(X) \vee \cdots \vee \varphi_\ell(X)$ is the conjunctive query obtained from Ψ by replacing each disjunction by a conjunction, that is $\bigwedge(\Psi) = \varphi_1(X) \wedge \cdots \wedge \varphi_\ell(X)$. Given a class of UCQs C, we set $\bigwedge(C) = \{\bigwedge(\Psi) \mid \Psi \in C\}$.

It will turn out that the structure of the class of combined queries $\bigwedge(C)$ determines the complexity of counting answers to UCQs in C, given that C has the following natural closure property: We say that C is *closed under deletions* if, for all $\Psi(X) = \varphi_1(X) \vee \cdots \vee \varphi_\ell(X)$ and for every $J \subseteq [\ell]$, the subquery $\bigvee_{j \in J} \varphi_j(X)$ is also contained in C. For example, any class of UCQs defined solely by the conjunctive queries admissible in the unions (such as unions of acyclic conjunctive queries) is closed under deletions. The following classification resolves the complexity of counting answers to UCQs in classes that are closed under deletions; we will see later that the closedness condition is necessary. Moreover, the tractability criterion depends solely on the structure of the combined query, and not on the terms in the CQ expansion, thus yielding, as desired, a much more concise and natural characterisation. As mentioned earlier, we first state the classification for quantifier-free UCQs.

Theorem 1.2. *Let C be recursively enumerable class of quantifier-free UCQs of bounded arity. If $\wedge(C)$ has bounded treewidth then #UCQ(C) is fixed-parameter tractable. If $\wedge(C)$ has unbounded treewidth and C is closed under deletions then #UCQ(C) is* $\mathrm{W}[1]$*-hard.*

We emphasise here that Theorem 1.2 is in terms of the simpler object $\wedge(C)$ instead of the complicated object $\Gamma(C)$.

If we allow UCQs with quantified variables in the class C then the situation becomes more intricate. Looking for a simple tractability criterion that describes the complexity of #UCQ(C) solely in terms of $\wedge(C)$ requires some additional effort. First, for a UCQ Ψ that has quantified variables, $\mathsf{contract}(\Psi)$ is not necessarily the same as Ψ, and therefore the treewidth of the contracts also plays a role. Moreover, the matching lower bound requires some conditions in addition to being closed under deletions. Nevertheless, our result is in terms of the simpler objects $\wedge(C)$ and $\mathsf{contract}(\wedge(C))$ rather than the more complicated $\Gamma(C)$ and $\mathsf{contract}(\Gamma(C))$. For Theorem 1.3, recall that a conjunctive query is *self-join-free* if each relation symbol occurs in at most one atom of the query.

Theorem 1.3. *Let C be a recursively enumerable class of UCQs of bounded arity. If $\wedge(C)$ and* $\mathsf{contract}(\wedge(C))$ *have bounded treewidth then #UCQ(C) is fixed-parameter tractable. Otherwise, if (I)–(III) are satisfed, then #UCQ(C) is* $\mathrm{W}[1]$*-hard.*

(I) C is closed under deletions.
(II) The number of existentially quantified variables of queries in C is bounded.
(III) The UCQs in C are unions of self-join-free conjunctive queries.

In the full version [31] we show that Theorem 1.3 is tight in the sense that, if any of these conditions is dropped, there are counterexamples to the claim that tractability is guided solely by $\wedge(C)$ and $\mathsf{contract}(\wedge(C))$.

Question Q2): Linear-Time Solvability. Now we turn to the question of linear-time solvability for a single fixed UCQ. The huge size of databases in modern applications motivates the question of which query problems are actually linear-time solvable. Along these lines, there is a lot of research for enumeration problems [7, 10, 11, 14, 19, 55].

The question whether counting answers to a conjunctive query φ can be achieved in time linear in the given database has been studied previously [47]. The corresponding dichotomy is well known and was discovered multiple times by different authors in different contexts.[1] In these results, the tractability criterion is whether φ is *acyclic*, i.e., whether it has a join tree (see [33]). The corresponding lower bounds are conditioned on a widely used complexity assumption from fine-grained complexity, namely the Triangle Conjecture. We define all of the complexity assumptions that we use in this work in Section 2. There we also formally define the size of a database (as the sum of the size of its signature, its universe, and its relations).

It is well-known that counting answers to quantifier-free conjunctive queries can be done in linear time if and only if the query is acyclic. The "only if" part relies on hardness assumptions from fine-grained complexity theory. Concretely, we have

[1]We remark that [10, Theorem 7] focuses on the special case of graphs and *near* linear time algorithms. However, in the word RAM model with $O(\log n)$ bits, a linear time algorithm is possible [19].

Theorem 1.4 (See Theorem 12 in [16], and [6, 7, 10]). *Let φ be a quantifier-free conjunctive query and suppose that* the Triangle Conjecture *is true. Then the number of answers of φ in a given database $\mathcal{D}$ can be computed in time linear in the size of $\mathcal{D}$ if and only if φ is acyclic.*

We note that the previous theorem is false if quantified variables were allowed as this would require the consideration of semantic acyclicity[2] (see [9]).

Theorem 1.4 yields an efficient way to check whether counting answers to a quantifier-free conjunctive query φ can be done in linear time: Just check whether φ is acyclic (in polynomial time, see for instance [33]). We investigate the corresponding question for *unions* of conjunctive queries. In stark contrast to Theorem 1.4, we show that there is no efficiently computable criterion that determines the linear-time tractability of counting answers to *unions* of conjunctive queries, unless some conjectures of fine-grained complexity theory fail.

We first observe that, as in the investigation of question Q1), one can obtain a criterion for linear-time solvability by expressing UCQ answer counts as linear combinations of CQ answer counts. Concretely, by a straightforward extension of previous results, we show that, assuming the Triangle Conjecture, a linear combination of CQ answer counts can be computed in linear time if and only if the answers to each #minimal CQ in the linear combination can be computed in linear time, that is, if each such CQ is acyclic. However, this criterion is again unwieldy in the sense that, for all we know, it may take time exponential in the size of the respective UCQ to determine whether this criterion holds.

In view of our results for question Q1) about fixed-parameter tractability, one might suspect that a more natural and simpler tractability criterion exists. However, it turns out that even under strong restrictions on the UCQs that we consider, an efficiently computable criterion is unlikely. We make this formal by studying the following meta problem.

Name: Meta
Input: A union Ψ of quantifier-free conjunctive queries.
Output: Is it possible to count answers to Ψ in time linear in the size of $\mathcal{D}$.

Restricting the input of Meta to quantifier-free queries is sensible as, without this restriction, the meta problem is known to be NP-hard even for conjunctive queries: If all variables are existentially quantified, then evaluating a conjunctive query can be done in linear time if and only if the query is semantically acyclic [55] (the "only if" relies on standard hardness assumptions). However, verifying whether a conjunctive query is semantically acyclic is already NP-hard [9]. In contrast, when restricted to quantifier-free conjunctive queries, the problem Meta is polynomially-time solvable according to Theorem 1.4.

We can now state our main result about the complexity of Meta. The hardness results hold under substantial additional input restrictions, which make these results stronger.

Theorem 1.5. *Meta can be solved in time $2^{O(\ell)} \cdot |\Psi|^{\mathrm{poly}(\log|\Psi|)}$, where ℓ is the number of conjunctive queries in the union, if the Triangle Conjecture is true. Moreover,*

- *If the Triangle Conjecture is true then Meta is* NP*-hard. If, additionally, ETH is true, then Meta cannot be solved in time $2^{o(\ell)}$.*
- *If SETH is true then Meta is* NP*-hard and cannot be solved in time $2^{o(\ell)}$.*
- *If the non-uniform ETH is true then Meta is* NP*-hard and Meta $\notin \bigcap_{\varepsilon>0} \mathrm{DTime}(2^{\varepsilon\cdot\ell})$.*

The lower bounds remain true even if Ψ is a union of self-join-free and acyclic conjunctive queries over a binary signature (that is, of arity 2).

[2]A conjunctive query is semantically acyclic if and only if its #core (Definition 2.8) is acyclic.

We make some remarks about Theorem 1.5. First, it may seem counterintuitive that the algorithmic part of this result relies on some lower bound conjectures. This is explained by the fact that an algorithmic result for META is actually a classification result for the underlying counting problem. The lower bound conjectures are the reason that the algorithm for META can answer that a linear-time algorithm is *not* possible for certain UCQs.

Second, while for counting the answers to a CQ in linear time the property of being acyclic is the right criterion, note that for unions of CQs, acyclicity is not even sufficient for tractability. Even when restricted to unions of acyclic conjunctive queries, the meta problem is NP-hard.

Third, we elaborate on the idea that we use to prove Theorem 1.5. As mentioned before, the algorithmic part of Theorem 1.5 comes from the well-known technique of expressing UCQ answer counts in terms of linear combinations of CQ answer counts, and establishing a corresponding complexity monotonicity property, see Section 2.3. The more interesting result is the hardness part. Here we discover a connection between the meta question stated in META, and a topological invariant, namely, the question whether the reduced Euler characteristic of a simplicial complex is non-zero. It is known that simplicial complexes with non-vanishing reduced Euler characteristic are evasive, and as such this property is also related to Karp's Evasiveness Conjecture (see e.g. the excellent survey of Miller [48]). We use the known fact that deciding whether the reduced Euler characteristic is vanishing is NP-hard [53]. Roughly, the reduction works as follows. Given some simplicial complex Δ, we carefully define a UCQ Ψ_Δ in such a way that only one particular term in the CQ expansion of Ψ_Δ determines the linear-time tractability of counting answers to Ψ_Δ. However, the coefficient of this term is zero precisely if the reduced Euler characteristic of Δ is vanishing.

Simplicial complexes also appeared in a related context in a work by Roth and Schmitt [52]. They show a connection between the complexity of counting induced subgraphs that fulfil some graph property and the question whether a simplicial complex associated with this graph property is non-zero. To solve their problem, it suffices to consider simplicial *graph* complexes, which are special simplicial complexes whose elements are subsets of the edges of a complete graph, and to encode these as induced subgraph counting problems. In contrast, to get our result we must encode arbitrary abstract simplicial complexes as UCQs and to show how to transfer the question about their Euler characteristic to a question about linear-time solvability of UCQs.

It turns out that, as additional consequences of our reduction in the proof of Theorem 1.5, we also obtain lower bounds for (approximately) computing the so-called Weisfeiler-Leman-dimension of a UCQ.

Consequences for the Weisfeiler-Leman-dimension of quantifier-free UCQs. During the last decade we have witnessed a resurge in the study of the *Weisfeiler-Leman-dimension* of graph classes and graph parameters [4, 8, 26, 32, 41, 49]. The Weisfeiler-Leman algorithm (WL-algorithm) and its higher-dimensional generalisations are important heuristics for graph isomorphism; for example, the 1-dimensional WL-algorithm is equivalent to the method of colour-refinement. We refer the reader to e.g. the EATCS Bulletin article of Arvind [4] for a concise and self-contained introduction; however, in this work we will use the WL-algorithm only in a black-box manner.

For each positive integer k, we say that two graphs G_1 and G_2 are *k-WL equivalent*, denoted by $G_1 \cong_k G_2$, if they cannot be distinguished by the k-dimensional WL-algorithm. A graph parameter π is called *k-WL invariant* if $G_1 \cong_k G_2$ implies $\pi(G_1) = \pi(G_2)$. Moreover, the *WL-dimension* of π is the minimum k for which π is k-WL invariant, if such a k exists, and ∞ otherwise (see e.g.[5]). The WL-dimension of a graph parameter π provides important information about the descriptive complexity of π [17]. Moreover, recent work of Morris et al. [49] shows that the WL-dimension of a graph parameter lower bounds the minimum dimension of a higher-order Graph Neural Network that computes the parameter.

The definitions of the WL-algorithm and the WL-dimension extend from graphs to labelled graphs, that is, directed multi-graphs with edge- and vertex-labels (see e.g. [44]). Formally, we say that a database is a *labelled graph* if its signature has arity at most 2, and if it contains no self-loops, that is, tuples of the form (v, v). Similarly, *(U)CQs on labelled graphs* have signatures of arity at most 2 and contain no atom of the form $R(v, v)$.

Definition 1.6 (WL-dimension). Let Ψ be a UCQ on labelled graphs. The *WL-dimension* of Ψ, denoted by $\mathsf{dim}_{\mathrm{WL}}(\Psi)$, is the minimum k such that, for any pair of labelled graphs $\mathcal{D}_1$ and $\mathcal{D}_2$ with $\mathcal{D}_1 \cong_k \mathcal{D}_2$, it holds that the number of answers to Ψ in $\mathcal{D}_1$ is the same as in $\mathcal{D}_2$. If no such k exists, then the WL-dimension is ∞.

Note that a CQ is a special case of a UCQ, so Definition 1.6 also applies when Ψ is a CQ φ.

It was shown very recently that the WL-dimension of a *quantifier-free conjunctive query* φ on labelled graphs is equal to the treewidth of the Gaifman graph of φ [44, 50]. Using known algorithms for computing the treewidth [15, 29] it follows that, for every fixed positive integer d, the problem of deciding whether the WL-dimension of φ is at most d can be solved in polynomial time (in the size of φ). Moreover, the WL-dimension of φ can be efficiently approximated in polynomial time. In stark contrast, we show that the computation of the WL-dimension of a *UCQ* is much harder; in what follows, we say that S is an *f-approximation* of k if $k \le S \le f(k) \cdot k$.

Theorem 1.7. *There exists an algorithm that computes a $O(\sqrt{\log k})$-approximation of the WL-dimension k of a quantifier-free UCQ on labelled graphs $\Psi = \varphi_1 \vee \cdots \vee \varphi_\ell$ in time $|\Psi|^{O(1)} \cdot O(2^\ell)$.*

Moreover, let $f : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ be any computable function. The problem of computing an f-approximation of $\mathsf{dim}_{\mathrm{WL}}(\Psi)$ given an input UCQ $\Psi = \varphi_1 \vee \cdots \vee \varphi_\ell$ is NP-hard, and, assuming ETH, an f-approximation of $\mathsf{dim}_{\mathrm{WL}}(\Psi)$ cannot be computed in time $2^{o(\ell)}$.

Finally, the computation of the WL-dimension of UCQs stays intractable even if we fix k.

Theorem 1.8. *Let k be any* fixed *positive integer. The problem of deciding whether the WL-dimension of a quantifier-free UCQ on labelled graphs $\Psi = \varphi_1 \vee \cdots \vee \varphi_\ell$ is at most k can be solved in time $|\Psi|^{O(1)} \cdot O(2^\ell)$.*

Moreover, the problem is NP*-hard and, assuming ETH, cannot be solved in time $2^{o(\ell)}$.*

1.2 Further Related Work

For exact counting it makes a substantial difference whether one wants to count answers to a conjunctive query or a union of conjunctive queries [22, 27]. However, for approximate counting, unions can generally be handled using a standard trick of Karp and Luby [40], and therefore fixed-parameter tractability results for approximately counting the answers to a conjunctive query also extend to unions of conjunctive queries [3, 30].

Counting and enumerating the answers to a union of conjunctive queries has also been studied in the context of dynamic databases [12, 13]. This line of research investigates the question whether linear-time dynamic algorithms are possible. Concretely, the question is whether, after a preprocessing step that builds a data structure in time linear in the size of the initial database, the number of answers to a fixed union of conjunctive queries can be returned in constant time with a constant-time update to the data structure, whenever there is a change to the database. Berkholz et al. show that for a conjunctive query such a linear-time algorithm is possible if and only if the CQ is *q-hierarchical* [12, Theorem 1.3]. There are acyclic CQs that are not q-hierarchical, for instance the query $\varphi(\{a, b, c, d\}) = E(a, b) \wedge E(b, c) \wedge E(c, d)$ is clearly acyclic — however, the sets of atoms that contain b and c, respectively, are neither comparable nor disjoint, and therefore φ is not q-hierarchical. So, there are queries for which counting in the static setting is easy, whereas it is hard in the dynamic setting. Berkholz et al. extend their result from CQs to UCQs [13, Theorem 4.5], where the criterion is whether the UCQ is *exhaustively q-hierarchical.* This property essentially means that, for every subset of the CQs in the union, if instead of taking the disjunction of these CQs we take the conjunction, then the resulting CQ should be q-hierarchical. Moreover, checking whether a CQ ϕ is q-hierarchical can be done in time polynomial in the size of ϕ. However, the straightforward approach of checking whether a UCQ is exhaustively q-hierarchical takes exponential time, and it is stated as an open problem in [13] whether this can be improved. In the dynamic setting this question remains open — however, in the static setting we show that, while for counting answers to CQs the criterion for linear-time tractability can be verified in polynomial time, this is not true for unions of conjunctive queries, subject to some complexity assumptions, as we have seen in Theorem 1.5.

1.3 Organisation of the Paper

The subsequent Section 2 introduces some preliminary material, and in Section 3 we prove the complexity classification of #UCQ(C) for deletion-closed classes C. The treatment of Meta and its connection to the WL-dimension is presented in the full version [31].

2 PRELIMINARIES

Due to the fine-grained nature of the questions we ask in this work (e.g. linear time counting vs non-linear time counting), it is important to specify the machine model. We use the standard word RAM model with $O(\log n)$ bits. The exact model makes a difference. For example, it is possible to count answers to quantifier-free acyclic conjunctive queries in linear time in the word RAM model [19], while Turing machines only achieve near linear time (or expected linear time) [10].

Due to the space constraints, we defer the introduction of some background material on parameterised complexity theory and relational databases to the full version.

2.1 Fine-grained Complexity Theory

In this work, we will rely on the following hypotheses from fine-grained complexity theory.

Conjecture 2.1 (ETH [39]). *3-SAT cannot be solved in time* $\exp(o(n))$*, where n denotes the number of variables of the input formula.*

Conjecture 2.2 (SETH [18, 39]). *For each* $\varepsilon > 0$ *there exists a positive integer* k *such that* k*-SAT cannot be solved in time* $O(2^{(1-\varepsilon)n})$*, where* n *denotes the number of variables of the input formula.*

Conjecture 2.3 (Non-uniform ETH [23]). *3-SAT is not contained in* $\bigcap_{\varepsilon>0} \mathrm{DTime}(\exp(\varepsilon n))$*, where* n *denotes the number of variables of the input formula.*

Conjecture 2.4 (Triangle Conjecture [1]). *There exists* $\gamma > 0$ *such that any (randomised) algorithm that decides whether a graph with* n *vertices and* m *edges contains a triangle takes time at least* $\Omega(m^{1+\gamma})$ *in expectation.*

2.2 Homomorphisms and Conjunctive Queries

We assume familiarity with the central notions of relational databases such as signatures, structures, and Gaifman graphs. We refer the reader to the full version [31] for a brief introduction.

Homomorphisms as Answers to CQs. Let $\mathcal{A}$ and $\mathcal{B}$ be structures over signatures $\tau_{\mathcal{A}} \subseteq \tau_{\mathcal{B}}$. A *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$ is a mapping $h : U(\mathcal{A}) \to U(\mathcal{B})$ such that for each relation symbol

$R \in \tau_{\mathcal{A}}$ with arity a and each tuple $\vec{t} = (t_1, \dots, t_a) \in R^{\mathcal{A}}$ we have that $h(\vec{t}) = (h(t_1), \dots, h(t_a)) \in R^{\mathcal{B}}$. We use $\mathsf{Hom}(\mathcal{A} \to \mathcal{B})$ to denote the set of homomorphisms from $\mathcal{A}$ to $\mathcal{B}$, and we use the lower case version $\mathsf{hom}(\mathcal{A} \to \mathcal{B})$ to denote the *number* of homomorphisms from $\mathcal{A}$ to $\mathcal{B}$.

Let φ be a conjunctive query with free variables $X = \{x_1, \dots, x_k\}$ and quantified variables $Y = \{y_1, \dots, y_d\}$. We can associate φ with a structure $\mathcal{A}_\varphi$ defined as follows: The universe of $\mathcal{A}_\varphi$ are the variables $X \cup Y$ and for each atom $R(\vec{t})$ of φ we add the tuple $\vec{t}$ to $R^{\mathcal{A}}$. It is well-known that, for each database $\mathcal{D}$, the set of answers of φ in $\mathcal{D}$ is precisely the set of assignments $a : X \to U(\mathcal{D})$ such that there is a homomorphism $h \in \mathsf{Hom}(\mathcal{A}_\varphi \to \mathcal{D})$ with $h|_X = a$. Since working with (partial) homomorphisms will be very convenient in this work, we will use the notation from [27] and (re)define a conjunctive query as a pair consisting of a relational structure $\mathcal{A}$ together with a set $X \subseteq U(\mathcal{A})$. The size of $(\mathcal{A}, X)$ is denoted by $|(\mathcal{A}, X)|$ and defined to be $|\mathcal{A}| + |X|$. Furthermore, the set of answers of $(\mathcal{A}, X)$ in $\mathcal{D}$, denoted by $\mathsf{Ans}((\mathcal{A}, X) \to \mathcal{D})$, is defined as $\{a : X \to U(\mathcal{D}) \mid \exists h \in \mathsf{Hom}(\mathcal{A} \to \mathcal{D}) : h|_X = a\}$. We then use $\mathsf{ans}((\mathcal{A}, X) \to \mathcal{D})$ to denote the number of answers, i.e., $\mathsf{ans}((\mathcal{A}, X) \to \mathcal{D}) := |\mathsf{Ans}((\mathcal{A}, X) \to \mathcal{D})|$.

We can now formally define the (parameterised) problem of counting answers to conjunctive queries. As is usual, we restrict the problem by a class C of allowed queries.

Name: #CQ(C)
Input: A conjunctive query $(\mathcal{A}, X) \in C$ and a database $\mathcal{D}$.
Parameter: $|(\mathcal{A}, X)|$.
Output: The number of answers $\mathsf{ans}((\mathcal{A}, X) \to \mathcal{D})$.

#Equivalence and #Minimality. In the realm of decision problems, it is well known that evaluating a conjunctive query is equivalent to evaluating the (homomorphic) core of the query, i.e., evaluating the minimal homomorphic-equivalent query. A similar, albeit slightly different notion of equivalence and minimality is required for counting answers to conjunctive queries. In what follows, we will provide the necessary definitions and properties of equivalence, minimality and cores for counting answers to conjunctive queries, and we refer the reader to [22] and to the full version of [27] for a more comprehensive discussion. To avoid confusion between the notions in the realms of decision and counting, we will from now on use the # symbol for the counting versions (see Definition 2.6).

Definition 2.5. Two conjunctive queries $(\mathcal{A}, X)$ and $(\mathcal{A}', X')$ are *isomorphic*, denoted by $(\mathcal{A}, X) \cong (\mathcal{A}', X')$, if there is an isomorphism b from $\mathcal{A}$ to $\mathcal{A}'$ with $b(X) = X'$.

Definition 2.6 (#Equivalence and #minimality (see [22, 27])). Two conjunctive queries $(\mathcal{A}, X)$ and $(\mathcal{A}', X')$ are *#equivalent*, denoted by $(\mathcal{A}, X) \sim (\mathcal{A}', X')$, if for every database $\mathcal{D}$ we have $\mathsf{ans}((\mathcal{A}, X) \to \mathcal{D}) = \mathsf{ans}((\mathcal{A}', X') \to \mathcal{D})$. A conjunctive query $(\mathcal{A}, X)$ is *#minimal* if there is no proper substructure $\mathcal{A}'$ of $\mathcal{A}$ such that $(\mathcal{A}, X) \sim (\mathcal{A}', X)$.

Observation 2.7. *The following are equivalent:*

(1) A conjunctive query $(\mathcal{A}, X)$ is #minimal.
(2) $(\mathcal{A}, X)$ has no #equivalent substructure that is induced by a set U with $X \subseteq U \subset U(\mathcal{A})$.
(3) Every homomorphism from $\mathcal{A}$ to itself that is the identity on X is surjective.

It turns out that #equivalence is the same as isomorphism if all variables are free, and it is the same as homomorphic equivalence if all variables are existentially quantified (see e.g. the discussion in Section 5 in the full version of [27]). Moreover, each quantifier-free conjunctive query is #minimal.

Definition 2.8 (#core). A *#core* of a conjunctive query $(\mathcal{A}, X)$ is a #minimal conjunctive query $(\mathcal{A}', X')$ with $(\mathcal{A}, X) \sim (\mathcal{A}', X')$.

It is well known (see e.g. [27]) that,for #minimal queries, #equivalence and isomorphism coincide. Thus the #core is unique up to isomorphisms; in fact, this allows us to speak of “the” #core of a conjunctive query.

Classification of #CQ(C) via Treewidth and Contracts. It is well known that the complexity of counting answers to a conjunctive query is governed by its treewidth, and by the treewidth of its contract [21, 27], which we define as follows.

Definition 2.9 (Contract). Let $(\mathcal{A}, X)$ be a conjunctive query, let $Y = U(\mathcal{A}) \setminus X$, and let G be the Gaifmann graph of $\mathcal{A}$. The *contract* of $(\mathcal{A}, X)$, denoted by $\mathsf{contract}(\mathcal{A}, X)$ is obtained from $G[X]$ by adding an edge between each pair of vertices u and v for which there is a connected component S in $G[Y]$ that is adjacent to both u and v, that is, there are vertices $x, y \in S$ such that $\{x, u\} \in E(G)$ and $\{y, v\} \in E(G)$. Given a class of conjunctive queries C, we write $\mathsf{contract}(C)$ for the class of all contracts of queries in C.

We note that there are multiple equivalent ways to define the contract of a query. For our purposes, the definition in [27] is most suitable. Also, the treewidth of the contract of a conjunctive query is an upper bound of what is called the query’s “star size” in [28] and its “dominating star size” in [27].

Chen and Mengel established the following classification for counting answers to conjunctive queries of bounded arity.

Theorem 2.10 ([21]). *Let C be a recursively enumerable class of conjunctive queries of bounded arity, and let C' be the class of #cores of queries in C. If the treewidth of C' and of $\mathsf{contract}(C')$ is bounded, then #CQ(C) is solvable in polynomial time. Otherwise, #CQ(C) is* W[1]*-hard.*

We point out that the W[1]-hard cases can further be partitioned into W[1]-complete, #W[1]-complete and even harder cases [21, 27].[3] However, for the purpose of this work, we are only interested in tractable and intractable cases (recall that W[1]-hard problems are not fixed-parameter tractable under standard assumptions from fine-grained and parameterised complexity theory, such as ETH).

Self-join-free Conjunctive Queries and Isolated Variables. A conjunctive query $(\mathcal{A}, X)$ is *self-join-free* if each relation of $\mathcal{A}$ contains at most one tuple. We say that a variable of a conjunctive query is *isolated* if it is not part of any relation.

Note that that adding/removing isolated variables to/from a conjunctive query does not change its treewidth or the treewidth of its #core. Further, it does not change the complexity of counting answers: Just multiply/divide by n^v, where n is the number of elements of the database and v it the number of added/removed isolated free variables. For this reason, we will allow ourselves in this work to freely add and remove isolated variables from the queries that we encounter. For the existence of homomorphisms we also observe the following.

Observation 2.11. *Let $(\mathcal{A}, X)$ be a conjunctive query, let X' be a superset of X and let $\mathcal{A}'$ be the structure obtained from $\mathcal{A}$ by adding an isolated variable for each $x \in X' \setminus X$. Then for all $a : X' \to U(\mathcal{D})$ we have that $a|_X \in \mathsf{Ans}((\mathcal{A}, X) \to \mathcal{D})$ iff $a \in \mathsf{Ans}((\mathcal{A}', X') \to \mathcal{D})$.*

2.3 UCQs and the Homomorphism Basis

A *union of conjunctive queries* (UCQ) Ψ is a tuple of structures $(\mathcal{A}_1, \dots, \mathcal{A}_{\ell(\Psi)})$ over the same signature together with a set of designated elements X (the free variables) that are in the universe of each of the structures. For each $J \subseteq [\ell(\Psi)]$, we define $\Psi|_J = ((A_j)_{j \in J}, X)$. If we restrict to a

[3]Those cases are: #W[2]-hard and #A[2]-complete.

single term of the union then we usually just write Ψ_i instead of $\Psi|_{\{i\}}$. Note that $\Psi_i = (\mathcal{A}_i, X)$ is simply a conjunctive query (rather than a union of CQs).

We will assume (without loss of generality) that, for any distinct i and i' in $[\ell(\Psi)]$, $U(\mathcal{A}_i) \cap U(\mathcal{A}_{i'}) = X$, i.e., that each CQ in the union has its own set of existentially quantified variables.

If each such conjunctive query is acyclic we say that Ψ is a union of acyclic conjunctive queries. Moreover, the arity of Ψ is the maximum arity of any of the $\mathcal{A}_i$. The size of Ψ is $|\Psi| = \sum_{i=1}^{\ell(\Psi)} |\Psi_i|$. The elements of X are the *free variables* of Ψ and $\ell(\Psi)$ is the number of CQs in the union. The set of *answers* of Ψ in a database $\mathcal{D}$, denoted by $\mathsf{Ans}(\Psi \to \mathcal{D})$ is defined as follows:

$$\mathsf{Ans}(\Psi \to \mathcal{D}) = \{a : X \to U(\mathcal{D}) \mid \exists i \in [\ell] : a \in \mathsf{Ans}(\Psi_i \to \mathcal{D})\}\,.$$

Again, we use the lower case version $\mathsf{ans}(\Psi \to \mathcal{D})$ to denote the *number* of answers of Ψ in $\mathcal{D}$.

In the definition of UCQs we assume that every CQ in the union has the same set of free variables, namely X. This assumption is without loss of generality. To see this, suppose that we have a union of CQs $(\mathcal{A}_1, X_1), \dots, (\mathcal{A}_\ell, X_\ell)$ with individual sets of free variables. Let $X = \bigcup_{i=1}^{\ell} X_i$ and, for each $i \in [\ell]$, let $\mathcal{A}'_i$ be the structure obtained from $\mathcal{A}_i$ by adding an isolated variable for each $x \in X \setminus X_i$. Then consider the UCQ $\Psi := ((\mathcal{A}'_1, \dots, \mathcal{A}'_\ell), X)$. If for some assignment $a : X \to U(\mathcal{D})$ it holds that there is an $i \in [\ell]$ such that $a|_{X_i} \in \mathsf{Ans}((\mathcal{A}_i, X_i) \to \mathcal{D})$. Then, according to Observation 2.11, this is equivalent to $a \in \mathsf{Ans}((\mathcal{A}'_i, X) \to \mathcal{D})$, which means that a is an answer of Ψ. So, without loss of generality we can work with Ψ, which uses the same set of free variables for each CQ in the union.

Now we define the parameterised problem of counting answers to UCQs. As usual, the problem is restricted by a class C of allowed queries with respect to which we classify the complexity.

Name: #UCQ(C)
Input: A UCQ $\Psi \in C$ together with a database $\mathcal{D}$.
Parameter: $|\Psi|$.
Output: The number of answers $\mathsf{ans}(\Psi \to \mathcal{D})$.

The next definition will be crucial for the analysis of the complexity of #UCQ(C).

Definition 2.12 (combined query $\wedge(\Psi)$). Let $\Psi = ((\mathcal{A}_1, , \dots, \mathcal{A}_\ell), X)$ be a UCQ. Then we define the *combined query* $\wedge(\Psi) = (\bigcup_{j\in[\ell]} A_j, X)$.

What follows is an easy, but crucial observation about $\wedge\left(\Psi|_J\right)$.

Observation 2.13. *Let $((\mathcal{A}_1, , \dots, \mathcal{A}_\ell), X)$ be a UCQ, and let $\emptyset \neq J \subseteq [\ell]$. For each database $\mathcal{D}$ and assignment $a : X \to U(\mathcal{D})$ we have*

$$a \in \mathsf{Ans}(\wedge\left(\Psi|_J\right) \to \mathcal{D}) \Leftrightarrow \forall j \in J : a \in \mathsf{Ans}(\Psi_j \to \mathcal{D}).$$

Definition 2.14 (Coefficient function c_Ψ). Let $\Psi = ((\mathcal{A}_1, , \dots, \mathcal{A}_\ell), X)$ be a UCQ. For each conjunctive query $(\mathcal{A}, X)$, we set $\mathcal{I}(\mathcal{A}, X) = \{J \subseteq [\ell] \mid (\mathcal{A}, X) \sim \wedge\left(\Psi|_J\right)\}$, and we define the *coefficient function* of Ψ as follows: $c_\Psi(\mathcal{A}, X) = \sum_{J\in\mathcal{I}(\mathcal{A},X)} (-1)^{|J|+1}$.

Using inclusion-exclusion, we can transform the problem of counting answers to Ψ into the problem of evaluating a linear combination of CQ answer counts. We include a proof in the full version [31] only for reasons of self-containment and note that the complexity-theoretic applications of this transformation, especially regarding lower bounds, have first been discovered by Chen and Mengel [22].

Lemma 2.15 ([22]). *Let $\Psi = ((\mathcal{A}_1, , \dots, \mathcal{A}_\ell), X)$ be a UCQ. For every database $\mathcal{D}$, $\mathsf{ans}(\Psi \to \mathcal{D}) = \sum_{(\mathcal{A},X)} c_\Psi(\mathcal{A}, X) \cdot \mathsf{ans}((\mathcal{A}, X) \to \mathcal{D})$, where the sum is over all equivalence classes of $\sim$.*

We conclude this subsection with the following two operations on classes of UCQs.

Definition 2.16 ($\Gamma(C)$ *and* $\bigwedge(C)$). Let C be a class of UCQs. $\Gamma(C)$ is the class of all $(\mathcal{A}, X)$ such that $(\mathcal{A}, X)$ is #minimal and there is $\Psi \in C$ with $c_\Psi(\mathcal{A}, X) \neq 0$. Let $\bigwedge(C) = \{\bigwedge(\Psi) \mid \Psi \in C\}$.

It was independently discovered by Chen and Mengel [22], and by Curticapean, Dell and Marx [24] that the computation of a linear combination of homomorphism counts is precisely as hard as computing its hardest term. Moreover, in the former work, Chen and Mengel also established this property in the more general context of linear combinations of conjunctive queries. Applying this principle to counting answers to UCQs (which we have seen to be equivalent to computing a linear combination in Lemma 2.15), we obtain the following two results; details are provided in the full version [31].

Corollary 2.17. *Let Ψ be a UCQ. For each $d \geq 1$, computing the function $\mathcal{D} \mapsto \mathsf{ans}(\Psi \to \mathcal{D})$ can be done in time $O(|\mathcal{D}|^d)$ if and only if for each #minimal $(\mathcal{A}, X)$ with $c_\Psi(\mathcal{A}, X) \neq 0$ the function $\mathcal{D} \mapsto \mathsf{ans}((\mathcal{A}, X) \to \mathcal{D})$ can be computed in time $O(|\mathcal{D}|^d)$.*

Corollary 2.18 (Implicitly also in [22]). *Let C be a recursively enumerable class of UCQs. The problems #UCQ(C) and #CQ($\Gamma(C)$) are interreducible with respect to parameterised Turing-reductions.*

3 PROOFS OF THEOREM 1.2 AND THEOREM 1.3

Let C be a class of UCQs. Recall that $\bigwedge(C)$ is the class of all conjunctive queries that are obtained just by substituting all $\vee$ by $\wedge$ in UCQs in C, whereas $\Gamma(C)$ in Theorem 1.1 is the much less natural class of #minimal queries that survive with a non-zero coefficient in the CQ expansion of a UCQ in C. The work of Chen and Mengel [22] implicitly also shows an upper bound for counting answers to UCQs from the class C in terms of the simpler objects $\bigwedge(C)$ and $\mathsf{contract}(\bigwedge(C))$, rather than in terms of the more complicated objects $\Gamma(C)$ and $\mathsf{contract}(\Gamma(C))$. We include a proof for completeness.

Lemma 3.1. *Let C be a recursively enumerable class of UCQs. Suppose that both $\bigwedge(C)$ and $\mathsf{contract}(\bigwedge(C))$ have bounded treewidth. Then #UCQ(C) is fixed-parameter tractable.*

Proof. Let $\Psi \in C$. Recall from the proof of Lemma 2.15 that, for every $\mathcal{D}$, $\mathsf{ans}(\Psi \to \mathcal{D}) = \sum_{\emptyset \neq J \subseteq [\ell]} (-1)^{|J|+1} \cdot \mathsf{hom}(\bigwedge(\Psi|_J) \to \mathcal{D})$. Hence #UCQ($C$) $\leq^{\mathsf{FPT}}$ #CQ($\hat{C}$) where $\hat{C}$ is $\{\bigwedge(\Psi|_J) \mid \Psi \in C \wedge \emptyset \neq J \subseteq [\ell(\Psi)]\}$. Finally, since $\bigwedge(\Psi|_J)$ is a subquery of $\bigwedge(\Psi)$ for each J, the treewidths of $\bigwedge(\Psi|_J)$ and $\mathsf{contract}(\bigwedge(\Psi|_J))$ are upper-bounded by the treewidths of $\bigwedge(\Psi)$ and $\mathsf{contract}(\bigwedge(\Psi))$, respectively. Consequently, the treewidths of $\hat{C}$ and $\mathsf{contract}(\hat{C})$ are bounded, and thus #CQ($\hat{C}$) is polynomial-time solvable by the classification of Chen and Mengel [21, Theorem 22] Since #UCQ(C) $\leq^{\mathsf{FPT}}$ #CQ($\hat{C}$), the lemma follows. □

Our goal is to relate the complexity of #UCQ(C) to the structure of $\bigwedge(C)$ with the hope of obtaining a more natural tractability criterion than the one given by Theorem 1.1. While we show in the full version [31] that this seems not always possible, we identify conditions under which a natural criterion based on $\bigwedge(C)$ is possible, both in the quantifier-free case (Section 3.1), and in the general case that allows quantified variables (Section 3.2).

A class of UCQs C is *closed under deletions* if, for every $\Psi = ((\mathcal{A}_1, \dots, \mathcal{A}_\ell), X) \in C$ and for every $\emptyset \neq J \subseteq [\ell]$, the UCQ $\Psi|_J$ is also contained in C. For example, any class of UCQs defined by prescribing the allowed conjunctive queries is closed under deletions. This includes, e.g., unions of acyclic conjunctive queries.

3.1 The Quantifier-free Case

As a warm-up, we start with the much simpler case of quantifier-free queries. Here, we only allow (unions of) conjunctive queries $(\mathcal{A}, X)$ satisfying $U(\mathcal{A}) = X$.

Lemma 3.2. *Let C be a recursively enumerable class of quantifier-free UCQs of bounded arity. Suppose that C is closed under deletions. If $\wedge(C)$ has unbounded treewidth then #UCQ(C) is* W[1]*-hard.*

Proof. We show that $\wedge(C) \subseteq \Gamma(C)$, which then proves the claim by Theorem 1.1. Recall from Definition 2.16 that $\Gamma(C) = \{(\mathcal{A}, X) \mid (\mathcal{A}, X)$ is #minimal and there is $\Psi \in C$ with $c_\Psi(\mathcal{A}, X) \neq 0\}$. Let $\Psi = ((\mathcal{A}_1, \dots, \mathcal{A}_\ell), X) \in C$. Note that, according to Observation 2.7, $\wedge(\Psi)$ is its own #core since it does not have existentially quantified variables. For the same reason, for each nonempty subset J of $[\ell]$, the query $\wedge(\Psi|_J)$ is its own #core. Now let $J \subseteq [\ell]$ be inclusion-minimal with the property that $\wedge(\Psi|_J)$ is isomorphic to $\wedge(\Psi)$. Since C is closed under deletions, the UCQ $\Psi|_J$ is contained in C. By the inclusion-minimality of J, Definition 2.14 ensures that $c_{\Psi|_J}(\wedge(\Psi)) = (-1)^{|J|+1} \neq 0$. As a consequence, $\wedge(\Psi) \in \Gamma(C)$, concluding the proof. □

From Lemmas 3.2 and 3.1 together with the fact that the contract of a quantifier-free query is the query itself, we obtain Theorem 1.2.

3.2 The General Case

Now we consider UCQs with existentially quantified variables. Here, a corresponding hardness result (Lemma 3.5) can be achieved under some additional assumptions. Note that the number of existentially quantified variables in a UCQ $\Psi = ((\mathcal{A}_1, \dots, \mathcal{A}_\ell), X)$ is equal to $\sum_{i=1}^{\ell} |U(\mathcal{A}_i) \setminus X|$. We first need the following two auxiliary results:

Lemma 3.3. *Let $(\mathcal{A}, X)$ and $(\mathcal{A}', X')$ be #equivalent conjunctive queries. Further, let G and G' be the Gaifman graphs of $\mathcal{A}$ and $\mathcal{A}'$, respectively. Then $G[X]$ and $G'[X']$ are isomorphic.*

Lemma 3.4. *Let $(\mathcal{A}, X)$ be a self-join-free conjunctive query. Let $\mathcal{A}'$ be the structure obtained from $\mathcal{A}$ by deleting all isolated variables in $U(\mathcal{A}) \setminus X$. Then $(\mathcal{A}', X)$ is the #core of $(\mathcal{A}, X)$.*

Lemma 3.5. *Let C be a recursively enumerable class of unions of self-join-free conjunctive queries with bounded arity. Suppose that C is closed under deletions and that there is a finite upper bound on the number of existentially quantified variables in queries in C. If either of $\wedge(C)$ or* contract$(\wedge(C))$ *have unbounded treewidth then #UCQ(C) is* W[1]*-hard.*

Proof. Let d be the maximum number of existentially quantified variables in a query in C. Assume first that $\wedge(C)$ has unbounded treewidth. We show that $\Gamma(C)$ has unbounded treewidth, which proves the claim by Theorem 1.1. To this end, let B be any positive integer. The goal is to find a conjunctive query in $\Gamma(C)$ with treewidth at least B. Since $\wedge(C)$ has unbounded treewidth, there is a UCQ $\Psi = ((\mathcal{A}_1, \dots, \mathcal{A}_\ell), X)$ in C such that $\wedge(\Psi)$ has treewidth larger than $d + B$. Note that, although all Ψ_i are self-join-free, $\wedge(\Psi)$ is not necessarily self-join-free. Let J be inclusion-minimal among the subsets of $[\ell]$ with the property that the #core of $\wedge(\Psi|_J)$ is isomorphic to the #core of $\wedge(\Psi)$. Since C is closed under deletions, the UCQ $\Psi|_J$ is contained in C. Let $(\mathcal{A}', X')$ be the #core of $\wedge(\Psi)$.

By inclusion-minimality of J, $c_{\Psi|_J}((\mathcal{A}', X')) = (-1)^{|J|+1} \neq 0$. As a consequence, $(\mathcal{A}', X') \in \Gamma(C)$. It remains to show that the treewidth of $(\mathcal{A}', X')$ is at least B. For this, let G be the Gaifmann graph of $\wedge(\Psi)$ and let G' be the Gaifmann graph of the #core of $\wedge(\Psi)$ (the Gaifman graph of $\mathcal{A}'$). First, deletion of a vertex can decrease the treewidth by at most 1. Thus, $G[X]$ has treewidth at least $d + B - d = B$. By Lemma 3.3, $G[X]$ and $G'[X']$ are isomorphic.

Therefore the treewidth of $G'[X']$, i.e., the treewidth of $(\mathcal{A}', X')$, is at least B. So we have shown that if the treewidth of $\wedge(C)$ is unbounded then so is the treewidth of $\Gamma(C)$.

In the second case, we assume that the contracts of queries in $\wedge(C)$ (see Definition 2.9) have unbounded treewidth. We introduce the following terminology: Let $(\mathcal{A}, X)$ be a conjunctive query and let $y \in U(\mathcal{A})$. The *degree of freedom* of y is the number of vertices in X that are adjacent to

y in the Gaifman graph of $\mathcal{A}$. Let $\hat{C}$ be the class of all conjunctive queries $(\mathcal{A}, X)$ such that there exists $\Psi = ((\mathcal{A}_1, \dots, \mathcal{A}_{\ell(\Psi)}), X)$ in C with $(\mathcal{A}, X) = (\mathcal{A}_i, X)$ for some $i \in [\ell(\Psi)]$. Since C is closed under deletions, $\hat{C} \subseteq C$. By the assumptions of the lemma, $\hat{C}$ consists only of self-join-free queries. Thus, by Lemma 3.4, each query in $\hat{C}$ is its own #core (up to deleting isolated variables). We will now consider the following cases:

(i) Suppose that $\hat{C}$ has unbounded degree of freedom. With Definition 2.9 it is straightforward to check that a quantified variable y with degree of freedom B induces a clique of size B in the contract of the corresponding query. Therefore, the contracts of the queries in $\hat{C}$ have unbounded treewidth. Consequently, $\#\mathrm{CQ}(\hat{C})$ is $\mathrm{W}[1]$-hard by the classification of Chen and Mengel [21, Theorem 22]. Since $\hat{C} \subseteq C$ the problem $\#\mathrm{CQ}(\hat{C})$ is merely a restriction of $\#\mathrm{UCQ}(C)$, the latter of which is thus $\mathrm{W}[1]$-hard as well.

(ii) Suppose that the degree of freedom of queries in $\hat{C}$ is bounded by a constant d'. We show that $\Gamma(C)$ has unbounded treewidth, which proves the claim by Theorem 1.1. To this end, let B be any positive integer. The goal is to find a conjunctive query in $\Gamma(C)$ with treewidth at least B. Since $\mathsf{contract}(\bigwedge(C))$ has unbounded treewidth, there is a UCQ $\Psi = (\mathcal{A}_1, \dots, \mathcal{A}_\ell), X)$ in C such that $\mathsf{contract}(\bigwedge(\Psi))$ has treewidth larger than $d + \binom{dd'}{2} + B$. We will show that $\bigwedge(\Psi)$ has treewidth larger than $d + B$, which, as we have argued previously, implies that $\Gamma(C)$ contains a query with treewidth at least B. To prove that $\bigwedge(\Psi)$ indeed has treewidth larger than $d+B$, let $\bigwedge(\Psi) = (\mathcal{A}, X)$ and let G be the Gaifman graph of $\mathcal{A}$. Let $Y = U(\mathcal{A}) \setminus X$ and recall from Definition 2.9 that $\mathsf{contract}(\mathcal{A}, X)$ is obtained from $G[X]$ by adding an edge between any pair of vertices u and v that are adjacent to a common connected component in $G[Y]$. Let $N \subseteq X$ be the set of all vertices in X that are adjacent to a vertex in Y and observe that $|N| \leq dd'$ since the number of existentially quantified variables and the degree of freedom are bounded by d and d', respectively. Thus, $\mathsf{contract}(\mathcal{A}, X)$ is obtained by adding at most $\binom{dd'}{2}$ edges to $G[X]$. The deletion of an edge can decrease the treewidth by at most 1, so $\mathsf{tw}(\bigwedge(\Psi)) = \mathsf{tw}(G) \geq \mathsf{tw}(G[X]) \geq \mathsf{tw}(\mathsf{contract}(\mathcal{A}, X)) - \binom{dd'}{2} > d + B$, which concludes Case (ii).

With all cases concluded, the proof is completed. □

From Lemmas 3.1 and 3.5 we directly obtain Theorem 1.3.

Remark 3.6. It turns out that all side conditions of Theorem 1.3 are necessary if we aim to classify $\#\mathrm{UCQ}(C)$ solely via $\bigwedge(C)$. To this end, we provide counter examples for each missing condition in the full version [31].

REFERENCES

[1] Amir Abboud and Virginia Vassilevska Williams. 2014. Popular Conjectures Imply Strong Lower Bounds for Dynamic Problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 434–443. https://doi.org/10.1109/FOCS.2014.53

[2] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases*. Addison-Wesley. http://webdam.inria.fr/Alice/

[3] Marcelo Arenas, Luis Alberto Croquevielle, Rajesh Jayaram, and Cristian Riveros. 2021. When is approximate counting for conjunctive queries tractable?. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, Samir Khuller and Virginia Vassilevska Williams (Eds.). ACM, 1015–1027. https://doi.org/10.1145/3406325.3451014

[4] Vikraman Arvind. 2016. The Weisfeiler-Lehman Procedure. *Bull. EATCS* 120 (2016). http://eatcs.org/beatcs/index.php/beatcs/article/view/442

[5] Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. 2022. On the Weisfeiler-Leman dimension of fractional packing. *Inf. Comput.* 288 (2022), 104803. https://doi.org/10.1016/j.ic.2021.104803

[6] Guillaume Bagan. 2009. *Algorithmes et complexité des problèmes d'énumération pour l'évaluation de requêtes logiques. (Algorithms and complexity of enumeration problems for the evaluation of logical queries).* Ph. D. Dissertation. University of Caen Normandy, France. https://tel.archives-ouvertes.fr/tel-00424232

[7] Guillaume Bagan, Arnaud Durand, and Etienne Grandjean. 2007. On Acyclic Conjunctive Queries and Constant Delay Enumeration. In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4646)*, Jacques Duparc and Thomas A. Henzinger (Eds.). Springer, 208–222. https://doi.org/10.1007/978-3-540-74915-8_18

[8] Pablo Barceló, Mikhail Galkin, Christopher Morris, and Miguel A. Romero Orth. 2022. Weisfeiler and Leman Go Relational. In *Learning on Graphs Conference, LoG 2022, 9-12 December 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 198)*, Bastian Rieck and Razvan Pascanu (Eds.). PMLR, 46. https://proceedings.mlr.press/v198/barcelo22a.html

[9] Pablo Barceló, Miguel Romero, and Moshe Y. Vardi. 2016. Semantic Acyclicity on Graph Databases. *SIAM J. Comput.* 45, 4 (2016), 1339–1376. https://doi.org/10.1137/15M1034714

[10] Suman K. Bera, Lior Gishboliner, Yevgeny Levanzov, C. Seshadhri, and Asaf Shapira. 2022. Counting Subgraphs in Degenerate Graphs. *J. ACM* 69, 3 (2022), 23:1–23:21. https://doi.org/10.1145/3520240

[11] Christoph Berkholz, Fabian Gerhardt, and Nicole Schweikardt. 2020. Constant delay enumeration for conjunctive queries: a tutorial. *ACM SIGLOG News* 7, 1 (2020), 4–33. https://doi.org/10.1145/3385634.3385636

[12] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. 2017. Answering Conjunctive Queries under Updates. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts (Eds.). ACM, 303–318. https://doi.org/10.1145/3034786.3034789

[13] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. 2018. Answering UCQs under Updates and in the Presence of Integrity Constraints. In *21st International Conference on Database Theory, ICDT 2018, March 26-29, 2018, Vienna, Austria (LIPIcs, Vol. 98)*, Benny Kimelfeld and Yael Amsterdamer (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 8:1–8:19. https://doi.org/10.4230/LIPIcs.ICDT.2018.8

[14] Christoph Berkholz and Nicole Schweikardt. 2019. Constant Delay Enumeration with FPT-Preprocessing for Conjunctive Queries of Bounded Submodular Width. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany (LIPIcs, Vol. 138)*, Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 58:1–58:15. https://doi.org/10.4230/LIPIcs.MFCS.2019.58

[15] Hans L. Bodlaender. 1996. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth. *SIAM J. Comput.* 25, 6 (1996), 1305–1317. https://doi.org/10.1137/S0097539793251219

[16] Johann Brault-Baron. 2013. *De la pertinence de l'énumération : complexité en logiques propositionnelle et du premier ordre. (The relevance of the list: propositional logic and complexity of the first order).* Ph. D. Dissertation. University of Caen Normandy, France. https://tel.archives-ouvertes.fr/tel-01081392

[17] Jin-yi Cai, Martin Fürer, and Neil Immerman. 1992. An optimal lower bound on the number of variables for graph identification. *Comb.* 12, 4 (1992), 389–410. https://doi.org/10.1007/BF01305232

[18] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. 2009. The Complexity of Satisfiability of Small Depth Circuits. In *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 5917)*, Jianer Chen and Fedor V. Fomin (Eds.). Springer, 75–85. https://doi.org/10.1007/978-3-642-11269-0_6

[19] Nofar Carmeli, Shai Zeevi, Christoph Berkholz, Alessio Conte, Benny Kimelfeld, and Nicole Schweikardt. 2022. Answering (Unions of) Conjunctive Queries using Random Access and Random-Order Enumeration. *ACM Trans. Database Syst.* 47, 3 (2022), 9:1–9:49. https://doi.org/10.1145/3531055

[20] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison (Eds.). ACM, 77–90. https://doi.org/10.1145/800105.803397

[21] Hubie Chen and Stefan Mengel. 2015. A Trichotomy in the Complexity of Counting Answers to Conjunctive Queries. In *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium (LIPIcs, Vol. 31)*, Marcelo Arenas and Martín Ugarte (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 110–126. https://doi.org/10.4230/LIPIcs.ICDT.2015.110

[22] Hubie Chen and Stefan Mengel. 2016. Counting Answers to Existential Positive Queries: A Complexity Classification. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Tova Milo and Wang-Chiew Tan (Eds.). ACM, 315–326. https://doi.org/10.1145/2902251.2902279

[23] Yijia Chen, Kord Eickmeyer, and Jörg Flum. 2012. The Exponential Time Hypothesis and the Parameterized Clique Problem. In *Parameterized and Exact Computation - 7th International Symposium, IPEC 2012, Ljubljana, Slovenia,*

September 12-14, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7535), Dimitrios M. Thilikos and Gerhard J. Woeginger (Eds.). Springer, 13–24. https://doi.org/10.1007/978-3-642-33293-7_4

[24] Radu Curticapean, Holger Dell, and Dániel Marx. 2017. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, Hamed Hatami, Pierre McKenzie, and Valerie King (Eds.). ACM, 210–223. https://doi.org/10.1145/3055399.3055502

[25] Víctor Dalmau and Peter Jonsson. 2004. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.* 329, 1-3 (2004), 315–323. https://doi.org/10.1016/j.tcs.2004.08.008

[26] Holger Dell, Martin Grohe, and Gaurav Rattan. 2018. Lovász Meets Weisfeiler and Leman. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic (LIPIcs, Vol. 107)*, Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 40:1–40:14. https://doi.org/10.4230/LIPIcs.ICALP.2018.40

[27] Holger Dell, Marc Roth, and Philip Wellnitz. 2019. Counting Answers to Existential Questions. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece (LIPIcs, Vol. 132)*, Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 113:1–113:15. https://doi.org/10.4230/LIPIcs.ICALP.2019.113

[28] Arnaud Durand and Stefan Mengel. 2013. Structural tractability of counting of solutions to conjunctive queries. In *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, Wang-Chiew Tan, Giovanna Guerrini, Barbara Catania, and Anastasios Gounaris (Eds.). ACM, 81–92. https://doi.org/10.1145/2448496.2448508

[29] Uriel Feige, Mohammad Taghi Hajiaghayi, and James R. Lee. 2008. Improved Approximation Algorithms for Minimum Weight Vertex Separators. *SIAM J. Comput.* 38, 2 (2008), 629–657. https://doi.org/10.1137/05064299X

[30] Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Zivný. 2022. Approximately Counting Answers to Conjunctive Queries with Disequalities and Negations. In *PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Leonid Libkin and Pablo Barceló (Eds.). ACM, 315–324. https://doi.org/10.1145/3517804.3526231

[31] Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Zivný. 2023. Counting Answers to Unions of Conjunctive Queries: Natural Tractability Criteria and Meta-Complexity. *CoRR* abs/2311.10634 (2023). https://doi.org/10.48550/ARXIV.2311.10634 arXiv:2311.10634

[32] Martin Fürer. 2017. On the Combinatorial Power of the Weisfeiler-Lehman Algorithm. In *Algorithms and Complexity - 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10236)*, Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos (Eds.). 260–271. https://doi.org/10.1007/978-3-319-57586-5_22

[33] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. 2014. Treewidth and Hypertree Width. In *Tractability: Practical Approaches to Hard Problems*, Lucas Bordeaux, Youssef Hamadi, and Pushmeet Kohli (Eds.). Cambridge University Press, 3–38. https://doi.org/10.1017/CBO9781139177801.002

[34] Georg Gottlob, Nicola Leone, and Francesco Scarcello. 2002. Hypertree decomposition and tractable queries. *J. Comput. System Sci.* 64, 3 (2002), 579–627. https://doi.org/10.1006/jcss.2001.1809

[35] Gianluigi Greco and Francesco Scarcello. 2014. Counting solutions to conjunctive queries: structural and hybrid tractability. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, Richard Hull and Martin Grohe (Eds.). ACM, 132–143. https://doi.org/10.1145/2594538.2594559

[36] Martin Grohe. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54, 1 (2007), 1:1–1:24. https://doi.org/10.1145/1206035.1206036

[37] Martin Grohe and Dániel Marx. 2014. Constraint Solving via Fractional Edge Covers. *ACM Transactions on Algorithms* 11, 1 (2014), 4:1–4:20. https://doi.org/10.1145/2636918

[38] Martin Grohe, Thomas Schwentick, and Luc Segoufin. 2001. When is the evaluation of conjunctive queries tractable?. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis (Eds.). ACM, 657–666. https://doi.org/10.1145/380752.380867

[39] Russell Impagliazzo and Ramamohan Paturi. 2001. On the Complexity of k-SAT. *J. Comput. Syst. Sci.* 62, 2 (2001), 367–375. https://doi.org/10.1006/jcss.2000.1727

[40] Richard M. Karp and Michael Luby. 1983. Monte-Carlo Algorithms for Enumeration and Reliability Problems. In *24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983*. IEEE Computer Society, 56–64. https://doi.org/10.1109/SFCS.1983.35

[41] Sandra Kiefer, Ilia Ponomarenko, and Pascal Schweitzer. 2019. The Weisfeiler-Leman Dimension of Planar Graphs Is at Most 3. *J. ACM* 66, 6 (2019), 44:1–44:31. https://doi.org/10.1145/3333003

[42] Phokion G. Kolaitis and Moshe Y. Vardi. 1995. On the Expressive Power of Datalog: Tools and a Case Study. *J. Comput. Syst. Sci.* 51, 1 (1995), 110–134. https://doi.org/10.1006/jcss.1995.1055

[43] Phokion G. Kolaitis and Moshe Y. Vardi. 2000. Conjunctive-Query Containment and Constraint Satisfaction. *J. Comput. Syst. Sci.* 61, 2 (2000), 302–332. https://doi.org/10.1006/jcss.2000.1713

[44] Matthias Lanzinger and Pablo Barceló. 2023. On the Power of the Weisfeiler-Leman Test for Graph Motif Parameters. *CoRR* abs/2309.17053 (2023). https://doi.org/10.48550/arXiv.2309.17053 arXiv:2309.17053

[45] Dániel Marx. 2010. Approximating fractional hypertree width. *ACM Trans. Algorithms* 6, 2 (2010), 29:1–29:17. https://doi.org/10.1145/1721837.1721845

[46] Dániel Marx. 2013. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM* 60, 6 (2013). https://doi.org/10.1145/2535926 Article No. 42.

[47] Stefan Mengel. 2021. A short note on the counting complexity of conjunctive queries. *CoRR* abs/2112.01108 (2021). arXiv:2112.01108 https://arxiv.org/abs/2112.01108

[48] Carl A. Miller. 2013. Evasiveness of Graph Properties and Topological Fixed-Point Theorems. *Found. Trends Theor. Comput. Sci.* 7, 4 (2013), 337–415. https://doi.org/10.1561/0400000055

[49] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 4602–4609. https://doi.org/10.1609/aaai.v33i01.33014602

[50] Daniel Neuen. 2023. Homomorphism-Distinguishing Closedness for Graphs of Bounded Tree-Width. *CoRR* abs/2304.07011 (2023). https://doi.org/10.48550/arXiv.2304.07011 arXiv:2304.07011

[51] Reinhard Pichler and Sebastian Skritek. 2013. Tractable counting of the answers to conjunctive queries. *J. Comput. Syst. Sci.* 79, 6 (2013), 984–1001. https://doi.org/10.1016/j.jcss.2013.01.012

[52] Marc Roth and Johannes Schmitt. 2020. Counting Induced Subgraphs: A Topological Approach to #W[1]-hardness. *Algorithmica* 82, 8 (2020), 2267–2291. https://doi.org/10.1007/s00453-020-00676-9

[53] Bjarke Hammersholt Roune and Eduardo Sáenz-de-Cabezón. 2013. Complexity and algorithms for Euler characteristic of simplicial complexes. *J. Symb. Comput.* 50 (2013), 170–196. https://doi.org/10.1016/j.jsc.2012.07.003

[54] Moshe Y. Vardi. 2000. Constraint Satisfaction and Database Theory: a Tutorial. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, May 15-17, 2000, Dallas, Texas, USA*, Victor Vianu and Georg Gottlob (Eds.). ACM, 76–85. https://doi.org/10.1145/335168.335209

[55] Mihalis Yannakakis. 1981. Algorithms for Acyclic Database Schemes. In *Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings*. IEEE Computer Society, 82–94.

Received December 2023; revised February 2024; accepted March 2024