Answering Counting Queries over DL-Lite Ontologies

Meghyn Bienvenu[1], **Quentin Manière**[1] and **Michaël Thomazo**[2]
[1]University of Bordeaux, CNRS, Bordeaux INP, LaBRI, Talence, France
[2]Inria, DI ENS, ENS, CNRS, University PSL, Paris, France
{meghyn.bienvenu, quentin.maniere}@u-bordeaux.fr, michael.thomazo@inria.fr

Abstract

Ontology-mediated query answering (OMQA) is a promising approach to data access and integration that has been actively studied in the knowledge representation and database communities for more than a decade. The vast majority of work on OMQA focuses on conjunctive queries, whereas more expressive queries that feature counting or other forms of aggregation remain largely unexplored. In this paper, we introduce a general form of counting query, relate it to previous proposals, and study the complexity of answering such queries in the presence of DL-Lite ontologies. As it follows from existing work that query answering is intractable and often of high complexity, we consider some practically relevant restrictions, for which we establish improved complexity bounds.

1 Introduction

Ontology-mediated query answering (OMQA) utilizes ontologies to provide a convenient vocabulary for query formulation and to capture domain knowledge that is exploited during the querying process to obtain more complete sets of answers [Poggi *et al.*, 2008; Bienvenu and Ortiz, 2015; Xiao *et al.*, 2018]. Much of the work on OMQA considers ontologies formulated using description logics (DLs), a family of knowledge representation languages that provide the logical foundations of the OWL web ontology language. Particular attention has been to the DL-Lite family of DLs [Calvanese *et al.*, 2007], which were developed with OMQA in mind and enjoy favorable computational properties.

The vast majority of work on OMQA supposes that user queries are given as conjunctive queries (CQs). However, there are many other kinds of database queries, beyond plain CQs, that are relevant in practice. This motivates research into the feasibility of adopting other database query languages for OMQA. While enriching CQs with either negated atoms or inequalities has been shown to lead to undecidability even in very restricted settings [Gutiérrez-Basulto *et al.*, 2015], the situation is more positive for navigational queries (like regular path queries), which can be adopted without losing decidability, sometimes even retaining tractable data complexity [Bienvenu *et al.*, 2015b].

Aggregate queries, which use numeric operators (e.g. count, sum, max) to summarize selected parts of a dataset, constitute another prominent class of database queries. Although such queries are widely used for data analysis, they have been little explored in context of OMQA. This may be partly due to the fact that it is not at all obvious how to define the semantics of such queries in the OMQA setting. A first exploration of aggregate queries in OMQA was conducted by Calvanese *et al.* (2008). They argued that the most straightforward adaptation of classical certain answer semantics to aggregate queries was unsatisfactory, as often values would differ from model to model, leading to no certain answers. For this reason, an epistemic semantics was proposed, in which variables involved in the aggregation are required to match to data constants. However, as discussed in [Kostylev and Reutter, 2015], this semantics can also give unintuitive results by ignoring ways of mapping aggregate variables to anonymous elements inferred due the ontology axioms. For instance, if no children of alex are listed in the data, then a query that asks to return the number of children will yield 0 under epistemic semantics, even if it can be inferred (e.g. due to a family tax benefit) that there must be at least 3 children. This led Kostylev and Reutter to define an alternative semantics for two kinds of counting queries (inspired by the COUNT and COUNT DISTINCT in SQL) which adopts a form of certain answer semantics but considers lower and upper bounds on the count value across different models. For the two considered logics (DL-Lite$_{\mathsf{core}}$ and DL-Lite$_{\mathcal{R}}$), only the lower bounds on the count value are non-trivial, and a complexity analysis shows that they are challenging to identify: coNP-data complexity for both logics, and Π_2^p-hard (resp. coNEXP-hard) in combined complexity for DL-Lite$_{\mathsf{core}}$ (resp. DL-Lite$_{\mathcal{R}}$). Several question were left unanswered by their work, including the exact combined complexity, the difficulty of recognizing the optimal lower bound, and the impact of allowing multiple aggregation variables.

This paper returns to the issue of handling counting queries in OMQA and makes several important contributions:

1. We propose a new notion of counting CQ that generalizes the two forms of queries from [Kostylev and Reutter, 2015] and allows arbitrarily many counting variables.
2. We show that existing complexity results for DL-Lite$_{\mathsf{core}}$ and DL-Lite$_{\mathcal{R}}$ KBs continue to hold for our more general notion of counting CQ, and provide an improved coNEXP

upper bound for the relevant case of finite-depth TBoxes.
3. We consider the impact of restricting the query structure, focusing on the class of rooted queries, in which every query variable must be connected to an answer variable or individual in the query graph. A recent result, obtained as part of a study of bag semantics for OMQA, identified a case in which rootedness leads to tractable data complexity for counting queries [Nikolaou *et al.*, 2019]. This motivates us to perform a more thorough investigation of rooted counting queries, which yields several improvements upon existing complexity bounds.
4. We prove that the problem of identifying the best certain interval is DP-complete in data complexity.

Our results close some questions that were left open by the work of Kostylev and Reutter and pave the way for further study of counting and aggregate queries in the OMQA setting.

An appendix with full proofs can be found in the long version of this paper, available on arXiv.

2 Preliminaries

We recall the basics of description logics (DLs), focusing on DL-Lite, see [Baader *et al.*, 2017] for more details.

Syntax and Semantics. A description logic vocabulary consists of a set $\mathsf{N_C}$ of *atomic concepts* (unary predicates), a set $\mathsf{N_R}$ of *atomic roles* (binary predicates), and a set $\mathsf{N_I}$ of *individual names* (constants). By *role*, we mean either an atomic role $P \in \mathsf{N_R}$ or an *inverse role* P^- (where $P \in \mathsf{N_R}$). We let $\mathsf{N_R^\pm}$ denote the set $\mathsf{N_R} \cup \{P^- \mid P \in \mathsf{N_R}\}$ of roles and use the notation R^- to mean P^- if $R = P \in \mathsf{N_R}$ and P if $R = P^-$.

A DL *knowledge base (KB)* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, consisting of an ABox $\mathcal{A}$ that contains facts about particular individuals and a TBox $\mathcal{T}$ that expresses general knowledge about the domain. Formally, an *ABox* is a finite set of *concept assertions* $A(b)$, with $A \in \mathsf{N_C}$ and $b \in \mathsf{N_I}$, and *role assertions* $P(a, b)$, with $P \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individuals in $\mathcal{A}$. A *TBox* is a finite set of axioms, whose syntax depends on the particular DL. In DL-Lite$_{\mathsf{core}}$, axioms take the form of *concept inclusions* $B_1 \sqsubseteq (\neg)B_2$, where each B_i is either A (for $A \in \mathsf{N_C}$) or $\exists R$ (with $R \in \mathsf{N_R^\pm}$). DL-Lite$_\mathcal{R}$ TBoxes additionally allow *role inclusions* $R_1 \sqsubseteq (\neg)R_2$, where $R_1, R_2 \in \mathsf{N_R^\pm}$.

Example 1. *Our example KB talks about leading (*LeadIn*) and supporting actors (*SuppIn*) in movies:*

$$\mathcal{A}_{\mathsf{act}} = \{\mathrm{ActsIn}(\mathsf{doona}, \mathsf{cloud}), \mathrm{SuppIn}(\mathsf{berry}, \mathsf{cloud}), \mathrm{SuppIn}(\mathsf{hanks}, \mathsf{cloud}), \mathrm{SuppIn}(\mathsf{hanks}, \mathsf{catch})\}$$
$$\mathcal{T}_{\mathsf{act}} = \{\mathrm{LeadIn} \sqsubseteq \mathrm{ActsIn}, \mathrm{SuppIn} \sqsubseteq \mathrm{ActsIn}, \exists \mathrm{SuppIn}^- \sqsubseteq \exists \mathrm{LeadIn}^-\}$$

An interpretation takes the form $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set (the *domain* of $\mathcal{I}$), and $\cdot^\mathcal{I}$ is a function that maps each $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, each $P \in \mathsf{N_R}$ to a binary relation $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and each $a \in \mathsf{N_I}$ to an element $a^\mathcal{I}$ of $\Delta^\mathcal{I}$. We make the *unique names assumption* (UNA) by requiring that $a^\mathcal{I} \neq b^\mathcal{I}$ for every $a, b \in \mathsf{N_I}$ with $a \neq b$. The function $\cdot^\mathcal{I}$ naturally extends to complex concepts and roles: $(\exists R)^\mathcal{I} = \{d \mid \exists d' : (d, d') \in R^\mathcal{I}\}$, $(P^-)^\mathcal{I} = \{(d_1, d_2) \mid (d_2, d_1) \in P^\mathcal{I}\}$, $(\neg B)^\mathcal{I} = \Delta^\mathcal{I} \setminus B^\mathcal{I}$, $(\neg R)^\mathcal{I} = \left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right) \setminus R^\mathcal{I}$. A (concept or role) inclusion $F \sqsubseteq G$ is satisfied in $\mathcal{I}$ if $F^\mathcal{I} \subseteq G^\mathcal{I}$; assertion $A(b)$ is satisfied in $\mathcal{I}$ if $b^\mathcal{I} \in A^\mathcal{I}$; $P(a, b)$ is satisfied in $\mathcal{I}$ if $(a^\mathcal{I}, b^\mathcal{I}) \in P^\mathcal{I}$. We call $\mathcal{I}$ a *model* of $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$, if it satisfies all inclusions and assertions in $\mathcal{K}$. A KB is *satisfiable* if has at least one model.

Queries. We recall that a *conjunctive query* (CQ) takes the form $\exists \boldsymbol{y}\, \psi(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are tuples of variables drawn from an infinite set of variables $\mathbf{V}$, and ψ is a conjunction of *atoms*, which can be either *concept atoms* $A(t_1)$ or *role atoms* $P(t_1, t_2)$, where $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$, and *terms* t_i are drawn from $\mathsf{N_I} \cup \boldsymbol{x} \cup \boldsymbol{y}$. Consider an interpretation $\mathcal{I}$ and CQ $q = \exists \boldsymbol{y}\, \psi(\boldsymbol{x}, \boldsymbol{y})$ with $|\boldsymbol{x}| = n$. A tuple $\boldsymbol{\alpha} \in \left(\Delta^\mathcal{I}\right)^n$ is an *answer to q in $\mathcal{I}$*, written $\mathcal{I} \models q(\boldsymbol{\alpha})$, if there is a *homomorphism of q into $\mathcal{I}$*, i.e., a function σ that maps the terms of q to elements of $\Delta^\mathcal{I}$ such that (i) $\sigma(a) = a^\mathcal{I}$ for $a \in \mathsf{N_I}$, (ii) $\sigma(t) \in A^\mathcal{I}$ for every atom $A(t)$ of q, and (iii) $(\sigma(t_1), \sigma(t_2)) \in P^\mathcal{I}$ for every atom $P(t_1, t_2)$ of q. A tuple $\boldsymbol{a} \in \mathsf{Ind}(\mathcal{A})^n$ is a *certain answer to q w.r.t. the KB $\mathcal{K}$* iff $\mathcal{I} \models q(\boldsymbol{a}^\mathcal{I})$ for every model $\mathcal{I}$ of $\mathcal{K}$.

Canonical Model. We recall the definition of the canonical model $\mathcal{C}_\mathcal{K}$ of a DL-Lite$_\mathcal{R}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. The domain of $\mathcal{C}_\mathcal{K}$ consists of $\mathsf{Ind}(\mathcal{A})$ and all words of the form $aR_1 \dots R_n$, with $a \in \mathsf{Ind}(\mathcal{A})$, $R_i \in \mathsf{N_R^\pm}$, and $n \geqslant 1$, such that:

- $\mathcal{K} \models \exists R_1(a)$ and there is no $R_1(a, b) \in \mathcal{A}$;
- for $1 \leq i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$.

We interpret individuals as themselves ($a^{\mathcal{C}_\mathcal{K}} = a$) and atomic concepts and roles as follows:

$$A^{\mathcal{C}_\mathcal{K}} = \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup \{aR_1 \dots R_n \in \Delta^{\mathcal{C}_\mathcal{K}} \setminus \mathsf{Ind}(\mathcal{A}) \mid \mathcal{T} \models \exists R_n^- \sqsubseteq A\}$$
$$P^{\mathcal{C}_\mathcal{K}} = \{(a, b) \mid P(a, b) \in \mathcal{A}\} \cup \{(e_1, e_2) \mid e_2 = e_1 R \text{ and } \mathcal{T} \models R \sqsubseteq P\} \cup \{(e_2, e_1) \mid e_2 = e_1 R \text{ and } \mathcal{T} \models R \sqsubseteq P^-\}$$

The term 'canonical model' is motivated by the following well-known property of $\mathcal{C}_\mathcal{K}$ (see e.g. [Calvanese *et al.*, 2007]):

Lemma 1. *Let $\mathcal{K}$ be a satisfiable DL-Lite$_\mathcal{R}$ KB. Then $\mathcal{C}_\mathcal{K} \models \mathcal{K}$ and if $\mathcal{I} \models \mathcal{K}$, there is a homomorphism of $\mathcal{C}_\mathcal{K}$ into $\mathcal{I}$.*

A useful corollary is that the *certain answers to a CQ q w.r.t. $\mathcal{K}$* are the tuples from $\mathsf{Ind}(\mathcal{A})$ that are *answers to q in $\mathcal{C}_\mathcal{K}$*.

Note that $\mathcal{C}_\mathcal{K}$ may be infinite. The *depth* of a TBox $\mathcal{T}$ is defined as the maximal length of any word that appears in the domain of $\mathcal{C}_\mathcal{K}$ for any KB $\mathcal{K}$ whose TBox is $\mathcal{T}$. If this number is finite, we say that $\mathcal{T}$ is a *finite-depth TBox*; such TBoxes can be identified in polynomial time [Bienvenu *et al.*, 2015a].

3 Counting Queries

We now introduce our formalization of counting queries. In addition to the set $\mathbf{V}$ of (classical) variables, we assume a second infinite set of counting variables $\mathbf{V_c}$, disjoint from $\mathbf{V}$.

Definition 1. *A counting conjunctive query (CCQ) q takes the form $q(\boldsymbol{x}) = \exists \boldsymbol{y} \exists \boldsymbol{z}\, \psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, where $\boldsymbol{x} \cup \boldsymbol{y} \subseteq \mathbf{V}$, $\boldsymbol{z} \subseteq \mathbf{V_c}$, and ψ is a conjunction of concept and role atoms whose terms are drawn from $\mathsf{N_I} \cup \boldsymbol{x} \cup \boldsymbol{y} \cup \boldsymbol{z}$. We call $\boldsymbol{x}$ (resp. $\boldsymbol{y}$, resp. $\boldsymbol{z}$) the* answer *(resp.* existential*, resp.* counting*) variables of q.*

We first define the semantics of counting queries on a single interpretation $\mathcal{I}$, by considering those pairs $(\boldsymbol{a}, n)$ such that n is the number of possible ways to map $\boldsymbol{z}$ into $\mathcal{I}$ when $\boldsymbol{x}$ is mapped to $\boldsymbol{a}$. Such pairs are called the *answers* to q in $\mathcal{I}$.

Definition 2. *A* match *of a CCQ $q(\boldsymbol{x}) = \exists \boldsymbol{y} \exists \boldsymbol{z}\ \psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ in $\mathcal{I}$ is a homomorphism[1] from q into $\mathcal{I}$. If a match σ maps $\boldsymbol{x}$ to $\boldsymbol{a}$, then the restriction of σ to $\boldsymbol{z}$ is called a* counting match (c-match) *of $q(\boldsymbol{a})$ in $\mathcal{I}$. The set of* answers to q in $\mathcal{I}$*, denoted $q^{\mathcal{I}}$, contains all pairs $(\boldsymbol{a}, q^{\mathcal{I}}_{\boldsymbol{a}})$, where $q^{\mathcal{I}}_{\boldsymbol{a}}$ is the number of distinct c-matches of $q(\boldsymbol{a})$ in $\mathcal{I}$.*

As has been previously noted (see e.g. [Kostylev and Reutter, 2015]), the exact count values of the answers in $q^{\mathcal{I}}$ are usually too specific to hold across models. Considering *bounds* on the exact value provides more insight, while still allowing unnamed elements to be counted. This motivates the following notion of answer interval.

Definition 3. *The set $[q]^{\mathcal{I}}$ of* answer intervals for a CCQ q in $\mathcal{I}$ *contains all pairs $(\boldsymbol{a}, [m, M])$ with $\boldsymbol{a} \in \mathsf{Ind}^{|\boldsymbol{x}|}$ and m, M integers such that $m \leq q^{\mathcal{I}}_{\boldsymbol{a}} \leq M$. The set $[q]^{\mathcal{K}}$ of* certain (counting) answers to q w.r.t. KB $\mathcal{K}$ *is obtained by considering those answer intervals that hold in all models of $\mathcal{K}$: $[q]^{\mathcal{K}} = \bigcap_{\mathcal{I} \models \mathcal{K}} [q]^{\mathcal{I}}$.*

Note that $(\boldsymbol{a}, [m, M]) \in [q]^{\mathcal{K}}$ does not imply that for any $n \in [m, M]$ there exists a model $\mathcal{I}$ in which $(\boldsymbol{a}, n) \in q^{\mathcal{I}}$.

Definition 1 is a proper generalization of the two forms of counting query considered by Kostylev and Reutter. Reusing their notations, a $Cntd()$-query $q(\boldsymbol{x}, Cntd(z)) = \exists \boldsymbol{y}\ \psi(\boldsymbol{x}, \boldsymbol{y}, z)$ corresponds to the CCQ $q(\boldsymbol{x}) = \psi(\boldsymbol{x}, \boldsymbol{y}, z)$, while a $Count()$-query $q(\boldsymbol{x}, Count()) = \exists \boldsymbol{y}\ \psi(\boldsymbol{x}, \boldsymbol{y})$ corresponds to the CCQ $q(\boldsymbol{x}) = \psi(\boldsymbol{x}, \emptyset, \hat{\boldsymbol{y}})$ (with $\hat{\boldsymbol{y}}$ a tuple of variables from $\mathbf{V_c}$ in bijection with $\boldsymbol{y}$). We will use the term *exhaustive* to refer to the latter CCQs, i.e. those in which every non-answer variable is a counting variable.

Example 2. *Reconsider the KB $\mathcal{K}_{\mathsf{act}} = (\mathcal{T}_{\mathsf{act}}, \mathcal{A}_{\mathsf{act}})$. We can use CCQs to count the pairs of actors (leading role, supporting role) having acted together (q_1), return movies together with a count of their supporting actors (q_2), and count the number of actors having acted with Tom Hanks (q_3):*

$$q_1 = \exists y \exists z_1 \exists z_2\ \mathrm{LeadIn}(z_1, y) \wedge \mathrm{SuppIn}(z_2, y)$$
$$q_2(x) = \exists z\ \mathrm{SuppIn}(z, x)$$
$$q_3 = \exists y \exists z\ \mathrm{ActsIn}(\mathsf{hanks}, y) \wedge \mathrm{ActsIn}(z, y)$$

According to our semantics, we have:

- $(\emptyset, [2, +\infty]) \in [q_1]^{\mathcal{K}_{\mathsf{act}}}$*, since z_2 can be mapped to either* berry *or* hanks*, and z_1 mapped to the lead actor (which must exist due to $\mathcal{T}_{\mathsf{act}}$). As the lead actors of the two films could be the same, $(\emptyset, [3, +\infty]) \notin [q_1]^{\mathcal{K}_{\mathsf{act}}}$.*
- $(\mathsf{cloud}, [2, +\infty]) \in [q_2]^{\mathcal{K}_{\mathsf{act}}}$*, mapping z to* berry *and* hanks*.*
- $(\emptyset, 5) \in q_3^{\mathcal{C}_{\mathcal{K}_{\mathsf{act}}}}$*, since in $\mathcal{C}_{\mathcal{K}_{\mathsf{act}}}$, we can map z to a named actor or the two elements standing in for the lead actors.*
- $(\emptyset, [5, +\infty]) \notin [q_3]^{\mathcal{K}_{\mathsf{act}}}$*, since the lead actors could possibly be the same or one of the named actors.*

The latter two points show that the canonical model does not yield the minimal number of matches.

[1] The notion of homomorphism of a CCQ is defined in the same way as for CQs, simply treating variables from $\mathbf{V_c}$ like those in $\mathbf{V}$.

	Data	Combined
DL-Lite$_{\mathsf{core}}$	coNP-**c**	Π^p_2-**h**, PP-**h** & in coNEXP
DL-Lite$_{\mathcal{R}}$	coNP-**c**	coNEXP-**h** & in coN2EXP coNEXP-**c** ($\mathcal{T}$ of finite depth)

Table 1: Data and combined complexity of CCQ answering

4 General Counting CQs

We shall consider the following CCQ answering decision problem: given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, CCQ q, and candidate answer $(\boldsymbol{a}, [m, M])$, decide whether $(\boldsymbol{a}, [m, M]) \in [q]^{\mathcal{K}}$.

As ontology languages, we will consider DL-Lite$_{\mathcal{R}}$ (which underlies OWL 2 QL) and its sublogic DL-Lite$_{\mathsf{core}}$. We know from [Kostylev and Reutter, 2015] that in these DLs, the least upper bound M can take one of three values (0, 1, or $+\infty$) and is easily computed. The argument[2] transfers to our more general notion of CCQ. We can therefore *restrict our attention to identifying certain answers of the form $(\boldsymbol{a}, [m, +\infty])$.*

We will consider the two usual complexity measures: *combined complexity* which is in terms of the size of the whole input ($\mathcal{T}$, $\mathcal{A}$, q, $\boldsymbol{a}$, m), and *data complexity* which is only in terms of the size of $\mathcal{A}$ and m ($\mathcal{T}$ and q are treated as fixed). We will assume that m is given in binary.

4.1 General Case

Table 1 displays complexity results for answering general CCQs over DL-Lite$_{\mathsf{core}}$ and DL-Lite$_{\mathcal{R}}$ TBoxes (we use '**-c**' and '**-h**' as abbreviations for '-complete' and '-hard').

With the exception of the PP-hardness result (discussed in Section 6.1), the lower bounds are inherited from [Kostylev and Reutter, 2015]. We will thus concentrate on the upper bounds from Table 1, which are obtained by generalizing and clarifying the constructions of Kostylev and Reutter. We give an overview of the proof both to give the flavor of the techniques involved and to enable us to discuss the necessary adaptations used to prove later results.

The proof constructs a decision procedure for the complementary problem of deciding whether $(\boldsymbol{a}, [m, +\infty]) \notin [q]^{\mathcal{K}}$. The latter holds iff there exists a *countermodel*, i.e., a model of $\mathcal{K}$ with fewer than m c-matches of $q(\boldsymbol{a})$. The main ingredient of the proof is the following theorem, which shows that it is sufficient to consider countermodels of bounded size.

Theorem 1. *For every DL-Lite$_{\mathcal{R}}$ (resp. DL-Lite$_{\mathsf{core}}$) KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and CCQ q, if there is a model of $\mathcal{K}$ with fewer than m c-matches of $q(\boldsymbol{a})$, then there exists one of size[3] $O(|\mathcal{A}|^{|\mathcal{T}|^{|q|+1}})$ (resp. $O(|\mathcal{A}|^{|q|})$).*

With Theorem 1 in hand, we can easily define nondeterministic procedures that witness the complexity upper bounds from Table 1: simply guess an interpretation of polynomial / exponential / double-exponential size (depending on the case) and verify whether it is a countermodel.

The proof of Theorem 1 starts with an arbitrary countermodel $\mathcal{I}$ and modifies it in order to make it smaller, being

[2] Briefly, the upper bound is 0 if the tuple is not a certain answer; otherwise, it is either 1 if $\boldsymbol{z} = \emptyset$, else $+\infty$.

[3] As usual, $|\mathcal{T}|$ (resp. $|\mathcal{A}|$, $|q|$) denotes the size of $\mathcal{T}$ (resp. $\mathcal{A}$, q).

careful not to introduce any new c-matches of $q(\boldsymbol{a})$. We first identify a relevant subset Δ^* of the domain of $\mathcal{I}$, consisting of the interpretations of all individual names from $\mathcal{A}$ and the images of all c-matches of $q(\boldsymbol{a})$. We then define a new interpretation that intuitively preserves Δ^* and replaces the rest of $\mathcal{I}$ with parts of the canonical model, to introduce a more regular structure. Formally, we fix a homomorphism f of $\mathcal{C}_\mathcal{K}$ into $\mathcal{I}$ (see Lemma 1) and consider the following mapping $f' : \Delta^{\mathcal{C}_\mathcal{K}} \to \Delta^* \cup \Delta^{\mathcal{C}_\mathcal{K}}$ from [Kostylev and Reutter, 2015]:

$$f'(d) \quad = \quad \begin{cases} f(d) & \text{if } f(d) \in \Delta^* \\ d & \text{otherwise} \end{cases}$$

We define the *interleaving*[4] $\mathcal{I}'$ of $\mathcal{I}$ as the image of $\mathcal{C}_\mathcal{K}$ by f', i.e., with domain $f'(\Delta^{\mathcal{C}_\mathcal{K}})$ and interpretation function $f' \circ \cdot^{\mathcal{C}_\mathcal{K}}$.

It is not difficult to prove that the interleaving $\mathcal{I}'$ is a model of $\mathcal{K}$. Moreover, by exhibiting a homomorphism ρ from $\mathcal{I}'$ to $\mathcal{I}$, we can translate matches of $\mathcal{I}'$ into matches in $\mathcal{I}$. As the images of c-matches of $q(\boldsymbol{a})$ are contained in Δ^*, which is left unchanged in $\mathcal{I}'$, the homomorphism ρ is in fact a one-to-one mapping of c-matches of $q(\boldsymbol{a})$ in $\mathcal{I}'$ to those in $\mathcal{I}$. This shows that $\mathcal{I}'$ is also a countermodel.

The interleaving $\mathcal{I}'$ may be arbitrarily large, even infinite. To reduce its size, an equivalence relation is introduced, and elements from $\Delta^{\mathcal{I}'} \setminus \Delta^*$ that belong to the same equivalence class are merged (elements from Δ^* are retained). In the case of DL-Lite$_\mathcal{R}$, there can be double-exponentially many equivalence classes, as elements are grouped based upon the properties of their $|q|$-neighborhoods, while for DL-Lite$_\mathsf{core}$, we can use a more refined relation with only exponentially many classes. This means that the resulting models are either of single- or double-exponential size w.r.t. combined complexity, depending on the chosen DL.

A crucial final step is to show that the merging of elements does not introduce any new c-matches of $q(\boldsymbol{a})$, so the resulting model is still a countermodel. This part of the argument, only sketched in [Kostylev and Reutter, 2015], requires a detailed and technical analysis of the construction to ensure that this property holds for our more general class of CCQs. We show that this is indeed the case, which answers a question left open by Kostylev and Reutter about counting CQs with both existential variables and multiple counting variables.

4.2 Case of Finite-Depth TBoxes

We give an improved upper bound for finite-depth TBoxes (which arguably cover many practical ontologies [Grau *et al.*, 2013]), pinpointing the exact combined complexity.

Theorem 2. *For finite-depth DL-Lite$_\mathcal{R}$ TBoxes, CCQ answering is* coNEXP*-complete w.r.t. combined complexity.*

Proof sketch. Fix a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. If $\mathcal{T}$ has finite depth, then $\mathcal{C}_\mathcal{K}$ contains at most $|\mathsf{Ind}(\mathcal{A})| \times |\mathcal{T}|^{|\mathcal{T}|}$ elements, which implies that, for every model $\mathcal{I}$ of $\mathcal{K}$, the interleaving of $\mathcal{I}$ is finite and of single exponential size in $|\mathcal{K}|$. Since the interleaving of a countermodel is itself a countermodel, this shows that the smallest countermodel is of single-exponential size, from which derives the improved coNEXP upper bound. □

[4]We have slightly modified the definition of interleaving to correct a small bug in the definition from [Kostylev and Reutter, 2015].

We note that the proofs of the coNP and Π_2^p lower bounds listed in Table 1 already use finite-depth TBoxes.

5 Rooted Counting CQs

We next explore whether structural restrictions on CCQs allow us to obtain lower complexity. As the lower bounds from [Kostylev and Reutter, 2015] use disconnected counting variables, a natural idea is to consider the subclass of *rooted* queries that were introduced in [Bienvenu *et al.*, 2012] and are believed to capture a large portion of real-world CQs.

Rooted CCQs can be defined analogously to rooted CQs. The definition utilizes the notion of a Gaifman graph of a CCQ, whose vertices are the query terms, and which has an undirected edge $\{t_1, t_2\}$ iff t_1, t_2 co-occur in a role atom.

Definition 4. *A CCQ $q(\boldsymbol{x}) := \exists \boldsymbol{y} \exists \boldsymbol{z} \psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is* rooted *if every connected component of the Gaifman graph of q contains at least one answer variable or individual name.*

Example queries q_2 and q_3 are rooted, while q_1 is not.

Rootedness has been shown to lower the complexity of reasoning in several settings. Most relevant to us is a recent result by Nikolaou *et al.* (2019) that rooted CQ answering under bag semantics[5] has tractable data complexity in DL-Lite$_\mathsf{core}$, and furthermore, the same holds for rooted versions of the $Count()$-queries of Kostylev and Reutter under suitable restrictions on the TBox. These techniques can be adapted to show tractability for arbitrary DL-Lite$_\mathsf{core}$TBoxes:

Theorem 3. ***(Implicit in [Nikolaou* et al.*, 2019; Cima* et al.*, 2019])*** *In DL-Lite$_\mathsf{core}$, exhaustive rooted CCQ answering is TC^0-complete[6] w.r.t. data complexity.*

Proof sketch. Nikolaou *et al.* prove that answering rooted CQs under bag semantics can be done via a rewriting to BCALC, whose evaluation problem is known to be in TC^0 due to [Libkin, 2001], see [Cima *et al.*, 2019] for discussion. Moreover, they further show that for a syntactically restricted class of DL-Lite$_\mathsf{core}$ TBoxes, it is possible to reduce exhaustive rooted CCQ answering to rooted CQ answering under bag semantics. To obtain TC^0 membership for unrestricted TBoxes, the BCALC rewriting can be adapted to set-based rather than bag interpretations. In the long version, we provide an alternative self-contained proof which directly constructs a family of TC^0 circuits. A matching lower bound has not been stated, but can be shown by a simple reduction (using an empty TBox) from the TC^0-complete problem that asks, given a binary string s and number k, whether the number of 1-bits in s exceeds k [Aehlig *et al.*, 2007]. □

The preceding result naturally leads us to ask whether rootedness also bring benefits for general CCQs. Unfortunately, we show that restricting to rooted CCQs (without exhaustiveness) does not allow us to escape existing hardness results:

[5]Bag semantics, which underly practical database systems, interprets relations using multisets rather than sets [Albert, 1991].

[6]We recall that TC^0 is a circuit complexity class defined similarly to AC^0 but additionally allowing threshhold gates. It is known that $\mathsf{AC}^0 \subsetneq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{LogSpace} \subseteq \mathsf{PTime}$.

Theorem 4. *In DL-Lite$_{\mathsf{core}}$, rooted CCQ answering is* coNP*-complete w.r.t. data complexity.*

Proof sketch. The proof borrows some ideas from the proofs of Lemmas 12 and 16 from [Kostylev and Reutter, 2015]. It proceeds by reduction from the well-known coNP-complete 3COL problem: given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, return yes iff $\mathcal{G}$ has *no* 3-coloring, i.e., a mapping from $\mathcal{V}$ to $\{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$ such that adjacent vertices map to different colors (equivalently: there is no monochromatic edge).

The reduction uses atomic roles Edge and Vertex to encode the graph and HasCol to assign colors. The TBox $\mathcal{T}_{\mathsf{col}}$ has a single axiom: $\exists \mathrm{Vertex}^- \sqsubseteq \exists \mathrm{HasCol}$. The ABox $\mathcal{A}_{\mathcal{G}}$ contains an individual v for each vertex $v \in \mathcal{V}$ and an assertion $\mathrm{Edge}(\mathsf{u}, \mathsf{v})$ for each edge $\{u, v\} \in \mathcal{E}$. All vertices are connected to a special root individual a: $\mathrm{Vertex}(\mathsf{a}, \mathsf{u})$, for each $u \in \mathcal{V}$. The three colors are represented by individuals r, g and b. To ensure that the query has matches in every model, we include a 'dummy' vertex individual $\mathsf{a_v}$ and the following assertions: $\mathrm{Vertex}(\mathsf{a}, \mathsf{a_v})$, $\mathrm{Edge}(\mathsf{a_v}, \mathsf{a_v})$, $\mathrm{HasCol}(\mathsf{a_v}, \mathsf{r})$, $\mathrm{HasCol}(\mathsf{a_v}, \mathsf{g})$, and $\mathrm{HasCol}(\mathsf{a_v}, \mathsf{b})$.

The query q is the conjunction of the two subqueries:

$$q^{edge} = \exists y_c\, \exists z_1\, \exists z_2\, \mathrm{Vertex}(\mathsf{a}, z_1) \wedge \mathrm{Vertex}(\mathsf{a}, z_2) \wedge \mathrm{Edge}(z_1, z_2) \wedge \mathrm{HasCol}(z_1, y_c) \wedge \mathrm{HasCol}(z_2, y_c)$$

$$q^{col} = \exists y\, \exists z\, \mathrm{Vertex}(\mathsf{a}, y) \wedge \mathrm{HasCol}(y, z)$$

serving respectively to detect monochromatic edges and to check whether any additional colors have been introduced.

By construction, there are at least 3 c-matches for $q(\emptyset)$ in any model of the KB $\mathcal{K}_{\mathsf{col}} = (\mathcal{T}_{\mathsf{col}}, \mathcal{A}_{\mathcal{G}})$. Moreover, it can be verified that $(\emptyset, [4, +\infty])$ is a certain answer to q w.r.t. $\mathcal{K}_{\mathsf{col}}$ iff $\mathcal{G}$ is not 3-colorable. □

Theorem 5. *In DL-Lite$_{\mathcal{R}}$, rooted CCQ answering is* coNEXP*-hard w.r.t. combined complexity.*

Proof sketch. The proof adapts a reduction from the exponential grid tiling problem (Lemma 18 from [Kostylev and Reutter, 2015]), the key difference being the use of existential query variables to access (and count) the colors and bits. □

6 Exhaustive Rooted Counting CQs

We have seen in Section 5 that the rootedness restriction is not by itself sufficient to lower the complexity of CCQ answering, whereas imposing both rootedness and exhaustiveness can sometimes yield better results. This motivates us to take a closer look at the case of exhaustive rooted CCQs. The emerging complexity landscape is summarized in Table 2.

Note that exhaustive CCQs constitute a very natural form of counting query, which ask for the number of different query matches for a given answer tuple. The query q_2 from Example 2 is an exhaustive rooted CCQ.

6.1 Exhaustive Rooted CCQs in DL-Lite$_{\mathsf{core}}$

We first consider DL-Lite$_{\mathsf{core}}$ KBs and pinpoint the precise combined complexity, which had not yet been considered.

An essential ingredient is the following result that shows that it is possible to focus on query matches in the canonical model. It can be obtained by adapting a similar result about canonical bag interpretations [Nikolaou *et al.*, 2019].

	Data	**Combined**
DL-Lite$_{\mathsf{core}}$	TC^0**-c**	PP**-c**
DL-Lite$_{\mathcal{R}}$	coNP**-c**	Π_2^p**-h**, PP**-h** & in coNEXP

Table 2: Complexity results for exhaustive rooted CCQs

Theorem 6. *For every DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K}$ and exhaustive rooted CCQ q, it holds that $[q]^{\mathcal{K}} = [q]^{\mathcal{C}_{\mathcal{K}}}$.*

Proof sketch. Exploiting the structure of DL-Lite$_{\mathsf{core}}$ canonical models, one can show that if σ_1, σ_2 are distinct matches of an exhaustive rooted CCQ q in $\mathcal{C}_{\mathcal{K}}$, then there exists a variable v such that $\sigma_1(v) \neq \sigma_2(v)$ and $\sigma_1(v), \sigma_2(v) \in \mathsf{Ind}(\mathcal{A})$. It follows that if we take an arbitrary model $\mathcal{I}$ of $\mathcal{K}$, and let f be a homomorphism of $\mathcal{C}_{\mathcal{K}}$ into $\mathcal{I}$, then f injectively maps query matches in $\mathcal{C}_{\mathcal{K}}$ to query matches in $\mathcal{I}$. □

We will also use the next lemma, implicit in [Bienvenu *et al.*, 2013], constraining the possible images of matches in $\mathcal{C}_{\mathcal{K}}$:

Lemma 2. *For every DL-Lite$_{\mathsf{core}}$ TBox $\mathcal{T}$ and CCQ q, we can construct in polynomial time a set of words $\Gamma_{q,\mathcal{T}}$ such that for every KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, match σ of q in $\mathcal{C}_{\mathcal{K}}$, and variable v of q: $\sigma(v) = aw$ for some $a \in \mathsf{Ind}(\mathcal{A})$ and $w \in \Gamma_{q,\mathcal{T}}$.*

We are now ready to show that the problem is PP-complete in combined complexity, and hence in PSpace.

Theorem 7. *In DL-Lite$_{\mathsf{core}}$, exhaustive rooted CCQ answering is* PP*-complete w.r.t. combined complexity.*

Proof sketch. The class PP contains all decision problems for which there exists a non-deterministic Turing machine (TM) such that, when the input is a 'yes' instance, then at least half of the computation paths accept, while on 'no' instances, less than half of the computation paths accept.

The lower bound is obtained by a reduction from the following PP-complete problem [Bailey *et al.*, 2007]: given a propositional formula ψ in CNF and number n, decide whether ψ has at least n satisfying assignments.

We sketch the TM used to show PP membership, which takes as input a DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, an exhaustive rooted CCQ $q(\boldsymbol{x})$, and candidate answer $(\boldsymbol{a}, [m, +\infty])$:

Phase 1. The TM constructs the set $\Gamma_{q,\mathcal{T}}$ from Lemma 2.

Phase 2. The TM guesses a mapping σ of the variables in q to elements from $\{aw \mid a \in \mathsf{Ind}(\mathcal{A}), w \in \Gamma_{q,\mathcal{T}}\}$. It then compares m with the number $C = |\Gamma_{q,\mathcal{T}}|^{|q|}$ of possible mappings and proceeds accordingly:

- if $m \geq \frac{C}{2} + 1$, the TM guesses an integer i with $0 \leq i \leq 2m - 3$ and accepts iff σ is a c-match of $q(\boldsymbol{a})$ and $i < C$;
- if $m < \frac{C}{2} + 1$, the TM guesses an integer i with $0 \leq i \leq 2C - 2m + 1$ and accepts iff σ is c-match for $q(\boldsymbol{a})$ or $i < C - 2m + 2$.

The guessed integer and comparisons ensure a suitable number of accepting paths. It can be verified that at least half of the paths are accepting iff $(\boldsymbol{a}, [m, +\infty]) \in [q]^{\mathcal{C}_{\mathcal{K}}}$. □

6.2 Exhaustive Rooted CCQs in DL-Lite$_\mathcal{R}$

We now turn to DL-Lite$_\mathcal{R}$ KBs. Our first result is negative: exhaustive rooted CCQs do not enjoy lower data complexity. This is shown by another reduction from 3COL which involves ideas from our proof of Theorem 4 and the proof of Lemma 16 from [Kostylev and Reutter, 2015].

Theorem 8. *In DL-Lite$_\mathcal{R}$, exhaustive rooted CCQ answering is* coNP*-complete w.r.t. data complexity.*

More positively, we can show an improved coNEXP upper bound in combined complexity for exhaustive rooted CCQs. We briefly sketch the proof, which involves highly non-trivial modifications to the argument used for general CCQs.

We first introduce a more refined notion of interleaving, which replaces the mapping f' by the following mapping f^*:

$$\begin{array}{rcl} f^*(a) & = & f(a) \\ f^*(\omega R) & = & \begin{cases} f(\omega R) & \text{if } f^*(\omega), f(\omega R) \in \Delta^* \\ f^*(\omega)R & \text{otherwise} \end{cases} \end{array}$$

It is possible to prove that when q is an exhaustive rooted CCQ, this modified interleaving yields a countermodel. Moreover, it has a very particular structure, essentially corresponding to the canonical model of the restriction of $f(\mathcal{C}_\mathcal{K})$ to Δ^* (viewed as an ABox). Importantly, this means that instead of guessing a whole countermodel, it suffices to guess an initial, exponential-size portion (the $|q|$-neighborhood of Δ^*), providing the basis for a coNEXP decision procedure.

Theorem 9. *In DL-Lite$_\mathcal{R}$, exhaustive rooted CCQ answering is in* coNEXP *w.r.t. combined complexity.*

7 Best Certain Answers

The definition of certain answers implies that if $(\boldsymbol{a}, [m, M]) \in [q]^\mathcal{K}$, then we also have $(\boldsymbol{a}, [m', M']) \in [q]^\mathcal{K}$ for every $m' \leq m$ and $M' \geq M$. It is naturally of interest to focus on certain answers providing the best bounds, i.e., those of the form $(\boldsymbol{a}, [\min_{\mathcal{I} \models \mathcal{K}} q^\mathcal{I}_{\boldsymbol{a}}, \max_{\mathcal{I} \models \mathcal{K}} q^\mathcal{I}_{\boldsymbol{a}}])$.

In this section, we show that the problem of identifying the best lower bound $(\min_{\mathcal{I} \models \mathcal{K}} q^\mathcal{I}_{\boldsymbol{a}})$ is DP-complete in data complexity. It is easily seen that checking whether m is such an optimal bound can be done in DP, by making a call to a coNP oracle (is $(\boldsymbol{a}, [m, +\infty]) \in [q]^\mathcal{K}$?) and an NP oracle (is $(\boldsymbol{a}, [m+1, +\infty]) \notin [q]^\mathcal{K}$?). The DP-hardness of this problem was left as an open question by Kostylev and Reutter.

Theorem 10. *The following problem is* DP*-hard in data complexity: given a DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, rooted CCQ q, tuple $\boldsymbol{a}$, and number m, decide whether $m = \min_{\mathcal{I} \models \mathcal{K}} q^\mathcal{I}_{\boldsymbol{a}}$.*

Proof sketch. We give a reduction from the following problem (DP-complete due to [Garey *et al.*, 1976]): given *planar* graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, decide if $\mathcal{G}_1 \in$ 3COL and $\mathcal{G}_2 \notin$ 3COL.

Let the TBox $\mathcal{T}_{\mathsf{col}}$ and ABoxes $\mathcal{A}_{\mathcal{G}_1}, \mathcal{A}_{\mathcal{G}_2}$ be defined as in the proof of Theorem 4. Rename the individuals to ensure $\mathsf{Ind}(\mathcal{A}_{\mathcal{G}_1}) \cap \mathsf{Ind}(\mathcal{A}_{\mathcal{G}_2}) = \emptyset$, then set $\mathcal{K} = (\mathcal{T}_{\mathsf{col}}, \mathcal{A}_{\mathcal{G}_1} \cup \mathcal{A}_{\mathcal{G}_2})$. Let q_1^{color} and q_1^{edge} (resp. q_2^{color} and q_2^{edge}) be defined as before, but using disjoint variables and the root individual from the $\mathcal{A}_{\mathcal{G}_1}$ (resp. $\mathcal{A}_{\mathcal{G}_2}$). The challenge is to make sure that we can determine the 3-colorability status of the two graphs solely by looking at the number of c-matches of the query. To be able to distinguish $\mathcal{G}_1$ from $\mathcal{G}_2$, we introduce an asymmetry by duplicating the color counter query for $\mathcal{G}_1$, i.e., create a copy q_0^{color} of q_1^{color} that uses fresh variables but the same root individual. We then take the query

$$q() := q_0^{color} \ \wedge \ q_1^{color} \ \wedge \ q_1^{edge} \ \wedge \ q_2^{color} \ \wedge \ q_2^{edge}.$$

We claim $(\boldsymbol{a}_\emptyset, [36, +\infty]) \in [q]^\mathcal{K}$ iff $\mathcal{G}_1 \in$ 3COL and $\mathcal{G}_2 \notin$ 3COL. This is proven by a case analysis, summarized here:

	$\mathcal{G}_1 \in$ 3COL	$\mathcal{G}_1 \notin$ 3COL
$\mathcal{G}_2 \in$ 3COL	27 $(= 3 \times 3 \times 3)$	48 $(= 4 \times 4 \times 3)$
$\mathcal{G}_2 \notin$ 3COL	36 $(= 3 \times 3 \times 4)$	64 $(= 4 \times 4 \times 4)$

Each of the four cells displays the least value of m such that $(\boldsymbol{a}_\emptyset, [m, +\infty]) \in [q]^\mathcal{K}$, under different assumptions on the 3-colorability of $\mathcal{G}_1$ and $\mathcal{G}_2$. To establish these values, one must first prove that every model has at least this many c-matches, and then exhibit a model that realizes the exact number. For the latter, we utilize our assumption that the graphs are planar, hence 4-colorable [Gonthier, 2008], which we use to show that the minimal number of c-matches is realized in a model that encodes proper 3- or 4-colorings of the graphs. □

The preceding reduction can be adapted to show DP-hardness also for the two kinds of CCQs from [Kostylev and Reutter, 2015], but without the rootedness restriction.

8 Conclusion & Future Work

We have revisited the issue of counting queries in OMQA and advanced our understanding of the complexity landscape, both by extending existing results to a more general notion of counting CQ and by exploring when structural restrictions on the ontology and query can lead to improved complexity.

There are several natural avenues for future study. A first challenging problem is to provide a full classification of the data complexity of ontology-mediated queries (i.e. query-ontology pairs), in order to identify further tractable cases. It would also be relevant to extend the complexity study to DLs with functional roles or quantified number restrictions, which would allow for non-trivial upper bounds on the number of matches. Tackling general CCQs for such DLs will likely require wholly different techniques from the model manipulations used in Section 4. However, a recent result by Cima *et al.* (2019) shows that the canonical model property (Theorem 6) holds also for DL-Lite$_\mathcal{F}$ (which extends DL-Lite$_{\mathsf{core}}$ with functional roles), and hence both TC^0 data complexity (Theorem 3) and our PP-completeness result (Theorem 7) for exhaustive rooted CCQs transfer to DL-Lite$_\mathcal{F}$.

Much remains to be explored for queries involving other kinds of aggregate functions (min, max, sum, average), which manipulate data values. Recent studies of bag semantics for OMQA [Nikolaou *et al.*, 2019; Cima *et al.*, 2019] and databases with incomplete information [Hernich and Kolaitis, 2017; Console *et al.*, 2017] provide important formal foundations for supporting such queries.

Acknowledgements

This work was partially supported by ANR project CQFD (ANR-18-CE23-0003).

References

[Aehlig *et al.*, 2007] Klaus Aehlig, Stephen Cook, and Phuong Nguyen. *Relativizing Small Complexity Classes and Their Theories*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[Albert, 1991] Joseph Albert. Algebraic properties of bag data types. In *Proceedings of the 17th International Conference on Very Large Data Bases (VLDB)*, pages 211–219, 1991.

[Baader *et al.*, 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[Bailey *et al.*, 2007] Delbert D. Bailey, Víctor Dalmau, and Phokion G. Kolaitis. Phase transitions of PP-complete satisfiability problems. *Discrete Applied Mathematics*, 155(12):1627 – 1639, 2007.

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Tutorial Lectures of the 11th Reasoning Web International Summer School*, pages 218–307, 2015.

[Bienvenu *et al.*, 2012] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query containment in description logics reconsidered. In *Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 2012.

[Bienvenu *et al.*, 2013] Meghyn Bienvenu, Magdalena Ortiz, Mantas Simkus, and Guohui Xiao. Tractable queries for lightweight description logics. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 768–774, 2013.

[Bienvenu *et al.*, 2015a] Meghyn Bienvenu, Stanislav Kikot, and Vladimir V. Podolskii. Tree-like queries in OWL 2 QL: Succinctness and complexity results. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 317–328, 2015.

[Bienvenu *et al.*, 2015b] Meghyn Bienvenu, Magdalena Ortiz, and Mantas Simkus. Regular path queries in lightweight description logics: Complexity and algorithms. *Journal of Artificial Intelligence Research (JAIR)*, 53:315–374, 2015.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2008] Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Camilo Thorne. Aggregate queries over ontologies. In *Proceedings of the 2nd International Workshop on Ontologies and Information Systems for the Semantic Web (ONISW)*, pages 97–104, 2008.

[Cima *et al.*, 2019] Gianluca Cima, Charalampos Nikolaou, Egor V. Kostylev, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Bag semantics of dl-lite with functionality axioms. In *Proceedings of the 18th International Semantic Web Conference (ISWC)*, pages 128–144, 2019.

[Console *et al.*, 2017] Marco Console, Paolo Guagliardo, and Leonid Libkin. On querying incomplete information in databases under bag semantics. In Carles Sierra, editor, *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 993–999, 2017.

[Garey *et al.*, 1976] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976.

[Gonthier, 2008] Georges Gonthier. Formal proof – The four-color theorem. *Notices of the American Mathematical Society*, 55(11):1382–1393, 2008.

[Grau *et al.*, 2013] Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. Acyclicity notions for existential rules and their application to query answering in ontologies. *Journal of Artificial Intelligence Research (JAIR)*, 47:741–808, 2013.

[Gutiérrez-Basulto *et al.*, 2015] Víctor Gutiérrez-Basulto, Yazmin Angélica Ibáñez-García, Roman Kontchakov, and Egor V. Kostylev. Queries with negation and inequalities over lightweight ontologies. *Journal of Web Semantics (JWS)*, 35:184–202, 2015.

[Hernich and Kolaitis, 2017] André Hernich and Phokion G. Kolaitis. Foundations of information integration under bag semantics. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017.

[Kostylev and Reutter, 2015] Egor V. Kostylev and Juan L. Reutter. Complexity of answering counting aggregate queries over DL-Lite. *Journal of Web Semantics (JWS)*, 33(1):94–111, 2015.

[Libkin, 2001] Leonid Libkin. Expressive power of SQL. In *Proceedings of the 8th International Conference on Database Theory (ICDT)*, pages 1–21, 2001.

[Nikolaou *et al.*, 2019] Charalampos Nikolaou, Egor V. Kostylev, George Konstantinidis, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Foundations of ontology-based data access under bag semantics. *Artificial Intelligence (AIJ)*, 274:91 – 132, 2019.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal of Data Semantics*, 10:133–173, 2008.

[Xiao *et al.*, 2018] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5511–5519, 2018.