Counting Query Answers over a *DL-Lite* Knowledge Base

Diego Calvanese[1,2] , **Julien Corman**[1] , **Davide Lanti**[1] and **Simon Razniewski**[3]
[1] Free University of Bozen-Bolzano, Italy
[2] Umeå University, Sweden
[3] Max-Planck-Institut für Informatik, Germany
{calvanese, corman, lanti}@inf.unibz.it, srazniew@mpi-inf.mpg.de

Abstract

Counting answers to a query is an operation supported by virtually all database management systems. In this paper we focus on counting answers over a Knowledge Base (KB), which may be viewed as a database enriched with background knowledge about the domain under consideration. In particular, we place our work in the context of Ontology-Mediated Query Answering/Ontology-based Data Access (OMQA/OBDA), where the language used for the ontology is a member of the *DL-Lite* family and the data is a (usually virtual) *set* of assertions. We study the *data complexity* of query answering, for different members of the *DL-Lite* family that include number restrictions, and for variants of *conjunctive queries with counting* that differ with respect to their shape (connected, branching, rooted). We improve upon existing results by providing P and coNP lower bounds, and upper bounds in P and LogSpace. For the LogSpace case, we have devised a novel query rewriting technique into first-order logic with counting.

1 Introduction

Counting answers to a query is an essential operation in data management, and is supported by virtually every database management system. In this paper, we focus on counting answers over a Knowledge Base (KB), which may be viewed as a database (DB) enriched with background knowledge about the domain of interest. In such a setting, counting may take into account two types of information: grounded assertions (typically DB records), and existentially quantified statements (typically statistics).

As a toy example, the following is an imaginary KB storing a parent/child relation, where explicit instances (e.g., Alice is the child of Kendall) coexist with existentially quantified ones (e.g., Parker has 3 children):

hasChild(Kendall, Alice)	*"Kendall has 2 children"*
hasChild(Jordan, Alice)	*"Parker has 3 children"*
hasChild(Parker, Bob)	*"A child has at most*
hasChild(Parker, Carol)	*2 parents"*

The presence of both types of information is common when integrating multiple data sources. One source may provide detailed records (e.g., one record per purchase, medical visit, etc.), whereas another source may only provide statistics (number of purchases, of visits, etc.), due to anonymization, access restriction, or simply because the data is recorded in this way.

In such scenarios, counting answers to a query over a KB may require operations that go beyond counting records. E.g., in our example, counting the minimal number of children that must exist according to the KB (where children can be explicit or existentially quantified elements in the range of hasChild) requires some non-trivial reasoning. The answer is 4: Bob or Carol may be the second child of Kendall, but Alice cannot be the third child of Parker (because Alice has two parents already), so a fourth child must exist.

One of the most extensively studied frameworks for query answering over a KB is Ontology Mediated Query Answering (OMQA)[1] [Calvanese *et al.*, 2008a; Bienvenu and Ortiz, 2015]. In OMQA, the background knowledge takes the form of a set of logical statements, called the *TBox*, and the records are a set of facts, called the *ABox*. TBoxes are in general expressed in Description Logics (DLs), which are decidable fragments of First-Order logic that typically can express the combination of explicit and existentially quantified instances mentioned above. Therefore OMQA may provide valuable insight about the computational problem of counting over such data (even though, in practice, DLs may not be the most straightforward way to represent such data).

For Conjunctive Queries (CQs) and Unions of CQs (UCQs), DLs have been identified with the remarkable property that query answering over a KB does not induce extra computational cost (w.r.t. worst-case complexity), when compared to query answering over a relational DB [Xiao *et al.*, 2018]. This key property has led to the development of numerous techniques that leverage the mature technology of relational DBs to perform query answering over a KB. In particular, the *DL-Lite* family [Calvanese *et al.*, 2007; Artale *et al.*, 2009] has been widely studied and adopted in OMQA/OBDA systems, resulting in the OWL 2 QL standard [Motik *et al.*, 2012].

Yet the problem of *counting* answers over a *DL-Lite* KB has seen relatively little interest in the literature. In particular, whether counting answers exhibits desirable computational properties analogous to query answering is still a partly open

[1]Also referred to as OBDA (for Ontology Based Data Access), when emphasis in placed on mappings connecting external data sources to a TBox [Xiao *et al.*, 2018].

question for such DLs. A key result for counting over *DL-Lite* KBs was provided by Kostylev and Reutter [2015], who also formalized the semantics we adopt in this paper (which we call *count semantics*). For CQs interpreted under count semantics, they show a coNP lower bound in *data complexity*, i.e., considering that the sizes of the query and TBox are fixed. However, their reduction relies on a CQ that computes the cross-product of two relations, which is unlikely to occur in practice. Later on, it was shown[2] by Nikolaou *et al.* [2019] that coNP-hardness still holds (for a more expressive DL) using a branching and cyclic CQ without cross-product. Building upon these results, we further investigate how query shape affects tractability.

Another important question is whether relational DB technologies may be leveraged for counting in OMQA, as done for boolean and enumeration queries. A key property here is *rewritability*, extensively studied for *DL-Lite* and UCQs [Calvanese *et al.*, 2007], i.e., the fact that a query over a KB may be rewritten as an equivalent UCQ over its ABox only, intuitively "compiling" part of the TBox into this new UCQ. An important result in this direction was provided by Nikolaou *et al.* [2019], but in the context of query answering under *bag semantics*. For certain *DL-Lite* variants, it is shown that queries that are *rooted* (i.e., with at least one constant or answer variable) can be rewritten as queries over the ABox. Despite there being a correspondence between bag semantics and count semantics, they show that these results do not automatically carry over to query answering under count semantics, due the way bag answers are computed in the presence of a KB.

So in this work, we further investigate the boundaries of tractability and rewritability for CQs with counting over a *DL-Lite* KB, with an emphasis on DLs that can express statistics about missing information. As is common for DBs, we focus on *data complexity*, i.e., computational cost in the size of the ABox (likely to grow much faster than the query or TBox).

Due to space limitations, the techniques used to obtain our results are only sketched, but full proofs are available in the extended version of this paper [Calvanese *et al.*, 2020].

2 Preliminaries and Problem Specification

We assume mutually disjoint sets $\mathsf{N_I}$ of *individuals* (a.k.a. *constants*), $\mathsf{N_E}$ of *anonymous individuals* (induced by existential quantification), $\mathsf{N_V}$ of *variables*, $\mathsf{N_C}$ of *concept names* (i.e., unary predicates, denoted with A), and $\mathsf{N_R}$ of *role names* (i.e., binary predicates, denoted with P). In the following, boldface letters, e.g., $\mathbf{c}$, denote tuples, and we treat tuples as sets.

Functions, Atoms. $\mathsf{dom}(f)$ and $\mathsf{range}(f)$ denote the domain and range of a function f. Given $D \subseteq \mathsf{dom}(f)$, the function f restricted to the elements in D is denoted $f|_D$. A function f is *constant-preserving* iff $c = f(c)$ for each $c \in \mathsf{dom}(f) \cap \mathsf{N_I}$. If $S \subseteq \mathsf{dom}(f)$, we use $f(S)$ for $\{f(s) \mid s \in S\}$. If $\mathbf{t} = \langle t_1, \ldots, t_n \rangle$ is is a tuple with elements in $\mathsf{dom}(f)$, we use $f(\mathbf{t})$ for $\langle f(t_1), \ldots, f(t_n) \rangle$.

An atom a has the form $A(s)$ or $P(s,t)$, with $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$, and $s, t \in \mathsf{N_I} \cup \mathsf{N_E} \cup \mathsf{N_V}$.

[2]The result was stated for the related setting of *bag semantics*, but the same reduction holds for count semantics as well.

$$R \longrightarrow P \mid P^- \qquad B \longrightarrow A \mid \geq_1 R \qquad C \longrightarrow A \mid \geq_n R$$

Figure 1: Syntax of *DL-Lite$_{core}$ roles R*, *basic concepts B*, and *concepts C*, where n denotes a positive integer, i.e., $n \in \mathbb{N}^+$.

Interpretations, Homomorphisms. An *interpretation* $\mathcal{I}$ is a FO structure $\langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$, where the *domain* $\Delta^\mathcal{I}$ is a non-empty subset of $\mathsf{N_I} \cup \mathsf{N_E}$, and the *interpretation function* $\cdot^\mathcal{I}$ is a function that maps each constant $c \in \mathsf{N_I}$ to itself (i.e., $c^\mathcal{I} = c$, in other words, we adopt the *standard names assumption*), each concept name $A \in \mathsf{N_C}$ to a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and each role name $P \in \mathsf{N_R}$ to a binary relation $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

Given an interpretation $\mathcal{I}$ and a constant-preserving function f with domain $\Delta^\mathcal{I}$, we use $f(\mathcal{I})$ to denote the interpretation defined by $\Delta^{f(\mathcal{I})} = f(\Delta^\mathcal{I})$ and $E^{f(\mathcal{I})} = f(E^\mathcal{I})$ for each $E \in \mathsf{N_C} \cup \mathsf{N_R}$. Given two interpretations $\mathcal{I}_1, \mathcal{I}_2$, we use $\mathcal{I}_1 \subseteq \mathcal{I}_2$ as a shortcut for $\Delta^{\mathcal{I}_1} \subseteq \Delta^{\mathcal{I}_2}$ and $E^{\mathcal{I}_1} \subseteq E^{\mathcal{I}_2}$, for each $E \in \mathsf{N_C} \cup \mathsf{N_R}$. A *homomorphism* h from $\mathcal{I}_1$ to $\mathcal{I}_2$ is a constant-preserving function with domain $\Delta^{\mathcal{I}_1}$ that verifies $h(\mathcal{I}_1) \subseteq \mathcal{I}_2$. We note that a set S of atoms with arguments in $\mathsf{N_I} \cup \mathsf{N_E}$ uniquely identifies an interpretation, which we denote with $\mathsf{inter}(S)$.

KBs, DLs, Models. A KB is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{A}$, called *ABox*, is a finite set of atoms with arguments in $\mathsf{N_I}$, and $\mathcal{T}$, called *TBox*, is a finite set of *axioms*. We consider DLs of the *DL-Lite* family [Artale *et al.*, 2009], starting with the logic *DL-Lite$_{core}$*, where each axiom has one of the forms *(i)* $B \sqsubseteq C$ (concept inclusion), *(ii)* $B \sqsubseteq \neg C$ (concept disjointness), or *(iii)* $R \sqsubseteq R'$ (role inclusion), where now and in the following, the symbols R, B, and C are defined according to the grammar of Figure 1, and are called respectively *roles*, *basic concepts*, and *concepts*. Concepts of the form $\geq_n R$ are called *number restrictions*. *DL-Lite$_{pos}$* allows only for axioms of form *(i)*, with the requirement that the number n in number restrictions may only be 1. In this work we study extensions to this logic along three orthogonal directions: *(1)* allowing also for axioms of form *(ii)*, indicated by replacing the subscript $_{pos}$ with $_{core}$; *(2)* allowing also for axioms of form *(iii)*, indicated by adding a superscript $^\mathcal{H}$; *(3)* allowing for arbitrary numbers in number restrictions, but only on the right-hand-side (RHS) of concept inclusion, indicated by adding a superscript $^{\mathcal{N}^-}$. We also use the superscript $^{\mathcal{H}^-}$ for logics with role inclusions, but with the restriction on TBoxes defined by Nikolaou *et al.* [2019], which disallows in a TBox $\mathcal{T}$ axioms of the form $B \sqsubseteq\ \geq_n R_1$ if $\mathcal{T}$ contains a role inclusion $R_1 \sqsubseteq R_2$, for some $R_2 \neq R_1$.

The semantics of DL constructs is specified as usual [Baader *et al.*, 2003]. An interpretation $\mathcal{I}$ is a *model* of $\langle \mathcal{T}, \mathcal{A} \rangle$ iff $\mathsf{inter}(\mathcal{A}) \subseteq \mathcal{I}$, and $E_1^\mathcal{I} \subseteq E_2^\mathcal{I}$ holds for each axiom $E_1 \sqsubseteq E_2$ in $\mathcal{T}$. A KB is *satisfiable* iff it admits at least one model. For readability, in what follows we focus on satisfiable KBs, that is, we use "a KB" as a shortcut for "a satisfiable KB". We use the binary relation $\sqsubseteq_\mathcal{T}$ over *DL-Lite* concepts and roles E_1, E_2 to denote entailment w.r.t. a TBox $\mathcal{T}$, defined by $E_1 \sqsubseteq_\mathcal{T} E_2$ iff $E_1^\mathcal{I} \subseteq E_2^\mathcal{I}$ for each model $\mathcal{I}$ of the KB $\langle \mathcal{T}, \emptyset \rangle$.

A key property of a *DL-Lite* KB $\mathcal{K}$ is the existence of a so-called *canonical model* $\mathcal{I}_{can}^\mathcal{K}$, unique up to isomorphism, s.t. there exists a homomorphism from $\mathcal{I}_{can}^\mathcal{K}$ to each model of $\mathcal{K}$. This model can be constructed via the *restricted chase* procedure by Calvanese *et al.* [2013], Botoeva *et al.* [2010].

Finally, we observe that axioms of the form $B \sqsubseteq \geq_n R$ can be expressed in the logic *DL-Lite*$^{\mathcal{H}}_{core}$, but with a possibly exponential blowup of the TBox (assuming n is encoded in binary). For instance, the axiom $A \sqsubseteq \geq_2 P$ can be expressed as $\{A \sqsubseteq \exists P_1, A \sqsubseteq \exists P_2, P_1 \sqsubseteq P, P_2 \sqsubseteq P, \exists P_1^- \sqsubseteq \neg \exists P_2^-\}$, with P_1, P_2 fresh DL roles.

CQs. A *Conjunctive Query* (CQ) q is an expression of the form $q(\mathbf{x}) \leftarrow p_1(\mathbf{t}_1), \ldots, p_n(\mathbf{t}_n)$, where each $p_i \in \mathsf{N_C} \cup \mathsf{N_R}$, $\mathbf{x} \subseteq \mathsf{N_V}$, each $\mathbf{t}_i \subseteq \mathsf{N_V} \cup \mathsf{N_I}$, and $p_1(\mathbf{t}_1), \ldots, p_n(\mathbf{t}_n)$ is syntactic sugar for the duplicate-free conjunction of atoms $p_1(\mathbf{t}_1) \wedge \cdots \wedge p_n(\mathbf{t}_n)$. Since all conjunctions in this work are duplicate-free, we sometimes treat them as sets of atoms. The variables in $\mathbf{x}$, called *distinguished*, are denoted by $\mathsf{dist}(q)$, $\mathsf{head}(q)$ denotes the *head* $q(\mathbf{x})$ of q, and $\mathsf{body}(q)$ denotes the *body* $\{p_1(\mathbf{t}_1), \ldots, p_n(\mathbf{t}_n)\}$ of q. We require safeness, i.e., $\mathbf{x} \subseteq \mathbf{t}_1 \cup \cdots \cup \mathbf{t}_n$. A query is *boolean* if $\mathbf{x}$ is the empty tuple.

Answers, Certain Answers. To define query answers under count semantics, we adapt the definitions by Cohen *et al.* [2007] and Kostylev and Reutter [2015]. A *match* for a query q in an interpretation $\mathcal{I}$ is a homomorphism from $\mathsf{body}(q)$ to $\mathcal{I}$. Then, an *answer* to q over $\mathcal{I}$ is a pair $\langle \omega, k \rangle$ s.t. $k \geq 1$, and there are *exactly* k matches $\rho_1, \ldots, \rho_k$ for q in $\mathcal{I}$ that verify $\omega = \rho_i|_{\mathsf{dist}(q)}$ for $i \in \{1, \ldots, k\}$. We use $\mathsf{ans}(q, \mathcal{I})$ to denote the set of answers to q over $\mathcal{I}$. Similarly, if $\mathcal{Q}$ is a set of queries, we use $\mathsf{ans}(\mathcal{Q}, \mathcal{I})$ to denote the set of all pairs $\langle \omega, \ell \rangle$ s.t. $\langle \omega, k \rangle \in \mathsf{ans}(q, \mathcal{I})$ for some k and $q \in \mathcal{Q}$, and $\ell = \sum\{k \mid \langle \omega, k \rangle \in \mathsf{ans}(q, \mathcal{I}), q \in \mathcal{Q}\}$. Answering a query over an interpretation (i.e., a DB) is also known as *query evaluation*. Finally, a pair $\langle \omega, k \rangle$ is a *certain answer* to a query q over a KB $\mathcal{K}$ iff $k \geq 1$ is the *largest* integer such that, for each model $\mathcal{I}$ of $\mathcal{K}$, $\langle \omega, k_{\mathcal{I}} \rangle \in \mathsf{ans}(q, \mathcal{I})$ for some $k_{\mathcal{I}} \geq k$. We use $\mathsf{certAns}(q, \mathcal{K})$ to denote the set of certain answers to q over $\mathcal{K}$.

Decision Problem. The decision problem defined by Kostylev and Reutter [2015] takes as input a query q, a mapping ω, a KB $\mathcal{K}$, and an integer k, and decides $\langle \omega, k \rangle \in \mathsf{certAns}(q, \mathcal{K})$. It is easy to see that an instance of this problem can be reduced (in linear time) to an instance where q is a boolean query and ω is the empty mapping, by introducing constants in $\mathsf{body}(q)$. We will use the following simplified setting for the complexity results below: if q is a boolean query and ε the empty mapping, we use $k = \mathsf{certCard}(q, \mathcal{K})$ as an abbreviation for $\langle \varepsilon, k \rangle \in \mathsf{certAns}(q, \mathcal{K})$. Then, the problem Count is stated as follows:

COUNT	**Input**: *DL-Lite* KB $\mathcal{K}$, boolean CQ q, $k \in \mathbb{N}^+$ **Decide**: $k = \mathsf{certCard}(q, \mathcal{K})$

Data complexity. As usual for query answering over DBs [Vardi, 1982] or KBs [Calvanese *et al.*, 2007], we distinguish between *combined* and *data* complexity. For the latter, we adopt the definition provided by Nikolaou *et al.* [2019], i.e., we measure data complexity in the cumulated size of the ABox and the input integer k (encoded in binary).

Query Shape. As we will see later, the shape of the input CQ may play a role for tractability. We define here the different query shapes used throughout the article. Because our focus is on queries with unary and binary atoms, we can use the Gaifman graph [Bienvenu *et al.*, 2017] of a CQ to characterize such shapes: the Gaifman graph $\mathcal{G}$ of a CQ q is the undirected graph whose vertices are the variables appearing in $\mathsf{body}(q)$, and that contains an edge between x_1 and x_2 iff $P(x_1, x_2) \in \mathsf{body}(q)$ for some role P.[3] We call q *connected* (denoted with $q \in \mathsf{CQ^C}$) if $\mathcal{G}$ is connected, *linear* ($q \in \mathsf{CQ^L}$) if the degree of each vertex in $\mathcal{G}$ is ≤ 2, and *acyclic* ($q \in \mathsf{CQ^A}$) if $\mathcal{G}$ is acyclic. We note that none of these three notions implies any of the other two. In addition, following Nikolaou *et al.* [2019], we call a CQ *rooted* ($q \in \mathsf{CQ^R}$) if each connected component in $\mathcal{G}$ contains at least one constant or one distinguished variable. Finally, a CQ q is *atomic* ($q \in \mathsf{AQ}$) if $|\mathsf{body}(q)| = 1$.

Rewritability. Given a query language $\mathcal{Q}$, a $\mathcal{Q}$-rewriting of a CQ q with respect to a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a $\mathcal{Q}$ query q' whose answers over $\mathsf{inter}(\mathcal{A})$ alone coincides with the certain answers to q over $\mathcal{K}$. For instance, for OMQA with boolean or enumeration queries, $\mathcal{Q}$ is traditionally the language of domain independent first-order queries, the logical underpinning of SQL. As for queries with counting, it has been shown by Grumbach and Milo [1996], Nikolaou *et al.* [2019] that counting answers over a relational DB can be captured by query languages with evaluation in LogSpace (data complexity).

3 Related Work

Query answering under count semantics can be viewed as a specific case of query answering under *bag semantics*, investigated notably by Grumbach and Milo [1996] and Libkin and Wong [1997], but for relational DBs rather than KBs. Instead, in our setting, and in line with the OMQA/OBDA literature, we assume that the input ABox is a set rather than a bag. The counting problem over sets has also been studied recently in the DB setting [Pichler and Skritek, 2013; Chen and Mengel, 2016], but from the perspective of combined complexity, where the shape of the query (e.g., bounded treewidth) plays a prominent role.

As for (*DL-Lite*) KBs, Calvanese *et al.* [2008b] define an alternative (*epistemic*) count semantics, which counts over all grounded tuples (i.e., over $\mathsf{N_I}$) entailed by the KB. Such a semantics does not account for existentially implied individuals, and thus cannot capture the statistics motivating our work.

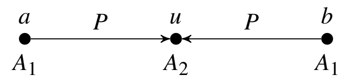
Instead, the work closest to ours, and which first introduced the count semantics that we adopt here, is the one of Kostylev and Reutter [2015], who first showed coNP-hardness of the Count problem for data complexity for *DL-Lite*$_{pos}$, with a reduction that uses a disconnected and cyclic query. coNP-membership is also shown for DLs up to *DL-Lite*$^{\mathcal{H}}_{core}$.

Nikolaou *et al.* [2019], Cima *et al.* [2019] have studied query answering over a KB under bag semantics, and provide a number of complexity results (including coNP-hardness) and query answering techniques (including a rewriting algorithm). Such semantics is clearly related to the count semantics, but there are notable differences as argued by Nikolaou *et al.* [2019]. In short, one cannot apply the intuitive idea of treating sets as bags with multiplicities 1. Hence algorithms and complexity

[3]This definition implies that the Gaifman graph of q has an edge from x to x if $P(x, x) \in \mathsf{body}(q)$.

results cannot be transferred between the two settings, and this already holds for ontology languages that allow for existential restrictions on the LHS of ontology axioms (note that *all* the logics considered in this paper allow for such construct). The following example by Nikolaou *et al.* [2019] illustrates this.

Example 1. Consider the KB $\mathcal{K} = \langle \{A_1 \sqsubseteq \exists P,\ \exists P^- \sqsubseteq A\}, \{A_1(a),\ A_1(b)\}\rangle$ and the query $q() \leftarrow A_2(y)$. If we evaluate this query in the count setting, then the answer is the empty tuple $\langle\rangle$ with cardinality 1, because of the following model:



However, such structure does not accurately represent a bag interpretation. In fact, under bag semantics every concept and property is associated to a bag of elements. Such bag can be seen as a function that returns, given an element, the number of times such element occurs in the bag. We build now a (minimal) bag interpretation $\mathcal{I}$ for $\mathcal{K}$. To satisfy $\mathcal{A}$, we set $A_1^{\mathcal{I}}(a) = 1$ and $A_1^{\mathcal{I}}(b) = 1$ To satisfy $A_1 \sqsubseteq \exists P$, we introduce a single element u (as above) and obtain $P^{\mathcal{I}}(a,u) = 1$ and $P^{\mathcal{I}}(b,u) = 1$. Therefore, $(P^-)^{\mathcal{I}}(u,a) = 1$ and $(P^-)^{\mathcal{I}}(u,b) = 1$, which, according to the semantics by Nikolaou *et al.* [2019], imply that $(\exists P^-)^{\mathcal{I}}(u) = 2$. Therefore, to satisfy $\exists P^- \sqsubseteq A_2$, it must be that $A_2^{\mathcal{I}}(u) = 2$. In fact, the certain answer to q over $\mathcal{I}$ under bag-semantics is the empty tuple $\langle\rangle$ with multiplicity 2.◂

4 Tractability and Intractability

We investigate now conditions for in/tractability (in data complexity) of COUNT, focusing on the impact of the shape of the query. We observe that the queries used by Kostylev and Reutter [2015] and Nikolaou *et al.* [2019] to show coNP-hardness are cyclic, and either disconnected or branching. Building upon these results, we further investigate whether cyclicity is necessary for intractability. Our results indicate that for certain DLs, non-connectedness or branching alone is a sufficient condition for intractability, whereas cyclicity is not. We start with a membership result:

Proposition 1. *COUNT is in P in data complexity for DL-Lite$_{pos}^{\mathcal{H}\mathcal{N}^-}$ and connected, linear CQs* ($\mathsf{CQ}^{\mathsf{CL}}$).

PROOF (SKETCH). We start with *DL-Lite*$_{pos}^{\mathcal{H}^-}$, and then discuss how to extend the proof to *DL-Lite*$_{pos}^{\mathcal{H}\mathcal{N}^-}$. If $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ is a *DL-Lite*$_{pos}^{\mathcal{H}\mathcal{N}^-}$ KB and q a boolean query in $\mathsf{CQ}^{\mathsf{CL}}$, consider the set $\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}})$ of all matches for $\mathsf{body}(q)$ over the canonical model $\mathcal{I}_{can}^{\mathcal{K}}$ of $\mathcal{K}$. Then viewing $\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}})$ as a relation (i.e., a set of tuples), let $F_{\min}$ be the set of all constant-preserving functions, whose domain is the set of all arguments in $\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}})$ and that minimize the number of resulting tuples when applied to $\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}})$. Because q is connected and linear, and thanks to the limited expressivity of *DL-Lite*$_{pos}^{\mathcal{H}\mathcal{N}^-}$, it can be shown that there must be some $f \in F_{\min}$ that verifies $|f(\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}}))| = |\mathsf{match}(q, f(\mathcal{I}_{can}^{\mathcal{K}}))|$. Since every model $\mathcal{I}$ of $\mathcal{K}$ verifies $\mathsf{match}(q, \mathcal{I}) \subseteq h(\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}}))$ for some homomorphism h, and because $f(\mathcal{I}_{can}^{\mathcal{K}})$ is a model of $\mathcal{K}$, it follows that $\mathsf{certCard}(q, \mathcal{K}) = |f(\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}}))|$. Then it can also be shown that $|f(\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}}))|$ can be computed in time polynomial in $|\mathcal{A}|$.

Now for *DL-Lite*$_{pos}^{\mathcal{H}\mathcal{N}^-}$, to account for cardinality restrictions, we associate in every interpretation $\mathcal{I}$ a cardinality $\mathsf{card}_{\mathcal{I}}(e)$ to each $e \in \Delta^{\mathcal{I}}$: cardinality 1 for elements of $\mathsf{N_I}$, and possibly more than 1 for elements of $\mathsf{N_E}$. E.g., if $\mathcal{K}$ implies that an element $a \in \mathsf{N_I}$ has 4 P-successors for some role P, and if there is only one $b \in \mathsf{N_I}$ s.t. $(a,b) \in P^{\mathcal{A}}$, then $(a,e) \in P^{\mathcal{I}_{can}^{\mathcal{K}}}$ for some $e \in \mathsf{N_E}$, and $\mathsf{card}_{\mathcal{I}_{can}^{\mathcal{K}}}(e) = 4 - 1 = 3$. Applying a function f to an interpretation $\mathcal{I}$ affects these cardinalities: for each $e \in \Delta^{f(\mathcal{I})}$, $\mathsf{card}_{f(\mathcal{I})}(e) = \max\{\mathsf{card}_{\mathcal{I}}(e') \mid f(e') = e\}$. Then we extend cardinality to a tuple $\mathbf{t}$ of elements, as $\mathsf{card}_{\mathcal{I}}(\mathbf{t}) = \prod_{e \in \mathbf{t}} \mathsf{card}_{\mathcal{I}}(e)$, and to a set T of tuples, as $\mathsf{card}_{\mathcal{I}}(T) = \sum_{\mathbf{t} \in T} \mathsf{card}_{\mathcal{I}}(\mathbf{t})$. In this extended setting, $\mathsf{certCard}(q, \mathcal{K}) = \mathsf{card}_{f(\mathcal{I})}(\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}}))$ for some function f that minimizes $\mathsf{card}_{f(\mathcal{I})}(\mathsf{match}(q, \mathcal{I}_{can}^{\mathcal{K}}))$. And this value can still be computed in time polynomial in $|\mathcal{A}|$. □

We now show that disconnectedness alone leads to intractability, i.e., cyclicity is not needed.

Proposition 2. *COUNT is coNP-hard in data complexity for DL-Lite$_{pos}$ and acyclic, linear, but disconnected CQs* ($\mathsf{CQ}^{\mathsf{AL}}$).

PROOF (SKETCH). The proof is a direct adaptation of the one provided by Kostylev and Reutter [2015]. We use a reduction from co-3-colorability to an instance of COUNT. Let $\mathcal{G} = \langle V, E\rangle$ be an undirected graph with vertices V, edges E, and without self-loops. The ABox is $\mathcal{A} = \{\mathsf{Vertex}(v) \mid v \in V\} \cup \{\mathsf{edge}(v_1, v_2) \mid (v_1, v_2) \in E\} \cup \{\mathsf{Blue(b)}, \mathsf{Green(g)}, \mathsf{Red(r)}, \mathsf{hasColor(a, b)}, \mathsf{hasColor(a, g)}, \mathsf{hasColor(a, r)}, \mathsf{edge(a, a)}\}$ for some fresh constants a, b, r, and g. The TBox is $\mathcal{T} = \{\mathsf{Vertex} \sqsubseteq \exists \mathsf{hasColor}, \exists \mathsf{hasColor}^- \sqsubseteq \mathsf{Color}\}$. And the (acyclic, non-branching) query is $q() \leftarrow \mathsf{Color}(c)$, $\mathsf{edge}(v_1, v_2)$, $\mathsf{hasColor}(v_1, c_1)$, $\mathsf{hasColor}(v_2, c_2)$, $\mathsf{Blue}(c_1)$, $\mathsf{Blue}(c_2)$, $\mathsf{edge}(v_3, v_4)$, $\mathsf{hasColor}(v_3, c_3)$, $\mathsf{hasColor}(v_4, c_4)$, $\mathsf{Green}(c_3)$, $\mathsf{Green}(c_4)$, $\mathsf{edge}(v_5, v_6)$, $\mathsf{hasColor}(v_5, c_5)$, $\mathsf{hasColor}(v_6, c_6)$, $\mathsf{Red}(c_5)$, $\mathsf{Red}(c_6)$. Then it can be verified that $4 = \mathsf{certCard}(q, \langle \mathcal{T}, \mathcal{A}\rangle)$ iff $\mathcal{G}$ is not 3-colorable. □

Next we show that linearity is required for tractability:

Proposition 3. *COUNT is coNP-hard in data complexity for DL-Lite$_{pos}^{\mathcal{H}}$ and acyclic, connected, but branching CQs* ($\mathsf{CQ}^{\mathsf{AC}}$).

Finally, we observe that the coNP upper bound provided by Kostylev and Reutter [2015] for *DL-Lite*$_{core}^{\mathcal{H}}$ extends to *DL-Lite*$_{core}^{\mathcal{H}\mathcal{N}^-}$,[4] since number restrictions can be encoded in *DL-Lite*$_{core}^{\mathcal{H}}$, as explained in Section 2.

Proposition 4. *COUNT is in coNP in data complexity for DL-Lite$_{core}^{\mathcal{H}\mathcal{N}^-}$ and arbitrary CQs* (CQ).

5 Rewritability and Non-rewritability

We now investigate conditions for rewritability. We start by showing P-hardness for DLs with role inclusions and disjointness, and atomic queries.

Proposition 5. *COUNT is P-hard in data complexity for DL-Lite$_{core}^{\mathcal{H}}$ and atomic queries* (AQ).

[4] With a technicality: the input integer k is not included in the notion of data complexity used by Kostylev and Reutter [2015].

PROOF (SKETCH). We show a LOGSPACE reduction from the co-problem of evaluating a boolean circuit where all gates are NAND gates [Greenlaw *et al.*, 1991] to COUNT. We view such a circuit as an interpretation C whose domain consists of the circuit inputs and gates. T_I, F_I, and T_O are unary predicates interpreted in C as the positive circuit inputs, the negative circuit inputs and the (unique) target gate respectively. P is a binary predicate s.t. $(q, g) \in P^C$ iff gate g has input q (where q can be either a circuit input or another gate).

The TBox $\mathcal{T}$ is defined by $\mathcal{T} = \mathcal{P} \cup \mathcal{T}_1 \cup \mathcal{T}_2$, where $\mathcal{P} = \{P_T \sqsubseteq P, P_F \sqsubseteq P\}$, $\mathcal{T}_1 = \{F_I \sqsubseteq F, T_I \sqsubseteq T, T_O \sqsubseteq T, T \sqsubseteq \neg F\}$ and $\mathcal{T}_2 = \{T \sqsubseteq \exists {P_F}^-, F \sqsubseteq (\geq_2 {P_T}^-), \exists P_T \sqsubseteq T, \exists P_F \sqsubseteq F\}$.[5] Intuitively, the unary predicates T and F correspond to gates that evaluate to true and false respectively in the circuit, and binary predicates P_T and P_F specialize P to positive and negative inputs. $\mathcal{T}_2$ encodes constraints pertaining to NAND gates: a positive gate must have at least one negative input, and a negative gate must have two positive inputs. Then $\mathcal{T}_1$ enforce that no gate can be both positive and negative, and that the circuit inputs and the output gate have the desired truth values.

Finally, as a technicality, the ABox $\mathcal{A}$ is an extension of C, i.e., $C \subseteq \mathsf{inter}(\mathcal{A})$. The domain of $\mathcal{A}$ contains 3 additional individuals t_1, t_2 and f, and it extends P^C with $\bigcup_{i \in (T_I)^C}\{P(f, i), P(t_1, i)\}$, $\bigcup_{i \in (F_I)^C}\{P(t_1, i), P(t_2, i)\}$, and $\{P(t_1, f), P(t_2, f), P(f, t_1), P(f, t_2), P(t_2, t_1), P(t_1, t_2)\}$.

Then it can be verified that C is a valid circuit iff there exists a model $\mathcal{I}$ of $\langle \mathcal{T}, \mathcal{A}\rangle$ s.t. $|P^{\mathcal{I}}| = |P^{\mathcal{A}}|$. Now let q be the query $q() \leftarrow P(x_1, x_2)$. It follows that C is not a valid circuit iff $|P^{\mathcal{A}}| + 1 = \mathsf{certCard}(q, \langle \mathcal{T}, \mathcal{A}\rangle)$. □

Assuming P $\not\subseteq$ LOGSPACE, this implies that for such DLs, even atomic queries cannot be rewritten into a query language whose evaluation is in LOGSPACE, which is sufficient to capture counting over relational databases. Interestingly, the reduction can be adapted so that it uses instead a query that is rooted, connected and linear (but not atomic).

Proposition 6. *COUNT is P-hard in data complexity for DL-Lite$^{\mathcal{H}}_{core}$ and rooted, connected, linear queries ($\mathsf{CQ}^{\mathsf{CLR}}$).*

We now focus on positive results, and rewriting algorithms.

5.1 Universal Model

We follow the notion of *universal model* proposed by Nikolaou *et al.* [2019]: a model $\mathcal{I}$ of a KB $\mathcal{K}$ is *universal* for a class of queries $\mathcal{Q}$ iff $\mathsf{ans}(q, \mathcal{I}) = \mathsf{certAns}(q, \mathcal{K})$ holds for every $q \in \mathcal{Q}$. Nikolaou *et al.* [2019] and Cima *et al.* [2019] investigated the existence of a universal model for queries evaluated under bag semantics. As we discussed in Section 3, these results carry over to the setting of count semantics, but only for ontology languages that disallow existential restriction on the LHS of ontology axioms. The existence of such model was proved over the class CQ^{R}, for the DL-Lite^b members up to *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ [Nikolaou *et al.*, 2019] and *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ [Cima *et al.*, 2019], with some syntactic restrictions. It was also shown that CQ^{R} queries can be rewritten into (*BCALC*) queries to be evaluated over the (bag) input ABox. Neither of these logics is able to encode numbers in the TBox though, therefore they cannot capture statistical information about missing data. And as discussed in the introduction, this information may be important in some applications [Chen and Mengel, 2016], and is one of the motivations behind our work. Note also that both logics allow for existentials on the LHS of axioms, and therefore these results do not carry over to count semantics.

Our first result shows the existence of a universal model for $\mathsf{CQ}^{\mathsf{CR}}$ and *DL-Lite*$^{\mathcal{N}^-}_{core}$, and queries evaluated under count semantics. Precisely, the canonical model $\mathcal{I}^{\mathcal{K}}_{can}$ obtained via the *restricted chase* from Calvanese *et al.* [2013] and Botoeva *et al.* [2010] is a universal model. From now on, we denote by $ch_i(\mathcal{K})$ the set of atoms obtained after applying the i-th chase step over the KB $\mathcal{K}$, and by $ch_\infty(\mathcal{K})$ the (possibly infinite) set of atoms obtained by an unbounded number of applications.

Proposition 7. *DL-Lite$^{\mathcal{N}^-}_{core}$ has a universal model w.r.t. COUNT over $\mathsf{CQ}^{\mathsf{CR}}$ queries.*

PROOF (SKETCH). Consider a $\mathsf{CQ}^{\mathsf{CR}}$ query q and a KB $\mathcal{K}$, and let $\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can})$ denote the set of all matches for $\mathsf{body}(q)$ over the canonical model $\mathcal{I}^{\mathcal{K}}_{can} := ch_\infty(\mathcal{K})$. Let $\mathcal{I}$ be a model of $\mathcal{K}$. Then there must exist a homomorphism τ from $\mathcal{I}^{\mathcal{K}}_{can}$ to $\mathcal{I}$. One immediately obtains that $\tau(\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can})) \subseteq \mathsf{match}(q, \tau(\mathcal{I}^{\mathcal{K}}_{can}))$, and therefore *(i)* $|\tau(\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can}))| \leq |\mathsf{match}(q, \tau(\mathcal{I}^{\mathcal{K}}_{can}))|$. Then relying on the fact that q is rooted, that the chase is restricted, and that *DL-Lite*$^{\mathcal{N}^-}_{core}$ does not allow for role subsumption, it can proven that *(ii)* $|\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can})| \leq |\tau(\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can}))|$. So from *(i)* and *(ii)* , $|\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can})| \leq |\mathsf{match}(q, \tau(\mathcal{I}^{\mathcal{K}}_{can}))|$ holds. Then, since τ is a homomorphism from $\mathcal{I}^{\mathcal{K}}_{can}$ to $\mathcal{I}$, $\tau(\mathcal{I}^{\mathcal{K}}_{can}) \subseteq \mathcal{I}$ must hold, and therefore $|\mathsf{match}(q, \tau(\mathcal{I}^{\mathcal{K}}_{can}))| \leq |\mathsf{match}(q, \mathcal{I})|$. We conclude that $|\mathsf{match}(q, \mathcal{I}^{\mathcal{K}}_{can})| \leq |\mathsf{match}(q, \mathcal{I})|$. □

5.2 Rewriting for *DL-Lite*$^{\mathcal{N}^-}_{core}$

We introduce $\mathsf{PerfectRef_{cnt}}$, a rewriting algorithm for *DL-Lite*$^{\mathcal{N}^-}_{core}$ inspired by $\mathsf{PerfectRef}$ [Calvanese *et al.*, 2006], and show its correctness. There is a fundamental complication in our setting, of which we provide an example. Consider a CQ q, a *DL-Lite*$^{\mathcal{N}^-}_{core}$ KB $\mathcal{K}$, and a query q' among those produced by $\mathsf{PerfectRef}$ or any other rewriting algorithm for CQs. Then, each match ω' for q' in $\mathsf{inter}(\mathcal{A})$ can be extended to the anonymous individuals so as to form a "complete" match ω for q in $\mathsf{inter}(ch_\infty(\mathcal{K}))$ in a certain number of ways (dictated by the axioms in the ontology). From now on, we call such number the *anonymous contribution relative to q'*. The following example shows that the anonymous contribution is related to the number restrictions occurring in $\mathcal{K}$.

Example 2. Consider the query $q(x) \leftarrow P(x, y)$, and the KB $\mathcal{K} = \langle \{A \sqsubseteq \geq_3 P\}, \{A(a)\}\rangle$. Starting from q, $\mathsf{PerfectRef}$ will produce, as part of the final rewriting, a query $q'(x) \leftarrow A(x)$. Note that there is a single match $\mu = \{x \mapsto a\}$ for q' over $\mathsf{inter}(\mathcal{A})$, and that μ can be extended into exactly three matches for q in $\mathsf{inter}(ch_\infty(\mathcal{K}))$, by mapping variable y into some anonymous individual. ◁

To deal with the fact that the anonymous contribution to a count is a non-fixed quantity that depends on the axioms in the ontology, our algorithm is substantially different from $\mathsf{PerfectRef}$ and significantly more complicated. It is also

[5] The axiom $\geq_2 {P_T}^-$ can be encoded into *DL-Lite*$^{\mathcal{H}}_{core}$, as explained in Section 2.

not related to the one by Nikolaou *et al.* [2019], which is based on *tree-witness rewriting* [Kikot *et al.*, 2012] rather than on PerfectRef, and was not designed for settings where the anonymous contribution is a non-fixed quantity.

Given a CQ q and a TBox $\mathcal{T}$, $\mathsf{PerfectRef_{cnt}}$ produces a *set* Q' of queries such that, for any ABox $\mathcal{A}$, $\mathsf{ans}(Q', \mathsf{inter}(\mathcal{A})) = \mathsf{certAns}(q, \langle\mathcal{T}, \mathcal{A}\rangle)$. Each query in Q' comes with a *multiplicative factor* that captures the anonymous contribution of each match for that query. Queries in Q' are expressed in a *target (query) language*, for convenience named FO(COUNT), which is a substantial enrichment of the one introduced in Section 2, but has a straightforward translation into SQL. Note that we use FO(COUNT) only to express the rewriting, while user queries over the KB are still plain CQs.

Following Cohen *et al.* [2007], FO(COUNT) allows one to explicitly specify a subset of the non-distinguished variables, called *aggregation variables*, for which we count the number of distinct mappings (so far, we implicitly assumed all non-distinguished variables as aggregation variables). It also allows for a *multiplicative factor* to be applied after the (count) aggregation operator, a restricted use of disjunctions, equalities between terms, atomic negation, and the use of *nested aggregation* in the form of a special $\exists^{=i}$ operator (which intuitively corresponds to a nested aggregation plus a boolean condition requiring the result of the aggregation to be equal to i).

A *query* in FO(COUNT) is a pair $\langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi\rangle$, where variables $\mathbf{x}$ are called *group-by variables*, variables $\mathbf{y}$ are called *aggregation variables* (intuitively, $\mathsf{cnt}(\mathbf{y})$ corresponds to the SQL construct COUNT DISTINCT), $\mathbf{x} \cap \mathbf{y} = \emptyset$, $\alpha \in \mathbb{N}$ is a positive multiplicative factor, and Π is a set of *rules* $\{q^k(\mathbf{x} : \mathbf{y}) \leftarrow \psi^k \mid 1 \le k \le m\}$. The symbol ':' in the head[6] of each rule is to distinguish between group-by and aggregation variables. Each ψ^k in Π is a conjunction $\psi^k_{pos} \wedge \psi^k_{neg} \wedge \psi^k_{eq} \wedge \psi^k_{\exists}$ of positive atoms (ψ^k_{pos}), negated atoms (ψ^k_{neg}), equalities between terms (ψ^k_{eq}), and special atoms ($\psi^k_{\exists}$), which we call $\exists$*-atoms*, of the form $\exists^{=i}_z. P(w, z)$, where $i \in \mathbb{N}_0$, $w \in \mathbf{x} \cup \mathbf{y}$, and z is a variable that occurs only once in q.

A mapping ρ is a match for ψ^k in an interpretation $\mathcal{I}$ if:

- $\rho(\psi^k_{pos}) \subseteq \mathcal{I}$;
- ρ satisfies all equalities in ψ^k_{eq};
- there is no $\rho' \supseteq \rho$ such that $\rho'(E(\mathbf{z})) \in \mathcal{I}$, for some $\neg E(\mathbf{z})$ in ψ^k_{neg};
- for each $\exists^{=i}_y R(w, z)$ in $\psi_\exists$, there are *exactly* i mappings $\rho_1, \ldots, \rho_i$ such that, for $j \in \{1, \ldots, i\}$ we have that $\rho_j \supseteq \rho$ and $\rho_j(R(w, z)) \subseteq \mathcal{I}$.

A mapping ρ is a match for Π in an interpretation $\mathcal{I}$, if for some $q(\mathbf{x} : \mathbf{y}) \leftarrow \psi$ in Π it is a match for ψ in $\mathcal{I}$. A mapping ω is an *answer* to $\langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi\rangle$ over $\mathcal{I}$ with cardinality $k \cdot \alpha$ iff there are *exactly* k mappings $\eta_1, \ldots, \eta_k$ such that, for $i \in \{1, \ldots, k\}$:

- $\omega = \eta_i|_{\mathbf{x}}$, and
- η_i can be extended to a match ρ for Π in $\mathcal{I}$ s.t. $\rho|_{\mathbf{x}\cup\mathbf{y}} = \eta_i$.

Note that our semantics also captures the case when the operator cnt() is over an empty set of variables (in that case, the k above would be equal to 1). This technicality is necessary for the presentation of the algorithm.

[6] Head and body of a rule are defined as for CQs.

We are now ready to introduce $\mathsf{PerfectRef_{cnt}}$. Consider a *satisfiable* knowledge base $\mathcal{K} = \langle\mathcal{T}, \mathcal{A}\rangle$, and a $\mathsf{CQ^{CR}}$ query $q(\mathbf{x}) \leftarrow \psi(\mathbf{x}, \mathbf{y})$. $\mathsf{PerfectRef_{cnt}}$ takes as input q and $\mathcal{T}$ and initializes the result set Q as

$$\{\langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot 1), \{q(\mathbf{x} : \mathbf{y}) \leftarrow \psi(\mathbf{x}, \mathbf{y})\}\rangle\}.$$

Then the algorithm expands Q by applying the rules *AtomRewrite*, *Reduce*, GE_α, and GE_β until saturation, with priority for *AtomRewrite* and *Reduce*. The set Q' obtained at the end of this process does not necessarily contain just queries (in the sense of our definition above), and hence needs to be *normalized* (see later).

To define the rules of the algorithm, we first need to introduce some notation. In the following, $P^-(w, z)$ stands for $P(z, w)$. Hence, also $R(w, z)$ when $R = P^-$ stands for $P(z, w)$. We use '_' to denote a fresh variable introduced during the execution of the algorithm. For a basic concept B, notation $\xi(B, w)$ stands for $B(w)$ if $B \in \mathsf{N_C}$, or $R(w, \_)$, if $B = \ge_1 R$. Given a set $\mathcal{B}$ of basic concepts, $\mathsf{subc}_\mathcal{T}(\mathcal{B})$ is defined as the set of basic concepts $\{B' \mid B' \sqsubseteq_\mathcal{T} B, B \in \mathcal{B}\}$. If ϕ, ψ are two conjunctions of atoms and a is an atom in ϕ, we use $\phi[a/\psi]$ (resp., $\phi[a/\top]$) to denote the conjunction identical to ϕ, but where a is replaced with ψ (resp., a is deleted from ϕ). By extension, if r is a rule, $r[a/\psi]$ denotes the rule $\mathsf{head}(r) \leftarrow \mathsf{body}(r)[a/\psi]$. If B is a basic concept and R a role, $\mathsf{card}_\mathcal{T}(B, R)$ denotes the maximal n s.t. $B \sqsubseteq \ge_n R \in \mathcal{T}$. A variable x is *bound in a rule* r if it is a group-by variable, or if it occurs more than once in the set of positive atoms of r. We say that x is α*-blocked* if it is bound, or if it occurs more than once in $\mathsf{head}(r)$, or if it occurs in some $\exists$-atom in $\mathsf{body}(r)$. Finally, x is β*-blocked* if it is bound, or if it occurs more than once in $\mathsf{head}(r)$, or if it occurs in some atom of the form $\exists^{=i}_z R(w, z)$ with $i > 0$.

To ease the presentation, for the exposition of rules GE_α and GE_β we will ignore details concerning the renaming of variables, and assume that variables belong to the input query.

AtomRewrite **($\rightsquigarrow_{AR}$).** $\{s\mathfrak{q}_1, \ldots, \mathfrak{q}_k\} \rightsquigarrow_{AR} \{\mathfrak{q}_1, \ldots, \mathfrak{q}_{k-1}, \mathfrak{q}'_k\}$ if

- $\mathfrak{q}_k = \langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi\rangle$;
- for some $r \in \Pi$, for some $E(\mathbf{z}) \in \mathsf{body}(r)$, either:
  - $E(\mathbf{z})$ is of the form $A(z)$, and $B \sqsubseteq A \in \mathcal{T}$, or
  - $E(\mathbf{z})$ is of the form $R(w, z)$, $B \sqsubseteq \ge_n R \in \mathcal{T}$, z is an unbound variable, and if $\mathsf{head}(r) = q(\mathbf{s} : \mathbf{t})$, then $z \notin \mathbf{t}$;

  $\mathfrak{q}'_k = \langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi \cup \{r[E(\mathbf{z})/\xi(B, w)]\}\rangle$.

Reduce **($\rightsquigarrow_R$).** $\{\mathfrak{q}_1, \ldots, \mathfrak{q}_k\} \rightsquigarrow_R \{\mathfrak{q}_1, \ldots, \mathfrak{q}_{k-1}, \mathfrak{q}'_k\}$ if

- $\mathfrak{q}_k = \langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi\rangle$;
- $\{E_1(\mathbf{z_1}), E\mathbf{z_2})\} \subseteq \mathsf{body}(r)$ for some $r \in \Pi$;
- σ is a most general unifier for $E(\mathbf{z}_1)$ and $E(\mathbf{z}_2)$; with the following restrictions:
  - a variable in $\mathbf{x}$ can map only to a variable in $\mathbf{x}$;
  - a variable in $\mathbf{y}$ can map only to a variable in $\mathbf{x} \cup \mathbf{y}$;
  - $\mathsf{dom}(\sigma) \subseteq \mathbf{z}_1 \cup \mathbf{z}_2$ and $\mathsf{range}(\sigma) \subseteq \mathbf{z}_1 \cup \mathbf{z}_2$.
- $\mathfrak{q}'_k = \langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi \cup \{\sigma(r[E(\mathbf{z}_2)/\top])\}\rangle$.

$\boldsymbol{GE_\alpha}$ **($\rightsquigarrow_{\ge_\alpha}$).** $\{\mathfrak{q}_1, \ldots, \mathfrak{q}_k\} \rightsquigarrow_{\ge_\alpha} \{\mathfrak{q}_1, \ldots, \mathfrak{q}_k\} \cup Q_k$ if

- $\mathfrak{q}_k = \langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi\rangle$

★ $R(w, z)$, with $w, z \in \mathbf{x} \cup \mathbf{y}$, is an atom such that

$$\Pi' := \left\{ R(w, z) \in \mathsf{body}(r) \;\middle|\; \begin{array}{l} r \in \Pi \text{ and} \\ y \text{ is a non-}\alpha\text{–blocked} \\ \text{aggregation variable} \end{array} \right\} \neq \emptyset$$

- Let $\psi_\exists$ be the conjunction of all exists-atoms in any rule $r \in \Pi$ (by construction, such conjunction is the same for all rules in Π). Then the conjunction $\psi_\exists \wedge \exists^0_y R(w,z)$ (seen as a set) must not appear in other rules from $\{\mathfrak{q}_1,\ldots,\mathfrak{q}_k\}$;
- $\mathcal{B}_k$ is the maximal set of basic concepts B such that $B \sqsubseteq \geq_{n_B} R \in \mathcal{T}$, for some n_B;
- ⋄ Q_k is defined as follows. First, let $\mathsf{part}(\mathcal{B}_k)$ denote the set of all pairs $\langle \mathcal{B}^1, \mathcal{B}^2 \rangle$ such that $\mathcal{B}^1 \subseteq \mathcal{B}_k$, $\mathcal{B}^2 \subseteq \mathcal{B}_k$, $\mathcal{B}^1 \neq \emptyset$, and $\mathsf{subc}_{\mathcal{T}}(\mathcal{B}^1) \cap \mathsf{subc}_{\mathcal{T}}(\mathcal{B}^2) = \emptyset$. Then, for a set $\mathcal{B}$ of basic concepts, let $\mathsf{cp}_{\mathcal{T}}(\mathcal{B})$ denote the cartesian product $\prod_{B \in \mathcal{B}} \mathsf{subc}_{\mathcal{T}}(B)$. And if $\mathcal{B}, \mathcal{B}'$ are two sets of basic concepts, we call *atomic decomposition* the formula $\mathsf{ad}(\mathcal{B}, \mathcal{B}')$, defined as:

  $$\bigwedge_{B \in \mathcal{B}} \xi(B,w) \wedge \bigwedge_{B \in \mathcal{B}', B' \in\, \mathsf{subc}_{\mathcal{T}}(B)} \neg \xi(B', w)$$

  If ψ is a formula, let $\mathsf{rpl}(r, \psi)$ designate the rule:

  $$q(\mathbf{s} : \mathbf{t} \setminus \{y\}) \leftarrow \mathsf{body}(r)[R(w,z)/\psi]$$

  Finally, if j is an integer, let $\mathsf{qh}(j)$ be the expression:

  $$Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y} \setminus \{y\}) \cdot j \cdot \alpha)$$

  We can now define Q_k as:

  $$\bigcup_{\substack{(\mathcal{B}^1,\mathcal{B}^2) \in \mathsf{part}(\mathcal{B}_k),\\ n = \max_{B \in \mathcal{B}^1} \mathsf{card}(B),\\ i \in [0..n-1]}} \{\langle \mathsf{qh}(n{-}i), \{\mathsf{rpl}(r, \mathsf{ad}(\mathcal{B}^3, \mathcal{B}^2) \wedge \exists^{=i}_y R(w,z)) \mid \mathcal{B}^3 \in \mathsf{cp}_{\mathcal{T}}(\mathcal{B}^1), r \in \Pi'\}\rangle\}$$

***GE*$_\beta$ ($\rightsquigarrow_{\geq_\beta}$).** This rule is defined as $\rightsquigarrow_{\geq_\alpha}$, but with the difference that conditions ⋆ and ⋄ are as follows:

- ⋆ $R(w,z)$ is an atom such that:

  $$\Pi' := \left\{ R(w,z) \in \mathsf{body}(r) \;\middle|\; \begin{array}{l} r \in \Pi \text{ and} \\ y \text{ is a non-}\beta\text{–blocked} \\ \text{aggregation variable} \end{array} \right\} \neq \emptyset$$

- ⋄ As item ⋄ for GE_α, with the additional condition that all atoms in which variable y occurs are removed.

Note that, once all rules have been applied to saturation, the resulting set $\boldsymbol{Q}'$ is technically not yet a set of queries, because of renamed variables, constants, or repetitions in the head of a rule. To transform each element $\langle Q(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi \rangle$ of $\boldsymbol{Q}'$ into a query, we *normalize* it by renaming the variables in rules in Π, based on their positions, according to $\mathbf{x}$ and $\mathbf{y}$, and by replacing constants and repeated variables in the head of a rule with suitable equalities in its body.

The intuition behind $\mathsf{PerfectRef}_{\mathsf{cnt}}$ is the following. First of all, we observe that the rewriting rules *AtomRewrite* and *Reduce* are analogous to their counterparts in the original PerfectRef algorithm. The restrictions on the unifier in *Reduce* are meant to limit the possible renamings of variables. Rewriting rules GE_α and GE_β extend the way existential quantification is handled in PerfectRef, and are the only ones eliminating aggregation variables from the rules in Π. Each time one such variable is eliminated, it can be potentially mapped in $(n{-}i)$ different ways into the anonymous part of the canonical model. The $\exists$-atoms, together with the relative atomic decompositions, check the number i of mappings that are already present in the ABox. The factor α keeps track of the number of ways variables eliminated in previous steps can be mapped into the anonymous part. Hence, the quantity $(n{-}i) \cdot \alpha$ captures the anonymous contribution relative to the query.

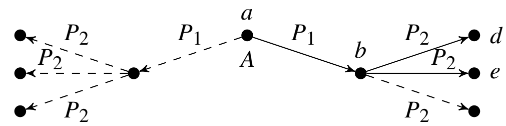


Figure 2: Chase model of Example 3. Solid arrows represent the information in the ABox, whereas dashed lines represent information implied by the ontology.

Example 3. Consider the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, with

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geq_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geq_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$

and the input CQ $q(x) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)$. The chase model of $\mathcal{K}$ is represented in Figure 2. The initialization step sets $\boldsymbol{Q} = \{\langle Q(x, \mathsf{cnt}(y_1,y_2) \cdot 1), \{q(x{:}y_1,y_2) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)\}\rangle\}$. Since y_2 is unbound, we can apply rule GE_α, which produces the following set $\boldsymbol{Q}'$:

$$\left\{ \begin{array}{l} \langle Q(x, \mathsf{cnt}(y_1,y_2) \cdot 1), \\ \quad \{q(x : y_1, y_2) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)\}\rangle, \\ \langle Q(x, \mathsf{cnt}(y_1) \cdot 3 - 0), \\ \quad \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists^{=0}_z P_2(y_1,z)\}\rangle, \\ \langle Q(x, \mathsf{cnt}(y_1) \cdot 3 - 1), \\ \quad \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists^{=1}_z P_2(y_1,z)\}\rangle, \\ \langle Q(x, \mathsf{cnt}(y_1) \cdot 3 - 2), \\ \quad \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists^{=2}_z P_2(y_1,z)\}\rangle \end{array} \right\}$$

Rule *Reduce* can now be triggered by the second, the third, and the last rule in $\boldsymbol{Q}'$. In particular, an application of *Reduce* on the second query leads to the query:

$$\begin{array}{l} \langle Q(x, \mathsf{cnt}(y_1) \cdot 3 - 0), \\ \quad \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists^{=0}_z P_2(y_1,z), \\ \quad\; q(x : y_1) \leftarrow A(x), P_1(x,y_1), \exists^{=0}_z P_2(y_1,z)\}\rangle \end{array}$$

On such query we can apply rule GE_β producing, among others, the following query:

$$\langle Q(x, 1 \cdot (2-1) \cdot 3), \{q(x : \langle\rangle) \leftarrow A(x), \exists^{=1}_z P_1(x,z)\}\rangle$$

Let us analyze the queries produced by $\mathsf{PerfectRef}_{\mathsf{cnt}}$ that return at least one answer. The query after the initialization step returns the number of paths (x, y_1, y_2) in $\mathcal{A}$ conforming to the structure dictated by the body of the input query. Since there are two such paths, such query returns the answer $\langle x \mapsto a, 2\rangle$. The queries generated by GE_α check for all sub-paths (x, y_1) of (x, y_1, y_2) such that x is an element of A, y_1 is a P_1-successor of x, and y_1 has less P_2-successors in the ABox than what the TBox prescribes. There is one such path in $\mathcal{I}^{\mathcal{K}}_{can}$, namely the one terminating in node b that has only two P_2-successors in $\mathcal{A}$. This path is captured by the fourth query in $\boldsymbol{Q}'$, which returns as answer $\langle x \mapsto a, 1\rangle$: indeed, there is a single way of extending this path into the anonymous part. The queries generated by GE_β are to be interpreted in a similar way. In particular, the query we highlighted retrieves the individual a, since this node has only one P_1-successor in $\mathcal{A}$ but it should have at least two P_1-successors according to $\mathcal{T}$. The answer to such query is $\langle x \mapsto a, 3\rangle$. Indeed, there are three ways of extending the match $x \mapsto a$ into the anonymous part. Summing up the numbers, we get that our set of queries returns the answer $\langle x \mapsto a, 6\rangle$, which indeed is the answer to our input query over the chase model from Figure 2. ◁

The algorithm terminates because the application of *AtomRewrite* and *Reduce* is blocked upon reaching saturation, and each application of GE_α and GE_β reduces the number of variables in ψ by 1. The following lemmas show the correctness of $\mathsf{PerfectRef}_{\mathsf{cnt}}$.

Lemma 8. *Consider a DL-Lite*$^{\mathcal{N}^-}_{core}$ *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *and a connected, rooted CQ q. Consider a query* Q' *belonging to the output of* $\mathsf{PerfectRef}_{\mathsf{cnt}}$ *over q and* $\mathcal{K}$*. Then, each match* ω' *for* Q' *in* $\mathsf{inter}(\mathcal{A})$ *can be extended into a match* ω *for q in* $\mathsf{inter}(ch_\infty(\mathcal{K}))$.

PROOF (SKETCH). The claim can be proved through a straightforward induction over the number of chase steps. □

The next lemma states that the opposite direction also holds, i.e., that *all* matches are retrieved.

Lemma 9. *Consider a DL-Lite*$^{\mathcal{N}^-}_{core}$ *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *and a connected, rooted CQ q. Every match* ω *for q in* $\mathsf{inter}(ch_\infty(\mathcal{K}))$ *is an extension of some match* ω' *for* Q' *in* $\mathsf{inter}(\mathcal{A})$*, where* Q' *belongs to the output of* $\mathsf{PerfectRef}_{\mathsf{cnt}}$.

PROOF (SKETCH). The proof follows the one by Calvanese *et al.* [2006], however one has to pay attention to the fact that here we deal with matches rather than with assignments for the distinguished variables. Another technical difference with that proof is that in our case the nodes in the *chase tree* are sets of atoms rather than single atoms. □

The last lemma tells us that our way of capturing the anonymous contribution is indeed correct.

Lemma 10. *Consider a DL-Lite*$^{\mathcal{N}^-}_{core}$ *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *and a connected, rooted CQ q. Consider a query* $\langle Q'(\mathbf{x}, \mathsf{cnt}(\mathbf{y}) \cdot \alpha), \Pi\rangle$ *belonging to the output of* $\mathsf{PerfectRef}_{\mathsf{cnt}}$ *over q and* $\mathcal{K}$*. Then, each match* ω' *for* Q' *in* $\mathsf{inter}(\mathcal{A})$ *can be extended into exactly* α *matches* ω *for q in* $\mathsf{inter}(ch_\infty(\mathcal{K}))$ *with* $\mathsf{range}(\omega \setminus \omega') \subseteq \mathsf{N_E}$.

PROOF (SKETCH). By induction on the number of applications of GE_α and GE_β. It uses Lemma 8 and the fact that variables are never eliminated by *AtomRewrite* or *Reduce*. The atomic decomposition in GE_α and GE_β guarantees that all combinations of number restrictions are considered. □

Proposition 11. *Consider a DL-Lite*$^{\mathcal{N}^-}_{core}$ *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *and a connected, rooted CQ q. Let* $\mathcal{Q}$ *be the set of queries returned by a run of* $\mathsf{PerfectRef}_{\mathsf{cnt}}$ *over q and* $\mathcal{K}$*. Then:*

$$\mathsf{ans}(\mathcal{Q}, \mathsf{inter}(\mathcal{A})) = \mathsf{certAns}(q, \mathcal{K})$$

PROOF (SKETCH). The claim follows from Lemmas 9 and 10, and by observing that the query Q' in Lemma 10 is unique due to $\exists^{=}i$ expressions, atomic decompositions, and the restrictions on GE_α and GE_β. Therefore, matches are not counted twice.□

The execution of $\mathsf{PerfectRef}_{\mathsf{cnt}}$ does not depend on the ABox. Considering that the evaluation of FO(COUNT) queries is in LOGSPACE in data complexity, this yields:

Proposition 12. *COUNT is in LOGSPACE in data complexity for DL-Lite*$^{\mathcal{N}^-}_{core}$ *and rooted, connected CQs.*

	AQ,CQ^{CL}	CQ^{AC}	CQ^{CLR},CQ^{CR}	CQ^{AL}	CQ
DL-Lite$_{pos}$	P	coNP	L	coNP-c	coNP-c
DL-Lite$^{\mathcal{H}}_{pos}$	P	coNP-c	coNP	coNP-c	coNP-c
DL-Lite$^{\mathcal{H}\mathcal{N}^-}_{pos}$	P	coNP-c	coNP	coNP-c	coNP-c
DL-Lite$_{core}$	coNP	coNP	L	coNP-c	coNP-c
DL-Lite$^{\mathcal{N}^-}_{core}$	coNP	coNP	L	coNP-c	coNP-c
DL-Lite$^{\mathcal{H}}_{core}$	P-h/coNP	P-h/coNP	P-h/coNP	coNP-c	coNP-c

Table 1: Summary of complexity results ('-h' stands for '-hard', and '-c' for '-complete'). New bounds proved here are in blue, bounds that directly follow in green, and already known bounds in black.

6 Conclusion and Perspectives

Table 1 summarizes our results for data complexity of query answering under count semantics for variants of CQs and *DL-Lite*. Among other observations, these results indicate that for certain DLs, whether a CQ is connected and branching affects tractability. An interesting open question in this direction is whether the P-membership result for *DL-Lite*$^{\mathcal{H}\mathcal{N}^-}_{pos}$ and AQ/CQ^{CL} is tight. Indeed, the P-hardness result provided for AQ holds for a more expressive DL (namely *DL-Lite*$^{\mathcal{H}}_{core}$), which allows for disjointness and arbitrary interactions between role subsumption and existential quantification.

A main contribution of this work is the query rewriting technique provided in Section 5. It shows that for connected and rooted CQs, and for variants of *DL-Lite* with neither disjointness nor role subsumption, rewritability into a variant of SQL with aggregates can be regained. An interesting open question is whether rewritability still holds for rooted queries and *DL-Lite*$^{\mathcal{H}\mathcal{N}^-}_{core}$, i.e., when allowing for restricted role subsumption.

Finally, it must be emphasized that this work is mostly theoretical, and does not deliver a practical algorithm for query answering under count semantics over *DL-Lite* KBs. In particular, the definition of data complexity that we adopted does not take into account the cardinality restrictions that may appear in the TBox. This is arguable: in scenarios where these restrictions may encode statistics, it is reasonable to consider that these numbers "grow" with the size of the data. The rewriting defined in Section 5 may produce a query whose size is exponential in such numbers (when they are encoded in binary). Therefore a natural continuation of this work is to investigate how arithmetic operations and nested aggregation can be used to yield a rewriting whose size does not depend on the numbers that appear in cardinality restrictions.

Acknowledgements

This research has been partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, by the Italian Basic Research (PRIN) project HOPE, by the EU H2020 project INODE, grant agreement 863410, by the CHIST-ERA project PACMEL, and by the project IDEE (FESR1133) through the European Regional Development Fund (ERDF) Investment for Growth and Jobs Programme 2014-2020.

References

[Artale *et al.*, 2009] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The *DL-Lite* family and relations. *J. of Artificial Intelligence Research*, 36:1–69, 2009.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web: Web Logic Rules – 11th Int. Summer School Tutorial Lectures (RW)*, volume 9203 of *LNCS*, pages 218–307. Springer, 2015.

[Bienvenu *et al.*, 2017] Meghyn Bienvenu, Stanislav Kikot, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. On the parametrised complexity of tree-shaped ontology-mediated queries in OWL 2 QL. In *Proc. of the 30th Int. Workshop on Description Logics (DL)*, volume 1879 of *CEUR Workshop Proc.*, http://ceur-ws.org/, 2017.

[Botoeva *et al.*, 2010] Elena Botoeva, Alessandro Artale, and Diego Calvanese. Query rewriting in *DL-Lite*$^{\mathcal{HN}}_{horn}$. In *Proc. of the 23rd Int. Workshop on Description Logics (DL)*, volume 573 of *CEUR Workshop Proc.*, http://ceur-ws.org/, pages 267–278, 2010.

[Calvanese *et al.*, 2006] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*, pages 260–270, 2006.

[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. *J. of Automated Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2008a] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Conjunctive query containment and answering under description logics constraints. *ACM Trans. on Comp. Logic*, 9(3):22.1–22.31, 2008.

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

[Calvanese *et al.*, 2013] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. *Artificial Intelligence*, 195:335–360, 2013.

[Calvanese *et al.*, 2020] Diego Calvanese, Julien Corman, Davide Lanti, and Simon Razniewski. Counting query answers over a DL-Lite knowledge base (Extended version). CoRR Tech. Rep. arXiv:2005.05886, arXiv.org e-Print archive, 2020. Available at http://arxiv.org/abs/2005.05886.

[Chen and Mengel, 2016] Hubie Chen and Stefan Mengel. Counting answers to existential positive queries: a complexity classification. In *Proc. of the 35th ACM Symp. on Principles of Database Systems (PODS)*, pages 315–326, 2016.

[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 *Proc. of the 18th Int. Sem. Web Conf. (ISWC)*, volume 11778 of *LNCS*, pages 128–144. Springer, 2019.

[Cohen *et al.*, 2007] Sara Cohen, Werner Nutt, and Yehoshua Sagiv. Deciding equivalences among conjunctive aggregate queries. *J. of the ACM*, 54(2):5, 2007.

[Greenlaw *et al.*, 1991] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. A compendium of problems complete for P. Technical report, Dep. of Computer Science, Univ. of New Hampshire, 1991.

[Grumbach and Milo, 1996] Stéphane Grumbach and Tova Milo. Towards tractable algebras for bags. *J. of Computer and System Sciences*, 52(3):570–588, 1996.

[Kikot *et al.*, 2012] Stanislav Kikot, Roman Kontchakov, and Michael Zakharyaschev. Conjunctive query answering with OWL 2 QL. In *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*, pages 275–285, 2012.

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

[Libkin and Wong, 1997] Leonid Libkin and Limsoon Wong. Query languages for bags and aggregate functions. *J. of Computer and System Sciences*, 55(2):241–272, 1997.

[Motik *et al.*, 2012] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. OWL 2 Web Ontology Language profiles (2nd ed.). W3C Rec., W3C, 2012. http://www.w3.org/TR/owl2-profiles/.

[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*, 274:91–132, 2019.

[Pichler and Skritek, 2013] Reinhard Pichler and Sebastian Skritek. Tractable counting of the answers to conjunctive queries. *J. of Computer and System Sciences*, 79(6):984–1001, 2013.

[Vardi, 1982] Moshe Y. Vardi. The complexity of relational query languages. In *Proc. of the 14th ACM Symp. on Theory of Computing (STOC)*, pages 137–146, 1982.

[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 *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 5511–5519. IJCAI Org., 2018.