Counting Query Answers over a DL-Lite Knowledge Base

Diego Calvanese¹,², Julien Corman¹, Davide Lanti¹ and Simon Razniewski³
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 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

¹Also referred to as OBDA (for Ontology Based Data Access), when emphasis is 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 by Nikolau 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 Calvanese et al. [2013], but in the context of query answering under bag semantics. For certain DL-Lite variants, it is shown that queries that are rootable (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 $N_I$ of individuals (a.k.a. constants), $N_V$ of anonymous individuals (induced by existential quantification), $N_C$ of variables, $N_K$ of concept names (i.e., unary predicates, denoted with $A$), and $N_R$ of role names (i.e., binary predicates, denoted with $P$). In the following, boldface letters, e.g., $c$, denote tuples, and we treat tuples as sets.

**Functions, Atoms.** $\text{dom}(f)$ and $\text{range}(f)$ denote the domain and range of a function $f$. Given $D \subseteq \text{dom}(f)$, 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 $c \in \text{dom}(f) \cap N_I$. If $S \subseteq \text{dom}(f)$, we use $f(S)$ for $\{f(s) \mid s \in S\}$. If $t = (t_1, \ldots, t_n)$ is a tuple with elements in $\text{dom}(f)$, we use $f(t)$ for $\{f(t_1), \ldots, f(t_n)\}$.

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

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

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

Given an interpretation $\mathcal{I}$ and a constant-preserving function $f$ with domain $\Delta_I$, we use $f(\cdot)$ to denote the interpretation defined by $\Delta_I^f = f(\Delta_I)$ and $E_I^f = f(E_I)$ for each $E_I \subseteq N_C \cup N_R$. Given two interpretations $\mathcal{I}_1, \mathcal{I}_2$, we use $\mathcal{I}_1 \equiv \mathcal{I}_2$ as a shortcut for $\Delta_I^1 \equiv \Delta_I^2$ and $E_I^1 \equiv E_I^2$, for each $E_I \subseteq N_C \cup N_R$. A homomorphism $h$ from $\mathcal{I}_1$ to $\mathcal{I}_2$ is a constant-preserving function with domain $\Delta_I^1$ that verifies $h(I_1) \subseteq I_2$. We note that a set $S$ of atoms with arguments in $N_I \cup N_E$ uniquely identifies an interpretation, which we denote with $\text{inter}(S)$.

**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 $N_I$ 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) $B \subseteq C$ (concept inclusion), (ii) $B \subseteq \neg C$ (concept disjointness), or (iii) $R \subseteq R'$ (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 $^\mathcal{K}$; (3) allowing for arbitrary numbers in number restrictions, but only on the right-hand-side (RHS) of concept inclusion, indicated by adding a superscript $^\mathcal{K}$. We also use the superscript $^\mathcal{K}$ for logics with role inclusions, but with the restriction on TBoxes defined by Nikolau et al. [2019], which disallows in a TBox axioms of the form $B \subseteq \geq n R_I$ if $\mathcal{T}$ contains a role inclusion $R_1 \subseteq R_2$, 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 $\text{inter}(\mathcal{A}) \subseteq \mathcal{I}$, and $E^\mathcal{I}_I \subseteq E^\mathcal{I}_I$ holds for each axiom $E_I \subseteq \mathcal{E}_2$ 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 $\subseteq_T$ over DL-Lite$_{core}$ concepts and roles $E_1, E_2$ to denote entailment w.r.t. a TBox $\mathcal{T}$, defined by $E_1 \subseteq_T E_2$ iff $E^\mathcal{I}_I \subseteq E^\mathcal{I}_I$ 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 $\mathcal{T}^\mathcal{K}_{can}$, unique up to isomorphism, s.t. there exists a homomorphism from $\mathcal{T}^\mathcal{K}_{can}$ 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 $B \sqsubseteq \exists x R$ can be expressed in the logic $\text{DL-Lite}^\text{H}$, but with a possibly exponential blowup of the TBox (assuming $n$ is encoded in binary). For instance, the axiom $A \sqsubseteq \exists x P$ can be expressed as $\{A \sqsubseteq \exists x \exists P, P \sqsubseteq \exists P, \exists P \sqsubseteq \exists P, \forall P \sqsubseteq \exists P \}$. CQs. A Conjointive Query (CQ) $q$ is an expression of the form $q(x) \leftarrow p_1(t_1), \ldots, p_n(t_n)$, where each $p_i \in N_R \cup N_A$, $x \subseteq N_V$, each $t_i \subseteq N_V \cup N_R$, and $p_1(t_1), \ldots, p_n(t_n)$ is syntactic sugar for the duplicate-free conjunction of atoms. A CQ is boolean if $x = \emptyset$.

Answers, Certain Answers. To define query answers under count semantics, we adopt the definitions by Cohen et al. [2007] and Kostylev and Reutter [2015]. A match for a query $q$ in an interpretation $I$ is a homomorphism from $\text{body}(q)$ to $I$. Then, an answer to $q$ over $I$ is a pair $(\omega, \pi)$ s.t. $\pi \geq 1$, and there are exactly $k$ matches $\pi_1, \ldots, \pi_k$ for $q$ in $I$ that verify $\omega = \pi_i \circ \text{ans}(q, I)$ for $i \in \{1, \ldots, k\}$. We use $\text{ans}(q, I)$ to denote the set of answers to $q$ over $I$. Similarly, if $Q$ is a set of queries, we use $\text{ans}(Q, I)$ to denote the set of all pairs $(\omega, \pi)$ s.t. $(\omega, k) \in \text{ans}(q, I)$ for some $k$ and $q \in Q$, and $\ell = \sum (k \mid (\omega, k) \in \text{ans}(q, I), q \in Q)$. Answering a query over an interpretation (i.e., a DB) is also known as query evaluation. Finally, a pair $(\omega, \pi)$ is a certain answer to a query $q$ over a KB $K$ iff $k \geq 1$ is the largest integer such that, for each model $I$ of $K$, $(\omega, k) \in \text{ans}(q, I)$ for some $k \geq 1$. We use $\text{certAns}(q, K)$ to denote the set of certain answers to $q$ over $K$.

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

| COUNT | Input: DL-Lite KB K, boolean CQ q, k ∈ N⁺ | Decide: k = certCard(q, 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 $G$ of a CQ $q$ is the undirected graph whose vertices are the variables appearing in $\text{body}(q)$, and that contains an edge between $x_1$ and $x_2$ if $P(x_1, x_2) \in \text{body}(q)$ for some role $P$. We call $q$ connected (denoted with $q \in \text{CQ}^\text{C}$) if $G$ is connected, linear ($q \in \text{CQ}^\text{L}$) if the degree of each vertex in $G$ is $\leq 2$, and acyclic ($q \in \text{CQ}^\text{A}$) if $G$ 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 ($q \in \text{CQ}^\text{R}$) if each connected component in $G$ contains at least one constant or one distinguished variable. Finally, a CQ $q$ is atomic ($q \in \text{AQ}$) if $|\text{body}(q)| = 1$.

Rewritability. Given a query language $Q$, a $Q$-rewriting of a CQ $q$ with respect to a KB $K = (\mathcal{T}, \mathcal{A})$ is a $Q$ query $q'$ whose answers over inter($\mathcal{A}$) alone coincides with the certain answers to $q$ over $K$. For instance, for OMQA with boolean or enumeration queries, $Q$ 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 $N_h$) 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$_{\text{pos}}$, with a reduction that uses a disconnected and cyclic query. coNP-membership is also shown for DLs up to DL-Lite$_{\text{core}}$.

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
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 Nikolau et al. [2019] illustrates this.

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

\[
\begin{array}{ccc}
A_1 & u & b \\
A_2 & p & p \\
\end{array}
\]

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 $I$ for $\mathcal{K}$. To satisfy $\mathcal{A}$, we set $A_1^I(a) = 1$ and $A_1^I(b) = 1$. To satisfy $A_1 \subseteq \exists P$, we introduce a single element $u$ (as above) and obtain $P^I(a, u) = 1$ and $P^I(b, u) = 1$. Therefore, $P^I(u, a) = 1$ and $P^I(u, b) = 1$, which, according to the semantics by Nikolau et al. [2019], imply that $(\exists P^I)I(u) = 2$. Therefore, to satisfy $\exists P^I \subseteq A_2$, it must be that $A_2^I(u) = 2$. In fact, the certain answer to $q$ over $I$ under bag-semantics is the empty tuple $\emptyset$ with multiplicity 2.

4 Tractability and Intractability

We investigate now conditions for intractability (in data complexity for DL-Lite$_{pos}$ and acyclic, linear, but disconnected CQs (CQL)).

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

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 $G = (V, E)$ be an undirected graph with vertices $V$, edges $E$, and without self-loops. The ABox is $\mathcal{A} = \{\text{Vertex}(v) \mid v \in V\} \cup \{\text{edge}(v_1, v_2) \mid (v_1, v_2) \in E\} \cup \{\text{Blue}(b), \text{Green}(g), \text{Red}(r), \text{hasColor}(a, b), \text{hasColor}(a, g), \text{hasColor}(a, r), \text{edge}(a, a)\}$ for some fresh constants $a, b, r,$ and $g$. The TBox is $T = \{\text{Vertex} \sqsubseteq \text{hasColor}, \exists\text{hasColor} \subseteq \text{Color}\}$. And the (acyclic, non-branching) query is $q() = \{\text{Color}(c), \text{edge}(v_1, v_2), \text{hasColor}(v_1, c_1), \text{hasColor}(v_2, c_2), \text{Blue}(c_1), \text{Blue}(c_2), \text{edge}(v_3, v_4), \text{hasColor}(v_3, c_3), \text{hasColor}(v_4, c_4), \text{Green}(c_3), \text{Green}(c_4), \text{edge}(v_5, v_6), \text{hasColor}(v_5, c_5), \text{hasColor}(v_6, c_6), \text{Red}(c_5), \text{Red}(c_6)\}$. Then it can be verified that $4 = \text{certCard}(q, \langle T, \mathcal{A} \rangle)$ iff $G$ 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$_{pos}$ and acyclic, connected, but branching CQs (CQ$^\text{AL}$).

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

Proposition 4. COUNT is in coNP in data complexity for DL-Lite$_{core}$ 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 disjunctions, and atomic queries.

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

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. $T_1$, $T_2$, and $T_0$ 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 $T$ is defined by $T = P \cup T_1 \cup T_2$, where $P = \{ P_T \subseteq P, P_F \subseteq P \}, T_1 = \{ F_1 \subseteq F, T_1 \subseteq T \}$, and $T_2 = \{ T \subseteq \exists P_T^F, F \subseteq (\exists P_T^F) \}$. Intuitively, the unary predicates $T_1$ and $T_2$ correspond to gates that evaluate to true and false respectively in the circuit, and binary predicates $P_T$ and $P_F$ specialize to positive and negative inputs. $T_2$ 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 $T_1$ enforces 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 $A$ is an extension of $C$, i.e., $C \subseteq \text{inter}(A)$. The domain of $A$ contains 3 additional individuals $t_1, t_2, f$, and it extends $P^C$ with $\cup_{i \in I} \{ P_i(t_1, t_2), P_i(t_1, f), P_i(f, t_1), P_i(f, f) \}$, and $\{ P(t_1, f), P(t_2, f), P(f, t_1), P(f, f), P(t_1, t_2), P(t_2, t_1) \}$.

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

Assuming $P \not\subseteq \text{LogSpace}$, 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$^H_{core}$ and rooted, connected, linear queries ($\text{CQ}^{GR}$).

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 $I$ of a KB $K$ is universal for a class of queries $Q$ iff $\text{ans}(q, I) = \text{certAns}(q, K)$ holds for every $q \in Q$. 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$^3$, for the DL-Lite$^a$ members up to DL-Lite$^b$. [Nikolaou et al., 2019] and DL-Lite$^b$ [Cima et al., 2019], with some syntactic restrictions. It was also shown that CQ$^3$ 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 CQ$^{GR}$ and DL-Lite$^N_{core}$, and queries evaluated under count semantics. Precisely, the canonical model $I^K_{can}$ obtained via the restricted chase from Calvanese et al. [2013] and Boteva et al. [2010] is a universal model. From now on, we denote by $\text{ch}_0(K)$ the set of atoms obtained after applying the i-th chase step over the KB $K$, and by $\text{ch}_0(K)$ (the possibly infinite) set of atoms obtained by an unbounded number of applications.

Proposition 7. DL-Lite$^N_{core}$ has a universal model w.r.t. COUNT over CQ$^{GR}$ queries.

Proof (Sketch). Consider a CQ$^{GR}$ query $q$ and a KB $K$, and let $\text{match}(q, I^K_{can})$ denote the set of all matches for body$(q)$ over the canonical model $I^K_{can} := \text{ch}_0(K)$. Let $I$ be a model of $K$. Then there must exist a homomorphism $\tau$ from $I^K_{can}$ to $I$. One immediately obtains that $\tau(\text{match}(q, I^K_{can})) \subseteq \text{match}(q, \tau(I^K_{can}))$, and therefore

$$\text{match}(q, I^K_{can}) \sqsubseteq \text{match}(q, \tau(I^K_{can}))$$

Then rely on the fact that $q$ is rooted, that the chase is restricted, and that DL-Lite$^N_{core}$ does not allow for role subsumption, it can proven that

$$\text{match}(q, I^K_{can}) \sqsubseteq \text{match}(q, \tau(I^K_{can}))$$

from (i) and (ii), $\text{match}(q, I^K_{can}) = \text{match}(q, \tau(I^K_{can}))$. Then, since $\tau$ is a homomorphism from $I^K_{can}$ to $I$, $\tau(I^K_{can}) \subseteq I$ must hold, and therefore $\text{match}(q, \tau(I^K_{can})) = \text{match}(q, I)$. We conclude that $\text{match}(q, I^K_{can}) = \text{match}(q, I)$. \hfill \Box

5.2 Rewriting for DL-Lite$^N_{core}$

We introduce PerfectRef$_{core}$, a rewriting algorithm for DL-Lite$^N_{core}$ inspired by 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$^N_{core}$ KB $K$, and a query $q'$ among those produced by PerfectRef or any other rewriting algorithm for CQs. Then, each match $\omega'$ for $q'$ in inter($A$) can be extended to the anonymous individuals so as to form a “complete” match $\omega$ for $q$ in inter($\text{ch}_0(K)$) 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 $K$.

Example 2. Consider the query $q(x) \leftarrow P(x, y)$, and the KB $K = \{ \{ A \subseteq \{ 1 \} \}, \{ A(a) \} \}$. Starting from $q$, 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 inter($A$), and that $\mu$ can be extended into exactly three matches for $q$ in inter($\text{ch}_0(K)$), 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 PerfectRef and significantly more complicated. It is also
We are now ready to introduce PerfectRef\textsubscript{out}. Consider a satisfiable knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, and a CQ\textsuperscript{CR} query $q(x) \leftarrow \psi(x, y)$. PerfectRef\textsubscript{out} takes as input $q$ and $\mathcal{T}$ and initializes the result set $Q$ as

$$\{(Q(x, \text{cnt}(y) \cdot 1), (q(x : y) \leftarrow \psi(x, y)))\}.$$  

Then the algorithm expands $Q$ by applying the rules AtomRewrite, Reduce, $GE_a$, and $GE_B$ 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, $P^r(w,z)$ stands for $P(z,w)$. Hence, also $R(w,z)$ when $R = P^r$ 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 $B \in N_C$, or $R(w,\_)$, if $B \supseteq \mathcal{A}$. Given a set $B$ of basic concepts, sub\textsubscript{B}($B$) is defined as the set of basic concepts $\{B' \mid B' \subseteq B, B \in B\}$. If $\phi, \psi$ are two conjunctions of atoms and $a$ is an atom in $\phi$, we use $\phi[a/\psi]$ (resp., $\phi[a/\perp]$) to denote the conjunction identical to $\phi$, but where $a$ is replaced with $\psi$ (resp., $a$ is deleted from $\phi$). By extension, if $r$ is a rule, $r[a/\psi]$ denotes the rule head($r$) $\leftarrow$ body($r$)[a/\psi]. If $B$ is a basic concept and $R$ a role, card\textsubscript{B}($B, R$) denotes the maximal $n \in \mathbb{N}$ such that $B \subseteq \mathcal{A}$, $R \subseteq \mathcal{T}$. 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 $\alpha$-blocked if it is bound, or if it occurs more than once in head($r$), or if it occurs in some $\exists$-atom in body($r$). Finally, $x$ is $\beta$-blocked if it is bound, or if it occurs more than once in head($r$), or if it occurs in some atom of the form $\exists \gamma R(w,z)$ with $\gamma > 0$.

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

**AtomRewrite ($\sim_{AR}$).** $\{q_1, \ldots, q_k\} \sim_{AR} \{q_1, \ldots, q_{k-1}, q_k'\}$ if

- $q_k = \langle Q(x, \text{cnt}(y) \cdot \alpha), \Pi \rangle$;
- for some $r \in \Pi$, for some $E(z)$ in body($r$), either:
  - $E(z)$ is of the form $A(z)$, and $B \subseteq A \subseteq \mathcal{T}$, or
  - $E(z)$ is of the form $R(w,z)$, $B \subseteq \mathcal{A}, R \subseteq \mathcal{T}, \mathcal{Z}$ is an unbound variable, and if head($r$) $= q(s : t)$, then $z \notin t$;
- $q_k' = \langle Q(x, \text{cnt}(y) \cdot \alpha), \Pi \cup \{r[E(z) / \xi(B,w)]\} \rangle$.

**Reduce ($\sim_{R}$).** $\{q_1, \ldots, q_k\} \sim_{R} \{q_1, \ldots, q_{k-1}, q_k'\}$ if

- $q_k = \langle Q(x, \text{cnt}(y) \cdot \alpha), \Pi \rangle$;
- $\{E_1(z_1), E_2(z_2)\} \subseteq \text{body}(r)$ for some $r \in \Pi$;
- $\sigma$ is a most general unifier for $E(z_1)$ and $E(z_2)$; with the following restrictions:
  - a variable in $x$ can map only to a variable in $x$;
  - a variable in $y$ can map only to a variable in $x \cup y$;
  - dom($\sigma$) $\subseteq z_1 \cup z_2$ and range($\sigma$) $\subseteq z_1 \cup z_2$;
- $q_k' = \langle Q(x, \text{cnt}(y) \cdot \alpha), \Pi \cup \{\sigma[r[E(z_2) / \xi(B,w)]]\} \rangle$.

**$GE_a$ ($\sim_{GE_a}$).** $\{q_1, \ldots, q_k\} \sim_{GE_a} \{q_1, \ldots, q_k \cup Q_k\}$ if

- $q_k = \langle Q(x, \text{cnt}(y) \cdot \alpha), \Pi \rangle$;
- $R(w,z)$, with $w, z \in x \cup y$, is an atom such that $\Psi = \{R(w,z) \in \text{body}(r) \mid r \in \Pi \text{ and } y \text{ is a non-}$-$\alpha$-blocked aggregation variable $\}$ $\neq \emptyset$.

---

\*Head and body of a rule are defined as for CQs.
Let \( \psi \) be the conjunction of all exists-atoms in any rule \( r \in R \) by construction, such conjunction is the same for all rules in \( \Pi \). Then the conjunction \( \psi \wedge \exists R(w,z) \) (seen as a set) must not appear in other rules from \( \{ q_1, \ldots, q_k \} \);  
• \( B_k \) is the maximal set of basic concepts \( B \) such that \( B \subseteq \exists R \subseteq T \). The \textit{answer} to such query is \( H_0 \). Indeed, there are three ways into the anonymous part of the canonical model. The inlimit the possible renamings of variables. Rewriting rules Reduce are analogous to their counterparts in the original \AtomRewrite set.  

\[ GE_\beta (\mapsto \mapsto_3). \]  
This rule is defined as \( \mapsto \mapsto_3 \), but with the difference that conditions \( \star \) and \( \circ \) are as follows:  

- \( \star \) \( R(w,z) \in \text{body}(r) \)  
- \( \circ \) As \( \circ \) for \( GE_a \), with the additional condition that all atoms in which variable \( \gamma \) occurs are removed.  

Note that, once all rules have been applied to saturation, the resulting set \( Q^* \) 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 \( Q(x, \text{cnt}(y) \cdot \alpha, \Pi) \) of \( Q^* \) into a query, we normalize it by renaming the variables in rules in \( \Pi \), based on their positions, according to \( x \) and \( y \), and by replacing constants and repeated variables in the head of a rule with suitable equalities in its body.  

The intuition behind \PerfectRef 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 \( GE_a \) and \( GE_\beta \) extend the way existential quantification is handled in \PerfectRef, and are the only ones eliminating aggregation variables from the \textit{rules} in \( \Pi \). 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 \( \alpha \) 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.
ways of extending the match \( \alpha \mapsto a \) into the anonymous part. Summing up the numbers, we get that our set of queries returns the answer \((x \mapsto a, 6)\), which indeed is the answer to our input query over the chase model from Figure 2.

The algorithm terminates because the application of \( \text{AtomRewrite} \) and \( \text{Reduce} \) is blocked upon reaching saturation, and each application of \( \text{GE}_a \) and \( \text{GE}_b \) reduces the number of variables in \( \psi \) by 1. The following lemmas show the correctness of \( \text{PerfectRefEnt} \).

**Lemma 8.** Consider a \( \text{DL-Lite} \) knowledge base \( \mathcal{K} = (T, \mathcal{A}) \) and a connected, rooted \( CQ \) \( q \). Consider a query \( Q' \) belonging to the output of \( \text{PerfectRefEnt} \) over \( q \) and \( \mathcal{K} \). Then, each match \( \omega' \) for \( Q' \) in \( \text{inter}(\mathcal{A}) \) can be extended into a match \( \omega \) for \( q \) in \( \text{inter}(\text{ch}_w(\mathcal{K})) \).

**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 \( \text{DL-Lite} \) knowledge base \( \mathcal{K} = (T, \mathcal{A}) \) and a connected, rooted \( CQ \) \( q \). Every match \( \omega \) for \( q \) in \( \text{inter}(\text{ch}_w(\mathcal{K})) \) is an extension of some match \( \omega' \) for \( Q' \) in \( \text{inter}(\mathcal{A}) \), where \( Q' \) belongs to the output of \( \text{PerfectRefEnt} \).

**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 \( \text{DL-Lite} \) knowledge base \( \mathcal{K} = (T, \mathcal{A}) \) and a connected, rooted \( CQ \) \( q \). Consider a query \( Q'(x, \text{cnt}(y) \cdot c, \Pi) \) belonging to the output of \( \text{PerfectRefEnt} \) over \( q \) and \( \mathcal{K} \). Then, each match \( \omega' \) for \( Q' \) in \( \text{inter}(\mathcal{A}) \) can be extended into exactly \( \alpha \) matches \( \omega \) for \( q \) in \( \text{inter}(\text{ch}_w(\mathcal{K})) \) with range \( (\omega \setminus \alpha) \subseteq N_b \).

**Proof (Sketch).** By induction on the number of applications of \( \text{GE}_a \) and \( \text{GE}_b \). It uses Lemma 8 and the fact that variables are never eliminated by \( \text{AtomRewrite} \) or \( \text{Reduce} \). The atomic decomposition in \( \text{GE}_a \) and \( \text{GE}_b \) guarantees that all combinations of number restrictions are considered.

**Proposition 11.** Consider a \( \text{DL-Lite} \) knowledge base \( \mathcal{K} = (T, \mathcal{A}) \) and a connected, rooted \( CQ \) \( q \). Let \( Q \) be the set of queries returned by a run of \( \text{PerfectRefEnt} \) over \( q \) and \( \mathcal{K} \). Then:

\[
\text{ans}(Q, \text{inter}(\mathcal{A})) = \text{certAns}(q, \mathcal{K})
\]

**Proof (Sketch).** The claim follows from Lemmas 9 and 10, and by observing that the query \( Q' \) in Lemma 10 is unique due to \( 3 \equiv \exists \) expressions, atomic decompositions, and the restrictions on \( \text{GE}_a \) and \( \text{GE}_b \). Therefore, matches are not counted twice.

The execution of \( \text{PerfectRefEnt} \) does not depend on the ABox. Considering that the evaluation of \( \text{FO} \text{(COUNT)} \) queries is in \( \text{LogSpace} \) in data complexity, this yields:

**Proposition 12.** \( \text{COUNT} \) is in \( \text{LogSpace} \) in data complexity for \( \text{DL-Lite} \) and rooted, connected \( CQs \).

<table>
<thead>
<tr>
<th>( AQ, \text{CO} )</th>
<th>( \text{CO}^{\text{CL}} )</th>
<th>( \text{CO}^{\text{AC}} )</th>
<th>( \text{CO}^{\text{GIR}, \text{CO}^{\text{CR}}} )</th>
<th>( \text{CO}^{\text{AL}} )</th>
<th>( \text{CO} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{DL-Lite}_\text{pass} )</td>
<td>( \text{P} )</td>
<td>( \text{coNP} )</td>
<td>( \text{L} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{pos} )</td>
<td>( \text{P} )</td>
<td>( \text{coNP} )</td>
<td>( \text{L} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{H} )</td>
<td>( \text{P} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{full} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{core} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
<td>( \text{coNP} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{pos} )</td>
<td>( \text{P} )</td>
<td>( \text{coNP} )</td>
<td>( \text{P-h} )</td>
<td>( \text{P-h} )</td>
<td>( \text{P-h} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{full} )</td>
<td>( \text{P} )</td>
<td>( \text{coNP} )</td>
<td>( \text{P-h} )</td>
<td>( \text{P-h} )</td>
<td>( \text{P-h} )</td>
</tr>
<tr>
<td>( \text{DL-Lite}_\text{core} )</td>
<td>( \text{P-h} )</td>
<td>( \text{coNP} )</td>
<td>( \text{P-h) }</td>
<td>( \text{P-h} )</td>
<td>( \text{P-h) }</td>
</tr>
</tbody>
</table>

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 Discussion**

Table 1 summarizes our results for data complexity of query answering under count semantics for variants of \( CQs \) and \( \text{DL-Lite} \). 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 \( \text{DL-Lite}\) and \( AQ/\text{CO}^{\text{CL}} \) is tight. Indeed, the P-hardness result provided for \( AQ \) holds for a more expressive DL (namely \( \text{DL-Lite} \)), 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 \( \text{DL-Lite} \) with neither disjointness nor role subsumption, rewritability in a variant of SQL with aggregates can be regained. An interesting open question is whether rewritability still holds for rooted queries and \( \text{DL-Lite}_\text{core} \), 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 \( \text{DL-Lite} \) 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


