Rewriting Count Queries over DL-Lite TBoxes with Number Restrictions*

Diego Calvanese[1,2], Julien Corman[1], Davide Lanti[1], and Simon Razniewski[3]

[1] Free University of Bozen-Bolzano, Italy, *lastname*@inf.unibz.it
[2] Umeå University, Sweden, diego.calvanese@umu.se
[3] Max-Planck-Institut für Informatik, Saarbrücken, Germany, srazniew@mpi-inf.mpg.de

Abstract. We propose a query rewriting algorithm for a restricted class of conjunctive queries evaluated under count semantics over a DL-Lite knowledge base. The target query language is an extension of relational algebra with aggregation and arithmetic functions, which can be translated into SQL. The algorithm supports number restrictions on the RHS of axioms in the input TBox, which can be used to encode statistics. The size of the output query remains linear in the binary encoding of these numbers, which improves upon previously proposed approaches.

1 Introduction

Counting answers to a query is an essential operation in data management, and is supported by virtually every relational database management system (RDBMS). 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. Our work falls within the scope of Ontology Mediated Query Answering (OMQA)[1] [7,3], where the background knowledge takes the form of a set of logical statements, called *TBox*, and the data is represented as a set of facts, called *ABox*. TBoxes are in general expressed in Description Logics (DLs), which are decidable fragments of first-order (FO) logic.

In this setting, counting answers to a query over a KB may require operations that go beyond counting ABox facts, as one also needs to take into account facts that can be inferred through the TBox. For instance, consider a scenario where a central repository keeps track of aggregated information about universities, such as their size, measured in number of students. And assume that these universities also keep detailed records about enrollment. These data sources may be integrated into a DL KB with a TBox of the form:

$$\begin{aligned} \mathit{Uni} &\sqsubseteq \geq_{1000} \mathit{enrolledIn}^- \\ \mathit{MedUni} &\sqsubseteq \geq_{5000} \mathit{enrolledIn}^- \\ \mathit{BigUni} &\sqsubseteq \geq_{20000} \mathit{enrolledIn}^- \end{aligned}$$

[1] Also referred to as OBDA (for Ontology Based Data Access), when emphasis is placed on mappings connecting external data sources to a TBox [16].

And an ABox of the form:

$$\begin{array}{l} \mathit{Uni}(\texttt{UniBZ}),\ \mathit{BigUni}(\texttt{UW}),\ \mathit{BigUni}(\texttt{TUW}),\ \ldots, \\ \mathit{region}(\texttt{UniBZ}, \texttt{SouthTyrol}),\ \mathit{region}(\texttt{UW}, \texttt{Vienna}),\ \mathit{region}(\texttt{TUW}, \texttt{Vienna}),\ \ldots, \\ \mathit{enrolledIn}(\texttt{Alice}, \texttt{UniBZ}),\ \mathit{enrolledIn}(\texttt{Bob}, \texttt{UW}),\ \mathit{enrolledIn}(\texttt{Carol}, \texttt{TUW}),\ \ldots \end{array}$$

In practice, the information about enrollment in such KB may be incomplete. First, aggregated data about a whole university may be available, but not the detailed records. The opposite may also hold: detailed records may be provided by a university, but the central repository may not have an aggregated value for it. In such scenarios, even if the data is incomplete, one may still want to compute a lower bound on the number of enrollments of students within each region. In more formal terms, one might be interested in counting the number of answers, as in the following query:

$$q(\mathsf{count}(x, y)) \leftarrow \mathit{enrolledIn}(x, y) \wedge \mathit{region}(y, \texttt{Vienna})$$

Answering such query may require arithmetic operations that combine numbers present in number restrictions in the TBox, and numbers obtained by counting ABox statements. For instance, in this example, assume that *TUW* and *UW* are the only two universities in the KB that are registered in the region of *Vienna*. Assume also that *UW* does not share enrollment records, whereas *TUW* shares them, however they may be incomplete (for instance, because some faculties within *TUW* have not yet communicated their information regarding enrollments). Then, the output value for $\mathsf{count}(x, y)$ should be the sum of: 20 000 on the one hand (for *UW*), and the largest value between 20 000 and n on the other hand (for *TUW*), where n is the number of enrollment records shared by *TUW*.

Query answering over DL KBs has been extensively studied for boolean and enumeration queries. With respect to the focus of this paper, a relevant result [6] is that for conjunctive queries (CQs) and unions of CQs (UCQs), answering a query over a KB expressed in certain lightweight DLs can be performed by *rewriting* it, according to the axioms in the TBox, into a relational algebra (RA) query over the ABox. This property, called *FO rewritability*, allows one to use an RDBMS for the evaluation of the rewritten query, thus leveraging already mature optimization techniques implemented in such systems. Among these DLs, the *DL-Lite* family [6,1] has been widely studied and adopted in OMQA/OBDA systems, resulting in the OWL 2 QL standard [13].

In comparison, the problem of *counting* answers to a CQ over a *DL-Lite* KB has seen little interest in the literature. A seminal result was provided in [11], showing that the problem is significantly harder already for relatively inexpressive DLs, with a shift from AC^0-membership to coNP-completeness in *data complexity*, i.e., assuming the sizes of the query and TBox to be fixed. Another key result was provided in [14], but for a slightly different semantics called *bag semantics*: for certain variants of *DL-Lite*, CQs that are *rooted* (i.e., with at least one constant or answer variable) can be rewritten into a variant of bag RA. However, this result is provided for DLs without number restrictions (like $\geq_{1000} \mathit{enrolledIn}^-$ in the example above). In addition, because bag semantics differs from the *count semantics* studied in [11], this rewritability result does not

directly apply to our setting. Finally, in [4,5], it was shown that under count semantics (and for some variants of *DL-Lite*), rooted[2] CQs can be rewritten into a variant of RA with aggregation and arithmetic functions. However, the provided rewriting algorithm may yield a query whose size is polynomial in the values of the numbers that occur in the TBox, i.e., exponential in the binary representation of such numbers, which is impractical.

In this paper, we improve upon this last result, showing how for the same DL, and under count semantics, rooted CQs can be rewritten into a different extension of RA, such that the number of algebraic operators in the output query does not depend on the numbers that occur in the TBox, but only on the number of axioms of the TBox. As a consequence, the size of the resulting query is polynomial in the numbers that appear in the number restrictions of the TBox, even when such numbers are represented in binary. This is an important step towards efficient query answering in such setting.

2 Preliminaries

We assume mutually disjoint countably infinite 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). An *atom* is an expression of the form $A(s)$ or $P(s,t)$, with $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$, and $s,t \in \mathsf{N_I} \cup \mathsf{N_E} \cup \mathsf{N_V}$. In the following, boldface letters like $\mathbf{t}$ denote tuples, and we sometimes treat tuples as sets. If $\mathbf{t} = (t_1, .., t_n)$ is a tuple and f a function defined for each t_i, we use $f(\mathbf{t})$ to denote the tuple $(f(t_1), .., f(t_n))$.

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* $\cdot^\mathcal{I}$ maps each constant $c \in \mathsf{N_I}$ to itself ($c^\mathcal{I} = c$, or 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 $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

If $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ is an interpretation, we may use $\mathcal{I}$ to denote the set of atoms $\{A(s) \mid A \in \mathsf{N_C}, s \in A^\mathcal{I}\} \cup \{P(s,t) \mid P \in \mathsf{N_R} \text{ and } (s,t) \in P^\mathcal{I}\}$. Conversely, if $\mathcal{S}$ is a set of unary and binary atoms with arguments in $\mathsf{N_I} \cup \mathsf{N_E}$, we may view it as the interpretation $\langle \Delta^\mathcal{S}, \cdot^\mathcal{S} \rangle$, where $\Delta^\mathcal{S}$ is the union of the arguments of all atoms in $\mathcal{S}$, and $\cdot^\mathcal{S}$ is defined by $A^\mathcal{S} = \{s \mid A(s) \in \mathcal{S}\}$, for $A \in \mathsf{N_C}$, and $P^\mathcal{S} = \{(s,t) \mid P(s,t) \in \mathcal{S}\}$, for $P \in \mathsf{N_R}$.

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*. In this

[2] Precisely, the result provided in [4,5] holds for rooted *and connected* CQs. This result immediately extends to rooted but disconnected CQs, by observing that the definition of rootedness requires each connected component to be rooted. Therefore, to answer such CQ, it suffices to compute the answers to each connected component separately, and then the cartesian product of these answers, multiplying the obtained cardinalities.

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

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

paper, we focus on (fragments of) *DL-Lite*$_{core}^{\mathcal{HN}^-}$ [4,5], a member of the the *DL-Lite* family [1] where each axiom is a *concept inclusion* of the form $B \sqsubseteq C$ or a *role inclusion* of the form $R_1 \sqsubseteq R_2$, following the grammar of Figure 1. From now on, we use the symbols B, C, P and R in accordance with this grammar, and we call these elements respectively *basic concepts*, *concepts*, *atomic roles* and *roles*. Concepts of the form $\geq_n R$ are called *number restrictions*. The fragment of *DL-Lite*$_{core}^{\mathcal{HN}^-}$ that disallows role inclusions and number restrictions where $n > 1$ is called *DL-Lite*$_{core}$ [1].

The evaluation $C^{\mathcal{I}}$ (resp. $R^{\mathcal{I}}$) of a concept C (resp. role R) over interpretation an $\mathcal{I}$ is defined inductively over the structure of C (resp. R) as usual (see [2] for a definition). An interpretation $\mathcal{I}$ is a *model* of $\langle \mathcal{T}, \mathcal{A} \rangle$ iff $\mathcal{A} \subseteq \mathcal{I}$ holds, and $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (resp. $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$) holds for each concept inclusion (resp. role inclusion) axiom $B \sqsubseteq C$ (resp. $R_1 \sqsubseteq R_2$) 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".

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 $\mathbf{x} \subseteq \mathsf{N_V}$, and each $p_i(\mathbf{t}_i)$ is an atom with arguments in $\mathsf{N_V} \cup \mathsf{N_I}$. In addition, we require *safeness*, i.e., $\mathbf{x} \subseteq \mathbf{t}_1 \cup \cdots \cup \mathbf{t}_n$. We use $\mathsf{dist}(q)$ to denote $\mathbf{x}$, called the *distinguished* variables of q, and we use $\mathsf{body}(q)$ to denote $\{p_1(\mathbf{t}_1), \ldots, p_n(\mathbf{t}_n)\}$. The *Gaifman graph* $\mathcal{G}_q$ of 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. Following [14], we call a CQ q *rooted* if each connected component in $\mathcal{G}_q$ contains at least one constant or distinguished variable.

We adopt the semantics proposed in [11] for counting answers to a CQ over a KB, which we call *count semantics*. If f is a function and D a subset of its domain, then we use $f|_D$ to denote f restricted to D. A *match* for a CQ q over an interpretation $\mathcal{I}$ is a homomorphism from $\mathsf{body}(q)$ to $\mathcal{I}$. Let M be the set of matches for q over $\mathcal{I}$, let ω be a mapping from $\mathsf{dist}(q)$ to $\mathsf{N_I}$, and let $\Gamma = \{\gamma \in M \mid \gamma|_{\mathsf{dist}(q)} = \omega\}$. Then the pair $\langle \omega, |\Gamma| \rangle$ is an *answer* to q over $\mathcal{I}$ iff $|\Gamma| \geq 1$. And we use $\mathsf{ans}(q, \mathcal{I})$ to denote the set of answers to q over $\mathcal{I}$. Finally, a pair $\langle \omega, k \rangle$ is a *certain answer* to q over a KB $\mathcal{K}$ (under *count semantics*) iff k is the *largest* element of $\mathbb{N} \cup \{+\infty\}$ 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 $\mathcal{K}$.

A property that has been extensively studied in the OBDA/OMQA literature is *FO rewritability* [6,1]. Query answering for a class $\mathcal{Q}$ of queries is FO rewritable for a DL $\mathcal{L}$ iff, for every $\mathcal{L}$ TBox $\mathcal{T}$ and $\mathcal{Q}$-query q, there is a FO query q' such that, for every ABox $\mathcal{A}$, the certain answers to q over $\langle \mathcal{T}, \mathcal{A} \rangle$ and the answers to q' over $\mathcal{A}$ (viewed as an interpretation) coincide. In particular, under

standard certain answer semantics for DLs (which differs from the definition of $\mathsf{certAns}(q, \mathcal{K})$ under count semantics given above), query answering for UCQs is FO rewritable for several members of the *DL-Lite* family. Since RA captures (domain-independent) FO logic, the evaluation of the query obtained via rewriting can be delegated to a RDBMS. Then in [14], this notion of rewritability has been adapted to a different target query language, called BCALC, which captures relational bag algebra [10], and can be translated into SQL queries with aggregation.

3 Related Work

Query answering under count semantics over a relational DB can be viewed as a specific case of query answering under *bag semantics*, investigated notably in [10,12]. 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 relational DB setting [15,9], but from the perspective of combined complexity, where the query is considered part of the input (i.e., its size is not assumed to be fixed).

As for (*DL-Lite*) KBs, in [8] an alternative (*epistemic*) count semantics is defined, 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.

Closer to our concern is the work presented in [11], which introduces the count semantics for CQs over a DL KB adopted in this paper. In particular the COUNT *problem*, which is the decision variant of query answering under count semantics, was shown in [11] to be coNP-hard in data complexity for the relatively inexpressive DL *DL-Lite*$_{core}$, even when negated concepts are forbidden. This implies that for this DL and arbitrary CQs, query answering under count semantics is not rewritable into a target query language for which query answering is easier than coNP in data complexity.

Then, rewritability of CQs over a *DL-Lite* KB was investigated in [14] for the related problem of query answering under bag semantics. In particular, a rewriting algorithm was provided for rooted CQs into the language BCALC (which can be evaluated in TC^0), and for a DL that extends *DL-Lite*$_{core}$ with a restricted form of role subsumption. This result does not immediately transfer to our setting though, for two reasons. First, this DL cannot express number restrictions on the right-hand side (RHS) of TBox axioms, and therefore cannot encode statistics like the ones from our motivating example. Second, as shown in [14], for *DL-Lite*$_{core}$ already, bag and count semantics differ in the presence of existential quantification on the left-hand side (LHS) of TBox axioms, which are typically used to express the *domain* and *range* of an atomic role.

To understand this difference, we recall the basics of query answering under bag semantics (and refer to [14] for a complete definition.) A *bag interpretation* $\mathcal{I}$ maps each atomic concept A (resp. atomic role P) to a bag $A^{\mathcal{I}}$ (resp. $P^{\mathcal{I}}$). Such bag can be seen as a function that maps each element of $\Delta^{\mathcal{I}}$ (resp. $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$)

to the number of times it occurs in the bag. This function $\cdot^{\mathcal{I}}$ is then extended to complex concepts and roles inductively (see [14]). And $\mathcal{I}$ satisfies an axiom $C_1 \sqsubseteq C_2$ (resp. $R_1 \sqsubseteq R_2$) iff $(C_1)^{\mathcal{I}}(d) \leq (C_2)^{\mathcal{I}}(d)$ for every $d \in \Delta^{\mathcal{I}}$ (resp. $(R_1)^{\mathcal{I}}(d_1, d_2) \leq (R_2)^{\mathcal{I}}(d_1, d_2)$ for every $(d_1, d_2) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$).

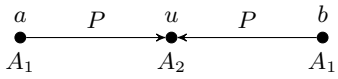
Now let $q(\mathbf{x}) \leftarrow p_1(\mathbf{t}_1), \ldots, p_n(\mathbf{t}_n)$ be a CQ, let $\mathcal{I}$ be a bag interpretation, let V be the set of variables that appear in q, let ω be a mapping from $\mathbf{x}$ to $\mathsf{N_I}$, let $\Gamma = \{\gamma : V \to \Delta^{\mathcal{I}} \mid \gamma|_{\mathsf{dist}(q)} = \omega\}$, and let $k = \sum_{\gamma \in \Gamma} \prod_{1 \leq i \leq n} (p_i)^{\mathcal{I}}(\gamma(\mathbf{t}_i))$. Then $\langle \omega, k \rangle$ is a *bag answer* to q over $\mathcal{I}$ iff $k \geq 1$. We use $\mathsf{bagAns}(q, \mathcal{I})$ to denote the set of bag answers to q over $\mathcal{I}$. And if $\mathcal{K}$ is a KB, we use $\mathsf{bagCertAns}(q, \mathcal{K})$ to denote the certain answers to q over $\mathcal{K}$ under bag semantics, defined analogously to certain answers under count semantics.[3]

The difference between bag and count semantics is illustrated in [14] with the following example:

Example 1. Consider the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where

$$\mathcal{T} = \{A_1 \sqsubseteq \exists P,\ \exists P^- \sqsubseteq A_2\}, \qquad \mathcal{A} = \{A_1(a),\ A_1(b)\},$$

and the query $q() \leftarrow A_2(y)$. If we evaluate this query under count semantics, then $\mathsf{certAns}(q, \mathcal{K}) = \{\langle \{\}, 1 \rangle\}$ (i.e., the only certain answer to q is the mapping with empty domain $\{\}$, with multiplicity 1), because the following structure is a model of $\mathcal{K}$:



However, such structure does not accurately represent a bag interpretation. Let us now build a (minimal) bag interpretation $\mathcal{I}$ for $\mathcal{K}$. To satisfy $\mathcal{A}$, we set $A_1^{\mathcal{I}}(a) = 1$ and $A_1^{\mathcal{I}}(b) = 1$. Then 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 proposed in [14], imply that $(\exists P^-)^{\mathcal{I}}(u) = 2$. Then in order for this bag interpretation to satisfy $\exists P^- \sqsubseteq A_2$, it must be the case that $(A_2)^{\mathcal{I}}(u) = 2$. So the only certain answer to q over $\mathcal{K}$ under bag semantics is the empty mapping with multiplicity 2, i.e. $\mathsf{bagCertAns}(q, \mathcal{K}) = \{\langle \{\}, 2 \rangle\}$. ■

It was also shown in [14] that query answering under bag and count semantics coincide for the DL that extends *DL-Lite*$_{core}$ with role inclusion, but disallows concepts of the form $\geq_1 R$ on the LHS of axioms. which we denote here with

[3] The notation used in [14] for certain answers under both bag and count semantics is slightly different from ours. Let $\mathsf{dist}(q) = \{x_1, .., x_n\}$. Then the certain answers to q under bag semantics are represented in [14] as a partial function $\mu : (\mathsf{N_C})^n \to \mathbb{N}^+$. Instead, we use $\mathsf{bagCertAns}(q, \mathcal{K}) = \{\langle \{x_1 \mapsto c_1, .., x_n \mapsto c_n\}, k \rangle \mid \mu(c_1, .., c_n) = k\}$. As for count semantics, let $\langle \omega, k \rangle \in \mathsf{certAns}(q, \mathcal{K})$. Then the definition provided in Section 2 implies that there is no $k' \neq k$ such that $\langle \omega, k' \rangle \in \mathsf{certAns}(q, \mathcal{K})$. Instead, in [11,14], $\langle \omega, k' \rangle$ is considered a certain answer under count semantics for each $1 \leq k' \leq k$.

DL-Lite$^{\mathcal{HF}}_{core}$. However, our focus is once again on logics that allow for number restrictions on the RHS of axioms (and thus can encode statistics). So a natural question is whether this result still holds for *DL-Lite*$^{\mathcal{HF}}_{core}$ extended with such axioms. Let *DL-Lite*$^{\mathcal{HN}^-\mathcal{F}}_{core}$ denote this DL (equivalently, *DL-Lite*$^{\mathcal{HN}^-\mathcal{F}}_{core}$ is the fragment of *DL-Lite*$^{\mathcal{HN}^-}_{core}$ that disallows concepts of the form $\geq_1 R$ on the LHS of axioms). To answer this question, the bag semantics proposed in [14] needs to be extended to concepts of the form $\geq_n R$. One way to do this is to define $(\geq_n R)^{\mathcal{I}}(d_1) = \left(\sum_{d_2 \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(d_1, d_2)\right) / n$, for any bag interpretation $\mathcal{I}$ and $d_1 \in \Delta^{\mathcal{I}}$, where "/" denotes integer division. With this definition, the proof provided in [14, Proposition 85] can be immediately extended to *DL-Lite*$^{\mathcal{HN}^-\mathcal{F}}_{core}$:

Proposition 2. *For any DL-Lite*$^{\mathcal{HN}^-\mathcal{F}}_{core}$ *KB* $\mathcal{K}$ *and CQ* q

$$\mathsf{certAns}(q, \mathcal{K}) = \mathsf{bagCertAns}(q, \mathcal{K})$$

Proof (sketch). The proof of Proposition 85 in [14] uses the fact that for any *DL-Lite*$^{\mathcal{HF}}_{core}$ KB $\mathcal{K}$ and model $\mathcal{I}$ of $\mathcal{K}$ under count semantics, there is a bag interpretation $\mathcal{I}_b$ that is a model of $\mathcal{K}$ under bag semantics, and verifies $\mathsf{ans}(q, \mathcal{I}) = \mathsf{bagAns}(q, \mathcal{I}_b)$ for any CQ q. This bag interpretation is defined for $M \in \mathsf{N_C} \cup \mathsf{N_R}$ by $M^{\mathcal{I}_b}(\mathbf{t}) = 1$ if $\mathbf{t} \in M^{\mathcal{I}}$, and $M^{\mathcal{I}_b}(\mathbf{t}) = 0$ otherwise. Now for any axiom of the form $A \sqsubseteq \geq_n R$, it can be verified that if $A^{\mathcal{I}} \subseteq (\geq_n R)^{\mathcal{I}}$, then $A^{\mathcal{I}_b}(d) \leq (\geq_n R)^{\mathcal{I}_b}(d)$ holds, for any individual d. So if $\mathcal{K}$ is now a *DL-Lite*$^{\mathcal{HN}^-\mathcal{F}}_{core}$ KB and $\mathcal{I}$ a model of $\mathcal{K}$, then $\mathcal{I}_b$ as defined above is still a model of $\mathcal{K}$ under bag semantics.

The proof of Proposition 85 in [14] also uses the fact that for any *DL-Lite*$^{\mathcal{HF}}_{core}$ bag KB $\mathcal{K}$ and model $\mathcal{I}$ of $\mathcal{K}$ under bag semantics, there is an interpretation $\mathcal{I}_s$ that is a model of $\mathcal{K}$ under count semantics, and such that for any CQ q, mapping ω over $\mathsf{dist}(q)$ and $k \in \mathbb{N}^+ \cup \{+\infty\}$, if $\langle \omega, k \rangle \in \mathsf{bagAns}(q, \mathcal{I})$, then $\langle \omega, k' \rangle \in \mathsf{ans}(q, \mathcal{I}_s)$ for some $k' \leq k$. This interpretation is defined for $M \in \mathsf{N_C} \cup \mathsf{N_R}$ by $\mathbf{t} \in M^{\mathcal{I}_s}$ iff $M^{\mathcal{I}}(\mathbf{t}) \geq 1$. Again, it can be verified that $\mathcal{I}_s$ is still a model of $\mathcal{K}$ if $\mathcal{K}$ is a *DL-Lite*$^{\mathcal{HN}^-\mathcal{F}}_{core}$ KB.

The rest follows from the original proof. □

Finally, in [4,5], for *DL-Lite*$^{\mathcal{N}^-}_{core}$ and under count semantics, a query rewriting algorithm was provided for rooted CQs into a variant of RA extended with aggregation and arithmetic functions. As for the rewriting of [14], the output query does not depend on the ABox. However, such algorithm is mostly of theoretical interest, and not well suited for implementation (see Section 4). Two negative results were also provided in [4,5], which further circumscribe the scope of rewritability. Specifically, for *DL-Lite*$^{\mathcal{H}}_{core}$, COUNT was shown to be P-hard both for rooted CQs and for CQs whose body contains a single atom. This implies that for this DL and these classes of queries, query answering under count semantics is not rewritable into a target query language for which query answering is easier than P.

4 Rewriting

Query answering under count semantics was shown in [4,5] to be rewritable for rooted CQs and $\mathit{DL\text{-}Lite}_{core}^{\mathcal{N}^-}$, into a target query language that extends RA with aggregation and some algebraic functions, thanks to a query rewriting algorithm inspired by PerfectRef [6]. However, the size of the rewritten query may be exponential in the size of the input TBox. More specifically, it may be exponential both in the number of axioms (precisely, in the depth of the deepest concept hierarchy that can be inferred from the TBox), and in the encoding of numbers that appear in number restrictions (if encoded in binary, thus polynomial in the value of such numbers). In many applications, it is reasonable to assume that the number of axioms in the TBox remains limited, so the first source of exponential blow-up may not be a major practical limitation. On the other hand, as illustrated in Section 1, values that appear in number restrictions (such as the one in $\geq_{20000} \mathit{enrolledIn}^-$) may depend on the size of the domain under consideration, and thus, in some applications, they are likely to be of the same order of magnitude as the size of the ABox itself. This might be unmanageable in practice, in scenarios where we have to deal with large amounts of data.

In the following, we describe how the rewriting algorithm of [4,5] can be adapted (for the same DL, source, and target query languages), so that the size of the output query remains linear in the size of the (binary) encoding of numbers in number restrictions. Moreover, this new rewriting guarantees that the number of algebraic operators in the output query only depends on the number of axioms of the TBox. The algorithm of [4,5] exploits the so-called *chase model* $\mathcal{I}_{can}^{\mathcal{K}}$ of $\mathcal{K}$. This model is such that, for $\mathit{DL\text{-}Lite}_{core}^{\mathcal{N}^-}$ and rooted CQs, $\mathsf{certAns}(q, \mathcal{K})$ and $\mathsf{ans}(q, \mathcal{I}_{can}^{\mathcal{K}})$ coincide.

Specifically, we illustrate how the rewriting algorithm from [4,5] can be modified by running it on the same example as the one used in [4,5]. Whenever relevant, we will explicitly point out the differences between the two algorithms. The rewriting from [4,5] generates a *set of queries* that can be directly translated into a SQL query. The target language for this rewriting makes use of nested aggregation in the form of special "atoms" of the form $\exists^{=k}.P(x, y)$, with $k \in \mathbb{N}$, which intuitively correspond to a SQL `COUNT DISTINCT` operation, together with a boolean condition stating that the result of the aggregation operation over y must be equal to k. If the TBox contains an axiom the form $C \sqsubseteq \geq_n R$, then for each $k \in [1..n]$, the rewriting of [4,5] generates one sub-query. Hence, the number of generated sub-queries is linear in n, and exponential in the binary encoding of n, which makes it unpractical.

In our variant, we drop the boolean condition and use instead the notation $z = \mathsf{count}(y).P(x, y)$, where z is now a variable storing the result of the same aggregation operation. Like in [4,5], we use negation, multiplication, and subtraction. In addition, our target language also requires the SQL function `greatest(x,y)` (that returns the largest value between `x` and `y`), and the aggregation operator `SUM`. We show through our running example that, despite our modifications, the target language for the rewriting can still be translated into SQL.

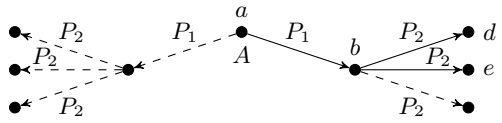


Fig. 2. Chase model of our running example. Solid arrows represent the information in the ABox, whereas dashed lines represent information implied by the KB.

Consider the following KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and CQ q:

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

$$q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$$

The chase model $\mathcal{I}^{\mathcal{K}}_{can}$ of $\mathcal{K}$ is represented in Figure 2. The rewriting algorithm operates on a set $\mathcal{Q}$ of queries. This set is initialized with a syntactic variant of the input query:

$$\langle Q(x, cnt = \mathsf{count}(y_1, y_2)),\ \{q(x, y_1, y_2) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)\} \rangle$$

Intuitively, such query corresponds to the following SQL query[4]

```
SELECT x, COUNT(DISTINCT y1, y2) as cnt
FROM A, P1, P2
WHERE A.x = P1.x A AND P1.y1 = P2.y1
GROUP BY x
```

Then each rewriting step selects a query Q in $\mathcal{Q}$, and extends $\mathcal{Q}$ with a set of new queries, obtained by applying some *rewriting rule* to Q, until saturation is reached. In the previous query, since variable y_2 is unbound, we can apply a rewriting step over atom $P_2(y_1, y_2)$ with respect to axiom $\exists P_1^- \sqsubseteq \geq_3 P_2$, producing the query:

$$\begin{array}{l} \langle Q(x, cnt = \mathsf{sum}(num)), \\ \quad \langle Q(x, y_1, num = \mathsf{greatest}(0, 3 - z)), \\ \quad\quad \{q(x, y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1)\} \wedge z = \mathsf{cnt}(y_2).\ P_2(y_1, y_2) \rangle \rangle \end{array}$$

This query corresponds to the following SQL query:

```
SELECT x, SUM(num) as cnt
FROM (SELECT x, y1, greatest(0, 3-z) as num
      FROM (SELECT x, y1
            FROM  A, P1, (SELECT x as _, y1 FROM P1) as P1a)
            WHERE A.x = P1.x AND P1.y1 = P1a.y1
           ) AS J1,
           (SELECT y1, COUNT(DISTINCT y2) as z
            FROM P2 GROUP BY y1
           ) AS J2
      WHERE J1.y1 = J2.y1
     )
GROUP BY x
```

[4] Note that the `COUNT DISTINCT` operator of SQL does not allow for multiple arguments. However, the desired result can be achieved through an injective concatenation operator, for instance making use of an additional fresh symbol.

The interested reader might observe that such query fully captures the three queries (one for each non-zero value of num) from [4,5]. On such a query, one can apply a variant of the "reduce" rule of PerfectRef [6], leading to the query:

$$\begin{array}{l}\langle Q(x, cnt = \mathsf{sum}(num)),\\ \quad \langle Q(x, y_1, num = \mathsf{greatest}(0, 3 - z)),\\ \quad\quad \left\{\begin{array}{l} q(x, y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1)\\ q(x, y_1) \leftarrow A(x), P_1(x, y_1)\end{array}\right\} \wedge z = \mathsf{cnt}(y_2).\ P_2(y_1, y_2)\rangle\rangle\end{array}$$

This query corresponds to the following SQL query:

```
SELECT x, SUM(num) as cnt
FROM (SELECT x, y1, greatest(0,3-z) as num
      FROM ((SELECT x, y1
             FROM A, P1, (SELECT x as _, y1 FROM P1) AS P1a
             WHERE A.x = P1.x AND P1.y1 = P1a.y1
            )
            UNION
            (SELECT x, y1 FROM A, P1 WHERE A.x = P1.x)
           ) AS J1,
           (SELECT y1, COUNT(DISTINCT y2) as z
            FROM P2
            GROUP BY y1
           ) AS J2
      WHERE J1.y1 = J2.y1
     )
GROUP BY x
```

Again, in the rewriting from [4,5], this very step produces three different queries: one for each non-zero value of num.

By applying another rewriting step over atom $P_1(x, y_1)$ and with respect to axiom $A \sqsubseteq \geq_2 P_1$, we obtain the following query:

$$\begin{array}{l}\langle Q(x, cnt = \mathsf{sum}(num)),\\ \quad \langle Q(x, num = (\mathsf{greatest}(0, (2 - z) \cdot 3),\\ \quad\quad \{q(x) \leftarrow A(x)\} \wedge z = \mathsf{cnt}(y_1).\ P_1(x, y_1)\rangle\rangle\end{array}$$

This query corresponds to the following SQL query:

```
SELECT x, SUM(num) as cnt
FROM (SELECT x, greatest(0,2-z) * 3 as num
      FROM (SELECT x FROM A) AS J1,
           (SELECT x, COUNT(DISTINCT y1) AS z
            FROM P1
            GROUP BY x
           ) AS J2
      WHERE J1.x = J2.x
     )
GROUP BY x
```

Let us now analyze the queries produced by the rewriting. 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, this query returns the answer $\{x \mapsto a, cnt \mapsto 2\}$. The query produced after the first rewriting step checks for all sub-paths (x, y_1) of (x, y_1, y_2) such that x is an instance 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 our query, which returns as answer $\{x \mapsto a, cnt \mapsto 1\}$: indeed, there is a single way of extending this path into the anonymous part. The last query has to be interpreted in a similar way. The query retrieves the individual a, since this node has only one P_1-successor in $\mathcal{A}$ but should have at least two P_1-successors according to $\mathcal{T}$. The answer to such query is $\{x \mapsto a, cnt \mapsto 3\}$. Indeed, there are three ways of extending the match $\{x \mapsto a\}$ into the anonymous part. Now, a last aggregation (with `SUM`) over all queries in $\mathcal{Q}$ yields the desired answer $\{x \mapsto a, cnt \mapsto 6\}$, which indeed corresponds to the answer $\langle\{x \mapsto a\}, 6\rangle$ to our input query over $\mathcal{I}^{\mathcal{K}}_{can}$.

5 Conclusions

In this paper, we have improved the query rewriting technique proposed in [4,5] for rooted CQs under count semantics over a *DL-Lite*$^{\mathcal{N}^-}_{core}$ KB, into a target query language that extends RA with aggregation and arithmetic functions. We have illustrated how the exponential blow-up of the output query in the numbers that occur in the TBox, characteristic of the rewriting of [4,5], can be avoided by extending the target rewriting language, specifically with the aggregate operator sum and with the binary function greatest. We also show that this enriched language can still be translated into SQL. Considering that numeric values that appear in the TBox may in practice be of the same order of magnitude as the size of the ABox, this new rewriting algorithm is a significant step towards a practical implementation of query answering under this semantics.

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, by the project IDEE (FESR1133) funded through the European Regional Development Fund (ERDF) Investment for Growth and Jobs Programme 2014-2020, by the project "South Tyrol Longitudinal study on Longevity and Ageing" (STyLoLa), funded through the 2017 Interdisciplinary Call issued by the Research Committee of the Free University of Bozen-Bolzano, and by the project "Ontology-based Geodata Integration, Visualization and Analysis" (OntoGeo), funded through the 2018 CRC Call issued by the Research Committee of the Free University of Bozen-Bolzano.

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The *DL-Lite* family and relations. *J. of Artificial Intelligence Research*, 36:1–69, 2009.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
3. M. Bienvenu and M. 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.
4. D. Calvanese, J. Corman, D. Lanti, and S. Razniewski. Counting query answers over a DL-Lite knowledge base. In *Proc. of the 29th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. IJCAI Org., 2020.
5. D. Calvanese, J. Corman, D. Lanti, and S. 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`.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
7. D. Calvanese, G. De Giacomo, and M. Lenzerini. Conjunctive query containment and answering under description logics constraints. *ACM Trans. on Comp. Logic*, 9(3):22.1–22.31, 2008.
8. D. Calvanese, E. Kharlamov, W. Nutt, and C. 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.
9. H. Chen and S. 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.
10. S. Grumbach and T. Milo. Towards tractable algebras for bags. *J. of Computer and System Sciences*, 52(3):570–588, 1996.
11. E. V. Kostylev and J. L. Reutter. Complexity of answering counting aggregate queries over DL-Lite. *J. of Web Semantics*, 33:94–111, 2015.
12. L. Libkin and L. Wong. Query languages for bags and aggregate functions. *J. of Computer and System Sciences*, 55(2):241–272, 1997.
13. B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language profiles (2nd ed.). W3C Rec., W3C, 2012. `http://www.w3.org/TR/owl2-profiles/`.
14. C. Nikolaou, E. V. Kostylev, G. Konstantinidis, M. Kaminski, B. Cuenca Grau, and I. Horrocks. Foundations of ontology-based data access under bag semantics. *Artificial Intelligence*, 274:91–132, 2019.
15. R. Pichler and S. Skritek. Tractable counting of the answers to conjunctive queries. *J. of Computer and System Sciences*, 79(6):984–1001, 2013.
16. G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. 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.