The Bag Semantics of Ontology-Based Data Access

Charalampos Nikolaou, Egor V. Kostylev, George Konstantinidis, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks
Department of Computer Science, University of Oxford, UK

Abstract

Ontology-based data access (OBDA) is a popular approach for integrating and querying multiple data sources by means of a shared ontology. The ontology is linked to each source by global-as-view (GAV) mappings [Lenzerini, 2002], which assign views over the data to ontology predicates. Motivated by the need for OBDA systems supporting bag semantics to aggregate queries, we propose a bag semantics for OBDA, where duplicate tuples in the views defined by the mappings are retained, as is the case in standard databases. We show that bag semantics makes conjunctive query answering in OBDA coNP-hard in data complexity. To regain tractability, we consider a rather general class of queries and show its rewritability to a generalisation of the relational calculus to bags.

1 Introduction

Ontology-based data access (OBDA) is an increasingly popular approach to enable uniform access to multiple data sources with diverging schemas [Poggi et al., 2008]. In OBDA, an ontology provides a unifying conceptual model for the data sources together with domain knowledge. The ontology is linked to each source by global-as-view (GAV) mappings [Lenzerini, 2002], which assign views over the data to ontology predicates. Users access the data by means of queries formulated using the vocabulary of the ontology; query answering amounts to computing the certain answers to the query over the union of ontology and the materialisation of the views defined by the mappings. The formalism of choice for representing ontologies in OBDA is the description logic DL-LiteR [Calvanese et al., 2007], which underpins OWL 2 QL [Motik et al., 2012]. DL-LiteR was designed to ensure that queries against the ontology are first-order rewritable; that is, they can be reformulated as a set of relational queries over the sources [Calvanese et al., 2007].

Example 1. A company stores data about departments and their employees in several databases. The sales department uses the schema $\text{SalEmployee}(\text{id}, \text{name}, \text{salary}, \text{loc}, \text{mngr})$, where attributes $\text{id}$, $\text{name}$, $\text{salary}$, $\text{loc}$, and $\text{mngr}$ stand for employee ID within the department, their name, salary, location, and name of their manager. In turn, the IT department stores data using the schema $\text{ITEmployee}(\text{id}, \text{surname}, \text{salary}, \text{city})$, where managers are not specified. To integrate employee data, the company relies on an ontology with TBox $\mathcal{T}_o$, which defines unary predicates such as $\text{SalEmp}$, $\text{ITEmp}$, and $\text{Mngr}$, and binary predicates such as $\text{hasMngr}$ relating employees to their managers. The following mappings determine the extension of the predicates based on the data, where each $\text{att}_i$ represents the attributes occurring only in the source:

- $\text{SalEmployee}(\text{name}, \text{att}_1) \rightarrow \text{SalEmp}(\text{name})$,
- $\text{SalEmployee}(\text{name}, \text{mngr}, \text{att}_2) \rightarrow \text{hasMngr}(\text{name}, \text{mngr})$,
- $\text{SalEmployee}(\text{mngr}, \text{att}_3) \rightarrow \text{Mngr}(\text{mngr})$,
- $\text{ITEmployee}(\text{surname}, \text{att}_4) \rightarrow \text{ITEmp}(\text{surname})$.

TBox $\mathcal{T}_o$ specifies the meaning of its vocabulary using inclusions (i) $\text{SalEmp} \sqsubseteq \text{Emp}$ and $\text{ITEmp} \sqsubseteq \text{Emp}$, which say that both sales and IT employees are company employees; (ii) $\exists \text{hasMngr} \sqsubseteq \text{Mngr}$, specifying the range of the $\text{hasMngr}$ relation, and (iii) $\text{Emp} \sqsubseteq \exists \text{hasMngr}$, requiring that employees have a (maybe unspecified) manager. Such inclusions influence query answering: when asking for the names of all company employees, the system will retrieve all relevant sales and IT employees; this is achieved via query rewriting, where the query is reformulated as the union of queries over the sales and IT databases.

OBDA has received a great deal of attention in recent years. Researchers have studied the limits of first-order rewritability in ontology languages [Calvanese et al., 2007; Artale et al., 2009], established bounds on the size of rewritings [Gottlob et al., 2014; Kikot et al., 2014], developed optimisation techniques [Kontchakov et al., 2014], and implemented systems well-suited for real-world applications [Calvanese et al., 2017; Calvanese et al., 2011].

An important observation about the conventional semantics of OBDA is that it is set-based: the materialisation of the views defined by the mappings is formalised as a virtual ABox consisting of a set of facts over the ontology predicates. This treatment is, however, in contrast with the semantics of database views, which is based on bags (multisets) and where duplicate tuples are retained by default. The distinction between set and bag semantics in databases is very significant in practice; in particular, it influences the evaluation of aggre-
gate queries, which combine various aggregation functions such as Min, Max, Sum, Count or Avg with the grouping functionality provided in SQL by the GroupBy construct.

**Example 2.** Consider the query asking for the number of employees named Lee. Assume there are two different employees named Lee, which are represented as different tuples in the sales database (e.g., tuples with the same employee name, but different ID). Under the conventional semantics of OBDA, the virtual ABox would contain a single fact SalEmp(Lee); hence, the query would wrongly return one, even under the semantics for counting aggregate queries in [Calvanese et al., 2008; Kostylev and Reutter, 2015]. The correct count can be obtained by considering the extension of SalEmp as a bag with multiple occurrences of Lee.

The goal of this paper is to propose and study a bag semantics for OBDA which is compatible with the semantics of standard databases and can provide a suitable foundation for the future study of aggregate queries. We focus on conjunctive query (CQ) answering over DL-LiteR ontologies under bag semantics, and our main contributions are as follows.

1. We propose the ontology language DL-LiteRbag and its restriction DL-LiteRbagcore, where ABoxes consist of a bag of facts, thus providing a faithful representation of the views defined by OBDA mappings. We define the semantics of query answering in this setting and show that it is compatible with the conventional set-based semantics.

2. We show that, in contrast to the set case, ontologies may not have a universal model (i.e., a single model over which all CQs can be correctly evaluated), and bag query answering becomes coNP-hard in data complexity even if we restrict ourselves to DL-LiteRcore ontologies.

3. To regain tractability, we study the class of rooted CQs [Bienvenu et al., 2012], where each connected component of the query graph is required to contain an individual or an answer variable. This is a very general class, which arguably captures most practical OBDA queries. We show that rooted CQs over DL-LiteRbagcore ontologies not only admit a universal model and enjoy favourable computational properties, but also allow for rewritings that can be directly evaluated over the bag ABox of the ontology. For the proofs of all results we refer to [Nikolau et al., 2017].

2 Preliminaries

**Syntax of Ontologies** We fix a vocabulary consisting of countably infinite and pairwise disjoint sets of individuals I (i.e., constants), variables X, atomic concepts C (unary predicates) and atomic roles R (binary predicates). A role is an atomic role P ∈ R or its inverse P−1. A concept is an atomic concept in C or an expression ∃R, where R is a role. An inclusion is an expression of the form S1 ⊑ S2 with S1 and S2 either both concepts or both roles. A disjointness axiom is an expression of the form Disj(S1, S2) with S1 and S2 either both concepts or both roles. A concept assertion is of the form A(a) with a ∈ I and A ∈ C. A role assertion is of the form P(a, b) with a, b ∈ I and P ∈ R. A DL-LiteR TBox is a finite set of inclusions and disjointness axioms. An ABox is a finite set of concept and role assertions. A DL-LiteR ontology is a pair ⟨T, A⟩ with T a DL-LiteR TBox and A an ABox. The ontology language DL-LiteRcore restricts DL-LiteR by disallowing inclusions and disjointness axioms for roles.

**Semantics of Ontologies** An interpretation I is a pair ⟨ΔI, I⟩, where the domain ΔI is a non-empty set, and the interpretation function I maps each a ∈ I to aI ∈ ΔI such that aI ≠ bI for all a, b ∈ I, each A ∈ C to a subset ΔI of ΔI × ΔI, and each P ∈ R to a subset PΔI of ΔI × ΔI. The interpretation function extends to concepts and roles as follows: (R−1)I = {(u, v) | (v, u) ∈ PI} and (∃RI)I = {u ∈ ΔI | (u, v) ∈ PI for some v ∈ ΔI}

An interpretation I satisfies ABox A if aI ∈ AI for all A(a) ∈ A and (aI, bI) ∈ PI for all P(a, b) ∈ A; I satisfies TBox T if SI ⊑ SI 2 for all S1 ⊆ S2 in T and SI 1 ∩ SI 2 = ∅ for all Disj(S1, S2) in T; I is a model of ontology ⟨T, A⟩ if it satisfies T and A. An ontology is satisfiable if it has a model.

**Queris** A conjunctive query (CQ) q(x) with answer variables x is a formula ∃y. φ(x, y), where x, y are (possibly empty) repetition-free tuples of variables and φ(x, y) is a conjunction of atoms of the form A(t), P(t1, t2) or z = t, where A ∈ C, P ∈ R, z ∈ x ∪ y, and t1, t2 ∈ x ∪ y ∪ I. If z is inessential, then we write q instead of q(x). If x is the empty tuple (∅), then q is Boolean. A union of CQs (UCQ) is a disjunction of CQs with the same answer variables.

The equality atoms in a CQ q(x) = ∃y. φ(x, y) yield an equivalence relation ∼ on terms x ∪ y ∪ I, and we write t for the equivalence class of a term t. The Gaifman graph of q(x) has a node t for each t ∈ x ∪ y ∪ I in φ, and an edge {t1, t2} for each atom in φ over t1 and t2. We assume that all CQs are safe: for each z ∈ x ∪ y, the class z contains a term mentioned in an atom of φ(x, y) that is not an equality.

The certain answers qI of a (U)CQ q(x) over a DL-LiteR ontology K are the set of all tuples α of individuals such that q(α) holds in every model of K. A class of queries Q₁ is rewritable to a class Q₂ for an ontology language O if for any q₁ ∈ Q₁ and TBox T in O, there is a q₂ ∈ Q₂ such that, for any ABox A in O with ⟨T, A⟩ satisfiable, q₁⟨T, A⟩ equals the answers to q₂ in (the least model of) A. Checking a q ∈ Q⟨T, A⟩ for a tuple α, (U)CQ q, and DL-LiteR ontology ⟨T, A⟩ is an NP-complete problem with AC⁰ data complexity (i.e., when T and q are fixed) [Calvanese et al., 2007]. The latter follows from the rewritability of UCQs to themselves for DL-LiteR.

**Bags** A bag over a set M is a function Ω : M → N₀, where N₀ is the set of nonnegative integers and infinity. The value Ω(c) is the multiplicity of c in M. A bag Ω is finite if there are finitely many c ∈ M with Ω(c) > 0 and there is no c with Ω(c) = ∞. The empty bag ∅ over M is the bag such that ∅(c) = 0 for all c ∈ M. Given bags Ω₁ and Ω₂ over M, let Ω₁ ⊑ Ω₂ if Ω₁(c) ≤ Ω₂(c) for each c ∈ M.

The intersection ∩, max union ∪, arithmetic union ⊔, and difference − are the binary operations defined for bags Ω₁ and Ω₂ under the same set M as follows: for every c ∈ M, (Ω₁ ∩ Ω₂)(c) = min{Ω₁(c), Ω₂(c)}, (Ω₁ ∪ Ω₂)(c) = max{Ω₁(c), Ω₂(c)}, (Ω₁ ⊔ Ω₂)(c) = Ω₁(c) + Ω₂(c), and (Ω₁ − Ω₂)(c) = max{0, Ω₁(c) − Ω₂(c)}; difference is well-defined only when Ω₂ is finite.

1 We adopt the unique name assumption for convenience; dropping it does not affect results (modulo minor changes of definitions).
In this section we present a bag semantics for DL-Lite$_R$ ontologies, define the associated query answering problem, and establish its intractability in data complexity.

We formalise ABoxes as bags of facts (rather than sets) in order to faithfully represent the materialised views over source data defined by OBDA mappings.

**Definition 3.** A bag ABox is a finite bag over the set of concept and role assertions. A DL-Lite$_R^b$ ontology is a pair $\langle T, A \rangle$ of a DL-Lite$_R$ TBox $T$ and a bag ABox $A$; the ontology is DL-Lite$_R^{core}$ if $T$ is a DL-Lite$_R^{core}$ TBox.

The semantics of DL-Lite$_R^b$ is based on bag interpretations $I$, with atomic concepts and roles mapped to bags of domain elements and pairs of elements, respectively, and where the interpretation function is extended to complex concepts and roles in the natural way; in particular, a concept $\exists P$ is interpreted as the bag projection of $P^I$ to the first component, where each occurrence of a pair $(u, v)$ in $P^I$ contributes to the multiplicity of domain element $u$ in $(\exists P)^I$.

**Definition 4.** A bag interpretation $I$ is a pair $\langle \Delta^I, \mathcal{L}^I \rangle$ defined the same as in the set case with the exception that $\Delta^I$ and $P^I$ are bags (not sets) over $\Delta^I$ and $\Delta^I \times \Delta^I$, respectively. The interpretation function extends to concepts and roles as follows: $(P^I)^I$ maps each $(u, v) \in \Delta^I \times \Delta^I$ to $P^I(u, v)$, and $(\exists R)^I$ maps each $u \in \Delta^I$ to $\sum_{v \in \Delta^I} R^I(v, u)$.

The definition of semantics of ontologies is as expected.

**Definition 5.** A bag interpretation $I = \langle \Delta^I, \mathcal{L}^I \rangle$ satisfies a bag ABox $A$ if $A(A(a)) \leq \Delta^I(a^I)$ for each concept assertion $A(a)$ in $A$ and $A(P(a, b)) \leq P^I(a^I, b^I)$ for each role assertion $P(a, b)$.

**Definition 7.** Let $q(x) = \exists y. \phi(x, y)$ be a CQ. The bag answers $q^I$ to $q$ over a bag interpretation $I = \langle \Delta^I, \mathcal{L}^I \rangle$ are defined as the bag over tuples of individuals from $\mathcal{I}$ of the same size as $x$ such that, for every such tuple $a$,

$$q^I(a) = \sum_{\lambda \in \Lambda} \prod_{S(t) \in \phi(a, x, y)} S^I(\lambda(t)),$$

where $\Lambda$ is the set of all valuations $\lambda: x \cup y \cup \mathcal{I} \rightarrow \Delta^I$ such that $\lambda(x) = a^I$, $\lambda(a) = a^I$ for each $a \in \mathcal{I}$, and $\lambda(z) = \lambda(t)$ for each $z = t$ in $\phi(x, y)$.

If $q$ is Boolean then $q^I$ are defined only for the empty tuple $\emptyset$. Also, conjunction $\phi(x, y)$ may contain repeated atoms, and hence can be seen as a bag of atoms; while repeated atoms are redundant in the set case, they are essential in the bag setting [Chaudhuri and Vardi, 1993] and thus the definition of $q^I(a)$ treats each copy of a query atom $S(t)$ separately.

The following definition of certain answers, capturing open-world query answering, is a reformulation of the definition in [Kostylev and Reutter, 2015] for counting queries. It is a natural extension of the set notion to bags: a query answer is certain for a given multiplicity if it occurs with at least that multiplicity in every bag model of the ontology.

**Definition 8.** The bag certain answers $q^K$ of a query $q$ over a DL-Lite$_R^b$ ontology $K$ are the bag $\langle I_{\mathcal{Q}} \rangle = q^K$.

We study the problem BAGCERT($Q$, $O$) of checking, given a query $q$ from a class of CQs $\mathcal{Q}$, ontology $K = \langle T, A \rangle$ from an ontology language $O$, tuple $a$ over $\mathcal{I}$, and number $k \in \mathbb{N}$, whether $q^K(a) \geq k$; data complexity of BAGCERT is studied under the assumption that $T$ and $q$ are fixed. Following [Grumbach and Milo, 1996], we assume that the multiplicities of assertions in $A$ and $k$ (if not infinity) are given unary.

**Example 9.** Let $q_{ex}(x) = \exists y. \text{hasMngr}(x, y)$ and $K_{ex}$ be as in Example 6. Then $q_{ex}^{\mathcal{Q}}(\text{Lee}) = 3$. Indeed, on the one hand, $q_{ex}^{\mathcal{Q}}(\text{Lee}) = 3$ for $I_{\mathcal{Q}}$ in Example 6. On the other, for any bag model $I$ of $K_{ex}$, $q_{ex}^{\mathcal{Q}}(\text{Lee}) = \sum_{x \in \Delta^I} \text{hasMngr}(\text{Lee}^I, x) \geq 3$, because $A_{ex}(\text{SalEmp}(\text{Lee})) = 3$ and $T_{ex}$ contains inclusions $\text{SalEmp} \sqsubseteq \text{Emp}$ and $\text{Emp} \sqsubseteq \exists \text{hasMngr}$.

The bag semantics can be seen as a generalisation of the set semantics of DL-Lite: first, satisfiability under bag semantics reduces to the set case; second, certain answers under bag and set semantics coincide if multiplicities are ignored.

**Proposition 10.** Let $\langle T, A \rangle$ be a DL-Lite$_R$ ontology and $\langle T, A' \rangle$ be a DL-Lite$_R^b$ ontology with the same TBox such that $\{S(t) \mid A'(S(t)) \geq 1\} = A$. Then, the following holds: 1. $\langle T, A \rangle$ is satisfiable if and only if $\langle T, A' \rangle$ is satisfiable; 2. for each CQ $q$ and tuple $a$ of individuals from $\mathcal{I}$, $a \in q^{(T, A)}$ if and only if $q^{(T, A')}(a) \geq 1$.

An important property of satisfiable DL-Lite$_R$ ontologies $K$ is the existence of so called universal models for CQs, that is, models $\mathcal{I}$ such that the certain answers to every CQ $q$ over $\mathcal{I}$ can be obtained by evaluating $q$ over $\mathcal{I}$ [Calvanese et al., 2007]. This notion extends naturally to bags.

**Definition 11.** A bag model $I$ of a DL-Lite$_R^b$ ontology $K$ is universal for a class of queries $\mathcal{Q}$ if $q^K = q^\mathcal{Q}$ for any $q \in \mathcal{Q}$.
Proposition 12. There exists a satisfiable DL-Lite$_{bag}$ ontology that has no universal bag model for the class of all CQs.

The lack of a universal model suggests that CQ answering under bag semantics is harder than in the set case. Indeed, this problem is coNP-hard in data complexity, which is in stark contrast to the AC$^0$ upper bound in the set case.

Theorem 13. BAGCERT[CQs, DL-Lite$_{bag}$core] is coNP-hard in data complexity.

4 Universal Models for Rooted Queries

Theorem 13 suggests that bag semantics is generally not well-suited for OBDA. Our approach to overcome this negative result is to consider a restricted class of CQs, introduced in the context of query optimisation in DLs [Bienvenu et al., 2012], called rooted: in a rooted CQ, each existential variable is connected in the Gaifman graph to an individual or an answer variable. Rooted CQs capture most practical queries; for example, they include all connected non-Boolean CQs.

Definition 14. A CQ $q(x)$ is rooted if each connected component of its Gaifman graph has a node with a term in $x \cup I$.

In contrast to arbitrary CQs, any satisfiable DL-Lite$_{bag}$ ontology admits a universal bag model for rooted CQs. Although we define such a model, called canonical, in a fully declarative way, it can be intuitively seen as the result of applying a variant of the restricted chase procedure [Calì et al., 2013] extended to bags. Starting from the ABox, the procedure successively “repairs” violations of $T$ by extending the interpretation of concepts and roles in a minimal way.

To formalise canonical models, we need two auxiliary notions. First, the concept closure $ccl_T[u, I]$ of an element $u \in \Delta^2$ in a bag interpretation $I = \langle \Delta^2 \times X \rangle$ over a TBox $T$ is the bag of concepts such that, for any concept $C$, $ccl_T[u, I](C)$ is the maximum value of $C^i_u(u)$ amongst all concepts $C^0$ satisfying $T = C^0 \subseteq C$. Second, the union $I \cup J$ of bag interpretations $I = \langle \Delta^2 \times X \rangle$ and $J = \langle \Delta^3 \times X \rangle$ with $a^2 = a^3$ for all $a \in I$ is the bag interpretation $\langle \Delta^2 \cup \Delta^3 \times X \cup J \rangle$ with $a^{2 \cup J} = a^2$ for $a \in I$ and $S^2 \cup J = S^2 \cup S^3$ for $S \subseteq C \cup R$.

Definition 15. The canonical bag model $C(K)$ of a DL-Lite$_{bag}$ core ontology $K = \langle T, A \rangle$ is the bag interpretation $\bigcup_{i \geq 0} C_i(K)$ with the bag interpretations $C_i(K) = \langle C_i(K), (K_i) \rangle$ defined as follows:

- $\Delta C_0 = I$, $a^{C_0}(a) = a$ for each $a \in I$, and $S^{C_0}(a) = A(S(a))$ for each $S \subseteq C \cup R$ and individuals $a$;
- for each $i > 0$, $\Delta C_i(K)$ is

  $\Delta C_i(K) = \{(w^d_{u, R}, \ldots, w^d_{u, R}) | u \in \Delta C_{i-1}(K), R$ a role, $\delta = ccl_T[u, C_{i-1}(K)](\exists R) - (\exists R)C_{i-1}(K)(u)\},$

where $w^d_{u, R}$ are fresh domain elements, called anonymous,

- $a^{C_i}(a) = a$ for all $a \in I$, and, for all $A \subseteq C$, $P \subseteq R$, and elements $u, v$,

  $A^{C_i}(u, v) = \{ccl_T[u, C_{i-1}(K)](A), \text{ if } u \in \Delta C_{i-1}(K), \}$

and

- $\Delta C_i(K) = \{ccl_T[u, C_{i-1}(K)](A), \text{ if } u \in \Delta C_{i-1}(K), \}$

0, otherwise,

$P^{C_i}(u, v) = \{c_{i-1}(K)(u, v), \text{ if } u, v \in \Delta C_{i-1}(K), \}$

1, if $v = w^{d}_{u, P}$ or $u = w^{d}_{v, P}, \}$

0, otherwise.

It is easily seen that $C(K)$ satisfies $K$ whenever $K$ is satisfiable. We next show that it is universal for rooted CQs.

Theorem 16. The canonical bag model $C(K)$ of a satisfiable DL-Lite$_{bag}$ core ontology $K$ is universal for rooted CQs.

Example 17. Consider an ontology $K_r = \langle T_r, A_r \rangle$ with

$T_r = \{\text{Emp} \subseteq \text{hasMngr}, \text{hasMngr} \subseteq \text{Mngr}\},$

$A_r(\text{Emp}(\text{Lee})) = A_r(\text{Mngr}(\text{Hill})) = 1.$

The canonical model $C(K_r)$ interprets (all with multiplicity 1) $\text{Emp}$ by $\text{Lee}$, $\text{Mngr}$ by $\text{Hill}$ and $w^1_{\text{Emp}, \text{hasMngr}}$, and $\text{hasMngr}$ by $(\text{Lee}, w^1_{\text{Lee}, \text{hasMngr}})$. Note that $C(K_r)$ is not universal for all CQs: for instance, $q_{\text{cnr}}(\langle \rangle) = 2$ for non-rooted $q_{\text{cnr}} = \exists x. \text{Mngr}(y)$, but $q_{\text{cnr}}(\langle \rangle) = 1$ for the model $T_{\text{ar}}$ interpreting $\text{Emp}$ by $\text{Lee}$, $\text{Mngr}$ by $\text{Hill}$, and $\text{Mngr}$ by $\text{Hill}$.

We conclude this section by showing an important property of rooted CQs, which justifies their favourable computational properties. As in the set case for arbitrary CQs, given a satisfiable DL-Lite$_{bag}$ core ontology $K$ and a rooted CQ $q$, $q_{\text{cnr}}$ can be computed over a small sub-interpretation of $C(K)$.

Theorem 18. Let $K$ be a satisfiable DL-Lite$_{bag}$ core ontology with $C(K) = \bigcup_{i \geq 0} C_i(K)$ and $q$ be a rooted CQ having $n$ atoms. Then, $q^{C_i}(K) = q^{C_i}(K)$.

5 Rewritability of Rooted Queries

Rewritability is key for OBDA, and we next establish to what extent rooted CQs over bag semantics are rewritable.

The first idea would be to use the analogy with the set case and rewrite to unions of CQs. There are two corresponding operations for bags: max union $\cup$ and arithmetic union $\cup$. So we may consider max unions $q_{\text{max}} = q_1(x) \lor \cdots \lor q_n(x)$ or arithmetic unions $q_{\text{ar}} = q_1(x) \lor \cdots \lor q_n(x)$ of CQs $q_i(x)$, $1 \leq i \leq n$, with the following semantics, for any interpretation $I$: $q_{\text{max}} = q_1^I \lor \cdots \lor q_n^I$ and $q_{\text{ar}} = q_1^I \lor \cdots \lor q_n^I$, respectively. Our first result is negative: rewriting to either of these classes is not possible even for DL-Lite$_{bag}$ core.

Proposition 19. The class of rooted CQs is rewritable neither to max nor to arithmetic unions of CQs for DL-Lite$_{bag}$ core.

Next we show that rooted queries are rewritable to BALG$_1$-queries: the class directly corresponding to the algebra BALG$_1$ for bags [Grumbach et al., 1996; Grumbach and Milo, 1996, 1997; Libkin and Wong, 1997]. Since BALG$_1$ \subseteq LOGSPACE [Grumbach and Milo, 1996], where BALG$_1$ is the complexity class for BALG$_1$ algebraic evaluation, rewritability to BALG$_1$-queries is highly desirable.

Intuitively, in addition to projection $\exists$, join $\land$, and unions $\lor$ and $\lor$, BALG$_1$ also allows for difference $\setminus$. Domain-dependent queries, inexpressible in algebraic query languages, are precluded by restrictions on the use of variables.

Definition 20. A BALG$_1$-query $q(x)$ with answer variables $x$ is one of the following, where $q_i$ are BALG$_2$-queries:

- $S(t)$, for $S \subseteq C \cup R$, $t$ tuple over $x \cup I$ mentioning all $x$;
- $q_1(x_1) \land q_2(x_2)$, for $x = x_1 \cup x_2$;
- $q_0(x_0) \land (x = t)$, for $x \subseteq x_0$, $t \subseteq X \cup I$, $x = x_0 \cup \{(t) \setminus I\}$;
- $\exists y. q_0(x, y) \lor q_1(x) \lor q_2(x) \lor q_1(x) \lor q_2(x)$.

1227
The semantics of $\text{BALG}_1$-queries is defined as follows.

**Definition 21.** The bag answers $q_I$ to a $\text{BALG}_1$-query $q(x)$ over a bag interpretation $I = (\Delta^F, ^C)$ is the bag of tuples over $I$ of the same size as $x$ inductively defined as follows, for each tuple $a$ and the corresponding mapping $\lambda$ such that $\lambda(x) = a^2$ and $\lambda(a) = a^2$ for all $a \in I$:

- $S^2(\lambda(t))$, if $q(x) = S(t)$;
- $q_1^2(\lambda(x_1)) \times q_2^2(\lambda(x_2))$, if $q(x) = q_1(x_1) \wedge q_2(x_2)$;
- $q_0^2(\lambda(x_0))$, if $q(x) = q_0(x_0) \vee (x = t)$ and $\lambda(x) = \lambda(t)$;
- $0$, if $q(x) = q_0(x_0) \wedge (x = t)$ and $\lambda(x) = \lambda(t)$;
- $\sum_{x_1, y \in \Delta^F} (q_1^2 \circ q_0^2)(a^2)$, if $q(x) = \exists y. q_0(x, y)$;
- $(q_1^2 \circ q_2^2)(a^2)$, if $q(x) = q_1(x_1) \circ p_1 q_2(x_2)$, where $p$ is $\cup$, $\vee$, or $\circ$.

The data complexity of $\text{BALG}_1$-query evaluation is obtained by showing that $\text{BALG}_1$-queries can be mapped to the $\text{BALG}_2$ algebra of [Grumbach and Milo, 1996].

**Proposition 22.** Given a fixed $\text{BALG}_1$-query $q(x)$, the problem of checking whether $q^{C(0, A)}(a) \geq k$ for a bag ABox $A$, tuple $a$, and $k \in \mathbb{N}^\infty$ is AC$^0$ reducible to $\text{BALG}_2$.

Our rewriting algorithm is inspired by the algorithm in [Kikot et al., 2012] for the set case of DL-Lite$_\mathcal{Q}$. Before going into details, we provide a high-level description.

The key observation is that the set of valuations of a CQ $q(x) = \exists y. \phi(x, y)$ over the bag canonical model $C(K)$ can be partitioned into subsets, each of which is characterised by variables $z \subseteq y$ that are sent to anonymous elements of $C(K)$. Hence, we can rewrite $q(x)$ for each of these subsets separately and then take an arithmetic union of the resulting queries, provided these queries are guaranteed to give the same answers as the corresponding subsets of valuations.

Our rewriting proceeds along the following steps.

**Step 1.** First, each $z$ is checked for realizability, that is, whether the subquery induced by $z$ can indeed be folded into the anonymous forest-shaped part of $C(K)$. This can be done without the ABox, looking only at the atoms of $q$ that link $z$ to other terms of $q$ (these linking atoms exist because $q$ is rooted). Non-realizable $z$ can be disregarded.

**Step 2.** For every realizable $z$, CQ $q(x)$ is replaced (for this $z$ in the arithmetic union) by a CQ $q_{\phi}(x)$ obtained from $q$ by replacing each maximal connected component of the subquery induced by $z$ by just one linking atom. This transformation is equivalence-preserving, because the anonymous part of $C(K)$ does not involve multiplicities other than 0 and 1.

**Step 3.** Finally, each resulting $q_{\phi}(x)$ is rewritten to a $\text{BALG}_1$-query $q_{\phi}(x)$ by “chasing back” each unary atom and each binary atom mentioning a variable in $z$ with the TBox: for the binary atoms it is also guaranteed, by means of difference, that the variable in $z$ is indeed mapped to the anonymous part, thus avoiding double-counting in the arithmetic union.

For the rest of this section, let us fix a rooted CQ $q(x) = \exists y. \phi(x, y)$ and a DL-Lite$_\mathcal{Q}$ TBox $T$. We start by formalising Step 1.

**Definition 23.** Given an ontology $K$ with a TBox $T$ and variables $z \subseteq y$, let $[q, z]^{C(K)}$ be the bag of tuples over $I$ such that, for each tuple $a$ of individuals, $[q, z]^{C(K)}(a) = \sum_{\lambda \in \Lambda_K} \prod_{S(t)} S(t) \oplus \phi(x, y) SC^{C(K)}(\lambda(t))$, where $\Lambda_K$ is the set of valuations $\lambda : x \cup y \cup I \rightarrow \Delta^{C(K)}$ such that $\lambda(x) = a$, $\lambda(a) = a$ for each $a \in I$, $\lambda(x) = \lambda(t)$ for each $x = t$ in $\phi(x, y)$, $\lambda(z)$ is an anonymous element for each $z \in z$, and $\lambda(y) \in I$ for each $y \in y \setminus z$.

Hence, the bag answers to $q$ can be partitioned as follows:

$$q^{C(K)} = \biguplus_{z \subseteq y} [q, z]^{C(K)}.$$

Variables $z \subseteq y$ are equality-consistent if $\phi(x, y)$ has no equality $z = t$ with $z \in z$ and $t \notin z$. If $z$ is not equality-consistent, then $[q, z]^{C(K)} = \emptyset$ and these $z$ can be disregarded in (1). Next, we show which other $z$ can be ignored.

**Definition 24.** Given equality-consistent $z \subseteq y$, variables $z' \subseteq z$ are maximally connected in the anonymous part (ma-connected) if $z' \subseteq z'$ for the equivalence class $z$ of any $z \in z'$ and the equivalence classes $z'$ are a maximal subset of $z$ connected in the Gaifman graph of $q$ via nodes in $z$.

Next we introduce several notations for ma-connected $z' \subseteq z$ with equality-consistent $z \subseteq y$. First, let $\phi_{z'}$ be the sub-conjunction of $\phi(x, y)$ that consists of all atoms mentioning at least one variable in $z'$ (these sub-conjunctions are disjoint for different $z'$). Second, since $q$ is rooted, $\phi_{z'}$ contains an atom $\alpha_{z'}$ of the form $P(t, z)$ or $P(z, t)$ with $z \in z'$ and $t \notin z$ (note that this definition may be non-deterministic). Third, let $q_{\phi_{z'}}() = \exists x'. \phi_{z'} \wedge \bigwedge_{t \in t_{z'}} (t = a) \wedge \bigwedge_{s \in s_{z'}} (s = a)$, where $t_{z'}$ are all such terms $t$, $a$ is an individual in $t_{z'}$ if it exists or a fresh individual otherwise, and $s_{z'} = t_{z'} \cap X$, (this definition may also be non-deterministic because of $a$). Notice that $q_{\phi_{z'}}$ is a Boolean CQ with possible equalities of individuals and inequalities, and we can define the bag answers of such a query $q'$ over a bag interpretation $I$ in the same way as for usual CQs in Definition 7 with the extra requirement that each contributing valuation $\lambda$ should satisfy $\lambda(x) \neq \lambda(t)$ for each inequality $x \neq t$ of $q'$ (and equalities of individuals are handled as usual equalities).

**Definition 25.** Given equality-consistent variables $z \subseteq y$, ma-connected $z' \subseteq z$ are realisable by TBox $T$ if $q_{\phi_{z'}}^{C(\{\tau', \alpha'\})}(\{\}) \geq 1$, where, for a fresh individual $b$, $\alpha'$ is the bag ABox having either only the assertion $P(a, b)$ (with multiplicity 1), when $\alpha_{z'} = P(t, z)$, or only $P(b, a)$, when $\alpha_{z'} = P(z, t)$.

This definition does not depend on the choice of $\alpha_{z'}$ and $a$. Indeed, if there are two atoms $P_1(t_1, z_1)$ and $P_2(z_2, z_2)$ satisfying the definition of $\alpha_{z'}$, then either $P_1 = P_2$ and both pairs $(t_1, z_1)$ and $(z_2, z_2)$ are mapped by a valuation of $q_{\phi_{z'}}$ to the same tuple, or $z'$ are not realisable regardless of the choice of $\alpha_{z'}$. Similarly, if $t_{z'}$ contains two individuals $a, a'$, then $q_{\phi_{z'}}$ has the equality $a = a'$, and hence $z'$ are not realisable regardless of this choice.

Intuitively, $z'$ are realisable if their corresponding subquery $q_{\phi_{z'}}$ is satisfied by the tree-shaped model induced by the TBox from a connection $\alpha_{z'}$ of $z'$ and the rest of the query. This definition does not essentially involve multiplicities, because all tuples of anonymous elements in the canonical model have multiplicity at most 1, and, hence, if $q_{\phi_{z'}}$ matches a part of the canonical model, it does so in a unique way. Thus, checking realizability is decidable using standard set-based techniques.
Definition 26. Variables $z \subseteq y$ are realisable by TBox $T$ if they are equality-consistent and each non-empty ma-connected subset of $z$ is realisable by $T$.

We proceed to Step 2. For realisable $z \subseteq y$, let $q_\alpha(x)$ be the CQ $\exists y'. \phi_z(x, y')$ such that $\phi_z(x, y')$ is obtained from $\phi(x, y)$ by replacing $\phi_\alpha$, for each ma-connected $z' \subseteq z$, with

$$\alpha_{z'} \land \bigwedge_{y \in t_{z'} \cap x, t \in t_{z'}} (y = t),$$

where $t_{z'}$ is as in $q_\alpha$, and $y'$ is the subset of $y$ remaining in $\phi_z$. In other words, $q_\alpha$ contains, for each $z'$, just one atom $\alpha_{z'}$ and equalities identifying $t_{z'}$ instead of conjunction $\phi_\alpha$ in $q$.

The following lemma justifies Steps 1 and 2. It says that in partitioning (1) we only need to iterate over tuples $z$ that are realisable by $T$ and can also replace $q$ with $q_\alpha$ for each $z$.

Lemma 27. For any ontology $K$ with TBox $T$ and $z \subseteq y$ with $q_\alpha(x) = \exists y'. \phi_z(x, y')$,

1. If $z$ is realisable by $T$ then $[q, z]^{(K)} = [q_\alpha, z \cap y']^{(K)}$;
2. If $z$ is not realisable by $T$ then $[q, z]^{(K)} = \emptyset$.

For Step 3, it suffices to rewrite each CQ $q_\alpha(x) = \exists y'. \phi_z(x, y')$ to a BALG$_1^2$-query $\tilde{q}_\alpha(x) = \exists y_z. \psi_z(x, y_z)$, for $y_z = y' \setminus z$, which is guaranteed to give $[q_\alpha, z \cap y']^{(K)}$ as the bag answers on the ABox in any ontology $K$ with TBox $T$. To this end, we use the following notation: for $t \in X \cup 1$, let $\zeta_{A}(t) = A(t)$ for $A \in C$, while $\zeta_{\exists R}(t) = \exists y. \{t(y), y\}$ and $\zeta_{\exists R'}(t) = \exists y. \{P(t, y), y\}$ for $P \in R$, where $y$ is a variable different from $t$. Then, formula $\psi_z(x, y_z)$ is obtained from $\phi_z(x, y')$ by replacing all atoms mentioning a term $t \in I \cup x \cup y_z$ or a variable $z \in z$ as follows:

- each $A(t)$ with $\bigvee_{T \models C \subseteq A} \zeta_{C}(t)$;
- each $P(t, z)$ with $\bigvee_{T \models C \supseteq P} \zeta_{C}(t) \land \zeta_{\exists P}(t)$;
- each $P(z, t)$ with $\bigvee_{T \models C \supseteq \exists P} \zeta_{C}(t) \land \zeta_{\exists P'}(t)$.

Note that $\phi_z(x, y')$ does not contain any atoms of the form $A(z)$ for $z \in z$, so $\psi_z(x, y_z)$ does not mention variables $z$. Also, atoms over roles without variables $z$ stay intact, because $T$ contains no role inclusions.

Finally, the rewriting of $q(x)$ over $T$ is the BALG$_1^2$-query

$$\tilde{q}(x) = \bigvee_{z \subseteq y \text{ realisable by } T} \tilde{q}_\alpha(x).$$

Example 28. Consider TBox $T_r$ from Example 17 and the rooted CQ $q'(x) = \exists y. \text{hasMngr}(x, y) \land \text{Mngr}(y)$. The query $\tilde{q}'(x) = \tilde{q}(x) \lor \tilde{q}_9(x)$, where $\tilde{q}(x)$ and $\tilde{q}_9(x)$ are

$$\exists y. \text{hasMngr}(x, y) \land \{\text{Mngr}(y) \lor \exists x. \text{hasMngr}(x, y)\},$$

is a rewriting of $q'$ over $T_r$, since $\{x\}$ and $y$ are realisable.

The following theorem establishes the correctness of our approach and leads to the main rewritability result.

Theorem 29. For any rooted CQ $q$ and DL-Lite$_{bag \text{core}}$ ontology $K = (T, A)$ we have that $q^{(K)}(\emptyset, A) = q^{(0, 1)}$. Proof.

Corollary 30. The class of rooted CQs is rewriteable for BALG$_1^2$-queries for DL-Lite$_{core}^{bag}$.

We conclude this section by establishing the complexity of rooted query answering. The bounds follow as an easy consequence of Theorem 18, Proposition 22, and Corollary 30.
References


