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 the sources using mappings, which assign views over the data to ontology predicates. Motivated by the need for OBDA systems supporting database-style 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-Lite \( \mathcal{R} \) [Calvanese et al., 2007], which underpins OWL 2 QL [Motik et al., 2012]. DL-Lite \( \mathcal{R} \) 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 \( \text{SalEmployee}(id, \text{name}, \text{salary}, \text{loc}, \text{mngr}) \), where attributes id, name, salary, loc, and 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 \( \text{ITEmployee}(id, \text{surname}, \text{salary}, \text{city}) \), where managers are not specified. To integrate employee data, the company relies on an ontology with TBox \( \mathcal{T}_{\text{ex}} \), which defines unary predicates such as \( \text{SalEmp}, \text{ITEmp}, \text{Mngr} \), 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:

\[
\begin{align*}
\text{SalEmployee(name, att_1)} & \rightarrow \text{SalEmp(name)}, \\
\text{SalEmployee(name, mngr, att_2)} & \rightarrow \text{hasMngr(name, mngr)}, \\
\text{SalEmployee(mngr, att_3)} & \rightarrow \text{Mngr(mngr)}, \\
\text{ITEmployee(surname, att_4)} & \rightarrow \text{ITEmp(surname)}. \\
\end{align*}
\]

TBox \( \mathcal{T}_{\text{ex}} \) 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 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 SalesEmp(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 SalesEmp 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-Lite\(_R\) ontologies under bag semantics, and our main contributions are as follows.

1. We propose the ontology language DL-Lite\(_R_{bag}\) and its restriction DL-Lite\(_{bag_{core}}\), 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 \(\text{coNP}\)-hard in data complexity even if we restrict ourselves to DL-Lite\(_R_{bag_{core}}\) 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-Lite\(_R_{bag_{core}}\) 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 [Nikolaou et al., 2017].

2 Preliminaries

Syntactic 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 \(P \in R\) or its inverse \(P^{-}\). A concept is an atomic concept in \(C\) or an expression \(\exists R\), where \(R\) is a role. An inclusion is an expression of the form \(S_1 \sqsubseteq S_2\) either both concepts or both roles. A disjunction axiom is an expression of the form \(\text{Disj}(S_1, S_2)\) with \(S_1\) and \(S_2\) either both concepts or both roles. A concept assertion is of the form \(A(a)\) with \(a \in I\) and \(A \in C\). A role assertion is of the form \(P(a, b)\) with \(a, b \in I\) and \(P \in R\). A DL-Lite\(_R\) TBox is a finite set of inclusions and disjunction axioms. An ABox is a finite set of concept and role assertions. A DL-Lite\(_R\) ontology is a pair \(\langle T, A \rangle\) with \(T\) a DL-Lite\(_R\) TBox and \(A\) an ABox. The ontology language DL-Lite\(_R_{core}\) restricts DL-Lite\(_R\) by disallowing inclusions and disjunction axioms for roles.

Semantics of Ontologies An interpretation \(I\) is a pair \(\langle \Delta^I, \cdot^I \rangle\), where the domain \(\Delta^I\) is a non-empty set, and the interpretation function \(\cdot^I\) maps each \(a \in I\) to \(a^I \in \Delta^I\) such that \(a^I \neq b^I\) for all \(a, b \in I\), each \(A \in C\) to a subset \(A^I\) of \(\Delta^I\) and each \(P \in R\) to a subset \(P^I\) of \(\Delta^I \times \Delta^I\). The interpretation function extends to concepts and roles as follows: \((\forall x)^I = \{(u, v) \mid (v, u) \in R^I\}\) and \((\exists x)^I = \{u \in \Delta^I \mid (u, v) \in R^I\}\) for some \(v \in \Delta^I\).

An interpretation \(I\) satisfies ABox \(A\) if \(a^I \in A^I\) for all \(A(a) \in A\) and \((a^I, b^I) \in P^I\) for all \(P(a, b) \in A\). \(I\) satisfies TBox \(T\) if \(S^I_1 \subseteq S^I_2\) and \(P^I_1 \cap P^I_2 = \emptyset\) for all \(S^I_1 \subseteq S^I_2\) in \(T\) and \(P^I_1 \cap P^I_2 = \emptyset\) for all \(S^I_1 \cap S^I_2 = \emptyset\) in \(T\). \(I\) is a model of ontology \(\langle T, A \rangle\) if it satisfies \(T\) and \(A\). An ontology is satisfiable if it has a model.

Queries A conjunctive query (CQ) \(q(x)\) with answer variables \(x\) is a formula \(3y. \phi(x, y)\), where \(x, y\) are (possibly empty) repetition-free tuples of variables and \(\phi(x, y)\) is a conjunction of atoms of the form \(A(t)_1\), \(P(t_1, t_2)_2\) or \(z = t\), where \(A \in C\), \(P \in R\), \(z \in x \cup y\), and \(t, t_1, t_2 \in x \cup y \cup I\). If \(x\) is inessential, then we write \(q\) instead of \(q(x)\). If \(x\) is the empty tuple \(\langle \rangle\), 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 \(q(x) = \exists y. \phi(x, y)\) yield an equivalence relation \(\sim\) on terms \(x \cup y \cup I\), and we write \(I\) for the equivalence class of a term \(t\). The Gaifman graph of \(q(x)\) has a node \(t\) for each \(t \in x \cup y \cup I\) in \(\phi\), and an edge \(\{t_1, t_2\}\) for each atom in \(\phi\) over \(t_1\) and \(t_2\). We assume that all CQs are safe: for each \(x \in x \cup y\), the class \(x\) contains a term mentioned in an atom of \(\phi(x, y)\) that is not an equality.

The certain answers \(q^{\Delta}\) to a \((U)CQ\) \(q(x)\) over a DL-Lite\(_R\) ontology \(K\) are the set of all tuples \(a\) of individuals such that \(q(a)\) holds in every model of \(K\). A class of queries \(Q_1\) is rewritable to a class \(Q_2\) for an ontology language \(O\) if for any \(q_1 \in Q_1\) and TBox \(T\) in \(O\), there is \(q_2 \in Q_2\) such that, for any ABox \(A\) in \(O\) with \(\langle T, A \rangle\) satisfiable, \(q_1^{\langle T, A \rangle}\) equals the answers to \(q_2\) in (the least model of) \(A\). Checking a \(q \in Q^{\langle T, A \rangle}\) for a tuple \(a\), \((U)CQ\) \(q\), and DL-Lite\(_R\) ontology \(\langle T, A \rangle\) is an NP-complete problem with AC^0 data complexity (i.e., when \(T\) and \(q\) are fixed) [Calvanese et al., 2007]. The latter follows from the rewritability of UCCs to themselves for DL-Lite\(_R\).

Bags A bag over a set \(M\) is a function \(\Omega : M \rightarrow \mathbb{N}_0^M\), where \(\mathbb{N}_0^M\) is the set of nonnegative integers and infinity. The value \(\Omega(c)\) is the multiplicity of \(c\) in \(M\). A bag \(\Omega\) is finite if there are finitely many \(c \in M\) with \(\Omega(c) > 0\) and there is no \(c\) with \(\Omega(c) = \infty\). The empty bag \(\emptyset\) over \(M\) is the bag such that \(\Omega(c) = 0\) for all \(c \in M\). Given bags \(\Omega_1\) and \(\Omega_2\) over \(M\), let \(\Omega_1 \subseteq \Omega_2\) if \(\Omega_1(c) \leq \Omega_2(c)\) for each \(c \in M\).

The intersection \(\cap\), max union \(\cup\), arithmetic union \(\oplus\), and difference \(\setminus\) are the binary operations defined for bags \(\Omega_1\) and \(\Omega_2\) over the same set \(M\) as follows: for every \(c \in M\), \((\Omega_1 \cap \Omega_2)(c) = \min(\Omega_1(c), \Omega_2(c))\), \((\Omega_1 \cup \Omega_2)(c) = \max(\Omega_1(c), \Omega_2(c))\), \((\Omega_1 \oplus \Omega_2)(c) = (\Omega_1(c) + \Omega_2(c))\), and \((\Omega_1 - \Omega_2)(c) = \max(0, \Omega_1(c) - \Omega_2(c))\); difference is well-defined only when \(\Omega_2\) is finite.

\footnote{We adopt the unique name assumption for convenience; dropping it does not affect results (modulo minor changes of definitions).}
3 DL-Lite_R with Bag Semantics

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^bag ontology is a pair (T, A) 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^bag 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 A^I 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 (A^I)^2.

Definition 4. A bag interpretation I is a pair (A^T, A) defined the same as in the set case with the exception that A^2 and P^2 are bags (not sets) over A^T and A^T x A^T, respectively. The interpretation function extends to concepts and roles as follows: (P^-^I) maps each (u, v) in A^T x A^T to P^I(u, v), and (∃R^I) maps each u in A^T to \sum_{v \in A^T} R^I(u, v).

The definition of semantics of ontologies is as expected.

Definition 5. A bag interpretation I = (T, A) satisfies a bag ABox A if A(A(a)) ≤ A^T(a^T) for each concept assertion A(a) in A and A(P(a, b)) ≤ P^I(a^T, b^T) for each role assertion P(a, b). Satisfaction of T is defined as in the set case, except that ⊆ and ∩ are applied to bags instead of sets.

Bag interpretation I is a bag model of the DL-Lite_R^bag ontology (T, A), written I |=^b (T, A), if it satisfies both T and A. The ontology is satisfiable if it has a bag model.

Example 6. Let K_ex = (T_ex, A_ex) be a DL-Lite_R^bag ontology with T_ex as in Example 1 and A_ex has hasEmp(lee) with multiplicity 3, Itemp(lee) and hasMngr(lee, hill) both with multiplicity 2 (and all other assertions with multiplicity 0). Let I_ex be the bag interpretation mapping individuals to themselves and with the following non-zero values:

SaeEmp^T_ex(lee) = Emp^T_ex(lee) = 3, Itemp^T_ex(lee) = 2,
hasMngr^T_ex(lee, hill) = 2, hasMngr^T_ex(lee, w) = 1,
Mngr^T_ex(hill) = 2, Mngr^T_ex(w) = 1,

where w is a fresh element. We can check that I_ex |=^b K_ex.

We now define the notion of query answering under bag semantics. We first define the answers q^I of a CQ q(x) over a bag interpretation I. Intuitively, q^I is a bag of tuples of individuals such that each valid embedding λ of the body of q into I contributes separately to the multiplicity of the tuple λ(x) in q^I; in turn, the contribution of each specific λ is the product of the multiplicities of the images of the query atoms under λ. The latter is in accordance with the interpretation of joins in the bag relational algebra and SQL, where the multiplicity of a tuple in a join is the product of the multiplicities of the joined tuples (e.g., see [Garcia-Molina et al., 2009]).

Definition 7. Let q(x) = \exists y. ϕ(x, y) be a CQ. The bag answers q^I to q over a bag interpretation I = (A^T, A) are defined as the bag over tuples of individuals from I of the same size as x such that, for every such tuple a,

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

where \(\Lambda\) is the set of all valuations \(\lambda : x \cup y \cup I \rightarrow A^T\) such that \(\lambda(x) = a^T\), \(\lambda(a) = a^T\) for each \(a \in I\), 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 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 if 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 to a query q over a DL-Lite_R^bag ontology K are the bag \(\{I \models q^K\}\).

We study the problem BAGCERT(K, O) of checking, given a query q from a class of CQs K, ontology K = (T, A) from an ontology language O, tuple a over I, and number k \(\in \mathbb{N}_0\), 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. hasMngr(x, y) and K_ex be as in Example 6. Then q^K_ex(lee) = 3. Indeed, on the one hand, q^K_ex(lee) = 3 for I_ex in Example 6. On the other, for any bag model I of K_ex, q^K_ex(lee) = \sum_{\lambda \in \Lambda} hasMngr^I(lee^I, y) \geq 3, because A_ex(hasEmp(lee)) = 3 and T_ex contains inclusions hasEmp \subseteq hasEmp and 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 (T, A) be a DL-Lite_R ontology and (T', A') be a DL-Lite_R^bag ontology with the same TBox such that \{S(t) | A'(S(t)) \geq 1\} = A. Then, the following holds:

1. (T, A) is satisfiable if and only if (T', A') is satisfiable;
2. for each CQ q and tuple a of individuals from I, a \(\in q^{(T',A')}(a)\) 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 I such that the certain answers to every CQ q over K can be obtained by evaluating q over I [Calvanese et al., 2007]. This notion extends naturally to bags.

Definition 11. A bag model I of a DL-Lite_R^bag ontology K is universal for a class of queries Q if q^K = q^I for any q \(\in Q\).

Unfortunately, in contrast to the set case, even DL-Lite_R^core ontologies may not admit a universal bag model for all CQs.
Proposition 12. There exists a satisfiable DL-Lite\textsubscript{bag}\textsubscript{core} 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\textsuperscript{0} upper bound in the set case.

Theorem 13. BagCert{CQs, DL-Lite\textsubscript{bag}\textsubscript{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 Theorem 13. There exists a satisfiable DL-Lite\textsubscript{bag} ontology C(K) that admits a universal bag model for root 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 [Call 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\textsubscript{T} [u, T] of an element u ∈ Δ\textsuperscript{2} in a bag interpretation T = (Δ\textsuperscript{2}, I) over a TBox T is the bag of concepts such that, for any concept C, ccl\textsubscript{T} [u, T](C) is the maximum value of C\textsubscript{0} (u) amongst all concepts C\textsubscript{0} satisfying T = C\textsubscript{0} ⊆ C. Second, the union I∪J of bag interpretations T = (Δ\textsuperscript{2}, I) and J = (Δ\textsuperscript{2}, J) with a\textsuperscript{2} = a\textsuperscript{2} for all a ∈ I is the bag interpretation (Δ\textsuperscript{2} ∪ Δ\textsuperscript{2}, I∪J) with a\textsuperscript{2} = a\textsuperscript{2}(a) = a\textsuperscript{2} for a ∈ I and S\textsuperscript{2} (J) = S\textsuperscript{2} (I)∪S\textsuperscript{2} (J) for S ∈ D \cup R.

Definition 15. The canonical bag model C(K) of a DL-Lite\textsubscript{bag} ontology K = (T, A) is the bag interpretation \[∪_{i>0} \mathcal{C}_{i}(K) \] with the bag interpretations \[\mathcal{C}_{i}(K) = (\Delta \mathcal{C}_{i}(K), \mathcal{C}_{i}(K)) \] defined as follows:

- Δ\textsubscript{C\textsubscript{0}} (K) = I, \(a\textsubscript{C\textsubscript{0}}(a) = a \text{ for each } a ∈ I, \) and \(S\textsubscript{C\textsubscript{0}}(a) = \) \(A(S(a)) \) for each \(S ∈ C \cup R \) and individuals \(a; \)
- for each \(i > 0, \Delta \mathcal{C}_{i}(K) \) is \(\Delta \mathcal{C}_{i-1}(K) = \{w\textsubscript{u,R}, \ldots, w\textsubscript{u,R} \mid u ∈ Δ \mathcal{C}_{i-1}(K), R \) a role, \(\delta \text{ ccl}_T [u, C_{i-1}(K)](\exists R) - (\exists R) \mathcal{C}_{i-1}(K)(u)\}, \) where \(w\textsubscript{u,R} \) are fresh domain elements, called anonymous, \(a\textsubscript{C_{i}}(a) = a \text{ for each } a ∈ I, \text{ and for all } A ∈ C, P ∈ R, \) and elements \(u, v, \)
- \(A\textsubscript{C_{i}}(K) (u) = \{\) \text{ccl}_T [u, C_{i-1}(K)] (A) if \(u ∈ Δ \mathcal{C}_{i-1}(K), \) 
- otherwise,
- \(p\textsubscript{C_{i}}(K) (u,v) = \{\) \text{ccl}_T [u, C_{i-1}(K)] (A) if \((u,v) ∈ Δ \mathcal{C}_{i-1}(K), \) 
- otherwise.

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

Theorem 16. The canonical bag model C(K) of a satisfiable DL-Lite\textsubscript{bag} ontology K is universal for root CQs.

Example 17. Consider an ontology K\textsubscript{r} = (T\textsubscript{r}, A\textsubscript{r}) with

\[
T\textsubscript{r} = \{\text{Emp} ⊆ \exists\text{hasMngr, } \exists\text{hasMngr} ⊆ \text{Mngr}\},
\]
\[
A\textsubscript{r}(\text{Emp}(\text{Lee})) = A\textsubscript{r}(\text{Mngr}(\text{Hill})) = 1.
\]

The canonical model C(K\textsubscript{r}) interprets (all with multiplicity 1) Emp by Lee, Mng by Hill and w\textsubscript{1}\textsubscript{Lee,hasMngr} and hasMngr by (Lee, w\textsubscript{1}\textsubscript{Lee,hasMngr}). Note that C(K\textsubscript{r}) is not universal for all CQs: for instance, \(q\textsubscript{ar} (\text{C}) = 2) \) for non-rooted q\textsubscript{ar} = \(∃y. \text{Mngr}(y), \) but \(q\textsubscript{ar} (\text{C}) = 1 \) for the model T\textsubscript{r} interpreting Emp by Lee, hasMngr by (Lee, Hill), and Mng by Hill.

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

Theorem 18. Let K be a satisfiable DL-Lite\textsubscript{bag} ontology with C(K) = ∪_{i≥0} C\textsubscript{i}(K) and q be a rooted CQ having n atoms. Then, \(q\textsubscript{C}\textsubscript{r} = q\textsubscript{C}\textsubscript{i}(K). \)

5 Rewritability of Rooted Queries

Rewritability is key for OBDA, and we next establish to what extent root 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 ∪ and arithmetic union ∪. So we may consider max unions \(q\textsubscript{max} = q\textsubscript{1}(x) \lor \cdots \lor q\textsubscript{n}(x) \) and arithmetic unions \(q\textsubscript{ar} = q\textsubscript{1}(x) \lor \cdots \lor q\textsubscript{n}(x) \) of CQs q\textsubscript{i}(x), \(1 ≤ i ≤ n, \) with the following semantics, for any interpretation T: \(q\textsubscript{max} = q\textsubscript{1}(x) \lor \cdots \lor q\textsubscript{n}(x) \) and \(q\textsubscript{ar} = q\textsubscript{1}(x) \lor \cdots \lor q\textsubscript{n}(x) \), respectively. Our first result is negative: rewriting to either of these classes is not possible even for DL-Lite\textsubscript{bag}.

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

Next we show that rooted queries are rewritable to BALG\textsubscript{1}-queries: the class directly corresponding to the algebra BALG\textsubscript{1} for bags [Grumbach et al., 1996; Grumbach and Milo, 1996; Libkin and Wong, 1997]. Since BALG\textsubscript{1} ⊆ LOGSPACE [Grumbach and Milo, 1996], where BALG\textsubscript{1} is the complexity class for BALG\textsubscript{1} algebra evaluation, rewritability to BALG\textsubscript{1}-queries is highly desirable.

Intuitively, in addition to projection ∃, join ∧, and unions ∨ and ∨, BALG\textsubscript{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\textsubscript{1}-query q(x) with answer variables x is one of the following, where q\textsubscript{i} are BALG\textsubscript{1}-queries:

- S(t), for S ∈ C \cup R, t tuple over \(x \cup \{t\} \) mentionining all x;
- \(q\textsubscript{1}(x) \land q\textsubscript{2}(x)\), for x = x\textsubscript{1} ∪ x\textsubscript{2};
- \(q\textsubscript{1}(x) \land (x = t)\), for x ∈ x\textsubscript{0}, t ∈ X ∪ I, x = x\textsubscript{0} \cup \{t\};
- \(∃y. q\textsubscript{0}(x, y); q\textsubscript{1}(x) \lor q\textsubscript{2}(x); q\textsubscript{1}(x) \lor q\textsubscript{2}(x); q\textsubscript{1}(x) \setminus q\textsubscript{2}(x). \)
The semantics of BALG₁-queries is defined as follows.

**Definition 21.** The bag answers $q^I$ to a BALG₁-query $q(x)$ over a bag interpretation $I = (Δ^I, □^I)$ 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 $λ$ such that $λ(x) = a^I$ and $λ(a) = a^I$ for all $a ∈ I$:

- $S^I(λ(t)))$, if $q(x) = S(t)$;
- $q^I_1(λ(x_1)) × q^I_2(λ(x_2))$, if $q(x) = q_1(x_1) \land q_2(x_2)$;
- $q^I_0(λ(x_0))$, if $q(x) = q_0(x_0) \land (x = t)$ and $λ(x) = λ(t)$;
- 0, if $q(x) = q_0(x_0) \land (x = t)$ and $λ(x) = 0$;
- $\sum_{λ'} \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \
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 $q_{z}(x)$ be the CQ $\exists y'. \phi_{z}(x, y')$ such that $\phi_{z}(x, y')$ is obtained from $\phi(x, y)$ by replacing $\phi_{z}$, for each ma-connected $z' \subseteq z$, with $\phi_{z'}$ and $y'$ is the subset of $y$ remaining in $\phi_{z}$. In other words, $q_{z}$ contains, for each $z'$, just one atom $\phi_{z'}$ and equalities identifying $t_{z'}$ instead of conjunction $\phi_{z'}$ 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 $q_{z}$ for each $z$.

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

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

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

- each $A(t)$ with $\bigvee_{t \in \text{C} \subseteq A} \zeta_{C}(t)$;
- each $P(t, z)$ with $\bigvee_{t \in \text{C} \subseteq \text{DP}} \zeta_{C}(t) \setminus \zeta_{\text{DP}}(t)$;
- each $P(z, t)$ with $\bigvee_{t \in \text{C} \subseteq \text{DP'}} \zeta_{C}(t) \setminus \zeta_{\text{DP'}}(t)$.

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

Finally, the rewriting of $\phi(q)$ over $T$ is the BALG$_{1}$-query $\hat{q}(x) = \bigvee_{z \subseteq y \text{realisable by } T} \hat{q}_{z}(x)$.

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 $\hat{q}'(x) = \hat{q}_{0}(x) \lor \hat{q}_{y}(x)$, where $\hat{q}_{0}(x)$ and $\hat{q}_{y}(x)$ are

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

is a rewriting of $q'$ over $T_{r}$, since $\emptyset$ and $y$ are realisable.

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

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

Corollary 30. The class of rooted CQs is rewriteable to BALG$_{1}$-queries for DL-Lite$_{\text{bag}}$-core.

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.

Theorem 31. BAGCERT$_{\text{bag}}$-rooted CQs, DL-Lite$_{\text{bag}}$-core is NP-complete and in logspace in data complexity.

However, the next theorem implies that rooted queries are not BALG$_{1}$-rewritable for unrestricted DL-Lite$_{\text{bag}}$ TBoxes.

Theorem 32. BAGCERT$_{\text{bag}}$-rooted CQs, DL-Lite$_{\text{bag}}$ is conP-hard in data complexity.

6 Related work

Query answering under bag semantics has received significant attention in the database literature [Libkin and Wong, 1994; Grumbach et al., 1996; Grumbach and Milo, 1996; Libkin and Wong, 1997]. These works study the relative expressive power of bag algebra primitives, the relationship with set-based algebras, and establish the data complexity of query answering. Such problems have also been recently studied in the setting of Semantic Web and SPARQL 1.1 in [Kaminski et al., 2016; Angles and Gutierrez, 2016].

Bag semantics in the context of Description Logics has been studied in [Jiang, 2010], where the author proposes a bag semantics for ALC and provides a tableau algorithm. In contrast to our work, their results are restricted to ontology satisfiability and do not encompass CQ answering.

CQ answering under bag semantics is closely related to answering Count aggregate queries. The semantics of aggregate queries for database settings with incomplete information, such as inconsistent databases and data exchange, have been studied in [Arens et al., 2003; Libkin, 2006; Afrati and Kolaitis, 2008]. As pointed out in [Kostylev and Reutter, 2015], these techniques are not directly applicable to ontologies. The practical solution in [Calvanese et al., 2008] is to give epistemic semantics to aggregate queries, where the query is evaluated over ABox facts entailed by the ontology; thus, the anonymous part of the ontology models is essentially ignored, and the semantics easily leads to counter-intuitive answers. To remedy these issues, [Kostylev and Reutter, 2015] propose a certain answer semantics for Count aggregate queries over ontologies and prove tight complexity bounds for DL-Lite$_{\text{R}}$ and DL-Lite$_{\text{core}}$. Similarly to our work, their semantics is open-world and considers all models of the ontology for query evaluation, which leads to more intuitive answers. The main difference resides in the definition of the ontology language, where they consider set ABoxes and adopt conventional set-based semantics for TBox axioms. Although DL-Lite$_{\text{bag}}$ is closely related to the logic in [Kostylev and Reutter, 2015], the two settings do not coincide even for set ABoxes. For example, if $A$ comprises only assertions $R(a, b)$ and $R(a, c)$ and $T$ comprises axiom $\exists R B$, then the query over $\langle T, A \rangle$ that counts the number of individuals $a$ in concept $B$ returns 1 in the setting of [Kostylev and Reutter, 2015], while the corresponding DL-Lite$_{\text{bag}}$ query returns 2.

7 Conclusion and Future Work

We have studied OBDA under bag semantics and identified a general class of rewritable queries over DL-Lite$_{\text{bag}}$-core ontologies. As our framework covers already the class of Count aggregate queries, in future work we plan to extend it to capture further aggregate functions and more expressive ontologies.
References


