Bag Semantics of DL-Lite with Functionality Axioms

Gianluca Cima[1(✉)], Charalampos Nikolaou[2,3(✉)], Egor V. Kostylev[2(✉)], Mark Kaminski[2(✉)], Bernardo Cuenca Grau[2(✉)], and Ian Horrocks[2(✉)]

[1] Sapienza Università di Roma, Rome, Italy
cima@diag.uniroma1.it

[2] Department of Computer Science, University of Oxford, Oxford, UK
{egor.kostylev,mark.kaminski,bernardo.cuenca.grau,ian.horrocks}@cs.ox.ac.uk

[3] Infor, Farnborough, UK
charalampos.nikolaou@infor.com

Abstract. Ontology-based data access (OBDA) is a popular approach for integrating and querying multiple data sources by means of an ontology, which is usually expressed in a description logic (DL) of DL-Lite family. The conventional semantics of OBDA and DLs is set-based—that is, duplicates are disregarded. This disagrees with the standard database bag (multiset) semantics, which is especially important for the correct evaluation of aggregate queries. In this article, we study two variants of bag semantics for query answering over $\mathit{DL\text{-}Lite}_{\mathcal{F}}$, extending basic $\mathit{DL\text{-}Lite}_{\mathrm{core}}$ with functional roles. For our first semantics, which follows the semantics of primary keys in SQL, conjunctive query (CQ) answering is coNP-hard in data complexity in general, but it is in TC^0 for the restricted class of rooted CQs; such CQs are also rewritable to the bag relational algebra. For our second semantics, the results are the same except that TC^0 membership and rewritability hold only for the restricted class of ontologies identified by a new notion of functional weak acyclicity.

1 Introduction

Ontology-based data access (OBDA) is an increasingly popular approach for integrating multiple relational data sources under a global schema [7,24,32]. In OBDA, an ontology provides a unifying conceptual model for the data sources, which is linked to each source by mappings assigning 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 usually the lightweight description logic $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ [8], which underpins OWL 2 QL [28]. $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ was designed to ensure that conjunctive queries (CQs) against the ontology are *first-order rewritable*—that is, they can be reformulated as relational database queries over the sources [8].

C. Ghidini et al. (Eds.): ISWC 2019, LNCS 11778, pp. 128–144, 2019.
https://doi.org/10.1007/978-3-030-30793-6_8

There is, however, an important semantic mismatch between standard database query languages, such as SQL, and OBDA: the former commit to a bag (multiset) semantics, where tuples are allowed to occur multiple times in query answers, whereas the latter is usually set-based, where multiplicities are disregarded. This semantic difference becomes apparent when evaluating queries with aggregation, where the multiplicities of tuples are important [1]. Motivated by the need to support database-style aggregate queries in OBDA systems and inspired by the work of Kostylev and Reutter [23] on the support of aggregates queries in $\mathit{DL\text{-}Lite}_{\mathcal{R}}$, Nikolaou et al. [30,31] proposed a bag semantics for $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ and OBDA, where duplicates in the views defined by the mappings are retained. The most common reasoning tasks of ontology satisfiability and query answering in this new language, called $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{R}}$, generalise the counterpart problems defined under the traditional set semantics. This generalisation does not come for free though as it raises the data complexity of query answering from AC^0 to coNP-hard, and this holds already for the core fragment $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathit{core}}$ of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{R}}$. To regain tractability, Nikolaou et al. [30,31] studied restrictions on CQs and showed that query answering for the class of so-called *rooted CQs* [6] becomes again tractable in data complexity. This result was obtained by showing that rooted CQs are rewritable to BCALC, a logical counterpart of the relational algebra BALG^1 for bags [15,26] whose evaluation problem is known to be in TC^0 in data complexity [25].

In this paper, building on the work of Nikolaou et al. [30,31], we consider the logic $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$—that is, the extension of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathit{core}}$ with functionality axioms. Such axioms comprise a desirable feature in description logics and OBDA since they are able to deliver various modelling scenarios encountered in information systems [9,10,27,33], such as key and identification constraints. We propose two alternative semantics for $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$, both of which generalise the standard set-based semantics, and which differ from each other in the way they handle functionality axioms. Our first semantics, called *SQL semantics*, interprets functionality axioms following the semantics of primary keys in SQL—that is, the interpretation of a functional role is required to be a set satisfying the key constraint in the sense that for each first component in the interpretation of a functional role there exists exactly one second component, and, moreover, the multiplicity of this relation between the components is exactly one. By contrast, our second semantics, called *multiplicity-respectful (MR) semantics*, retains the key constraint requirement but allows for several copies of the same pair in the interpretation.

Our results are summarised below. First, we study how the two semantics relate to the set-based semantics of $\mathit{DL\text{-}Lite}_{\mathcal{F}}$ and to each other in terms of the standard reasoning tasks of satisfiability checking and query answering. On the one hand, we show that under the MR semantics both problems generalise the corresponding ones under set semantics. On the other hand, under the SQL semantics the notion of satisfiability becomes stronger than under set semantics, while query answering for satisfiable ontologies again generalises set semantics. Second, we investigate whether the class of rooted CQs is rewritable to BCALC under each of the semantics. For the SQL semantics, we obtain a positive answer, which implies that query answering is feasible in TC^0 in data complexity. For the MR semantics, however, we obtain LogSpace-hardness of query answering even for the simple class of instance queries, which prevents rewritability to BCALC (under the usual complexity-theoretic assumptions). To address this, we identify a class of TBoxes, called *functionally weakly acyclic*, for which rooted CQs become rewritable to BCALC, thus regaining feasibility of query answering.

The rest of the paper is organised as follows. Section 2 introduces the relevant background. Section 3 defines the SQL and MR semantics as extensions of the bag semantics proposed in [30,31] accounting for functionality axioms, and relates the new semantics to the set semantics and to each other. Section 4 studies the query answering problem, establishing the rewritability and complexity results. Last, Sect. 5 discusses related work and Sect. 6 concludes the paper.

2 Preliminaries

We start by defining *DL-Lite*$_\mathcal{F}$ ontologies as well as the notions of query answering and rewriting over such ontologies, all over the usual set semantics [4,8], after which we summarise the bag semantics of queries in databases [15,26,31].

Syntax of *DL-Lite*$_\mathcal{F}$. We fix a vocabulary consisting of countably infinite and pairwise disjoint sets of *individuals* $\mathbf{I}$ (i.e., constants), *atomic concepts* $\mathbf{C}$ (i.e., unary predicates) and *atomic roles* $\mathbf{R}$ (i.e., binary predicates). A *role* is an atomic role $P \in \mathbf{R}$ or its *inverse* P^-. A *concept* is an atomic concept in $\mathbf{C}$ or an expression $\exists R$ with R a role. Expressions $C_1 \sqsubseteq C_2$ and $\mathsf{Disj}(C_1, C_2)$ with C_1, C_2 concepts are *inclusion* and *disjointness axioms*, respectively. An expression $(\mathsf{funct}\ R)$ with R a role is a *functionality axiom*. A *DL-Lite*$_\mathcal{F}$ *TBox* is a finite set of inclusion, disjointness, and functionality axioms. A *concept assertion* is $A(a)$ with $a \in \mathbf{I}$ and $A \in \mathbf{C}$, and a *role assertion* is $P(a, b)$ with $a, b \in \mathbf{I}$ and $P \in \mathbf{R}$. A (*set*) *ABox* is a finite set of concept and role assertions. A *DL-Lite*$_\mathcal{F}$ *ontology* is a pair $(\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ a *DL-Lite*$_\mathcal{F}$ TBox and $\mathcal{A}$ an ABox. A *DL-Lite*$_{\text{core}}$ *ontology* is the same except that functionality axioms are disallowed.

Semantics of *DL-Lite*$_\mathcal{F}$. A (*set*) *interpretation* $\mathcal{I}$ is a pair $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the *domain* $\Delta^\mathcal{I}$ is a non-empty set, and the *interpretation function* $\cdot^\mathcal{I}$ maps each $a \in \mathbf{I}$ to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$ such that $a^\mathcal{I} \neq b^\mathcal{I}$ for all distinct $a, b \in \mathbf{I}$ (i.e., as usual for *DL-Lite* we adopt the UNA—that is, the unique name assumption), each $A \in \mathbf{C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and each $P \in \mathbf{R}$ to $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. Interpretation function $\cdot^\mathcal{I}$ extends to non-atomic concepts and roles as follows, for each $P \in \mathbf{R}$ and each role R:

$$(P^-)^\mathcal{I} = \{(u, u') \mid (u', u) \in P^\mathcal{I}\},\ (\exists R)^\mathcal{I} = \{u \in \Delta^\mathcal{I} \mid \exists u' \in \Delta^\mathcal{I} :\ (u, u') \in R^\mathcal{I}\}.$$

An interpretation $\mathcal{I}$ *satisfies* a *DL-Lite*$_\mathcal{F}$ TBox $\mathcal{T}$ if $C_1^\mathcal{I} \subseteq C_2^\mathcal{I}$ for each inclusion axiom $C_1 \sqsubseteq C_2$ in $\mathcal{T}$, $C_1^\mathcal{I} \cap C_2^\mathcal{I} = \emptyset$ for each disjointness axiom $\mathsf{Disj}(C_1, C_2)$ in $\mathcal{T}$, and $v_1 = v_2$ for each $(u, v_1), (u, v_2)$ in $R^\mathcal{I}$ with functionality axiom $(\mathsf{funct}\ R)$ in $\mathcal{T}$. Interpretation $\mathcal{I}$ *satisfies* an ABox $\mathcal{A}$ if $a^\mathcal{I} \in A^\mathcal{I}$ for all $A(a) \in \mathcal{A}$ and $(a^\mathcal{I}, b^\mathcal{I}) \in P^\mathcal{I}$

for all $P(a,b) \in \mathcal{A}$. An interpretation $\mathcal{I}$ is a *model* of an ontology $(\mathcal{T},\mathcal{A})$ if it satisfies both $\mathcal{T}$ and $\mathcal{A}$. An ontology is *satisfiable* if it has a model. Checking satisfiability of a *DL-Lite*$_\mathcal{F}$ ontology is NLogSpace-complete in general and in AC^0 if the TBox is fixed [4,8] (note however that the latter problem becomes P-complete if the UNA is dropped).

Queries over *DL-Lite*$_\mathcal{F}$. A *conjunctive query* (*CQ*) $q(\mathbf{x})$ with *answer* variables $\mathbf{x}$ is a formula $\exists \mathbf{y}.\, \phi(\mathbf{x},\mathbf{y})$, where $\mathbf{x}$, $\mathbf{y}$ are (possibly empty) repetition-free disjoint tuples of variables from a set $\mathbf{X}$ disjoint from $\mathbf{I}$, $\mathbf{C}$ and $\mathbf{R}$, and $\phi(\mathbf{x},\mathbf{y})$ is a conjunction of atoms of the form $A(t)$, $P(t_1,t_2)$ or $(z = t)$, where $A \in \mathbf{C}$, $P \in \mathbf{R}$, $z \in \mathbf{x} \cup \mathbf{y}$, and $t, t_1, t_2 \in \mathbf{x} \cup \mathbf{y} \cup \mathbf{I}$. If $\mathbf{x}$ is inessential, then we write q instead of $q(\mathbf{x})$. The equality atoms $(z = t)$ in $\phi(\mathbf{x},\mathbf{y})$ yield an equivalence relation $\sim$ on terms $\mathbf{x} \cup \mathbf{y} \cup \mathbf{I}$, and we write $\tilde{t}$ for the equivalence class of a term t. The *Gaifman graph* of $q(\mathbf{x})$ has a node $\tilde{t}$ for each $t \in \mathbf{x} \cup \mathbf{y} \cup \mathbf{I}$ in ϕ, and an edge $\{\tilde{t}_1, \tilde{t}_2\}$ for each atom in ϕ over t_1 and t_2. We assume that all CQs are *safe*—that is, for each $z \in \mathbf{x} \cup \mathbf{y}$, $\tilde{z}$ contains a term mentioned in an atom of $\phi(\mathbf{x},\mathbf{y})$ that is not equality. A CQ $q(\mathbf{x})$ is *rooted* if each connected component of its Gaifman graph has a node with a term in $\mathbf{x} \cup \mathbf{I}$ [6]. A *union* of CQs (*UCQ*) is a disjunction of CQs with the same answer variables. The *certain answers* $q^\mathcal{K}$ to a (U)CQ $q(\mathbf{x})$ over a *DL-Lite*$_\mathcal{F}$ ontology $\mathcal{K}$ are the set of all tuples $\mathbf{a}$ of individuals such that $q(\mathbf{a})$ holds in every model of $\mathcal{K}$. Checking whether a tuple of individuals is in the certain answers to a (U)CQ over a *DL-Lite*$_\mathcal{F}$ ontology is an NP-complete problem with AC^0 data complexity (i.e., when the query and TBox are fixed) [4,8]. The latter follows from the *rewritability* of the class of UCQs to itself over *DL-Lite*$_\mathcal{F}$—that is, from the fact that for each UCQ q and *DL-Lite*$_\mathcal{F}$ TBox $\mathcal{T}$ there is a UCQ q_1 such that $q^{(\mathcal{T},\mathcal{A})} = q_1^{(\emptyset,\mathcal{A})}$ for each ABox $\mathcal{A}$ [8].

Bags. A *bag* over a set M is a function $\Omega : M \to \mathbb{N}_0^\infty$, where $\mathbb{N}_0^\infty$ is the set $\mathbb{N}_0$ of non-negative integers extended with the (positive) infinity ∞. The value $\Omega(c)$ is the *multiplicity* of element c in Ω. A bag Ω 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 $\emptyset(c) = 0$ for each $c \in M$. A bag Ω_1 over M is a *subbag* of a bag Ω_2 over M, in symbols $\Omega_1 \subseteq \Omega_2$, if $\Omega_1(c) \le \Omega_2(c)$ for each $c \in M$. Often we will use an alternative syntax for bags: for instance, we will write $\{\!|\, c : 5, d : 3 \,|\!\}$ for the bag that assigns 5 to c, 3 to d, and 0 to all other elements. We use the following common operators on bags [15,26]: the *intersection* $\cap$, *maximal union* $\cup$, *arithmetic union* $\uplus$, and *difference* $-$ are the binary operators defined, for bags Ω_1 and Ω_2 over a set M, and for every $c \in M$, as follows:

$$(\Omega_1 \cap \Omega_2)(c) = \min\{\Omega_1(c), \Omega_2(c)\}, \quad (\Omega_1 \cup \Omega_2)(c) = \max\{\Omega_1(c), \Omega_2(c)\},$$
$$(\Omega_1 \uplus \Omega_2)(c) = \Omega_1(c) + \Omega_2(c), \quad (\Omega_1 - \Omega_2)(c) = \max\{0, \Omega_1(c) - \Omega_2(c)\}.$$

Note that bag difference is well-defined only if $\Omega_2(c)$ is a finite number for each $c \in M$. The unary *duplicate elimination* operator ε is defined for a bag Ω over M and for each $c \in M$ as $(\varepsilon(\Omega))(c) = 1$ if $\Omega(c) > 0$ and $(\varepsilon(\Omega))(c) = 0$ otherwise.

Queries over Bags. Following [31], a BCALC *query* $\Phi(\mathbf{x})$ with (a tuple of) *answer* variables $\mathbf{x}$ is any of the following, for Ψ, Ψ_1, and Ψ_2 BCALC queries:

- $S(\mathbf{t})$, where $S \in \mathbf{C} \cup \mathbf{R}$ and $\mathbf{t}$ is a tuple over $\mathbf{x} \cup \mathbf{I}$ mentioning all $\mathbf{x}$;
- $\Psi_1(\mathbf{x}_1) \wedge \Psi_2(\mathbf{x}_2)$, where $\mathbf{x} = \mathbf{x}_1 \cup \mathbf{x}_2$;
- $\Psi(\mathbf{x}_0) \wedge (x = t)$, where $x \in \mathbf{x}_0$, $t \in \mathbf{X} \cup \mathbf{I}$, and $\mathbf{x} = \mathbf{x}_0 \cup (\{t\} \setminus \mathbf{I})$;
- $\exists \mathbf{y}.\, \Psi(\mathbf{x}, \mathbf{y})$, where $\mathbf{y}$ is a tuple of distinct variables from $\mathbf{X}$ that are not in $\mathbf{x}$;
- $\Psi_1(\mathbf{x})\, \mathsf{op}\, \Psi_2(\mathbf{x})$, where $\mathsf{op} \in \{\vee, \veebar, \setminus\}$; or
- $\delta\, \Psi(\mathbf{x})$.

In particular, all UCQs are syntactically BCALC queries. BCALC queries are evaluated over bag database instances, which are, in the context of this paper, *bag ABoxes*—that is, finite bags over the set of concept and role assertions. The *bag answers* $\Phi^{\mathcal{A}}$ to a BCALC query $\Phi(\mathbf{x})$ over a bag ABox $\mathcal{A}$ is the finite bag over $\mathbf{I}^{|\mathbf{x}|}$ defined inductively as follows, for every tuple $\mathbf{a}$ over $\mathbf{I}$ with $|\mathbf{a}| = |\mathbf{x}|$, where $\nu : \mathbf{x} \cup \mathbf{I} \to \mathbf{I}$ is the function such that $\nu(\mathbf{x}) = \mathbf{a}$ and $\nu(a) = a$ for all $a \in \mathbf{I}$:

- $\Phi^{\mathcal{A}}(\mathbf{a}) = \mathcal{A}(S(\nu(\mathbf{t})))$, if $\Phi(\mathbf{x}) = S(\mathbf{t})$;
- $\Phi^{\mathcal{A}}(\mathbf{a}) = \Psi_1^{\mathcal{A}}(\nu(\mathbf{x}_1)) \times \Psi_2^{\mathcal{A}}(\nu(\mathbf{x}_2))$, if $\Phi(\mathbf{x}) = \Psi_1(\mathbf{x}_1) \wedge \Psi_2(\mathbf{x}_2)$;
- $\Phi^{\mathcal{A}}(\mathbf{a}) = \Psi^{\mathcal{A}}(\nu(\mathbf{x}_0))$, if $\Phi(\mathbf{x}) = \Psi(\mathbf{x}_0) \wedge (x = t)$ and $\nu(x) = \nu(t)$;
- $\Phi^{\mathcal{A}}(\mathbf{a}) = 0$, if $\Phi(\mathbf{x}) = \Psi(\mathbf{x}_0) \wedge (x = t)$ and $\nu(x) \neq \nu(t)$;
- $\Phi^{\mathcal{A}}(\mathbf{a}) = \sum_{\nu' : \mathbf{y} \to \mathbf{I}} \Psi^{\mathcal{A}}(\mathbf{a}, \nu'(\mathbf{y}))$, if $\Phi(\mathbf{x}) = \exists \mathbf{y}.\, \Psi(\mathbf{x}, \mathbf{y})$;
- $\Phi^{\mathcal{A}}(\mathbf{a}) = (\Psi_1^{\mathcal{A}}\, \mathsf{op}\, \Psi_2^{\mathcal{A}})(\mathbf{a})$, if $\Phi(\mathbf{x}) = \Psi_1(\mathbf{x})\, \mathsf{op}'\, \Psi_2(\mathbf{x})$, where op is $\cup$, $\uplus$, or $-$, and op' is $\vee$, $\veebar$, or $\setminus$, respectively;
- $\Phi^{\mathcal{A}}(\mathbf{a}) = \big(\varepsilon(\Psi^{\mathcal{A}})\big)(\mathbf{a})$, if $\Phi(\mathbf{x}) = \delta\, \Psi(\mathbf{x})$.

As shown in [31], BCALC is a logical counterpart of the bag relational algebra BALG^1 [15], with the same expressive power. Evaluation of a fixed BALG^1 (and hence BCALC) query is in TC^0 [25] (i.e., between AC^0 and LogSpace).

3 *DL-Lite$_{\mathcal{F}}$* Under Bag Semantics

In this section we introduce the bag version *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ of the ontology language *DL-Lite*$_{\mathcal{F}}$ by proposing two semantics and then study their properties and relationships. Both semantics extend the bag semantics of *DL-Lite*$_{\mathsf{core}}$ proposed by Nikolaou et al. [30,31] but differ in their interpretation of functionality axioms.

3.1 Syntax and Semantics of *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$

Syntactically, *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ is the same as *DL-Lite*$_{\mathcal{F}}$ except that assertions in ABoxes may have arbitrary finite multiplicities—that is, bag ABoxes are considered instead of set ABoxes. Thus, at the syntax level *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ is a conservative extension of *DL-Lite*$_{\mathcal{F}}$ since each set ABox can be seen as a bag ABox with assertion multiplicities 0 and 1.

Definition 1. *A DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ ontology *is a pair* $(\mathcal{T}, \mathcal{A})$ *of a DL-Lite*$_{\mathcal{F}}$ *TBox* $\mathcal{T}$ *and a bag ABox* $\mathcal{A}$. *A DL-Lite*$^{\mathsf{b}}_{\mathsf{core}}$ ontology *is the same except that* $\mathcal{T}$ *is DL-Lite*$_{\mathsf{core}}$.

The semantics of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontologies is based on bag interpretations, which are the same as set interpretations except that concepts and roles are interpreted as bags rather than sets. The extension of the interpretation function to non-atomic concepts and roles is defined in a way that respects the multiplicities: for example, the concept $\exists P$ for an atomic role P is interpreted by a bag interpretation $\mathcal{I}$ as the bag projection of $P^{\mathcal{I}}$ to its first component, where each occurrence of a pair (u, v) in $P^{\mathcal{I}}$ contributes separately to the multiplicity of u in $(\exists P)^{\mathcal{I}}$.

Definition 2. *A* bag interpretation $\mathcal{I}$ *is a pair* $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *where the* domain $\Delta^{\mathcal{I}}$ *is a non-empty set, and the* interpretation function $\cdot^{\mathcal{I}}$ *maps each individual* $a \in \mathbf{I}$ *to an element* $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ *such that* $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ *for all distinct* $a, b \in \mathbf{I}$*, each atomic concept* $A \in \mathbf{C}$ *to a bag* $A^{\mathcal{I}}$ *over* $\Delta^{\mathcal{I}}$*, and each atomic role* $P \in \mathbf{R}$ *to a bag* $P^{\mathcal{I}}$ *over* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$*. Interpretation function* $\cdot^{\mathcal{I}}$ *extends to non-atomic concepts and roles as follows, for all* $P \in \mathbf{R}$*,* R *a role, and* $u, u' \in \Delta^{\mathcal{I}}$*:*

$$(P^{-})^{\mathcal{I}}(u, u') = P^{\mathcal{I}}(u', u) \qquad \text{and} \qquad (\exists R)^{\mathcal{I}}(u) = \sum\nolimits_{u' \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(u, u').$$

Note that, same as in the set case, we adopt the UNA by requiring different individuals be interpreted by different domain elements.

We are now ready to present our two semantics of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$. Both semantics extend the semantics of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{core}$ considered in [31], but handle the functional axioms differently. Our first semantics, called SQL, follows the semantics of primary keys in SQL: if R is a functional role then for every domain element u of a model there exists at most one element u' related to u by R; moreover, the multiplicity of the tuple (u, u') in R cannot be more than one. Our second semantics, called MR (i.e., multiplicity-respectful), allows more freedom for functional roles: same as before, only one u' may be related to u by a functional role R, but the multiplicity of (u, u') may be arbitrary.

Definition 3. *A bag interpretation* $\mathcal{I}$ satisfies *an inclusion axiom* $C_1 \sqsubseteq C_2$ *if* $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$*. It* satisfies *a disjointness axiom* $\mathsf{Disj}(C_1, C_2)$ *if* $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} = \emptyset$*. It* satisfies *a functionality axiom* $(\mathsf{funct}\ R)$ under SQL semantics *(or* SQL-satisfies*, for short) if* $u' = u''$ *and* $R^{\mathcal{I}}(u, u') = R^{\mathcal{I}}(u, u'') = 1$ *for every* u*,* u'*, and* u'' *in* $\Delta^{\mathcal{I}}$ *such that* $R^{\mathcal{I}}(u, u') > 0$ *and* $R^{\mathcal{I}}(u, u'') > 0$*; it satisfies* $(\mathsf{funct}\ R)$ under MR semantics *(or* MR-satisfies*) if the same holds except that the requirement* $R^{\mathcal{I}}(u, u') = R^{\mathcal{I}}(u, u'') = 1$ *is not imposed.*

For X *being* SQL *or* MR*, a bag interpretation* $\mathcal{I}$ X-satisfies *a* $\mathit{DL\text{-}Lite}_{\mathcal{F}}$ *TBox* $\mathcal{T}$*, written* $\mathcal{I} \models_X \mathcal{T}$*, if it satisfies every inclusion and disjointness axiom in* $\mathcal{T}$ *and* X*-satisfies every functionality axiom in* $\mathcal{T}$*. A bag interpretation* $\mathcal{I}$ satisfies *a bag ABox* $\mathcal{A}$*, written* $\mathcal{I} \models \mathcal{A}$*, if* $\mathcal{A}(A(a)) \leq A^{\mathcal{I}}(a^{\mathcal{I}})$ *and* $\mathcal{A}(P(a, b)) \leq P^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$ *for each concept assertion* $A(a)$ *and role assertion* $P(a, b)$*, respectively. A bag interpretation* $\mathcal{I}$ *is an* X-model *of a* $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ *ontology* $(\mathcal{T}, \mathcal{A})$*, written* $\mathcal{I} \models_X (\mathcal{T}, \mathcal{A})$*, if* $\mathcal{I} \models_X \mathcal{T}$ *and* $\mathcal{I} \models \mathcal{A}$*. A* $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ *ontology is* X-satisfiable *if it has an* X*-model.*

Since MR-satisfaction is a relaxation of SQL-satisfaction, every SQL-model of a $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontology is also an MR-model of this ontology. However, as illustrated by the following example, the opposite does not hold.

Example 1. Consider an online store that employs atomic concept $\mathsf{Customer}$ and atomic roles $\mathsf{hasItem}$, $\mathsf{placedBy}$ for recording the items ordered by customers in a purchase. A sample $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontology recording customers' orders is $\mathcal{K}_{\mathrm{ex}} = (\mathcal{T}_{ex}, \mathcal{A}_{ex})$ with

$$\mathcal{T}_{ex} = \{\exists \mathsf{hasItem} \sqsubseteq \exists \mathsf{placedBy},\ \exists \mathsf{placedBy}^- \sqsubseteq \mathsf{Customer},\ (\mathsf{funct}\ \mathsf{placedBy})\} \text{ and}$$
$$\mathcal{A}_{ex} = \{\!|\ \mathsf{hasItem}(o, i_1) : 1,\ \mathsf{hasItem}(o, i_2) : 1,\ \mathsf{placedBy}(o, c) : 1,\ \mathsf{Customer}(c) : 1\ |\!\}.$$

Let $\mathcal{I}_{ex}$ be the bag interpretation that interprets all individuals by themselves, and the atomic roles and concepts as follows: $\mathsf{Customer}^{\mathcal{I}_{ex}} = \{\!|\ c : 2\ |\!\}$, $\mathsf{hasItem}^{\mathcal{I}_{ex}} = \{\!|\ (o, i_1) : 1, (o, i_2) : 1\ |\!\}$, and $\mathsf{placedBy}^{\mathcal{I}_{ex}} = \{\!|\ (o, c) : 2\ |\!\}$. It is immediate that $\mathcal{I}_{ex}$ is an MR-model of $\mathcal{K}_{\mathrm{ex}}$ but not a SQL-model. ◁

To conclude this section, we note that each semantics has its advantages and drawbacks. Indeed, on the one hand, SQL semantics is compatible with primary keys in SQL, so a large fragment of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ under this semantics can be easily simulated by a SQL engine. On the other hand, one can show that entailment of axioms under set and bag semantics coincides only for the case of MR models; this means that the adoption of MR semantics does not affect the standard TBox reasoning services implemented in ontology development tools. So neither of the two semantics is clearly preferable to the other.

3.2 Queries over $\boldsymbol{\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}}$

We next define the answers $q^{\mathcal{I}}$ to a CQ $q(\mathbf{x})$ over a bag interpretation $\mathcal{I}$ as the bag of tuples of individuals such that each valid embedding λ of the atoms in q into $\mathcal{I}$ contributes separately to the multiplicity of the tuple $\lambda(\mathbf{x})$ in $q^{\mathcal{I}}$, and where the contribution of each specific λ is the product of the multiplicities of the images of the query atoms under λ in $\mathcal{I}$. This may be seen as usual CQ answering under bag semantics over relational databases when the interpretation is seen as a bag database instance [12]. In fact, when q is evaluated over this bag database instance as a BCALC query (see Sect. 2), it produces exactly $q^{\mathcal{I}}$.

Definition 4. *Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a CQ and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a bag interpretation. The* bag answers *$q^{\mathcal{I}}$ to q over $\mathcal{I}$ are the bag over tuples of individuals from $\mathbf{I}$ of size $|\mathbf{x}|$ such that, for every such tuple $\mathbf{a}$,*

$$q^{\mathcal{I}}(\mathbf{a}) \;\; = \;\; \sum\nolimits_{\lambda \in \Lambda} \prod\nolimits_{S(\mathbf{t}) \text{ in } \phi(\mathbf{x},\mathbf{y})} S^{\mathcal{I}}(\lambda(\mathbf{t})),$$

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

Note that conjunction $\phi(\mathbf{x}, \mathbf{y})$ in a CQ 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 [12,18], and thus in the definition of $q^{\mathcal{I}}(\mathbf{a})$ each occurrence of a query atom $S(\mathbf{t})$ is treated separately in the product.

The following definition of certain answers, which captures open-world query answering, is a natural extension of certain answers for $\mathit{DL\text{-}Lite}_{\mathcal{F}}$ to bags. For $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{core}$, this definition coincides with the one in [31] for both semantics.

Definition 5. *For X being* SQL *or* MR*, the X*-bag certain answers $q_X^{\mathcal{K}}$ *to a CQ q over a DL-Lite$^{\mathsf{b}}_{\mathcal{F}}$ ontology $\mathcal{K}$ are the bag $\bigcap_{\mathcal{I}\models_X \mathcal{K}} q^{\mathcal{I}}$.*

Note that in this definition the intersection is the bag intersection, and we assume that the intersection of zero bags (which is relevant when $\mathcal{K}$ is not X-satisfiable) assigns ∞ to all tuples over $\mathbf{I}$.

The (data complexity version of the) decision problem corresponding to computing the X-bag certain answers to a CQ q over an ontology with a *DL-Lite*$_{\mathcal{F}}$ TBox $\mathcal{T}$, for X begin SQL or MR, is defined as follows, assuming that all numbers in the input are represented in unary.

BagCert$_X[q, \mathcal{T}]$

Input:	ABox $\mathcal{A}$, tuple $\mathbf{a}$ of individuals from $\mathbf{I}$, and $k \in \mathbb{N}_0^{\infty}$.
Question:	Is $q_X^{(\mathcal{T},\mathcal{A})}(\mathbf{a}) \geq k$?

The idea of bag certain answers is illustrated by the following example.

Example 2. Recall ontology $(\mathcal{T}_{ex}, \mathcal{A}_{ex})$ and interpretation $\mathcal{I}_{ex}$ specified in Example 1, and let $q(x) = \exists y.\, \mathsf{placedBy}(x, y) \wedge \mathsf{Customer}(y)$ be the rooted CQ requesting orders placed by customers. The bag answers $q^{\mathcal{I}_{ex}}$ to q over interpretation $\mathcal{I}_{ex}$ is the bag $\{\!| o : 4 |\!\}$. Moreover, it is not hard to see that the MR-bag certain answers to q over $(\mathcal{T}_{ex}, \mathcal{A}_{ex})$ coincide with bag $q^{\mathcal{I}_{ex}}$, and that $q_{\mathrm{SQL}}^{(\mathcal{T}_{ex},\mathcal{A}_{ex})}(a) = \infty$ for every $a \in \mathbf{I}$ since $(\mathcal{T}_{ex}, \mathcal{A}_{ex})$ does not have any SQL-model. ◁

Besides the complexity of query answering, an important related property of any description logic is query rewritability: since TBoxes are much more stable than ABoxes in practice, it is desirable to be able to rewrite a query and a TBox into another query so that the answers to the original query over each satisfiable ontology with this TBox are the same as the answers to the rewriting over the ABox alone. The rewriting may be in a richer query language than the language of the original query, provided we have an efficient query engine for the target language; it is important, however, that the rewriting does not depend on the ABox. As mentioned above, rewritings of (U)CQs to UCQs are usually considered in the set setting. In our bag setting, the source language is CQs and the target language is BCALC, which can be easily translated to SQL.

Definition 6. *For X being* SQL *or* MR*, a* BCALC *query Φ is an X*-rewriting *of a CQ q with respect to a DL-Lite$_{\mathcal{F}}$ TBox $\mathcal{T}$ if $q_X^{(\mathcal{T},\mathcal{A})} = \Phi^{\mathcal{A}}$ for every bag ABox $\mathcal{A}$ with $(\mathcal{T}, \mathcal{A})$ X-satisfiable. A class $\mathcal{Q}$ of CQs is X*-rewritable *to a class $\mathcal{Q}'$ of* BCALC *queries over a sublanguage $\mathcal{L}$ of DL-Lite$_{\mathcal{F}}$ if, for every CQ in $\mathcal{Q}$ and TBox in $\mathcal{L}$, there is an X-rewriting of the CQ with respect to the TBox in $\mathcal{Q}'$.*

Since evaluation of fixed BCALC queries is in TC^0 [25], rewritability to BCALC implies TC^0 data complexity of query answering provided rewritings are effectively constructible. BagCert$_X[q, \mathcal{T}]$ is coNP-hard even for *DL-Lite*$^{\mathsf{b}}_{\mathrm{core}}$ ontologies (for both X) [31], which precludes efficient query answering and (constructive) BCALC rewritability (under the usual complexity-theoretic assumptions). However, rewritability and TC^0 complexity of query answering are regained for rooted CQs, which are common in practice. The main goal of this paper is to understand to what extent these results transfer to $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$.

We next establish some basic properties of the proposed bag semantics and relate them to the standard set semantics. The following theorem states that satisfiability and query answering under the set semantics and MR semantics are essentially equivalent when multiplicities are ignored, while SQL semantics is in a sense stronger as only one direction of the statements holds.

Theorem 1. *The following statements hold for every* $\mathit{DL\text{-}Lite}_{\mathcal{F}}$ *TBox* $\mathcal{T}$ *and every bag ABox* $\mathcal{A}$ *(recall that* ε *is the duplicate elimination operator):*

1. *if* $(\mathcal{T}, \mathcal{A})$ *is* SQL*-satisfiable then* $(\mathcal{T}, \varepsilon(\mathcal{A}))$ *is satisfiable; and*
2. *for every tuple* $\mathbf{a}$ *over* $\mathbf{I}$*, if* $\mathbf{a} \in q^{(\mathcal{T},\varepsilon(\mathcal{A}))}$ *then* $q^{(\mathcal{T},\mathcal{A})}_{\mathrm{SQL}}(\mathbf{a}) \geq 1$*, and the converse holds whenever* $(\mathcal{T}, \mathcal{A})$ *is* SQL*-satisfiable.*

The same holds when MR *semantics is considered instead of* SQL*; moreover, in this case the converses of both statements hold unconditionally.*

In fact, the converse direction of statement 1 does not hold for SQL semantics; indeed, the $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontology $\mathcal{K}_{\mathrm{ex}}$ of Example 1 is not SQL-satisfiable but ontology $(\mathcal{T}_{\mathrm{ex}}, \varepsilon(\mathcal{A}_{\mathrm{ex}}))$ is satisfiable.

Statement 1 for MR semantics implies that we can check MR-satisfiability of $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontologies using standard techniques for $\mathit{DL\text{-}Lite}_{\mathcal{F}}$ under the set semantics; in particular, we can do it in AC^0 for fixed TBoxes. The following proposition says that for SQL semantics the problem is not much more difficult.

Proposition 1. *The problem of checking whether a* $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ *ontology is SQL-satisfiable is in* TC^0 *when the TBox is fixed.*

Finally, note that, since every SQL-model of a $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontology is also an MR-model, $q^{\mathcal{K}}_{\mathrm{MR}} \subseteq q^{\mathcal{K}}_{\mathrm{SQL}}$ for every CQ q and $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ ontology $\mathcal{K}$; it is not difficult to see that the inclusion may be strict even if $\mathcal{K}$ is SQL-satisfiable.

4 Rewriting and Query Answering in $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$

We next study rewritability of rooted CQs to BCALC over $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$ under our two semantics (recall that the class of all CQs are not rewritable even over $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathit{core}}$ [31]). We first show that for SQL semantics and satisfiable ontologies we can apply the same rewriting as for $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathit{core}}$ [31], which implies TC^0 data complexity of query answering. However, MR semantics is more complex, because, as we show, even simple rooted CQs (in particular, instance queries) have LogSpace-hard query answering, which precludes rewritability (assuming $\mathrm{TC}^0 \subsetneq$ LogSpace). To address this limitation, we introduce a new acyclicity condition on TBoxes, for which we show that the rewritability is regained.

4.1 SQL Semantics

The key ingredient for rewritability and tractability of CQ answering in many description logics is the existence of a universal model.

Definition 7. *For X being* SQL *or* MR, *an X-model $\mathcal{I}$ of a DL-Lite$^{\mathsf{b}}_{\mathcal{F}}$ ontology $\mathcal{K}$ is X-*universal *for a class of CQs $\mathcal{Q}$ if $q_X^{\mathcal{K}} = q^{\mathcal{I}}$ for every $q \in \mathcal{Q}$.*

In the set case, it is well-known that if the ontology is satisfiable, then the so-called canonical interpretation, which can be constructed by the chase procedure, is always a universal model for all CQs [4,8]. Nikolaou et al. generalised this idea to *DL-Lite*$^{\mathsf{b}}_{core}$ [31] and rooted CQs, and it turns out that their canonical interpretation is a universal model for rooted CQs also for *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ under SQL semantics. Before we give the main construction, we introduce the relevant notions from [31].

The *concept closure* $\mathsf{ccl}_{\mathcal{T}}[u,\mathcal{I}]$ of an element $u \in \Delta^{\mathcal{I}}$ in a bag interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ over a TBox $\mathcal{T}$ is the bag of concepts such that, for any concept C,

$$\mathsf{ccl}_{\mathcal{T}}[u,\mathcal{I}](C) = \max\{C_0^{\mathcal{I}}(u) \mid \mathcal{T} \models C_0 \sqsubseteq C\}.$$

In other words, $\mathsf{ccl}_{\mathcal{T}}[u,\mathcal{I}](C)$ is the minimal multiplicity of $C^{\mathcal{J}}(u)$ required for an extension $\mathcal{J}$ of $\mathcal{I}$ to satisfy TBox $\mathcal{T}$ locally in u.

The *union* $\mathcal{I} \cup \mathcal{J}$ of bag interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ interpreting all the individuals in the same way—that is, such that $a^{\mathcal{I}} = a^{\mathcal{J}}$ for each $a \in \mathbf{I}$—is the bag interpretation $(\Delta^{\mathcal{I}} \cup \Delta^{\mathcal{J}}, \cdot^{\mathcal{I}\cup\mathcal{J}})$ with $a^{\mathcal{I}\cup\mathcal{J}} = a^{\mathcal{I}}$ for all $a \in \mathbf{I}$ and $S^{\mathcal{I}\cup\mathcal{J}} = S^{\mathcal{I}} \cup S^{\mathcal{J}}$ for all atomic concepts and roles $S \in \mathbf{C} \cup \mathbf{R}$.

Finally, given a bag ABox $\mathcal{A}$ we denote with $\mathcal{I}_{\mathcal{A}} = (\Delta^{\mathcal{I}_{\mathcal{A}}}, \cdot^{\mathcal{I}_{\mathcal{A}}})$ the *standard interpretation* of $\mathcal{A}$ that is defined as follows: $\Delta^{\mathcal{I}_{\mathcal{A}}} = \mathbf{I}$, $a^{\mathcal{I}_{\mathcal{A}}} = a$ for each $a \in \mathbf{I}$, and $S^{\mathcal{I}_{\mathcal{A}}}(\mathbf{a}) = \mathcal{A}(S(\mathbf{a}))$ for each $S \in \mathbf{C} \cup \mathbf{R}$ and tuple of individuals $\mathbf{a}$.

Definition 8 (Nikolaou et al. [31]). *The* SQL-canonical bag interpretation *$\mathcal{C}_{\mathrm{SQL}}(\mathcal{K})$ of a DL-Lite$^{\mathsf{b}}_{\mathcal{F}}$ ontology $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is the bag interpretation that is the union $\bigcup_{i\geq 0} \mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})$ of the bag interpretations $\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K}) = (\Delta^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})}, \cdot^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})})$ such that $\mathcal{C}^0_{\mathrm{SQL}}(\mathcal{K}) = \mathcal{I}_{\mathcal{A}}$ and, for each $i > 0$, $\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})$ is constructed from $\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})$ as follows:*

- *$\Delta^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})}$ extends $\Delta^{\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})}$ by fresh* anonymous *elements $w^1_{u,R}, \dots, w^{\delta}_{u,R}$ for each $u \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})}$ and role R with*
$$\delta = \mathsf{ccl}_{\mathcal{T}}[u, \mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})](\exists R) - (\exists R)^{\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})}(u);$$
- *$a^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})} = a$ for all $a \in \mathbf{I}$, and, for all $A \in \mathbf{C}$, $P \in \mathbf{R}$ and u, v in $\Delta^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})}$,*
$$A^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})}(u) = \begin{cases} \mathsf{ccl}_{\mathcal{T}}[u, \mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})](A), & \text{if } u \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})}, \\ 0, & \text{otherwise}, \end{cases}$$
$$P^{\mathcal{C}^i_{\mathrm{SQL}}(\mathcal{K})}(u,v) = \begin{cases} P^{\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})}(u,v), & \text{if } u, v \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{SQL}}(\mathcal{K})}, \\ 1, & \text{if } u = w^j_{v,P} \text{ or } v = w^j_{u,P^-}, \\ 0, & \text{otherwise}. \end{cases}$$

The following example illustrates the construction of SQL-canonical models.

Example 3. Consider the *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ ontology $\mathcal{K}'_{ex} = (\mathcal{T}'_{ex}, \mathcal{A}'_{ex})$ with

$$\mathcal{T}'_{ex} = \{\mathsf{Order} \sqsubseteq \exists\mathsf{placedBy},\ \mathsf{Customer} \sqsubseteq \exists\mathsf{placedBy}^-,\ (\mathsf{funct}\ \mathsf{placedBy})\} \text{ and}$$
$$\mathcal{A}'_{ex} = \{\!|\, \mathsf{Order}(o) : 1,\ \mathsf{placedBy}(o, c) : 1,\ \mathsf{Customer}(c) : 4 \,|\!\}.$$

To compute the SQL-canonical interpretation $\mathcal{C}_{\mathrm{SQL}}(\mathcal{K}'_{ex})$ of $\mathcal{K}'_{ex}$, we first set $\mathcal{C}^0_{\mathrm{SQL}}(\mathcal{K}'_{ex}) = \mathcal{I}_{\mathcal{A}'_{ex}}$. For the second step we take $\mathsf{Order}^{\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})} = \mathsf{Order}^{\mathcal{C}^0_{\mathrm{SQL}}(\mathcal{K}'_{ex})}$ and $\mathsf{Customer}^{\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})} = \mathsf{Customer}^{\mathcal{C}^0_{\mathrm{SQL}}(\mathcal{K}'_{ex})}$ as neither of the concepts subsumes another concept in $\mathcal{T}'_{ex}$. The interpretation of $\mathsf{placedBy}$ by $\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})$ is then determined by the concept closures of o and c for the concepts $\exists\mathsf{placedBy}$ and $\exists\mathsf{placedBy}^-$ over $\mathcal{T}'_{ex}$, respectively. Since the former is equal to the multiplicity that o has in $(\exists\mathsf{placedBy})^{\mathcal{C}^0_{\mathrm{SQL}}(\mathcal{K}'_{ex})}$, no new $\exists\mathsf{placedBy}$-successor is added for o. However, the latter is larger than the multiplicity of c in $(\exists\mathsf{placedBy}^-)^{\mathcal{C}^0_{\mathrm{SQL}}(\mathcal{K}'_{ex})}$ by three, and hence c must be associated with new anonymous $\exists\mathsf{placedBy}^-$-successors $w^1_{c,\exists\mathsf{placedBy}^-}, \ldots, w^3_{c,\exists\mathsf{placedBy}^-}$. Therefore, $\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})$ has domain $\mathbf{I} \cup \{w^1_{c,\exists\mathsf{placedBy}^-}, \ldots, w^3_{c,\exists\mathsf{placedBy}^-}\}$, and interprets concepts and roles as follows:

$$\mathsf{Order}^{\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})} = \{\!|\, o : 1 \,|\!\}, \quad \mathsf{Customer}^{\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})} = \{\!|\, c : 4 \,|\!\}, \text{ and}$$
$$\mathsf{placedBy}^{\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})} = \{\!|\, (o, c) : 1, (w^1_{c,\exists\mathsf{placedBy}^-}, c) : 1, \ldots, (w^3_{c,\exists\mathsf{placedBy}^-}, c) : 1 \,|\!\}.$$

Since there is no violation of axioms in $\mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})$, the process terminates at the following step, and we take $\mathcal{C}_{\mathrm{SQL}}(\mathcal{K}'_{ex}) = \mathcal{C}^2_{\mathrm{SQL}}(\mathcal{K}'_{ex}) = \mathcal{C}^1_{\mathrm{SQL}}(\mathcal{K}'_{ex})$. $\lhd$

We are ready to show that the SQL-canonical bag interpretation is indeed SQL-universal for rooted CQs.

Theorem 2. *The* SQL*-canonical bag interpretation of an* SQL*-satisfiable DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ *ontology* $\mathcal{K}$ *is an* SQL*-universal model for the class of rooted CQs.*

Having this result at hand, we can reuse the rewriting of rooted CQs over *DL-Lite*$^{\mathsf{b}}_{\mathrm{core}}$ introduced in [31] for the SQL semantics of *DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$.

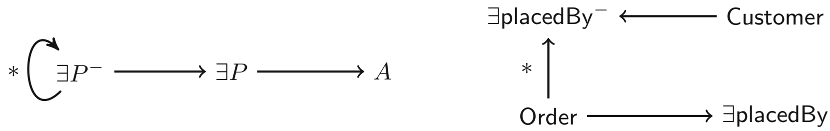
Corollary 1. *Rooted CQs are* SQL*-rewritable to BCALC over DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$.

Since the proof of rewritability in [31] is constructive, SQL-satisfiability is in TC^0, and BCALC evaluation is in TC^0, rooted CQ answering is also in TC^0.

Corollary 2. *Problem* $\textsc{BagCert}_{\mathrm{SQL}}[q, \mathcal{T}]$ *is in* TC^0 *for every rooted CQ* q *and DL-Lite*$_{\mathcal{F}}$ *TBox* $\mathcal{T}$.

4.2 MR Semantics

We begin the study of MR semantics by proving LogSpace-hardness for answering even very simple rooted CQs (in particular, instance queries), which emphasises the difference with SQL semantics. Since answering BCALC queries is in TC^0, this result says that such queries are unlikely to be MR-rewritable to BCALC.



(a) (b)

Fig. 1. The functional dependency graphs of TBoxes $\mathcal{T}$ and $\mathcal{T}'_{ex}$ from Example 4

Theorem 3. *There exist a rooted CQ of the form $A(a)$ with $A \in \mathbf{C}$ and $a \in \mathbf{I}$, and a DL-Lite$_\mathcal{F}$ TBox $\mathcal{T}$ such that* $\textsc{BagCert}_{\mathrm{MR}}[A(a), \mathcal{T}]$ *is* LogSpace*-hard.*

Proof (Sketch). The proof is by an AC^0 reduction from the 1GAP decision problem which is a prototypical complete problem for LogSpace (under AC^0 reductions) [17,20]. The input of 1GAP consists of a directed acyclic graph $H = (V, E)$ with nodes V and edges E such that each node has at most one outgoing edge, and two nodes s, t in V, and the question is whether t is reachable from s in H. For the reduction, we define a *DL-Lite*$^{\mathsf{b}}_\mathcal{F}$ ontology $(\mathcal{T}, \mathcal{A}_H)$ over atomic concept A and role P, where the *DL-Lite*$_\mathcal{F}$ TBox $\mathcal{T}$ comprises axioms $\exists P^- \sqsubseteq \exists P$, $(\mathsf{funct}\ P)$, and $\exists P^- \sqsubseteq A$, and $\mathcal{A}_H$ is the bag ABox defined as follows, for individuals a_v, for each $v \in V$, and $a_\star$:

$$\mathcal{A}_H(P(a_1, a_2)) = \begin{cases} 1, & \text{if } a_1 = a_v \text{ and } a_2 = a_u \text{ for } (v, u) \in E, \\ |V|, & \text{if } a_1 = a_\star \text{ and } a_2 = a_s, \\ 0, & \text{otherwise.} \end{cases}$$

Now t is reachable from s in H if and only if $q_{\mathrm{MR}}^{(\mathcal{T}, \mathcal{A}_H)}() \geq |V|$ for $q = A(a_t)$. □

Recalling that evaluation of BCALC queries is in TC^0, the previous theorem implies that even very simple rooted CQs are unlikely to be MR-rewritable to BCALC. Next we introduce a restriction on TBoxes which, as we will see, guarantees MR-rewritability. The restriction is based on the notions of *functional dependency graphs* and *functional weakly acyclic* TBoxes that respectively specialise the notions of dependency graphs and weak acyclicity defined for sets of tuple-generating dependencies in the context of data exchange [14].

Definition 9. *The* functional dependency graph $G_\mathcal{T}$ *of a DL-Lite*$_\mathcal{F}$ $\mathcal{T}$ *is the directed graph that has all the concepts appearing in* $\mathcal{T}$ *as nodes, a usual edge*

(C_1, C_2) for each $C_1 \sqsubseteq C_2$ in $\mathcal{T}$, and a special edge $(C_1, \exists R^-)^*$ for each $C_1 \sqsubseteq \exists R$ with $(\mathsf{funct}\ R)$ *in $\mathcal{T}$, where, for $P \in \mathbf{R}$, R^- is P if R is P^-. TBox $\mathcal{T}$ is* functionally weakly acyclic *if $G_\mathcal{T}$ has no cycle through a special edge. The* f-depth *of such a TBox $\mathcal{T}$ is the maximum number of special edges along a path in $G_\mathcal{T}$.*

Example 4. The functional dependency graphs of TBoxes $\mathcal{T}$ and $\mathcal{T}'_{ex}$ specified respectively in the proof of Theorem 3 and in Example 3 are depicted in Fig. 1. From the graph of Fig. 1a, we have that the functional depth of $\mathcal{T}$ is ∞; thus, $\mathcal{T}$ is not functionally weakly acyclic. From the graph of Fig. 1b, we have that the functional depth of $\mathcal{T}'_{ex}$ is 1; thus, $\mathcal{T}'_{ex}$ is functionally weakly acyclic. $\triangleleft$

Note that the SQL-canonical interpretation of an MR-satisfiable *DL-Lite*$^\mathsf{b}_\mathcal{F}$ ontology $\mathcal{K}$ specified in Definition 8 is not always an MR-model of $\mathcal{K}$ (e.g., consider ontology $(\{A \sqsubseteq \exists P, (\mathsf{funct}\ P)\}, \{\!|\, A(e) : 2 \,|\!\})$). Below we introduce the construction of MR-canonical interpretations that always results in MR-models for MR-satisfiable ontologies, and start with the auxiliary notion of *closure*.

The *closure* $\mathcal{L}(\mathcal{K})$ of a *DL-Lite*$^\mathsf{b}_\mathcal{F}$ ontology $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is the union $\bigcup_{i\geq 0} \mathcal{L}^i(\mathcal{K})$ of bag interpretations $\mathcal{L}^i(\mathcal{K}) = (\Delta^{\mathcal{L}^i(\mathcal{K})}, \cdot^{\mathcal{L}^i(\mathcal{K})})$ with $\Delta^{\mathcal{L}^i(\mathcal{K})} = \mathbf{I}$ such that $\mathcal{L}^0(\mathcal{K}) = \mathcal{I}_\mathcal{A}$ and, for each $i \geq 1$, $\mathcal{L}^i(\mathcal{K})$ extends $\mathcal{L}^{i-1}(\mathcal{K})$ so that $a^{\mathcal{L}^i(\mathcal{K})} = a$ for all $a \in \mathbf{I}$, and, for all $A \in \mathbf{C}$, $P \in \mathbf{R}$, and $a, b, c, c' \in \mathbf{I}$,

$$A^{\mathcal{L}^i(\mathcal{K})}(a) = \mathsf{ccl}_\mathcal{T}[a, \mathcal{L}^{i-1}(\mathcal{K})](A),$$

$$P^{\mathcal{L}^i(\mathcal{K})}(a,b) = \begin{cases} 0, & \text{if } P^{\mathcal{L}^{i-1}(\mathcal{K})}(a,b) = 0, \\ \max\{\ell_P(a,b), \ell_{P^-}(b,a)\}, & \text{otherwise, where} \end{cases}$$

$$\ell_R(c,c') = \begin{cases} \mathsf{ccl}_\mathcal{T}[c, \mathcal{L}^{i-1}(\mathcal{K})](\exists R), & \text{if } (\mathsf{funct}\ R) \text{ is in } \mathcal{T}, \\ R^{\mathcal{L}^{i-1}(\mathcal{K})}(c,c'), & \text{otherwise.} \end{cases}$$

In fact, if the TBox of $\mathcal{K}$ is functionally weakly acyclic then the closure can be computed in a finite number of steps that does not depend on the ABox.

Proposition 2. *For every DL-Lite$^\mathsf{b}_\mathcal{F}$ ontology $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with a functionally weakly acyclic TBox $\mathcal{T}$ we have $\mathcal{L}(\mathcal{K}) = \bigcup_{i=0}^{d_\mathcal{T}+1} \mathcal{L}^i(\mathcal{K})$.*

The example below demonstrates the notion of closure.

Example 5. Consider the *DL-Lite*$^\mathsf{b}_\mathcal{F}$ ontology $\mathcal{K}'_{ex} = (\mathcal{T}'_{ex}, \mathcal{A}'_{ex})$ with $\mathcal{T}'_{ex}$ as in Example 3 and $\mathcal{A}'_{ex} = \{\!|\, \mathsf{Order}(o) : 3,\ \mathsf{placedBy}(o,c) : 1,\ \mathsf{Customer}(c) : 4 \,|\!\}$. Following the definition of closure on $\mathcal{K}'_{ex}$, we initialise $\mathcal{L}^0(\mathcal{K}'_{ex})$ to $\mathcal{I}_{\mathcal{A}'_{ex}}$ and then, for the next step we trivially have that $\mathsf{Order}^{\mathcal{L}^1(\mathcal{K}'_{ex})} = \mathsf{Order}^{\mathcal{I}_{\mathcal{A}'_{ex}}} = \{\!|\, o : 3 \,|\!\}$ and $\mathsf{Customer}^{\mathcal{L}^1(\mathcal{K}'_{ex})} = \mathsf{Customer}^{\mathcal{I}_{\mathcal{A}'_{ex}}} = \{\!|\, c : 4 \,|\!\}$ since both Order and $\mathsf{Customer}$ do not subsume any concept in $\mathcal{T}'_{ex}$. Then, it can be easily seen that $\mathsf{placedBy}^{\mathcal{L}^1(\mathcal{K}'_{ex})}$ includes only tuple (o,c) with a non-zero multiplicity expressed as the maximum of $\mathsf{ccl}_{\mathcal{T}'_{ex}}[o, \mathcal{L}^0(\mathcal{K}'_{ex})](\exists \mathsf{placedBy}) = 3$ and $\mathsf{placedBy}^{\mathcal{L}^0(\mathcal{K}'_{ex})}(o,c) = 1$; thus $\mathsf{placedBy}^{\mathcal{L}^1(\mathcal{K}'_{ex})} = \{\!|\, (o,c) : 3 \,|\!\}$. Since all axioms in $\mathcal{K}'_{ex}$ are now satisfied, we obtain that $\mathcal{L}^2(\mathcal{K}'_{ex}) = \mathcal{L}^1(\mathcal{K}'_{ex})$; thus $\mathcal{L}(\mathcal{K}'_{ex}) = \bigcup_{i=0}^{d_{\mathcal{T}'_{ex}}+1} \mathcal{L}^i(\mathcal{K}'_{ex}) = \mathcal{L}^2(\mathcal{K}'_{ex})$. $\triangleleft$

We use the closure in the following definition of MR-canonical interpretations. Note the difference in handling functional and non-functional roles when creating anonymous elements, which always produces a most general possible interpretation in each case.

Definition 10. *The* MR-canonical bag interpretation $\mathcal{C}_{\mathrm{MR}}(\mathcal{K})$ *of a DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ *ontology* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *is the union* $\bigcup_{i \geq 0} \mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})$ *such that* $\mathcal{C}^0_{\mathrm{MR}}(\mathcal{K}) = \mathcal{L}(\mathcal{K})$ *and, for each* $i \geq 1$, $\mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})$ *is obtained from* $\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})$ *as follows:*

- $\Delta^{\mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})}$ *extends* $\Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}$ *by*
  - *a fresh* anonymous *element* $w_{u,R}$ *for each* $u \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}$ *and each role* R *with* $(\mathsf{funct}\ R) \in \mathcal{T}$, $\mathsf{ccl}_{\mathcal{T}}[u, \mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})](\exists R) > 0$, *and* $(\exists R)^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}(u) = 0$,
  - *fresh* anonymous *elements* $w^1_{u,R}, \ldots, w^{\delta}_{u,R}$ *for each* $u \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}$ *and each role* R *with* $(\mathsf{funct}\ R) \notin \mathcal{T}$ *and* $\delta = \mathsf{ccl}_{\mathcal{T}}[u, \mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})](\exists R) - (\exists R)^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}(u)$;
- $a^{\mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})} = a$ *for all* $a \in \mathbf{I}$, *and, for all* $A \in \mathbf{C}$, $P \in \mathbf{R}$, *and* u, v *in* $\Delta^{\mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})}$,

$$A^{\mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})}(u) = \begin{cases} \mathsf{ccl}_{\mathcal{T}}[u, \mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})](A), & \text{if } u \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}, \\ 0, & \text{otherwise}, \end{cases}$$

$$P^{\mathcal{C}^i_{\mathrm{MR}}(\mathcal{K})}(u,v) = \begin{cases} P^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}(u,v), & \text{if } u, v \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})}, \\ \mathsf{ccl}_{\mathcal{T}}[u, \mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})](\exists P), & \text{if } u \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})} \text{ and } v = w_{u,P}, \\ \mathsf{ccl}_{\mathcal{T}}[v, \mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})](\exists P^-), & \text{if } v \in \Delta^{\mathcal{C}^{i-1}_{\mathrm{MR}}(\mathcal{K})} \text{ and } u = w_{v,P^-}, \\ 1, & \text{if } v = w^j_{u,P} \text{ or } u = w^j_{v,P^-}, \\ 0, & \text{otherwise.} \end{cases}$$

MR-canonical bag interpretations are illustrated in the following example.

Example 6. Consider the functionally weakly acyclic ontology $\mathcal{K}'_{ex} = (\mathcal{T}'_{ex}, \mathcal{A}'_{ex})$ and its closure $\mathcal{L}(\mathcal{K}'_{ex})$ specified in Example 5. Following Definition 10, the MR-canonical interpretation $\mathcal{C}_{\mathrm{MR}}(\mathcal{K}'_{ex})$ is constructed on the basis of $\mathcal{L}(\mathcal{K}'_{ex})$ by first setting $\mathcal{C}^0_{\mathrm{MR}}(\mathcal{K}'_{ex}) = \mathcal{L}(\mathcal{K}'_{ex})$. Then, for the next step we set

$$\mathsf{Order}^{\mathcal{C}^1_{\mathrm{MR}}(\mathcal{K}'_{ex})} = \mathsf{Order}^{\mathcal{C}^0_{\mathrm{MR}}(\mathcal{K}'_{ex})} \text{ and } \mathsf{Customer}^{\mathcal{C}^1_{\mathrm{MR}}(\mathcal{K}'_{ex})} = \mathsf{Customer}^{\mathcal{C}^0_{\mathrm{MR}}(\mathcal{K}'_{ex})}$$

as neither Order nor $\mathsf{Customer}$ subsumes any concept in $\mathcal{T}'_{ex}$, and set $\mathsf{placedBy}^{\mathcal{C}^1_{\mathrm{MR}}(\mathcal{K}'_{ex})} = \{\!|\, (o, c) : 3, (w^1_{c,\exists\mathsf{placedBy}^-}, c) : 1 \,|\!\}$. The latter follows by the fact that o has already a $\mathsf{placedBy}$-successor in $\mathcal{C}^0_{\mathrm{MR}}(\mathcal{K})$ while at the same time the multiplicity of c in the extension of $\exists\mathsf{placedBy}^-$ under $\mathcal{C}^1_{\mathrm{MR}}(\mathcal{K})$ must be increased by 1 so that inclusion $\mathsf{Customer} \sqsubseteq \exists\mathsf{placedBy}^-$ is satisfied; since $(\mathsf{funct}\ \mathsf{placedBy}^-)$ is not in $\mathcal{T}'_{ex}$, this must be done by introducing a fresh anonymous element $w^1_{c,\exists\mathsf{placedBy}^-}$ to the domain of $\mathcal{C}^1_{\mathrm{MR}}(\mathcal{K})$ and making it a $\exists\mathsf{placedBy}^-$-successor of c. All axioms are satisfied in $\mathcal{C}^1_{\mathrm{MR}}(\mathcal{K}'_{ex})$, and hence $\mathcal{C}_{\mathrm{MR}}(\mathcal{K}'_{ex}) = \mathcal{C}^1_{\mathrm{MR}}(\mathcal{K}'_{ex})$. ◁

As the following theorem says, the MR-canonical bag interpretation is an MR-universal model, as desired.

Theorem 4. *The* MR*-canonical bag interpretation* $\mathcal{C}_{\mathrm{MR}}(\mathcal{K})$ *of an* MR*-satisfiable DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ *ontology* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *with* $\mathcal{T}$ *functionally weakly acyclic is an* MR*-universal model for the class of rooted CQs.*

Example 7. Consider ontology $\mathcal{K}'_{ex}$ and its MR-canonical interpretation $\mathcal{C}_{\mathrm{MR}}(\mathcal{K}'_{ex})$ as in Example 6. Consider also rooted CQs $q_1(x) = \exists y.\, \mathsf{placedBy}(x, y)$ and $q_2(x) = \exists y.\, \mathsf{placedBy}(y, x)$. It is straightforward to verify that the MR-bag certain answers to q_1 and q_2 over $\mathcal{K}'_{ex}$ are respectively given by bags $q_{1\,\mathrm{MR}}^{\mathcal{K}'_{ex}} = \{\!|\, o : 3 \,|\!\}$ and $q_{2\,\mathrm{MR}}^{\mathcal{K}'_{ex}} = \{\!|\, c : 4 \,|\!\}$, and that these bags coincide with the bag answers to q_1 and q_2 over $\mathcal{C}_{\mathrm{MR}}(\mathcal{K}'_{ex})$, respectively. This supports our expectation that $\mathcal{C}_{\mathrm{MR}}(\mathcal{K}'_{ex})$ is an MR-universal model of $\mathcal{K}'_{ex}$ for the class of rooted CQs. ◁

By adapting and extending the techniques in [31], we establish that rooted CQs are MR-rewritable to BCALC over the restricted ontology language.

Theorem 5. *Rooted CQs are* MR*-rewritable to BCALC over DL-Lite*$^{\mathsf{b}}_{\mathcal{F}}$ *with functionally weakly acyclic TBoxes.*

Hence, under the restrictions, query answering is indeed feasible in TC^0.

Corollary 3. *Problem* $\textsc{BagCert}_{\mathrm{MR}}[q, \mathcal{T}]$ *is in* TC^0 *for every rooted CQ* q *and functionally weakly acyclic DL-Lite*$_{\mathcal{F}}$ *TBox* $\mathcal{T}$*.*

5 Related Work

Jiang [19] was the first to propose a bag semantics for the DL $\mathcal{ALC}$, which is however incompatible with SQL and incomparable to the semantics developed in this work. Motivated by the semantic differences arising between the set-based theory and bag-based practice of OBDA and data exchange settings, Nikolaou et al. [30,31] as well as Hernich and Kolaitis [16] studied respectively the foundations of OBDA and data exchange settings under a bag semantics compatible with SQL. To the best of our knowledge, our work, which builds on [30,31], is the first one to study the interaction of functionality and inclusion axioms under a bag semantics. A bag semantics for functional dependencies, which generalises our SQL semantics, has been studied before by Köhler and Link [22] who, however, studied only schema design issues. Owing to the aforementioned works and the work by Console et al. [13], there is now a better understanding of CQ answering under bag semantics for frameworks managing incomplete information. This latter problem is closely related to answering queries using aggregate functions the semantics of which has been studied before in the context of inconsistent databases [3], data exchange [2], and *DL-Lite* [11,23], where the resulting frameworks do not treat bags as first-class citizens. Handling bags through sets was also the approach followed in the 90's by Mumick et al. [29] for supporting bags in Datalog and recently by Bertossi et al. [5] for Datalog$^{\pm}$.

6 Conclusions and Future Work

In this paper, we studied two bag semantics for functionality axioms: our first SQL semantics follows the bag semantics of SQL for primary keys, while the second MR semantics is more general and gives more modelling freedom. Combining the semantics with the bag semantics of $\mathit{DL\text{-}Lite}_{\mathit{core}}$ of [30,31], we studied the problems of satisfiability, query answering, and rewritability for the resulting logical language $\mathit{DL\text{-}Lite}^{\mathsf{b}}_{\mathcal{F}}$. It is interesting to see how our work generalises to the case of n-ary predicates. This case has been studied only recently in the context of data exchange settings [16] and Datalog$^{\pm}$ [5], which, however, do not consider functional dependencies. We also anticipate our work will be useful for laying the foundations of aggregate queries in SPARQL under entailment regimes [21].

Acknowledgements. This research was supported by the SIRIUS Centre for Scalable Data Access and the EPSRC projects DBOnto, MaSI^3, and ED^3.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Afrati, F.N., Kolaitis, P.G.: Answering aggregate queries in data exchange. In: Proceedings of PODS, pp. 129–138 (2008)
3. Arenas, M., Bertossi, L.E., Chomicki, J., He, X., Raghavan, V., Spinrad, J.P.: Scalar aggregation in inconsistent databases. Theor. Comput. Sci. **296**(3), 405–434 (2003)
4. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. **36**, 1–69 (2009)
5. Bertossi, L.E., Gottlob, G., Pichler, R.: Datalog: bag semantics via set semantics. In: Proceedings of ICDT, pp. 16:1–16:19 (2019)
6. Bienvenu, M., Lutz, C., Wolter, F.: Query containment in description logics reconsidered. In: Proceedings of KR, pp. 221–231 (2012)
7. Calvanese, D., et al.: Ontop: answering SPARQL queries over relational databases. Semant. Web **8**(3), 471–487 (2017)
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. J. Autom. Reason. **39**(3), 385–429 (2007)
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Path-based identification constraints in description logics. In: Proceedings of KR, pp. 231–241 (2008)
10. Calvanese, D., De Giacomo, G., Lenzerini, M.: Identification constraints and functional dependencies in description logics. In: Proceedings of IJCAI, pp. 155–160 (2001)
11. Calvanese, D., Kharlamov, E., Nutt, W., Thorne, C.: Aggregate queries over ontologies. In: Proceedings of ONISW, pp. 97–104 (2008)
12. Chaudhuri, S., Vardi, M.Y.: Optimization of real conjunctive queries. In: Proceedings of PODS, pp. 59–70 (1993)
13. Console, M., Guagliardo, P., Libkin, L.: On querying incomplete information in databases under bag semantics. In: Proceedings of IJCAI, pp. 993–999 (2017)

14. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. **336**(1), 89–124 (2005)
15. Grumbach, S., Milo, T.: Towards tractable algebras for bags. J. Comput. Syst. Sci. **52**(3), 570–588 (1996)
16. Hernich, A., Kolaitis, P.G.: Foundations of information integration under bag semantics. In: Proceedings of LICS, pp. 1–12 (2017)
17. Immerman, N.: Languages that capture complexity classes. SIAM J. Comput. **16**(4), 760–778 (1987)
18. Ioannidis, Y.E., Ramakrishnan, R.: Containment of conjunctive queries: beyond relations as sets. ACM Trans. Database Syst. **20**(3), 288–324 (1995)
19. Jiang, Y.: Description logics over multisets. In: Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web, pp. 1–12 (2010)
20. Jones, N.D.: Space-bounded reducibility among combinatorial problems. J. Comput. Syst. Sci. **11**(1), 68–85 (1975)
21. Kaminski, M., Kostylev, E.V., Cuenca Grau, B.: Query nesting, assignment, and aggregation in SPARQL 1.1. ACM Trans. Database Syst. **42**(3), 17:1–17:46 (2017)
22. Köhler, H., Link, S.: Armstrong axioms and Boyce-Codd-Heath normal form under bag semantics. Inf. Process. Lett. **110**(16), 717–724 (2010)
23. Kostylev, E.V., Reutter, J.L.: Complexity of answering counting aggregate queries over DL-Lite. Web Semant. **33**, 94–111 (2015)
24. Lenzerini, M.: Data integration: a theoretical perspective. In: Proceedings of PODS, pp. 233–246 (2002)
25. Libkin, L.: Expressive power of SQL. Theor. Comput. Sci. **296**(3), 379–404 (2003)
26. Libkin, L., Wong, L.: Query languages for bags and aggregate functions. J. Comput. Syst. Sci. **55**(2), 241–272 (1997)
27. Lutz, C., Areces, C., Horrocks, I., Sattler, U.: Keys, nominals, and concrete domains. J. Artif. Intell. Res. **23**, 667–726 (2005)
28. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles (Second Edition). W3C recommendation, W3C (2012). http://www.w3.org/TR/owl2-profiles/
29. Mumick, I.S., Pirahesh, H., Ramakrishnan, R.: The magic of duplicates and aggregates. In: Proceedings of VLDB, pp. 264–277 (1990)
30. Nikolaou, C., Kostylev, E.V., Konstantinidis, G., Kaminski, M., Cuenca Grau, B., Horrocks, I.: The bag semantics of ontology-based data access. In: Proceedings of IJCAI, pp. 1224–1230 (2017)
31. Nikolaou, C., Kostylev, E.V., Konstantinidis, G., Kaminski, M., Cuenca Grau, B., Horrocks, I.: Foundations of ontology-based data access under bag semantics. Artif. Intell. **274**, 91–132 (2019)
32. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Semant. **10**, 133–173 (2008)
33. Toman, D., Weddell, G.E.: On keys and functional dependencies as first-class citizens in description logics. J. Autom. Reason. **40**(2–3), 117–132 (2008)