Artificial Intelligence 274 (2019) 91–132

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Foundations of ontology-based data access under bag semantics ☆

Charalampos Nikolaou [a,*], Egor V. Kostylev [a], George Konstantinidis [b,a], Mark Kaminski [a], Bernardo Cuenca Grau [a], Ian Horrocks [a]

[a] Department of Computer Science, University of Oxford, UK
[b] Department of Electronics and Computer Science, University of Southampton, UK

ARTICLE INFO

Article history:
Received 12 January 2018
Received in revised form 10 February 2019
Accepted 12 February 2019
Available online 15 February 2019

Keywords:
Ontology-based data access
Description logics
Bag semantics
Query answering
Query rewriting

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 to ontology predicates views over the data. The conventional semantics of OBDA is set-based—that is, the extension of the views defined by the mappings does not contain duplicate tuples. This treatment is, however, in disagreement with the standard semantics of database views and database management systems in general, which is based on bags and where duplicate tuples are retained by default. The distinction between set and bag semantics in databases is very significant in practice, and it influences the evaluation of aggregate queries.

In this article, we propose and study a bag semantics for OBDA which provides a solid foundation for the future study of aggregate and analytic queries. Our semantics is compatible with both the bag semantics of database views and the set-based conventional semantics of OBDA. Furthermore, it is compatible with existing bag-based semantics for data exchange recently proposed in the literature. We show that adopting a bag semantics makes conjunctive query answering in OBDA coNP-hard in data complexity. To regain tractability of query answering, we consider suitable restrictions along three dimensions, namely, the query language, the ontology language, and the adoption of the unique name assumption. Our investigation shows a complete picture of the computational properties of query answering under bag semantics over ontologies in the DL-Lite family.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Ontology-based data access (OBDA) is an increasingly popular approach to enable uniform access to multiple data sources with diverging schemas [2–7]. An ontology in OBDA is represented using a fragment of first-order logic and provides a unifying conceptual model for the data sources. The ontology is linked to the schema of each data source by *global-as-view* (GAV) mappings [8], which declaratively assign views over the data to predicates in the vocabulary of the ontology. Users of OBDA systems are typically unaware of the details of the source schemas or the mappings, and access the data by means of

☆ This article extends our IJCAI-2017 conference publication [1].

* Corresponding author.
E-mail addresses: charalampos.nikolaou@cs.ox.ac.uk (C. Nikolaou), egor.kostylev@cs.ox.ac.uk (E.V. Kostylev), g.konstantinidis@soton.ac.uk (G. Konstantinidis), mark.kaminski@cs.ox.ac.uk (M. Kaminski), bernardo.cuenca.grau@cs.ox.ac.uk (B. Cuenca Grau), ian.horrocks@cs.ox.ac.uk (I. Horrocks).

https://doi.org/10.1016/j.artint.2019.02.003
0004-3702/© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

(conjunctive) queries formulated using only the vocabulary of the ontology. The answers to a query are those that logically follow from the union of the ontology and the materialisation over the sources of the views defined by the mappings.

The formalism of choice for representing ontologies in OBDA is the description logic *DL-Lite*$_{\mathcal{R}}$ [9], which underpins the Web ontology language OWL 2 QL [10]. *DL-Lite*$_{\mathcal{R}}$ is a language designed to ensure that all input queries to the OBDA system are *first-order rewritable*—that is, the answers to every input query can be obtained by first reformulating the query as a set of relational queries over the source schemas, and then evaluating the reformulated queries over the source data [2]. In practice, such reformulation involves two steps known as *rewriting* and *unfolding* [8,11]. In the rewriting step, the original query is transformed into a first-order query that captures the relevant information in the ontology; in turn, in the unfolding step, the query computed in the rewriting step is further rewritten as a set of relational queries over the schemas of the sources using the relevant mappings.

OBDA has received a great deal of attention in recent years. Researchers have studied the limits of first-order rewritability in ontology languages [9,12], established bounds on the size of rewritings [13,14], developed optimisation techniques [15–17], and implemented systems well-suited for real-world applications [3,5].

Example 1. Consider a music encyclopedia that collects metadata about music records and makes it available to the public. The encyclopedia gathers data from record labels, which are maintained in separate relational tables. For instance, it contains the table Columbia(art_nm, r_title, r_year, r_date, r_loc) that provides the names of the records (r_title) artists (art_nm) have cut on the record label Columbia together with their release years (r_year), recording dates (r_date), and locations (r_loc). In addition, we consider the table Verve_Wind(name, title, year) that provides information about the names of the records (title) artists (name) playing wind instruments have cut on Verve (for the sake of illustration, we assume that records by Verve are organised in different tables according to the type of instrument played by the lead performer; in our running example, we restrict ourselves to wind instruments). The following is a relational database instance $\mathcal{D}_{ex}$ providing information about the records that trumpeter Miles Davis and pianist Keith Jarrett have cut on these two labels.

Columbia:	art_nm	r_title	r_year	r_date	r_loc
	M. Davis	*Kind of Blue*	*1959*	*2/3/59 and 22/4/59*	*New York*
	M. Davis	*A Tribute to Jack Johnson*	*1971*	*7/4/70*	*New York*
	K. Jarrett	*Expectations*	*1972*	*5/4/72–27/4/72*	*New York*

Verve_Wind:	name	title	year
	M. Davis	*Ascenseur pour l'Échafaud*	*1958*

To integrate this data, the music encyclopedia relies on a *DL-Lite*$_{\mathcal{R}}$ ontology with TBox $\mathcal{T}_{ex}$, which defines unary predicates, called *concepts*, such as Musician, WindPlayer, and Record, and binary predicates, called *roles*, such as hasMusician. TBox $\mathcal{T}_{ex}$ describes the meaning of these predicates using the following axioms, called *inclusions*:

$$\mathsf{WindPlayer} \sqsubseteq \mathsf{Musician}, \quad \text{saying that every player of a wind instrument is a musician,}$$
$$\mathsf{Record} \sqsubseteq \exists \mathsf{hasMusician}, \quad \text{saying that every record is associated to some musician.}$$

The extension of the concepts and roles in the ontology based on the data in Columbia and Verve_Wind tables is determined by a set of GAV mapping assertions of the form $\Phi(x) \leadsto A(x)$ or $\Psi(x, y) \leadsto P(x, y)$, where $\Phi(x)$ and $\Psi(x, y)$ are SQL queries, A is a concept, and P is a role. Such mappings can also be seen as database views with names A or P and definitions $\Phi(x)$ or $\Psi(x, y)$, respectively. The following set Σ_{ex} of mappings is used to populate concepts Musician, WindPlayer, and Record, as well as role hasMusician with data from $\mathcal{D}_{ex}$:

σ_1: SELECT art_nm AS x FROM Columbia $\leadsto$ Musician(x),
σ_2: SELECT name AS x FROM Verve_Wind $\leadsto$ WindPlayer(x),
σ_3: SELECT r_title AS x FROM Columbia $\leadsto$ Record(x),
σ_4: SELECT title AS x FROM Verve_Wind $\leadsto$ Record(x),
σ_5: SELECT r_title AS x, art_nm AS y FROM Columbia $\leadsto$ hasMusician(x, y),
σ_6: SELECT title AS x, name AS y FROM Verve_Wind $\leadsto$ hasMusician(x, y).

Finally, user queries are formulated using only the vocabulary of $\mathcal{T}_{ex}$, and users are typically unaware of the schema of $\mathcal{D}_{ex}$ and the definition of the mappings. ◁

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, called *assertions*, over the ontology predicates. This treatment is, however, in disagreement with the standard semantics of database views and database management systems in general, which is based on bags and where duplicate tuples are retained by default [18,19]. The

distinction between set and bag semantics in databases is significant in practice; in particular, it influences the evaluation of aggregate queries that combine various aggregation functions such as MIN, MAX, SUM, COUNT, and AVG with the grouping functionality provided in SQL by the GROUP BY construct.

The mismatch between the set semantics of OBDA and the bag semantics of database views manifests itself already in our running example.

Example 2. Consider TBox $\mathcal{T}_{ex}$, mappings Σ_{ex}, and database instance $\mathcal{D}_{ex}$ specified in Example 1. Consider also the query $q_{ex}(x) = \mathsf{Musician}(x)$, which asks for all musicians. Under the conventional OBDA semantics, the virtual ABox $\mathcal{A}_{ex}$ corresponding to $\mathcal{D}_{ex}$ and Σ_{ex} comprises the following assertions:

$\mathsf{Musician}(\mathit{M.\ Davis})$, $\mathsf{Musician}(\mathit{K.\ Jarrett})$, $\mathsf{WindPlayer}(\mathit{M.\ Davis})$,
$\mathsf{Record}(\mathit{Kind\ of\ Blue})$, $\mathsf{Record}(\mathit{A\ Tribute\ to\ Jack\ Johnson})$, $\mathsf{Record}(\mathit{Expectations})$,
$\mathsf{Record}(\mathit{Ascenseur\ pour\ l'Échafaud})$,
$\mathsf{hasMusician}(\mathit{Kind\ of\ Blue}, \mathit{M.\ Davis})$, $\mathsf{hasMusician}(\mathit{A\ Tribute\ to\ Jack\ Johnson}, \mathit{M.\ Davis})$,
$\mathsf{hasMusician}(\mathit{Expectations}, \mathit{K.\ Jarrett})$, $\mathsf{hasMusician}(\mathit{Ascenseur\ pour\ l'Échafaud}, \mathit{M.\ Davis})$.

The answers to $q_{ex}(x)$ over ontology $\langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$ are given by the set $\{\mathit{M.\ Davis}, \mathit{K.\ Jarrett}\}$ since *M. Davis* and *K. Jarrett* are the only instances of concept Musician entailed by the ontology $\langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$.

To compute these answers in practice, however, an OBDA system would exploit the first-order rewritability property of OBDA. In particular, it would first rewrite $q_{ex}(x)$ into the union of queries $\mathsf{Musician}(x)$ and $\mathsf{WindPlayer}(x)$ using inclusion $\mathsf{WindPlayer} \sqsubseteq \mathsf{Musician}$ in $\mathcal{T}_{ex}$. Then, in a second step, the system would unfold each disjunct of the rewritten query into a query over the database $\mathcal{D}_{ex}$ using the mappings in Σ_{ex}. The result is the SQL query $\Phi_{ex}(x)$ comprising the union of the SQL queries $\Phi_{\sigma_1}(x)$ and $\Phi_{\sigma_2}(x)$ mentioned on the left-hand side of mappings σ_1 and σ_2, respectively. The query $\Phi_{ex}(\mathrm{x})$ is finally evaluated directly over $\mathcal{D}_{ex}$.

According to the semantics of OBDA, the answers to $\Phi_{ex}(x)$ over ontology $\langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$ should coincide with the evaluation of SQL query $\Phi_{ex}(\mathrm{x})$ over $\mathcal{D}_{ex}$. This is, however, not the case in our example. In particular, evaluating $\Phi_{ex}(\mathrm{x})$ over $\mathcal{D}_{ex}$ yields a *bag* containing two occurrences of *M. Davis* and one occurrence of *K. Jarrett*. This is because duplicates in the answers to SQL queries are kept by default unless duplicate elimination is explicitly requested by using the DISTINCT operator in the SELECT clause of a query.[1]

This discrepancy between OBDA semantics and the semantics of database views may occur even if the TBox of the ontology is empty. In particular, in such a case the evaluation of $q_{ex}(x)$ over ABox $\mathcal{A}_{ex}$ does not coincide with the evaluation of the rewritten query ($\Phi_{\sigma_1}(\mathrm{x})$ in this case) over $\mathcal{D}_{ex}$. $\lhd$

Example 2 suggests that the conventional approach to OBDA can faithfully represent only a subset of GAV mapping assertions—those whose SQL query contains the DISTINCT operator in the top-level SELECT clause.

In this paper, we propose and study a bag semantics for OBDA, which provides a solid foundation for future research on aggregate and analytic queries. Our semantics is compatible with *(i)* the bag semantics of database views; *(ii)* the set-based conventional semantics of OBDA; and *(iii)* the bag semantics recently proposed by Hernich and Kolaitis [20] in the context of data exchange.

1.1. Contributions and organisation

The contributions and organisation of this paper are as follows. In Section 3 we introduce the bag semantics of an OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ consisting of a *DL-Lite*$_{\mathcal{R}}$ TBox $\mathcal{T}$, a set of GAV mappings Σ, and a (bag) database instance $\mathcal{D}$. We also define the notion of certain answers to conjunctive queries as well as the associated query answering problem.

In Section 4 we define the ontology language *DL-Lite*$_{\mathcal{R}}^{\mathrm{b}}$ and two of its natural fragments. A distinctive feature of *DL-Lite*$_{\mathcal{R}}^{\mathrm{b}}$ is that ABoxes are bags of assertions rather than sets. Syntactically, this language allows for the same TBoxes as *DL-Lite*$_{\mathcal{R}}$, but their semantics also takes multiplicities into account. We show that, as in the case of conventional OBDA, the certain answers to a query over an OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ can be characterised as those that logically follow from the union of the TBox $\mathcal{T}$ and a *virtual bag ABox* $\mathcal{A}_{\Sigma,\mathcal{D}}$ representing the materialisation of the views defined by the mappings Σ over the database $\mathcal{D}$. As a result, the data complexity of OBDA query answering under bag semantics coincides with that of query answering over *DL-Lite*$_{\mathcal{R}}^{\mathrm{b}}$.

In Section 5 we then establish the relationship between bag and set semantics of *DL-Lite*$_{\mathcal{R}}^{\mathrm{b}}$ and *DL-Lite*$_{\mathcal{R}}$, respectively. In particular, we show that, on the one hand, satisfiability checking in *DL-Lite*$_{\mathcal{R}}^{\mathrm{b}}$ reduces to satisfiability checking in *DL-Lite*$_{\mathcal{R}}$ and, on the other hand, query answers under bag and set semantics coincide if multiplicities are ignored. There are, however, key properties of the conventional semantics of *DL-Lite*$_{\mathcal{R}}$ that are no longer satisfied by the bag semantics: unlike the set

[1] Bag semantics offers two types of union, called *maximal* and *arithmetic*; the first computes the maximum number of occurrences for every tuple in the provided operands whereas the second adds these numbers up. The answer we provide here is based on maximal union for reasons that will be made clear later on in this article. The use of arithmetic union does not affect our motivation.

case, $DL\text{-}Lite^{\mathsf{b}}_{\mathcal{R}}$ ontologies may not have a universal model for conjunctive queries—that is, a single model the answers over which are precisely the certain answers to all such queries; moreover, query answers may be sensitive to the adoption of the *unique name assumption* (*UNA*).

In Section 6 we show that conjunctive query answering under bag semantics is computationally more challenging than in the set case. In particular, we establish three incomparable coNP lower bounds for the data complexity of the problem. We first show that it is coNP-hard even if we restrict the ontology language to $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{CORE}}$—that is, the language where role inclusions are disallowed—regardless of whether the UNA is adopted or not. Second, we show that without the UNA the problem is hard even if the ontologies do not have existential quantification on the right-hand side of concept inclusions (i.e., in the $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{RDFS}}$ ontology language) and the queries are restricted to the so-called *rooted conjunctive queries* [21]—that is, conjunctive queries with all their connected components containing at least one constant or answer variable; this class of queries comprises most practical OBDA queries. The third hardness result is established for the same settings as in the second case except that the UNA is adopted, but there are no restrictions on inclusion axioms.

In Section 7 we make a first step on the way to regain tractability of conjunctive query answering. In particular, we show that rooted conjunctive queries admit a universal model over $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{CORE}}$ ontologies, regardless of the adoption of the UNA.

In Section 8 we employ this result to show that rooted conjunctive queries are rewritable over $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{CORE}}$ to queries in a bag analogue of relational calculus, BCALC, which can be evaluated directly on the ABox of the ontology. Using known results on bag databases, we conclude that the corresponding query answering problem is tractable, in particular, in LogSpace.

In Section 9 we establish similar results for arbitrary conjunctive queries and $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{RDFS}}$—that is, the language that allows for role inclusions, but does not allow for existential quantification on the right-hand side of concept inclusions; however, these positive results hold only when the UNA is adopted, while we already know that without the UNA the problem is coNP-hard and hence non-rewritable to BCALC.

In Section 10 we combine the results of the previous two sections and establish rewritability and tractability of query answering for the ontology language $DL\text{-}Lite^{\mathsf{b}}_{\mathcal{R}^-}$ capturing both $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{CORE}}$ and $DL\text{-}Lite^{\mathsf{b}}_{\mathsf{RDFS}}$. This language allows for both role inclusions and existential quantification on the right-hand side of concept inclusions. To establish rewritability, however, the language essentially forbids interaction between these two features. Also, the setting inherits all of the restrictions imposed on the previous cases—that is, it adopts the UNA and considers only rooted conjunctive queries.

Finally, in Section 11 we provide a comprehensive discussion of related work, and in Section 12 we discuss possible extensions of our OBDA framework.

2. Preliminaries

In this section we recapitulate the basic definitions that we use in the remainder of the paper. In Section 2.1 we introduce the syntax and (set-based) model-theoretic semantics of the standard ontology and query languages for OBDA. Then, in Section 2.2, we introduce conjunctive queries and define the associated query answering problem. In Section 2.3 we review the common operations on bags. Finally, in Section 2.4 we specify a bag relational calculus that we will exploit later on to express query rewritings over ontologies; our calculus is embeddable into the bag algebra for relational databases by Grumbach and Milo [22], which we discuss in the accompanying Appendix.[2]

2.1. Syntax and semantics of $DL\text{-}Lite_{\mathcal{R}}$ ontologies

We fix a vocabulary consisting of countably infinite and pairwise disjoint sets of *individuals* $\mathbf{I}$ (or *constants*), *variables* $\mathbf{X}$, *atomic concepts* $\mathbf{C}$ (unary predicates) and *atomic roles* $\mathbf{R}$ (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 *existentially quantified concept* $\exists R$, where R is a role. An *inclusion axiom* (or just *inclusion*) is an expression of the form $S_1 \sqsubseteq S_2$ with S_1 and S_2 either both concepts, in which case we speak of a *concept inclusion*, or both roles, in which case we speak of a *role inclusion*. A *disjointness axiom* is an expression of the form $\mathsf{Disj}(S_1, S_2)$ with S_1 and S_2 either both concepts or both roles. A $DL\text{-}Lite_{\mathcal{R}}$ *TBox* is a finite set of inclusions and disjointness axioms. A *concept assertion* is of the form $A(a)$ with $a \in \mathbf{I}$ and $A \in \mathbf{C}$. A *role assertion* is of the form $P(a, b)$ with $a, b \in \mathbf{I}$ and $P \in \mathbf{R}$. A $DL\text{-}Lite_{\mathcal{R}}$ *ABox* is a finite set of concept and role assertions. A $DL\text{-}Lite_{\mathcal{R}}$ *ontology* is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$ with $\mathcal{T}$ a $DL\text{-}Lite_{\mathcal{R}}$ TBox and $\mathcal{A}$ a $DL\text{-}Lite_{\mathcal{R}}$ ABox.

An *interpretation* I (or *set interpretation*, when the context matters) is a pair $\langle \Delta^I, \cdot^I \rangle$, where the *domain* Δ^I is a non-empty set, and the *interpretation function* $\cdot^I$ maps each individual $a \in \mathbf{I}$ to an element $a^I \in \Delta^I$, each atomic concept $A \in \mathbf{C}$ to a subset A^I of Δ^I, and each atomic role $P \in \mathbf{R}$ to a subset P^I of $\Delta^I \times \Delta^I$. The interpretation function extends to other concepts and roles as follows:

$$(R^-)^I = \{(u, v) \mid (v, u) \in R^I\},$$
$$(\exists R)^I = \{u \mid \exists v \text{ such that } (u, v) \in R^I\}.$$

[2] There are alternative algebraic query languages for bags, such as that by Libkin and Wong [23]; however, their expressive power is equivalent to that of Grumbach and Milo's algebra and hence the choice of an underpinning algebraic query language is immaterial to the results in this article.

Interpretation I is *finite* if so is Δ^I. An interpretation $I = \langle \Delta^I, \cdot^I \rangle$ *satisfies* the *unique name assumption* (or *UNA*) whenever I interprets distinct individuals from $\mathbf{I}$ with distinct elements from Δ^I—that is, the inequality $a^I \neq b^I$ holds when $a, b \in \mathbf{I}$ with $a \neq b$. An interpretation I *satisfies* an inclusion $S_1 \sqsubseteq S_2$ if $S_1^I \subseteq S_2^I$, and it *satisfies* a disjointness axiom $\mathsf{Disj}(S_1, S_2)$ if $S_1^I \cap S_2^I = \emptyset$. Interpretation I *satisfies* a TBox $\mathcal{T}$, written $I \models \mathcal{T}$, if I satisfies every axiom in $\mathcal{T}$. A TBox $\mathcal{T}$ *entails* an axiom α, written $\mathcal{T} \models \alpha$, if every interpretation satisfying $\mathcal{T}$ also satisfies α; TBox $\mathcal{T}$ *entails* α *under the UNA* if the same holds for every interpretation satisfying the UNA. An interpretation I *satisfies* an ABox $\mathcal{A}$ if $a^I \in A^I$ for all $A(a) \in \mathcal{A}$ and $(a^I, b^I) \in P^I$ for all $P(a, b) \in \mathcal{A}$. An interpretation is a *model* of a *DL-Lite*$_\mathcal{R}$ ontology $\langle \mathcal{T}, \mathcal{A} \rangle$ if it satisfies $\mathcal{T}$ and $\mathcal{A}$. An ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is *satisfiable* if it has a model, and it is *satisfiable under the UNA* if it has a model satisfying the UNA.

It is well-known that a *DL-Lite*$_\mathcal{R}$ ontology is satisfiable if and only if it is satisfiable under the UNA, and this can be tested in NLogSpace in general and in AC^0 if the TBox is fixed; similarly, a *DL-Lite*$_\mathcal{R}$ TBox entails an axiom if and only if it entails the axiom under the UNA [24].

In this paper, we will also consider the following three sublanguages of *DL-Lite*$_\mathcal{R}$:

- *DL-Lite*$_{\text{CORE}}$ restricts *DL-Lite*$_\mathcal{R}$ by disallowing in TBoxes role inclusions and role disjointness axioms;
- *DL-Lite*$_{\text{RDFS}}$ restricts *DL-Lite*$_\mathcal{R}$ by disallowing in TBoxes concept and role disjointness axioms as well as existentially quantified concepts on the right-hand side of concept inclusions;
- *DL-Lite*$_{\mathcal{R}^-}$ restricts *DL-Lite*$_\mathcal{R}$ by disallowing in TBoxes $\mathcal{T}$ inclusions of the form $C \sqsubseteq \exists R$ whenever $\mathcal{T}$ contains inclusion $R \sqsubseteq S$ for some role S different from R.

Note that *DL-Lite*$_{\text{CORE}}$ and *DL-Lite*$_{\text{RDFS}}$ are incomparable fragments of *DL-Lite*$_\mathcal{R}$, whereas *DL-Lite*$_{\mathcal{R}^-}$ extends both *DL-Lite*$_{\text{CORE}}$ and *DL-Lite*$_{\text{RDFS}}$. The restrictions imposed in *DL-Lite*$_{\mathcal{R}^-}$ limit the interaction between concept and role inclusions in TBoxes.

2.2. Queries over ontologies

A *conjunctive query* (or *CQ*) $q(\mathbf{x})$ with *answer* variables $\mathbf{x}$ is a formula $\exists \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})$ in first-order logic with equality, where $\mathbf{x}$ and $\mathbf{y}$ are (possibly empty) repetition-free tuples of variables from $\mathbf{X}$ 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 clear from the context, then we may write q instead of $q(\mathbf{x})$. The equality atoms of the form $(z = t)$ in a CQ $q(\mathbf{x}) = \exists \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})$ 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 $P(t_1, t_2)$ in ϕ. In what follows, we (silently) assume that all CQs $q(\mathbf{x}) = \exists \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})$ are *safe*—that is, such that for each $z \in \mathbf{x} \cup \mathbf{y}$, the class $\tilde{z}$ contains either an individual from $\mathbf{I}$ or a variable mentioned in an atom of $\phi(\mathbf{x}, \mathbf{y})$ that is not an equality. A CQ is *Boolean* if its answer variables $\mathbf{x}$ are the empty tuple $\langle\rangle$. Furthermore, following Bienvenu et al. [21], a CQ $q(\mathbf{x})$ is *rooted* if each connected component of its Gaifman graph has a node with a term in $\mathbf{x} \cup \mathbf{I}$. A *union* of CQs (*UCQ*) is a disjunction of CQs with the same answer variables. A UCQ is *Boolean* (or *rooted*, or both) if so are all of its component CQs.

The *answers* q^I to a (U)CQ $q(\mathbf{x})$ over an interpretation I are the set of all tuples $\mathbf{a}$ of individuals from $\mathbf{I}$ with $|\mathbf{a}| = |\mathbf{x}|$ such that the formula $q(\mathbf{a})$ holds in I (where $|\mathbf{a}|$ and $|\mathbf{x}|$ are the sizes of $\mathbf{a}$ and $\mathbf{x}$, respectively). The *certain answers* to a (U)CQ $q(\mathbf{x})$ over a *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{K}$ are the intersection of the answers to $q(\mathbf{x})$ over all models of $\mathcal{K}$. The *certain answers* $q^\mathcal{K}$ to $q(\mathbf{x})$ over $\mathcal{K}$ *under the UNA* are the intersection of the answers to $q(\mathbf{x})$ over all models of $\mathcal{K}$ satisfying the UNA. In fact, for *DL-Lite*$_\mathcal{R}$, the (usual) certain answers always coincide with the certain answers under UNA, and checking whether a tuple of individuals is in the certain answers to a (U)CQ q over a *DL-Lite*$_\mathcal{R}$ ontology $\langle \mathcal{T}, \mathcal{A} \rangle$ is an NP-complete problem with AC^0 data complexity (i.e., when $\mathcal{T}$ and q are fixed) [9,24]. The latter follows from the rewritability of the class of UCQs to itself over *DL-Lite*$_\mathcal{R}$ [9]. Informally, the key ideas for rewritability is as follows. First, every *DL-Lite*$_\mathcal{R}$ ontology possesses a so-called canonical interpretation, which is a model if the ontology is satisfiable, and which is homomorphically embeddable to every other model. Moreover, this model is universal for UCQs in the sense that the answers to every UCQ on the canonical interpretation coincide with the certain answers over the ontology. Finally, for every UCQ and every TBox it is always possible to construct another UCQ such that, for every ABox, the answers to the original UCQ over the universal model of the resulting ontology are the same as the answers to the new UCQ over just the ABox. We will formally define the notions of rewritability, canonical interpretation, and universal model later in the article.

To conclude, we note that our definition of CQs is slightly non-standard in that it allows for equality atoms, which are usually regarded as inessential. Making equalities explicit in the query will be convenient later on when computing query rewritings, where we will sometimes need to force an answer variable to become equal to another answer variable or to an individual. This is, however, just a technicality; in particular, none of our complexity lower bounds to query answering or negative rewritability results depend on the presence of equalities in the query.

2.3. 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 of non-negative integers $\mathbb{N}_0$ extended with infinity ∞. The value $\Omega(c)$ is the *multiplicity* of 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 satisfying $\emptyset(c) = 0$ for all $c \in M$. Given bags Ω_1 and

Ω_2 over M, we say that Ω_1 *is a subbag of* Ω_2, in symbols $\Omega_1 \subseteq \Omega_2$, if $\Omega_1(c) \leq \Omega_2(c)$ for each $c \in M$. Often, especially in examples, 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.

In this paper, we use the following common set of operators encountered in the literature on algebras over bags [22,23,25–27]. The *intersection* $\cap$, *maximal union* $\cup$, *arithmetic union* $\uplus$, and *difference* $-$ are the binary operators defined for bags Ω_1 and Ω_2 over the same set M, and for every $c \in M$ as follows:

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

Note that bag difference is well-defined only when Ω_2 does not assign ∞ to any element in M. Also, the unary *duplicate elimination* ε operator is defined for a bag Ω over a set M and for every $c \in M$ as follows:

$$\big(\varepsilon(\Omega)\big)(c) = \begin{cases} 1, & \text{if } \Omega(c) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Note that, for every two finite bags Ω_1 and Ω_2 over the same set M, the following identities hold [26]:

$$\Omega_1 \cup \Omega_2 = \Omega_1 \uplus (\Omega_2 - \Omega_1) \qquad \text{and} \qquad \Omega_1 \cap \Omega_2 = (\Omega_1 \uplus \Omega_2) - (\Omega_1 \cup \Omega_2). \tag{1}$$

However, these identities may not hold if the bags are infinite.

2.4. A calculus for querying bag databases

A *database schema* is a non-empty finite set $\mathbf{S}$ of predicates with non-negative arities that are disjoint from $\mathbf{C}$ and $\mathbf{R}$. Given a database schema $\mathbf{S}$ and a set of constants (i.e., individuals) $\mathbf{I}$, a database *fact* is an expression of the form $S(\mathbf{a})$, where $S \in \mathbf{S}$ and $\mathbf{a}$ is a tuple of constants from $\mathbf{I}$ of size equal to the arity of S. Then, a *bag database instance* $\mathcal{D}$ is a finite bag over all the facts over $\mathbf{S}$ and $\mathbf{I}$.

Grumbach and Milo [22] proposed an algebraic query language for bag databases called BALG, which is sufficiently powerful to capture relational algebra over bags [19]. BALG allows for nesting of bags and can be seen as the union of the sublanguages BALG^k, $k \geq 1$, each of which allowing for up to $k-1$ levels of nesting. In this article we restrict ourselves to BALG^1, which does not allow for bag nesting. Grumbach and Milo [22] studied the data complexity of answering BALG^1 queries under a unary encoding of numbers in the input and showed that it is strictly "sandwiched" between complexity classes AC^0 and LogSpace; it is therefore tractable, but strictly harder than the data complexity of relational queries over set databases. We defer a full treatment of Grumbach and Milo's algebra and its associated decision problem to Appendix A.

We next introduce BCALC—a calculus for querying bag databases based on Grumbach and Milo's algebra. Using a calculus formulation instead of an algebraic one will significantly simplify the presentation of our query rewriting algorithms later on. In Appendix B we show that our calculus can be easily embedded into BALG^1 and thus inherits its LogSpace upper bound for data complexity of query answering.

The syntax of BCALC for a database schema, formally presented in the following inductive definition, extends the syntax of (U)CQs given in Section 2.2 with several new operations. Domain-dependent queries, inexpressible in algebraic query languages, are precluded by introducing restrictions on the use of variables (intuitively, a query is *domain-dependent* if its answers over a fixed database instance may change when the underlying set of constants is modified; see [28] for details).

Definition 3. Given a database schema $\mathbf{S}$ and a set of constants $\mathbf{I}$, a BCALC *query* $\Phi(\mathbf{x})$ with *answer* variables $\mathbf{x}$ is any of the following, where Ψ, Ψ_1, and Ψ_2 are BCALC queries:

- $S(\mathbf{t})$, where $S \in \mathbf{S}$ and $\mathbf{t}$ is a tuple over $\mathbf{x} \cup \mathbf{I}$ of size equal to the arity of S mentioning all variables in $\mathbf{x}$;
- $\Psi_1(\mathbf{x}_1) \wedge \Psi_2(\mathbf{x}_2)$, 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 $\mathbf{x}$;
- $\Psi_1(\mathbf{x}) \,\mathrm{op}\, \Psi_2(\mathbf{x})$, where $\mathrm{op} \in \{\vee, \veebar, \setminus\}$; or
- $\delta\, \Psi(\mathbf{x})$.

A BCALC query is *positive* if it does not mention the difference operator $\setminus$. A positive BCALC query is a BCALC *conjunctive query* (*CQ*) if it additionally does not mention operators $\vee$, $\veebar$, and δ. A BCALC *maximal* (or *arithmetic*) *union of CQs* is a BCALC query of the form $\Phi_1(\mathbf{x}) \vee \cdots \vee \Phi_n(\mathbf{x})$ (or of the form $\Phi_1(\mathbf{x}) \veebar \cdots \veebar \Phi_n(\mathbf{x})$, respectively), where each Φ_i is a BCALC CQ.

Next we formally define the semantics of BCALC queries, which are bags of tuples of constants.

Definition 4. The *bag answers* $\Phi^{\mathcal{D}}$ to a BCALC query $\Phi(\mathbf{x})$ over a bag database instance $\mathcal{D}$ is the finite bag over $\mathbf{I}^{|\mathbf{x}|}$ defined inductively by the following equations 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 $a \in \mathbf{I}$:

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

The decision problem of query answering for BCALC is defined as follows, where all numbers in the input are assumed to be represented in unary and the bag (i.e., the database instance) is explicitly defined only for a finite number of facts while the multiplicities of all other facts are assumed to be 0.

QueryAnswering[BCALC]

Input: BCALC query $\Phi(\mathbf{x})$, bag database instance $\mathcal{D}$, tuple $\mathbf{a}$ of constants over $\mathbf{I}$, and number $k \in \mathbb{N}_0$.
Question: Is $\Phi^{\mathcal{D}}(\mathbf{a}) \geq k$?

The *data complexity* of this problem is the complexity when query Φ is considered to be fixed and only $\mathcal{D}$, $\mathbf{a}$, and k form the input.

The LogSpace upper bound for the data complexity of QueryAnswering[BCALC] is obtained by showing that for each BCALC query Φ one can construct a BALG^1 algebra expression E_Φ such that the bag answers to Φ over every bag database $\mathcal{D}$ coincide with the bag answers to E_Φ over $\mathcal{D}$; this serves the need, because, as we already mentioned, BALG^1 can be evaluated in LogSpace. The proof of this claim, which is rather technical but conceptually straightforward, is deferred to Appendix B.

Proposition 5. QueryAnswering[BCALC] *is in* LogSpace *in data complexity.*

We conclude by observing that in the literature on query optimisation under bag semantics [29–31] it is common to encounter the notion of *bag-set semantics* for databases, where input database instances are sets—that is, do not allow for multiplicities greater than 1—while permitting query answers and views to be bags. The bag semantics we consider in this article generalises the bag-set semantics, and restricting ourselves to bag-set semantics would not change any of our results.

3. Ontology-based data access under bag semantics

In this section, we introduce our OBDA framework as a natural generalisation of that by Poggi et al. for OBDA under set semantics [2]. We start by defining the syntax of OBDA settings.

Definition 6. A *bag OBDA setting* is a triple $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ where

- $\mathcal{T}$ is a *DL-Lite*$_{\mathcal{R}}$ TBox;
- Σ is a set of *global-as-view* (*GAV*) *mapping assertions* (or *mappings*) of the form

$$\Phi(x) \rightsquigarrow A(x) \quad \text{or} \quad \Psi(x, y) \rightsquigarrow P(x, y),$$

  where Φ and Ψ are BCALC queries, while A and P are an atomic concept and atomic role, respectively; and
- $\mathcal{D}$ is a bag database instance.

A couple of observations about Definition 6 are in order. First, recall that in all our motivating examples so far we have written mappings using SQL queries, which reflects the way in which mappings are defined in practice. To formally study OBDA, however, the use of a bag query language close to first-order logic, such as BCALC, is more appropriate. Second, in contrast to the definitions by Poggi et al. [2], we do not allow for function symbols on the right-hand side of mappings.

This restriction does not affect the computational properties of query answering in OBDA under set semantics [2], and it is adopted in most theoretical papers on data integration [8,32]; it is also immaterial to our technical results, and yet allows us to simplify the presentation.

Example 7. The mappings Σ_{ex} in Example 1 are equivalently expressed using BCALC as follows:

$$\begin{aligned}
\sigma_1 &= \exists y_1, y_2, y_3, y_4.\, \mathsf{Columbia}(x, y_1, y_2, y_3, y_4) \quad \rightsquigarrow \quad \mathsf{Musician}(x),\\
\sigma_2 &= \exists y_1, y_2.\, \mathsf{Verve_Wind}(x, y_1, y_2) \quad \rightsquigarrow \quad \mathsf{WindPlayer}(x),\\
\sigma_3 &= \exists y_1, y_2, y_3, y_4.\, \mathsf{Columbia}(y_1, x, y_2, y_3, y_4) \quad \rightsquigarrow \quad \mathsf{Record}(x),\\
\sigma_4 &= \exists y_1, y_2.\, \mathsf{Verve_Wind}(y_1, x, y_2) \quad \rightsquigarrow \quad \mathsf{Record}(x),\\
\sigma_5 &= \exists y_1, y_2, y_3.\, \mathsf{Columbia}(y, x, y_1, y_2, y_3) \quad \rightsquigarrow \quad \mathsf{hasMusician}(x, y),\\
\sigma_6 &= \exists y_1.\, \mathsf{Verve_Wind}(y, x, y_1) \quad \rightsquigarrow \quad \mathsf{hasMusician}(x, y). \qquad \triangleleft
\end{aligned}$$

The semantics of bag OBDA settings is based on *bag interpretations* I, which are defined as set interpretations (see Section 2.1) with the exception that concepts and roles are now interpreted as bags rather than sets. The extension of the interpretation function to non-atomic concepts and roles is defined in a natural way: for example, the concept $\exists P$ for an atomic role P is interpreted as the bag projection of the interpretation P^I of P to its first component, where each occurrence of a pair (u, v) in P^I contributes separately to the multiplicity of a domain element u in $(\exists P)^I$.

Definition 8. A *bag interpretation* I is a pair $\langle \Delta^I, \cdot^I \rangle$ where the *domain* Δ^I is a non-empty set, and the *interpretation function* $\cdot^I$ maps each individual $a \in \mathbf{I}$ to an element $a^I \in \Delta^I$, each atomic concept $A \in \mathbf{C}$ to a bag A^I over Δ^I, and each atomic role $P \in \mathbf{R}$ to a bag P^I over $\Delta^I \times \Delta^I$. Interpretation function $\cdot^I$ extends to complex concepts $(P^-)^I$ and $(\exists R)^I$, for $P \in \mathbf{R}$ and R a role, as follows, for all $u, v \in \Delta^I$:

$$(P^-)^I(u, v) = P^I(v, u) \qquad \text{and} \qquad (\exists R)^I(u) = \sum_{v \in \Delta^I} R^I(u, v).$$

A bag interpretation I is *finite* if Δ^I is a finite set and I assigns a finite bag to each $A \in \mathbf{C}$ and each $P \in \mathbf{R}$.

We are now ready to specify the semantics of bag OBDA settings in terms of bag interpretations (note that satisfaction of axioms is defined in the same way as in the set case, but the symbols $\subseteq$, $\cap$, and $\emptyset$ denote the subbag relation, bag intersection, and the empty bag, respectively).

Definition 9. A bag interpretation I *satisfies* the *UNA* if $a^I \neq b^I$ whenever $a, b \in \mathbf{I}$ with $a \neq b$. A bag interpretation I *satisfies* an inclusion $S_1 \sqsubseteq S_2$ if $S_1^I \subseteq S_2^I$, and it *satisfies* a disjointness axiom $\mathsf{Disj}(S_1, S_2)$ if $S_1^I \cap S_2^I = \emptyset$. A bag interpretation I *satisfies* a TBox $\mathcal{T}$, written $I \models^{\mathsf{b}} \mathcal{T}$, if I satisfies every axiom in $\mathcal{T}$. A TBox $\mathcal{T}$ *entails* an axiom α *under bag semantics*, written $\mathcal{T} \models^{\mathsf{b}} \alpha$, if every bag interpretation satisfying $\mathcal{T}$ satisfies α; $\mathcal{T}$ *entails* α *under bag semantics and the UNA* if only bag interpretations satisfying the UNA are considered.

A bag interpretation I *satisfies* a set of mappings Σ *with respect to* a bag database instance $\mathcal{D}$ if the following holds, for all mappings $\Phi(x) \rightsquigarrow A(x)$ and $\Psi(x, y) \rightsquigarrow P(x, y)$ in Σ, and all individuals $a, a_1, a_2 \in \mathbf{I}$:

$$\sum_{b \in \mathbf{I}:\ b^I = a^I} \Phi^{\mathcal{D}}(b) \leq A^I(a^I) \qquad \text{and} \qquad \sum_{b_i \in \mathbf{I}:\ b_i^I = a_i^I,\ i=1,2} \Psi^{\mathcal{D}}(b_1, b_2) \leq P^I(a_1^I, a_2^I).$$

A bag interpretation I is a *model* of a bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$, denoted by $I \models^{\mathsf{b}} \langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$, if $I \models^{\mathsf{b}} \mathcal{T}$ and I satisfies Σ with respect to $\mathcal{D}$. An OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ is *satisfiable* if it has a model, and it is *satisfiable under the UNA* if it has a model satisfying the UNA.

Note that if a bag interpretation I satisfies the UNA, then the notion of satisfaction for a set of mappings Σ as defined above becomes equivalent to requiring that inequalities $\Phi^{\mathcal{D}}(a) \leq A^I(a^I)$ and $\Psi^{\mathcal{D}}(a_1, a_2) \leq P^I(a_1^I, a_2^I)$ hold for all mappings $\Phi(x) \rightsquigarrow A(x)$ and $\Psi(x, y) \rightsquigarrow P(x, y)$ in Σ, and all individuals $a, a_1, a_2 \in \mathbf{I}$.

Example 10. Consider the bag OBDA setting $\langle \mathcal{T}_{ex}, \Sigma_{ex}, \mathcal{D}_{ex} \rangle$ in Example 1. Let I_{ex} be the bag interpretation with domain $\Delta^{I_{ex}} = \mathbf{I}$ that maps all individuals to themselves, and assigns bags to concepts and roles as follows:

$$\begin{aligned}
\mathsf{Musician}^{I_{ex}} &= \{\!|\, \textit{M. Davis} : 2,\ \textit{K. Jarrett} : 1 \,|\!\},\\
\mathsf{WindPlayer}^{I_{ex}} &= \{\!|\, \textit{M. Davis} : 1 \,|\!\},\\
\mathsf{Record}^{I_{ex}} &= \{\!|\, \textit{Kind of Blue} : 1,\ \textit{A Tribute to Jack Johnson} : 1,\ \textit{Expectations} : 1,\\
&\qquad \textit{Ascenseur pour l'Échafaud} : 1 \,|\!\},\\
\mathsf{hasMusician}^{I_{ex}} &= \{\!|\, (\textit{Kind of Blue},\ \textit{M. Davis}) : 1,\ (\textit{A Tribute to Jack Johnson},\ \textit{M. Davis}) : 1,\\
&\qquad (\textit{Expectations},\ \textit{K. Jarrett}) : 1,\ (\textit{Ascenseur pour l'Échafaud},\ \textit{M. Davis}) : 1 \,|\!\}.
\end{aligned}$$

To show that $I_{ex} \models^{\mathrm{b}} \langle \mathcal{T}_{ex}, \Sigma_{ex}, \mathcal{D}_{ex} \rangle$, we argue that I_{ex} satisfies Σ_{ex} with respect to $\mathcal{D}_{ex}$ and that $I_{ex} \models^{\mathrm{b}} \mathcal{T}_{ex}$. For the former, we first compute the bag answers to the BCALC queries $\Phi_1, \ldots, \Phi_6$ appearing on the left-hand side of the mappings $\sigma_1, \ldots, \sigma_6$ over database instance $\mathcal{D}_{ex}$. These are specified as follows:

$$\begin{aligned}
\Phi_1^{\mathcal{D}_{ex}} &= \{\!|\, \textit{M. Davis} : 2,\ \textit{K. Jarrett} : 1 \,|\!\},\\
\Phi_2^{\mathcal{D}_{ex}} &= \{\!|\, \textit{M. Davis} : 1 \,|\!\},\\
\Phi_3^{\mathcal{D}_{ex}} &= \{\!|\, \textit{Kind of Blue} : 1,\ \textit{A Tribute to Jack Johnson} : 1,\ \textit{Expectations} : 1 \,|\!\},\\
\Phi_4^{\mathcal{D}_{ex}} &= \{\!|\, \textit{Ascenseur pour l'Échafaud} : 1 \,|\!\},\\
\Phi_5^{\mathcal{D}_{ex}} &= \{\!|\, (\textit{Kind of Blue},\ \textit{M. Davis}) : 1,\ (\textit{A Tribute to Jack Johnson},\ \textit{M. Davis}) : 1,\ (\textit{Expectations},\ \textit{K. Jarrett}) : 1 \,|\!\},\\
\Phi_6^{\mathcal{D}_{ex}} &= \{\!|\, (\textit{Ascenseur pour l'Échafaud},\ \textit{M. Davis}) : 1 \,|\!\}.
\end{aligned}$$

It is now immediate to verify that the inequalities stipulated by Definition 9 hold for $\sigma_1, \ldots, \sigma_6$; thus, I_{ex} satisfies Σ_{ex} with respect to $\mathcal{D}_{ex}$. We next argue that $I_{ex} \models^{\mathrm{b}} \mathcal{T}_{ex}$. Indeed, by Definition 8, the interpretation of $\exists\mathsf{hasMusician}$ is the bag

$$\{\!|\, \textit{Kind of Blue} : 1,\ \textit{A Tribute to Jack Johnson} : 1,\ \textit{Expectations} : 1,\ \textit{Ascenseur pour l'Échafaud} : 1 \,|\!\},$$

so $C^{I_{ex}}$ is a subbag of $D^{I_{ex}}$ for each inclusion $C \sqsubseteq D$ in $\mathcal{T}_{ex}$, as required. ◁

We next discuss an important aspect of Definition 9 concerning the presence of different mappings defining the same view. In such cases, the extension of the view intuitively corresponds to the union of the answers to the queries specified on the left-hand side of the contributing mappings. However, bag query languages, such as BCALC, come with two versions of the union operation: maximal and arithmetic union. Moreover, in different settings one of these unions can be more intuitive and preferable than the other. On the one hand, Definition 9 tacitly commits to the maximal union by requiring that a model for Σ and $\mathcal{D}$ satisfies each contributing mapping independently. On the other hand, this is not a limitation of our OBDA framework since GAV mapping assertions can always be rewritten to reflect the alternative choice based on arithmetic union.

Example 11. Consider the bag OBDA setting $\langle \mathcal{T}_{ex}, \Sigma'_{ex}, \mathcal{D}_{ex} \rangle$ obtained from our running example by augmenting Σ_{ex} to the set Σ'_{ex} that additionally contains mapping

$$\sigma_7 = \quad \exists y_1, y_2.\, \mathsf{Verve\_Wind}(x, y_1, y_2) \quad \rightsquigarrow \quad \mathsf{Musician}(x).$$

Note that both mappings σ_1 and σ_7 define the extension of concept Musician. By Definition 9 interpretations such as I_{ex} in Example 10, which interpret Musician as the maximal union of the bags corresponding to the musicians mentioned in the Columbia and Verve_Wind tables, are valid models of $\langle \mathcal{T}_{ex}, \Sigma'_{ex}, \mathcal{D}_{ex} \rangle$. Let us now define Σ''_{ex} as the mappings obtained from Σ_{ex} by replacing σ_1 by

$$\big(\exists y_1, y_2, y_3, y_4.\, \mathsf{Columbia}(x, y_1, y_2, y_3, y_4)\big) \vee \big(\exists y_1, y_2.\, \mathsf{Verve\_Wind}(x, y_1, y_2)\big) \quad \rightsquigarrow \quad \mathsf{Musician}(x).$$

In this case, every model of $\langle \mathcal{T}_{ex}, \Sigma''_{ex}, \mathcal{D}_{ex} \rangle$ interprets Musician as the arithmetic union of the bags corresponding to musicians in the relevant tables. In particular, interpretation I_{ex} in Example 10 is not a model, as required. ◁

We are now ready to define CQ answering under bag semantics. We first define the answers q^I to a CQ $q(\mathbf{x})$ over a bag interpretation I; this is a natural extension to (possibly infinite) interpretations of the notion of bag answers to a CQ over a bag database (see Section 2.4). Specifically, q^I is a bag of tuples of individuals such that each valid embedding λ of the atoms in q into I contributes separately to the multiplicity of the tuple $\lambda(\mathbf{x})$ in q^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 I.

Definition 12. Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a CQ and $I = \langle \Delta^I, \cdot^I \rangle$ be a bag interpretation. The *bag answers* q^I to q over I are the bag over tuples of individuals from $\mathbf{I}$ of size $|\mathbf{x}|$ such that, for every such tuple $\mathbf{a}$,

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

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

If q is Boolean then the bag answers q^I are defined only for the empty tuple $\langle\rangle$. Also, conjunction $\phi(\mathbf{x}, \mathbf{y})$ 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 [29,33], and thus the definition of $q^I(\mathbf{a})$ should be read in the way that it treats each copy of a query atom $S(\mathbf{t})$ separately in the product.

The following definition of certain answers, which captures open-world query answering, is a natural extension of the set notion to bags: a query answer is certain with multiplicity k if it is an answer with multiplicity at least k over every model of the OBDA setting.

Definition 13. The *bag certain answers* $q^{\langle \mathcal{T}, \Sigma, \mathcal{D}\rangle}$ to a CQ q over a bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D}\rangle$ are the bag

$$\bigcap_{I \models^{\mathrm{b}} \langle \mathcal{T}, \Sigma, \mathcal{D}\rangle} q^I.$$

The *bag certain answers under the UNA* are defined in the same way except that the intersection ranges only over the models satisfying the UNA.

Example 14. Consider the OBDA setting $\langle \mathcal{T}_{ex}, \Sigma_{ex}, \mathcal{D}_{ex}\rangle$ of our running example. Let $\mathcal{T}'_{ex}$ augment $\mathcal{T}_{ex}$ with the additional inclusion $\exists \mathsf{hasMusician}^- \sqsubseteq \mathsf{Musician}$ specifying the range of role $\mathsf{hasMusician}$, and let $q_{ex}(x) = \mathsf{Musician}(x)$. We argue that $q_{ex}^{\langle \mathcal{T}'_{ex}, \Sigma_{ex}, \mathcal{D}_{ex}\rangle}(\mathit{M.\ Davis}) = 3$. On the one hand, consider interpretation J_{ex} extending I_{ex} in Example 10 by setting $\mathsf{Musician}^{J_{ex}} = \{\!|\, \mathit{M.\ Davis} : 3,\ \mathit{K.\ Jarrett} : 1 \,|\!\}$; it can be easily checked that J_{ex} is a model of $\langle \mathcal{T}'_{ex}, \Sigma_{ex}, \mathcal{D}_{ex}\rangle$ satisfying $q_{ex}^{J_{ex}}(\mathit{M.\ Davis}) = 3$. On the other hand, we argue that every bag model I of $\langle \mathcal{T}'_{ex}, \Sigma_{ex}, \mathcal{D}_{ex}\rangle$ satisfies $q_{ex}^{I}(\mathit{M.\ Davis}) \geq 3$. The fact that I satisfies Σ_{ex} with respect to $\mathcal{D}_{ex}$ (and, in particular, mappings σ_5 and σ_6) implies that $\mathsf{hasMusician}^I$ associates with *M. Davis* at least three elements. So, $(\exists \mathsf{hasMusician}^-)^I(\mathit{M.\ Davis}) \geq 3$. But then, since $\mathcal{T}'_{ex}$ contains $\exists \mathsf{hasMusician}^- \sqsubseteq \mathsf{Musician}$ and I satisfies $\mathcal{T}'_{ex}$, we have that $\mathsf{Musician}^I(\mathit{M.\ Davis}) \geq 3$. Therefore, $q_{ex}^{\langle \mathcal{T}'_{ex}, \Sigma_{ex}, \mathcal{D}_{ex}\rangle}(\mathit{M.\ Davis}) = 3$ holds as well. ◁

The decision problem BagCertObda$[\mathcal{Q}, \mathcal{O}]$ corresponding to computing the bag certain answers to a CQ from a class $\mathcal{Q}$ over an OBDA setting with a TBox from an ontology language $\mathcal{O}$ (i.e., *DL-Lite*$_{\mathcal{R}}$ or one of its sublanguages) is defined as follows, where we again assume that all numbers in the input are represented in unary and the bag is explicitly defined only for a finite number of facts.

BagCertObda$[\mathcal{Q}, \mathcal{O}]$

Input: CQ q from $\mathcal{Q}$, bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D}\rangle$ with $\mathcal{T}$ from $\mathcal{O}$, tuple $\mathbf{a}$ of individuals from $\mathbf{I}$, and number $k \in \mathbb{N}_0^\infty$.

Question: Is $q^{\langle \mathcal{T}, \Sigma, \mathcal{D}\rangle}(\mathbf{a}) \geq k$?

The UNA version BagCertObda$^{\mathrm{UNA}}[\mathcal{Q}, \mathcal{O}]$ of this problem is defined in the same way as BagCertObda$[\mathcal{Q}, \mathcal{O}]$ except that the certain answers are considered under the UNA. The *data complexity* of these problems is the complexity when the query q, TBox $\mathcal{T}$, and mappings Σ are considered to be fixed, and only $\mathcal{D}$, $\mathbf{a}$, and k form the input.

4. The ontology language *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$

In Section 4.1 we define the ontology language *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ and its natural fragments, where the distinctive feature of *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ is that ABoxes consist of bags of facts rather than sets. We then show in Section 4.2 that, analogously to the case of conventional OBDA, the materialisation of the mappings over the sources can be represented by a virtual bag ABox; as a result, the data complexity of OBDA query answering under bag semantics coincides with that of query answering over *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$.

4.1. The syntax and semantics of *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$

We start by introducing the notion of a bag ABox and describing its semantics in terms of bag interpretations.

Definition 15. A *bag ABox* is a finite bag over the set of concept and role assertions. A bag interpretation $I = \langle \Delta^I, \cdot^I\rangle$ *satisfies* a bag ABox $\mathcal{A}$, written $I \models^{\mathrm{b}} \mathcal{A}$, if, for each concept assertion $A(a)$ and role assertion $P(a_1, a_2)$, the following holds:

$$\sum_{b \in \mathbf{I}:\ b^I = a^I} \mathcal{A}(A(b)) \leq A^I(a^I) \quad \text{and} \quad \sum_{b_i \in \mathbf{I}:\ b_i^I = a_i^I,\ i=1,2} \mathcal{A}(P(b_1, b_2)) \leq P^I(a_1^I, a_2^I).$$

If a bag interpretation I satisfies the UNA, then ABox satisfaction amounts to checking whether the inequalities $\mathcal{A}(A(a)) \leq A^I(a^I)$ and $\mathcal{A}(P(a_1, a_2)) \leq P^I(a_1^I, a_2^I)$ hold for each concept and role assertion $A(a)$ and $P(a_1, a_2)$, respectively. We can now introduce the notion of a bag ontology and define the ontology language *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ and its fragments.

Definition 16. A *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ *ontology* is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$ of a *DL-Lite*$_{\mathcal{R}}$ TBox $\mathcal{T}$ and bag ABox $\mathcal{A}$. The sublanguages *DL-Lite*$^{\mathrm{b}}_{\mathrm{CORE}}$, *DL-Lite*$^{\mathrm{b}}_{\mathrm{RDFS}}$, and *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}^-}$ of *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ are defined in the same way except that only *DL-Lite*$_{\mathrm{CORE}}$, *DL-Lite*$_{\mathrm{RDFS}}$, and *DL-Lite*$_{\mathcal{R}^-}$ TBoxes are allowed, respectively.

A bag interpretation I is a *model* of a *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontology $\langle \mathcal{T}, \mathcal{A} \rangle$, written $I \models^{\mathrm{b}} \langle \mathcal{T}, \mathcal{A} \rangle$, if $I \models^{\mathrm{b}} \mathcal{T}$ and $I \models^{\mathrm{b}} \mathcal{A}$. A *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontology is *satisfiable* if it has a model; it is *satisfiable under the UNA* if it has a model satisfying the UNA.

The following definition of certain answers, which captures open-world query answering, is a natural extension of the set notion for *DL-Lite*$_{\mathcal{R}}$ to bags: a query answer is certain for a given multiplicity if it occurs with at least that multiplicity in the bag answers to the query over every model of the ontology.

Definition 17. The *bag certain answers* $q^{\mathcal{K}}$ to a CQ q over a *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontology $\mathcal{K}$ are the bag $\bigcap_{I \models^{\mathrm{b}} \mathcal{K}} q^I$. The *bag certain answers under the UNA* are defined in the same way except that only models satisfying the UNA are considered in the intersection.

Similarly to the OBDA case, the decision problem corresponding to computing the bag certain answers to a CQ from a class $\mathcal{Q}$ over an ontology in a bag ontology language $\mathcal{O}$ (e.g., *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ or one of its sublanguages) is defined as follows, where we again assume that all numbers in the input are represented in unary and the bag is explicitly defined only for a finite number of facts.

BagCert$[\mathcal{Q}, \mathcal{O}]$

Input: CQ q from $\mathcal{Q}$, ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ from $\mathcal{O}$, tuple $\mathbf{a}$ of individuals from $\mathbf{I}$, and number $k \in \mathbb{N}_0^\infty$.

Question: Is $q^{\mathcal{K}}(\mathbf{a}) \geq k$?

The UNA version BagCert$^{\mathrm{UNA}}[\mathcal{Q}, \mathcal{O}]$ of this problem is defined in the same way as BagCert$[\mathcal{Q}, \mathcal{O}]$ except that the certain answers are considered under the UNA. The *data complexity* of this problem is the complexity when the query q and the TBox $\mathcal{T}$ are considered to be fixed, and only $\mathcal{A}$, $\mathbf{a}$, and k form the input.

We conclude this section by introducing the notion of rewritability for our bag semantics. Our definition is analogous to that of Calvanese et al. [9] for the set case. Note that a bag ABox $\mathcal{A}$ can be seen as a database instance, so we can write $\Phi^{\mathcal{A}}$ for a BCALC query Φ over the unary and binary predicates for atomic concepts and roles, respectively.

Definition 18. A BCALC query Φ is a *rewriting* of a CQ q with respect to a TBox $\mathcal{T}$ if all the individuals, atomic concepts and atomic roles of Φ appear in q or $\mathcal{T}$, and $q^{\langle \mathcal{T}, \mathcal{A} \rangle} = \Phi^{\mathcal{A}}$ for every bag ABox $\mathcal{A}$ with satisfiable $\langle \mathcal{T}, \mathcal{A} \rangle$. A class of CQs $\mathcal{Q}$ is *rewritable* to a class of BCALC queries $\mathcal{Q}'$ over an ontology language $\mathcal{O}$ if, for every query in $\mathcal{Q}$ and every TBox in $\mathcal{O}$, there exists in $\mathcal{Q}'$ a rewriting of the query with respect to the TBox. *Rewritings* and *rewritability under the UNA* are defined in the same way except that only ontologies satisfiable under the UNA and certain answers under the UNA are considered.

Note that by restricting the signature of Φ to that of q and $\mathcal{T}$ in this definition we are considering the problem of finding a *pure* rewriting of q with respect to $\mathcal{T}$. In the set case, it was shown that pure rewritings may have to be of larger size than their *impure* counterparts [13]. However, our negative results on rewritability (i.e., Propositions 45 and 63) do not depend on this restriction, and may be shown for the more general case; we impose this restriction to facilitate exposition of some proofs.

Note also that in the set case the target query language for rewriting is typically a class of UCQs, since Calvanese et al. [9] showed that arbitrary CQs are rewritable to UCQs over *DL-Lite*$_{\mathcal{R}}$. In the case of bags, however, we will see that the situation is markedly different, and we will focus on BCALC as the target language for rewriting.

4.2. Relationship to query answering in OBDA

In conventional OBDA, the certain answers to a query over $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ can be characterised as those that logically follow from the union of the TBox $\mathcal{T}$ and the *virtual ABox* $\mathcal{A}_{\Sigma, \mathcal{D}}$, which represents the materialisation of the views defined by the mappings Σ over the database instance $\mathcal{D}$. As a result, query answering in OBDA amounts to query answering over *DL-Lite*$_{\mathcal{R}}$, and the rewritability and data complexity properties of both problems coincide [2].

In what follows, we show that an analogous correspondence holds under bag semantics. We start by introducing the notion of a *virtual bag ABox*, which captures the materialisation of the views specified in a bag OBDA setting.

Definition 19. The *virtual DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ABox of the bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ is the bag ABox $\mathcal{A}_{\Sigma,\mathcal{D}}$ defined as follows, for all $A \in \mathbf{C}$, $P \in \mathbf{R}$, and $a, a_1, a_2 \in \mathbf{I}$:

$$\mathcal{A}_{\Sigma,\mathcal{D}}(A(a)) = \max\left\{\Phi^{\mathcal{D}}(a) \mid \Phi(x) \rightsquigarrow A(x) \in \Sigma\right\},$$
$$\mathcal{A}_{\Sigma,\mathcal{D}}(P(a_1, a_2)) = \max\left\{\Psi^{\mathcal{D}}(a_1, a_2) \mid \Psi(x, y) \rightsquigarrow P(x, y) \in \Sigma\right\}.$$

The following example illustrates the notion of virtual ABoxes.

Example 20. The virtual ABox $\mathcal{A}_{ex}$ of the bag OBDA setting $\langle \mathcal{T}_{ex}, \Sigma_{ex}, \mathcal{D}_{ex} \rangle$ where $\mathcal{T}_{ex}$ is specified in Example 1 and $\Sigma_{ex}, \mathcal{D}_{ex}$ in Example 7 is the following bag of assertions:

$$\{\!\!\{ \mathsf{Musician}(\textit{M. Davis}) : 2,\ \mathsf{Musician}(\textit{K. Jarrett}) : 1,\ \mathsf{WindPlayer}(\textit{M. Davis}) : 1,$$
$$\mathsf{Record}(\textit{Kind of Blue}) : 1,\ \mathsf{Record}(\textit{A Tribute to Jack Johnson}) : 1,$$
$$\mathsf{Record}(\textit{Expectations}) : 1,\ \mathsf{Record}(\textit{Ascenseur pour l'Échafaud}) : 1,$$
$$\mathsf{hasMusician}(\textit{Kind of Blue},\ \textit{M. Davis}) : 1,\ \mathsf{hasMusician}(\textit{A Tribute to Jack Johnson},\ \textit{M. Davis}) : 1,$$
$$\mathsf{hasMusician}(\textit{Expectations},\ \textit{K. Jarrett}) : 1,\ \mathsf{hasMusician}(\textit{Ascenseur pour l'Échafaud},\ \textit{M. Davis}) : 1 \}\!\!\}. \quad \triangleleft$$

The following lemma shows that the models of a bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ coincide with those models satisfying the TBox $\mathcal{T}$ and the virtual bag ABox $\mathcal{A}_{\Sigma,\mathcal{D}}$.

Lemma 21. *For every bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ and every bag interpretation I, we have that $I \models^{b} \langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ if and only if $I \models^{b} \langle \mathcal{T}, \mathcal{A}_{\Sigma,\mathcal{D}} \rangle$.*

Proof. It suffices to show that I is a model of $\mathcal{A}_{\Sigma,\mathcal{D}}$ if and only if I satisfies Σ with respect to $\mathcal{D}$ as in Definition 9. For this, let I be a model of $\mathcal{A}_{\Sigma,\mathcal{D}}$ and, for all $S \in \mathbf{C} \cup \mathbf{R}$ and tuples $\mathbf{a}$ of individuals, let $\Psi_{S,\mathbf{a}}(\mathbf{x}) \rightsquigarrow S(\mathbf{x})$ be the mapping in Σ such that $\Psi^{\mathcal{D}}_{S,\mathbf{a}}(\mathbf{a}) = \max\left\{\Phi^{\mathcal{D}}(\mathbf{a}) \mid \Phi(\mathbf{x}) \rightsquigarrow S(\mathbf{x}) \in \Sigma\right\}$ where $\mathbf{x}$ has the same arity as $\mathbf{a}$. By the definition of $I \models^{\mathrm{b}} \mathcal{A}_{\Sigma,\mathcal{D}}$ and Definition 19, the following inequality holds for every $S \in \mathbf{C} \cup \mathbf{R}$, tuples $\mathbf{a}$ of individuals, and mapping $\Phi(\mathbf{x}) \rightsquigarrow S(\mathbf{x})$ in Σ:

$$S^I(\mathbf{a}^I) \;\geq\; \sum_{\mathbf{b} \text{ over } \mathbf{I}:\ \mathbf{b}^I = \mathbf{a}^I} \mathcal{A}_{\Sigma,D}(S(\mathbf{b})) \;=\; \sum_{\mathbf{b} \text{ over } \mathbf{I}:\ \mathbf{b}^I = \mathbf{a}^I} \Psi^{\mathcal{D}}_{S,\mathbf{b}}(\mathbf{b}) \;\geq\; \sum_{\mathbf{b} \text{ over } \mathbf{I}:\ \mathbf{b}^I = \mathbf{a}^I} \Phi^{\mathcal{D}}(\mathbf{b}).$$

By Definition 9, this is equivalent to requiring that I satisfies Σ with respect to $\mathcal{D}$, as desired. □

Having Lemma 21 at our disposal, we can relate the problems of satisfiability checking and query answering for bag OBDA settings to the corresponding problems for *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$.

Theorem 22. *The following statements hold:*

1. *a bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ is satisfiable if and only if $\langle \mathcal{T}, \mathcal{A}_{\Sigma,\mathcal{D}} \rangle$ is satisfiable;*
2. *for every bag OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ and every CQ q we have $q^{\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle} = q^{\langle \mathcal{T}, \mathcal{A}_{\Sigma,\mathcal{D}} \rangle}$; and*
3. *BagCertObda$[\mathcal{Q}, \mathcal{O}]$ and BagCert$[\mathcal{Q}, \mathcal{O}]$ are mutually reducible in LogSpace with respect to data complexity, for $\mathcal{Q}$ a class of CQs and for $\mathcal{O}$ a sublanguage of DL-Lite$^{\mathrm{b}}_{\mathcal{R}}$ or DL-Lite$^{\mathrm{b}}_{\mathcal{R}}$ itself.*

All three statements also hold when the problems are considered under the UNA.

Proof. We concentrate on the general case; the case of the UNA is analogous. The first two statements are direct consequences of Lemma 21 and the definitions of satisfiability and certain answers.

To show the third statement, we start by reducing BagCertObda$[\mathcal{Q}, \mathcal{O}]$ to BagCert$[\mathcal{Q}, \mathcal{O}]$. For this, fix an instance of BagCertObda$[\mathcal{Q}, \mathcal{O}]$ consisting of a CQ $q \in \mathcal{Q}$, an OBDA setting $\langle \mathcal{T}, \Sigma, \mathcal{D} \rangle$ with TBox $\mathcal{T}$ in $\mathcal{O}$, a tuple $\mathbf{a}$ of individuals, and a number k. By Statement 2, BagCertObda$[\mathcal{Q}, \mathcal{O}]$ is true on the aforementioned instance if and only if BagCert$[\mathcal{Q}, \mathcal{O}]$ is true on an instance consisting of q, $\langle \mathcal{T}, \mathcal{A}_{\Sigma,\mathcal{D}} \rangle$, $\mathbf{a}$, and k, where $\mathcal{A}_{\Sigma,\mathcal{D}}$ is the virtual ABox corresponding to Σ and $\mathcal{D}$. Because $\mathcal{T}$ and Σ are of fixed size, it is clear by Definition 19 and Proposition 5 on the data complexity of BCALC that the construction of $\mathcal{A}_{\Sigma,\mathcal{D}}$ is feasible in LogSpace.

The reduction of BagCert$[\mathcal{Q}, \mathcal{O}]$ to BagCertObda$[\mathcal{Q}, \mathcal{O}]$ is straightforward: for an instance of BagCert$[\mathcal{Q}, \mathcal{O}]$ consisting of a CQ $q \in \mathcal{Q}$, an ontology $\langle \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{O}$, a tuple $\mathbf{a}$ of individuals, and a number k, we just consider an instance of BagCertObda$[\mathcal{Q}, \mathcal{O}]$ consisting of q, $\langle \mathcal{T}, \Sigma, \mathcal{A} \rangle$, $\mathbf{a}$, and k, where $\mathcal{A}$ is considered as a bag database instance and Σ is a set of identity mappings $S(\mathbf{x}) \rightsquigarrow S(\mathbf{x})$ for all atomic concepts and roles S appearing in $\mathcal{A}$. □

This theorem allows us to talk only about *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontologies in the rest of the paper, silently assuming that all the results apply to OBDA settings as well.

5. Relationship of bag and set semantics in the context of *DL-Lite*$_{\mathcal{R}}$

In this section we discuss the main similarities and differences between our bag semantics of *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ and the conventional set semantics of *DL-Lite*$_{\mathcal{R}}$. First, in Section 5.1, we argue that our bag semantics can be seen as a generalisation of the set semantics, in the sense that, on the one hand, satisfiability checking in *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ reduces to satisfiability checking in *DL-Lite*$_{\mathcal{R}}$ and, on the other hand, query answers under bag and set semantics coincide if multiplicities are ignored. There are, however, key properties of the conventional semantics for *DL-Lite*$_{\mathcal{R}}$ that are no longer satisfied by the bag semantics. In particular, in Section 5.2, we discuss the influence of the UNA on query answering and show fundamental differences with the set case. Furthermore, in Section 5.3, we show that a universal model—a representative model of a satisfiable ontology over which each CQ can be correctly evaluated—may not exist under bag semantics; this is in contrast to the set case, where the fact that a universal model always exists is key to ensuring favourable computational properties of query answering.

5.1. Satisfiability, entailment of axioms, and query answering

The following theorem shows that our bag semantics is compatible with the conventional set semantics of *DL-Lite*$_{\mathcal{R}}$. The first statement in the theorem shows that satisfiability under bag semantics reduces to the set case: to check whether a *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontology $\mathcal{K}'$ is satisfiable, it suffices to check satisfiability of the *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{K}$ obtained from $\mathcal{K}'$ by setting all non-zero multiplicities in the ABox to 1. The second statement establishes that entailment of axioms under set and bag semantics coincide; this means that the adoption of bag semantics does not affect the standard TBox reasoning services implemented in ontology development tools. Finally, the third statement shows that certain answers under bag and set semantics coincide if multiplicities are ignored—that is, a tuple is a set certain answer to a query with respect to an ontology if and only if it is also a bag certain answer with multiplicity at least one. All three statements in the theorem hold regardless of whether the UNA is adopted.

Theorem 23. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ ontology and let $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ be a DL-Lite$^{\mathrm{b}}_{\mathcal{R}}$ ontology with the same TBox such that $\mathcal{A} = \{S(\mathbf{t}) \mid \mathcal{A}'(S(\mathbf{t})) \geq 1\}$. Then, the following statements hold:*

1. *$\mathcal{K}$ is satisfiable if and only if $\mathcal{K}'$ is satisfiable;*
2. *$\mathcal{T} \models \alpha$ if and only if $\mathcal{T} \models^{\mathrm{b}} \alpha$, for each DL-Lite$_{\mathcal{R}}$ axiom α; and*
3. *$\mathbf{a} \in q^{\mathcal{K}}$ if and only if $q^{\mathcal{K}'}(\mathbf{a}) \geq 1$, for each CQ q and each tuple $\mathbf{a}$ over $\mathbf{I}$.*

All three statements also hold when the problems are considered under the UNA.

Proof. We concentrate on the general case; the case of the UNA is analogous.
(*Statement 1*) Assume that $\mathcal{K}$ has a model $I = \langle \Delta^I, \cdot^I \rangle$. Let $I' = \langle \Delta^I, \cdot^{I'} \rangle$ be the bag interpretation defined as follows, for each $a \in \mathbf{I}$, $A \in \mathbf{C}$, $P \in \mathbf{R}$, and $u, v \in \Delta^I$:

$$a^{I'} = a^I, \qquad A^{I'}(u) = \begin{cases} \infty, & \text{if } u \in A^I, \\ 0, & \text{otherwise,} \end{cases} \qquad P^{I'}(u, v) = \begin{cases} \infty, & \text{if } (u, v) \in P^I, \\ 0, & \text{otherwise.} \end{cases}$$

Bag interpretation I' satisfies $\mathcal{A}'$ and all axioms in $\mathcal{T}$. Thus, I' is a model of $\mathcal{K}'$ and, therefore, $\mathcal{K}'$ is satisfiable, as required. Conversely, suppose that $\mathcal{K}'$ has a model $I' = \langle \Delta^{I'}, \cdot^{I'} \rangle$. We construct an interpretation $I = \langle \Delta^{I'}, \cdot^I \rangle$ as follows, for each $a \in \mathbf{I}$, $A \in \mathbf{C}$, $P \in \mathbf{R}$, and $u, v \in \Delta^{I'}$:

$$\begin{aligned} a^I &= a^{I'}, \\ u \in A^I &\text{ if and only if } A^{I'}(u) > 0, \\ (u, v) \in P^I &\text{ if and only if } P^{I'}(u, v) > 0. \end{aligned}$$

Interpretation I is a model of $\mathcal{K}$ by construction, which completes the proof of Statement 1.

(*Statement 2*) We show the claim by considering a case for each kind of axiom in $\mathcal{T}$.

Let first α be $S_1 \sqsubseteq S_2$ where S_1 and S_2 are either both concepts or both roles. To show that $\mathcal{T} \models^{\mathrm{b}} S_1 \sqsubseteq S_2$ implies $\mathcal{T} \models S_1 \sqsubseteq S_2$, assume that $\mathcal{T} \models^{\mathrm{b}} S_1 \sqsubseteq S_2$ but $\mathcal{T} \not\models S_1 \sqsubseteq S_2$. Then, the following *DL-Lite*$_{\mathcal{R}}$ ontology must be satisfiable [9]: $\langle \mathcal{T} \cup \{S \sqsubseteq S_1,\ \mathsf{Disj}(S, S_2)\}, \{S(\mathbf{a})\} \rangle$, where S is fresh. By Statement 1, the *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontology consisting of the same TBox and a bag ABox $\mathcal{A}' = \{\!\{ S(\mathbf{a}) : 1 \}\!\}$ is satisfiable. Thus, there exists a bag interpretation I such that $I \models^{\mathrm{b}} \mathcal{T}$, $S^I \subseteq S_1^I$, $S^I \cap S_2^I = \emptyset$, and $S^I(\mathbf{u}) > 0$ for some tuple $\mathbf{u}$ of domain elements. From this we derive that $S_1^I(\mathbf{u}) > 0$ and $S_2^I(\mathbf{u}) = 0$ and hence $S_1^I \not\subseteq S_2^I$, which then implies $\mathcal{T} \not\models^{\mathrm{b}} S_1 \sqsubseteq S_2$, contradicting our assumption.

We now show that $\mathcal{T} \models S_1 \sqsubseteq S_2$ implies $\mathcal{T} \models^{\mathrm{b}} S_1 \sqsubseteq S_2$. Let $\mathcal{T}'$ be the TBox extending $\mathcal{T}$ with the following inclusions for each role inclusion $R_1 \sqsubseteq R_2$ in $\mathcal{T}$: $\exists R_1 \sqsubseteq \exists R_2$, $\exists R_1^- \sqsubseteq \exists R_2^-$, and $R_1^- \sqsubseteq R_2^-$, for R_1^- and R_2^- the inverses of R_1 and R_2, respectively. Following [34], $\mathcal{T} \models S_1 \sqsubseteq S_2$ implies that either $\mathcal{T} \models \mathsf{Disj}(S_1, S_1)$, or there exists a chain of inclusions $T_0 \sqsubseteq T_1, \ldots, T_{n-1} \sqsubseteq T_n$ in $\mathcal{T}'$ such that $T_0 = S_1$ and $T_n = S_2$. In the first case, we have that $\mathcal{T} \models^{\mathrm{b}} \mathsf{Disj}(S_1, S_1)$ by definition. In the second case, the chain of inclusions in $\mathcal{T}'$ implies that $\mathcal{T}' \models^{\mathrm{b}} S_1 \sqsubseteq S_2$ since $T_0^I \subseteq T_1^I \subseteq \cdots \subseteq T_n^I$ should hold for every bag interpretation I satisfying $\mathcal{T}'$. Then, $\mathcal{T} \models^{\mathrm{b}} S_1 \sqsubseteq S_2$ follows from the fact that every bag interpretation satisfying $\mathcal{T}$ satisfies also the additional inclusions in $\mathcal{T}'$ by construction.

Let now α be $\mathsf{Disj}(S_1, S_2)$ where S_1, S_2 are either both concepts or both roles. If $\mathcal{T} \not\models^{\mathrm{b}} \mathsf{Disj}(S_1, S_2)$, then there exists a bag interpretation I such that $I \models^{\mathrm{b}} \mathcal{T}$ and $S_1^I \cap S_2^I \neq \emptyset$. Let I' be the set interpretation constructed in the proof of Statement 1 on the basis of I. By construction we have that $I' \models \mathcal{T}$ and $S_1^{I'} \cap S_2^{I'} \neq \emptyset$; thus $\mathcal{T} \not\models \mathsf{Disj}(S_1, S_2)$, as required. The other direction can be shown in exactly the same way.

(*Statement 3*) First, let $\mathbf{a} \in q^{\mathcal{K}}$—that is, let $\mathbf{a}$ belong to the certain answers to q over $\mathcal{K}$—and assume for the sake of contradiction that $q^{\mathcal{K}'}(\mathbf{a}) = 0$. The latter means that there exists a model I' of $\mathcal{K}'$ such that $q^{I'}(\mathbf{a}) = 0$. Consider the interpretation I constructed on the basis of I' as in the proof of Statement 1. Interpretation I is a model of $\mathcal{K}$ such that $I \not\models q(\mathbf{a})$, which yields a contradiction. Thus, $q^{\mathcal{K}'}(\mathbf{a}) \geq 1$, as required. The other direction can be shown in exactly the same way. □

5.2. Unique name assumption

As we mentioned before, in the set case general satisfiability and satisfiability under the UNA coincide for *DL-Lite*$_{\mathcal{R}}$, and the same holds for axiom entailment. The following corollary, which states similar claims for bag semantics, is an immediate consequence of Statements 1 and 2 of Theorem 23 and this property.

Corollary 24. *A DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ *ontology is satisfiable if and only if it is satisfiable under the UNA. A DL-Lite*$_{\mathcal{R}}$ *TBox entails an axiom under bag semantics if and only if it entails this axiom under bag semantics and the UNA.*

Artale et al. [24] showed that query answering over satisfiable *DL-Lite*$_{\mathcal{R}}$ ontologies is also independent of whether the UNA is adopted or not; indeed the UNA may influence query answers under set semantics only if the ontology language allows for some form of equality (e.g., functionality constraints). We next argue that the situation is markedly different under bag semantics. The following proposition shows that the UNA can influence query answering under bag semantics as soon as role inclusions are allowed in the ontology language.

Proposition 25. *There exists a satisfiable DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ *ontology $\mathcal{K}$ and a rooted CQ q such that the (general) certain answers to q over $\mathcal{K}$ differ from the certain answers under the UNA.*

Proof. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a *DL-Lite*$^{\mathrm{b}}_{\mathcal{R}}$ ontology with $\mathcal{T} = \{P' \sqsubseteq P\}$ and $\mathcal{A} = \{\!|\, P(a, b_1) : 1, P'(a, b_2) : 1 \,|\!\}$. Let also $q = \exists y.\, P(a, y)$. Under the UNA, we have that $q^{\mathcal{K}}(\langle\rangle) = 2$. Indeed, in all models I satisfying the UNA, individuals b_1 and b_2 are interpreted as different elements; as a result, in each such I, the element a^I is associated with at least two elements in P^I. In contrast, if the UNA is not adopted, then the certain answer is 1, which is witnessed by an interpretation that maps b_1 and b_2 to the same element of the domain. □

As we will see later on (in Corollary 43), the presence of role inclusions is crucial for this mismatch, and the certain answers to rooted CQs over *DL-Lite*$^{\mathrm{b}}_{\mathrm{CORE}}$ ontologies do not depend on whether the UNA is adopted.

5.3. Universal models

An important property of each satisfiable *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{K}$ is the existence of so-called universal models for CQs—that is, models I such that the certain answers to every CQ q over $\mathcal{K}$ can be obtained by evaluating q over I [9]. Existence of such universal models is critical to the favourable computational properties of *DL-Lite*$_{\mathcal{R}}$. The notion of a universal model for bags is the same as for sets.

Definition 26. A model I of an ontology $\mathcal{K}$ is *universal* for a class of queries $\mathcal{Q}$ if $q^{\mathcal{K}} = q^I$ for all $q \in \mathcal{Q}$. It is *universal under the UNA* if the certain answers under the UNA are considered.

In the set case, it is well-known that universal models for the class of CQs always exist for satisfiable ontologies $\mathcal{K}$. In fact, they are canonical interpretations—that is, interpretations that can be obtained by a restricted chase procedure applied to $\mathcal{K}$ [35]. It is also well-known that a model of $\mathcal{K}$ is universal for the class of CQs if and only if it can be homomorphically embedded into every other model of $\mathcal{K}$ [9]. Unfortunately, in contrast to the set case, even *DL-Lite*$^{\mathrm{b}}_{\mathrm{CORE}}$ ontologies may not admit universal models for all CQs.

Proposition 27. *There exists a satisfiable DL-Lite$^{\flat}_{\text{CORE}}$ ontology $\mathcal{K}$ that has neither a universal model nor a universal model under the UNA for the class of all CQs.*

Proof. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be the *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology with $\mathcal{T} = \{A \sqsubseteq \exists P, \exists P^- \sqsubseteq B\}$ and $\mathcal{A} = \{\!\{ A(a) : 1, B(b) : 1 \}\!\}$. Consider the bag interpretation I_1 with domain $\{a, b\}$ that interprets the individuals by themselves and interprets the concepts and roles as follows:

$$A^{I_1} = \{\!\{ a : 1 \}\!\}, \qquad B^{I_1} = \{\!\{ b : 1 \}\!\}, \qquad P^{I_1} = \{\!\{ (a, b) : 1 \}\!\}.$$

Similarly, consider the bag interpretation I_2 with domain $\{a, b, u\}$ that interprets the individuals by themselves and interprets concepts and roles as follows:

$$A^{I_2} = \{\!\{ a : 1 \}\!\}, \qquad B^{I_2} = \{\!\{ b : 1,\ u : 1 \}\!\}, \qquad P^{I_2} = \{\!\{ (a, u) : 1 \}\!\}.$$

It is immediate to verify that both I_1 and I_2 are models of $\mathcal{K}$. Moreover, for the Boolean CQs

$$q_r = P(a, b) \qquad \text{and} \qquad q_{nr} = \exists y.\, B(y),$$

we have that

$$\begin{aligned} q_r^{I_1}(\langle\rangle) &= 1, \qquad q_r^{I_2}(\langle\rangle) = 0, \\ q_{nr}^{I_1}(\langle\rangle) &= 1, \qquad q_{nr}^{I_2}(\langle\rangle) = 2; \end{aligned}$$

thus, neither model is universal for $\{q_r, q_{nr}\}$.

Suppose now there is a universal model I for $\{q_r, q_{nr}\}$. Then, since $q_r^I(\langle\rangle)$ must be 0, we have that $P^I(a^I, b^I) = 0$. Since assertion $A(a)$ occurs in $\mathcal{A}$ with multiplicity 1 and inclusion $A \sqsubseteq \exists P$ belongs to $\mathcal{T}$, we have that $P^I(a, v) \geq 1$ for some $v \in \Delta^I$ distinct from element b^I. Since inclusion $\exists P^- \sqsubseteq B$ is in $\mathcal{T}$, it follows that $B^I(v) \geq 1$, and, hence, $q_{nr}^I(\langle\rangle) \geq 2$, contradicting universality of I.

Finally, note that the proof also works under the UNA. □

6. Lower bounds for the data complexity of query answering under bag semantics

The lack of universal models illustrated in Section 5.3 suggests that CQ answering under bag semantics is computationally more challenging than in the set case. In this section, we show that this is indeed the case and establish three incomparable coNP lower bounds in data complexity. These are in stark contrast to the well-known AC^0 upper bound in the set case for CQ answering over *DL-Lite*$_{\mathcal{R}}$.

The first lower bound is given in Theorem 28, where we show that CQ answering is coNP-hard even if we restrict the ontology language to *DL-Lite*$^{\flat}_{\text{CORE}}$ regardless of the adoption of the UNA. The second and third lower bounds are established in Theorem 29, where we show similar coNP-hardness results for the cases where the query language is restricted to the class of rooted CQs and the ontology language is allowed to contain role inclusions.

Theorem 28. *Both* BagCert$\big[$CQs, *DL-Lite*$^{\flat}_{\text{CORE}}\big]$ *and* BagCert$^{\text{UNA}}\big[$CQs, *DL-Lite*$^{\flat}_{\text{CORE}}\big]$ *are* coNP*-hard in data complexity.*

Proof. We prove that there exists a *DL-Lite*$^{\flat}_{\text{CORE}}$ TBox $\mathcal{T}$ and a Boolean CQ q such that checking whether $q^{\langle \mathcal{T}, \mathcal{A} \rangle}(\langle\rangle) \geq k$ for an input bag ABox $\mathcal{A}$ and $k \in \mathbb{N}_0^{\infty}$ is coNP-hard regardless of whether the UNA is adopted or not. Following ideas of Kostylev and Reutter [36], we provide a reduction of the complement of the 3-colourability problem for directed graphs, a well-known coNP-complete problem, to query answering.

We first address the case of BagCert$^{\text{UNA}}$. Let $G = \langle V, E \rangle$ be a directed graph with vertices V and edges E. We construct a *DL-Lite*$_{\text{CORE}}$ TBox $\mathcal{T}$ and a Boolean CQ q, neither of which depends on G, as well as a bag ABox $\mathcal{A}_G$ based on G, such that G is not 3-colourable if and only if $q^{\langle \mathcal{T}, \mathcal{A}_G \rangle}(\langle\rangle) \geq 3 \times |V| + 2$.

First, let $\mathcal{T}$ consist of the inclusions

$$\mathit{Vertex} \sqsubseteq \exists \mathit{hasColour} \qquad \text{and} \qquad \exists \mathit{hasColour}^- \sqsubseteq \mathit{Colour},$$

where *Vertex* and *Colour* are atomic concepts, and *hasColour* is an atomic role, and let q be the Boolean CQ

$$\exists x, y, z, w.\, \mathit{Edge}(x, y) \wedge \mathit{hasColour}(x, z) \wedge \mathit{hasColour}(y, z) \wedge \mathit{Colour}(w).$$

Then, let $\mathcal{A}_G$ be the bag ABox defined as given next, where we use an individual a_v for each vertex $v \in V$, an individual a representing an auxiliary "vertex", individuals r, g, and b representing three colours, atomic role *Edge*, as well as the concepts and roles introduced before:

- $\mathit{Vertex}(a_v)$ has multiplicity 1, for each vertex $v \in V$;
- $\mathit{Edge}(a_{v_1}, a_{v_2})$ has multiplicity 1, for each edge $(v_1, v_2) \in E$;

- $Colour(r)$ has multiplicity $|V|+1$ for colour r;
- $Colour(g)$ and $Colour(b)$ each has multiplicity $|V|$ for colours g and b;
- $Vertex(a)$, $Edge(a,a)$, and $hasColour(a,r)$ each has multiplicity 1 for the auxiliary "vertex" a; and
- all other assertions have multiplicity 0.

Concept $Vertex$ and role $Edge$ are used to encode G. The role $hasColour$ represents a colour assignment to the vertices of G, where inclusions $Vertex \sqsubseteq \exists hasColour$ and $\exists hasColour^- \sqsubseteq Colour$ necessitate the association of each vertex to a colour. Concept $Colour$ provides a sufficient number of pre-defined copies of the three colours; every proper colour assignment of G shall use at most $|V|$ times each of these colours. We next exploit these properties for showing that

$$G \text{ is not 3-colourable if and only if } q^{\langle \mathcal{T}, \mathcal{A}_G \rangle}(\langle\rangle) \geq 3 \times |V| + 2.$$

First, let G be not 3-colourable. Consider an arbitrary model I of $\langle \mathcal{T}, \mathcal{A}_G \rangle$. We next show that $q^I(\langle\rangle) \geq 3 \times |V| + 2$. Since I is a model of bag ABox $\mathcal{A}_G$, the valuation λ defined as $\lambda(x) = \lambda(y) = a^I$ and $\lambda(z) = \lambda(w) = r^I$ contributes to $q^I(\langle\rangle)$ a multiplicity of at least $|V|+1$; this is because $\mathcal{A}_G$ contains assertion $Colour(r)$ with multiplicity $|V|+1$ and assertions $Vertex(a)$, $Edge(a,a)$, and $hasColour(a,r)$ with multiplicity 1. Similarly, each valuation that differs from λ by sending w to either g^I or b^I contributes to $q^I(\langle\rangle)$ a multiplicity of at least $|V|$. We have two possibilities for I: either there exists an element $u \in \Delta^I$ different from r^I, g^I and b^I such that $Colour^I(u) \geq 1$ or not. In the first case the valuation that differs from λ by sending w to u instead of r^I contributes to $q^I(\langle\rangle)$ a multiplicity of at least 1, so overall we have $q(\langle\rangle)^I \geq 3 \times |V| + 2$, as required. In the second case we can consider a colour assignment γ to V such that, for every $v \in V$, $\gamma(v)$ is red if $hasColour^I(a_v^I, r^I) \geq 1$, it is green if $hasColour^I(a_v^I, g^I) \geq 1$, and it is blue if $hasColour^I(a_v^I, b^I) \geq 1$ (if there are several possible options for some v we can just pick any of them). Since G is not 3-colourable, there exists an edge (v_1, v_2) in E such that $\gamma(v_1) = \gamma(v_2)$. Consider the valuation λ' such that $\lambda'(x) = a_{v_1}^I$, $\lambda'(y) = a_{v_2}^I$, $\lambda'(z)$ is one of r^I, g^I and b^I corresponding to the colour of v_1 and v_2 under γ, and $\lambda'(w) = r^I$. By construction, λ' contributes to $q^I(\langle\rangle)$ a multiplicity of at least $|V|+1$. Therefore, overall we have that $q(\langle\rangle)^I \geq 3 \times |V| + 2$, as required.

Assume now that G is 3-colourable. It suffices to show that there exists a model I of $\langle \mathcal{T}, \mathcal{A}_G \rangle$ for which $q^I(\langle\rangle) < 3 \times |V| + 2$. Since G is 3-colourable, there is an assignment $\gamma : V \to \{r, g, b\}$ such that, for every $(v_1, v_2) \in E$, $\gamma(v_1) \neq \gamma(v_2)$. Consider a bag interpretation I with the domain consisting of all the individuals (i.e., $\Delta^I = \{a_v \mid v \in V\} \cup \{a, r, g, b\}$) that interprets all the individuals by themselves, and such that $Vertex^I$, $Edge^I$ and $Colour^I$ are defined precisely according to $\mathcal{A}_G$ (e.g., $Vertex^I(c) = \mathcal{A}_G(Vertex(c))$ for every individual c), while

$$hasColour^I(u_1, u_2) = \begin{cases} 1, & \text{if } u_1 = a_v \text{ and } u_2 = \gamma(v) \text{ for } v \in V, \\ 1, & \text{if } u_1 = a \text{ and } u_2 = r, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, interpretation I is defined on the basis of the 3-colouring of G. By construction, I is a model of $\langle \mathcal{T}, \mathcal{A}_G \rangle$. Next, we show that $q^I(\langle\rangle) = 3 \times |V| + 1$. First, we observe that the first three atoms $Edge(x,y)$, $hasColour(x,z)$, and $hasColour(y,z)$ of q match exactly once (i.e., under the valuation sending x and y to a, and z to r). Next, there are precisely three possibilities for variable w, namely r, g, and b, contributing multiplicity $3 \times |V| + 1$ in total. Consequently, $q^I(\langle\rangle) = 3 \times |V| + 1$, as desired.

We now address the case of BagCert by discussing the required modifications in the aforementioned reduction. For this, it is enough to ensure that, first, the auxiliary "vertex" a is not interpreted by the same element as any of the vertices of G; and, second, that the colour individuals r, g, and b are interpreted by pairwise different elements. To ensure this, we use atomic concepts V_a, V_G, Red, $Green$, and $Blue$. We add the following disjointness axioms to TBox $\mathcal{T}$: $\mathsf{Disj}(V_a, V_G)$, $\mathsf{Disj}(Red, Green)$, $\mathsf{Disj}(Red, Blue)$, and $\mathsf{Disj}(Green, Blue)$. We also modify bag ABox $\mathcal{A}_G$ by setting the multiplicity of $V_a(a)$, $Red(r)$, $Green(g)$, $Blue(b)$, and $V_G(a_v)$, for every vertex $v \in V$, to 1 (and the multiplicity of all other assertions over the new concepts to 0). Following the same argumentation as for the case of BagCert$^{\text{UNA}}$, we can show that the above reduction works when the UNA is dropped. □

Note that the query constructed in the proof of Theorem 28 is not rooted; furthermore, the use of the disconnected atom $Colour(w)$ in the query is instrumental to the correctness of the reduction. In Section 8 we show that rooted CQs are rewritable to BCALC over $DL\text{-}Lite^{\flat}_{\text{CORE}}$ regardless of the adoption of the UNA—that is, the problems are in LogSpace in data complexity.

Unfortunately, the restriction to rooted CQs alone is not sufficient to ensure tractability of query answering for bag ontology languages allowing for role inclusions. In the first part of Theorem 29 we show that answering rooted CQs is intractable (coNP-hard) even if we restrict ourselves to $DL\text{-}Lite^{\flat}_{\text{RDFS}}$ ontologies, which allow for role inclusions while at the same time disallowing existential quantification on the right-hand side of concept inclusions. This lower bound, however, critically depends on the fact that the UNA is not adopted; indeed, in Section 9 we will show that all CQs (and not just rooted ones) are rewritable to BCALC over $DL\text{-}Lite^{\flat}_{\text{RDFS}}$ under the UNA. On the other hand, even if adopting the UNA can make rooted CQ answering easier, in the second part of Theorem 29 we show that it remains intractable in general: answering rooted CQs over $DL\text{-}Lite^{\flat}_{\mathcal{R}}$ is coNP-hard under the UNA.

Theorem 29. BagCert$\left[\text{rooted CQs}, \mathit{DL\text{-}Lite}^{b}_{\text{RDFS}}\right]$ *and* BagCert$^{\text{UNA}}\left[\text{rooted CQs}, \mathit{DL\text{-}Lite}^{b}_{\mathcal{R}}\right]$ *are* coNP*-hard in data complexity.*

Proof. We first prove the claim for the case of BagCert, and then show how to adapt the proof to the case of BagCert$^{\text{UNA}}$. The proof is again by reduction of the complement of the 3-colouring problem for directed graphs; however, the reduction is more involved. Let $G = \langle V, E\rangle$ be a directed graph with vertices V and edges E. Next, we define a *DL-Lite*$_{\mathcal{R}}$ TBox $\mathcal{T}$ and Boolean rooted CQ q, neither of which depends on G, as well as a bag ABox A_G, based on G, such that G is not 3-colourable if and only if $q^{\langle\mathcal{T},\mathcal{A}_G\rangle}(\langle\rangle) \geq 3 \times |V| + 2$.

First, let $\mathcal{T}$ consist of a single role inclusion

$$\mathit{hasColour} \sqsubseteq \mathit{Colour},$$

where *hasColour* and *Colour* are atomic roles. Let also q be the following Boolean rooted CQ, where a_0 is a "root" individual, and *Edge*, *Beg*, *End* and *Vertex* are atomic roles:

$$\exists x_v, x_c, y_e, y_v^1, y_v^2, y_c.\, \mathit{Vertex}(a_0, x_v) \wedge \mathit{Colour}(x_v, x_c) \wedge \mathit{Edge}(a_0, y_e) \wedge \mathit{Beg}(y_e, y_v^1) \wedge \mathit{hasColour}(y_v^1, y_c) \wedge \mathit{End}(y_e, y_v^2) \wedge \mathit{hasColour}(y_v^2, y_c).$$

Finally, let the bag ABox $\mathcal{A}_G$ mention the "root" individual a_0 of q, individuals a_v and c_v associated to vertices $v \in V$, individuals r, g, and b corresponding to the three colours, individuals a_e associated to edges $e \in E$, as well as an auxiliary "vertex" individual a and "edge" individual a_*; let also $\mathcal{A}_G$ assign 1 to the following assertions (and 0 to all others):

- $\mathit{Vertex}(a_0, a_v)$, $\mathit{Colour}(a_v, r)$, $\mathit{Colour}(a_v, g)$, $\mathit{Colour}(a_v, b)$, $\mathit{hasColour}(a_v, c_v)$, for each vertex $v \in V$,
- $\mathit{Edge}(a_0, a_e)$, $\mathit{Beg}(a_e, a_{v_1})$, $\mathit{End}(a_e, a_{v_2})$, for each $e = (v_1, v_2)$ in E,
- $\mathit{Vertex}(a_0, a)$, $\mathit{Colour}(a, r)$, $\mathit{hasColour}(a, r)$, and
- $\mathit{Edge}(a_0, a_*)$, $\mathit{Beg}(a_*, a)$, $\mathit{End}(a_*, a)$.

Having the reduction complete, next we show that it is correct—that is, that

$$G \text{ is not 3-colourable if and only if } q^{\langle\mathcal{T},\mathcal{A}_G\rangle}(\langle\rangle) \geq 3 \times |V| + 2.$$

Intuitively, every model of $\langle\mathcal{T}, \mathcal{A}_G\rangle$ has $3 \times |V| + 1$ contributing valuations for q that send the subquery of q over the y variables to the (interpretations of) the assertions over the auxiliary a_*, a and r, while the subquery over the x variables to the assertions over one of a_v and a, and one of r, g, and b. Then, if some c_v is interpreted as neither r, nor g, nor b, we can construct one more valuation sending x_c to the interpretation of c_v (here we make use of the TBox $\mathcal{T}$). Otherwise, the identifications of c_v can be seen as a colouring of the vertices (represented by a_v individuals), and every valid colouring corresponds to the model possessing exactly $3 \times |V| + 1$ valuations. Next, we make this intuition formal. In fact, in the both directions of the correctness proof we make use of the following fact.

Claim 30. *For every bag interpretation I satisfying all assertions in $\mathcal{A}_G$, $q^I(\langle\rangle) \geq 3 \times |V| + 1$.*

Proof. Let I be a bag interpretation satisfying ABox $\mathcal{A}_G$. Consider all the valuations λ such that $\lambda(y_e) = a_*^I$, $\lambda(y_v^1) = \lambda(y_v^2) = a^I$, $\lambda(y_c) = r^I$, as well as $\lambda(x_v)$ is one of a_v, for $v \in V$, and $\lambda(x_c)$ is one of r^I, g^I, and b^I. Since I satisfies $\mathcal{A}_G$, each of these valuations contribute at least 1 to $q^I(\langle\rangle)$, and there are overall $3 \times |V|$ of them. Note that we rely only on the cardinality of the ABox here, so even if the interpretations of the individuals are not pairwise distinct—that is, if the UNA is violated—and some of these valuations may coincide, the total contribution of these valuations is still at least $3 \times |V|$ by Definition 15. Consider now the valuation λ' that is the same as before on y_e, y_v^1, y_v^2 and y_c, but such that $\lambda(x_v) = a^I$ and $\lambda(x_c) = r^I$. This valuation also contributes a multiplicity of at least 1. Moreover, for the same reason as before, the contribution of each of the considered valuations is separate—that is, the total contribution is at least $3 \times |V| + 1$. □

Having this claim at hand, we are ready to show correctness of the reduction. Let first G be not 3-colourable. Consider an arbitrary model I of $\langle\mathcal{T}, \mathcal{A}_G\rangle$. Since I satisfies all assertions in $\mathcal{A}_G$, by Claim 30 we know that there are valuations that contribute $3 \times |V| + 1$ to $q^I(\langle\rangle)$. So, it is enough to show that there is a valuation with a non-zero and different contribution. We have two possibilities: either there is a vertex $v \in V$ such that c_v^I is distinct from r^I, g^I, and b^I, or not.

In the first case, consider such a vertex v and the valuation λ that is the same as in Claim 30 on y_c, y_v^1, y_v^2 and y_c, but such that $\lambda(x_v) = a_v^I$ and $\lambda(x_c) = c_v^I$. On the one hand, $\mathcal{A}(\mathit{hasColour}(a_v, c_v)) = 1$ and $\mathcal{T}$ has the inclusion $\mathit{hasColour} \sqsubseteq \mathit{Colour}$, so $\mathit{Colour}^I(a_v^I, c_v^I) \geq 1$ and, therefore, the contribution of this valuation is at least 1. On the other hand, we have not considered this valuation yet, because c_v^I is different from r^I, g^I, and b^I by assumption.

Consider now the second case—that is, the case when c_v^I is among r^I, g^I and b^I for each v. Construct a colour assignment γ to vertices such that, for each such vertex v, $\gamma(v)$ is red if $c_v^I = r^I$, it is green if $c_v^I = g^I$, and it is blue if $c_v^I = b^I$ (if some of r^I, g^I, and b^I coincide, then there are many of such assignments and γ can be any of them). We know that G does not have a valid colouring, so there is an edge $e = (v_1, v_2)$ in E with both v_1 and v_2 assigned to the same colour. For brevity,

Table 1
Data complexity of BagCert[$\mathcal{Q}, \mathcal{O}$] and BagCert$^{\text{UNA}}$[$\mathcal{Q}, \mathcal{O}$] with references to the corresponding theorems (the bounds without references follow immediately from the others).

UNA	$\mathcal{Q}$	$\mathcal{O}$			
		DL-Lite$^{\flat}_{\text{CORE}}$	*DL-Lite*$^{\flat}_{\text{RDFS}}$	*DL-Lite*$^{\flat}_{\mathcal{R}^-}$	*DL-Lite*$^{\flat}_{\mathcal{R}}$
No	CQs	coNP-hard [Theorem 28]	coNP-hard	coNP-hard	coNP-hard
	Rooted CQs	in LogSpace [Corollary 59]	coNP-hard [Theorem 29]	coNP-hard	coNP-hard
Yes	CQs	coNP-hard [Theorem 28]	in LogSpace [Corollary 68]	coNP-hard	coNP-hard
	Rooted CQs	in LogSpace	in LogSpace	in LogSpace [Corollary 79]	coNP-hard [Theorem 29]

consider only the case when this colour is red; the other two cases are symmetric. Let λ be the valuation that agrees with a valuation in Claim 30 on the variables x_v and x_c, and follows the following assignment for the rest of the variables: $\lambda(y_e) = a_e^I$, $\lambda(y_v^1) = a_{v_1}^I$, $\lambda(y_v^2) = a_{v_2}^I$, $\lambda(y_c) = r^I$. On the one hand, the contribution of this valuation to $q^I(\langle\rangle)$ is at least 1 by construction. On the other hand, this valuation is different from the ones considered in Claim 30.

Therefore, in both cases we have that $q^I(\langle\rangle) \geq 3 \times |V| + 2$, as required.

Let now G be 3-colourable—that is, there is a colour assignment γ to V such that the vertices of each edge are coloured differently. We next show that $q^{\langle\mathcal{T},\mathcal{A}_G\rangle}(\langle\rangle) < 3 \times |V| + 2$. To this end, consider the bag interpretation I defined as follows:

- for each vertex $v \in V$, c_v^I is r if $\gamma(v)$ is red, it is g if $\gamma(v)$ is green, and it is b if $\gamma(v)$ is blue;
- all other individuals are interpreted by themselves; and
- all the atomic roles are interpreted as dictated by the ABox (i.e., $S^I(u_1^I, u_2^I) = \mathcal{A}_G(S(u_1, u_2))$ for every atomic role S and every pair of individuals u_1, u_2).

On the one hand, interpretation I is a model of $\langle\mathcal{T}, \mathcal{A}_G\rangle$ by construction, because the identification of all c_v^I with one of r, g and b makes I satisfy the inclusion of $\mathcal{T}$ for the only relevant assertions $\mathit{hasColour}(a_v, c_v)$. On the other hand, $q^I(\langle\rangle) = 3 \times |V| + 1$: indeed, it is at least $3 \times |V| + 1$ by Claim 30, and it is immediate to check that there are no more valuations with a non-zero contribution to $q^I(\langle\rangle)$. So, I is a witness for the fact that $q^{\langle\mathcal{T},\mathcal{A}_G\rangle}(\langle\rangle) < 3 \times |V| + 2$.

We are left to show the second part of the theorem—that is, coNP-hardness of BagCert for the case when the UNA is adopted, but arbitrary *DL-Lite*$_{\mathcal{R}}$ TBoxes are allowed. In fact, we can essentially repurpose the same reduction as in the first part. The only modifications in the reduction are that the ABox $\mathcal{A}_G$ does not have assertions $\mathit{hasColour}(a_v, c_v)$, for $v \in V$ (i.e., does not use individuals c_v at all), while the TBox $\mathcal{T}$ additionally has the inclusion $\exists \mathit{Vertex}^- \sqsubseteq \exists \mathit{hasColour}$. Then, the correctness proof goes along the same lines as in the first case, except that anonymous domain elements u_v, which are enforced by the new inclusion for each $v \in V$, are used instead of the c_v^I. □

Since the data complexity of BCALC is strictly contained in LogSpace, the lower bounds in Theorems 28 and 29 imply non-rewritability to BCALC for the relevant query and ontology languages.

Corollary 31. *The class of all CQs is not rewritable to* BCALC *over DL-Lite*$^{\flat}_{\text{CORE}}$*, both in general and under the UNA. The class of rooted CQs is not rewritable to* BCALC *over DL-Lite*$^{\flat}_{\text{RDFS}}$ *in general and over DL-Lite*$^{\flat}_{\mathcal{R}}$ *under the UNA.*

In the following sections, we investigate how to regain tractability of query answering and rewritability to BCALC by considering suitable restrictions on the query and ontology languages that allow us to circumvent the bounds in Theorems 28 and 29. In Sections 7 and 8 we focus on *DL-Lite*$^{\flat}_{\text{CORE}}$ and show that the class of rooted CQs is rewritable to BCALC both in general and under the UNA. In Section 9 we focus on the ontology language *DL-Lite*$^{\flat}_{\text{RDFS}}$ and show that all CQs (and not just rooted ones) are rewritable to BCALC under the UNA. Finally, in Section 10 we show rewritability of rooted CQs over *DL-Lite*$^{\flat}_{\mathcal{R}^-}$, which extends both *DL-Lite*$^{\flat}_{\text{CORE}}$ and *DL-Lite*$^{\flat}_{\text{RDFS}}$, under the UNA. For the convenience of the reader, we summarise in Table 1 all the data complexity results proved in this paper.

7. Universal models for rooted conjunctive queries over *DL-Lite*$^{\flat}_{\text{CORE}}$ ontologies

Our next main goal is to show tractability of answering rooted CQs over *DL-Lite*$^{\flat}_{\text{CORE}}$ in data complexity and their BCALC rewritability, both regardless of the adoption of the UNA. Towards this goal, in this section we show that every satisfiable *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology admits a universal model for rooted CQs, both in general and under the UNA. To this end, we proceed as in the set case: we first define a special bag interpretation for each *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology, which we call *canonical*, and then, after developing dedicated machinery, prove that it is indeed universal for the class of rooted CQs when the ontology is satisfiable. However, in contrast to the set case, the requirement for CQs to be rooted is crucial here: recall Proposition 27, where we constructed a *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology that does not have a universal model for all CQs.

To formalise canonical bag interpretations, we need two auxiliary notions. First, the *concept closure* $\mathsf{ccl}_{\mathcal{T}}[u, I]$ of an element $u \in \Delta^I$ in a bag interpretation $I = \langle\Delta^I, \cdot^I\rangle$ over a TBox $\mathcal{T}$ is the bag of concepts such that, for every concept C,

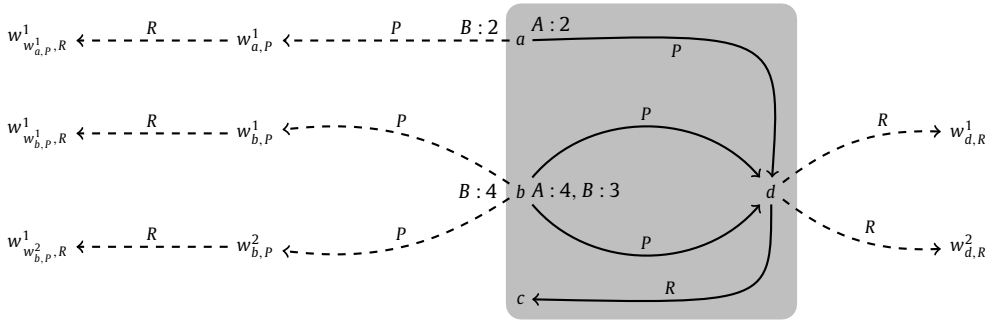

Fig. 1. The canonical bag interpretation of the *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology considered in Example 33.

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

In other words, $\mathsf{ccl}_{\mathcal{T}}[u, I](C)$ is the maximum value of $C_0^I(u)$ amongst all concepts C_0 satisfying $\mathcal{T} \models C_0 \sqsubseteq C$—that is, $\mathsf{ccl}_{\mathcal{T}}[u, I](C)$ is the minimal multiplicity of $C^J(u)$ required for an extension J of I to satisfy TBox $\mathcal{T}$ locally in u.

Second, the *union* $I \cup J$ of two bag interpretations $I = \langle \Delta^I, \cdot^I \rangle$ and $J = \langle \Delta^J, \cdot^J \rangle$ interpreting all the individuals in the same way—that is, such that $a^I = a^J$ for all $a \in \mathbf{I}$—is the bag interpretation $\langle \Delta^I \cup \Delta^J, \cdot^{I \cup J} \rangle$ with $a^{I \cup J} = a^I$ for all individuals $a \in \mathbf{I}$ and $S^{I \cup J} = S^I \cup S^J$ for all atomic concepts and roles $S \in \mathbf{C} \cup \mathbf{R}$ (recall that S^I and S^J are bags, so $S^I \cup S^J$ is the bag maximal union).

Definition 32. The *canonical bag interpretation* $Can(\mathcal{K})$ of a *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is the bag interpretation that is the union $\bigcup_{i \geq 0} Can_i(\mathcal{K})$ of the bag interpretations $Can_i(\mathcal{K})$ defined as follows:

- $Can_0(\mathcal{K}) = \langle \Delta^{Can_0(\mathcal{K})}, \cdot^{Can_0(\mathcal{K})} \rangle$ is the bag interpretation corresponding to bag ABox $\mathcal{A}$—that is, such that $\Delta^{Can_0(\mathcal{K})} = \mathbf{I}$, $a^{Can_0(\mathcal{K})} = a$ for each $a \in \mathbf{I}$, and $S^{Can_0(\mathcal{K})}(\mathbf{a}) = \mathcal{A}(S(\mathbf{a}))$ for each $S \in \mathbf{C} \cup \mathbf{R}$ and individuals $\mathbf{a}$;
- for each $i > 0$, $Can_i(\mathcal{K}) = \langle \Delta^{Can_i(\mathcal{K})}, \cdot^{Can_i(\mathcal{K})} \rangle$ extends $Can_{i-1}(\mathcal{K})$ by satisfying all the inclusions that are not satisfied in $Can_{i-1}(\mathcal{K})$—that is,

$$\Delta^{Can_i(\mathcal{K})} = \Delta^{Can_{i-1}(\mathcal{K})} \cup \big\{ w^1_{u,R}, \dots, w^{\delta}_{u,R} \mid u \in \Delta^{Can_{i-1}(\mathcal{K})} \text{ and } R \text{ is a role such that } \delta = \mathsf{ccl}_{\mathcal{T}}[u, Can_{i-1}(\mathcal{K})](\exists R) - (\exists R)^{Can_{i-1}(\mathcal{K})}(u) \big\},$$

where $w^j_{u,R}$ are fresh domain elements, called *anonymous*, and, for all $a \in \mathbf{I}$, $A \in \mathbf{C}$, $P \in \mathbf{R}$, and domain elements u and v,

$$a^{Can_i(\mathcal{K})} = a,$$

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

$$P^{Can_i(\mathcal{K})}(u, v) = \begin{cases} P^{Can_{i-1}(\mathcal{K})}(u, v), & \text{if } u, v \in \Delta^{Can_{i-1}(\mathcal{K})}, \\ 1, & \text{if } u = w^j_{v,P} \text{ or } v = w^j_{u,P^-}, \\ 0, & \text{otherwise}. \end{cases}$$

We have just defined canonical bag interpretations in a declarative way. Note, however, that they can also be obtained by applying a variant of the restricted chase procedure [35] extended to bags—a procedure where, starting from the ABox, violations of the inclusions in the TBox are successively "repaired" by extending the interpretation of concepts and roles in a minimal way. We now illustrate Definition 32 with an example.

Example 33. Consider the *DL-Lite*$^{\flat}_{\text{CORE}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ specified as

$$\mathcal{T} = \{A \sqsubseteq B,\ B \sqsubseteq \exists P,\ \exists P^- \sqsubseteq \exists R\} \quad \text{and} \quad \mathcal{A} = \{\!|\, A(a) : 2,\ A(b) : 4,\ B(b) : 3,\ P(b, d) : 2,\ P(a, d) : 1,\ R(d, c) : 1 \,|\!\}.$$

The canonical bag interpretation of $\mathcal{K}$ is depicted in Fig. 1. Let us now follow the steps involved in its construction according to Definition 32. Interpretation $Can_0(\mathcal{K})$, enclosed in the shaded area, reflects the bag ABox $\mathcal{A}$: interpretations of concepts are shown by labels of the individuals, which include the multiplicities (e.g., label $A : 2$ of a indicates that $A^{Can_0(\mathcal{K})}(a) = 2$), while interpretations of roles are shown by solid lines (e.g., two lines between b and d labelled by P indicate that $P^{Can_0(\mathcal{K})}(b, d) = 2$). Interpretation $Can_1(\mathcal{K})$ increases the multiplicities of the elements a and b in the interpretation of B (e.g., b has a label $B : 3$ inside the shaded area indicating that $B^{Can_0(\mathcal{K})}(b) = 3$ and a label $B : 4$ outside of it indicating

that $B^{Can_1(\mathcal{K})}(b) = 4$), introduces the anonymous domain elements $w^1_{b,P}$, $w^2_{b,P}$, $w^1_{a,P}$, $w^1_{d,R}$ and $w^2_{d,R}$, and extends the interpretations of the roles accordingly, which is shown by dashed lines. Interpretation $Can_2(\mathcal{K})$ further introduces three anonymous elements and extends the interpretations of the roles. No further changes occur for $i > 2$, and hence $Can(\mathcal{K}) = Can_0(\mathcal{K}) \cup Can_1(\mathcal{K}) \cup Can_2(\mathcal{K})$. $\lhd$

Inspecting Definition 32 and Example 33, we observe that every canonical bag interpretation interprets each concept with a bag of individuals but only with a set of anonymous elements; similarly, multiplicities greater than 1 in the interpretations of roles are possible only for pairs of elements that are both individuals. This is an important property of canonical bag interpretations, which we are going to use in this section.

Note that the canonical bag interpretation satisfies the UNA. The following is another simple and intuitive observation, which holds regardless of the UNA and can be checked by the construction.

Proposition 34. *If a DL-Lite$^{\mathsf{b}}_{\text{CORE}}$ ontology is satisfiable then its canonical bag interpretation is its model.*

In the rest of this section, we show that the canonical bag interpretations of satisfiable *DL-Lite*$^{\mathsf{b}}_{\text{CORE}}$ ontologies are universal models for the class of rooted CQs regardless of the adoption of the UNA. There are two key ideas here, which are similar to the set case, but more subtle.

First, the canonical bag interpretation of a satisfiable *DL-Lite*$^{\mathsf{b}}_{\text{CORE}}$ ontology admits a homomorphism of a special type to every model of the ontology. Such homomorphisms, which we call *multiplicity-preserving on the individuals*, have a hybrid nature: for the concept interpretations, they preserve multiplicities of the interpretations of individuals, but are not required to do so for anonymous elements; similarly, for role interpretations, they preserve multiplicities of the pairs having at least one element being the interpretation of an individual, but are not required to do so for pairs of anonymous elements.

Second, each valuation of a rooted CQ over the canonical bag interpretation sends at least one term of each connected component to the interpretation of an individual. So, since *DL-Lite*$^{\mathsf{b}}_{\text{CORE}}$ does not allow for role inclusions, if two such valuations are different, then they are different on the non-anonymous part of the canonical interpretation. Moreover, the canonical bag interpretation is essentially set-based on the anonymous elements. Therefore, a valuation contributing to the answers and its multiplicity are determined solely by the non-anonymous part of the image of the valuation.

Putting these two ideas together, we can conclude that the composition of a valuation of a rooted CQ over the canonical bag interpretation and a homomorphism from the canonical interpretation to another model that is multiplicity-preserving on the individuals is also a valuation, and its contribution to the certain answers over the latter model is at least as large as the contribution of the former valuation over the canonical interpretation; moreover, different valuations over the canonical interpretation contribute independently to the certain answers over the latter model. This means that the canonical bag interpretation has the smallest possible certain answers to every rooted CQ—that is, by definition, it is the universal model for the class of such queries.

Even though these ideas may seem quite intuitive, their formalisation requires additional machinery, which we do not have yet. The problem is that with the current terminology we cannot unambiguously refer to each particular occurrence of an element in a bag, which is highly desirable for the formalisation. Therefore, we start by introducing new terminology for bags and other bag-based notions.

Definition 35. An *enumerated bag* (or *e-bag*) Θ over a set M is a set of pairs $[c\!:\!m]$ with $c \in M$ and positive integer $m \in \mathbb{N}$, such that if $[c\!:\!m] \in \Theta$ then $[c\!:\!m-1] \in \Theta$ for all $m \in \mathbb{N}$.

There is a straightforward one-to-one correspondence between bags and e-bags, and we call the e-bag corresponding to a bag Ω the *enumerated version* of Ω and denote it Ω^{e}. We can extend this notation to bag interpretations and consider the *enumerated version* I^{e} of a bag interpretation $I = \langle \Delta^I, \cdot^I \rangle$ defined as the pair $\langle \Delta^I, \cdot^{I^{\mathsf{e}}} \rangle$ such that $a^{I^{\mathsf{e}}} = a^I$ for each individual a and $S^{I^{\mathsf{e}}} = (S^I)^{\mathsf{e}}$ for each concept or role S.

The use of enumerated versions of interpretations allows us to refer, in an unambiguous way, to the different occurrences of elements and pairs of elements in the bags corresponding to concepts and roles, respectively. An enumerated homomorphism between two interpretations is then defined as a standard homomorphism that additionally establishes a correspondence for each enumerated tuple of elements in each bag of the relevant bag interpretations.

Definition 36. Given two bag interpretations I and J, an *enumerated homomorphism* (or *e-homomorphism*) from $I^{\mathsf{e}} = \langle \Delta^I, \cdot^{I^{\mathsf{e}}} \rangle$ to $J^{\mathsf{e}} = \langle \Delta^J, \cdot^{J^{\mathsf{e}}} \rangle$ is a family $(h, h_S, \ldots)$, with $S \in \mathbf{C} \cup \mathbf{R}$, of functions

$$\begin{aligned} h &: \ \Delta^I \to \Delta^J, \\ h_S &: \ S^{I^{\mathsf{e}}} \to S^{J^{\mathsf{e}}}, \quad \text{for all } S \in \mathbf{C} \cup \mathbf{R}, \end{aligned}$$

such that

- $h(a^{I^{\mathsf{e}}}) = a^{J^{\mathsf{e}}}$ for each $a \in \mathbf{I}$,
- $h_A([u\!:\!m]) = [h(u)\!:\!\ell]$ for all $A \in \mathbf{C}$ and $[u\!:\!m] \in A^{I^{\mathsf{e}}}$, where ℓ is a number in $\mathbb{N}$,

– $h_P([(u, v):m]) = [(h(u), h(v)):\ell]$ for all $P \in \mathbf{R}$ and $[(u, v):m] \in P^{I^e}$, where ℓ is a number in $\mathbb{N}$.

To handle some cases uniformly, we sometimes write $h_{P^-}([(v, u):m])$ instead of $h_P([(u, v):m])$, for $P \in \mathbf{R}$.

E-homomorphisms have no essential differences with usual homomorphisms because they can send several enumerated tuples to just one without any restrictions. In contrast, the next definition formalises the aforementioned idea of multiplicity preservation: e-homomorphisms that are multiplicity-preserving on the individuals preserve multiplicities on the non-anonymous part of the source interpretation.

Definition 37. An e-homomorphism $(h, h_S, \ldots)$ from $I^e = \langle \Delta^I, \cdot^{I^e} \rangle$ to $J^e = \langle \Delta^J, \cdot^{J^e} \rangle$ is *multiplicity-preserving on individuals* $\mathbf{I}$ if, for each $a \in \mathbf{I}$, the following holds, where $u = a^{I^e}$:

– $h_A([u:m]) \neq h_A([u:\ell])$ for all $A \in \mathbf{C}$ and all $[u:m], [u:\ell] \in A^{I^e}$ with $m \neq \ell$,
– $h_R([(u, v_1):m]) \neq h_R([(u, v_2):\ell])$ for all roles R and all $[(u, v_1):m], [(u, v_2):\ell] \in R^{I^e}$ with $v_1 \neq v_2$ or $m \neq \ell$.

The following lemma then formalises the first key idea about the canonical bag interpretation in terms of e-homomorphisms that are multiplicity-preserving on the individuals.

Lemma 38. *For every DL-Lite$^{\flat}_{\text{CORE}}$ ontology $\mathcal{K}$ and every model I of $\mathcal{K}$ there exists an e-homomorphism from $Can^e(\mathcal{K})$ to I^e that is multiplicity-preserving on* $\mathbf{I}$.

Proof. Consider a *DL-Lite$^{\flat}_{\text{CORE}}$* ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ with the canonical bag interpretation $Can(\mathcal{K}) = \bigcup_{i \geq 0} Can_i(\mathcal{K})$ such that $Can_i(\mathcal{K}) = \langle \Delta^{Can_i(\mathcal{K})}, \cdot^{Can_i(\mathcal{K})} \rangle$ and a model $I = \langle \Delta^I, \cdot^I \rangle$ of $\mathcal{K}$. We first define a witnessing e-homomorphism $(h, h_S, \ldots)$ from $Can^e(\mathcal{K})$ to I^e for the elements in $\Delta^{Can_0(\mathcal{K})}$—that is, for the (interpretations of the) individuals $\mathbf{I}$—that is multiplicity-preserving on $\mathbf{I}$, then extend it to the elements introduced in $Can_1(\mathcal{K})$—that is, to the anonymous elements on the first level of the canonical bag interpretation—and, finally, inductively define it on all other elements.

To begin, recall that $a = a^{Can_0(\mathcal{K})}$ for every $a \in \mathbf{I}$ and let $h : \Delta^{Can_0(\mathcal{K})} \to \Delta^I$ be such that $h(a^{Can_0(\mathcal{K})}) = a^I$ for every $a \in \Delta^{Can_0(\mathcal{K})}$—that is, for every $a \in \mathbf{I}$. We now define function $h_S : S^{Can^e_0(\mathcal{K})} \to S^{I^e}$ for every $S \in \mathbf{C} \cup \mathbf{R}$. For this, consider a tuple of individuals $\mathbf{a}$ such that $S^{Can_0(\mathcal{K})}(\mathbf{a}) = k$, for $k \in \mathbb{N}$—that is, such that $[\mathbf{a}:m] \in S^{Can^e_0(\mathcal{K})}$ for all $m \in \mathbb{N}$ with $m \leq k$. By the definition of $Can_0(\mathcal{K})$, we have that $\mathcal{A}(S(\mathbf{a})) = k$. Since I is a model of $\mathcal{K}$, it satisfies $\mathcal{A}$, and, in particular,

$$S^I(\mathbf{a}^I) \quad \geq \sum_{\mathbf{b} \text{ tuple over } \mathbf{I}:\ \mathbf{b}^I = \mathbf{a}^I} \mathcal{A}(S(\mathbf{b})) \ \geq \ \mathcal{A}(S(\mathbf{a})) \ = \ k.$$

As a result, since $h(\mathbf{a}) = \mathbf{a}^I$, we have that $[h(\mathbf{a}):m] \in S^{I^e}$ for all $m \leq k$; thus we can set $h_S([\mathbf{a}:m]) = [h(\mathbf{a}):m]$ for all such m. By construction, $(h, h_S, \ldots)$ satisfies all conditions stipulated by Definitions 36 and 37, thus it is an e-homomorphism from $Can^e_0(\mathcal{K})$ to I^e that is multiplicity-preserving on $\mathbf{I}$.

We now show the claim for $Can^e_1(\mathcal{K})$. To this end, we extend $(h, h_S, \ldots)$ from the case of $Can^e_0(\mathcal{K})$. By construction of $Can^e_1(\mathcal{K})$, it is enough to extend h_A on $\Delta^{Can_0(\mathcal{K})} = \mathbf{I}$ for $A \in \mathbf{C}$, to define h on the anonymous elements introduced to the domain of $Can^e_1(\mathcal{K})$, and to define h_R on the role links between these anonymous elements and corresponding individuals.

For the extension for atomic concepts, consider an arbitrary concept name $A \in \mathbf{C}$ and an element $u \in \Delta^{Can_0(\mathcal{K})} = \mathbf{I}$. By definition, we have $A^{Can_1(\mathcal{K})}(u) = \mathsf{ccl}_{\mathcal{T}}[u, Can_0(\mathcal{K})](A)$. Therefore, to show that it is possible to extend h_A in such a way that different enumerated elements are sent to different enumerated elements, it suffices to prove that $A^I(u^I) \geq \mathsf{ccl}_{\mathcal{T}}[u, Can_0(\mathcal{K})](A)$. By the definition of concept closure, there must exist a concept C such that $\mathcal{T} \models C \sqsubseteq A$ and $C^{Can_0(\mathcal{K})}(u) = \mathsf{ccl}_{\mathcal{T}}[u, Can_0(\mathcal{K})](A)$. If C is an atomic concept, then $C^{Can_0(\mathcal{K})}(u) = \mathcal{A}(C(u))$ by construction, and $A^I(u^I) \geq \mathcal{A}(C(u))$ follows from the fact that I is a model of both $\mathcal{A}$ and $\mathcal{T}$, and the fact that $\mathcal{T} \models C \sqsubseteq A$ is equivalent to $\mathcal{T} \models^{\flat} C \sqsubseteq A$ (see Statement 2 of Theorem 23). If C is $\exists P$ or $\exists P^-$, then $C^{Can_0(\mathcal{K})}(u) = \sum_{a \in \mathbf{I}} \mathcal{A}(P(u, a))$ or $C^{Can_0(\mathcal{K})}(u) = \sum_{a \in \mathbf{I}} \mathcal{A}(P(a, u))$, respectively, and the argument is analogous.

For the extension for the introduced anonymous elements and corresponding role links, it is enough to consider arbitrary $P \in \mathbf{R}$ and $u \in \Delta^{Can_0(\mathcal{K})} = \mathbf{I}$. By definition, $\Delta^{Can_1(\mathcal{K})}$ extends $\Delta^{Can_0(\mathcal{K})}$ with fresh anonymous elements $w^j_{u,P}$, for $j = 1, \ldots, \mathsf{ccl}_{\mathcal{T}}[u, Can_0(\mathcal{K})](\exists P) - (\exists P)^{Can_0(\mathcal{K})}(u)$, and $P^{Can_1(\mathcal{K})}(u, w^j_{u,P})$ is set to 1 for each of these elements; the same is done for P^- instead of P. We consider only the first set, since the second can be handled in the same way. To extend h to the new elements $w^j_{u,P}$ and h_P to the pairs $(u, w^j_{u,P})$, it suffices to show that $(\exists P)^I(u^I) \geq \mathsf{ccl}_{\mathcal{T}}[u, Can_0(\mathcal{K})](\exists P)$. This can be done in exactly the same way as the proof of $A^I(u^I) \geq \mathsf{ccl}_{\mathcal{T}}[u, Can_0(\mathcal{K})](A)$ in the atomic concept case, by taking $\exists P$ instead of A.

To complete the proof we argue that having an e-homomorphism $(h, h_S, \ldots)$ from $Can^e_i(\mathcal{K})$ to I^e that is multiplicity-preserving on $\mathbf{I}$, for $i \geq 1$, we can extend it to $Can^e_{i+1}(\mathcal{K})$. This can be shown analogously to the case of $Can^e_1(\mathcal{K})$. The only additional observation is that for every $i \geq 1$ and every $S \in \mathbf{C} \cup \mathbf{R}$, bag $S^{Can_{i+1}(\mathcal{K})}$ extends $S^{Can_i(\mathcal{K})}$ with tuples $\mathbf{u}$ of elements that are all anonymous, for which $S^{Can_i(\mathcal{K})}(\mathbf{u}) = 1$ by definition. □

We next move to the formalisation of the second intuitive idea. Recall that a Boolean CQ can be seen as a bag of atoms. Therefore, in the following definition we can consider the *enumerated version* q^{e} of a Boolean CQ q, which is the e-bag of its atoms; this formulation allows us to distinguish different occurrences of atoms in q. Then, an enumerated valuation from a CQ to an interpretation is essentially an e-homomorphism where the CQ is seen as a bag interpretation.

Definition 39. An *enumerated valuation* (*e-valuation*) of a Boolean CQ q over a bag interpretation $I = \langle \Delta^I, \cdot^I \rangle$ is a family $(\nu, \nu_S, \ldots)$, for $S \in \mathbf{C} \cup \mathbf{R}$, of the following functions, where q_S is the subquery of q consisting of all its atoms over S:

$$\begin{aligned} \nu &: \ \mathbf{y} \cup \mathbf{I} \to \Delta^I, \\ \nu_S &: \ q_S^{\mathrm{e}} \to S^{I^{\mathrm{e}}}, \qquad \text{for all } S \in \mathbf{C} \cup \mathbf{R}, \end{aligned}$$

such that

- $\nu(a) = a^I$ for each $a \in \mathbf{I}$,
- $\nu(y) = \nu(t)$ for all equality atoms $y = t$ in q,
- $\nu_A([A(t){:}m]) = [\nu(t){:}\ell]$ for all $A \in \mathbf{C}$ and $[A(t){:}m] \in q_A^{\mathrm{e}}$, where ℓ is a number in $\mathbb{N}$, and
- $\nu_P([P(t_1, t_2){:}m]) = [(\nu(t_1), \nu(t_2)){:}\ell]$ for all $P \in \mathbf{R}$ and $[P(t_1, t_2){:}m] \in q_P^{\mathrm{e}}$, where ℓ is a number in $\mathbb{N}$.

We sometimes write $\nu_{P^-}([P^-(t_2, t_1){:}m])$ instead of $\nu_P([P(t_1, t_2){:}m])$, for $P \in \mathbf{R}$.

It is straightforward to check that the number of e-valuations of a Boolean CQ q over a bag interpretation I is precisely the multiplicity of the empty tuple in the certain answers to q over I.

Proposition 40. *The number of e-valuations of a Boolean CQ q over a bag interpretation I is $q^I(\langle\rangle)$.*

The following lemma formalises the second idea: if two e-valuations over the canonical bag interpretation coincide on all the (enumerated occurrences of the) atoms of a rooted CQ that involve terms evaluating to (the interpretations of) individuals, then they are the same e-valuation.

Lemma 41. *Let q be a rooted Boolean CQ and $\mathcal{K}$ be a DL-Lite$_{\mathrm{CORE}}^{b}$ ontology. If two e-valuations $(\nu^1, \nu_S^1, \ldots)$ and $(\nu^2, \nu_S^2, \ldots)$ of q over $Can(\mathcal{K})$ are different, then there exists an individual $a \in \mathbf{I}$, an atom $S(\mathbf{t})$ in q, a number $m \in \mathbb{N}$, and $i \in \{1, 2\}$ such that $\nu^i(a)$ appears in the tuple $\nu^i(\mathbf{t})$ and $\nu_S^1([S(\mathbf{t}){:}m]) \neq \nu_S^2([S(\mathbf{t}){:}m])$.*

Proof. Let e-valuations $(\nu^1, \nu_{S'}^1, \ldots)$ and $(\nu^2, \nu_{S'}^2, \ldots)$ of q^{e} over $Can^{\mathrm{e}}(\mathcal{K})$ be different, but, for the sake of contradiction, $\nu_S^1([S(\mathbf{t}){:}m]) = \nu_S^2([S(\mathbf{t}){:}m])$ for all $a \in \mathbf{I}$, $[S(\mathbf{t}){:}m] \in q^{\mathrm{e}}$ and $i \in \{1, 2\}$ such that $\nu^i(a)$ is in $\nu^i(\mathbf{t})$. Since the e-valuations are different, there exists $[S(\mathbf{t}){:}m] \in q^{\mathrm{e}}$ such that $\nu_S^1([S(\mathbf{t}){:}m]) \neq \nu_S^2([S(\mathbf{t}){:}m])$. Moreover, by assumption $\mathbf{t}$ consists of only variables. We consider only the case when $S(\mathbf{t})$ is $P(x_1, x_2)$, where $P \in \mathbf{R}$ (and the case when $S(\mathbf{t})$ is $A(x)$ for $A \in \mathbf{C}$ can be handled in the same way).

Boolean CQ q is rooted, so there exists a sequence

$$[R_1(t_0', t_1){:}m_1], [R_2(t_1', t_2){:}m_2], \ldots, [R_k(t_{k-1}', t_k){:}m_k]$$

such that $t_0' \in \mathbf{I}$, $[R_k(t_{k-1}', t_k){:}m_k]$ is either $[P(x_1, x_2){:}m]$ or $[P^-(x_2, x_1){:}m]$, and, for each $j = 1, \ldots, k$, $t_j' \sim t_j$ and either $[R_j(t_{j-1}', t_j){:}m_j]$ is in q^{e}, if R_j is an atomic role, or $[P_j(t_j', t_{j-1}){:}m_j]$ is in q^{e}, if $R_j = P_j^-$ for an atomic role P_j.

We claim that

$$\nu_{R_j}^1([R_j(t_{j-1}', t_j){:}m_j]) = \nu_{R_j}^2([R_j(t_{j-1}', t_j){:}m_j]) \tag{2}$$

for all $j = 1, \ldots, k$ (which, in particular, contradicts our assumption on $[P(x_1, x_2){:}m]$). To prove this claim, suppose for the sake of contradiction that it is not the case, and let $i \in \{1, \ldots, k\}$ be the smallest number such that (2) does not hold. By assumption, we know that $\nu^i(t_{j-1}') \neq \nu^i(a)$ for both $i = 1, 2$ and every $a \in \mathbf{I}$ (therefore, $j \neq 1$, because $t_0' \in \mathbf{I}$). However, since j is the smallest number, $\nu^1(t_{j-1}') = \nu^2(t_{j-1}')$. So, the element $u = \nu^1(t_{j-1}')$ in the canonical bag interpretation $Can(\mathcal{K}) = \bigcup_{i \geq 0} Can_i(\mathcal{K})$ was not introduced in $Can_0(\mathcal{K})$, which implies, by construction, that $(\exists R_j)^{Can(\mathcal{K})}(u) \leq 1$. In fact, since $(\nu^1, \nu_S^1, \ldots)$ is an e-valuation, $(\exists R_j)^{Can(\mathcal{K})}(u) = 1$—that is, there exists just one $v \in \Delta^{Can(\mathcal{K})}$ such that $R_j^{Can(\mathcal{K})}(u, v) \geq 1$, and, moreover, $R_j^{Can(\mathcal{K})}(u, v) = 1$. In other words, $[(u, v){:}1] \in R_j^{Can^{\mathrm{e}}(\mathcal{K})}$, but $[(u, v){:}2] \notin R_j^{Can^{\mathrm{e}}(\mathcal{K})}$. Since $(\nu^1, \nu_S^1, \ldots)$ and $(\nu^2, \nu_S^2, \ldots)$ are e-valuations, $\nu_{R_j}^1$ and $\nu_{R_j}^2$ send $[R_j(t_{j-1}', t_j){:}m_j]$ to some enumerated pairs in $R_i^{Can^{\mathrm{e}}(\mathcal{K})}$, which, by assumption, are different. However, we also know that $\nu^1(t_{j-1}) = \nu^2(t_{j-1})$, so the only possibility for both $\nu_{R_j}^1([R_j(t_{j-1}', t_j){:}m_j])$ and $\nu_{R_j}^2([R_j(t_{j-1}', t_j){:}m_j])$ is $[(u, v){:}1]$. Therefore, our assumption on the existence of j was wrong and (2) indeed holds for all j. In particular, it holds for $j = k$, which contradicts the fact that $\nu_P^1([P(x_1, x_2){:}m]) \neq \nu_P^2([P(x_1, x_2){:}m])$. □

Lemma 41 relies both on the fact that there are no role inclusions in the TBox and on the fact that the CQ is rooted. It is easy to construct counter-examples to this lemma if any one of these requirements is violated.

Having Lemmas 38 and 41 at hand, we are ready to prove that, for satisfiable *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ ontologies, the canonical bag interpretation is the universal model for the class of rooted CQs. The idea is that the composition of an e-valuation of a rooted CQ and an e-homomorphism that is multiplicity-preserving on the individuals is also an e-valuation; moreover, the mapping between e-valuations defined by the e-homomorphism in this way is injective and therefore preserving the size of the domain of the mapping.

Theorem 42. *The canonical bag interpretation $Can(\mathcal{K})$ of a satisfiable DL-Lite$^{\mathsf{b}}_{\mathrm{CORE}}$ ontology $\mathcal{K}$ is a universal model for the class of rooted CQs. The same holds if universality is considered under the UNA.*

Proof. First, note that it is enough to consider only Boolean rooted CQs, because the required property for a non-Boolean rooted CQ $q(\mathbf{x})$ follows from the property for all Boolean CQs obtained from $q(\mathbf{x})$ by replacing variables $\mathbf{x}$ by individuals from $\mathbf{I}$. For a Boolean rooted CQ q it is enough to show that for every *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ ontology $\mathcal{K}$, every model I of $\mathcal{K}$, and every e-valuation $(\nu, \nu_S, \ldots)$ of q over $Can(\mathcal{K})$ there exists a unique e-valuation $(\nu', \nu'_S, \ldots)$ of q over I. By Lemma 38 we know that there exists an e-homomorphism $(h, h_S, \ldots)$ from $Can^{\mathsf{e}}(\mathcal{K})$ to I^{e} that is multiplicity-preserving on $\mathbf{I}$. Therefore, we can take the composition $(\nu, \nu_S, \ldots) \circ (h, h_S, \ldots) = (\nu \circ h, \nu_S \circ h_S, \ldots)$ as $(\nu', \nu'_S, \ldots)$; indeed, the result of this composition is an e-valuation of q over I and, by Lemma 41, this result is unique among all e-valuations of q over $Can(\mathcal{K})$.

Given that $Can(\mathcal{K})$ satisfies the UNA, this result implies also that $Can(\mathcal{K})$ is universal under the UNA for the class of rooted CQs and satisfiable *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ ontologies $\mathcal{K}$. □

The following is an important corollary of Theorem 42 and Corollary 24, which allows us to forget the UNA in the rest of the paper when talking about rooted CQ answering over *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$.

Corollary 43. *The certain answers to rooted CQs over DL-Lite$^{\mathsf{b}}_{\mathrm{CORE}}$ ontologies do not depend on the adoption of the UNA.*

Another important corollary of Theorem 42, the structural properties of rooted CQs, and the definition of the canonical interpretation is that, similarly to the set case, the bag certain answers $q^{\mathcal{K}}$ to a rooted CQ q over a satisfiable *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ ontology $\mathcal{K}$ can be computed over the sub-interpretation $Can_n(\mathcal{K})$ of $Can(\mathcal{K})$ with n depending only on q.

Corollary 44. *If $\mathcal{K}$ is a satisfiable DL-Lite$^{\mathsf{b}}_{\mathrm{CORE}}$ ontology with $Can(\mathcal{K}) = \bigcup_{i \geq 0} Can_i(\mathcal{K})$ and q is a rooted CQ having n atoms, then $q^{\mathcal{K}} = q^{Can_n(\mathcal{K})}$.*

8. Rewritability of rooted conjunctive queries over *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$

First-order rewritability of CQs is a key property of *DL-Lite* query answering under set semantics. In this section, we show rewritability of rooted CQs over *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ to BCALC (recall that by Corollary 43 this result is agnostic to the adoption of the UNA).

8.1. Non-rewritability to BCALC unions of conjunctive queries

In the case of set semantics, the target language for rewritings is that of unions of conjunctive queries (UCQs). There are two natural counterparts to UCQs in the bag setting: BCALC maximal union of CQs and BCALC arithmetic union of CQs. Our first result is negative and in stark contrast to the set case: in general, rewriting to either of these classes of BCALC queries is not possible, even over *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$.

Proposition 45. *Rooted CQs are rewritable neither to* BCALC *maximal nor to* BCALC *arithmetic unions of CQs over DL-Lite$^{\mathsf{b}}_{\mathrm{CORE}}$.*

Proof. We first prove the claim for BCALC maximal unions of CQs. Consider the satisfiable *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over atomic concepts A and B, and atomic role P with TBox $\mathcal{T} = \{A \sqsubseteq \exists P,\ \exists P^- \sqsubseteq B\}$, and ABox $\mathcal{A} = \{\!|\, A(a) : 3,\ P(a,b) : 2,\ B(b) : 3 \,|\!\}$. Let also $q(x) = \exists y.\, P(x,y) \wedge B(y)$. Then, $Can(\mathcal{K})$ is the interpretation with domain $\Delta^{Can(\mathcal{K})} = \mathbf{I} \cup \{w^1_{a,P}\}$ that interprets individuals by themselves and predicates as follows:

$$A^{Can(\mathcal{K})} = \{\!|\, a : 3 \,|\!\}, \qquad P^{Can(\mathcal{K})} = \{\!|\, (a,b) : 2,\ (a, w^1_{a,P}) : 1 \,|\!\}, \qquad B^{Can(\mathcal{K})} = \{\!|\, b : 3,\ w^1_{a,P} : 1 \,|\!\}.$$

Evaluating q over $Can(\mathcal{K})$, we get $q^{Can(\mathcal{K})}(a) = 7$ for individual a. Assume for the sake of contradiction that there exists a rewriting of q to a BCALC maximal union $\Phi(x)$ of CQs with respect to $\mathcal{T}$. By the semantics of the BCALC maximal union, there exists a BCALC CQ q_0 in Φ with $q_0^{\mathcal{A}}(a) = q^{Can(\mathcal{K})}(a)$. Observe that $\mathcal{A}$ contains three distinct assertions with multiplicities 3, 2, and 3. Hence, whenever there is a valuation of the terms of q_0 that maps an atom of q_0 to one of these assertions, the multiplicity is either 2 or 3. Because q_0 is a CQ, every valuation of q_0 contributes to $q_0^{\mathcal{A}}(a)$ a multiplicity that

is a multiple of 2 or 3. Since 7 is prime, there can be no valuation contributing multiplicity 7. Moreover, there are only two ways to get 7 as an instance of a polynomial with coefficients 2 and 3, namely, $2+2+3$ and $2\times 2+3$. For the former sum, this means that there exist three distinct valuations contributing to $q_0^{\mathcal{A}}(a)$ with multiplicities 2, 2, and 3, respectively, which is impossible given the fact that, to get 2, query q_0 must be set equal to $\exists y.\, P(x,y)$, which excludes the possibility of getting the multiplicity 3. For the latter sum, there must exist two distinct valuations contributing to $q_0^{\mathcal{A}}(a)$ with multiplicities 4 and 3, respectively, which is again impossible given the fact that, to get 4, query q_0 must be set to $\exists y, z.\, P(x,y) \wedge P(x,z)$ or to $\exists y.\, P(x,y) \wedge P(x,y)$, which in either case excludes the possibility of getting the multiplicity 3.

We now prove the claim for BCALC arithmetic unions of CQs. Consider the *DL-Lite*$_{\text{CORE}}^{\text{b}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ over atomic concepts A and B with TBox $\mathcal{T} = \{A \sqsubseteq B\}$ and ABox $\mathcal{A} = \{\!\{ A(a):3,\ A(b):2,\ B(a):2,\ B(b):3 \}\!\}$, and the rooted query $q(x) = B(x)$. Observe that $q^{\mathcal{K}} = B^{Can(\mathcal{K})} = \{\!\{ a:3,\ b:3 \}\!\}$. We have the following possible cases for a BCALC arithmetic union $\Phi(x)$ of CQs:

- if $\Phi(x) = A(x)$, then $\Phi^{\mathcal{A}} = \{\!\{ a:3,\ b:2 \}\!\}$;
- if $\Phi(x) = B(x)$, then $\Phi^{\mathcal{A}} = \{\!\{ a:2,\ b:3 \}\!\}$;
- if $\Phi(x) = A(x) \wedge B(x)$, then $\Phi^{\mathcal{A}} = \{\!\{ a:6,\ b:6 \}\!\}$;
- if $\Phi(x) = A(x) \vee B(x)$, then $\Phi^{\mathcal{A}} = \{\!\{ a:5,\ b:5 \}\!\}$;
- if $\Phi(x)$ contains $A(x)$ at least twice, then $\Phi^{\mathcal{A}}(a) \geq 9$; and
- if $\Phi(x)$ contains $B(x)$ at least twice, then $\Phi^{\mathcal{A}}(b) \geq 9$.

So, none of these cases satisfies $\Phi^{\mathcal{A}} = B^{Can(\mathcal{K})}$, which is required for a rewriting of q with respect to $\mathcal{T}$. □

8.2. General ideas for rewritability to BCALC queries

Next, we show that rooted CQs are rewritable to a richer fragment of BCALC over *DL-Lite*$_{\text{CORE}}^{\text{b}}$, which also features the operation of difference. This ensures LogSpace membership in data complexity of query answering. Our rewriting algorithm is inspired by that of Kikot et al. [37] for *DL-Lite*$_{\mathcal{R}}$. The key observation behind our approach is that, for a *DL-Lite*$_{\text{CORE}}^{\text{b}}$ ontology $\mathcal{K}$ and a rooted CQ $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x},\mathbf{y})$, the bag answers to q over the canonical bag interpretation $Can(\mathcal{K})$, which, by Theorem 42, coincide with the bag certain answers to q over $\mathcal{K}$, can be partitioned as

$$q^{Can(\mathcal{K})} \;=\; \biguplus_{\mathbf{z} \subseteq \mathbf{y}} [q, \mathbf{z}]^{Can(\mathcal{K})}, \tag{3}$$

where each $[q,\mathbf{z}]^{Can(\mathcal{K})}$ is the bag of answers to q over $Can(\mathcal{K})$ supported by valuations of $q(\mathbf{x})$ over $Can(\mathcal{K})$ that send all the variables in a subset $\mathbf{z}$ of the variables $\mathbf{y}$ to anonymous elements and all other variables to individuals. Next, we define such partitions formally.

Definition 46. Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x},\mathbf{y})$ be a rooted CQ and let $\mathcal{K}$ be a *DL-Lite*$_{\text{CORE}}^{\text{b}}$ ontology. Given a subset $\mathbf{z}$ of variables $\mathbf{y}$, let $[q,\mathbf{z}]^{Can(\mathcal{K})}$ be the bag of tuples over $\mathbf{I}$ such that, for each tuple $\mathbf{a}$ of individuals,

$$[q,\mathbf{z}]^{Can(\mathcal{K})}(\mathbf{a}) \;=\; \sum_{\lambda \in \Lambda_{\mathbf{z}}} \;\prod_{S(\mathbf{t}) \text{ in } \phi(\mathbf{x},\mathbf{y})} S^{Can(\mathcal{K})}(\lambda(\mathbf{t})),$$

where $\Lambda_{\mathbf{z}}$ is the set of valuations $\lambda : \mathbf{x} \cup \mathbf{y} \cup \mathbf{I} \to \Delta^{Can(\mathcal{K})}$ such that $\lambda(\mathbf{x}) = \mathbf{a}$, $\lambda(a) = a$ for each $a \in \mathbf{I}$, $\lambda(x) = \lambda(t)$ for each $x = t$ in $\phi(\mathbf{x},\mathbf{y})$, $\lambda(z)$ is an anonymous element for each $z \in \mathbf{z}$, and $\lambda(y) \in \mathbf{I}$ for each $y \in \mathbf{y} \setminus \mathbf{z}$.

Following this key observation, given a rooted CQ, the rewriting algorithm first constructs a set of BCALC queries each one accounting for the bag $[q,\mathbf{z}]^{Can(\mathcal{K})}$ for a subset $\mathbf{z}$ and then adjoins them using the arithmetic union to produce the actual rewriting, which is still a BCALC query. In particular, every subset $\mathbf{z}$ is processed along the following three steps:

1. $\mathbf{z}$ is checked for $\mathcal{T}$-realisability—that is, whether the corresponding subquery can be folded to the anonymous part of a canonical bag interpretation—and disregarded from consideration in (3) if the check fails;
2. each connected component of the subquery corresponding to $\mathbf{z}$ is replaced in the query of $[q,\mathbf{z}]^{Can(\mathcal{K})}$ by a single representative role atom; and
3. each concept atom and each representative atom is rewritten to a BCALC query that takes into account the TBox and the fact that $\mathbf{z}$ should be sent to anonymous elements.

In the following three sections we formalise each of these steps and prove their correctness.

8.3. Step 1: checking for realisability

In the first step, every subset $\mathbf{z}$ of existentially quantified variables $\mathbf{y}$ in a rooted CQ $q(\mathbf{x})$ is checked for $\mathcal{T}$-realisability. Intuitively, $\mathbf{z}$ is $\mathcal{T}$-realisable if the subquery of $q(\mathbf{x})$ induced by $\mathbf{z}$ can be "folded" into the anonymous forest-shaped part of

Fig. 2. Gaifman graph of CQ in Example 47 and its subgraph corresponding to $\{y_2, \ldots, y_5\}$.

$Can(\mathcal{K})$ for some ontology $\mathcal{K}$ having TBox $\mathcal{T}$. Therefore, non-realisable $\mathbf{z}$ cannot contribute to partitioning (3) and their associated subqueries can be disregarded for the purpose of query rewriting. To provide the formal definition of $\mathcal{T}$-realisability, we need to introduce some preliminary definitions and notations.

Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and $\mathcal{T}$ be a *DL-Lite*$_{\text{CORE}}$ TBox.

First, recall that the Gaifman graph G of q has the equivalence class $\tilde{t}$ for each term $t \in \mathbf{x} \cup \mathbf{y} \cup \mathbf{I}$ in ϕ as a node, and an edge $\{\tilde{t}_1, \tilde{t}_2\}$ for each atom $P(t_1, t_2)$ in ϕ. A subset $\mathbf{z}$ of $\mathbf{y}$ is *equality-consistent* if $\tilde{z} \subseteq \mathbf{z}$ for every $z \in \mathbf{z}$.

Second, each equality-consistent subset $\mathbf{z}$ of $\mathbf{y}$ has a corresponding subgraph $G|_{\mathbf{z}}$ of Gaifman graph G—that is, the subgraph on the set of nodes $\{\tilde{z} \mid z \in \mathbf{z}\}$. This subgraph may have several connected components; a subset $\mathbf{v}$ of $\mathbf{z}$ is *maximally connected* if it is also equality-consistent and the subgraph of $G|_{\mathbf{z}}$ corresponding to $\mathbf{v}$ is a connected component of $G|_{\mathbf{z}}$. Therefore, $\mathbf{z}$ can be partitioned to its maximal connected subsets.

Third, for every maximally connected subset $\mathbf{v}$ of an equality-consistent $\mathbf{z} \subseteq \mathbf{y}$ and for each individual a, we define the query

$$q^a_{\mathbf{v}} = \exists \mathbf{v}, \mathbf{v}'.\, \phi_{\mathbf{v}} \wedge \bigwedge_{t \in \mathbf{t}_{\mathbf{v}}} (t = a) \;\wedge\; \bigwedge_{v \in \mathbf{v}} (v \neq a),$$

where $\phi_{\mathbf{v}}$ is the conjunction of all atoms in ϕ mentioning at least one variable in $\mathbf{v}$, $\mathbf{t}_{\mathbf{v}}$ is the set of all terms appearing in $\phi_{\mathbf{v}}$ but not in $\mathbf{v}$, and $\mathbf{v}' = (\mathbf{x} \cup \mathbf{y}) \cap \mathbf{t}_{\mathbf{v}}$. This query is a Boolean CQ, except that it may have equalities of two individuals and inequalities of terms. The semantics of CQs in Definition 12 can be extended to such queries in a straightforward way: the additional requirement on each valuation λ contributing to the sum is that λ should satisfy $\lambda(x) \neq \lambda(t)$ for each inequality atom $(x \neq t)$ in the query (and the requirement for equalities of individuals is the same as for usual equalities).

Example 47. Consider the rooted CQ

$$q(x) = \exists \mathbf{y}.\, P(x, y_1) \wedge P(x, y_2) \wedge P(x, y_3) \wedge P(x, y_4) \wedge P(d, y_4) \wedge R(y_3, y_5) \wedge (y_1 = c) \wedge (y_3 = y_4)$$

with $\mathbf{y} = y_1, \ldots, y_5$ over atomic roles P and R, and its Gaifman graph, depicted in Fig. 2a. Observe that no subset of $\mathbf{y}$ containing y_1 is equality-consistent because q contains equality $y_1 = c$ and c is not in $\mathbf{y}$. Furthermore, every subset of $\mathbf{y}$ containing y_3 or y_4 but not both is also not equality-consistent. However, $\mathbf{z} = \{y_2, \ldots, y_5\}$ is equality-consistent. The corresponding subgraph $G|_{\mathbf{z}}$ is depicted in Fig. 2b. It has two connected components, and therefore $\mathbf{z}$ partitions into two maximally connected subsets, $\mathbf{v}_1 = \{y_2\}$ and $\mathbf{v}_2 = \{y_3, y_4, y_5\}$. For the first, we have $\phi_{\mathbf{v}_1} = P(x, y_2)$, $\mathbf{t}_{\mathbf{v}_1} = \{x\}$, and, for an individual a,

$$q^a_{\mathbf{v}_1} = \exists y_2, x.\, P(x, y_2) \wedge (x = a) \wedge (y_2 \neq a). \tag{4}$$

For the second, we have $\phi_{\mathbf{v}_2} = P(x, y_3) \wedge P(x, y_4) \wedge P(d, y_4) \wedge R(y_3, y_5) \wedge (y_3 = y_4)$, $\mathbf{t}_{\mathbf{v}_2} = \{x, d\}$, and

$$q^d_{\mathbf{v}_2} = \exists \mathbf{v}_2, x.\, \phi_{\mathbf{v}_2} \wedge (x = d) \wedge (d = d) \wedge (y_3 \neq d) \wedge (y_4 \neq d) \wedge (y_5 \neq d). \quad \lhd \tag{5}$$

We are now ready to define the notion of realisability, which is inspired by the notion of tree witnesses proposed by Kikot et al. [37] in the context of query rewriting over *DL-Lite*$_{\mathcal{R}}$.

Definition 48. Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and $\mathcal{T}$ be a *DL-Lite*$_{\text{CORE}}$ TBox. A subset $\mathbf{z}$ of variables $\mathbf{y}$ is *$\mathcal{T}$-realisable* if it is equality-consistent and every maximally connected subset $\mathbf{v}$ of $\mathbf{z}$ satisfies the following conditions, where, as before, $\phi_{\mathbf{v}}$ is the conjunction of all atoms in ϕ mentioning at least one variable in $\mathbf{v}$ and $\mathbf{t}_{\mathbf{v}}$ is the set of all terms appearing in $\phi_{\mathbf{v}}$ but not in $\mathbf{v}$:

1. there is at most one individual in $\mathbf{t}_{\mathbf{v}}$;
2. all the atoms in $\phi_{\mathbf{v}}$ mentioning terms in $\mathbf{t}_{\mathbf{v}}$ are over the same atomic role $P_{\mathbf{v}}$ and have these terms at the same position $p_{\mathbf{v}} \in \{1, 2\}$;
3. $(q^a_{\mathbf{v}})^{Can(\mathcal{K}_{\mathbf{v}})}(\langle\rangle) \geq 1$, where a is the individual in $\mathbf{t}_{\mathbf{v}}$ if it exists or a fresh individual otherwise, and, for a fresh individual b, $\mathcal{K}_{\mathbf{v}}$ is $\langle \mathcal{T}, \{\!|\, P_{\mathbf{v}}(a, b) : 1 \,|\!\}\rangle$ if $p_{\mathbf{v}} = 1$ or $\langle \mathcal{T}, \{\!|\, P_{\mathbf{v}}(b, a) : 1 \,|\!\}\rangle$ if $p_{\mathbf{v}} = 2$.

Note that realisability checking is clearly decidable. In particular, by Corollary 44, Condition 3 can be checked over a bounded fragment of the relevant canonical bag interpretation.

Example 49. Consider the *DL-Lite*$_{\text{CORE}}$ TBox $\mathcal{T} = \{A \sqsubseteq \exists P,\ \exists P^- \sqsubseteq \exists R\}$ over atomic concept A and atomic roles P and R, the rooted CQ $q(x)$ specified in Example 47, and the equality-consistent subset $\mathbf{z} = \{y_2, \dots, y_5\}$ with two maximally connected subsets, $\mathbf{v}_1 = \{y_2\}$ and $\mathbf{v}_2 = \{y_3, y_4, y_5\}$. Subset $\mathbf{z}$ is $\mathcal{T}$-realisable. To see this, note first that Conditions 1 and 2 are immediately satisfied by both $\mathbf{v}_1$ and $\mathbf{v}_2$, with no individuals in $\mathbf{t}_{\mathbf{v}_1}$, one individual d in $\mathbf{t}_{\mathbf{v}_2}$, $P_{\mathbf{v}_1} = P_{\mathbf{v}_2} = P$ and $p_{\mathbf{v}_1} = p_{\mathbf{v}_2} = 1$. To verify Condition 3 for $\mathbf{v}_1$, note that $\mathcal{K}_{\mathbf{v}_1} = \langle \mathcal{T}, \{\!|\, P(a,b) : 1 \,|\!\} \rangle$ for fresh individuals a and b, and therefore

$$A^{Can(\mathcal{K}_{\mathbf{v}_1})} = \emptyset, \qquad P^{Can(\mathcal{K}_{\mathbf{v}_1})} = \{\!|\, (a,b) : 1 \,|\!\}, \qquad R^{Can(\mathcal{K}_{\mathbf{v}_1})} = \{\!|\, (b, w^1_{b,R}) : 1 \,|\!\};$$

so $q^a_{\mathbf{v}_1}$, defined in (4), evaluates to 1 for $\langle\rangle$ over $Can(\mathcal{K}_{\mathbf{v}_1})$. Condition 3 for $\mathbf{v}_2$ can be verified in exactly the same way, except that $\mathcal{K}_{\mathbf{v}_2}$ and $Can(\mathcal{K}_{\mathbf{v}_2})$ have d instead of a, and $q^d_{\mathbf{v}_2}$ is defined in (5). ◁

The next lemma establishes the key property of realisability: for any ABox, a non-realisable $\mathbf{z}$ cannot contribute to the partitioning (3) of the bag query answers over the canonical bag interpretation.

Lemma 50. *Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and let $\mathcal{T}$ be a DL-Lite*$_{\text{CORE}}$ *TBox. If a subset $\mathbf{z}$ of $\mathbf{y}$ is not $\mathcal{T}$-realisable, then $[q, \mathbf{z}]^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)} = \emptyset$ for every bag ABox $\mathcal{A}$.*

Proof. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{A}$ is an arbitrary bag ABox, and $\mathbf{z}$ be a subset of $\mathbf{y}$ that is not $\mathcal{T}$-realisable. By Definition 48, either $\mathbf{z}$ is not equality-consistent, or there exists a maximally connected subset of $\mathbf{z}$ for which one of the three conditions does not hold.

It is straightforward to check that $[q, \mathbf{z}]^{Can(\mathcal{K})} = \emptyset$ if $\mathbf{z}$ is not equality-consistent: indeed, in this case the CQ contains an equality atom $(y = t)$ with $y \in \mathbf{z}$ and $t \notin \mathbf{z}$, which cannot be satisfied over the canonical bag interpretation $Can(\mathcal{K})$ by any valuation contributing to $[q, \mathbf{z}]^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)}$ because y is mapped to anonymous elements of $Can(\mathcal{K})$ by such valuations, whereas t is mapped to individuals.

Similarly, if there is a maximally connected $\mathbf{v} \subseteq \mathbf{z}$ for which Condition 1 does not hold, then $[q, \mathbf{z}]^{Can(\mathcal{K})} = \emptyset$: indeed, if $\mathbf{t}_{\mathbf{v}}$ contains two different individuals, then no contributing valuation can send $\mathbf{v}$ to anonymous elements, because $\mathbf{v}$ is connected, while $Can(\mathcal{K})$ has independent tree-shaped anonymous parts connected to these two individuals.

Next, if there is a maximally connected $\mathbf{v} \subseteq \mathbf{z}$ for which Condition 2 does not hold, then again $[q, \mathbf{z}]^{Can(\mathcal{K})} = \emptyset$: indeed, if $\phi_{\mathbf{v}}$ contains two atoms over different atomic roles with terms in $\mathbf{t}_{\mathbf{v}}$ or two atoms over the same atomic role but with the terms in $\mathbf{t}_{\mathbf{v}}$ in different positions, then no contributing valuation can send $\mathbf{v}$ to anonymous elements by the same reasons as in the case of Condition 1 and the fact that no anonymous element is connected to another element in two different ways in $Can(\mathcal{K})$ by construction.

Finally, consider the case when there is a maximally connected $\mathbf{v} \subseteq \mathbf{z}$ for which Conditions 1 and 2 hold but Condition 3 does not. Reasoning as in the previous two cases, we conclude that all variables in $\mathbf{v}$ are sent by every contributing valuation to anonymous elements generated by the same individual in the canonical bag interpretation, and all terms in $\mathbf{t}_{\mathbf{v}}$ are sent to this individual. By Condition 2, every atom in $\phi_{\mathbf{v}}$ that has a term in $\mathbf{t}_{\mathbf{v}}$ is over atomic role $P_{\mathbf{v}}$ and has the term in position $p_{\mathbf{v}}$. Therefore, $(q^a_{\mathbf{v}})^{Can(\mathcal{K}_{\mathbf{v}})}(\langle\rangle)$ is the factor corresponding to $\phi_{\mathbf{v}}$ in the multiplicity of every valuation satisfying the conditions of $[q, \mathbf{z}]^{Can(\mathcal{K})}$, and $(q^a_{\mathbf{v}})^{Can(\mathcal{K}_{\mathbf{v}})}(\langle\rangle) = 0$ means that there are no valuations with non-zero contribution. □

8.4. Step 2: replacing subqueries with representatives

Consider a $\mathcal{T}$-realisable subset $\mathbf{z}$ of the variables $\mathbf{y}$ in a rooted CQ $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ for a *DL-Lite*$_{\text{CORE}}$ TBox $\mathcal{T}$. As established in the previous section, maximally connected subsets $\mathbf{v}$ of $\mathbf{z}$ are always disjoint, so the corresponding conjunctions of atoms $\phi_{\mathbf{v}}$ from ϕ that mention at least one variable in $\mathbf{v}$ do not share atoms either. So, $q(\mathbf{x})$ can be put into the following form for the given $\mathbf{z}$ in a unique way, where $\psi_{\mathbf{z}}$ consists of all atoms in $\phi(\mathbf{x}, \mathbf{y})$ not contained in any $\phi_{\mathbf{v}}$:

$$q(\mathbf{x}) \quad = \quad \exists \mathbf{y}.\, \psi_{\mathbf{z}} \wedge \bigwedge_{\substack{\text{maximally} \\ \text{connected } \mathbf{v} \subseteq \mathbf{z}}} \phi_{\mathbf{v}}. \tag{6}$$

The rewriting step introduced in this section can be informally explained as follows. By definition, every valuation contributing to bag $[q, \mathbf{z}]^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)}$ with an arbitrary bag ABox $\mathcal{A}$ sends $\mathbf{z}$ to anonymous elements of the canonical bag interpretation. So, since each $\phi_{\mathbf{v}}$ is connected and mentions a variable from $\mathbf{z}$ in each of its atoms as well as a term outside $\mathbf{z}$, conjunction $\phi_{\mathbf{v}}$ contributes to every valuation for $[q, \mathbf{z}]^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)}$ a multiplicity of at most 1 (recall that the canonical bag interpretation involves only multiplicities 0 and 1 in the anonymous part). Moreover, whether $\phi_{\mathbf{v}}$ can be appropriately embedded into $Can(\langle \mathcal{T}, \mathcal{A} \rangle)$ depends solely on whether $\phi_{\mathbf{v}}$ can be embedded into the canonical bag interpretation of the ontology consisting of $\mathcal{T}$ and a prototypical bag ABox that comprises a single assertion with multiplicity 1 in such a way

that all terms of $\phi_{\mathbf{v}}$ that are outside $\mathbf{z}$ are sent to one of the individuals of the assertion. Therefore, when computing $[q, \mathbf{z}]^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)}$, the whole $\phi_{\mathbf{v}}$ can be replaced in q by just a single role atom mentioning a representative variable and a term not in $\mathbf{v}$ as well as several equalities identifying all the terms not in $\mathbf{v}$.

Next, we formalise this idea and prove its correctness.

Definition 51. Let $q(\mathbf{x}) = \exists \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and let $\mathcal{T}$ be a *DL-Lite*$_{\text{CORE}}$ TBox. For every $\mathcal{T}$-realisable subset $\mathbf{z}$ of $\mathbf{y}$, let

$$q_{\mathbf{z}}(\mathbf{x}) \quad = \quad \exists \mathbf{y}_{\mathbf{z}}, \mathbf{z}'. \psi_{\mathbf{z}} \wedge \bigwedge_{\substack{\text{maximally} \\ \text{connected } \mathbf{v} \subseteq \mathbf{z}}} \left(\alpha_{\mathbf{v}} \wedge \bigwedge_{\substack{y \in (\mathbf{x} \cup \mathbf{y}) \cap \mathbf{t}_{\mathbf{v}}, \\ t \in \mathbf{t}_{\mathbf{v}}}} (y = t) \right), \tag{7}$$

where $\psi_{\mathbf{z}}$ is defined as in (6), $\mathbf{y}_{\mathbf{z}}$ is the set of all variables in $\mathbf{y}$ appearing in $\psi_{\mathbf{z}}$, atom $\alpha_{\mathbf{v}}$ is defined as follows, for every maximally connected $\mathbf{v} \subseteq \mathbf{z}$ with terms $\mathbf{t}_{\mathbf{v}}$, role $P_{\mathbf{v}}$, and position $p_{\mathbf{v}}$ defined as in Definition 48, as well as for a term $t \in \mathbf{t}_{\mathbf{v}}$ and a fresh variable $y_{\mathbf{v}}$:

$$\alpha_{\mathbf{v}} = \begin{cases} P_{\mathbf{v}}(t, y_{\mathbf{v}}), & \text{if } p_{\mathbf{v}} = 1, \\ P_{\mathbf{v}}(y_{\mathbf{v}}, t), & \text{if } p_{\mathbf{v}} = 2, \end{cases}$$

and $\mathbf{z}'$ is the set of all variables $y_{\mathbf{v}}$ introduced for the atoms $\alpha_{\mathbf{v}}$.

Formally speaking, this definition is non-deterministic, because the terms t in the atoms $\alpha_{\mathbf{v}}$ are chosen arbitrarily from $\mathbf{t}_{\mathbf{v}}$. However, this choice does not influence the semantics of $q_{\mathbf{z}}(\mathbf{x})$ because of the equalities introduced in (7). Therefore, we assume that $q_{\mathbf{z}}(\mathbf{x})$ is well-defined.

Example 52. Consider the rooted CQ $q(x)$, *DL-Lite*$_{\text{CORE}}$ TBox $\mathcal{T}$, and $\mathcal{T}$-realisable subset $\mathbf{z}$ with maximally connected subsets $\mathbf{v}_1$ and $\mathbf{v}_2$ introduced in Examples 47 and 49. We know that $\mathbf{t}_{\mathbf{v}_1} = \{x\}$, $\mathbf{t}_{\mathbf{v}_2} = \{x, d\}$, $P_{\mathbf{v}_1} = P_{\mathbf{v}_2} = P$, and $p_{\mathbf{v}_1} = p_{\mathbf{v}_2} = 1$, so $\alpha_{\mathbf{v}_1} = P(x, y_{\mathbf{v}_1})$, $\alpha_{\mathbf{v}_2} = P(x, y_{\mathbf{v}_2})$, and

$$q_{\mathbf{z}}(x) = \exists y_1, y_{\mathbf{v}_1}, y_{\mathbf{v}_2}. P(x, y_1) \wedge (y_1 = c) \ \wedge \ \Big(P(x, y_{\mathbf{v}_1}) \wedge (x = x) \Big) \ \wedge \ \Big(P(x, y_{\mathbf{v}_2}) \wedge (x = d) \wedge (x = x) \Big).$$

In contrast to q, CQ $q_{\mathbf{z}}$ does not contain any R atoms. Indeed, such atoms are not needed: sending y_3 and y_4 to the same anonymous element w and given that w must be a P-successor of d in the canonical bag interpretation, it follows that w must have an R-successor due to inclusion $\exists P^- \sqsubseteq \exists R$ in $\mathcal{T}$. Thus, we only need a representative $y_{\mathbf{v}_1}$ for y_3 and y_4. $\lhd$

In the second step, our algorithm generates the query $q_{\mathbf{z}}(\mathbf{x})$ for each $\mathcal{T}$-realisable subset $\mathbf{z}$ of $\mathbf{y}$. The next lemma justifies this step by showing that we can consider in (3) only $\mathcal{T}$-realisable $\mathbf{z}$ and replace q with $q_{\mathbf{z}}$ for each such $\mathbf{z}$.

Lemma 53. *Let $q(\mathbf{x}) = \exists \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and let $\mathcal{T}$ be a DL-Lite*$_{\text{CORE}}$ *TBox. If a subset $\mathbf{z}$ of $\mathbf{y}$ is $\mathcal{T}$-realisable then $[q, \mathbf{z}]^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)} = [q_{\mathbf{z}}, \mathbf{z}']^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)}$ for every bag ABox $\mathcal{A}$, where $\mathbf{z}'$ as in Definition 51.*

Proof. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{A}$ is an arbitrary bag ABox, and let $\mathbf{z}$ be a $\mathcal{T}$-realisable subset of $\mathbf{y}$. First, for every valuation $\lambda \in \Lambda_{\mathbf{z}}$ as in Definition 46 of $[q, \mathbf{z}]^{Can(\mathcal{K})}$, let $\lambda_{\mathbf{z}}$ be the valuation of $\mathbf{x} \cup \mathbf{y}_{\mathbf{z}} \cup \mathbf{z}' \cup \mathbf{I}$ to $\Delta^{Can(\mathcal{K})}$ that is the same as λ on $\mathbf{I}$ and all the terms of $\psi_{\mathbf{z}}$, and, for each maximally connected subset $\mathbf{v}$ of $\mathbf{z}$, $\lambda_{\mathbf{z}}(y_{\mathbf{v}}) = \lambda(y)$, where y is a variable in $\mathbf{v}$ such that $\phi_{\mathbf{v}}$ contains an atom $P_{\mathbf{v}}(t, y)$ or an atom $P_{\mathbf{v}}(y, t)$, for some $t \in \mathbf{t}_{\mathbf{v}}$; in other words, $\lambda_{\mathbf{z}}$ is the same as λ except that each $\mathbf{v}$ is replaced by its representative with the corresponding value. It suffices to show that every valuation $\lambda \in \Lambda_{\mathbf{z}}$ contributes to $[q, \mathbf{z}]^{Can(\mathcal{K})}$ the same multiplicity as $\lambda_{\mathbf{z}}$ contributes to $[q_{\mathbf{z}}, \mathbf{z}']^{Can(\mathcal{K})}$.

Consider first a valuation $\lambda \in \Lambda_{\mathbf{z}}$ contributing to $[q, \mathbf{z}]^{Can(\mathcal{K})}$ a non-zero multiplicity. By construction, $Can(\mathcal{K})$ interprets all concepts with multiplicity at most 1 on the anonymous elements and all roles with multiplicity at most 1 on all pairs with at least one anonymous element. Thus, the contribution of λ is equal to the contribution of (the relevant part of) λ to the evaluation of $\psi_{\mathbf{z}}$. So, it is enough to show that, for each maximally connected $\mathbf{v} \subseteq \mathbf{z}$, $\lambda_{\mathbf{z}}(\alpha_{\mathbf{v}})$ has multiplicity 1 in $Can(\mathcal{K})$, and all the equalities on $\mathbf{t}_{\mathbf{v}}$ introduced in (7) to $q_{\mathbf{z}}$ hold for $\lambda_{\mathbf{z}}$ (i.e., for λ, because they coincide on $\mathbf{t}_{\mathbf{v}}$). The former holds immediately by construction of $\lambda_{\mathbf{z}}$, while the latter follows from the fact that $\mathbf{v}$ are connected in q and sent to anonymous elements by λ, while $Can(\mathcal{K})$ is tree-shaped on the anonymous elements.

Consider now a valuation $\lambda \in \Lambda_{\mathbf{z}}$ such that $\lambda_{\mathbf{z}}$ contributes to $[q_{\mathbf{z}}, \mathbf{z}']^{Can(\mathcal{K})}$ a non-zero multiplicity. Similarly to the previous case, the contribution of $\lambda_{\mathbf{z}}$ is equal to the contribution of (the relevant part of) $\lambda_{\mathbf{z}}$ to the evaluation of $\psi_{\mathbf{z}}$. So, it is enough to show that, for each maximally connected $\mathbf{v} \subseteq \mathbf{z}$, $\lambda(\phi_{\mathbf{v}})$ has multiplicity 1 in $Can(\mathcal{K})$. However, this follows from the fact that $\mathbf{z}$ is $\mathcal{T}$-realisable (in particular, Condition 3) and the fact that $\lambda_{\mathbf{z}}(\alpha_{\mathbf{v}})$ has multiplicity 1 in $Can(\mathcal{K})$. $\square$

8.5. Step 3: rewriting atoms to BCALC queries

In the last step, our algorithm first transforms each CQ $q_{\mathbf{z}}(\mathbf{x})$ computed in Step 2 for a $\mathcal{T}$-realisable $\mathbf{z}$ to a BCALC query $\Phi_{\mathbf{z}}(\mathbf{x})$ satisfying equality $[q_{\mathbf{z}}, \mathbf{z}']^{Can(\langle\mathcal{T},\mathcal{A}\rangle)} = \Phi_{\mathbf{z}}^{\mathcal{A}}$ and then constructs the final rewriting of the input CQ $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ with respect to the input *DL-Lite*$_{\text{CORE}}$ TBox $\mathcal{T}$ to the BCALC query

$$\Phi_q(\mathbf{x}) \;=\; \bigvee_{\mathcal{T}\text{-realisable } \mathbf{z}\subseteq\mathbf{y}} \Phi_{\mathbf{z}}(\mathbf{x}). \tag{8}$$

The intuition behind the construction of $\Phi_{\mathbf{z}}(\mathbf{x})$ from $q_{\mathbf{z}}(\mathbf{x})$ and $\mathcal{T}$ hinges on the observation that, for every bag ABox $\mathcal{A}$, the multiplicities of individuals in the interpretations of concepts and roles in $Can(\langle\mathcal{T},\mathcal{A}\rangle)$ are determined by the multiplicities of the assertions in $\mathcal{A}$ as follows:

- for a concept C, the multiplicity of an individual a in $C^{Can(\langle\mathcal{T},\mathcal{A}\rangle)}$ is the maximum multiplicity of a in the interpretation of the concepts subsumed by C with respect to $\mathcal{T}$ in the bag interpretation corresponding to $\mathcal{A}$;
- for a role P, the multiplicity of a pair of individuals (a, b) in $P^{Can(\langle\mathcal{T},\mathcal{A}\rangle)}$ coincides with the multiplicity of assertion $P(a, b)$ in $\mathcal{A}$ (which is justified by the fact that *DL-Lite*$_{\text{CORE}}$ TBoxes do not allow for role inclusions).

Therefore, for a role R, the number of anonymous R-successors of an individual a in the canonical bag interpretation is precisely the multiplicity of a in the interpretation of the concept $\exists R$ minus the number of individual R-successors of a in the ABox. We can then exploit these observations to construct a BCALC query $\Phi_{\mathbf{z}}(\mathbf{x})$ such that the bag answers to $\Phi_{\mathbf{z}}(\mathbf{x})$ over every ABox $\mathcal{A}$ coincide with the bag $[q_{\mathbf{z}}, \mathbf{z}']^{Can(\langle\mathcal{T},\mathcal{A}\rangle)}$ (where all valuations contributing to the latter bag map $\mathbf{z}'$ to anonymous elements and the rest to individuals). For this, it suffices to apply to $q_{\mathbf{z}}$, which has the form (7), the following replacements:

- each atom over an atomic concept A in the conjunction $\psi_{\mathbf{z}}$ of $q_{\mathbf{z}}$ with a BCALC query retrieving the maximum multiplicity over all concepts subsumed by A in $\mathcal{T}$;
- each atom $\alpha_{\mathbf{v}} = P_{\mathbf{v}}(t, y_{\mathbf{v}})$, for a maximally connected $\mathbf{v} \subseteq \mathbf{z}$, with a BCALC query that subtracts the number of $P_{\mathbf{v}}$-successors in the ABox from the maximum multiplicity over all concepts subsumed by $\exists P_{\mathbf{v}}$ in $\mathcal{T}$; and
- each atom $\alpha_{\mathbf{v}} = P_{\mathbf{v}}(y_{\mathbf{v}}, t)$, for a maximally connected $\mathbf{v} \subseteq \mathbf{z}$, with a BCALC query that subtracts the number of $P_{\mathbf{v}}$-predecessors in the ABox from the maximum multiplicity over all concepts subsumed by $\exists P_{\mathbf{v}}^{-}$ in $\mathcal{T}$.

Note that atoms over atomic roles in $\psi_{\mathbf{z}}$ are left intact because $\mathcal{T}$ does not allow for role inclusions.

Next, we formalise this intuition and define the BCALC query $\Phi_{\mathbf{z}}(\mathbf{x})$ in terms of $q_{\mathbf{z}}(\mathbf{x})$ and $\mathcal{T}$.

Definition 54. Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ, $\mathcal{T}$ be a *DL-Lite*$_{\text{CORE}}$ TBox, and let $\mathbf{z}$ be a $\mathcal{T}$-realisable subset of $\mathbf{y}$. The BCALC query $\Phi_{\mathbf{z}}(\mathbf{x})$ is obtained from $q_{\mathbf{z}}(\mathbf{x})$ by replacing

- each occurrence of an atom $A(t)$ in $\psi_{\mathbf{z}}$ with

$$\bigvee_{\mathcal{T}\models C\sqsubseteq A} \zeta_C(t), \tag{9}$$

- for each maximally connected $\mathbf{v} \subseteq \mathbf{z}$, the atom $P_{\mathbf{v}}(t, y_{\mathbf{v}})$ and the atom $P_{\mathbf{v}}(y_{\mathbf{v}}, t)$ with the following BCALC query, where $R_{\mathbf{v}}$ is $P_{\mathbf{v}}$ and $P_{\mathbf{v}}^{-}$, respectively:

$$\left(\bigvee_{\mathcal{T}\models C\sqsubseteq\exists R_{\mathbf{v}}} \zeta_C(t)\right) \setminus \zeta_{\exists R_{\mathbf{v}}}(t), \tag{10}$$

where, for every concept C and term t, and for a fresh variable y,

$$\zeta_C(t) = \begin{cases} C(t), & \text{if } C \text{ is an atomic concept,} \\ \exists y.\, \xi_R(t, y), & \text{if } C \text{ is of the form } \exists R, \end{cases}$$

and, for every atomic role P and terms t_1 and t_2, $\xi_P(t_1, t_2) = P(t_1, t_2)$ and $\xi_{P^-}(t_1, t_2) = P(t_2, t_1)$.

Before proving correctness of the last step of the rewriting, we illustrate the definitions on our running example.

Example 55. Consider the *DL-Lite*$_{\text{CORE}}$ TBox $\mathcal{T} = \{\mathsf{Record} \sqsubseteq \exists \mathsf{hasMusician}, \exists \mathsf{hasMusician}^{-} \sqsubseteq \mathsf{Musician}\}$ and the rooted CQ

$$q(x) = \exists y.\, \mathsf{hasMusician}(x, y) \wedge \mathsf{Musician}(y).$$

There are two subsets of y, namely $\emptyset$ and y, and it is immediate to check that both of them are $\mathcal{T}$-realisable. Moreover,

$$q_\emptyset(x) = q(x) \quad \text{and} \quad q_y(x) = \exists y'.\, \mathsf{hasMusician}(x, y').$$

For the first of these CQs, all valuations contributing to the bag $[q_\emptyset, \emptyset]^{Can(\mathcal{K})}$ map all variables of $q_\emptyset$ to individuals, for every $\mathcal{K}$ with TBox $\mathcal{T}$. Therefore, the atom $\mathsf{hasMusician}(x, y)$ remains intact in $\Phi_\emptyset$ whereas the atom $\mathsf{Musician}(y)$ is rewritten to a BCALC query retrieving the multiplicity of an individual a in $\mathsf{Musician}^{Can(\mathcal{K})}$, which is equal to the maximum multiplicity of a amongst the concepts Musician and $\exists \mathsf{hasMusician}^-$ over the ABox of $\mathcal{K}$. As a result,

$$\Phi_\emptyset(x) = \exists y.\, \mathsf{hasMusician}(x, y) \wedge \big(\mathsf{Musician}(y) \vee \exists z.\, \mathsf{hasMusician}(z, y)\big).$$

For the second CQ, q_y, all valuations contributing to the bag $[q_y, y']^{Can(\mathcal{K})}$ map y' to an anonymous element and x to an individual, for every $\mathcal{K}$ with $\mathcal{T}$. Hence, each such valuation contributes to $[q_y, y']^{Can(\mathcal{K})}$ multiplicity 1, while all valuations λ agreeing on x contribute to $[q_y, y']^{Can(\mathcal{K})}$ an overall multiplicity equal to the number of anonymous $\mathsf{hasMusician}$-successors of $\lambda(x)$. This number is the multiplicity of $\lambda(x)$ in the interpretation of $\exists \mathsf{hasMusician}$ under $Can(\mathcal{K})$ minus the number of individual $\mathsf{hasMusician}$-successors of $\lambda(x)$. Inspecting $\mathcal{T}$, we finally derive

$$\Phi_y(x) = \big(\mathsf{Record}(x) \vee \exists y'.\, \mathsf{hasMusician}(x, y')\big) \setminus \exists y''.\, \mathsf{hasMusician}(x, y'').$$

Consider now the ABox

$$\mathcal{A} = \{\!|\, \mathsf{Record}(\mathit{Expectations}) : 2,\ \mathsf{hasMusician}(\mathit{Expectations}, \mathit{K.\,Jarrett}) : 1 \,|\!\}.$$

Evaluating the two rewritings on this ABox, we get $\Phi_\emptyset^\mathcal{A} = \Phi_y^\mathcal{A} = \{\!|\, \mathit{Expectations} : 1 \,|\!\}$. Therefore, we have $\Phi_q^\mathcal{A} = (\Phi_\emptyset \uplus \Phi_y)^\mathcal{A} = \{\!|\, \mathit{Expectations} : 2 \,|\!\}$, which is equal to $q^{Can(\langle\mathcal{T},\mathcal{A}\rangle)} = q^{\langle\mathcal{T},\mathcal{A}\rangle}$ as expected. ◁

The following lemma establishes correctness of Step 3 of our rewriting approach as formalised in Definition 54.

Lemma 56. *Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and $\mathcal{T}$ a DL-Lite$_{\text{CORE}}$ TBox. Then $[q_\mathbf{z}, \mathbf{z}']^{Can(\langle\mathcal{T},\mathcal{A}\rangle)} = \Phi_\mathbf{z}^\mathcal{A}$ for every bag ABox $\mathcal{A}$ and every $\mathcal{T}$-realisable subset $\mathbf{z}$ of $\mathbf{y}$ with $\mathbf{z}'$ as in Definition 51.*

Proof. We first claim that it is enough to show that, for every $\mathcal{T}$-realisable subset $\mathbf{z}$ of $\mathbf{y}$ and every maximally consistent $\mathbf{v} \subseteq \mathbf{z}$, the bag answers to the rewritings (9) and (10) of atoms $A(t)$ and $\xi_{R_\mathbf{v}}(t, y_\mathbf{v})$, respectively, in $q_\mathbf{z}$ over every bag ABox $\mathcal{A}$ are equal to the bag answers $[A(t), \emptyset]^{Can(\mathcal{K})}$ and $[\exists y_\mathbf{v}.\, \xi_{R_\mathbf{v}}(t, y_\mathbf{v}), y_\mathbf{v}]^{Can(\mathcal{K})}$, respectively, for $\mathcal{K} = \langle\mathcal{T}, \mathcal{A}\rangle$. Indeed, on the one hand, $\Phi_\mathbf{z}$ is obtained from $q_\mathbf{z}$ by applying only these replacements; on the other hand, the atoms that are not rewritten in $\Phi_\mathbf{z}$ are of the form $P(t_1, t_2)$ or $(t_1 = t_2)$, for terms t_1 and t_2 that are mapped to individuals by each valuation $\lambda \in \Lambda_\mathbf{z}$, so such atoms do not need to be rewritten by the fact that atoms $P(t_1, t_2)$ satisfy $P^{Can(\mathcal{K})}(\lambda(t_1), \lambda(t_2)) = \mathcal{A}(P(\lambda(t_1), \lambda(t_2)))$ for every $\lambda \in \Lambda_\mathbf{z}$, whereas equalities do not contribute to multiplicities.

We now argue the correctness of replacements (9) and (10). By the definitions of canonical bag interpretation and concept closure, for all individuals a and concepts C,

$$\begin{aligned} C^{Can(\mathcal{K})}(a) = C^{Can_1(\mathcal{K})}(a) = \mathsf{ccl}_\mathcal{T}[a, Can_0(\mathcal{K})](C) = \\ \max(\, \max\{\mathcal{A}(A_0(a)) \mid A_0 \in \mathbf{C},\ \mathcal{T} \models A_0 \sqsubseteq C\}, \\ \max\{\textstyle\sum_{b \in \mathbf{I}} \mathcal{A}(P_0(a, b)) \mid P_0 \in \mathbf{R},\ \mathcal{T} \models \exists P_0 \sqsubseteq C\},\ \max\{\textstyle\sum_{b \in \mathbf{I}} \mathcal{A}(P_0(b, a)) \mid P_0 \in \mathbf{R},\ \mathcal{T} \models \exists P_0^- \sqsubseteq C\}\,). \end{aligned} \tag{11}$$

First, by substituting A for C in (11) and by the semantics of BCALC queries, we immediately derive the following for every $a \in \mathbf{I}$ and $A \in \mathbf{C}$:

$$[A(a), \emptyset]^{Can(\mathcal{K})} = \left(\bigvee_{\mathcal{T} \models C \sqsubseteq A} \zeta_C(a) \right)^\mathcal{A},$$

which proves the claim for concept atoms. For $R_\mathbf{v}$ atoms, note that, for each individual a, $[\exists y_\mathbf{v}.\, \xi_{R_\mathbf{v}}(a, y_\mathbf{v}), y_\mathbf{v}]^{Can(\mathcal{K})}$ is the number of anonymous $R_\mathbf{v}$-successors of a in the bag canonical interpretation, which is $(\exists R_\mathbf{v})^{Can(\mathcal{K})}(a) - (\exists R_\mathbf{v})^\mathcal{A}(a)$. Therefore, by substituting $\exists R_\mathbf{v}$ for C in (11) and by the semantics of BCALC queries, we get the following for every $a \in \mathbf{I}$ and $A \in \mathbf{C}$:

$$[\exists y_\mathbf{v}.\, \xi_{R_\mathbf{v}}(a, y_\mathbf{v}), y_\mathbf{v}]^{Can(\mathcal{K})} = \left(\left(\bigvee_{\mathcal{T} \models C \sqsubseteq \exists R_\mathbf{v}} \zeta_C(a) \right) \setminus \xi_{\exists R_\mathbf{v}}(a) \right)^\mathcal{A},$$

which proves the claim for $R_\mathbf{v}$ atoms. □

8.6. Rewriting and complexity

Putting the results of the previous three sections together, we obtain the following theorem, which establishes the correctness of our rewriting approach.

Theorem 57. *For every rooted CQ q and every DL-Lite$^{\flat}_{\text{CORE}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ we have that $q^{Can(\mathcal{K})} = \Phi_q^{\mathcal{A}}$.*

Proof. Consider a rooted CQ $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ and a DL-Lite$^{\flat}_{\text{CORE}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. By (3), by Lemmas 50, 53, and 56, and by (8) we have the sequence of equalities

$$q^{Can(\mathcal{K})} = \biguplus_{\mathbf{z} \subseteq \mathbf{y}} [q, \mathbf{z}]^{Can(\mathcal{K})} = \biguplus_{\mathcal{T}\text{-realisable } \mathbf{z} \subseteq \mathbf{y}} [q_{\mathbf{z}}, \mathbf{z}']^{Can(\mathcal{K})} = \biguplus_{\mathcal{T}\text{-realisable } \mathbf{z} \subseteq \mathbf{y}} \Phi_{\mathbf{z}}^{\mathcal{A}} = \left(\bigvee_{\mathcal{T}\text{-realisable } \mathbf{z} \subseteq \mathbf{y}} \Phi_{\mathbf{z}} \right)^{\mathcal{A}} = \Phi_q^{\mathcal{A}},$$

which proves the statement of the theorem. □

Theorems 42 and 57, together with the fact that Φ_q does not depend on $\mathcal{A}$, imply our main rewritability result.

Corollary 58. *The class of rooted CQs is rewritable to* BCALC *over DL-Lite$^{\flat}_{\text{CORE}}$.*

Recall that by Corollary 43 the rewritability result applies regardless of the adoption of the UNA. Note also that we need to manipulate only finite bags when evaluating the rewriting Φ_q. Since BCALC maximal union is expressible via BCALC arithmetic union and difference for such bags according to equation (1), we can strengthen Corollary 58 and claim that there always exists a rewriting that uses only $\wedge$, $\vee$, $\setminus$, equalities, and existential quantification.

We conclude with the LogSpace upper bound on the data complexity of the query answering for rooted CQs over DL-Lite$^{\flat}_{\text{CORE}}$. We can decide this problem by checking the ontology for non-satisfiability as in the usual set setting and, if the check fails, evaluating the BCALC rewriting of the input query on the ABox. The algorithm is correct by Statement 1 of Theorem 23 on the equivalence of satisfiability under bag and set semantics, and by Corollaries 43 and 58. Both steps can be done in LogSpace by Proposition 5 on the LogSpace membership of the query answering problem for BCALC and the results obtained by Calvanese et al. [9].

Corollary 59. BagCert$\big[$rooted CQs, DL-Lite$^{\flat}_{\text{CORE}}\big]$ *and* BagCert$^{\text{UNA}}\big[$rooted CQs, DL-Lite$^{\flat}_{\text{CORE}}\big]$ *are in* LogSpace *in data complexity.*

9. Rewritability of conjunctive queries over DL-Lite$^{\flat}_{\text{RDFS}}$ under UNA

In this section we show BCALC rewritability of CQs over DL-Lite$^{\flat}_{\text{RDFS}}$ under the UNA as well as tractability of the corresponding query answering problem in data complexity. This result holds only under the UNA since, as shown in Theorem 29, even rooted CQ answering over DL-Lite$^{\flat}_{\text{RDFS}}$ is coNP-hard in data complexity if the UNA is dropped. Note, however, that the results of this section hold for arbitrary CQs and not just rooted ones.

We proceed analogously to the case of DL-Lite$^{\flat}_{\text{CORE}}$ described in the previous two sections; however, the absence of existential quantification on the right-hand side of concept inclusions considerably simplifies the exposition.

First, to formalise canonical bag interpretations for DL-Lite$^{\flat}_{\text{RDFS}}$, we introduce the notion of *role closure*, which is analogous to that of concept closure in Section 7. The role closure $\mathsf{rcl}_{\mathcal{T}}[(u, v), I]$ of a pair (u, v) of elements $u, v \in \Delta^I$ in a bag interpretation $I = \langle \Delta^I, \cdot^I \rangle$ over a TBox $\mathcal{T}$ is the bag of roles such that, for every role R,

$$\mathsf{rcl}_{\mathcal{T}}[(u, v), I](R) = \max\{R_0^I(u, v) \mid \mathcal{T} \models R_0 \sqsubseteq R\}.$$

In other words, $\mathsf{rcl}_{\mathcal{T}}[(u, v), I](R)$ is the maximum value of $R_0^I(u, v)$ amongst all roles R_0 satisfying $\mathcal{T} \models R_0 \sqsubseteq R$.

We are now ready to define canonical bag interpretations for DL-Lite$^{\flat}_{\text{RDFS}}$ ontologies.

Definition 60. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$^{\flat}_{\text{RDFS}}$ ontology and $Can_0(\mathcal{K})$ be the bag interpretation corresponding to $\mathcal{A}$—that is, such that $\Delta^{Can_0(\mathcal{K})} = \mathbf{I}$, $a^{Can_0(\mathcal{K})} = a$ for each $a \in \mathbf{I}$, and $S^{Can_0(\mathcal{K})}(\mathbf{a}) = \mathcal{A}(S(\mathbf{a}))$ for each $S \in \mathbf{C} \cup \mathbf{R}$ and tuple of individuals $\mathbf{a}$. Let also $Can_{\mathbf{R}}(\mathcal{K})$ be the bag interpretation with domain $\mathbf{I}$ that interprets individuals and atomic concepts as $Can_0(\mathcal{K})$ and, for each atomic role P and individuals a, b,

$$P^{Can_{\mathbf{R}}(\mathcal{K})}(a, b) = \mathsf{rcl}_{\mathcal{T}}[(a, b), Can_0(\mathcal{K})](P).$$

The *canonical bag interpretation* $Can(\mathcal{K})$ of $\mathcal{K}$ is the bag interpretation with domain $\mathbf{I}$ that interprets individuals and atomic roles as $Can_{\mathbf{R}}(\mathcal{K})$, and, for each atomic concept A and individual a, satisfies

$$A^{Can(\mathcal{K})}(a) = \mathsf{ccl}_{\mathcal{T}}[a, Can_{\mathbf{R}}(\mathcal{K})](A).$$

There are two main differences between the canonical bag interpretations for *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ and *DL-Lite*$^{\mathsf{b}}_{\text{CORE}}$ ontologies. On the one hand, canonical bag interpretations for *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ do not involve anonymous domain elements; this is because the logic does not support existentially quantified concepts on the right-hand side of inclusions. On the other hand, canonical bag interpretations for *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ need to satisfy role inclusions, which is ensured using the notion of role closure. As expected, the aforementioned definitions of a canonical bag interpretation coincide for ontologies that are both in *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ and *DL-Lite*$^{\mathsf{b}}_{\text{CORE}}$.

We next argue that the canonical bag interpretation in Definition 60 is a universal model for CQs under the assumptions in this section.

Proposition 61. *The canonical bag interpretation of a DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ *ontology is a universal model for the class of all CQs under the UNA.*

Proof. First, note that every *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ ontology is satisfiable under the UNA, because it does not have any disjointness axioms (or any other axioms that may cause an inconsistency). Moreover, the canonical bag interpretation $\mathit{Can}(\mathcal{K})$ is a model of a *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ ontology $\mathcal{K}$ by construction. Second, every bag interpretation I satisfying the UNA has the corresponding bag interpretation I_s satisfying the *standard name assumption*—that is, I_s is the same as I except that, for every $a \in \mathbf{I}$, the individual a itself is used in I_s as an element instead of a^I; moreover, $q^I = q^{I_s}$ for every query q. Therefore, it is enough to show that, for every model I of $\mathcal{K}$, I_s contains $\mathit{Can}(\mathcal{K})$, in the sense that $\Delta^{\mathit{Can}(\mathcal{K})} \subseteq \Delta^{I_s}$ and $S^{\mathit{Can}(\mathcal{K})} \subseteq S^{I_s}$ for every $S \in \mathbf{C} \cup \mathbf{R}$. However, this follows from the definitions of a bag interpretation and the canonical bag interpretation, because concept and role closures essentially encode satisfaction of the inclusions. □

Example 62. Consider the *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is the TBox defined as

$$\mathcal{T} = \{\exists \mathsf{hasMusician} \sqsubseteq \mathsf{Record},\ \mathsf{playsOn}^- \sqsubseteq \mathsf{hasMusician}\}$$

and $\mathcal{A}$ is the bag ABox defined as

$$\mathcal{A} = \{\!\{\, \mathsf{Record}(\mathit{Expectations}) : 1,\ \mathsf{hasMusician}(\mathit{Expectations}, \mathit{K.\ Jarrett}) : 1,\ \mathsf{playsOn}(\mathit{P.\ Motian}, \mathit{Expectations}) : 1 \,\}\!\}.$$

The canonical bag interpretation $\mathit{Can}(\mathcal{K})$ is the bag interpretation having domain $\Delta^{\mathit{Can}(\mathcal{K})} = \mathbf{I}$, interpreting all individuals by themselves, and assigning bags to atomic concepts and atomic roles as follows:

$$\begin{aligned}
\mathsf{Record}^{\mathit{Can}(\mathcal{K})} &= \{\!\{\, \mathit{Expectations} : 2 \,\}\!\},\\
\mathsf{playsOn}^{\mathit{Can}(\mathcal{K})} &= \{\!\{\, (\mathit{P.\ Motian}, \mathit{Expectations}) : 1 \,\}\!\},\\
\mathsf{hasMusician}^{\mathit{Can}(\mathcal{K})} &= \{\!\{\, (\mathit{Expectations}, \mathit{K.\ Jarrett}) : 1,\ (\mathit{Expectations}, \mathit{P.\ Motian}) : 1 \,\}\!\}.
\end{aligned}$$

By Definition 60, the bag interpretations $\mathit{Can}_0(\mathcal{K})$ and $\mathit{Can}_{\mathbf{R}}(\mathcal{K})$ are involved in the construction of $\mathit{Can}(\mathcal{K})$. These are specified as follows. Bag interpretation $\mathit{Can}_0(\mathcal{K})$ reflects the bag ABox $\mathcal{A}$ and maps Record to $\{\!\{\, \mathit{Expectations} : 1 \,\}\!\}$, hasMusician to $\{\!\{\, (\mathit{Expectations}, \mathit{K.\ Jarrett}) : 1 \,\}\!\}$, and playsOn to $\{\!\{\, (\mathit{P.\ Motian}, \mathit{Expectations}) : 1 \,\}\!\}$. Bag interpretation $\mathit{Can}_{\mathbf{R}}(\mathcal{K})$ extends $\mathit{Can}_0(\mathcal{K})$ by setting the multiplicity of tuple $(\mathit{Expectations}, \mathit{P.\ Motian})$ in role hasMusician to 1, thus, satisfying the role inclusion axiom in $\mathcal{T}$. Then, interpretation $\mathit{Can}(\mathcal{K})$ is defined on the basis of $\mathit{Can}_{\mathbf{R}}(\mathcal{K})$ by setting the multiplicity of element *Expectations* in Record to 2, thus, satisfying the concept inclusion in $\mathcal{T}$.

Observe that satisfying concept inclusions only after role inclusions have been satisfied is crucial for deriving a model of $\mathcal{K}$. This is because satisfaction of role inclusions may result in the increase of an individual's multiplicity in the interpretation of a concept, which clearly affects satisfaction of concept inclusions. Indeed, it is easy to verify that the interpretation resulting from the parallel satisfaction of the inclusions in $\mathcal{T}$ is not a model of $\mathcal{K}$, because it assigns to role Record the bag $\{\!\{\, \mathit{Expectations} : 1 \,\}\!\}$, hence, violating the concept inclusion in $\mathcal{T}$. ◁

Next, we investigate BCALC rewritability of CQs over *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ under the UNA. We start by arguing that rooted CQs can be rewritten to neither BCALC maximal nor BCALC arithmetic unions of CQs.

Proposition 63. *The class of rooted CQs is rewritable neither to* BCALC *maximal nor to* BCALC *arithmetic unions of CQs over DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ *under the UNA.*

Proof. For BCALC maximal unions of CQs, consider the Boolean rooted CQ $q = A(a)$ for an atomic concept A and individual a, and the *DL-Lite*$^{\mathsf{b}}_{\text{RDFS}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ with $\mathcal{T} = \{P \sqsubseteq R,\ \exists R \sqsubseteq A\}$ and $\mathcal{A} = \{\!\{\, P(a,b) : 1,\ R(a,c) : 1 \,\}\!\}$, for atomic roles P and R as well as individuals b and c. By Proposition 61, we have $q^{\mathcal{K}}(\langle\rangle) = q^{\mathit{Can}(\mathcal{K})}(\langle\rangle)$; moreover, $\Delta^{\mathit{Can}(\mathcal{K})} = \mathbf{I}$, while

$$A^{\mathit{Can}(\mathcal{K})} = \{\!\{\, a : 2 \,\}\!\}, \qquad P^{\mathit{Can}(\mathcal{K})} = \{\!\{\, (a,b) : 1 \,\}\!\}, \qquad R^{\mathit{Can}(\mathcal{K})} = \{\!\{\, (a,b) : 1,\ (a,c) : 1 \,\}\!\}.$$

As a result, we have $q^{\mathcal{K}}(\langle\rangle) = 2$. Assume now for the sake of contradiction that there exists a BCALC maximal union of CQs Φ such that $\Phi^{\mathcal{A}}(\langle\rangle) = 2$. By the semantics of maximal union, this means that Φ contains a CQ $q_0 = \exists \mathbf{y}.\, \phi(\mathbf{y})$ satisfying $q_0^{\mathcal{A}}(\langle\rangle) = 2$. There are only two ways in which this is possible:

- if there is only one valuation $\lambda : \mathbf{y} \cup \mathbf{I} \to \mathbf{I}$ for q_0 contributing multiplicity 2 to $q_0^{\mathcal{A}}(\langle\rangle)$, then $\phi(\mathbf{y})$ must contain an atom $S(\mathbf{t})$ such that $S^{\mathcal{A}}(\lambda(\mathbf{t})) = 2$; however, this is not possible because $\mathcal{A}$ does not have any assertions with multiplicity 2;
- there exist two distinct valuations for q_0, each contributing multiplicity 1; however, this is not possible because $\mathcal{A}$ does not have two assertions with multiplicity 1 over the same atomic concept or role.

As a result, we conclude that Φ cannot contain any such CQ q_0, and hence Φ cannot be a rewriting of q with respect to $\mathcal{T}$.

Finally, non-rewritability to BCALC arithmetic unions of CQs follows already from the proof of the second part of Proposition 45, where the relevant TBox consists only of an inclusion between two atomic concepts. □

In what follows, we show how to construct a BCALC rewriting of an arbitrary CQ with respect to a *DL-Lite*$_{\text{RDFS}}$ TBox. The construction is much simpler than that for *DL-Lite*$^{\flat}_{\text{CORE}}$ in Section 8 since the canonical bag interpretation of a *DL-Lite*$^{\flat}_{\text{RDFS}}$ ontology does not contain anonymous elements. In particular, there is no need to identify $\mathcal{T}$-realisable subsets of existentially quantified variables and to transform the input CQ accordingly, which implies that Steps 1 and 2 in Section 8 are no longer required. As a result, a BCALC rewriting of the CQ can be directly constructed following an analogous approach to that of Step 3 in Section 8, and yet the operations of arithmetic union and difference are no longer required. Note, however, that the BCALC query resulting from the rewriting is not a BCALC maximal union of CQs since it interleaves maximal unions with existential quantification.

Definition 64. Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a CQ and $\mathcal{T}$ be a *DL-Lite*$_{\text{RDFS}}$ TBox. The BCALC query $\Phi_q(\mathbf{x})$ is obtained from $q(\mathbf{x})$ by replacing each occurrence of an atom $A(t)$ and an atom $P(t_1, t_2)$ with

$$\bigvee_{\mathcal{T} \models C \sqsubseteq A} \zeta_C(t) \qquad \text{and} \qquad \bigvee_{\mathcal{T} \models R \sqsubseteq P} \xi_R(t_1, t_2), \qquad \text{respectively,} \tag{12}$$

where, for every concept C and term t, and for a fresh variable y,

$$\zeta_C(t) = \begin{cases} C(t), & \text{if } C \text{ is an atomic concept,} \\ \exists y.\, \Big(\bigvee_{\mathcal{T} \models R_0 \sqsubseteq R} \xi_{R_0}(t, y) \Big), & \text{if } C \text{ is of the form } \exists R, \end{cases}$$

and, for every atomic role P and terms t_1 and t_2, $\xi_P(t_1, t_2) = P(t_1, t_2)$ and $\xi_{P^-}(t_1, t_2) = P(t_2, t_1)$.

So, analogously to Definition 54 in the *DL-Lite*$^{\flat}_{\text{CORE}}$ case, we replace each concept atom $A(t)$ with a BCALC maximal union of all atoms over t corresponding to concepts subsumed by A in $\mathcal{T}$; in contrast to Definition 54, however, we also need to take into account role inclusions, which results in the inclusion of further disjuncts in the second part of the definition of $\zeta_C(t)$. Then, each role atom $P(t_1, t_2)$ is replaced by the BCALC maximal union of all binary atoms over t_1 and t_2 corresponding to roles subsumed by P in $\mathcal{T}$.

We now provide an example illustrating the construction of a rewriting.

Example 65. Consider TBox $\mathcal{T}$ from Example 62 and the Boolean CQ $q = \exists y_1, y_2.\, \mathsf{Record}(y_1) \wedge \mathsf{hasMusician}(y_1, y_2)$. Following Definition 64, the relevant concepts for rewriting the atom $\mathsf{Record}(y_1)$ are Record itself, $\exists \mathsf{hasMusician}$, and $\exists \mathsf{playsOn}^-$ as these are all the concepts subsumed by Record in $\mathcal{T}$. Similarly, the relevant roles for rewriting the atom $\mathsf{hasMusician}(y_1, y_2)$ are hasMusician itself and $\mathsf{playsOn}^-$ as these are all the roles subsumed by hasMusician in $\mathcal{T}$. Thus atoms $\mathsf{Record}(y_1)$ and $\mathsf{hasMusician}(y_1, y_2)$ are replaced in q with expressions

$$\zeta_{\mathsf{Record}}(y_1) \vee \zeta_{\exists \mathsf{hasMusician}}(y_1) \vee \zeta_{\exists \mathsf{playsOn}^-}(y_1) = \mathsf{Record}(y_1) \vee \exists y.\, (\mathsf{hasMusician}(y_1, y) \vee \mathsf{playsOn}(y, y_1)) \vee \exists y.\, \mathsf{playsOn}(y, y_1)$$

and $\quad \xi_{\mathsf{hasMusician}}(y_1, y_2) \vee \xi_{\mathsf{playsOn}^-}(y_1, y_2) = \mathsf{hasMusician}(y_1, y_2) \vee \mathsf{playsOn}(y_2, y_1),$

respectively, resulting in the rewriting

$$\Phi_q = \exists y_1, y_2.\, \big(\mathsf{Record}(y_1) \vee \exists y.\, (\mathsf{hasMusician}(y_1, y) \vee \mathsf{playsOn}(y, y_1)) \vee \exists y.\, \mathsf{playsOn}(y, y_1)\big) \wedge \big(\mathsf{hasMusician}(y_1, y_2) \vee \mathsf{playsOn}(y_2, y_1)\big).$$

Consider now ABox

$$\mathcal{A} = \{\!|\, \mathsf{Record}(\mathit{Expectations}) : 1,\ \mathsf{hasMusician}(\mathit{Expectations}, \mathit{K.\,Jarrett}) : 1,\ \mathsf{playsOn}(\mathit{P.\,Motian}, \mathit{Expectations}) : 1 \,|\!\},$$

from Example 62 and the canonical bag interpretation $\mathit{Can}(\mathcal{K})$ of the *DL-Lite*$^{\flat}_{\text{RDFS}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, specified there. On the one hand, evaluating q over $\mathit{Can}(\mathcal{K})$ results in the bag $q^{\mathit{Can}(\mathcal{K})} = \{\!|\, \langle\rangle : 4 \,|\!\}$. On the other hand, it is immediate to check that Φ_q also evaluates to $\{\!|\, \langle\rangle : 4 \,|\!\}$ over $\mathcal{A}$. ◁

We next prove correctness of the rewriting formalised in Definition 64.

Theorem 66. *For every CQ q and every DL-Lite$^{\flat}_{\text{RDFS}}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ we have that $q^{Can(\mathcal{K})} = \Phi_q^{\mathcal{A}}$.*

Proof. Let $q(\mathbf{x}) = \exists \mathbf{y}.\ \phi(\mathbf{x}, \mathbf{y})$ be a CQ and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be *DL-Lite*$^{\flat}_{\text{RDFS}}$ ontology. To prove the claim, it suffices to show that for each atom $A(t)$ and $P(t_1, t_2)$ in $\phi(\mathbf{x}, \mathbf{y})$, and for each valuation $\lambda : \mathbf{x} \cup \mathbf{y} \cup \mathbf{I} \to \Delta^{Can(\mathcal{K})}$, the multiplicities $A^{Can(\mathcal{K})}(\lambda(t))$ and $P^{Can(\mathcal{K})}(\lambda(t_1), \lambda(t_2))$ are equal to the multiplicities of $\lambda(t)$ and $(\lambda(t_1), \lambda(t_2))$ in the bag answers to the BCALC queries of (12) over ABox $\mathcal{A}$, respectively.

Canonical interpretation $Can(\mathcal{K})$ assigns to every atomic concept A and every atomic role P the bag defined as $A^{Can(\mathcal{K})}(u) = \mathsf{ccl}_{\mathcal{T}}[u, Can_{\mathbf{R}}(\mathcal{K})](A)$ and $P^{Can(\mathcal{K})}(u, v) = \mathsf{rcl}_{\mathcal{T}}[(u, v), Can_0(\mathcal{K})](P)$ for all elements $u, v \in \Delta^{Can(\mathcal{K})}$, respectively. By unfolding the definition of closures, we get the following, for every atom $A(t)$ and $P(t_1, t_2)$ in $\phi(\mathbf{x}, \mathbf{y})$, and every valuation λ:

$$\begin{aligned} A^{Can(\mathcal{K})}(\lambda(t)) = \max(\ & \max\{\mathcal{A}(A_0(\lambda(t))) \mid A_0 \in \mathbf{C},\ \mathcal{T} \models A_0 \sqsubseteq A\}, \\ & \max\{(\exists R)^{Can_{\mathbf{R}}(\mathcal{K})}(\lambda(t)) \mid R \text{ a role},\ \mathcal{T} \models \exists R \sqsubseteq A\}), \end{aligned} \tag{13}$$

$$\begin{aligned} P^{Can(\mathcal{K})}(\lambda(t_1), \lambda(t_2)) = \max(\ & \max\{\mathcal{A}(P_0(\lambda(t_1), \lambda(t_2))) \mid P_0 \in \mathbf{R},\ \mathcal{T} \models P_0 \sqsubseteq P\}, \\ & \max\{\mathcal{A}(P_0(\lambda(t_2), \lambda(t_1))) \mid P_0 \in \mathbf{R},\ \mathcal{T} \models P_0^- \sqsubseteq P\}). \end{aligned} \tag{14}$$

Consider first the right-hand side of (13). Observe that the first subexpression involving the max function can be equivalently written as $\big(\bigvee_{A_0 \in \mathbf{C},\ \mathcal{T} \models A_0 \sqsubseteq A} A_0(t)\big)^{\mathcal{A}}(\lambda(t))$. As for the second subexpression, it can be equivalently written as $\big(\bigcup_{\mathcal{T} \models \exists R \sqsubseteq A} (\exists R)^{Can_{\mathbf{R}}(\mathcal{K})}\big)(\lambda(t))$ where bag $(\exists R)^{Can_{\mathbf{R}}(\mathcal{K})}$ can be further written as $\big(\exists y'.\ \big(\bigvee_{\mathcal{T} \models R_0 \sqsubseteq R} \xi_{R_0}(t, y')\big)\big)^{\mathcal{A}}$. Thus, by substituting $\bigvee$ for $\cup$ in the aforementioned expression and for the outer max function in (13), and by the definition of $\zeta_C(t)$ in Definition 64, we derive

$$A^{Can(\mathcal{K})}(\lambda(t)) = \Big(\bigvee\nolimits_{\mathcal{T} \models C \sqsubseteq A} \zeta_C(t)\Big)^{\mathcal{A}}(\lambda(t)),$$

as required. To complete the proof of the theorem, observe that from (14) and the semantics of BCALC queries we immediately obtain $P^{Can(\mathcal{K})}(\lambda(t_1), \lambda(t_2)) = \big(\bigvee_{\mathcal{T} \models R \sqsubseteq P} \xi_R(t_1, t_2)\big)^{\mathcal{A}}(\lambda(t_1), \lambda(t_2))$, as required. □

From Proposition 61 and Theorem 66 we obtain the rewritability result of this section.

Corollary 67. *The class of all CQs is rewritable to positive* BCALC *over DL-Lite*$^{\flat}_{\text{RDFS}}$ *under the UNA.*

All *DL-Lite*$^{\flat}_{\text{RDFS}}$ ontologies are satisfiable, so the data complexity upper bound for the corresponding query answering problem can be established as a straightforward consequence of Proposition 5 and Corollary 67.

Corollary 68. BAGCERT$^{\text{UNA}}$[CQs, *DL-Lite*$^{\flat}_{\text{RDFS}}$] *is in* LOGSPACE *in data complexity.*

10. Rewritability of rooted conjunctive queries over *DL-Lite*$^{\flat}_{\mathcal{R}^-}$ under UNA

In this section we consider BCALC rewritings of rooted CQs over the ontology language *DL-Lite*$^{\flat}_{\mathcal{R}^-}$, which extends both *DL-Lite*$^{\flat}_{\text{CORE}}$ and *DL-Lite*$^{\flat}_{\text{RDFS}}$. As defined in Section 2.1, this language provides all the constructs available in *DL-Lite*$_{\mathcal{R}}$, but allows concept inclusions of the form $C \sqsubseteq \exists R$ only for roles R that do not have more general roles in the ontology.

Similarly to the case of *DL-Lite*$^{\flat}_{\text{RDFS}}$ considered in the previous section, we focus on the semantics that adopts the UNA; this is justified by Theorem 29, where we showed that (rooted) CQ answering over ontology languages allowing for role inclusions is coNP-hard in data complexity if the UNA is not adopted. However, in contrast to the previous section, we focus on rooted CQs rather than general CQs; this choice is justified by Theorem 28, where we established coNP-hardness of CQ answering over *DL-Lite*$^{\flat}_{\text{CORE}}$ under the UNA.

The strategy behind our proof of BCALC rewritability of rooted CQs over *DL-Lite*$^{\flat}_{\mathcal{R}^-}$ is to seamlessly combine the techniques developed in the previous three sections for *DL-Lite*$^{\flat}_{\text{CORE}}$ and *DL-Lite*$^{\flat}_{\text{RDFS}}$. We start with the definition of the canonical bag interpretation. The construction is divided into two steps: first, we build the canonical bag interpretation of the *DL-Lite*$^{\flat}_{\text{RDFS}}$ sub-ontology consisting only of role inclusions, and then use the resulting interpretation as the starting point for the recursive model construction done for *DL-Lite*$^{\flat}_{\text{CORE}}$, where we now ignore the role inclusions.

Definition 69. The *canonical bag interpretation* $Can(\mathcal{K})$ of a *DL-Lite*$^{\flat}_{\mathcal{R}^-}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is the bag interpretation $\bigcup_{i \geq 0} Can_i(\mathcal{K})$, where $Can_0(\mathcal{K})$ is the canonical bag interpretation of the *DL-Lite*$^{\flat}_{\text{RDFS}}$ ontology obtained from $\mathcal{K}$ by keeping only the role inclusions in $\mathcal{T}$ (as defined in Definition 60), and, for each $i \geq 1$, interpretation $Can_i(\mathcal{K})$ is defined on the basis of $Can_{i-1}(\mathcal{K})$ as in Definition 32 by considering only the concept inclusions in $\mathcal{T}$.

It is immediate to check that this definition generalises the definitions of canonical bag interpretations for *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ and *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$. We next illustrate the aforementioned definition with an example.

Example 70. Consider the *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where

$$\mathcal{T} = \{\mathsf{Record} \sqsubseteq \exists \mathsf{hasMusician},\ \exists \mathsf{hasMusician}^- \sqsubseteq \mathsf{Musician},\ \mathsf{playsOn}^- \sqsubseteq \mathsf{hasMusician}\},$$

$$\mathcal{A} = \{\!|\ \mathsf{Record}(\mathit{Expectations}) : 2,\ \mathsf{playsOn}(\mathit{K.\,Jarrett},\ \mathit{Expectations}) : 1\ |\!\}.$$

The canonical bag interpretation $\mathit{Can}(\mathcal{K})$ has domain $\Delta^{\mathit{Can}(\mathcal{K})} = \mathbf{I} \cup \{w^1_{\mathit{Expectations},\exists \mathsf{hasMusician}}\}$, interprets all the individuals by themselves, and assigns bags to atomic concepts and atomic roles as follows:

$$\begin{aligned}
\mathsf{Record}^{\mathit{Can}(\mathcal{K})} &= \{\!|\ \mathit{Expectations} : 2\ |\!\},\\
\mathsf{Musician}^{\mathit{Can}(\mathcal{K})} &= \{\!|\ \mathit{K.\,Jarrett} : 1,\ w^1_{\mathit{Expectations},\exists \mathsf{hasMusician}} : 1\ |\!\},\\
\mathsf{playsOn}^{\mathit{Can}(\mathcal{K})} &= \{\!|\ (\mathit{K.\,Jarrett},\ \mathit{Expectations}) : 1\ |\!\},\\
\mathsf{hasMusician}^{\mathit{Can}(\mathcal{K})} &= \{\!|\ (\mathit{Expectations},\ \mathit{K.\,Jarrett}) : 1,\ (\mathit{Expectations},\ w^1_{\mathit{Expectations},\exists \mathsf{hasMusician}}) : 1\ |\!\}. \quad \lhd
\end{aligned}$$

Note that the canonical bag interpretation of a satisfiable *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is always a model of $\mathcal{K}$ by construction and the fact that $\mathcal{T}$ does not have concept inclusions of the form $C \sqsubseteq \exists R$ whenever R has a more general role in $\mathcal{T}$. This fact guarantees that the subinterpretation of $\mathit{Can}(\mathcal{K})$ corresponding to the set of concept inclusions in $\mathcal{T}$, namely $\bigcup_{i \geq 1} \mathit{Can}_i(\mathcal{K})$, does not violate the role inclusions in $\mathcal{T}$ that have been already satisfied in $\mathit{Can}_0(\mathcal{K})$. Given this property, the following is a generalisation of Proposition 34 for the canonical bag interpretations of *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ ontologies, which holds again regardless of whether the UNA is adopted or not (recall that this was automatic for the case of *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$ because this language does not allow for any inconsistencies).

Proposition 71. *If a DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ *ontology is satisfiable then its canonical bag interpretation is its model.*

We next show that the canonical bag interpretation is universal for the class of rooted CQs. For this, we establish the counterpart of Lemma 38 for the case of *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ ontologies.

Lemma 72. *For every DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ *ontology* $\mathcal{K}$ *and every model* I *of* $\mathcal{K}$ *there exists an e-homomorphism from* $\mathit{Can}^e(\mathcal{K})$ *to* I^e *that is multiplicity-preserving on* $\mathbf{I}$.

Proof. The proof goes along the same lines as the proof of Lemma 38. The only difference is the way in which we define a witnessing multiplicity-preserving e-homomorphism $(h, h_S, \ldots)$ for atomic roles and the elements in $\Delta^{\mathit{Can}_0(\mathcal{K})}$—that is, on the (interpretations of the) individuals $\mathbf{I}$. However, the existence of such an e-homomorphism follows from the proof of Proposition 61, which shows universality of the canonical bag interpretation for *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$. Indeed, the fact that for every model I of $\mathcal{K}$ the corresponding model I_s (i.e., the model satisfying the standard name assumption) is such that $\Delta^{\mathit{Can}_0(\mathcal{K})} \subseteq \Delta^{I_s}$ and $S^{\mathit{Can}_0(\mathcal{K})} \subseteq S^{I_s}$ for every $S \in \mathbf{C} \cup \mathbf{R}$ implies the existence of an e-homomorphism. □

Using Lemma 72 and the fact that Lemma 41 transfers trivially to *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ ontologies, we can conclude that Theorem 42 (as well as Proposition 61) generalises to the case of *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$.

Theorem 73. *The canonical bag interpretation* $\mathit{Can}(\mathcal{K})$ *of a satisfiable DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ *ontology* $\mathcal{K}$ *is a universal model for the class of rooted CQs under the UNA.*

We next move to the BCALC rewritability of rooted CQs over *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$. We start by pointing out that this class of queries is rewritable to neither BCALC maximal nor to BCALC arithmetic unions of CQs over *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ under the UNA, which is a direct consequence of Propositions 45 and 63, as well as the fact that *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ extends both *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ and *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$. In fact, the rewriting algorithm can be seen as a combination of the corresponding algorithms for *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ and *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$, described in Sections 8 and 9, respectively. When given as input a *DL-Lite*$_{\mathcal{R}^-}$ TBox $\mathcal{T}$ and a rooted CQ $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$, the algorithm considers each subset $\mathbf{z}$ of $\mathbf{y}$ independently and then takes the BCALC arithmetic union of the resulting rewritings. In particular, for each such $\mathbf{z}$, we proceed according to the following three steps:

1. as specified in Definition 48, $\mathbf{z}$ is checked for $\mathcal{T}_{\mathbf{C}}$-realisability, where $\mathcal{T}_{\mathbf{C}}$ is the set of concept inclusions in $\mathcal{T}$, and disregarded from consideration if the check fails,
2. as specified in Definition 51, each maximally connected component of the subquery corresponding to $\mathbf{z}$ is replaced by a single representative role atom, resulting in a CQ; and

3. all atoms are rewritten to a BCALC query that takes into account the TBox and the fact that $\mathbf{z}$ should be mapped to anonymous elements.

The only essential difference to the algorithm for *DL-Lite*$^{\flat}_{\text{CORE}}$ is in the last step, where we now take into account also the presence of role inclusions, as in the case of *DL-Lite*$^{\flat}_{\text{RDFS}}$.

The construction of the rewriting is formalised by the following definition.

Definition 74. Let $q(\mathbf{x}) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$ be a rooted CQ and $\mathcal{T}$ a *DL-Lite*$_{\mathcal{R}^-}$ TBox. For a $\mathcal{T}_{\mathbf{C}}$-realisable subset $\mathbf{z}$ of $\mathbf{y}$, let $\Phi_{\mathbf{z}}(\mathbf{x})$ be the query obtained from the CQ $q_{\mathbf{z}}(\mathbf{x})$, given in Definition 51, by replacing

– each occurrence of an atom $A(t)$ and an atom $P(t_1, t_2)$ with

$$\bigvee_{\mathcal{T}_{\mathbf{C}} \models C \sqsubseteq A} \zeta_C(t) \qquad \text{and} \qquad \bigvee_{\mathcal{T} \models R \sqsubseteq P} \xi_R(t_1, t_2), \qquad \text{respectively,} \tag{15}$$

– the atom $P_{\mathbf{v}}(t, y_{\mathbf{v}})$ and the atom $P_{\mathbf{v}}(y_{\mathbf{v}}, t)$ for each maximally connected $\mathbf{v} \subseteq \mathbf{z}$ with the following BCALC query, where $R_{\mathbf{v}}$ is $P_{\mathbf{v}}$ and $P_{\mathbf{v}}^-$, respectively:

$$\left(\bigvee_{\mathcal{T}_{\mathbf{C}} \models C \sqsubseteq \exists R_{\mathbf{v}}} \zeta_C(t) \right) \setminus \zeta_{\exists R_{\mathbf{v}}}(t), \tag{16}$$

where, for every concept C and term t, and for a fresh variable y,

$$\zeta_C(t) = \begin{cases} C(t), & \text{if } C \text{ is an atomic concept,} \\ \exists y.\, \left(\bigvee_{\mathcal{T} \models R_0 \sqsubseteq R} \xi_{R_0}(t, y) \right), & \text{if } C \text{ is of the form } \exists R, \end{cases}$$

and, for every atomic role P and terms t_1, t_2, $\xi_P(t_1, t_2) = P(t_1, t_2)$ and $\xi_{P^-}(t_1, t_2) = P(t_2, t_1)$.

Finally, let

$$\Phi_q(\mathbf{x}) = \bigvee_{\mathcal{T}_{\mathbf{C}}\text{-realisable } \mathbf{z} \subseteq \mathbf{y}} \Phi_{\mathbf{z}}(\mathbf{x}).$$

The structure of the rewriting in Definition 74 is similar to that for *DL-Lite*$^{\flat}_{\text{CORE}}$ in Definition 54. The main differences are as follows:

– when rewriting atoms not having variables in $\mathbf{z}$, we take into account the role inclusions as in the *DL-Lite*$^{\flat}_{\text{RDFS}}$ case; and
– when rewriting role atoms $P_{\mathbf{v}}(t, y_{\mathbf{v}})$ and $P_{\mathbf{v}}(y_{\mathbf{v}}, t)$ for maximally connected subsets $\mathbf{v}$ of variables, we distinguish between different types of subconcepts of $\exists P_{\mathbf{v}}$ and $\exists P_{\mathbf{v}}^-$, respectively: for atomic concepts we follow the rewriting for *DL-Lite*$^{\flat}_{\text{CORE}}$, while for existentially quantified subconcepts we take into account the role inclusions as in the *DL-Lite*$^{\flat}_{\text{RDFS}}$ case.

Example 75. Consider TBox $\mathcal{T}$ from Example 70 and the rooted CQ $q(x) = \exists y.\, \mathsf{hasMusician}(x, y) \wedge \mathsf{Musician}(y)$, same as in Example 55. As before, there are two $\mathcal{T}_{\mathbf{C}}$-realisable subsets of y, namely $\emptyset$ and y, and

$$q_{\emptyset}(x) = q(x) \quad \text{and} \quad q_y(x) = \exists y'.\, \mathsf{hasMusician}(x, y').$$

The rewritings of these CQs are

$$\Phi_{\emptyset}(x) = \exists y.\, \big(\mathsf{hasMusician}(x, y) \vee \mathsf{playsOn}(y, x)\big) \wedge \big(\mathsf{Musician}(y) \vee \exists y'.\, \big(\mathsf{hasMusician}(y', y) \vee \mathsf{playsOn}(y, y')\big)\big),$$
$$\Phi_y(x) = \big(\mathsf{Record}(x) \vee \exists y''.\, (\mathsf{hasMusician}(x, y'') \vee \mathsf{playsOn}(y'', x))\big) \setminus \exists y''.\, \big(\mathsf{hasMusician}(x, y'') \vee \mathsf{playsOn}(y'', x)\big).$$

Evaluating the two rewritings on ABox $\mathcal{A}$ from Example 70, we get $\Phi_{\emptyset}^{\mathcal{A}} = \Phi_y^{\mathcal{A}} = \{\!\{ \mathit{Expectations} : 1 \}\!\}$. Therefore, $\Phi_q^{\mathcal{A}} = (\Phi_{\emptyset} \vee \Phi_y)^{\mathcal{A}} = \{\!\{ \mathit{Expectations} : 2 \}\!\}$, which is equal to $q^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)} = q^{\langle \mathcal{T}, \mathcal{A} \rangle}$ as expected. ◁

We show the correctness of the approach by means of the following generalisation of Lemma 56.

Lemma 76. *Let $q(\mathbf{x})$ be a rooted CQ and $\mathcal{T}$ a DL-Lite$_{\mathcal{R}^-}$ TBox. Then $[q_{\mathbf{z}}, \mathbf{z}']^{Can(\langle \mathcal{T}, \mathcal{A} \rangle)} = \Phi_{\mathbf{z}}^{\mathcal{A}}$ for every bag ABox $\mathcal{A}$ and every $\mathcal{T}_{\mathbf{C}}$-realisable subset $\mathbf{z}$ of $\mathbf{y}$.*

Proof. The proof is similar to the proofs of Lemma 56 and Theorem 66. Consider an arbitrary bag ABox $\mathcal{A}$ and let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be the resulting *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ ontology. We need to show that the bag answers over each bag ABox $\mathcal{A}$ to the rewritings of atoms $A(t)$, $P(t_1, t_2)$ and $\xi_{P_\mathbf{v}}(t, y_\mathbf{v})$ in $q_\mathbf{z}$ as these appear on the left- and right-hand side of (15), as well as in (16), respectively, are equal to the bag answers $[A(t), \emptyset]^{Can(\mathcal{K})}$, $[P(t_1, t_2), \emptyset]^{Can(\mathcal{K})}$, and $[\exists y_\mathbf{v}.\, \xi_{P_\mathbf{v}}(t, y_\mathbf{v}), y_\mathbf{v}]^{Can(\mathcal{K})}$, respectively.

We now argue the correctness of replacements (15) and (16) in Definition 74. To begin, let $\mathcal{T}_\mathbf{R}$ be the set of role inclusions in $\mathcal{T}$, let $I_\mathcal{A} = \langle \Delta^{I_\mathcal{A}}, \cdot^{I_\mathcal{A}} \rangle$ be the bag interpretation corresponding to ABox $\mathcal{A}$—that is, the interpretation defined as $\Delta^{I_\mathcal{A}} = \mathbf{I}$, $a^{I_\mathcal{A}} = a$ for all $a \in \mathbf{I}$, and $S^{I_\mathcal{A}}(\mathbf{a}) = \mathcal{A}(S(\mathbf{a}))$ for all $S \in \mathbf{C} \cup \mathbf{R}$ and tuples of individuals $\mathbf{a}$, and let $Can_\mathbf{R}(\langle \mathcal{T}_\mathbf{R}, \mathcal{A} \rangle)$ be the bag interpretation for the *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$ ontology $\langle \mathcal{T}_\mathbf{R}, \mathcal{A} \rangle$ defined on the basis of $I_\mathcal{A}$ according to Definition 60 (note that we use $I_\mathcal{A}$ instead of $Can_0(\mathcal{K})$ since the definition of the latter in *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ differs from the one in *DL-Lite*$^{\mathsf{b}}_{\mathrm{RDFS}}$). By the definitions of canonical bag interpretations for *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ ontologies as well as concept and role closures, for all individuals a and b, concepts C, and atomic roles P, we have the following equalities:

$$C^{Can(\mathcal{K})}(a) = C^{Can_1(\mathcal{K})} = \mathsf{ccl}_{\mathcal{T}_\mathbf{C}}[a, Can_0(\mathcal{K})](C) = \max(\max\{\mathcal{A}(A_0(a)) \mid A_0 \in \mathbf{C},\ \mathcal{T} \models A_0 \sqsubseteq C\},\ \max\{(\exists R)^{Can_\mathbf{R}(\langle \mathcal{T}_\mathbf{R}, \mathcal{A} \rangle)}(a) \mid R \text{ a role},\ \mathcal{T}_\mathbf{C} \models \exists R \sqsubseteq C\}), \quad (17)$$

$$P^{Can(\mathcal{K})}(a, b) = \mathsf{rcl}_{\mathcal{T}_\mathbf{R}}[(a, b), I_\mathcal{A}](P) = \mathsf{rcl}_{\mathcal{T}}[(a, b), I_\mathcal{A}](P) = \max(\max\{\mathcal{A}(P_0(a, b)) \mid P_0 \in \mathbf{R},\ \mathcal{T} \models P_0 \sqsubseteq P\},\ \max\{\mathcal{A}(P_0(b, a)) \mid P_0 \in \mathbf{R},\ \mathcal{T} \models P_0^- \sqsubseteq P\}). \quad (18)$$

From (17), (18), and the semantics of BCALC queries, we immediately derive the following, for every $a, b \in \mathbf{I}$, $A \in \mathbf{C}$, and $P \in \mathbf{R}$ (see also the derivations for (13) and (14) in the proof of Theorem 66):

$$[A(a), \emptyset]^{Can(\mathcal{K})} = \left(\bigvee_{\mathcal{T}_\mathbf{C} \models C \sqsubseteq A} \zeta_C(a) \right)^{\mathcal{A}} \quad \text{and} \quad [P(a, b), \emptyset]^{Can(\mathcal{K})} = \left(\bigvee_{\mathcal{T} \models R \sqsubseteq P} \xi_R(a, b) \right)^{\mathcal{A}},$$

which proves the claim for atoms of the form $A(t)$ and $P(t_1, t_2)$. The claim for $P_\mathbf{v}$ atoms is proved using (17) and similarly to the proof of Lemma 56. □

Using this lemma, the following theorem can be proved in exactly the same way as Theorem 57.

Theorem 77. *For every rooted CQ $q(\mathbf{x})$ and every DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ *ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ we have that $q^{Can(\mathcal{K})} = \Phi_q^\mathcal{A}$.*

Combining Theorem 73 and Theorem 77, we derive the following corollary.

Corollary 78. *The class of rooted CQs is rewritable to* BCALC *over DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$ *under the UNA.*

Similarly to Corollary 58 in the *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$ case, we can strengthen Corollary 78 and claim that there always exists a rewriting that uses only $\wedge$, $\vee$, $\setminus$, equalities, and existential quantification.

We conclude this section by establishing the data complexity of rooted CQ answering over *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$. Problem BagCert$^{\mathrm{UNA}}$[rooted CQs, *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$] can be decided in the same way as BagCert[rooted CQs, *DL-Lite*$^{\mathsf{b}}_{\mathrm{CORE}}$] (see Section 8.6) with the only difference that the rewriting Φ_q is now computed as in Definition 74. Correctness of the algorithm follows from Theorems 73 and 77, and yields the following data complexity bound.

Corollary 79. BagCert$^{\mathrm{UNA}}$[rooted CQs, *DL-Lite*$^{\mathsf{b}}_{\mathcal{R}^-}$] *is in* LogSpace *in data complexity.*

11. Related work

In this section, we establish bridges between our bag semantics for OBDA and existing work in the literature on data exchange and query answering over *DL-Lite*.

11.1. Bag semantics in data exchange

Hernich and Kolaitis [20] have recently studied CQ answering under bag semantics in the context of data exchange. As is customary in conventional treatments of data exchange [32], their setting considers disjoint source and target database schemas, which are related via source-to-target GLAV mappings. In the data exchange literature it is common to also consider dependencies over the target schema (which can equivalently be seen as ontological axioms); however, the semantics by Hernich and Kolaitis is defined only under the assumption that no such dependencies exist.

Definition 80. A *bag data exchange setting* is a tuple $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle$ where

- $\mathbf{S}$ is a *source* database schema;
- $\mathbf{T}$ is a *target* database schema disjoint from $\mathbf{S}$;
- Σ is a finite set of *global-and-local-as-view* (*GLAV*) mapping assertions (or mappings) of the form

$$q(\mathbf{x}) \rightsquigarrow p(\mathbf{x})$$

  where $q(\mathbf{x})$ is a CQ over $\mathbf{S}$ and $p(\mathbf{x})$ is a CQ over $\mathbf{T}$;[3] and
- $\mathcal{D}$ is a bag database instance over $\mathbf{S}$.

In the conventional set-based case, a solution of a data exchange setting is a finite database over the target schema that satisfies all the mappings together with the source database. Hernich and Kolaitis [20] defined two possible generalisations of this notion to bag semantics.

Definition 81. Let $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle$ be a bag data exchange setting. A bag database instance $\mathcal{B}$ over $\mathbf{T}$ is

- an *incognizant solution* (or *i-solution*), if $q^{\mathcal{D}} \subseteq p^{\mathcal{B}}$ for every mapping $q(\mathbf{x}) \rightsquigarrow p(\mathbf{x})$ in Σ;
- a *cognizant solution* (or *c-solution*), if, for each mapping $\sigma = q(\mathbf{x}) \rightsquigarrow p(\mathbf{x})$ in Σ, there exists a bag database instance $\mathcal{B}_\sigma$ over $\mathbf{T}$ with $q^{\mathcal{D}} \subseteq p^{\mathcal{B}_\sigma}$ such that

$$\biguplus_{\sigma \in \Sigma} \mathcal{B}_\sigma \subseteq \mathcal{B}.$$

In other words, the incognizant semantics adopts the maximal union approach for interpreting mappings defining the same view—that is, when applied to GAV mappings $q_1(\mathbf{x}) \rightsquigarrow S(\mathbf{x})$ and $q_2(\mathbf{x}) \rightsquigarrow S(\mathbf{x})$ for a predicate S, and a database $\mathcal{D}$, an incognizant solution $\mathcal{B}$ must satisfy $q_1^{\mathcal{D}} \cup q_2^{\mathcal{D}} \subseteq S^{\mathcal{B}}$; in contrast, the cognizant semantics adopts the arithmetic union approach and requires that a solution $\mathcal{B}$ satisfies $q_1^{\mathcal{D}} \uplus q_2^{\mathcal{D}} \subseteq S^{\mathcal{B}}$.

Query answering in bag data exchange settings is defined as the problem of computing the bag certain answers to a query over the target schema with respect to the set of solutions.

Definition 82. Let $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle$ be a bag data exchange setting and let q be a CQ over $\mathbf{T}$. The *incognizant certain answers* $\text{i-certain}_{\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle}(q)$ to q with respect to $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle$ are defined as follows:

$$\text{i-certain}_{\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle}(q) = \bigcap_{\substack{\mathcal{B} \text{ is an i-solution for} \\ \langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle}} q^{\mathcal{B}}.$$

The *cognizant certain answers* $\text{c-certain}_{\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle}(q)$ to q with respect to $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle$ are defined in the same way except that c-solutions are considered in the intersection instead of i-solutions.

We are now ready to establish the connection between our OBDA framework and that of Hernich and Kolaitis for data exchange. Note that there are several differences between our OBDA settings, introduced in Definition 6, and data exchange settings. First, target schemas in OBDA are restricted to predicates of arity one and two, whereas in data exchange the arity of target predicates is unbounded. Second, data exchange mappings have CQs on both sides, whereas OBDA mappings have arbitrary BCALC queries on the source side, but only atomic queries of the form $A(x)$ and $P(x, y)$ on the target side. Third, an OBDA setting comes with a TBox, whereas the data exchange setting in [20] does not allow for any dependencies over the target schema. Finally, from a semantics point of view, certain answers in OBDA are defined in terms of (possibly infinite) models, whereas certain answers in data exchange are defined in terms of (finite) database solutions.

We next show that, despite these mismatches, our semantics is compatible with that of Hernich and Kolaitis. For this we note that OBDA and data exchange settings are syntactically compatible if we assume target predicates of arity only one and two, absence of an ontology, and restrict ourselves to GAV mappings with CQs on the source side and atoms $A(x)$ and $P(x, y)$ on the target side. Under these assumptions, we argue that our semantics coincides with the incognizant semantics of Hernich and Kolaitis; furthermore, their cognizant semantics can be simulated in our setting by means of a suitable rewriting of their mappings. Note also that the following proposition is stated for the OBDA semantics without the UNA; however, adopting the UNA would not affect this result, because in case of an empty ontology the two semantics coincide.

Proposition 83. *Let $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D} \rangle$ be a data exchange setting such that $\mathbf{T}$ consists only of unary and binary predicates and all mappings in Σ have atoms of the form $A(x)$ or $P(x, y)$ on the target side. Then, for every CQ q over $\mathbf{T}$,*

[3] Note that Hernich and Kolaitis do not allow for equality atoms in CQs, but this does not affect expressivity.

1. *i-certain*$_{\langle \mathbf{S},\mathbf{T},\Sigma,\mathcal{D}\rangle}(q) = q^{\langle \emptyset,\Sigma,\mathcal{D}\rangle}$, *and*
2. *c-certain*$_{\langle \mathbf{S},\mathbf{T},\Sigma,\mathcal{D}\rangle}(q) = q^{\langle \emptyset,\Sigma',\mathcal{D}\rangle}$, *where* Σ' *is obtained from* Σ *by replacing, for each atom* $S(\mathbf{x})$*, all the mappings with* $S(\mathbf{x})$ *on the target side with*

$$\bigvee^{\cdot}_{q(\mathbf{x})\rightsquigarrow S(\mathbf{x})\ in\ \Sigma} \quad q(\mathbf{x}) \ \rightsquigarrow \ S(\mathbf{x}).$$

Proof. Given a bag database $\mathcal{B}$ over $\mathbf{T}$, let $I_{\mathcal{B}}$ be the bag interpretation corresponding to $\mathcal{B}$—that is, the interpretation having the set of individuals (i.e., constants) appearing in $\mathcal{B}$ as the domain and satisfying $S^{I_{\mathcal{B}}} = S^{\mathcal{B}}$ for every predicate $S \in \mathbf{T}$. We claim that, for every bag database $\mathcal{B}$ over $\mathbf{T}$,

1. $\mathcal{B}$ is an i-solution for $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D}\rangle$ if and only if $I_{\mathcal{B}}$ is a model of $\langle \emptyset, \Sigma, \mathcal{D}\rangle$, and
2. $\mathcal{B}$ is a c-solution for $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D}\rangle$ if and only if $I_{\mathcal{B}}$ is a model of $\langle \emptyset, \Sigma', \mathcal{D}\rangle$.

The first claim follows from Definitions 81 and 9 of the semantics of data exchange and OBDA, respectively.

Consider now the second claim. The fact that $\mathcal{B}$ is a c-solution for $\langle \mathbf{S}, \mathbf{T}, \Sigma, \mathcal{D}\rangle$ means that there are bag database instances $\mathcal{B}_\sigma$, for $\sigma \in \Sigma$, such that $q^{\mathcal{D}} \subseteq (S(\mathbf{x}))^{\mathcal{B}_\sigma}$ for each $\sigma = q(\mathbf{x}) \rightsquigarrow S(\mathbf{x})$ and $\biguplus_{\sigma\in\Sigma} \mathcal{B}_\sigma \subseteq \mathcal{B}$. This is equivalent to the fact that, for each atom $S(\mathbf{x})$,

$$\biguplus_{q(\mathbf{x})\rightsquigarrow S(\mathbf{x})\ \text{in}\ \Sigma} \quad q^{\mathcal{D}} \subseteq S^{I_{\mathcal{B}}},$$

which means that $I_{\mathcal{B}}$ is a model of $\langle \emptyset, \Sigma', \mathcal{D}\rangle$ by the definition of Σ'. To conclude the proof, note that, since an ontology with an empty TBox has a unique finite model that is minimal with respect to bag inclusion, by Lemma 21, it is enough to consider only finite interpretations when computing the bag certain answers to q. □

Note that the second statement in this proposition is essentially illustrated in Example 11.

We conclude this section by observing that, at least in principle, we could have defined a "cognizant" bag semantics for *DL-Lite*$^{\flat}_{\mathcal{R}}$, which is based on arithmetic union rather than maximal union. Under such a semantics, inclusions $A \sqsubseteq C$ and $B \sqsubseteq C$ would be satisfied by a bag interpretation I only if $A^I \uplus B^I \subseteq C^I$; this is in contrast to our current "incognizant" semantics where we require that $A^I \cup B^I \subseteq C^I$. Such a semantics would, however, come with clear disadvantages. For example, every model I of the ontology with TBox $\{A \sqsubseteq B,\ \ B \sqsubseteq A, C \sqsubseteq B\}$ and ABox $\{\!\{ A(a) : 1,\ \ B(a) : 1,\ \ C(a) : 1 \}\!\}$ would need to satisfy

$$A^I \subseteq B^I, \qquad B^I \subseteq A^I, \qquad C^I \subseteq B^I, \qquad A^I \uplus C^I \subseteq B^I,$$

and, hence, would have to include infinitely many occurrences of element a^I in both A^I and B^I.

11.2. Count aggregate queries over ontologies

Our approach to OBDA query answering under bag semantics is closely related to existing approaches for answering conjunctive counting aggregate queries over *DL-Lite*$_{\mathcal{R}}$ [36,38]. Indeed, under bag semantics, CQs are intrinsically equipped with counting power: the result of evaluating a CQ over a (set or bag) interpretation under bag semantics is a bag of answer tuples, where each tuple comes with a multiplicity (i.e., a "count").

Calvanese et al. [38] proposed an epistemic semantics, where query answers are obtained by evaluating the query over the (finite) set of all ABox facts entailed by the ontology. Although this approach is well-suited for practical implementations, it can easily lead to counter-intuitive answers. To remedy this, Kostylev and Reutter [36] proposed a certain answers semantics that requires the query to be evaluated over all models of the ontology, which yields more intuitive answers at the expense of increased computational cost.

In what follows we take a closer look at Kostylev and Reutter's framework, which is formalised in the following definition. The main difference between their setting and ours is that they consider set ABoxes and conventional set-based semantics of ontologies.

Definition 84. A *count aggregate query* is the expression $q_{\mathrm{c}}(\mathbf{x}, \mathsf{Count}()) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$, where $\phi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over atomic concepts and roles. A (set) interpretation I *satisfies* $q_{\mathrm{c}}(\mathbf{a}, m)$, for a tuple $\mathbf{a}$ over $\mathbf{I}$ with $|\mathbf{a}| = |\mathbf{x}|$ and a number $m \in \mathbb{N}_0^\infty$, if there are exactly m valuations $\lambda : \mathbf{x} \cup \mathbf{y} \cup \mathbf{I} \to \Delta^I$ with $\lambda(\mathbf{x}) = \mathbf{a}^I$ and $\lambda(a) = a^I$ for each $a \in \mathbf{I}$ that make $\phi(\mathbf{x}, \mathbf{y})$ true in I. A number $n \in \mathbb{N}_0^\infty$ is in the *count aggregate certain answers* $Cert(q_{\mathrm{c}}, \mathbf{a}, \mathcal{K})$ to q_{c} for a tuple of individuals $\mathbf{a}$ and *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{K}$ if $n \leq \min\{m \in \mathbb{N}_0^\infty \mid \text{there is } I \models \mathcal{K} \text{ satisfying } q_{\mathrm{c}}(\mathbf{a}, m)\}$. *Count aggregate certain answers* $Cert^{\mathsf{UNA}}(q_{\mathrm{c}}, \mathbf{a}, \mathcal{K})$ under the UNA are defined in the same way except that only interpretations satisfying the UNA are considered.

Intuitively, the count associated to $\mathbf{a}$ in the count aggregate certain answers to q_c is the minimum number of matching valuations over all models of the ontology. The following proposition establishes a correspondence between our setting and theirs: under suitable restrictions on the TBox, assuming only set ABoxes, and adopting the UNA, the certain answers to CQs under our bag semantics coincide with the answers to the corresponding count query as given in Definition 84.

Proposition 85. *For every DL-Lite$_\mathcal{R}$ ontology $\mathcal{K}^s = \langle \mathcal{T}, \mathcal{A}^s \rangle$ such that $\mathcal{T}$ does not contain any inclusions of the form $\exists R \sqsubseteq C$, every count aggregate query $q_c(\mathbf{x}, \mathsf{Count}()) = \exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$, every tuple of individuals $\mathbf{a}$, and every $n \in \mathbb{N}_0^\infty$,*

$$n \in Cert^{\mathsf{UNA}}(q_c, \mathbf{a}, \mathcal{K}^s) \text{ if and only if } n \leq q^{\mathcal{K}^b, \mathsf{UNA}}(\mathbf{a}),$$

where q is the CQ $\exists \mathbf{y}.\, \phi(\mathbf{x}, \mathbf{y})$, $\mathcal{K}^b$ is the DL-Lite$^b_\mathcal{R}$ ontology obtained from $\mathcal{K}^s$ by considering $\mathcal{A}^s$ as a bag ABox (i.e., as the bag ABox assigning 1 to all assertions in $\mathcal{A}^s$ and 0 to all others), and $q^{\mathcal{K}^b, \mathsf{UNA}}$ is the certain answers to q over $\mathcal{K}^b$ under the UNA.

Proof. First, we claim that for every set interpretation I^s that is a model of $\mathcal{K}^s$ there exists a bag interpretation I^b that is a model of $\mathcal{K}^b$ such that, for every tuple of individuals $\mathbf{a}$, $q^{I^b}(\mathbf{a}) = m$ where m is the number with I^s satisfying $q_c(\mathbf{a}, m)$. Indeed, given such an I^s we can take as I^b the bag version of I^s—that is, the bag interpretation such that $S^{I^b}(\mathbf{b}) = 1$ for an atomic concept or role S and individuals $\mathbf{b}$ if $\mathbf{b} \in S^{I^s}$ and $S^{I^b}(\mathbf{b}) = 0$ otherwise. Bag interpretation I^b is a model of $\mathcal{K}^b$ because the restriction on the TBox rules out any forced increase of multiplicities, while $q^{I^b}(\mathbf{a}) = m$ holds for every $\mathbf{a}$ because each valuation contributes to $q^{I^b}(\mathbf{a})$ with multiplicity 0 and 1, and the ones with 1 are precisely those that turn ϕ to true.

Second, we claim that for every bag interpretation I^b that is a model of $\mathcal{K}^b$ there exists a set interpretation I^s that is a model of $\mathcal{K}^s$ such that I^s satisfies $q_c(\mathbf{a}, m)$ for every tuple of individuals $\mathbf{a}$ and for a number $m \leq q^{I^b}(\mathbf{a})$. Indeed, given such an I^b we can take as I^s the "characteristic" version of I^b—that is, the set interpretation such that $\mathbf{b} \in S^{I^s}$ for an atomic concept or role S and individuals $\mathbf{b}$ if and only if $S^{I^b}(\mathbf{b}) \geq 1$. Set interpretation I^s is a model of $\mathcal{K}^s$ just by definition, while I^s additionally satisfies $q_c(\mathbf{a}, m)$ for a tuple $\mathbf{a}$ and a number $m \leq q^{I^b}(\mathbf{a})$ because, by construction, a valuation contributes to $q^{I^b}(\mathbf{a})$ a multiplicity greater than 0 if and only if it turns ϕ to true.

These two claims immediately imply the statement of the proposition. □

Note, however, that both the UNA and the restriction on TBoxes are necessary for Proposition 85 to hold, and dropping any of these makes the two frameworks incompatible. Indeed, if the UNA is not adopted, then, for the simple ontology $\langle \emptyset, \{A(a), A(b)\} \rangle$, the aggregate query counting $A(y)$ has certain answer 1, while the corresponding CQ has the empty tuple with multiplicity 2 in the answer. Similarly, for the ontology $\langle \{B \sqsubseteq \exists P, \exists P^- \sqsubseteq A\}, \{B(a), B(b)\} \rangle$, we have the same situation if we drop the restriction on the TBox.

We conclude by pointing out that the work by Kostylev and Reutter is also strongly related to existing approaches in the database literature for answering counting aggregate queries in the presence of incomplete information [39–41]. As observed by Kostylev and Reutter, however, these approaches are not directly applicable to answering counting queries in the presence of an ontology, and we refer to [36] for a detailed discussion.

11.3. Other related work

Jiang [42] proposed a bag semantics for the description logic $\mathcal{ALC}$. The author focuses on satisfiability checking and provides a tableaux-based decision procedure. Their semantics is, however, incomparable to ours. For example, concepts of the form $\exists R.\top$ are not interpreted as the bag projection on the first argument of role R, which makes the semantics incompatible with SQL.

Query answering and optimisation under bag semantics have received significant attention in the database literature [22,23,27,29–31,33,43]. 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, query containment, and query equivalence. More recently, Console et al. [44] studied query answering under bag semantics in incomplete relational databases. Query answering and its data complexity under bag semantics have been recently studied as well in the setting of the Semantic Web query languages [45,46].

12. Conclusion and future work

In this article, we have proposed a novel bag semantics for OBDA and studied the computational properties of its associated query answering problems. The key advantage of our semantics is that it allows us to faithfully represent arbitrary GAV mappings (and not just those whose source query involves duplicate elimination) in a way that is compatible with SQL. Furthermore, our semantics is compatible with existing bag-based semantics for databases with incomplete information and data exchange.

We see many interesting directions for future work. First, we are planning to extend the query language to allow for database-style aggregate functions and to study suitable restrictions ensuring rewritability of such queries. Second, it would

be interesting to try to push the rewritability boundaries for CQs to include some constant-free Boolean queries. For this, an interesting starting point could be the notion of *local concepts* and *queries* proposed by Gutiérrez-Basulto et al. [47] as a means of regaining decidability of query answering over *DL-Lite*$_{\mathcal{R}}$ for the class of CQs with inequalities. Finally, our rewriting algorithms are not designed with an efficient implementation in mind. We plan to develop a practically applicable rewriting algorithm for our bag semantics.

Acknowledgements

This work was supported by the Royal Society under a University Research Fellowship, the EPSRC projects ED3 (EP/N014359/1) and DBOnto (EP/L012138/1), and the Research Council of Norway via the Sirius SFI (237889).

Appendix A. BALG^1: algebraic query language for bag databases

In this appendix we introduce the algebraic query language for bag databases of Grumbach and Milo [22], called BALG^1. Queries in BALG^1 are algebra expressions built upon predicates in a database schema by the composition of the operations defined in Section 2.3 for bags, as well as the operations of *tupling* τ, *attribute projection* α, *bagging* β, *Cartesian product* $\times$, *selection* σ, and *projection* π.[4] The semantics of these algebra expressions is defined formally below, where the operations are grouped into those that operate on tuples over the database domain $\mathbf{I}$ and those that operate on bags of tuples.

1. Operations on tuples:
   - *tupling* $\tau(a_1, ..., a_k)$ for $a_1, \ldots, a_k \in \mathbf{I}$ is the k-ary tuple $(a_1, ..., a_k)$;
   - *attribute projection* $\alpha_i(\mathbf{a})$ for a tuple $\mathbf{a}$ over $\mathbf{I}$ of size k and $i \in [1, k]$ is the i-th component of $\mathbf{a}$;
   - *bagging* $\beta(\mathbf{a})$ for a tuple $\mathbf{a}$ over $\mathbf{I}$ is the bag of tuples over $\mathbf{I}$ of size $|\mathbf{a}|$ assigning 1 to $\mathbf{a}$ and 0 to other tuples.
2. Operations on bags of tuples:
   - *intersection* $\cap$, *maximal union* $\cup$, *arithmetic union* $\uplus$, *difference* $-$, and *duplicate elimination* ε on bags of tuples over $\mathbf{I}$ of the same size are defined as in Section 2.3;
   - *Cartesian product* $\Omega_1 \times \Omega_2$ of bags Ω_1 and Ω_2 of tuples of size k and ℓ, respectively, over $\mathbf{I}$ is the bag of tuples of size $k + \ell$ assigning $\Omega_1(\mathbf{a}) \times \Omega_2(\mathbf{b})$ to the concatenation $\mathbf{a}, \mathbf{b}$ of each two tuples $\mathbf{a}$ and $\mathbf{b}$ of size k and ℓ, respectively;
   - *selection* $\sigma_{E_1(X)=E_2(X)}(\Omega)$ of a bag Ω of tuples over $\mathbf{I}$ of size k for BALG^1 algebra expressions E_1 and E_2 with a tuple variable X is the bag of tuples over $\mathbf{I}$ of size k that assigns $\Omega(\mathbf{a})$ to each $\mathbf{a}$ with $E_1(\mathbf{a}) = E_2(\mathbf{a})$ and 0 to all other tuples;
   - *projection* $\pi_{i_1,...,i_n}(\Omega)$ of a bag Ω of tuples over $\mathbf{I}$ of size k for $i_1, \ldots, i_n \in [1, k]$ and $n \geq 1$ is the bag of tuples of size n over $\mathbf{I}$ that assigns to each tuple $\mathbf{a}'$ of size n the sum of $\Omega(\mathbf{a})$ over all $\mathbf{a}$ that have $\mathbf{a}'$ as components $i_1, \ldots, i_n$.

The *bag answers* $E^{\mathcal{D}}$ to a BALG^1 algebra expression E (operating on bags) over a database instance $\mathcal{D}$ over the same schema is the bag of tuples resulting in the evaluation of E on $\mathcal{D}$.

We now define the decision problem corresponding to computing the bag answers to BALG^1 algebra expressions as it was introduced in [22], where we again assume that all numbers in the input are represented in unary and the bag database instance is explicitly defined only for a finite number of facts.

QueryAnswering$^{=}$[BALG^1]

Input: BALG^1 algebra expression E, bag database instance $\mathcal{D}$, tuple $\mathbf{a}$ of constants over $\mathbf{I}$, and number $k \in \mathbb{N}_0$.

Question: Is $E^{\mathcal{D}}(\mathbf{a}) = k$?

The *data complexity* of this problem is the complexity when the expression E is considered to be fixed and only $\mathcal{D}$, $\mathbf{a}$, and k form the input.

We stress here that in this problem the question is whether the equality $\Phi^{\mathcal{D}}(\mathbf{a}) = k$ holds. However, we are more interested in the threshold version QueryAnswering[BALG^1] where $\Phi^{\mathcal{D}}(\mathbf{a}) \geq k$ is checked instead.

The following proposition shows that, in data complexity, our problem is not any harder than the one of Grumbach and Milo.

Proposition 86. QueryAnswering[BALG^1] *is* AC^0 *reducible to* QueryAnswering$^{=}$[BALG^1] *in data complexity.*

[4] Note that the language of Grumbach and Milo [22] includes a restructuring operation instead of projection. Our convenient deviation does not change the expressivity of the language since these two operations are expressible via each other in the presence of the other operations [22].

Proof. Let E be a fixed BALG^1 algebra expression, $\mathcal{D}$ be a bag database instance, $\mathbf{a}$ be a tuple of constants over $\mathbf{I}$, and k be a number in $\mathbb{N}_0$. Consider the bag database instance $\mathcal{D}_0$ over $\mathbf{I}$ and schema extended with a fresh predicate T of arity $|\mathbf{a}|$ in which T assigns k to $\mathbf{a}$ and 0 to all other tuples of size $|\mathbf{a}|$, while every other predicate is as in $\mathcal{D}$.

Let $E_0 = T - E$. We claim that $E^{\mathcal{D}}(\mathbf{a}) \geq k$ if and only if $E_0^{\mathcal{D}_0}(\mathbf{a}) = 0$. Indeed, assuming $E^{\mathcal{D}}(\mathbf{a}) \geq k$, we derive $E_0^{\mathcal{D}_0}(\mathbf{a}) = 0$. Similarly, assuming $E^{\mathcal{D}}(\mathbf{a}) < k$, we derive $E_0^{\mathcal{D}_0}(\mathbf{a}) > 0$.

The above many-one reduction is computable, for each $\mathcal{D}$, $\mathbf{a}$, and k, by a Boolean circuit with arbitrary fan-in AND and OR gates whose depth depends only on E. We conclude that language $\{(\mathcal{D}, \mathbf{a}, k) \mid E^{\mathcal{D}}(\mathbf{a}) \geq k\}$ is contained in $\{(\mathcal{D}, \mathbf{a}, k) \mid E^{\mathcal{D}}(\mathbf{a}) = k\}$ under LogSpace-uniform AC^0 reductions, as required. □

Grumbach and Milo [22] studied the expressive power of QueryAnswering$^{=}$[BALG^1] and showed that it is strictly in between AC^0 and LogSpace. Therefore, we can conclude the following fact.

Corollary 87. QueryAnswering[BALG^1] *is in* LogSpace *in data complexity.*

Appendix B. Relationship between BCALC and BALG^1

Theorem 88. *For each* BCALC *query* Φ *there is a* BALG^1 *algebra expression* E_Φ *such that* $\Phi^{\mathcal{D}} = E_\Phi^{\mathcal{D}}$ *for every bag database instance* $\mathcal{D}$.

Proof. In this proof we assume that all the variables $\mathbf{X}$ are globally ordered, and each tuple of repetition-free variables has its variables according to this order; moreover, for a BCALC query $\exists\mathbf{y}.\Psi(\mathbf{x},\mathbf{y})$ we assume that all $\mathbf{y}$ are after all $\mathbf{x}$ in the order, which is done without loss of generality, because BCALC is agnostic to renaming of variables.

We define BALG^1 algebra expression E_Φ for each BCALC query Φ by induction on the structure of Φ as follows:

- if $\Phi(x_1,\ldots,x_n) = S(t_1,\ldots,t_k)$ for a predicate S and a tuple of terms $t_1,\ldots,t_k$ with the first occurrence of each x_i in a position p_i, then, for $c_1,\ldots,c_k \in \mathbf{I}$ such that c_j is fresh if t_j is a variable and c_j is t_j otherwise for each j,

$$E_\Phi = \pi_{p_1,\ldots,p_n}\big(\sigma_{\alpha_1(X)=\alpha_{m_1}(X)}\big(\cdots\sigma_{\alpha_k(X)=\alpha_{m_k}(X)}\big(S \times \beta(\tau(c_1,\ldots,c_k))\big)\cdots\big)\big),$$

  where, for each j, $m_j = p_i$ if t_j is x_i and $m_j = j+k$ otherwise;
- if $\Phi(\mathbf{x}) = \Psi(\mathbf{x}) \wedge (x_1 = x_2)$ with x_1 and x_2 in positions i and j in $\mathbf{x}$, then $E_\Phi = \sigma_{\alpha_i(X)=\alpha_j(X)}(E_\Psi)$;
- if $\Phi(x_1,\ldots,x_k) = \Psi(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_k) \wedge (x_i = x_j)$ for $i,j \in [1,k]$, $i \neq j$, then $E_\Phi = \pi_{1,\ldots,j-1,i,j+1,\ldots,k}(E_\Psi)$;
- if $\Phi(x_1,\ldots,x_k) = \Psi(x_1,\ldots,x_k) \wedge (x_i = a)$ for $i \in [1,k]$, then $E_\Phi = \pi_{1,\ldots,k}(\sigma_{\alpha_i(X)=\alpha_{k+1}(X)}(E_\Psi \times \beta(\tau(a))))$;
- if $\Phi(x_1,\ldots,x_k) = \Psi_1(x_{i_1},\ldots,x_{i_m}) \wedge \Psi_2(x_{j_1},\ldots,x_{j_n})$ with $\{i_1,\ldots,i_m,j_1,\ldots,j_n\} = \{1,\ldots,k\}$, then

$$E_\Phi = \pi_{p_1,\ldots,p_k}\big(\sigma_{\alpha_{m+1}(X)=\alpha_{s_1}(X)}\big(\cdots\sigma_{\alpha_{m+n}(X)=\alpha_{s_n}(X)}\big(E_{\Psi_1} \times E_{\Psi_2}\big)\cdots\big)\big),$$

  where, for each $\ell \in [1,k]$, p_ℓ is the position of the first occurrence of ℓ in $i_1,\ldots,i_m,j_1,\ldots,j_n$ and, for each $\ell \in [1,n]$, s_ℓ is the position of the first occurrence of j_ℓ in $i_1,\ldots,i_m,j_1,\ldots,j_n$;
- if $\Phi(\mathbf{x}) = \exists\mathbf{y}.\Psi(\mathbf{x},\mathbf{y})$, then $E_\Phi = \pi_{1,\ldots,|\mathbf{x}|}(E_\Psi)$;
- if $\Phi(\mathbf{x}) = \Psi_1(\mathbf{x}) \vee \Psi_2(\mathbf{x})$, then $E_\Phi = E_{\Psi_1} \cup E_{\Psi_2}$;
- if $\Phi(\mathbf{x}) = \Psi_1(\mathbf{x}) \veebar \Psi_2(\mathbf{x})$, then $E_\Phi = E_{\Psi_1} \uplus E_{\Psi_2}$;
- if $\Phi(\mathbf{x}) = \Psi_1(\mathbf{x}) \setminus \Psi_2(\mathbf{x})$, then $E_\Phi = E_{\Psi_1} - E_{\Psi_2}$; and
- if $\Phi(\mathbf{x}) = \delta\,\Psi(\mathbf{x})$, then $E_\Phi = \varepsilon(E_\Psi)$.

It is now straightforward to check that $\Phi^{\mathcal{D}} = E_\Phi^{\mathcal{D}}$ for every bag database $\mathcal{D}$. □

The following fact is a direct consequence of Corollary 87 and Proposition 88.

Proposition 88 QueryAnswering[BCALC] *is in* LogSpace *in data complexity.*

References

[1] C. Nikolaou, E.V. Kostylev, G. Konstantinidis, M. Kaminski, B. Cuenca Grau, I. Horrocks, The bag semantics of ontology-based data access, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 1224–1230.

[2] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, J. Data Semant. 10 (2008) 133–173.

[3] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, D.F. Savo, The MASTRO system for ontology-based data access, Semant. Web 2 (1) (2011) 43–53.

[4] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, Ontologies and databases: the DL-Lite approach, in: Proceedings of 5th International Summer School on Reasoning Web: Semantic Technologies for Information Systems, Tutorial Lectures, 2009, pp. 255–356.

[5] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, G. Xiao, Ontop: answering SPARQL queries over relational databases, Semant. Web 8 (3) (2017) 471–487.

[6] E. Kharlamov, D. Hovland, E. Jiménez-Ruiz, D. Lanti, H. Lie, C. Pinkel, M. Rezk, M.G. Skjæveland, E. Thorstensen, G. Xiao, D. Zheleznyakov, I. Horrocks, Ontology based access to exploration data at Statoil, in: Part II of the Proceedings of the 14th International Semantic Web Conference, 2015, pp. 93–112.
[7] E. Kharlamov, T.P. Mailis, K. Bereta, D. Bilidas, S. Brandt, E. Jiménez-Ruiz, S. Lamparter, C. Neuenstadt, Ö.L. Özçep, A. Soylu, C. Svingos, G. Xiao, D. Zheleznyakov, D. Calvanese, I. Horrocks, M. Giese, Y.E. Ioannidis, Y. Kotidis, R. Möller, A. Waaler, A semantic approach to polystores, in: Proceedings of the IEEE International Conference on Big Data, 2016, pp. 2565–2573.
[8] M. Lenzerini, Data integration: a theoretical perspective, in: Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 2002, pp. 233–246.
[9] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: the DL-Lite family, J. Autom. Reason. 39 (3) (2007) 385–429.
[10] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web ontology language profiles, second edition, W3C Recommendation.
[11] A. Doan, A.Y. Halevy, Z.G. Ives, Principles of Data Integration, Morgan Kaufmann, 2012.
[12] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, J. Artif. Intell. Res. 36 (2009) 1–69.
[13] G. Gottlob, S. Kikot, R. Kontchakov, V.V. Podolskii, T. Schwentick, M. Zakharyaschev, The price of query rewriting in ontology-based data access, Artif. Intell. 213 (2014) 42–59.
[14] M. Bienvenu, S. Kikot, R. Kontchakov, V.V. Podolskii, M. Zakharyaschev, Ontology-mediated queries: combined complexity and succinctness of rewritings via circuit complexity, J. ACM 65 (5) (2018) 28.
[15] F.D. Pinto, D. Lembo, M. Lenzerini, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, D.F. Savo, Optimizing query rewriting in ontology-based data access, in: Joint 2013 EDBT/ICDT Conferences, EDBT '13 Proceedings, Genoa, Italy, March 18–22, 2013, 2013, pp. 561–572.
[16] R. Kontchakov, M. Rezk, M. Rodriguez-Muro, G. Xiao, M. Zakharyaschev, Answering SPARQL queries over databases under OWL 2 QL entailment regime, in: Part I of the Proceedings of the 13th International Semantic Web Conference, 2014, pp. 552–567.
[17] J.F. Sequeda, M. Arenas, D.P. Miranker, OBDA: query rewriting or materialization? in practice, both!, in: The Semantic Web – ISWC 2014 – 13th International Semantic Web Conference, Riva del Garda, Italy, October 19–23, 2014, Proceedings, Part I, 2014, pp. 535–551.
[18] A. Gupta, I.S. Mumick, Materialized Views: Techniques, Implementations, and Applications, MIT Press, 1999.
[19] H. García-Molina, J.D. Ullman, J. Widom, Database Systems: The Complete Book, 2nd edition, Pearson Education, 2009.
[20] A. Hernich, P.G. Kolaitis, Foundations of information integration under bag semantics, in: Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, 2017, pp. 1–12.
[21] M. Bienvenu, C. Lutz, F. Wolter, Query containment in description logics reconsidered, in: Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning, 2012, pp. 221–231.
[22] S. Grumbach, T. Milo, Towards tractable algebras for bags, J. Comput. Syst. Sci. 52 (3) (1996) 570–588.
[23] L. Libkin, L. Wong, Query languages for bags and aggregate functions, J. Comput. Syst. Sci. 55 (2) (1997) 241–272.
[24] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, J. Artif. Intell. Res. 36 (2009) 1–69.
[25] U. Dayal, N. Goodman, R.H. Katz, An extended relational algebra with control over duplicate elimination, in: Proceedings of the ACM Symposium on Principles of Database Systems, 1982, pp. 117–123.
[26] J. Albert, Algebraic properties of bag data types, in: Proceedings of the 17th International Conference on Very Large Data Bases, 1991, pp. 211–219.
[27] S. Grumbach, L. Libkin, T. Milo, L. Wong, Query languages for bags: expressive power and complexity, SIGACT News 27 (2) (1996) 30–44.
[28] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[29] S. Chaudhuri, M.Y. Vardi, Optimization of real conjunctive queries, in: Proceedings of the 12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1993, pp. 59–70.
[30] T.S. Jayram, P.G. Kolaitis, E. Vee, The containment problem for REAL conjunctive queries with inequalities, in: Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 2006, pp. 80–89.
[31] S. Cohen, Equivalence of queries that are sensitive to multiplicities, VLDB J. 18 (3) (2009) 765–785.
[32] R. Fagin, P.G. Kolaitis, R.J. Miller, L. Popa, Data exchange: semantics and query answering, Theor. Comput. Sci. 336 (1) (2005) 89–124.
[33] Y.E. Ioannidis, R. Ramakrishnan, Containment of conjunctive queries: beyond relations as sets, ACM Trans. Database Syst. 20 (3) (1995) 288–324.
[34] D. Lembo, V. Santarelli, D.F. Savo, Graph-based ontology classification in OWL 2 QL, in: Proceedings of the 10th International Conference on The Semantic Web: Semantics and Big Data, 2013, pp. 320–334.
[35] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: query answering under expressive relational constraints, J. Artif. Intell. Res. 48 (2013) 115–174.
[36] E.V. Kostylev, J.L. Reutter, Complexity of answering counting aggregate queries over DL-Lite, J. Web Semant. 33 (2015) 94–111.
[37] S. Kikot, R. Kontchakov, M. Zakharyaschev, Conjunctive query answering with OWL 2 QL, in: Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning, 2012, pp. 275–285.
[38] D. Calvanese, E. Kharlamov, W. Nutt, C. Thorne, Aggregate queries over ontologies, in: Proceedings of the 2nd International Workshop on Ontologies and Information Systems for the Semantic Web, 2008, pp. 97–104.
[39] M. Arenas, L.E. Bertossi, J. Chomicki, X. He, V. Raghavan, J.P. Spinrad, Scalar aggregation in inconsistent databases, Theor. Comput. Sci. 296 (3) (2003) 405–434.
[40] L. Libkin, Data exchange and incomplete information, in: Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 2006, pp. 60–69.
[41] F.N. Afrati, P.G. Kolaitis, Answering aggregate queries in data exchange, in: Proceedings of the 27th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 2008, pp. 129–138.
[42] Y. Jiang, Description logics over multisets, in: Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web, 2010, pp. 1–12.
[43] L. Libkin, L. Wong, New techniques for studying set languages, bag languages and aggregate functions, in: Proceedings of the 13th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1994, pp. 155–166.
[44] L.L. Marco Console, Paolo Guagliardo, On querying incomplete information in databases under bag semantics, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 993–999.
[45] M. Kaminski, E.V. Kostylev, B. Cuenca Grau, Query nesting, assignment, and aggregation in SPARQL 1.1, ACM Trans. Database Syst. 42 (3) (2017) 17.
[46] R. Angles, C. Gutierrez, The multiset semantics of SPARQL patterns, in: Part I of the Proceedings of the 15th International Semantic Web Conference, 2016, pp. 20–36.
[47] V. Gutiérrez-Basulto, Y.A. Ibáñez-García, R. Kontchakov, E.V. Kostylev, Queries with negation and inequalities over lightweight ontologies, J. Web Semant. 35 (2015) 184–202.