Query-based Comparison of Mappings in Ontology-based Data Access

Meghyn Bienvenu, Riccardo Rosati

Query-based Comparison of Mappings in Ontology-based Data Access

Meghyn Bienvenu
LIRMM - CNRS, INRIA,
& Université de Montpellier
Montpellier, France

Riccardo Rosati
DIAG
Sapienza Università di Roma
Rome, Italy

Abstract

An ontology-based data access (OBDA) system is composed of one or more data sources, an ontology that provides a conceptual view of the data, and declarative mappings that relate the data and ontology schemas. In order to debug and optimize such systems, it is important to be able to analyze and compare OBDA specifications. Recent work in this direction compared specifications using classical notions of equivalence and entailment, but an interesting alternative is to consider query-based notions, in which two specifications are deemed equivalent if they give the same answers to the considered query or class of queries for all possible data sources. In this paper, we define such query-based notions of entailment and equivalence of OBDA specifications and investigate the complexity of the resulting analysis tasks when the ontology is formulated in (fragments of) *DL-Lite*$_R$.

1 Introduction

Ontology-based data access (OBDA) (Poggi et al. 2008) is a recent paradigm that proposes the use of an *ontology* as a conceptual, reconciled view of the information stored in a set of existing *data sources*. The connection between the ontology and the data sources is provided by declarative *mappings* that relate the elements of the ontology with the elements of the data sources. The ontology layer is the virtual interface used to access data, through *queries* formulated in the vocabulary of the ontology.

Due to the recent availability of techniques and systems for query processing in this setting (Calvanese et al. 2011; Rodriguez-Muro, Kontchakov, and Zakharyaschev 2013), the OBDA approach has begun to be experimented in real applications (see e.g. (Antonioli et al. 2013; Kharlamov et al. 2013; Giese et al. 2015)). In these projects, the construction, debugging and maintenance of the *OBDA specification*, consisting of the ontology, the schemas of the data sources, and the mapping, is a non-trivial task. In fact, the size and the complexity of the ontology and, especially, the mappings makes the management of such specifications a practical issue in these projects. Providing formal tools for supporting the above activities is therefore crucial for the successful deployment of OBDA solutions.

In addition, the OBDA specification plays a major role in query answering, since its form may affect the system performance in answering queries: different, yet semantically equivalent specifications may give rise to very different execution times for the same query. Thus, the study of notions of equivalence and formal comparison of OBDA specifications is also important for optimizing query processing in OBDA systems. Indeed, some systems already implement forms of optimization based on such transformations (see, e.g., (Rodriguez-Muro, Kontchakov, and Zakharyaschev 2013)).

Most of the work on OBDA thus far has focused on the query answering task, often in a simplified setting without mappings. Very little attention has been devoted to the formal analysis of OBDA specifications. The first approach that explicitly focuses on the formal analysis of OBDA specifications is (Lembo et al. 2014; 2015), whose aim is the discovery of redundancies and inconsistencies in mappings. That work utilizes the classical notions of logical equivalence and entailment to compare OBDA specifications. While it is very natural to resort to such classical notions, an appealing alternative in many applications is to adopt *query-based* notions of equivalence and entailment, in which two specifications are compared with respect to a given query or a given class of queries, and are deemed equivalent if they give the same answers to the considered queries for all possible data sources. This idea has already been investigated in several works by the DL community for the purpose of comparing TBoxes and knowledge bases (Konev et al. 2009; Lutz and Wolter 2010; Kontchakov, Wolter, and Zakharyaschev 2010; Konev et al. 2011; Botoeva et al. 2014; 2015), as well as in the data exchange and schema mapping literature (Fagin et al. 2008; Gottlob, Pichler, and Savenkov 2011). To the best of our knowledge, it has never been explicitly considered for OBDA specifications.

Conjunctive queries (CQs) are the type of queries most commonly considered in OBDA. Therefore, a first natural choice would be to compare OBDA specifications with respect to the whole class of CQs. We thus define a notion of *CQ-entailment* between OBDA specifications, which is a natural generalization of previously studied notions of query entailment between ontologies. We also consider the important subclass of *instance queries (IQs)* and analyze the notion of *IQ-entailment* between specifications. Moreover, in many application contexts, only a (small) set of predefined queries are available or of interest to the user(s). In such cases, it may be more appropriate to tailor the comparison of specifications to a specific set of queries. For this reason,

we also study the notions of *single CQ-entailment* and *single IQ-entailment*, which compare specifications with respect to a single CQ or IQ.

Each of the preceding notions of entailment induces a corresponding notion of equivalence of OBDA specifications. Being weaker than classical logical equivalence, such query-based notions of equivalence lay the formal foundations for more powerful types of optimizations of OBDA specifications, as it is possible to consider a larger set of semantically well-founded transformations and simplifications of the ontology and mapping. The notions of equivalence based upon particular queries can be used to check that modifications to the specification do not inadvertently impact the answers to queries that should not be concerned by the changes.

We present a first investigation of the computational complexity of deciding the above forms of entailment for a pair of OBDA specifications. We consider three different languages (linear, GAV and GLAV) for expressing mappings in OBDA. As ontology languages, we focus on lightweight description logics (DLs) from the *DL-Lite* family (Calvanese et al. 2007; Artale et al. 2009), which are a popular choice for OBDA. More precisely, we consider the logic *DL-Lite*$_R$ (also known as *DL-Lite*$_{core}^{\mathcal{H}}$), which is the basis for the OWL 2 QL profile (Motik et al. 2012), as well as its fragments *DL-Lite*$_{core}$ and *DL-Lite*$_{RDFS}$. The latter is the DL analogue of the RDFS standard (Guha and Brickley 2014).

The paper is structured as follows. After some preliminaries, we formally define in Section 3 the different notions of entailment between OBDA specifications. In Section 4, we provide characterization results that allow us to restrict the number and form of databases and ABoxes that need to be considered when deciding entailment. These results will be used in Section 5 as central tools in our analysis of the complexity of query entailment between OBDA specifications formulated in the DL and mapping languages mentioned above. To obtain our complexity results, we also exploit connections to query containment in the presence of signature restrictions (Bienvenu, Lutz, and Wolter 2012) and KB query inseparability (Botoeva et al. 2014). The results of our complexity analysis are summarized in Figure 1. As shown in the table, the complexity of query-based entailment between OBDA specifications ranges from NL (nondeterministic logarithmic space) for (single) IQ-entailment with linear mappings (and the three considered DL-Lite dialects) to EXPTIME for CQ-entailment in *DL-Lite*$_R$ (for the three types of mapping).

For lack of space, some proofs have been deferred to the appendix of the long version (Bienvenu and Rosati 2016).

2 Preliminaries

We start from four pairwise disjoint countably infinite sets of names: the set $\mathsf{N_C}$ of concept names (unary predicates), the set $\mathsf{N_R}$ of role names (binary predicates), the set $\mathsf{N_{rel}}$ of relation names (predicates of arbitrary arity), and the set $\mathsf{N_I}$ of constant names (also called individuals). We will call $\mathsf{N_C} \cup \mathsf{N_R}$ the *DL signature* and $\mathsf{N_{rel}}$ the *database signature*. Given a syntactic object O, we denote by $\mathsf{sig}(O)$ the set of concept names, role names, and relation names occurring in O, and by $\mathsf{const}(O)$ the set of constant names occurring in O. If $\vec{a}$ is a tuple of constants and $\Xi \subseteq \mathsf{N_I}$, then the notation $\vec{a} \subseteq \Xi$ means that every constant in $\vec{a}$ belongs to Ξ. As usual, we use $|O|$ for the *size of* O, i.e. the total number of occurrences of symbols in O.

We recall that a DL *knowledge base* (KB) is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$ composed of a TBox $\mathcal{T}$, expressing intensional knowledge, and an ABox $\mathcal{A}$, specifying factual information about particular individuals. Formally, the *TBox* is a finite set of axioms built from $\mathsf{N_C}$ and $\mathsf{N_R}$, whose form depends on the DL in question. The *ABox* $\mathcal{A}$ is a finite set of concept and role *assertions* (ground facts) of the forms $A(a)$ and $R(a, b)$, with $a, b \in \mathsf{N_I}$, $A \in \mathsf{N_C}$, and $R \in \mathsf{N_R}$.

As mentioned in the introduction, we focus on the description logic *DL-Lite*$_R$ and on two fragments of this logic. A *DL-Lite*$_R$ TBox consists of a finite set of concept inclusions $B \sqsubseteq C$ and role inclusions $S \sqsubseteq Q$, where B, C, S, and Q are defined according to the following syntax (where A is a concept name and R is a role name):

$$B \to A \mid \exists S \quad C \to B \mid \neg B \quad S \to R \mid R^- \quad Q \to S \mid \neg S$$

A *DL-Lite*$_{core}$ TBox only allows for concept inclusions of the form $B \sqsubseteq C$, while a *DL-Lite*$_{RDFS}$ TBox allows for concept inclusions $B \sqsubseteq A$ and role inclusions $S \sqsubseteq S'$.

The semantics of DL KBs is based upon *interpretations*, which take the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, and the function $\cdot^{\mathcal{I}}$ maps every $A \in \mathsf{N_C}$ to $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, every $R \in \mathsf{N_R}$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every $c \in \mathsf{N_I}$ to $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ is straightforwardly extended to arbitrary concepts and roles, e.g. $(\exists R)^{\mathcal{I}} = \{e_1 \mid (e_1, e_2) \in R^{\mathcal{I}}\}$ and $(R^-)^{\mathcal{I}} = \{(e_2, e_1) \mid (e_1, e_2) \in R^{\mathcal{I}}\}$. An interpretation $\mathcal{I}$ satisfies an inclusion $G \sqsubseteq H$ if $G^{\mathcal{I}} \subseteq H^{\mathcal{I}}$, and it satisfies an assertion $A(a)$ (resp. $R(a, b)$) if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$). We call $\mathcal{I}$ a model of a KB $\langle \mathcal{T}, \mathcal{A} \rangle$ if it satisfies all inclusions in $\mathcal{T}$ and assertions in $\mathcal{A}$. If $\langle \mathcal{T}, \mathcal{A} \rangle$ has no models, it is unsatisfiable, written $\langle \mathcal{T}, \mathcal{A} \rangle \models \bot$.

Every satisfiable *DL-Lite*$_R$ KB $(\mathcal{T}, \mathcal{A})$ possesses a *canonical model* $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ constructed as follows. The domain $\Delta^{\mathcal{T},\mathcal{A}}$ of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ consists of all words $aR_1 \dots R_n$ ($n \geq 0$) such that $a \in \mathsf{const}(\mathcal{A})$, $R_i \in \{R, R^- \mid R \in \mathsf{N_R}\}$, and:

- if $n \geq 1$, then $\mathcal{T}, \mathcal{A} \models \exists R_1(a)$;
- for $1 \leq i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$.

The interpretation function is defined as follows:

$$\begin{aligned}
a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= a \text{ for all } a \in \mathsf{N_I} \\
A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \{a \in \mathsf{const}(\mathcal{A}) \mid \mathcal{T}, \mathcal{A} \models A(a)\} \\
&\qquad \cup \{aR_1 \dots R_n \mid n \geq 1 \text{ and } \mathcal{T} \models \exists R_n^- \sqsubseteq A\} \\
R^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \{(a, b) \mid R(a, b) \in \mathcal{A}\} \cup \\
&\qquad \{(w_1, w_2) \mid w_2 = w_1 S \text{ and } \mathcal{T} \models S \sqsubseteq R\} \cup \\
&\qquad \{(w_2, w_1) \mid w_2 = w_1 S \text{ and } \mathcal{T} \models S^- \sqsubseteq R\}
\end{aligned}$$

Observe that the elements of $\Delta^{\mathcal{T},\mathcal{A}}$ can be naturally viewed as a set of *trees*, rooted at the constants.

We now introduce OBDA specifications, which consist of a DL ontology and a mapping, with the latter serving to specify the semantic relationship holding between elements

of the DL and database vocabularies. In this paper, we focus on the well-known class of GLAV ('global-and-local-as-view') mappings (Doan, Halevy, and Ives 2012).

Mappings are formally defined as follows. An *atom* is an expression $P(\vec{t})$ where P is a predicate and $\vec{t}$ is a tuple of *terms* (variables and constants). A *(GLAV) mapping assertion* m is an expression of the form $q_s(\vec{x}) \rightarrow q_o(\vec{x})$, where $q_s(\vec{x})$ (the *body* of m, denoted $body(m)$) is a conjunction of atoms over predicates from $\mathsf{N_{rel}}$ and constants from $\mathsf{N_I}$, $q_o(\vec{x})$ (called the *head* of m, $head(m)$) is a conjunction of atoms using predicates from $\mathsf{N_C} \cup \mathsf{N_R}$ and constants from $\mathsf{N_I}$, and $\vec{x}$, called the *frontier variables* of m, are the variables that appear both in q_o and in q_s. We use $\mathsf{fvars}(m)$ (resp. $\mathsf{vars}(m)$, $\mathsf{terms}(m)$) to refer to the frontier variables (resp. variables, terms) of m. The *arity* of m is the number of its frontier variables. When $q_o(\vec{x})$ has the form $P(\vec{x})$ (i.e., $q_o(\vec{x})$ is a single atom whose arguments are $\vec{x}$), we call m a *GAV* mapping assertion. A *linear* mapping assertion is a GAV assertion whose body consists of a single atom. A *(GLAV) mapping* $\mathcal{M}$ is a set of mapping assertions. A *GAV (resp. linear) mapping* is a mapping constituted of GAV (resp. linear) mapping assertions. Without loss of generality, we assume that in every mapping $\mathcal{M}$, every pair of distinct mapping assertions uses pairwise disjoint sets of variables.

An *OBDA specification* is a pair $\Gamma = \langle \mathcal{T}, \mathcal{M} \rangle$, where $\mathcal{T}$ is a TBox and $\mathcal{M}$ is a mapping. Given a mapping assertion m of arity n and an n-tuple of constants $\vec{a}$, we denote by $m(\vec{a})$ the assertion obtained from m by replacing the frontier variables with the constants in $\vec{a}$.

Given a set of atoms AT, the function gr returns a set $gr(AT)$ of ground atoms obtained from AT by replacing every variable symbol x with a fresh constant symbol c_x that does not occur in the considered mapping or database. We assume without loss of generality that if $AT \neq AT'$, then $gr(AT)$ and $gr(AT')$ use distinct fresh constants. Note that there is a *natural isomorphism* from AT to $gr(AT)$ which maps every constant in AT to itself and every variable x to the corresponding constant symbol c_x in $gr(AT)$.

In this paper, a *database* (instance) is a finite set of ground atoms using relation names from $\mathsf{N_{rel}}$ and constant names from $\mathsf{N_I}$. Given a mapping $\mathcal{M}$ and a database D, we define the *ABox for D and $\mathcal{M}$*, denoted as $\mathcal{A}_{\mathcal{M},D}$, as follows:

$$\{\beta \in gr(head(m(\vec{a}))) \mid m \in \mathcal{M} \text{ and } D \models \exists \vec{y}.body(m(\vec{a}))\}$$

where we assume that $\vec{y}$ are the variables occurring in $body(m(\vec{a}))$. If $\mathcal{A} \subseteq \mathcal{A}_{\mathcal{M},D}$, then we denote by $\mathsf{const}^{\exists}(\mathcal{A})$ the constants in $\mathcal{A}$ that occur neither in D nor in $\mathcal{M}$ (i.e., the fresh constants introduced when grounding mapping heads).

Remark 1. In the case of GAV mappings, the ABox $\mathcal{A}_{\mathcal{M},D}$ is guaranteed to be of polynomial size, whereas for GLAV mappings, it can be exponentially large due to presence of existential variables in mapping heads. Indeed, take $D = \{T_i(0), T_i(1) \mid 1 \leq i \leq n\}$ and suppose $\mathcal{M}$ contains

$$T_1(x_1) \wedge \ldots \wedge T_n(x_n) \rightarrow \exists z R_1(z, x_1) \wedge \ldots \wedge R_n(z, x_n)$$

For each $\tau = (v_1, \ldots, v_n) \in \{0,1\}^n$, we create a fresh constant c_τ and include $R_1(c_\tau, v_1), \ldots, R_n(c_\tau, v_n)$ in $\mathcal{A}_{\mathcal{M},D}$.

Given an OBDA specification $\Gamma = \langle \mathcal{T}, \mathcal{M} \rangle$ and a database D, we define the *models of* Γ *and* D, denoted $Mods(\Gamma, D)$, as the set of models of the KB $\langle \mathcal{T}, \mathcal{A}_{\mathcal{M},D} \rangle$. When $Mods(\Gamma, D) = \emptyset$, we write $\langle \mathcal{T}, \mathcal{M}, D \rangle \models \bot$.

We are interested in the problem of answering instance queries and conjunctive queries over a pair composed of an OBDA specification and a database. A *conjunctive query (CQ)* takes the form $q(\vec{x}) = \exists \vec{y}(\alpha_1 \wedge \ldots \wedge \alpha_n)$ where every α_i is an atom whose arguments are either constants or variables from $\vec{x} \cup \vec{y}$. The free variables $\vec{x}$ are called *answer variables*, and a CQ is called *Boolean* if it has no answer variables. For a non-Boolean CQ q with answer variables $x_1, \ldots, x_k$, a tuple of constants $\vec{a} = \langle a_1, \ldots, a_k \rangle$ occurring in $\mathcal{A}$ is said to be a *certain answer for q w.r.t. $\mathcal{K}$* just in the case that $\mathcal{K} \models q(\vec{a})$, where $q(\vec{a})$ is the Boolean query obtained from q by replacing each x_i by a_i. An *instance query (IQ)* is a CQ consisting of a single atom of the form $A(x)$ or $R(x, y)$, with A concept name, R role name, and x, y distinct answer variables. We use CQ (resp. IQ) to refer to the set of all CQs (resp. IQs) over the DL signature $\mathsf{N_C} \cup \mathsf{N_R}$.

Given an OBDA specification $\Gamma = \langle \mathcal{T}, \mathcal{M} \rangle$, a database D, and a CQ q, we define the *certain answers for q w.r.t.* (Γ, D) as the tuples of constants from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M})$ that are certain answers for q w.r.t. $\langle \mathcal{T}, \mathcal{A}_{\mathcal{M},D} \rangle$. In particular, for Boolean CQs, we say that q is entailed by (Γ, D), denoted by $(\Gamma, D) \models q$ (or $\langle \mathcal{T}, \mathcal{M}, D \rangle \models q$), if $\mathcal{I} \models q$ for every $\mathcal{I} \in Mods(\Gamma, D)$. Note that for non-Boolean queries, we only consider constants occurring either in D or in $\mathcal{M}$ and thereby exclude the fresh constants in $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M},D})$.

Importantly, canonical models characterize CQ answering over *DL-Lite$_R$* KBs. More precisely: for every CQ q and every tuple $\vec{a}$ of constants occurring in $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle \models q(\vec{a})$ iff $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q(\vec{a})$. Given an OBDA specification $\Gamma = \langle \mathcal{T}, \mathcal{M} \rangle$ and a data instance D, we use $\mathcal{I}_{\Gamma,D}$ to denote the canonical model for the KB $\langle \mathcal{T}, \mathcal{A}_{\mathcal{M},D} \rangle$. The preceding property immediately carries over to the OBDA setting:

Proposition 1. *Given a DL-Lite$_R$ OBDA specification $\Gamma = \langle \mathcal{T}, \mathcal{M} \rangle$, a database D, a CQ q and a tuple $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M})$, we have $(\Gamma, D) \models q(\vec{a})$ iff $\mathcal{I}_{\Gamma,D} \models q(\vec{a})$.*

3 Query-based Entailment for OBDA Specifications

We start by recalling the classical notion of entailment between OBDA specifications.

Definition 1 (Logical entailment)**.** An OBDA specification $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ *logically entails* $\langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, written $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathsf{log}} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ if and only the first-order theory $\mathcal{T}_1 \cup \mathcal{M}_1$ logically entails the first-order theory $\mathcal{T}_2 \cup \mathcal{M}_2$.

We next formally define the different notions of query-based entailment between OBDA specifications considered in this paper. First, we introduce a notion of entailment that compares specifications based upon the constraints they impose regarding consistency.

Definition 2 ($\bot$-entailment)**.** An OBDA specification $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ *$\bot$-entails* $\langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, written $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\bot} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, iff, for every database D,

$$\langle \mathcal{T}_2, \mathcal{M}_2, D \rangle \models \bot \quad \Rightarrow \quad \langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models \bot$$

Next, we define a notion of query entailment between OBDA specifications with respect to a *single* query.

Definition 3 (Single query entailment)**.** Let q be a query. An OBDA specification $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ *q-entails* $\langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, written $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_q \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, iff $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_\bot \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ and for every database D and tuple $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$:

$$\langle \mathcal{T}_2, \mathcal{M}_2, D \rangle \models q(\vec{a}) \quad \Rightarrow \quad \langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models q(\vec{a})$$

When q is an IQ (resp. CQ), we call the preceding entailment relation *single IQ-entailment* (resp. *single CQ-entailment*).

We can generalize the previous definition to classes of queries as follows.

Definition 4 (Query entailment)**.** Let $\mathcal{L}$ be a (possibly infinite) set of queries. An OBDA specification $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ *$\mathcal{L}$-entails* $\langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, written $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathcal{L}} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ iff $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_\bot \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ and $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_q \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ for every query $q \in \mathcal{L}$. When $\mathcal{L} = \mathsf{IQ}$, we call the preceding entailment relation *IQ-entailment*, and for $\mathcal{L} = \mathsf{CQ}$, we use the term *CQ-entailment*.

Note that each notion of entailment induces a notion of equivalence between OBDA specifications, corresponding to the case when the entailment holds in both directions (we omit the formal definitions due to space limitations).

The following property immediately follows from the preceding definitions.

Proposition 2. *Let $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle, \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ be two OBDA specifications, and let $\mathcal{L}_1$ be a set of queries. Then, $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathsf{log}} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ implies $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathcal{L}_1} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$. Moreover, if $\mathcal{L}_2 \subseteq \mathcal{L}_1$, then $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathcal{L}_1} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ implies $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathcal{L}_2} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$.*

As a consequence of Proposition 2, we have that logical entailment implies CQ-entailment, and CQ-entailment implies IQ-entailment. The converse implications do not hold, as the following examples demonstrate.

Example 1. We start by illustrating the difference between logical entailment and CQ-entailment. Consider a database containing instances for the relation *EXAM*(*studentName,courseName,grade,date*). Then, let $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$, where

$$\mathcal{T}_1 = \{\textit{Student} \sqsubseteq \textit{Person},\ \textit{PhDStudent} \sqsubseteq \textit{Student}\}$$
$$\mathcal{M}_1 = \{\textit{EXAM}(x, y, z, w) \rightarrow \textit{Student}(x)\}$$

and let $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, where $\mathcal{T}_2 = \{\textit{Student} \sqsubseteq \textit{Person}\}$ and $\mathcal{M}_2 = \mathcal{M}_1$. It is immediate to verify that $\Gamma_2 \not\models_{\mathsf{log}} \Gamma_1$. However, we have that $\Gamma_2 \models_{\mathsf{CQ}} \Gamma_1$. Indeed, $\Gamma_2 \models_{\mathsf{CQ}} \Gamma_1$ can be intuitively explained by the fact that the mapping $\mathcal{M}_1$ does not retrieve any instances of the concept *PhDStudent* (and there are no subclasses that can indirectly populate it), so the presence of the inclusion *PhDStudent* $\sqsubseteq$ *Student* in $\mathcal{T}_1$ does not have any effect on query answering; in particular, every CQ that mentions the concept *PhDStudent* cannot be entailed both under Γ_1 and under Γ_2. Notice also that, if we modify the mapping $\mathcal{M}_1$ to map *PhDStudent* instead of *Student* (i.e., if $\mathcal{M}_1$ were $\{\textit{EXAM}(x, y, z, w) \rightarrow \textit{PhDStudent}(x)\}$), then CQ-entailment between Γ_2 and Γ_1 would no longer hold.

Next, consider $\Gamma_3 = \langle \mathcal{T}_3, \mathcal{M}_3 \rangle$, where $\mathcal{T}_3 = \emptyset$ and

$$\mathcal{M}_3 = \{\textit{EXAM}(x, y, z, w) \rightarrow \textit{Student}(x),$$
$$\textit{EXAM}(x, y, z, w) \rightarrow \textit{Person}(x)\}$$

Again, it it immediate to see that $\Gamma_3 \not\models_{\mathsf{log}} \Gamma_2$, while we have that $\Gamma_3 \models_{\mathsf{CQ}} \Gamma_2$. Indeed, $\Gamma_3 \models_{\mathsf{CQ}} \Gamma_2$ follows informally from the fact that the mapping $\mathcal{M}_3$ is able to "extensionally" simulate the inclusion *Student* $\sqsubseteq$ *Person* of $\mathcal{T}_2$, which is sufficient for Γ_3 to entail every CQ in the same way as Γ_2.

Example 2. We slightly modify Example 1 to show the difference between CQ-entailment and IQ-entailment. Consider $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M} \rangle$ and $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M} \rangle$ where

$$\mathcal{T}_1 = \{\textit{Student} \sqsubseteq \textit{Person},\ \textit{Student} \sqsubseteq \exists \textit{TakesCourse}\}$$
$$\mathcal{T}_2 = \{\textit{Student} \sqsubseteq \textit{Person}\}$$
$$\mathcal{M} = \{\textit{EXAM}(x, y, z, w) \rightarrow \textit{Student}(x)\}$$

It is easily verified that $\Gamma_2 \not\models_{\mathsf{CQ}} \Gamma_1$. Indeed, take the Boolean CQ $\exists x, y\, \textit{TakesCourse}(x, y)$: for every database D, this query is not entailed by the pair (Γ_2, D), while this is not the case when the specification is Γ_1. On the other hand, we have that $\Gamma_2 \models_{\mathsf{IQ}} \Gamma_1$: in particular, for every database D and for every pair of individuals a, b, neither (Γ_1, D) nor (Γ_2, D) entails the IQ $\textit{TakesCourse}(a, b)$. Finally, let q be the non-Boolean CQ $\exists x\, \textit{TakesCourse}(x, y)$: then, it can be easily verified that the single CQ-entailment $\Gamma_2 \models_q \Gamma_1$ holds; while for the CQ q' of the form $\exists y\, \textit{TakesCourse}(x, y)$, the single CQ-entailment $\Gamma_2 \models_{q'} \Gamma_1$ does not hold.

Some of our proofs will exploit results about query entailment for DL knowledge bases, a problem which was first investigated in (Botoeva et al. 2014) and further studied in (Botoeva et al. 2015). The latter work introduces a generalized notion of entailment parameterized both by a query signature and a set of constants, which is defined as follows:

Definition 5. Let Σ be a DL signature and Υ a set of constants. A DL KB $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ *(Σ-Υ)-entails* a DL KB $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle$ iff $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle \models q(\vec{a})$ for every CQ q with $\mathsf{sig}(q) \subseteq \Sigma$ and every tuple $\vec{a} \subseteq \Upsilon \cap \mathsf{const}(\mathcal{A}_2)$.

The relationship between the two notions of query entailment is witnessed by the following property:

Proposition 3. *Let $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle, \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ be two OBDA specifications. Then, $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ iff, for every database D, the KB $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1, D} \rangle$ (Σ-Υ)-entails the KB $\mathcal{K}_2 = \langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2, D} \rangle$, where $\Sigma = \mathsf{sig}(\mathcal{K}_2)$ and $\Upsilon = \mathsf{const}(\mathcal{A}_{\mathcal{M}_2, D}) \setminus \mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_2, D})$.*

4 Characterization Results

A key difficulty in developing procedures for checking the various notions of entailment from the previous section is that the definitions quantify over *all possible databases* D. In this section, we show that when ontologies are formulated in *DL-Lite*$_R$, it is sufficient to consider a finite set of databases. The exact number and shape of these databases will depend on the notion of entailment considered.

To simplify the exposition, we will assume throughout this section that $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ and $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ are OBDA specifications such that $\mathcal{T}_1, \mathcal{T}_2$ are formulated in *DL-Lite*$_R$, and $\mathcal{M}_1, \mathcal{M}_2$ are GLAV mappings.

We first consider $\bot$-entailment and show that it can be characterized in terms of a small number of small databases.

Theorem 1. *$\Gamma_1 \models_\bot \Gamma_2$ if and only if $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models \bot$ for every database D obtained by:*

(i) taking two mapping assertions m_1, m_2 from $\mathcal{M}_2$,
(ii) selecting atoms $\alpha_1 \in head(m_1)$ and $\alpha_2 \in head(m_2)$,
(iii) identifying in m_1 and m_2 some variables from α_1 and α_2 in such a way that $\langle \mathcal{T}, gr(\{\alpha_1, \alpha_2\})\rangle \models \bot$, and
(iv) setting D equal to $gr(body(m_1) \cup body(m_2))$.

Proof. The one direction is immediate from the definitions. For the other direction, let us suppose that $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models \bot$ for every database D obtained as in the theorem statement. Let us further suppose that $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models \bot$, where D_0 may be any database. We thus have $\langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2,D_0}\rangle \models \bot$. It is well known that every minimal unsatisfiable subset of a *DL-Lite*$_R$ KB contains at most two ABox assertions, so there must exist a subset $\mathcal{A}' \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ with $|\mathcal{A}'| \leq 2$ such that $\langle \mathcal{T}_2, \mathcal{A}'\rangle \models \bot$. Let γ be the conjunction of atoms obtained by taking for each ABox assertion in $\mathcal{A}'$, a mapping assertion that produced it, identifying those variables (and only those variables) needed to produce the ABox assertion(s), and then taking the conjunction of the atoms in the bodies. We observe that by construction $D_\gamma = gr(\gamma)$ satisfies the conditions of the theorem and is such that $\langle \mathcal{T}_1, \mathcal{M}_1, D_\gamma\rangle \models \bot$. By construction, there is a homomorphism of D_γ into the original database D_0. It follows that $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models \bot$. □

Next we consider entailment with respect to a specific CQ and show that it suffices to consider a finite (but exponential) number of small databases. We further show that, instead of considering all query answers obtainable from $\mathcal{A}_{\mathcal{M}_2,D}$, we can focus on a polynomial subset $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$.

Theorem 2. *Let q be a CQ. Then $\Gamma_1 \models_q \Gamma_2$ iff $\Gamma_1 \models_\bot \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every database D, every tuple $\vec{a}$ from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$, and every ABox $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ obtained as follows:*
(i) take $k \leq |q|$ mapping assertions $m_1, \ldots, m_k \in \mathcal{M}_2$,
(ii) identify some of the variables in $\cup_{i=1}^k \mathsf{fvars}(m_i)$, and
(iii) let $D = gr(\cup_{i=1}^k body(m_i))$ with natural isomorphism f, and[1] $\mathcal{A}_2 = \cup_{i=1}^k gr(f(head(m_i)))$.

Proof. Again the one direction is immediate. To show the non-trivial direction, let us suppose that $\Gamma_1 \models_\bot \Gamma_2$ and that $\langle \mathcal{T}_2, \mathcal{M}_2, D\rangle \models q(\vec{c})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{c})$ for every tuple $\vec{c}$ and database D satisfying the conditions of the theorem. Further suppose that $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models q_0(\vec{a})$. We focus on the interesting case in which $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \not\models \bot$ and $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \not\models \bot$. Since $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models q_0(\vec{a})$, we have $\langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2,D_0}\rangle \models q_0(\vec{a})$. It is a well-known property of *DL-Lite*$_R$ that there exists a subset $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D_0}$ with $|\mathcal{A}_2| \leq |q_0|$ such that $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q_0(\vec{a})$. Let $\sim$ be the least equivalence relation over the assertions in $\mathcal{A}_2$ such that $\beta \sim \gamma$ for every pair of assertions $\beta, \gamma \in \mathcal{A}_2$ that share a constant from $\mathsf{const}^\exists(\mathcal{A}_2)$. Next let $\mathcal{B}_1, \ldots, \mathcal{B}_k$ be the ABoxes corresponding to the equivalence classes of $\sim$. By construction, $\mathcal{A}_2$ is the disjoint union of $\mathcal{B}_1, \ldots, \mathcal{B}_k$; in particular, we have that $k \leq |\mathcal{A}_2|$ (hence, $k \leq |q_0|$). Moreover, it follows from the definition of $\sim$ and the construction of $\mathcal{A}_{\mathcal{M}_2,D_0}$ that for every $\mathcal{B}_i$, there exists a mapping assertion $m_i \in \mathcal{M}_2$ and a homomorphism h_i of $body(m_i)$ into D_0 such that $\mathcal{B}_i \subseteq gr(h_i(head(m_i)))$. Now let $FV = \cup_{i=1}^k \mathsf{fvars}(m_i)$, and let $\equiv$ be the smallest equivalence relation on FV that satisfies the following property: $u \equiv v$ if u and v appear in m_i and m_j respectively, and $h_i(u) = h_j(v)$. Select one variable from each equivalence class, and let $m'_1, \ldots, m'_k$ be obtained from $m_1, \ldots, m_k$ by replacing each frontier variable by the representative of its equivalence class (i.e., we merge those frontier variables that are mapped to the same constants). Define a function h from $\cup_{i=1}^k \mathsf{terms}(body(m'_i))$ to $\mathsf{const}(D_0)$ by mapping constants to themselves, every $v \in \mathsf{vars}(m'_i) \setminus \mathsf{fvars}(m'_i)$ to $h_i(v)$, and every $v \in \mathsf{fvars}(m'_i)$ to $h_j(v)$, where $v \in \mathsf{fvars}(m_j)$ (note that at least one such j must exist, and if v occurs in more than one m_j, the result will be the same no matter which we choose). Observe that for every $1 \leq i \leq k$, h is a homomorphism from $body(m'_i)$ to D_0 that is injective on $\mathsf{fvars}(m'_i)$, and such that $h(head(m'_i)) = h_i(head(m_i))$.

We set $D' = gr(body(m'_1) \cup \ldots \cup body(m'_k))$. We then let f be the natural isomorphism from $\cup_{i=1}^k body(m'_i)$ to D', and set $\mathcal{A}'_2 = \cup_{i=1}^k gr(f(head(m'_i))$. Clearly, D' and $\mathcal{A}'_2$ satisfy the conditions of the theorem statement. We can define an isomorphism g from $\cup_{i=1}^k gr(h(head(m'_i)))$ to $\mathcal{A}'_2 = \cup_{i=1}^k gr(f(head(m'_i))$ as follows:

- $g(c) = c$ for every constant c in $head(m'_i)$
- $g(d_v) = d'_v$ for every $v \in \mathsf{vars}(head(m'_i)) \setminus \mathsf{fvars}(m'_i)$ which is grounded to the constant d_v (resp. d'_v) in $\cup_{i=1}^k gr(h_i(head(m'_i)))$ (resp. $\mathcal{A}'_2$)
- $g(h(v)) = f(v)$ for every $v \in \mathsf{fvars}(m'_i)$

As $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q_0(\vec{a})$, $\mathcal{A}_2 \subseteq \cup_{i=1}^k gr(h_i(head(m_i)))$, and $h(head(m'_i)) = h_i(head(m_i))$, it follows that $\langle \mathcal{T}_2, \mathcal{A}'_2\rangle \models q_0(g(\vec{a}))$. Applying our assumption, we obtain $\langle \mathcal{T}_1, \mathcal{M}_1, D'\rangle \models q_0(g(\vec{a}))$. We next remark that the function which maps $d \in \mathsf{const}(D')$ to $h(f^{-1}(d))$ defines a homomorphism from D' to D_0. Moreover, if we take the tuple $g(\vec{a})$ and replace every constant d in $g(\vec{a}) \cap \mathsf{const}(D')$ by $h(f^{-1}(d))$, we get the tuple $\vec{a}$. We thus obtain $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_0(\vec{a})$. □

If we replace CQs by IQs, the preceding theorem can be refined as follows:

Theorem 3. *Let q be an IQ. Then $\Gamma_1 \models_q \Gamma_2$ iff $\Gamma_1 \models_\bot \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every database D, every tuple $\vec{a}$ from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$, and every ABox $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ obtained as follows:*
(i) taking a mapping assertion m from $\mathcal{M}_2$ and choosing an atom $\alpha \in head(m)$,
(ii) possibly identifying in m the (at most two) frontier variables appearing in α, and
(iii) letting $D = gr(body(m))$ with natural isomorphism f, and $\mathcal{A}_2 = gr(f(head(m)))$.

Proof. We simply note that in the proof of Theorem 2, if we restrict our attention to instance queries, then we have $k = 1$, and it is only necessary to identify those variables in the head atom of the mapping that leads to introducing the single ABox assertion of interest. □

[1] By $f(head(m))$ we mean the result of replacing $v \in \mathsf{fvars}(m)$ in $head(m)$ by $f(v)$ (and leaving all other terms untouched).

We now consider entailment with respect to query classes. For CQ-entailment, the characterization is actually a bit simpler than for single-CQ entailment, since we only need to consider the database-ABox pairs induced from single mapping assertions (rather than sets of mapping assertions).

Theorem 4. *$\Gamma_1 \models_{\mathsf{CQ}} \Gamma_2$ iff $\Gamma_1 \models_\bot \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every CQ q, every database D, every tuple $\vec{a}$ from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$, and every ABox $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ obtained as follows:*

(i) take a mapping assertion m from $\mathcal{M}_2$,
(ii) possibly identify some of the variables in $\mathsf{fvars}(m)$, and
(iii) let $D = gr(body(m))$ with natural isomorphism f, and $\mathcal{A}_2 = gr(f(head(m)))$.

Proof. Suppose that $\Gamma_1 \models_\bot \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every CQ q and every database D and ABox $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ obtained as in the theorem statement. Further suppose that $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models q_0(\vec{a})$. We focus on the interesting case in which both $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \not\models \bot$ and $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \not\models \bot$. We let $\mathcal{A}_{\mathcal{M}_1,D_0}$ and $\mathcal{A}_{\mathcal{M}_2,D_0}$ be the corresponding ABoxes, and let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the canonical models of $\langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1,D_0}\rangle$ and $\langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2,D_0}\rangle$ respectively. Since $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models q_0(\vec{a})$, we have $\mathcal{I}_2 \models q_0(\vec{a})$ by Proposition 1, and thus there must exist a match π for $q_0(\vec{a})$ in $\mathcal{I}_2$ (i.e., a homomorphism of q_0 into $\mathcal{I}_2$ that sends the answer variables $\vec{x}$ of q_0 to $\vec{a}$). This match gives rise to an equivalence relation $\sim_\pi$ on the atoms in q, defined as follows: $\alpha \sim_\pi \alpha'$ just in the case that α and α' share a variable v such that $\pi(v) \in (\Delta^{\mathcal{I}_2} \setminus \mathsf{const}(\mathcal{A}_{\mathcal{M}_2,D_0})) \cup \mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_2,D_0})$ that is, $\pi(v)$ is not a constant from D_0 nor $\mathcal{M}_2$. This equivalence relation in turn induces a partition of q_0 into connected subqueries $q_1, \ldots, q_n$, where every q_i corresponds to one of the equivalence classes of $\sim_\pi$. By construction, each of the queries q_i satisfies exactly one of the following conditions:

- q_i consists of a single atom all of whose variables are mapped by π to constants from $\mathsf{const}(D_0) \cup \mathsf{const}(\mathcal{M}_2)$;
- every atom in q_i contains a variable that is not mapped to a constant occurring in D_0 or $\mathcal{M}_2$, and at least one term in q_i is mapped to a constant in $\mathcal{A}_{\mathcal{M}_2,D}$;
- q_i contains no constants and none of its variables are mapped to constants in $\mathcal{A}_{\mathcal{M}_2,D}$.

We associate a tuple $\vec{x}_i$ of answer variables with each q_i. If q_i satisfies the first or second condition, then every variable in q_i that is mapped to a constant *from* $\mathsf{const}(D_0) \cup \mathsf{const}(\mathcal{M}_2)$ is designated as an answer variable. If the third condition holds, then q_i has no answer variables. By construction, we have $\mathcal{I}_2 \models q_i(\vec{b}_i)$, where $\vec{b}_i = \pi(\vec{x}_i)$.

Our aim is to show that $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_0(\vec{a})$. Because of the way we defined the queries $q_i(\vec{x}_i)$ and tuples $\vec{b}_i$, it is sufficient to show that $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_i(\vec{b}_i)$, for every $1 \leq i \leq n$. Indeed, suppose this is the case, and let π'_i be a match of $q_i(\vec{b}_i)$ in the canonical model $\mathcal{I}_1$ of $\langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1,D_0}\rangle$. If $v \in \vec{x}_i \cap \vec{x}_j$, then we must have $\pi'_i(v) = \pi'_j(v) = \pi(v)$. It follows that we can define a match π' for q_0 in $\mathcal{I}_0$ by setting $\pi'(v) = \pi'_i(v)$, where i is chosen such that $v \in \vec{x}_i$.

We briefly sketch the argument that $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_i(\vec{b}_i)$ (details are in the appendix). The key step is to show, using properties of canonical models, that there exists a mapping assertion $m_i \in \mathcal{M}_2$ and a homomorphism h_i of $body(m_i)$ into D_0 such that $\langle \mathcal{T}_2, gr(h_i(head(m_i)))\rangle \models q_i(\vec{b}_i)$. We let m'_i be obtained from m_i by identifying frontier variables y, z whenever $h_i(y) = h_i(z)$. We apply the assumption to the pair consisting of the database $D_{m'_i} = gr(body(m'_i))$ and the ABox $gr(f_i(head(m'_i)))$ (with f_i the natural isomorphism) to obtain $\langle \mathcal{T}_1, \mathcal{M}_1, D_{m'_i}\rangle \models q_i(\vec{b}'_i)$, with $\vec{b}'_i$ the tuple of constants from $D_{m'_i}$ corresponding to $\vec{b}_i$. From this, we can derive $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_i(\vec{b}_i)$. □

For IQ entailment, it suffices to consider the same databases and ABoxes as for the single IQ case.

Theorem 5. *$\Gamma_1 \models_{\mathsf{IQ}} \Gamma_2$ if and only if $\Gamma_1 \models_\bot \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every IQ q and every database-ABox pair $(D, \mathcal{A}_2)$ and tuple of constants $\vec{a}$ that satisfies the conditions of Theorem 3.*

Finally, we close this section with the following result, which shows that if CQ entailment holds, then this is witnessed by a polynomial-size subset of the ABox $\mathcal{A}_{\mathcal{M}_1,D}$.

Theorem 6. *Let $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$, and suppose that $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ with $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. Then there exists a subset $\mathcal{A}_1 \subseteq \mathcal{A}_{\mathcal{M}_1,D}$ with $|\mathcal{A}_1| \leq 2|\mathcal{T}_2| \cdot ((2|\mathcal{T}_2|+1) \cdot |\mathcal{M}_1| \cdot (|\mathcal{A}_2| + 2|\mathcal{T}_2| \cdot (|D| + |\mathcal{M}_1|)))^2$, such that $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{A}_1\rangle \models q(\vec{a})$ for every $q(\vec{a})$ as above.*

Proof. Let $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ be such that $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ such that $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. We consider the interesting case in which $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \not\models \bot$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \not\models \bot$. We let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the canonical models of $\langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1,D}\rangle$ and $\langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2,D}\rangle$ respectively. By assumption, $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ with $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. It follows from results in (Botoeva et al. 2015) that the latter property is true iff there is a homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$ that is the identity on the constants in D and $\mathcal{M}_2$, or more precisely, a function $h : \Delta^{\mathcal{I}_2} \to \Delta^{\mathcal{I}_1}$ such that:

- $h(e) = e$ for every $e \in \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$
- $e \in A^{\mathcal{I}_2}$ implies $h(e) \in A^{\mathcal{I}_1}$
- $(e, e') \in R^{\mathcal{I}_2}$ implies $(h(e), h(e')) \in R^{\mathcal{I}_1}$

Note that the constants in $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_2,D})$ need not be mapped to themselves.

Our objective is to show how to use h to construct a homomorphism g from $\mathcal{I}_2$ to $\mathcal{I}_1$ with the property that the set $\Theta = \{a \mid aw \in \mathsf{range}(g)\}$ has size polynomially bounded in the size of D, $\mathcal{A}_2$, $\mathcal{T}_2$, and $\mathcal{M}_1$. This will allow us to identify a small subset of $\mathcal{A}_{\mathcal{M}_1,D}$ that is sufficient to infer all of the queries involving constants from D or $\mathcal{M}_2$ that are entailed by $\langle \mathcal{T}_2, \mathcal{A}_2\rangle$. To construct the desired homomorphism g, we will define a sequence $g_0, g_1, g_2 \ldots$ of

entailment	mapping	*DL-Lite*$_{RDFS}$	*DL-Lite*$_{core}$	*DL-Lite*$_R$
logical	linear	NL	NL	NL
	G(L)AV	NP	NP	NP
$\bot$	linear	trivial	NL	NL
	G(L)AV	trivial	NP	NP
(single) IQ	linear	NL	NL	NL
	G(L)AV	NP	NP	NP
CQ	linear	NL	NL-hard, in PTIME	EXPTIME
	GAV	NP	NP	EXPTIME
	GLAV	NP	NP-hard, in Π_2^p	EXPTIME
single CQ	linear	Π_2^p	Π_2^p	Π_2^p
	G(L)AV	Π_2^p	Π_2^p	Π_2^p

Figure 1: Complexity of query-based entailment between OBDA specifications. All results are completeness results, unless otherwise indicated.

partial homomorphisms from $\mathcal{I}_2$ to $\mathcal{I}_1$ such that g_{i+1} extends g_i for every $i \geq 0$ (that is, $\mathsf{dom}(g_i) \subseteq \mathsf{dom}(g_{i+1})$ and $g_{i+1}(e) = g_i(e)$ for all $e \in \mathsf{dom}(g_i)$). We will ensure that for every $e \in \Delta^{\mathcal{I}_2}$, there is some i such that $e \in \mathsf{dom}(g_i)$, which allows us to take g to be the limit of this sequence.

For the initial partial homomorphism g_0, we include in the domain $\mathsf{dom}(g_0)$: *(i)* all constants in $\mathsf{const}(\mathcal{A}_2)$, and *(ii)* all elements awR such that $h(e) \notin \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$ for every e that is a prefix of awR, and set $g_0(e) = h(e)$ for every $e \in \mathsf{dom}(g_0)$. Intuitively, g_0 copies the value of h for all constants in $\mathcal{A}_2$ as well as for all elements e such that every 'ancestor' of e is mapped to an element of $\Delta^{\mathcal{I}_1} \setminus (\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1))$.

At each stage $i \geq 1$, we will extend g_{i-1} to some additional elements from $\Delta^{\mathcal{I}_2} \setminus \mathsf{const}(\mathcal{A}_2)$ in such a way that the cardinality of the set $\{a \mid aw \in \mathsf{range}(g_i)\}$ does not exceed the required polynomial bound. The construction is lengthy and rather involved (see the appendix for details), but the basic intuition is as follows: we exploit the regularity of canonical models in order to identify subtrees of $\mathcal{I}_2$ that are similar in structure, and we modify the homomorphism h so similar subtrees are mapped in the same way into $\mathcal{I}_1$. □

5 Complexity Results

In this section, we investigate the computational properties of the different notions of entailment between OBDA specifications defined in the previous section.

The results of our complexity analysis are displayed in Figure 1. In what follows, we formally state the different complexity results and provide some ideas about the proofs.

We begin by considering the complexity of deciding classical entailment between OBDA specifications.

Theorem 7. *For DL-Lite*$_{RDFS}$*, DL-Lite*$_{core}$ *and DL-Lite*$_R$*, logical entailment between OBDA specifications is* NP*-complete for GAV and GLAV mappings, and NL-complete for linear mappings.*

Proof. Let $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$, $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$. First, it is easy to see that $\Gamma_1 \models_{\mathsf{log}} \Gamma_2$ iff (i) $\mathcal{T}_1 \models \mathcal{T}_2$; and (ii) $\Gamma_1 \models_{\mathsf{log}} \mathcal{M}_2$. In the case of a *DL-Lite*$_R$ TBox, property (i) can be decided in NL (Artale et al. 2009). Property (ii) can be decided by an algorithm that, for every assertion $m \in \mathcal{M}_2$, first builds a database D corresponding to $gr(body(m))$, and then checks whether $\langle \Gamma_1, D \rangle$ entails the CQ corresponding to the head of m whose frontier variables have been replaced by the corresponding constants. This algorithm runs in NP in the case of GAV and GLAV mappings. and it can be made to run in NL in the case of linear mappings (see the appendix for details), which implies the overall upper bounds in the theorem statement. For both *DL-Lite*$_{core}$ and *DL-Lite*$_{RDFS}$ TBoxes, the lower bound for GAV mappings can be obtained through an easy reduction of CQ containment to logical entailment, while the lower bounds for linear mappings follow from a reduction of the entailment of a concept inclusion axiom in a TBox that is both in *DL-Lite*$_{core}$ and in *DL-Lite*$_{RDFS}$. □

Using Theorem 1, we can pinpoint the complexity of $\bot$-entailment. We remark that $\bot$-entailment is trivial in *DL-Lite*$_{RDFS}$, since every KB in this logic is satisfiable.

Theorem 8. *For both DL-Lite*$_R$ *and DL-Lite*$_{core}$ *TBoxes, the $\bot$-entailment problem between OBDA specifications is NP-complete in the case of GAV / GLAV mappings, and NL-complete in the case of linear mappings.*

Proof. We know from Theorem 1 that $\Gamma_1 \models_\bot \Gamma_2$ iff $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models \bot$ for every database D satisfying the conditions of the theorem statement. For the GAV / GLAV case, we compute these databases in polynomial time and for every such database D, we guess a polynomial-size proof that $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models \bot$. For the linear case, we observe that the databases satisfying the conditions of Theorem 1 correspond to taking the conjunction of two single-atom mapping bodies and possibly performing a bounded number of variable unifications, and (representations of) such databases can be enumerated in logarithmic space. For every such database D, we can check using an NL oracle whether $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models \bot$. Since $\mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$, we obtain an NL procedure. □

We next analyze the complexity of single CQ-entailment, showing it to be Π_2^p-complete for all of the DLs and mapping languages considered in this paper.

Theorem 9. *The single CQ-entailment problem is Π_2^p-complete for OBDA specifications based upon DL-Lite*$_R$ *TBoxes and GLAV mappings. The lower bound holds even for linear mappings and when both TBoxes are empty,*

Proof. For the upper bound, consider two OBDA specifications $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ and $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$. From Theorems 1 and 2, we know that $\Gamma_1 \not\models_q \Gamma_2$ if and only if one of the following holds:

- there is a database D satisfying the conditions of Theorem 1 such that $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \not\models \bot$;
- there is a pair $(D, \mathcal{A}_2)$ satisfying the conditions of Theorem 2 such that $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$, $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \not\models \bot$, and $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \not\models q(\vec{a})$.

The first item can be checked using an NP oracle (by Theorem 8). To check the second item, we remark that for every pair $(D, \mathcal{A}_2)$ satisfying the conditions of Theorem 2, the number of facts in D (resp. $\mathcal{A}_2$) cannot exceed $|q| \cdot \mathsf{maxbody}$ (resp. $|q| \cdot \mathsf{maxhead}$) where $\mathsf{maxbody}$ (resp. $\mathsf{maxhead}$) is the maximum number of atoms appearing in the body (resp.

head) of a mapping assertion in $\mathcal{M}_2$. It follows that to show that the second item above is violated, we can guess a pair $(D, \mathcal{A}_2)$ respecting these size bounds, together with a tuple of constants $\vec{a}$ and a polynomial-size proof that $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$, and then we can verify using an NP oracle that $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \not\models q(\vec{a})$. We therefore obtain a Σ_2^p procedure for deciding the complement of our problem.

For the lower bound, we utilize a result from (Bienvenu, Lutz, and Wolter 2012) on query containment over signature-restricted ABoxes. In that paper, it is shown how, given a 2QBF $\forall\vec{u}\exists\vec{v}\varphi(\vec{u},\vec{v})$, one can construct a TBox $\mathcal{T}$, Boolean CQs q_1 and q_2, and a signature Σ such that $\forall\vec{u}\exists\vec{v}\varphi(\vec{u},\vec{v})$ is valid iff $\mathcal{T}, \mathcal{A} \models q_1 \Rightarrow \mathcal{T}, \mathcal{A} \models q_2$ for all Σ-ABoxes $\mathcal{A}$. We will not detail the construction but simply remark that the same TBox $\mathcal{T} = \{T \sqsubseteq V, F \sqsubseteq V\}$ is used for all QBFs, the signature Σ is given by $(\mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q_1) \cup \mathsf{sig}(q_2)) \setminus \{V\}$, and the query q_2 is such that $V \notin \mathsf{sig}(q_2)$.

In what follows, we will show how given $\mathcal{T}$, q_1, q_2, and Σ as above, we can reduce the problem of testing whether $\mathcal{T}, \mathcal{A} \models q_1$ implies $\mathcal{T}, \mathcal{A} \models q_2$ for all Σ-ABoxes to the problem of single CQ entailment. We will use Σ for our database instances, and we create two copies $\Sigma_1 = \{P^1 \mid P \in \Sigma\}$ and $\Sigma_2 = \{P^2 \mid P \in \Sigma\}$ of the signature Σ to be used in the head of mapping assertions. Next, we define sets of mapping assertions $\mathsf{copy}^1(\Sigma)$ and $\mathsf{copy}^2(\Sigma)$ that simply copy all of the predicates in Σ into the corresponding symbol in Σ_1 (resp. Σ_2). Formally, for $j \in \{1, 2\}$,

$$\begin{aligned}\mathsf{copy}^j(\Sigma) = &\{A(x) \rightarrow A^j(x) \mid A \in \Sigma \cap \mathsf{N_C}\} \cup \\ &\{R(x,y) \rightarrow R^j(x,y) \mid R \in \Sigma \cap \mathsf{N_R}\}\end{aligned}$$

We further define, given a data signature Λ_1 and DL signature Λ_2, a set $\mathsf{pop}(\Lambda_1, \Lambda_2)$ of mapping assertions that populates the relations in Λ_2 using all possible combinations of the constants appearing in tuples over Λ_1:

$$\begin{aligned}\mathsf{pop}(\Lambda_1,\Lambda_2) = &\{P(\vec{x}) \rightarrow A(x) \mid P \in \Lambda_1, A \in \Lambda_2 \cap \mathsf{N_C}, x \in \vec{x}\} \\ &\cup \{P(\vec{x}) \rightarrow R(x,x) \mid P \in \Lambda_1, \\ &\quad R \in \Lambda_2 \cap \mathsf{N_R}, x \in \vec{x}\}\end{aligned}$$

Using $\mathsf{copy}^1(\Sigma)$, $\mathsf{copy}^2(\Sigma)$, $\mathsf{pop}(\Sigma, \Sigma^1)$, and $\mathsf{pop}(\Sigma, \Sigma^2)$, we construct the following mappings:

$$\begin{aligned}\mathcal{M}_1 = &\mathsf{pop}(\Sigma, \Sigma^1 \cup \{V\}) \cup \mathsf{copy}^2(\Sigma) \\ \mathcal{M}_2 = &\mathsf{copy}^1(\Sigma) \cup \mathsf{pop}(\Sigma, \Sigma^2) \\ &\cup \{T(x) \rightarrow V(x), F(x) \rightarrow V(x)\}\end{aligned}$$

Observe that both mappings are linear. For the query, we let q_1' (resp. q_2') be obtained from q_1 (resp. q_2) by replacing every predicate P by P^1(resp. P^2). We also rename variables so that q_1' and q_2' do not share any variables. We then let q be the CQ obtained by taking the conjunction of q_1' and q_2' and existentially quantifying all variables. In the appendix, we show that $\langle \emptyset, \mathcal{M}_1 \rangle \models_q \langle \emptyset, \mathcal{M}_2 \rangle$ iff $\mathcal{T}, \mathcal{A} \models q_1 \Rightarrow \mathcal{T}, \mathcal{A} \models q_2$ for all Σ-ABoxes. By combining this with the reduction from (Bienvenu, Lutz, and Wolter 2012), we obtain a reduction from universal 2QBF to the single CQ-entailment problem, establishing Π_2^p-hardness of the latter. □

If we consider IQs instead, the complexity drops to either NP- or NL-complete.

Theorem 10. *For DL-Lite$_{RDFS}$, DL-Lite$_{core}$ and DL-Lite$_R$, the single IQ-entailment problem between OBDA specifications is NP-complete for both GAV and GLAV mappings, and is NL-complete for linear mappings.*

Proof. We give the arguments for GAV and GLAV mappings (for the linear case, see the appendix). For the NP upper bound, consider specifications $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ and $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$, and let q be an IQ. By Theorem 3, $\Gamma_1 \models_q \Gamma_2$ iff $\Gamma_1 \models_\perp \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models q(\vec{a})$ for every database-ABox pair $(D, \mathcal{A}_2)$ obtained in the manner described by the theorem statement.

We already know that it is in NP to test whether $\Gamma_1 \models_\perp \Gamma_2$. For the second property, observe that there are only polynomially many pairs $(D, \mathcal{A}_2)$ satisfying the required conditions, since each corresponds to choosing a mapping assertion m in $\mathcal{M}_2$, an atom $\alpha \in \mathit{head}(m)$, and deciding whether or not to identify the (at most two) variables in α. For every such pair $(D, \mathcal{A}_2)$, we compute in polynomial time the set of Boolean IQs β obtained by instantiating the IQ q with constants from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$ for which $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models \beta$. For every such β, we guess a polynomial-size proof that $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models \beta$. If all of our polynomially many guesses succeed, the procedure returns yes; else it returns no. By grouping the guesses together, we obtain an NP decision procedure.

The NP lower bound is by reduction from the NP-complete CQ containment problem: given two CQs q_1, q_2 both having a single answer variable x, we have that q_1 is contained in q_2 just in the case that $\langle \emptyset, \{q_2 \rightarrow A(x)\} \rangle \models_{A(x)} \langle \emptyset, \{q_1 \rightarrow A(x)\} \rangle$, where $A \notin \mathsf{sig}(q_1) \cup \mathsf{sig}(q_2)$. □

Finally, we consider entailment with respect to entire classes of queries. We show that testing CQ-entailment is significantly more difficult than for single CQs. Both the upper and lower bounds use recent results on KB query inseparability (Botoeva et al. 2014; 2015).

Theorem 11. *For DL-Lite$_R$, the CQ-entailment problem between OBDA specifications is EXPTIME-complete for linear, GAV and GLAV mappings.*

Proof. We start with the proof of the membership results. Consider OBDA specifications $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ and $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$. By Theorem 4, $\Gamma_1 \models_{\mathsf{CQ}} \Gamma_2$ if and only if $\Gamma_1 \models_\perp \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models q(\vec{a})$ for every query $q(\vec{a})$ and database-ABox pair $(D, \mathcal{A}_2)$ satisfying the conditions of the theorem statement. We know that testing $\Gamma_1 \models_\perp \Gamma_2$ can be done in NP (Theorem 8). To decide whether the second property holds, we consider each of the single exponentially many database-ABox pairs $(D, \mathcal{A}_2)$, and for every such pair, we set $N_{D,\mathcal{A}_2} = 2|\mathcal{T}_2| \cdot ((2|\mathcal{T}_2| + 1) \cdot |\mathcal{M}_1| \cdot (|\mathcal{A}_2| + 2|\mathcal{T}_2| \cdot (|D| + |\mathcal{M}_1|)))^2$. By Theorem 6 and Proposition 3, we know that $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models q(\vec{a})$ for every CQ $q(\vec{a})$ with $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$ iff there is a subset $\mathcal{A}_1 \subseteq \mathcal{A}_{\mathcal{M}_1, D}$ with $|\mathcal{A}_1| \leq N_{D,\mathcal{A}_2}$ such that the KB $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ (Σ, Υ)-entails the KB $\mathcal{K}_2 = \langle \mathcal{T}_2, \mathcal{A}_2 \rangle$, where $\Sigma = \mathsf{sig}(\mathcal{K}_2)$ and $\Upsilon = \mathsf{const}(\mathcal{A}_2) \setminus \mathsf{const}^\exists(\mathcal{A}_2)$. The latter

check can be performed in EXPTIME, as proven in (Botoeva et al. 2015). We thus obtain an EXPTIME procedure by iterating over all exponentially many pairs $(D, \mathcal{A}_2)$ and ABoxes $\mathcal{A}_1 \subseteq \mathcal{A}_{\mathcal{M}_1,D}$ with $|\mathcal{A}_1| \leq N_{D,\mathcal{A}_2}$, performing the KB query entailment check for every such combination, and outputting yes if for every pair $(D, \mathcal{A}_2)$, there is some ABox $\mathcal{A}_1$ for which the entailment check succeeds.

Our lower bound also exploits recent results on query inseparability of *DL-Lite*$_R$ KBs. In (Botoeva et al. 2014), the following problem is shown to be EXPTIME-complete: given *DL-Lite*$_R$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ that are consistent with the ABox $\{A(c)\}$, decide whether the certain answers for q w.r.t. $\langle \mathcal{T}_2, \{A(c)\}\rangle$ are contained in those for $\langle \mathcal{T}_1, \{A(c)\}\rangle$ for every CQ q with $\mathsf{sig}(q) \subseteq \mathsf{sig}(\mathcal{T}_2)$. To reduce this problem to the CQ-entailment problem for OBDA specifications, we take the following linear mapping which populates the concept A with all constants appearing in the unary database relation A' (refer to the proof of Theorem 9 for the definition of pop): $\mathcal{M}_1 = \mathcal{M}_2 = \mathsf{pop}(\{A'\}, \{A\})$. To complete the proof, we prove in the appendix that $\langle \mathcal{T}_1, \mathcal{M}_1\rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2\rangle$ iff $\langle \mathcal{T}_2, \{A(c)\}\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \{A(c)\}\rangle \models q(\vec{a})$ for every CQ q with $\mathsf{sig}(q) \subseteq \mathsf{sig}(\mathcal{T}_2)$. □

The complexity of CQ-entailment is significantly lower for *DL-Lite*$_{core}$ and *DL-Lite*$_{RDFS}$:

Theorem 12. *For DL-Lite*$_{RDFS}$*, the CQ-entailment problem between OBDA specifications is NL-complete for linear mappings and NP-complete for GAV and GLAV mappings; for DL-Lite*$_{core}$*, the problem is in PTIME for linear mappings, NP-complete for GAV mappings, and in* Π_2^p *for GLAV mappings.*

Proof. We give the arguments for *DL-Lite*$_{core}$ (the rest of the proof is in the appendix). For linear mappings, membership in PTIME easily follows from the fact that, by Theorem 4, to check $\langle \mathcal{T}_1, \mathcal{M}_1\rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2\rangle$ it is sufficient to construct in polynomial time each of the databases D from the theorem statement, together with the corresponding ABoxes $\mathcal{A}_{\mathcal{M}_1,D}$ and $\mathcal{A}_2$, and then check whether the KB $\langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1,D}\rangle$ $(\mathsf{sig}(\mathcal{K}_2), \mathsf{const}(\mathcal{A}_2))$-entails the KB $\langle \mathcal{T}_2, \mathcal{A}_2\rangle$ (note: that since we consider linear mappings, $\mathsf{const}^{\exists}(\mathcal{A}_2) = \emptyset$). We return yes if all of these checks succeed. Since KB query entailment *without constant restrictions* was proven to be in PTIME for *DL-Lite*$_{core}$ KBs (Botoeva et al. 2014), membership in PTIME follows.

For GAV mappings, NP-hardness can be proven by an easy reduction of the CQ containment problem. As for the upper bound, we know from Theorem 4 that to check $\langle \mathcal{T}_1, \mathcal{M}_1\rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2\rangle$ it is sufficient to check a polynomial number of database-ABox pairs $(D, \mathcal{A}_2)$. For each such pair $(D, \mathcal{A}_2)$, we guess a subset $\mathcal{A}' \subseteq \mathcal{A}_{\mathcal{M}_1,D}$ (together with a polynomial proof that every $\alpha \in \mathcal{A}'$ belongs to $\mathcal{A}_{\mathcal{M}_1,D}$) and then check in PTIME that $\langle \mathcal{T}_1, \mathcal{A}'\rangle$ $(\mathsf{sig}(\mathcal{K}_2), \mathsf{const}(\mathcal{A}_2))$-entails the KB $\langle \mathcal{T}_2, \mathcal{A}_2\rangle$, returning yes if all checks succeed. By grouping all of the above guesses into a single one, we obtain an NP decision procedure.

Finally, in the case of GLAV mappings, we can show membership in Π_2^p as follows. By Theorem 4, $\langle \mathcal{T}_1, \mathcal{M}_1\rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2\rangle$ iff $\langle \mathcal{T}_1, \mathcal{M}_1\rangle \models_{\bot} \langle \mathcal{T}_2, \mathcal{M}_2\rangle$ (which can be checked in NP by Theorem 8) and for every pair $(D, \mathcal{A}_2)$ constructed according to the theorem statement, we have $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ with $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. Because of Theorem 6, we know that the latter holds iff there exists a subset $\mathcal{A}_1 \subseteq \mathcal{A}_{\mathcal{M}_1,D}$ whose size is bounded polynomially in $|D|$, $|\mathcal{A}_2|$, $|\mathcal{T}_2|$, and $|\mathcal{M}_1|$ such that the KB $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1\rangle$ (Σ, Υ)-entails the KB $\mathcal{K}_2 = \langle \mathcal{T}_2, \mathcal{A}_2\rangle$, where $\Sigma = \mathsf{sig}(\mathcal{K}_2)$ and $\Upsilon = \mathsf{const}(\mathcal{A}_2) \setminus \mathsf{const}^{\exists}(\mathcal{A}_2)$. Since KB entailment with both signature and constant restrictions is NP-complete for *DL-Lite*$_{core}$ KBs (Botoeva et al. 2015), it is in NP to decide whether such a subset $\mathcal{A}_1$ exists. We can thus define a Π_2^p procedure which ranges over all possible pairs $(D, \mathcal{A}_2)$, and for every such pair, uses an NP oracle to check both that $\langle \mathcal{T}_1, \mathcal{M}_1\rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2\rangle$ and that there exists a subset $\mathcal{A}_1 \subseteq \mathcal{A}_{\mathcal{M}_1,D}$ with the required properties. □

Our final result shows that IQ-entailment has the same complexity as single IQ-entailment. The proof exploits Theorem 5 and proceeds similarly to the proof of Theorem 10.

Theorem 13. *For DL-Lite*$_{RDFS}$*, DL-Lite*$_{core}$ *and DL-Lite*$_R$*, the IQ-entailment problem between OBDA specifications is NP-complete for both GAV and GLAV mappings, and is NL-complete for linear mappings.*

6 Conclusion and Future Work

In this paper, we have introduced notions of query-based entailment of OBDA specifications and have analyzed the complexity of checking query-based entailment for different classes of queries and mappings and for TBoxes formulated in *DL-Lite*$_R$ and its sublogics *DL-Lite*$_{core}$ and *DL-Lite*$_{RDFS}$.

The present work constitutes only a first step towards a full analysis of query-based forms of comparing OBDA specifications and can be extended in several directions:

- First, we could introduce a query signature and only test entailment for queries formulated in the given signature, as has been done for TBox and KB query inseparability (Botoeva et al. 2014). In fact, for GAV mappings (which are the most commonly used in practice), our complexity upper bounds continue to hold in the presence of a query signature. We leave open whether this is the case for CQ-entailment with GLAV mappings.
- Second, it would be interesting to extend the computational analysis of query entailment to other DLs beyond *DL-Lite*$_R$. We expect that the techniques from this paper can be extended to handle Horn versions of DL-Lite, but that fundamentally new ideas will be needed for DLs allowing for role functionality or recursion.
- Third, other forms of mapping beyond GAV and GLAV could be analyzed. In particular, we would like to see whether decidability of query entailment is preserved if we add some restricted form of inequality or negation to the mapping bodies. This is especially relevant in the context of R2RML (Das, Cyganiak, and Sundara 2012), the W3C language for mapping relational databases to RDF. The query-based comparison of specifications based upon R2RML mappings is in general undecidable, since

R2RML can express arbitrary SQL queries over the database. It would thus be interesting to identify fragments for which query-based entailment is decidable.

- Fourth, to explore the impact of restricting the set of possible databases, we could extend the computational analysis to database schemas with integrity constraints.
- Finally, it would be interesting to develop and experiment practical algorithms for comparing OBDA specifications using query-based notions of entailment.

7 Acknowledgements

This work was partially supported by the EU (FP7 project Optique, grant n. FP7-318338) and the French National Research Agency (ANR JCJC grant n. ANR-12-JS02-007-01).

References

Antonioli, N.; Castanò, F.; Civili, C.; Coletta, S.; Grossi, S.; Lembo, D.; Lenzerini, M.; Poggi, A.; Savo, D. F.; and Virardi, E. 2013. Ontology-based data access: the experience at the Italian Department of Treasury. In *Proc. of the 25th Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, Industrial Track, 9–16.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The *DL-Lite* family and relations. *J. Artif. Intell. Res. (JAIR)* 36:1–69.

Bienvenu, M., and Rosati, R. 2016. Long version of this paper (with appendix). Available at `http://www.lirmm.fr/~meghyn/papers/BieRos-KR16-long.pdf`.

Bienvenu, M.; Lutz, C.; and Wolter, F. 2012. Query containment in description logics reconsidered. In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*.

Botoeva, E.; Kontchakov, R.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2014. Query inseparability for description logic knowledge bases. In *Proc. of 14th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*.

Botoeva, E.; Kontchakov, R.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2015. Games for query inseparability of description logic knowledge bases. To appear in *Artif. Intell. J. (AIJ)*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Autom. Reasoning (JAR)* 39(3):385–429.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. The Mastro system for ontology-based data access. *Semantic Web J.* 2(1):43–53.

Das, S.; Cyganiak, R.; and Sundara, S. 2012. R2RML: RDB to RDF mapping language. W3C recommendation, W3C. `http://www.w3.org/TR/2012/REC-r2rml-20120927/`.

Doan, A.; Halevy, A. Y.; and Ives, Z. G. 2012. *Principles of Data Integration*. Morgan Kaufmann.

Fagin, R.; Kolaitis, P. G.; Nash, A.; and Popa, L. 2008. Towards a theory of schema-mapping optimization. In *Proc. of the 27th Symposium on Principles of Database Systems (PODS)*, 33-42.

Giese, M.; Soylu, A.; Vega-Gorgojo, G.; Waaler, A.; Haase, P.; Jiménez-Ruiz, E.; Lanti, D.; Rezk, M.; Xiao, G.; Özçep, Ö. L.; and Rosati, R. 2015. Optique: Zooming in on big data. *IEEE Computer* 48(3):60–67.

Gottlob, G.; Pichler, R.; and Savenkov, V. 2011. Normalization and optimization of schema mappings. *Very Large Database J.* 20(2):277–302.

Guha, R., and Brickley, D. 2014. RDF schema 1.1. W3C recommendation, W3C. `http://www.w3.org/TR/2014/REC-rdf-schema-20140225/`.

Kharlamov, E.; Giese, M.; Jiménez-Ruiz, E.; Skjæveland, M. G.; Soylu, A.; Zheleznyakov, D.; Bagosi, T.; Console, M.; Haase, P.; Horrocks, I.; Marciuska, S.; Pinkel, C.; Rodriguez-Muro, M.; Ruzzi, M.; Santarelli, V.; Savo, D. F.; Sengupta, K.; Schmidt, M.; Thorstensen, E.; Trame, J.; and Waaler, A. 2013. Optique 1.0: Semantic access to big data: The case of Norwegian Petroleum Directorate's FactPages. In *Proc. of the ISWC Posters & Demos Track*, 65–68.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2009. Formal properties of modularisation. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, *LNCS* vol. 5445, 25–66.

Konev, B.; Kontchakov, R.; Ludwig, M.; Schneider, T.; Wolter, F.; and Zakharyaschev, M. 2011. Conjunctive query inseparability of OWL 2 QL TBoxes. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI)*.

Kontchakov, R.; Wolter, F.; and Zakharyaschev, M. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell. J. (AIJ)* 174(15):1093–1141.

Lembo, D.; Mora, J.; Rosati, R.; Savo, D. F.; and Thorstensen, E. 2014. Towards mapping analysis in ontology-based data access. In *Proc. of the 8th Int. Conf. on Web Reasoning and Rule Systems (RR)*, 108–123.

Lembo, D.; Mora, J.; Rosati, R.; Savo, D. F.; and Thorstensen, E. 2015. Mapping analysis in ontology-based data access: Algorithms and complexity. In *Proc. of the 14th Int. Semantic Web Conf. (ISWC)*, 217–234.

Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic EL. *J. Symb. Comput. (JSC)* 45(2):194–228.

Motik, B.; Grau, B. C.; Horrocks, I.; Fokoue, A.; and Wu, Z. 2012. OWL 2 Web Ontology Language profiles (2nd edition). W3C recommendation, W3C. `http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/`.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *J. on Data Semantics* X:133–173.

Rodriguez-Muro, M.; Kontchakov, R.; and Zakharyaschev, M. 2013. Ontology-based data access: Ontop of databases. In *Proc. of the 12th Int. Semantic Web Conf. (ISWC)*, *LNCS* vol. 8218, 558–573.

A Proof details for Section 4

We complete the proof of Theorem 4.

Proof of Theorem 4 (continued). Suppose that $\Gamma_1 \models_\perp \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every CQ q and every database D and ABox $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ obtained as in the theorem statement. Let us further suppose that $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models q_0(\vec{a})$, where D_0 is another database. The first possibility is that $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models \perp$, in which case $\Gamma_1 \models_\perp \Gamma_2$ yields $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models \perp$ and thus, $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_0(\vec{a})$. The other possibility is that $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \models q_0(\vec{a})$ and $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \not\models \perp$. If $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models \perp$, we are done.

Otherwise, we are in the case in which both $\langle \mathcal{T}_2, \mathcal{M}_2, D_0\rangle \not\models \perp$ and $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \not\models \perp$. The beginning of the argument for this case was given in the body of the text. We complete the proof by showing that $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_i(\vec{b}_i)$, for every $1 \leq i \leq n$.

We begin by establishing the following claim, which shows that only a small number of assertions in $\mathcal{A}_{\mathcal{M}_2,D_0}$ are needed to derive $q_i(\vec{b}_i)$.

Claim. For every $1 \leq i \leq n$, there exists a mapping assertion $m_i \in \mathcal{M}_2$ and a homomorphism h_i of $body(m_i)$ into D_0 such that $\langle \mathcal{T}_2, gr(h_i(head(m_i)))\rangle \models q_i(\vec{b}_i)$.

Proof of claim. First suppose that q_i is of the first type. Then $q_i = \alpha$ for some atom $\alpha \in q_0$ such that all variables in α are mapped by π to constants from D_0 or $\mathcal{M}_2$. It is a well-known property of *DL-Lite*$_R$ (see e.g., (Calvanese et al. 2007)) that there must exist an assertion $\beta \in \mathcal{A}_{\mathcal{M}_2,D_0}$ such that $\langle \mathcal{T}_2, \{\beta\}\rangle \models \pi(\alpha)$. Since $\beta \in \mathcal{A}_{\mathcal{M}_2,D_0}$, there is some mapping $m_i \in \mathcal{M}_2$ and a homomorphism h_i of $body(m_i)$ into D_0 such that $\beta \in gr(h_i(head(m_i)))$, and hence such that $\langle \mathcal{T}_2, gr(h_i(head(m_i)))\rangle \models q_i(\vec{b}_i)$.

Next suppose that q_i is of the second type. We know that for every pair of atoms $\alpha, \alpha' \in q_i$, there exists a sequence of atoms $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_{\ell-1}, \alpha_\ell = \alpha'$ such that for every $0 \leq j < \ell$, there exists $v_i \in \mathsf{vars}(\alpha_j) \cap \mathsf{vars}(\alpha_{j+1})$ such that $h(v_i) \in (\Delta^{\mathcal{I}_2} \setminus \mathsf{const}(\mathcal{A}_{\mathcal{M}_2,D_0})) \cup \mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_2,D_0})$. We then note that an element $aw \in (\Delta^{\mathcal{I}_2} \setminus \mathsf{const}(\mathcal{A}_{\mathcal{M}_2,D_0}))$ is only connected via roles to elements of the form aw' (with w' possibly empty), and constants from $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_2,D_0})$ can only be connected via roles to constants that occur in the same mapping head. It follows that there must exist a mapping assertion $m_i \in \mathcal{M}_2$ and a homomorphism h_i of $body(m_i)$ into D_0 such that all constants in $\{a \mid \exists v \in \mathsf{vars}(q_i), h(v) = aw\}$ appear in $gr(h_i(head(m_i)))$.

Finally, consider the third case, in which q_i contains no constants and none of its variables are mapped to constants in $\mathcal{A}_{\mathcal{M}_2,D}$. In this case, we know that $\{h(v) \mid v \in q_i\}$ forms a connected subset of $\Delta^{\mathcal{I}_2} \setminus \mathsf{const}(\mathcal{A}_{\mathcal{M}_2,D_0})$, and thus there must exist a constant $a \in \mathsf{const}(\mathcal{A}_{\mathcal{M}_2,D})$ and a role R such that every element in $\{h(v) \mid v \in q_i\}$ takes the form aRw for some (possibly empty) word w. Since a occurs in $\mathcal{A}_{\mathcal{M}_2,D_0}$, there must exist a mapping assertion $m_i \in \mathcal{M}_2$ and a homomorphism h_i of $body(m_i)$ into D_0 such that $a \in \mathsf{const}(gr(h_i(head(m_i))))$. (*end proof of claim*)

Let m_i and h_i be as in the preceding claim, and let m_i' be obtained from m_i by identifying all pairs of frontier variables y, z such that $h_i(y) = h_i(z)$. By construction, h_i is an injective homomorphism of $body(m_i')$ into D_0, and $\langle \mathcal{T}_2, gr(h_i(head(m_i')))\rangle \models q_i(\vec{b}_i)$.

We set $D_{m_i'} = gr(body(m_i'))$, and we let f_i be the natural isomorphism from $body(m_i')$ into $D_{m_i'}$ that maps each variable in $body(m_i')$ to the corresponding constant in $D_{m_i'}$. Observe that $D_{m_i'} = f_i(body(m_i'))$. We define two further functions relating constants in D_0 and $D_{m_i'}$:

- a function g_i mapping the constants in $D_{m_i'}$ to constants in D_0 defined by setting $g_i(c) = h_i(f_i^{-1}(c))$;
- a function g_i' mapping the constants in D_0 that occur in the image of h_i to constants in $D_{m_i'}$ defined by setting $g_i'(c) = f_i(h_i^{-1}(c))$.

Note that this definition relies upon the injectivity of h_i and f_i, which ensures that h_i^{-1} and f_i^{-1} are well defined.

We can extend g_i' to all of the constants occurring in $gr(h_i(head(m_i')))$ by mapping the unique constant in $gr(h_i(head(m_i')))$ corresponding to the existentially quantified variable u in $head(m_i')$ to the unique constant in $gr(f_i(head(m_i')))$ corresponding to u. Observe that g_i' defines a homomorphism (in fact, an isomorphism) of $gr(h_i(head(m_i')))$ into $gr(f_i(head(m_i')))$.

From $\langle \mathcal{T}_2, gr(h_i(head(m_i')))\rangle \models q_i(\vec{b}_i)$ and the fact that g_i' is a homomorphism of $gr(h_i(head(m_i')))$ into $gr(f_i(head(m_i')))$, it follows that $\langle \mathcal{T}_2, gr(f_i(head(m_i')))\rangle \models q_i(g_i'(\vec{b}_i))$, and hence $\langle \mathcal{T}_2, \mathcal{M}_2, D_{m_i'}\rangle \models q_i(g_i'(\vec{b}_i))$. By construction, $D_{m_i'}$ satisfies the conditions of the theorem statement, so we can apply our initial assumption to obtain $\langle \mathcal{T}_1, \mathcal{M}_1, D_{m_i'}\rangle \models q_i(g_i'(\vec{b}_i))$.

As $f_i^{-1}(D_{m_i'}) = body(m_i')$ and $h_i(body(m_i')) \subseteq D_0$, it follows that $g_i(D_{m_i'}) \subseteq D_0$. In other words, g_i defines a homomorphism of $D_{m_i'}$ into D_0. It follows that $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_i(g_i(g_i'(\vec{b}_i)))$. Since $g_i(g_i'(\vec{b}_i)) = \vec{b}_i$, this yields $\langle \mathcal{T}_1, \mathcal{M}_1, D_0\rangle \models q_i(\vec{b}_i)$. □

Proof of Theorem 5. We can use a similar argument to that used in the proof of Theorem 4. The main differences can be summarized as follows:

- We do not need to break down the instance query q_0 into subqueries, as it consists of a single atom, all of whose variables are mapped to constants from D_0 or $\mathcal{M}_2$.
- We prove the following stronger claim: there exists a mapping assertion $m \in \mathcal{M}_2$, an atom $\alpha \in m$, and a homomorphism h of $body(m)$ into D_0 such that $\langle \mathcal{T}_2, gr(h(\alpha))\rangle \models q_0(\vec{a})$.
- Since we are only concerned with the head atom α, we let m' be the mapping obtained by unifying the at most two variables occurring in α, if they are mapped to the same constant by h.

The latter point ensures that the database $D_{m'}$ is of the required form. □

In preparation for the proof of Theorem 6, we introduce some additional notation and recall some facts about the canonical model construction.

The notation $\leq$ will be used to indicate the *prefix relation* between words (or domain elements): $u < u'$ means that there exists a (possibly empty) word u'' such that $u' = uu''$; for the strict prefix relation (when u'' must be non-empty), we use $<$ instead. Note that if $e, e' \in \Delta^{\mathcal{T},\mathcal{A}}$ and $e \leq e'$, then e and e' begin with the same constant. By construction, the set of elements in $\Delta^{\mathcal{T},\mathcal{A}}$ is *prefix-closed*: if $e' \in \Delta^{\mathcal{T},\mathcal{A}}$ and $e \leq e'$, then $e' \in \Delta^{\mathcal{T},\mathcal{A}}$. Thus, the elements of $\Delta^{\mathcal{T},\mathcal{A}}$ can be naturally viewed as a set of trees. We call $w \in \Delta^{\mathcal{T},\mathcal{A}}$ an *ancestor* of $w' \in \Delta^{\mathcal{T},\mathcal{A}}$ if $w < w'$; w is the *parent* of w' if $w' = wR$.

With every element $e \in \Delta^{\mathcal{T},\mathcal{A}}$ is naturally associated a subinterpretation $\mathcal{I}^{\mathcal{T},\mathcal{A}}|_e$ obtained by restricting $\mathcal{I}^{\mathcal{T},\mathcal{A}}$ to those elements e' of $\Delta^{\mathcal{I}}$ such that $e \leq e'$. Observe that if $e = awR$ and $e = bw'R$ are elements of $\Delta^{\mathcal{T},\mathcal{A}}$ that end with the same role R, then the function f mapping $awRw''$ to $bw'Rw''$ defines an isomorphism from $\mathcal{I}^{\mathcal{T},\mathcal{A}}|_e$ to $\mathcal{I}^{\mathcal{T},\mathcal{A}}|_{e'}$.

In the following proof, given a (partial) function f, we will use $\mathsf{dom}(f)$ to refer to the domain of f (i.e., the elements for which f is defined), and we will use $\mathsf{range}(f)$ to refer to the range of f.

For the sake of readability, we provide the full proof of Theorem 6 (including the parts that were included in the proof sketch in the main text).

Proof of Theorem 6. Suppose that $\mathcal{A}_2 \subseteq \mathcal{A}_{\mathcal{M}_2,D}$ and that $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ such that $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. First consider the case in which $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models \bot$, and hence $\langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1,D}\rangle \models \bot$. It is well known that if an ABox is inconsistent with a *DL-Lite*$_R$ TBox, then there exists a subset of the ABox of cardinality at most two that is inconsistent with the TBox. We can thus extract a subset $\mathcal{A}_1 \subseteq \mathcal{A}_{\mathcal{M}_1,D}$ such that $|\mathcal{A}_1| \leq 2$ and $\langle \mathcal{T}_1, \mathcal{A}_1\rangle \models \bot$.

Let us now consider the more interesting case in which $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \not\models \bot$. By definition, $\langle \mathcal{T}_2, \mathcal{M}_2, D\rangle \not\models \bot$, so $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \not\models \bot$. We let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the canonical models of $\langle \mathcal{T}_1, \mathcal{A}_{\mathcal{M}_1,D}\rangle$ and $\langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2,D}\rangle$ respectively. By assumption, $\langle \mathcal{T}_2, \mathcal{A}_2\rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{M}_1, D\rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ with $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. It follows from results in (Botoeva et al. 2015) that the latter property is true just in the case that there is a homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$ that is the identity on the constants from D and $\mathcal{M}_2$. More precisely, there is a function $h : \Delta^{\mathcal{I}_2} \to \Delta^{\mathcal{I}_1}$ such that:

- $h(e) = e$ for every $e \in \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$
- $e \in A^{\mathcal{I}_2}$ implies $h(e) \in A^{\mathcal{I}_1}$
- $(e, e') \in P^{\mathcal{I}_2}$ implies $(h(e), h(e')) \in P^{\mathcal{I}_1}$

Note that the constants in $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_2,D})$ need not be mapped to themselves.

Our objective is to show how to use h to construct a homomorphism g from $\mathcal{I}_2$ to $\mathcal{I}_1$ with the property that the set $\Theta = \{a \mid aw \in \mathsf{range}(g)\}$ has size polynomially bounded in the size of D, $\mathcal{A}_2$, $\mathcal{T}_2$, and $\mathcal{M}_1$ (more precisely: of size at most $|\mathcal{A}_2| \cdot |\mathcal{M}_1| + 2|D| \cdot |\mathcal{T}_2| \cdot |\mathcal{M}_1|$). This will allow us to identify a small subset of $\mathcal{A}_{\mathcal{M}_1,D}$ that is sufficient to infer all of the queries involving constants from D or $\mathcal{M}_2$ that are entailed by $\langle \mathcal{T}_2, \mathcal{A}_2\rangle$. To construct the desired homomorphism g, we will define a sequence $g_0, g_1, g_2 \ldots$ of partial homomorphisms from $\mathcal{I}_2$ to $\mathcal{I}_1$ such that g_{i+1} extends g_i for every $i \geq 0$ (that is, $\mathsf{dom}(g_i) \subseteq \mathsf{dom}(g_{i+1})$ and $g_{i+1}(e) = g_i(e)$ for all $e \in \mathsf{dom}(g_i)$). We will ensure that for every $e \in \Delta^{\mathcal{I}_2}$, there is some i such that $e \in \mathsf{dom}(g_i)$, which allows us to take g to be the limit of this sequence. In what follows, we will use Θ_i to refer to the set $\{a \mid aw \in \mathsf{range}(g_i)\}$.

For the initial partial homomorphism g_0, we set $\mathsf{dom}(g_0)$ equal to

$$\mathsf{const}(\mathcal{A}_2) \cup \{awR \mid h(e) \notin \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1) \text{ for all } e < awR\}$$

and set $g_0(e) = h(e)$ for every $e \in \mathsf{dom}(g_0)$. Intuitively, g_0 copies the value of h for all constants in $\mathcal{A}_2$ as well as for all elements e such that every ancestor of e is mapped to an element of $\Delta^{\mathcal{I}_1} \setminus (\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1))$. Because of the way we defined $\mathsf{dom}(g_0)$ and the fact that roles can only link constants from $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_1,D})$ to constants issuing from the same grounded mapping head, we know that if $g_0(b) = h(b) = au$, $bw \in \mathsf{dom}(g_0)$, and $g_0(bw) = h(bw) = a'u'$, then there exists a mapping assertion $m \in \mathcal{M}_1$ such that $D \models \exists \vec{y}.body(m(\vec{a}))$ and $a, a' \in \mathsf{const}(gr(head(m(\vec{a}))))$. Thus, $|\Theta_0| \leq |\mathcal{A}_2| \cdot |\mathcal{M}_1|$.

At each stage $i \geq 1$, we will extend g_{i-1} to some additional elements from $\Delta^{\mathcal{I}_2} \setminus \mathsf{const}(\mathcal{A}_2)$. The idea is to exploit the regularity of canonical models in order to identify subtrees of $\mathcal{I}_2$ with similar structure and to modify the homomorphism h so that similar subtrees are mapped in the same way into $\mathcal{I}_1$. We will use a table $\mathsf{MapTrees}$ to keep track of how we have mapped some representative subtrees and to reuse this information when extending our partial homomorphisms. Formally, $\mathsf{MapTrees}$ will contain tuples of the form (d, R, W, σ, e) where:

- d is a constant from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$
- $R \in \mathsf{N}_\mathsf{R}^{\pm} \cap \mathsf{sig}(\mathcal{T}_2)$
- W is a (possibly infinite) prefix-closed set of words over the alphabet $\mathsf{N}_\mathsf{R}^{\pm} \cap \mathsf{sig}(\mathcal{T}_2)$
- σ is a function mapping words in W to elements in $\Delta^{\mathcal{I}_1}$
- e is an element from $\Delta^{\mathcal{I}_2}$

where we use $\mathsf{N}_\mathsf{R}^{\pm}$ as an abbreviation for $\{P, P^- \mid P \in \mathsf{N}_\mathsf{R}\}$. Intuitively, the tuple (d, R, W, σ, e) means that when the parent of an element $e' \in \Delta^{\mathcal{I}_2}$ is mapped to d and the final letter in e' is R, then we need to extend the homomorphism to all elements in $\{e'w \mid w \in W\}$ by mapping $e'w$ to $\sigma(w)$. The final argument e indicates that σ copies information from the subtree rooted at e (this will be formalized below). Like g, the table $\mathsf{MapTrees}$ will be constructed in stages; we will use $\mathsf{MapTrees}_i$ to denote the state of $\mathsf{MapTrees}$ at the end of stage i. We start with an empty table ($\mathsf{MapTrees}_0 = \emptyset$), and at every stage, we will add zero or one new tuples ($\mathsf{MapTrees}_{i-1} \subseteq \mathsf{MapTrees}_i$ and $|\mathsf{MapTrees}_i \setminus \mathsf{MapTrees}_{i-1}| \leq 1$).

For correctness and to achieve the required bound on the cardinality of the sets Θ_i, we will ensure that the following conditions are satisfied by g_i and $\mathsf{MapTrees}_i$ for every $i \geq 0$:

(P1) If $\mathsf{MapTrees}_i \setminus \mathsf{MapTrees}_{i-1} = \{(d, R, W, \sigma, e)\}$, then:

- $\mathsf{MapTrees}_{i-1}$ does not contain any tuple $(d, R, \ldots)$
- $e \in \mathsf{dom}(g_{i-1})$ and $g_{i-1}(e) = d$
- for every $w \in W$: $eRw \in \Delta^{\mathcal{I}_2} \setminus \mathsf{dom}(g_{i-1})$, $eRw \in \mathsf{dom}(g_i)$, and $g_i(eRw) = h(eRw) = \sigma(w)$
- $\mathsf{dom}(g_i) \setminus \mathsf{dom}(g_{i-1}) = \{eRw \mid w \in W\}$

(P2) If $\mathsf{MapTrees}_i = \mathsf{MapTrees}_{i-1}$, then there exists $(d, R, W, \sigma, e) \in \mathsf{MapTrees}_{i-1}$ and $e'R \in \Delta^{\mathcal{I}_2} \setminus \mathsf{dom}(g_{i-1})$ such that:

- $e' \neq e$, $e' \in \mathsf{dom}(g_{i-1})$, and $g_{i-1}(e') = d$
- for every $w \in W$, $e'Rw \in \Delta^{\mathcal{I}_2}$, $e'Rw \in \mathsf{dom}(g_i)$, $g_i(e'Rw) = \sigma(w)$
- $\mathsf{dom}(g_i) \setminus \mathsf{dom}(g_{i-1}) = \{e'Rw \mid w \in W\}$

(P3) For every tuple $(d, R, W, \sigma, e) \in \mathsf{MapTrees}_i$ and every leaf $w \in W$ (that is, there is no $w' \in W$ with $w < w'$), we have $\sigma(w) \in \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$.

(P4) g_i is a partial homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$ that is the identity on constants from D and $\mathcal{M}_2$.

(P5) For every function σ appearing in the fourth column of $\mathsf{MapTrees}_i$, there exists a mapping assertion $m \in \mathcal{M}_1$ such that $D \models \exists \vec{y}.body(m(\vec{a}))$ and $\{a \mid aw \in \mathsf{range}(\sigma)\} \subseteq \mathsf{const}(gr(head(m(\vec{a}))))$.

Observe that the preceding properties are trivially satisfied by the initial function g_0 and table $\mathsf{MapTrees}_0$.

We now explain how to construct g_i and $\mathsf{MapTrees}_i$ starting from g_{i-1} and $\mathsf{MapTrees}_{i-1}$. We first select an element $awR \in \Delta^{\mathcal{I}_2}$ such that $aw \in \mathsf{dom}(g_{i-1})$, $awR \notin \mathsf{dom}(g_{i-1})$, and there is no $a'w'R' \in \Delta^{\mathcal{I}_2}$ with $a'w' \in \mathsf{dom}(g_{i-1})$ and $aw'R' \notin \mathsf{dom}(g_{i-1})$ such that either:

- $|a'w'| < |aw|$, or
- $|a'w'| = |aw|$ and there exists j such that $a'w' \in \mathsf{dom}(g_j)$ but $aw \notin \mathsf{dom}(g_j)$.

It follows from properties (**P1**)-(**P3**) and the definition of g_0 that $g_{i-1}(aw) \in \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$. There are two cases to consider:

Case 1: $\mathsf{MapTrees}_{i-1}$ does not contain any tuple of the form $(g_{i-1}(aw), R, W, \sigma, e)$

We define a set of W^R_{aw} of words as follows:

$$W^R_{aw} = \{w' \mid \text{for every } w'' < w', \\ h(awRw') \notin \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)\}$$

and we set $\sigma^R_{aw}(w') = h(awRw')$ for every $w' \in W^R_{aw}$. We let $\mathsf{MapTrees}_i$ be the result of adding the tuple $(g_{i-1}(aw), R, W^R_{aw}, \sigma^R_{aw})$ to $\mathsf{MapTrees}_{i-1}$, and we extend g_{i_1} to the set of elements $\{awRw' \mid w' \in W^R_{aw}\}$ by setting $g_i(awRw') = h(awRw')$ (equivalently, $g_i(awRw') = \sigma^R_{aw}(w')$). By construction, g_i and $\mathsf{MapTrees}_i$ satisfy properties (**P1**) and (**P3**), and (**P2**) trivially holds. It remains to show that (**P5**) and (**P4**) hold.

For property (**P4**), we begin by showing that $g_{i-1}(aw) = h(aw)$. Suppose for a contradiction that this is not the case. Then we know that $aw \notin \mathsf{dom}(g_0)$ (since $g_0(e) = h(e)$ for all $e \in \mathsf{dom}(g_0)$). Let j be such that $aw \in \mathsf{dom}(g_j) \setminus \mathsf{dom}(g_{j-1})$. We must have $\mathsf{MapTrees}_j = \mathsf{MapTrees}_{j-1}$, since otherwise, by property (**P1**), we would have $g_j(aw) = h(aw)$, hence $g_{i-1}(aw) = h(w)$. By properties (**P1**) and (**P2**), there exist $(c, S, W, \sigma, e) \in \mathsf{MapTrees}_{j-1}$ and $auR \in \Delta^{\mathcal{I}_2} \setminus \mathsf{dom}(g_{j-1})$ such that:

- $au \in \mathsf{dom}(g_{j-1})$ and $g_{j-1}(au) = c$
- there exists $u' \in W$ such that $aw = auSu'$
- $g_j(aw) = g_j(auSu') = \sigma(u')$

Since $(c, S, W, \sigma, e) \in \mathsf{MapTrees}_{j-1}$ and $u' \in W$, the element eSu' belongs to $\mathsf{dom}(g_{j-1})$ by (**P1**), whereas $aw = auSu' \notin \mathsf{dom}(g_{j-1})$. Moreover, because of the way we select elements for treatment, we know that $|e| \leq |au|$, hence $|eSu'| \leq |auSu'| = |aw|$. It follows that at stage i, our strategy for selecting elements would lead us to choose eSu' rather than aw, a contradiction. We have thus established that $g_{i-1}(aw) = h(aw)$. Since aw is the only element in $\mathsf{dom}(g_{i-1})$ that is connected via a role to an element in $\mathsf{dom}(g_i) \setminus \mathsf{dom}(g_{i-1})$, $g_i(e'') = h(e'')$ for every $e'' \in \mathsf{dom}(g_i) \setminus \mathsf{dom}(g_{i-1})$, and h is a homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$, it follows that g_i is a partial homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$. Thus, g_i satisfies property (**P4**).

Finally, we consider property (**P5**). Recall that we have defined $\mathsf{dom}(\sigma^R_{aw}) = W^R_{aw}$ and $\sigma^R_{aw}(w') = h(awRw')$ for every $w' \in W^R_{aw}$. Thus, $\{b \mid bv \in \mathsf{range}(\sigma^R_{aw})\} = \{b \mid h(awRw') = bv \text{ for some } w' \in W^R_{aw}\}$. Since g_i is a partial homomorphism (property (**P4**)), we know that if two elements $e_1, e_2 \in \{awRw' \mid w' \in W^R_{aw}\}$ are related by a role in $\mathcal{I}_2$, then $h(e_1)$ and $h(e_2)$ must also be related by a role in $\mathcal{I}_1$. By the structure of the ABox $\mathcal{A}_{\mathcal{M}_1,D}$ and the definition of canonical models, if there is a role between $h(e_1)$ and $h(e_2)$, then one of the following must hold:

- $h(e_2) = h(e_1)T$ for some role T
- $h(e_1) = h(e_2)T$ for some role T
- $h(e_1), h(e_2) \in \mathsf{const}(\mathcal{A}_{\mathcal{M}_1,D})$ and there exists a mapping assertion $m \in \mathcal{M}_1$ such that $D \models \exists \vec{y}.body(m(\vec{a}))$ and $h(e_1), h(e_2)$ occur in a role assertion in $gr(head(m(\vec{a})))$.

Note that in the first two cases, $h(e_1)$ and $h(e_2)$ share the same initial constant. Recall also that by construction, $g_i(awRw') \notin \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$ for every $w' \in W^R_{aw}$ that is not a leaf of W^R_{aw}, and by property (**P3**), $g_i(awRw') \in \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$ for every leaf $w' \in W^R_{aw}$. It follows that every path from $h(awR)$ to $h(awRw')$ with $w' \in W^R_{aw}$ a leaf in W^R_{aw} contains exactly one element from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$, namely, $h(awRw')$. Earlier in the path, we can move between adjacent elements in the tree rooted at a constant, between constants from $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_1,D})$ that occur in the same grounded mapping head, or at the end of the path, from a constant from $\mathsf{const}^{\exists}(\mathcal{A}_{\mathcal{M}_1,D})$ to a constant from $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_1)$. Thus, there must exist a single mapping assertion $m \in \mathcal{M}_1$ such that $D \models \exists \vec{y}.body(m(\vec{a}))$ and $\{b \mid h(awRw') = bv \text{ for some } w' \in W^R_{aw}\} \subseteq \mathsf{const}(gr(head(m(\vec{a}))))$.

Case 2: $\mathsf{MapTrees}_{i-1}$ contains $(g_{i-1}(aw), R, W, \sigma, e)$

We extend g_i to set of elements $\{awRw' \mid w' \in W\}$ by setting $g_i(awRw') = \sigma(w')$. We do not modify the table: $\mathsf{MapTrees}_i = \mathsf{MapTrees}_{i-1}$. Properties (**P1**), (**P3**), and (**P5**) are trivially satisfied for g_i and $\mathsf{MapTrees}_i$. For (**P2**), we note that since $(g_{i-1}(aw), R, W, \sigma, e) \in \mathsf{MapTrees}_{i-1}$ but $awR \notin \mathsf{dom}(g_{i-1})$, we must have $aw \neq e$. The remainder of the first bullet, as well as the other two bullets, are satisfied by construction. To show the remaining property (**P4**), first note that by property (**P1**), we have $g_i(aw) = g_{i-1}(e)$ and $g_i(awRw') = g_{i-1}(eRw')$ for every $w' \in W$ Since g_{i-1} satisfies (**P4**), we know that g_{i-1} is a partial homomorphism. To conclude the argument, we recall that the subinterpretation $\mathcal{I}_2|_{awR}$ is isomorphic to $\mathcal{I}_2|_{eR}$, with isomorphism f given by $f(awRw') = ewRw'$.

We have defined the sequence of partial homomorphisms $g_0, g_1, g_2, \ldots$ from $\mathcal{I}_2$ to $\mathcal{I}_1$. Our selection strategy ensures that for every element $e \in \Delta^{\mathcal{I}_2}$ there exists i such that $e \in \mathsf{dom}(g_i)$. Moreover, if $e \in \mathsf{dom}(g_i)$, then $g_i(e) = g_j(e)$ for every $j \geq i$. We can thus define g as follows: $g(e) = g_i(e)$ for some (equivalently, every) i with $e \in \mathsf{dom}(g_i)$. Since every g_i is a partial homomorphism, it follows that g is a (total) homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$ that is the identify on $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$.

Let us now show that $\Theta = \{a \mid aw \in \mathsf{range}(g)\}$ is of size at most $|\mathcal{A}_2| \cdot |\mathcal{M}_1| + 2|\mathcal{T}_2| \cdot |\mathcal{M}_1| \cdot (|D| + |\mathcal{M}_1|)$. As noted earlier, $|\Theta_0| \leq |\mathcal{A}_2| \cdot |\mathcal{M}_1|$. Let us now consider the size of $\Theta \setminus \Theta_0 = \{a \mid aw \in \mathsf{range}(g) \setminus \mathsf{range}(g_0)\}$. We know from properties (**P1**) and (**P2**) that for every $e \in \mathsf{dom}(g_i) \setminus \mathsf{dom}(g_{i-1})$, there exists a tuple $(d, R, W, \sigma, e') \in \mathsf{MapTrees}_i$ such that $g_i(e) \in \mathsf{range}(\sigma)$. Thus, we need to determine the cardinality of

$$\{a \mid aw \in \mathsf{range}(\sigma) \text{ for some } (d, R, W, \sigma, e') \in \mathsf{MapTrees}\}$$

Since there can be at most one tuple of the form (d, R, W, σ, e') per pair (d, R), the total number of different functions σ cannot exceed $2|\mathcal{T}_2| \cdot (|\mathsf{const}(D)| + |\mathsf{const}(\mathcal{M}_1)|)$ (here $2|\mathcal{T}_2|$ is used to bound the possible values of R). By property (**P5**), for every such function σ, there exists a mapping assertion $m \in \mathcal{M}_1$ such that $D \models \exists \vec{y}.body(m(\vec{a}))$ and $\{a \mid aw \in \mathsf{range}(\sigma)\} \subseteq \mathsf{const}(gr(head(m(\vec{a}))))$. The cardinality of the latter set cannot exceed the maximal number of terms appearing in the head of a mapping assertion in $\mathcal{M}_1$, and hence is bounded above by $|\mathcal{M}_1|$. Putting all of this together yields the desired bound of $|\mathcal{A}_2| \cdot |\mathcal{M}_1| + 2|\mathcal{T}_2| \cdot |\mathcal{M}_1| \cdot (|D| + |\mathcal{M}_1|)$.

Now we let $\mathcal{A}_0$ be the restriction of $\mathcal{A}_{\mathcal{M}_1,D}$ to the constants in Θ. To define $\mathcal{A}_1$, for every constant $c \in \mathcal{A}_0$ such that there exists d with $(c, d) \in R^{\mathcal{I}_1}$, we pick one such constant d, which we will refer to by $\mathsf{wit}(c, R)$. Then the ABox $\mathcal{A}_1$ is defined as follows:

$$\mathcal{A}_1 = \mathcal{A}_0 \cup \{R(c, \mathsf{wit}(c, R)) \mid c \in \mathsf{const}(\mathcal{A}_0), \exists d\, R(c, d) \in \mathcal{A}_0\}$$

Note that R may be an inverse role r^-, in which case we use $R(a, b)$ to refer to the assertion $r(b, a)$. Take $\mathcal{I}_2'$ to be the restriction of $\mathcal{I}_2$ to $\{aw \mid a \in \Theta\}$, and let $\mathcal{J}$ be the canonical model of the KB $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle$. It is readily verified, using the definition of canonical models and of the ABox $\mathcal{A}_1$, that the identify function is a homomorphism of $\mathcal{I}_2'$ into $\mathcal{J}$. Since g is a homomorphism of $\mathcal{I}_1$ into $\mathcal{I}_2$ whose range is contained in $\{aw \mid a \in \Theta\}$, g must also be a homomorphism of $\mathcal{I}_1$ into $\mathcal{I}_2'$, and hence also into $\mathcal{J}$. Moreover, g maps all constants in $\mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$ to themselves. We can thus conclude that $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \models q(\vec{a})$ implies $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle \models q(\vec{a})$ for every Boolean CQ $q(\vec{a})$ with $\vec{a} \subseteq \mathsf{const}(D) \cup \mathsf{const}(\mathcal{M}_2)$. □

B Proof details for Section 5

We provide details of the argument for the NL upper bound in the proof of Theorem 7.

Proof of Theorem 7 (continued).

Proof. We consider the case in which $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$, $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ both have linear mappings, and we explain how to decide in NL whether $\Gamma_1 \models_{\mathsf{log}} \mathcal{M}_2$. The procedure will consider in turn each of the mapping assertions in $\mathcal{M}_2$, which can be done in logarithmic space by keeping only the id of mapping assertions in memory, rather than the assertions themselves. For each mapping assertion $m \in \mathcal{M}_2$, we do not build the database $D = gr(body(m))$ explicitly, but instead exploit the fact that D has precisely the same structure as $body(m)$, which is available in the input. In other words, instead of replacing every variable by a constant, we simply treat the existing variables as constants. We store in memory (a binary representation of) the atom $\alpha = head(m)$; this requires only logarithmic space since the arity of α is at most two. Our aim is to determine whether $\langle \Gamma_1, body(m) \rangle \models \alpha$, where the variables in $body(m)$ and α are treated as constants. Because $\mathcal{T}_1$ is a *DL-Lite*$_R$ TBox, we know that this entailment holds iff there exists a mapping assertion $m' \in \mathcal{M}_1$ and a homomorphism h from $body(m')$ to $body(m)$ such that $\langle \mathcal{T}_1, h(head(m')) \rangle \models \alpha$. Moreover, since we are working with linear mappings, both $body(m')$ and $body(m)$ consist of a single atom, which means that $body(m')$ and $body(m)$ must use the same relation name and h is the unique function such that $h(body(m')) = body(m)$. We can thus proceed as follows:

1. We guess the (id of a) mapping assertion $m' \in \mathcal{M}_1$.
2. We check in the input that $body(m)$ and $body(m')$ use the same relation name.
3. We iterate over the positions $1 \leq i \leq \mathsf{arity}(P)$ and over positions $i < j \leq \mathsf{arity}(P)$ (by representing the positions in binary, we need only logarithmic space to store the pair (i, j)). If $body(m')$ contains the same term in positions i and j, then we check that the term in position i of $body(m)$ is the same as that in position j.
4. We compute and store binary representations of the (at most two) frontier variables in m' Then, for each frontier variable v, we scan $body(m')$ for the first position i in which v occurs, and we memorize the term occurring in position i of $body(m)$.
5. We let β be obtained by taking $head(m')$ and replacing each frontier variable v by the term in m that we memorized in the previous step. A binary representation of the atom β can be stored in logarithmic space.

6. Finally, since IQ answering in *DL-Lite*$_R$ is in NL, we can make a call to an NL oracle to verify that $\langle \mathcal{T}_1, \beta \rangle \models \alpha$.
7. If all of the preceding checks succeeded, we return yes. Otherwise, we return no.

Note that if the checks in Steps 2 and 3 succeed, then the atom β constructed in Step 5 is precisely $h(head(m'))$. Thus, the procedure correctly checks whether $\langle \Gamma_1, body(m) \rangle \models \alpha$. Moreover, since the procedure runs in logarithmic space with access to an NL oracle and $\mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$, we obtain an NL procedure. □

The next lemma establishes the claim from the proof of Theorem 9.

Lemma 1. *$\langle \emptyset, \mathcal{M}_1 \rangle \models_q \langle \emptyset, \mathcal{M}_2 \rangle$ iff $\langle \mathcal{T}, \mathcal{A} \rangle \models q_1 \Rightarrow \langle \mathcal{T}, \mathcal{A} \rangle \models q_2$ for all Σ-ABoxes.*

Proof. First suppose that $\langle \emptyset, \mathcal{M}_1 \rangle \models_q \langle \emptyset, \mathcal{M}_2 \rangle$, and let $\mathcal{A}$ be an ABox such that $\mathsf{sig}(\mathcal{A}) \subseteq \Sigma$ and $\langle \mathcal{T}, \mathcal{A} \rangle \models q_1$. Let D be the database instance consisting of the facts in $\mathcal{A}$. Since $\mathcal{M}_2$ contains $\mathsf{copy}^1(\Sigma)$ as well as $\{T(x) \rightarrow V(x), F(x) \rightarrow V(x)\}$, it follows that $\langle \emptyset, \mathcal{M}_2, D \rangle \models q_1'$. Moreover, because $\mathcal{M}_2$ contains $\mathsf{pop}(\Sigma, \Sigma^2)$, the corresponding ABox $\mathcal{A}_{\mathcal{M}_2, D}$ contains for every constant c in D (equiv. $\mathcal{A}$), every Σ^2-fact whose only constant is c. More precisely, for every such constant c, the ABox $\mathcal{A}_{\mathcal{M}_2, D}$ contains $A^2(c)$ for every concept name $A \in \Sigma$ and $R^2(c, c)$ for every role name $R \in \Sigma$. Using the fact that q_2' is Boolean and only uses predicates from Σ^2, we can infer that $\langle \emptyset, \mathcal{M}_2, D \rangle \models q_2'$ and hence that $\langle \emptyset, \mathcal{M}_2, D \rangle \models q$. By our assumption that $\langle \emptyset, \mathcal{M}_1 \rangle \models_q \langle \emptyset, \mathcal{M}_2 \rangle$, we obtain $\langle \emptyset, \mathcal{M}_1, D \rangle \models q$, which implies that $\langle \emptyset, \mathcal{M}_1, D \rangle \models q_2'$. We then observe that $\mathcal{A}_{\mathcal{M}_1, D}$ contains $P^2(\vec{c})$ iff $\mathcal{A}$ contains the corresponding assertion $P(\vec{c})$. It follows that the homomorphism witnessing that $\langle \emptyset, \mathcal{M}_1, D \rangle \models q_2'$ can be reproduced in $\mathcal{A}$ using the original predicates, so $\mathcal{A} \models q_2$.

For the other direction, suppose that for all Σ-ABoxes, $\langle \mathcal{T}, \mathcal{A} \rangle \models q_1$ implies $\langle \mathcal{T}, \mathcal{A} \rangle \models q_2$. Let D be a database instance such that $\langle \emptyset, \mathcal{M}_2, D \rangle \models q$. It follows that $\langle \emptyset, \mathcal{M}_2, D \rangle \models q_1'$. Note that since the left-hand sides of mapping assertions in $\mathcal{M}_2$ only use predicates from Σ, we may assume w.l.o.g. that $\mathsf{sig}(D) \subseteq \Sigma$. Let $\mathcal{A}_D$ be the Σ-ABox containing the facts in D. From $\langle \emptyset, \mathcal{M}_2, D \rangle \models q_1'$ and the fact that the two inclusions in $\mathcal{T}$ simulate the effect of the mapping assertions $T(x) \rightarrow V(x)$ and $F(x) \rightarrow V(x)$, we can infer that $\langle \mathcal{T}, \mathcal{A}_D \rangle \models q_1$. Applying our assumption, we obtain $\langle \mathcal{T}, \mathcal{A}_D \rangle \models q_2$, which yields $\langle \emptyset, \mathcal{M}_1, D \rangle \models q_2'$ because of the mapping assertions in $\mathsf{copy}^2(\Sigma)$. We can also show that $\langle \emptyset, \mathcal{M}_1, D \rangle \models q_1'$ using the mapping assertions in $\mathsf{pop}(\Sigma, \Sigma^1 \cup \{V\})$, from which we obtain $\langle \emptyset, \mathcal{M}_1, D \rangle \models q$, as desired. □

We give the missing proof of the linear case for Theorem 10.

Proof of Theorem 10 (continued). In the case of linear mappings, we can again use Theorem 3 to restrict the number and form of the databases that need to be considered. By Theorem 8, we can check in NL whether $\Gamma_1 \models_\bot \Gamma_2$. We can next iterate over all of pairs $(D, \mathcal{A}_2)$ constructed in the manner specified in Theorems 3. Observe that because $\mathcal{M}_2$ is a linear mapping, every pair $(D, \mathcal{A}_2)$ corresponds to take a mapping assertion $m \in \mathcal{M}_2$, possibly unifying the (at most two) frontier variables in m, and setting $D = gr(body(m))$ and $\mathcal{A}_2 = gr(head(m))$. As in the proof of Theorems 7 and 8, we can work directly with $body(m)$ and $head(m)$, rather than introducing new constants, and we do not need to explicitly construct $body(m)$ but can instead store the id of m and the decision of whether or not to merge the frontier variables. We next iterate (again in logarithmic space) over all (unary or binary) tuples $\vec{a} \subseteq \mathsf{terms}(head(m))$ of the same arity as q. For every such tuple $\vec{a}$, we make a call to an NL oracle to check whether the IQ $q(\vec{a})$ is entailed from $\langle \mathcal{T}_2, body(m) \rangle$. If the response is yes, then we make a second NL oracle call to check whether $\langle \Gamma_1, body(m) \rangle \models q(\vec{a})$ (refer to the proof of Theorem 7 for a more detailed description of how to perform this entailment check in NL). The procedure returns no if there is some m obtained from a mapping assertion in $\mathcal{M}_2$ by possibly unifying the frontier variables and some tuple $\vec{a} \subseteq \mathsf{terms}(head(m))$ for which $\langle \mathcal{T}_2, body(m) \rangle \models q(\vec{a})$ and $\langle \Gamma_1, body(m) \rangle \not\models q(\vec{a})$; otherwise, it returns yes. The correctness of the procedure is easily verified, and since $\mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$, we obtain an NL decision procedure.

For the NL lower bound, we reduce concept subsumption to single IQ entailment with linear mappings as follows: $\mathcal{T} \models A \sqsubseteq B$ iff $\langle \mathcal{T}, \{A'(x) \rightarrow A(x)\} \rangle \models_{B(x)} \langle \mathcal{T}, \{A'(x) \rightarrow B(x)\} \rangle$. □

The following lemma, which will be used in the proof of Theorem 12, shows that if we consider OBDA specifications based upon *DL-Lite*$_{RDFS}$ TBoxes and GAV mappings, then CQ-entailment and IQ-entailment coincide.

Lemma 2. *Consider OBDA specifications $\Gamma_1 = \langle \mathcal{T}_1, \mathcal{M}_1 \rangle$ and $\Gamma_2 = \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ where $\mathcal{T}_1, \mathcal{T}_2$ are DL-Lite$_{RDFS}$ TBoxes and $\mathcal{M}_1, \mathcal{M}_2$ are GAV mappings. Then $\Gamma_1 \models_{\mathsf{CQ}} \Gamma_2$ iff $\Gamma_1 \models_{\mathsf{IQ}} \Gamma_2$.*

Proof. By Proposition 2, $\Gamma_1 \models_{\mathsf{CQ}} \Gamma_2$ implies $\Gamma_1 \models_{\mathsf{IQ}} \Gamma_2$. For the other direction, suppose that $\Gamma_1 \models_{\mathsf{IQ}} \Gamma_2$ and $\langle \mathcal{T}_2, \mathcal{M}_2, D \rangle \models q(\vec{a})$. Since $\mathcal{T}_2$ is formulated in *DL-Lite*$_{RDFS}$, we know that it does not contain any existentials on the right-hand-side of inclusions. It must thus be the case that there is a function $h : \mathsf{terms}(q) \rightarrow \mathsf{const}(\mathcal{A}_{\mathcal{M}_2, D})$ that maps every constant to itself and the tuple of answer variables of q to the tuple of constants $\vec{a}$ and which satisfies $\{h(\alpha) \mid \alpha \in q\} \subseteq \mathcal{A}_{\mathcal{M}_2, D}$. From $\Gamma_1 \models_{\mathsf{IQ}} \Gamma_2$, we can infer that $\{h(\alpha) \mid \alpha \in q\} \subseteq \mathcal{A}_{\mathcal{M}_1, D}$, and hence that $\langle \mathcal{T}_1, \mathcal{M}_1, D \rangle \models q(\vec{a})$. This establishes that $\Gamma_1 \models_{\mathsf{CQ}} \Gamma_2$. □

We now complete the proof of Theorem 12 by proving the complexity results for *DL-Lite*$_{RDFS}$.

Proof of Theorem 12 (continued).

For linear mappings, NL-hardness is easily obtained by a reduction of concept subsumption in *DL-Lite*$_{RDFS}$ similar to the previous ones. More precisely, it is immediate to verify that $\mathcal{T} \models A \sqsubseteq B$ iff $\langle \mathcal{T}, \{T(x) \rightarrow A(x)\} \rangle \models_{\mathsf{CQ}} \langle \mathcal{T}, \{T(x) \rightarrow A(x), T(x) \rightarrow B(x)\} \rangle$. (Note that this same

reduction works for *DL-Lite*$_{core}$.) Membership in NL is an immediate consequence of Lemma 2 and Theorem 13.

For GAV mappings, NP-hardness is easily obtained through a reduction of the containment problem for conjunctive queries. To show membership in NP for *DL-Lite*$_{RDFS}$ and GLAV mappings, we let $\mathsf{sat}(\mathcal{M}, \mathcal{T})$ denote the *saturation of* $\mathcal{M}$ *by* $\mathcal{T}$, which is obtained by replacing each mapping assertion $m \in \mathcal{M}$ by the mapping assertion m' which has the same body as m, but its head contains all atoms in $head(m)$, *plus* all atoms that can be inferred from $head(m)$ using $\mathcal{T}$. For example, if $\mathcal{T} = \{A \sqsubseteq B, \exists S^- \sqsubseteq D, R \sqsubseteq S\}$, we would replace the mapping assertion $P(x, y, z) \rightarrow \exists w.A(x) \wedge R(x, w)$ by

$$P(x, y, z) \rightarrow \exists w.A(x) \wedge R(x, w) \wedge S(x, w) \wedge B(x) \wedge D(w)$$

It is not hard to see that for every database D, the ABoxes $\mathcal{A}_{\mathcal{M},D}$ and $\mathcal{A}_{\emptyset,\mathsf{sat}(\mathcal{M},\mathcal{T})}$ are identical (modulo renaming of the constants issuing from the grounding of existential variables). It follows that $\langle \mathcal{T}, \mathcal{M} \rangle \equiv_{\mathsf{CQ}} \langle \emptyset, \mathsf{sat}(\mathcal{M}, \mathcal{T}) \rangle$. Thus, we have $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathsf{CQ}} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ iff $\langle \emptyset, \mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \rangle \models_{\mathsf{CQ}} \langle \emptyset, \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2) \rangle$. We next argue that the latter holds just in the case that $\mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \models_{\mathsf{log}} \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2)$. The direction from $\mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \models_{\mathsf{log}} \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2)$ to $\langle \emptyset, \mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \rangle \models_{\mathsf{CQ}} \langle \emptyset, \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2) \rangle$ is a simple consequence of Proposition 2. For the other direction, suppose that $\langle \emptyset, \mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \rangle \models_{\mathsf{CQ}} \langle \emptyset, \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2) \rangle$, and let $m \in \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2)$. Let $D_m = gr(body(m))$ be the database corresponding to the body of m, let $q = head(m)$, and let $\vec{b}$ be the constants in D_m corresponding to the grounding of the frontier variables of m. By construction, we have $\langle \emptyset, \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2), D_m \rangle \models q(\vec{b})$, and so applying our assumption, we obtain $\langle \emptyset, \mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1), D_m \rangle \models q(\vec{b})$. It follows that $\mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \models m$, which yields $\mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \models_{\mathsf{log}} \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2)$. We thus obtain the following NP decision procedure for CQ-entailment for *DL-Lite*$_{RDFS}$ TBoxes and GLAV mappings: (i) construct in polynomial time the saturated mappings $\mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1)$ and $\mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2)$, (ii) test whether $\mathsf{sat}(\mathcal{M}_1, \mathcal{T}_1) \models_{\mathsf{log}} \mathsf{sat}(\mathcal{M}_2, \mathcal{T}_2)$ using the NP procedure described in the proof of Theorem 7, and (iii) return yes iff the procedure in (ii) returned yes. □

Proof of Theorem 13. The proof of all the upper bounds follows from Theorem 10 and from the fact that $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_{\mathsf{IQ}} \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ iff $\langle \mathcal{T}_1, \mathcal{M}_1 \rangle \models_\alpha \langle \mathcal{T}_2, \mathcal{M}_2 \rangle$ for every instance query α over $\mathsf{sig}(\langle \mathcal{T}_2, \mathcal{A}_{\mathcal{M}_2,D} \rangle)$. Since the number of such queries α is linearly bounded (and the queries can be enumerated in logarithmic space), it follows that all the upper bounds of Theorem 10 extend to the IQ-entailment problem.

All the NL lower bounds for all the DLs and linear mappings follow from an easy reduction of the problem of concept subsumption in a TBox constituted of inclusions between atomic concepts (such a TBox belongs to all the DLs considered), which is an NL-hard problem.

Finally, it is easy to prove the NP lower bound for the case of empty TBoxes and GAV mappings through a reduction of containment between conjunctive queries: given two Boolean CQs q_1, q_2, q_1 is contained in q_2 if and only if $\langle \emptyset, \{q_2 \rightarrow A(a)\} \rangle \models_{\mathsf{IQ}} \langle \emptyset, \{q_1 \rightarrow A(a)\} \rangle$. □