Non-Monotonic Ontology-based Abstractions of Data Services

Gianluca Cima[1] , **Maurizio Lenzerini**[1] , **Antonella Poggi**[1,2]
[1]Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma
[2]Dipartimento di Lettere e Culture Moderne, Sapienza Università di Roma
{cima, lenzerini, poggi}@diag.uniroma1.it

Abstract

In Ontology-Based Data Access (OBDA), a domain ontology is linked to the data sources of an organization in order to query, integrate and manage data through the concepts and relations of the domain of interest, thus abstracting from the structure and the implementation details of the data layer. While the great majority of contributions in OBDA in the last decade have been concerned with the issue of computing the answers of queries expressed over the ontology, recent papers address a different problem, namely the one of providing suitable abstractions of data services, i.e., characterizing or explaining the semantics of queries over the sources in terms of queries over the domain ontology. Current works on this subject are based on expressing abstractions in terms of unions of conjunctive queries (UCQs). In this paper we advocate the use of a non-monotonic language for this task. As a first contribution, we present a simple extension of UCQs with non-monotonic features, and show that non-monotonicity provides more expressive power in characterizing the semantics of data services. A second contribution is to prove that, similarly to the case of monotonic abstractions, depending on the expressive power of the languages used to specify the various components of the OBDA system, there are cases where neither perfect nor approximated abstractions exist for a given data service. As a third contribution, we single out interesting special cases where the existence of abstractions is guaranteed, and we present algorithms for computing such abstractions in these cases.

1 Introduction

The *Ontology-Based Data Access* (OBDA) paradigm (Poggi et al. 2008) has been the subject of many investigations in the last decade. An OBDA specification Σ consists of a triple $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O}$ is an ontology expressed in Description Logic (DL) (Baader et al. 2003), $\mathcal{S}$ is the schema of the data sources forming the data layer of an information system, and $\mathcal{M}$ is a mapping between the source schema and the ontology. The ontology is a logic-based representation of the underlying domain, and the mapping specifies the relationship between the data at the sources and the elements in the ontology. Thus, OBDA provides a means for managing data through the lens of an ontology (Lenzerini 2018), and enables the application of Knowledge Representation and Reasoning principles and techniques to various data management tasks.

As testified by (Xiao et al. 2018; Ortiz 2018; Bienvenu 2016), the vast majority of papers dealing with OBDA concentrate on query answering. The rewriting approach to this problem is as follows: given a user query q expressed over the ontology, find the so-called ontology-to-source rewriting of q, i.e., a query over the source schema that, once executed over the data, provides the certain answers to q.

Recent papers (Cima 2017; Lutz, Marti, and Sabellek 2018; Cima, Lenzerini, and Poggi 2019) address a different issue in OBDA: starting from a query $q_{\mathcal{S}}$ expressed over the sources, the goal is to find the so-called *source-to-ontology rewriting* (*s-to-o rewriting* for short) of $q_{\mathcal{S}}$, i.e., a query $q_{\mathcal{O}}$ over the ontology that is equivalent to the original query, modulo the ontology and the mapping. Thus, $q_{\mathcal{O}}$ represents an abstraction of the data service represented by $q_{\mathcal{S}}$ in terms of the domain ontology, obtained through the mapping. Such notion of abstraction is relevant in different data management scenarios, such as the ones discussed in (Cima, Lenzerini, and Poggi 2019; Lutz, Marti, and Sabellek 2018). As a notable example, it can be used for providing a semantic explanation, i.e., a formulation in terms of the domain vocabulary, of services expressed over the data layer, such as queries, and other data analytics tasks.

Example 1. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be as follows:*

$$\mathcal{O} = \{ \mathsf{Infected} \sqsubseteq \exists \mathsf{HadContacts}, \mathsf{Married} \sqsubseteq \mathsf{HadContacts}, \mathsf{Patient0} \sqsubseteq \mathsf{Infected}, \mathsf{Recovered} \sqsubseteq \mathsf{Infected} \}$$

$$\mathcal{S} = \{ s_1, s_2, s_3, s_4, s_5 \}$$

$$\mathcal{M} = \left\{ \begin{array}{lcl} s_1(x) & \rightarrow & \mathsf{Infected}(x), \\ s_2(x_1,x_2) & \rightarrow & \mathsf{HadContacts}(x_1,x_2), \\ s_3(x_1,x_2) & \rightarrow & \mathsf{Married}(x_1,x_2), \\ \exists y. s_1(x) \wedge s_4(x,y) & \rightarrow & \mathsf{Patient0}(x), \\ s_5(x) & \rightarrow & \mathsf{Recovered}(x) \end{array} \right\}$$

We aim at an abstraction for the data service $q_{\mathcal{S}}(x) \leftarrow s_1(x)$ w.r.t. Σ. One can verify that there is no query over $\mathcal{O}$ that precisely describes the data service $q_{\mathcal{S}}$ in terms of $\mathcal{O}$. On the other hand, if we are happy with a sound approximation of $q_{\mathcal{S}}$ expressed as UCQ over $\mathcal{O}$, then the query returning all the certain answers of Patient0 *is the best we can achieve. Observe, however, that if we extend the language used to express abstractions with non-monotonic features, then it is not difficult to see that the best sound characterization of $q_{\mathcal{S}}$ in terms of $\mathcal{O}$ is the query returning the union of the certain answers of* Patient0 *with those certain answers of* Infected *that are* not *certain answers of* Recovered*.*

In this paper, we start an investigation of the notion of abstraction in the case where s-to-o rewritings can be expressed in a non-monotonic query language. Obviously, one basic issue to address in this endeavour is selecting the non-monotonic query language. Our choice in this paper is to use *EQL-Lite*(UCQ) (Calvanese et al. 2007a), that is based on a variant of the well-known first-order modal logic of knowledge/belief (Levesque 1984; Reiter 1990; Levesque and Lakemeyer 2001). The language incorporates a minimal knowledge operator $\mathbf{K}$, which is used to formalize the epistemic state of the knowledge base. Informally, the formula $\mathbf{K}\phi$ is read as "ϕ is known to hold (by the knowledge base)". Queries in *EQL-Lite*(UCQ) can use conjunction, negation, and existential quantification, and have atoms that are expressed exactly as $\mathbf{K}\phi$, where ϕ is a union of conjunctive query (UCQ). With this combination of operators we can ask for those x such that a given $\phi(x)$ is not known to hold, and this is crucial for characterizing a set of tuples that are not certain answers to a given source query. By exploiting such features, one can show that the best sound characterization of the query $q_{\mathcal{S}}$ mentioned in Example 1 is $q_{\mathcal{O}}(x) \leftarrow \mathbf{K}(\mathsf{Patient0}(x)) \vee (\mathbf{K}(\mathsf{Infected}(x)) \wedge \neg\mathbf{K}(\mathsf{Recovered}(x)))$. The epistemic operator enables also other interesting features. For instance, we can distinguish between asking for those x such that it is known that there is y for which $R(x, y)$ holds (where y can be unknown), and asking for those x such that there is y for which $R(x, y)$ is known to hold (and therefore y is known).

The issue of using non-monotonic query languages in OBDA has been rarely addressed. Our work is actually the first to consider s-to-o rewritings that exploit non-monotonicity in order to provide powerful abstractions of data services. We believe that non-monotonic abstractions can be extremely useful for providing more informative explanations of data services or data sources. Referring to Example 1, the non-monotonic abstraction of $q_{\mathcal{S}}$ makes it clear that no instances of Recovered can be inferred to be stored in source s_1, except for those that are also instances of Patient0. These kinds of characterizations for the content of data sources or data services cannot be produced if we just use UCQs as target language.

Our contributions in this paper can be summarized as follows. Considering as a starting point a specific framework considered in (Cima, Lenzerini, and Poggi 2019), we propose the language *EQL-Lite*(UCQ) for expressing queries over the ontology, and we discuss why such language provides a better means to compute abstractions of data services compared to the language of UCQs (Section 3). In particular, we show that there are cases where the perfect s-to-o rewriting is expressible as an *EQL-Lite*(UCQ) query, but *not* as a UCQ. Also, there are cases where the maximally sound s-to-o rewriting exists in the class of *EQL-Lite*(UCQ) queries, but *not* in the class of UCQ, and cases where such maximally sound s-to-o rewriting is a better approximation than the analogous in the class of UCQ. On the other hand, we also show that, similarly to UCQs (Cima, Lenzerini, and Poggi 2019), it may happen that no maximally sound s-to-o rewritings exists in the class of *EQL-Lite*(UCQ) queries, and the same holds for minimally complete s-to-o rewritings (Section 4). In order to address the issue of non-expressibility, we explore two special cases. In the first case, we limit the mapping language, and consider the so-called one-to-one mapping, where each mapping assertion links one source relation to one ontology element (Section 5). In the second case, we limit the query language, and we consider a weak version of *EQL-Lite*(UCQ), where both nested negation and union are disallowed (Section 6). In both special cases, we address the problem of computing the minimally complete and the maximally sound s-to-o rewriting of a source query, presenting algorithms whenever possible.

2 Preliminaries

We assume basic knowledge about databases (Abiteboul, Hull, and Vianu 1995) and Description Logics (DLs) (Baader et al. 2003).

Database and Queries. A *database schema* (or simply *schema*) $\mathcal{S}$ is a finite set of predicate symbols, each with a specific arity. An $\mathcal{S}$-*database* D is a finite set of *facts* $s(\vec{c})$, where s is an n-ary predicate symbol of $\mathcal{S}$, and $\vec{c} = (c_1, \ldots, c_n)$ is an n-tuple of constants, each taken from a denumerable infinite set of symbols denoted by Const.

In its general form, an $\mathcal{L}$-*query* q over a schema $\mathcal{S}$ is a function in a certain class $\mathcal{L}$ that can be *evaluated* over an $\mathcal{S}$-database D to return the set of *answers* q^D, each answer being a tuple of constants. We assume to deal with databases supporting queries in First-Order Logic (FOL). Let $\vec{t}$ be a (possibly empty) tuple of terms, each term being either a constant or a variable, and $\mathcal{S}$ a schema. In general, a *query q with target list $\vec{t}$ over $\mathcal{S}$* has the form $q(\vec{t}) \leftarrow \phi(\vec{x})$, where ϕ (called the *body* of the query) is a FOL open formula whose free variables $\vec{x}$ (also called *distinguished variables*) are the variables appearing in $\vec{t}$. When the target list is empty, we say that q is *boolean*. Also, we accept queries whose body is $\bot$ or $\top$. If q is a query of arity n with target list $\vec{t}$, and $\vec{c}$ is an n-tuple of constants, then by $q(\vec{t}/\vec{c})$ we denote the boolean query obtained from $q(\vec{t})$ by substituting each t_i with c_i. A *conjunctive query (CQs)* CQ is a FOL query whose body is an existentially quantified finite conjunction of atoms. We often write a CQ in the form $q(\vec{t}) \leftarrow \phi(\vec{x}, \vec{y})$, where in the body of the query we explicitly indicate with $\vec{x}$ the distinguished variables of q (i.e., the variables in $\vec{t}$), and with $\vec{y}$ its existentially quantified variables. We also use the notation $\phi(\vec{x}, Y)$ instead of $\phi(\vec{x}, \vec{y})$, where $Y = \{y_1, \ldots, y_k\}$ is the set of variables in $\vec{y}$, and we write $\exists Y$ instead of $\exists y_1.\exists y_2.\ldots.\exists y_k$. Also, we write $\exists y_1.\exists y_2.\ldots.\exists y_k.\phi$ as $\exists y_1, y_2, \ldots, y_k.\phi$. Given a CQ $q(\vec{t}) \leftarrow \phi(\vec{x}, \vec{y})$, we say that an existential variable y in $\vec{y}$ is a *join existential variable* if it occurs more than once in ϕ. The *conjunctive queries with join-free existential variables (CQJFEs)* are CQs with no join existential variables. A UCQ is a *union* of a finite set of CQs with same arity, each called a *disjunct* of the query. We assume that the semantics of FOL queries is known. We only point out that, for every D, if the body of q is $\bot$, then $q^D = \emptyset$, and if the body of q is $\top$, then q^D is the set of all tuples of constants in D whose arity is the arity of the target list of q.

Ontologies. A DL *ontology* $\mathcal{O}$ is simply a TBox (set of terminological axioms) expressed in a specific DL ($\mathcal{O}$ can be trivially seen as comprising a schema). We are interested in DL ontologies expressed in *DL-Lite*$_{\mathcal{R}}^{-}$, the fragment of the DL *DL-Lite*$_{\mathcal{R}}$ (Calvanese et al. 2004; Calvanese et al. 2007b)[1] without disjointness axioms. The semantics of a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology $\mathcal{O}$ is specified through the usual notion of FOL interpretation (simply interpretation, in the following) $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\mathcal{I}$ is a model of $\mathcal{O}$, denoted by $\mathcal{I} \models \mathcal{O}$, if it satisfies every axiom in $\mathcal{O}$.

OBDA. An *Ontology-based Data Access (OBDA) specification* (Poggi et al. 2008) consists of a triple $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O}$ is a DL ontology, $\mathcal{S}$, called *source schema*, is a schema, and $\mathcal{M}$ is a *mapping*, i.e., a finite set of mapping assertions relating $\mathcal{S}$ to $\mathcal{O}$. Mapping assertions are FOL implications of the form $\forall \vec{x}. \phi(\vec{x}, Y) \rightarrow \varphi(\vec{x}, Z)$, where $\phi(\vec{x}, Y)$ and $\psi(\vec{x}, Z)$ are bodies of a CQ over $\mathcal{S}$ and $\mathcal{O}$, respectively (Lenzerini 2002; Doan, Halevy, and Ives 2012). Mapping assertions of the above form, also written as $\phi(\vec{x}, Y) \rightarrow \varphi(\vec{x}, Z)$, are called *GLAV (Global-and-Local-as-View)* mapping assertions. Special cases of GLAV mapping assertions are called *pure GAV* and *one-to-one*. A pure GAV mapping assertion is a GLAV mapping assertion in which the right-hand side of the implication is simply an atom without existential variables, constants, or repeated variables, i.e, it is either of the form $A(x)$, or $P(x_1, x_2)$, with x_1, x_2 different variables. A one-to-one mapping assertion is a GLAV mapping assertion where both $\phi(\vec{x}, Y)$ and $\psi(\vec{x}, Z)$ are simply atoms without constants or repeated variables.

For the semantics of an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, we consider interpretations $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{O}$ with $\Delta^{\mathcal{I}} = \mathsf{Const}$[2]. An interpretation $\mathcal{I}$ for $\mathcal{O}$ is a *model for* Σ *relative to an* $\mathcal{S}$*-database* D if (*i*) $\mathcal{I} \models \mathcal{O}$, and (*ii*) $\langle D, \mathcal{I} \rangle \models \mathcal{M}$, i.e., the FOL interpretation constituted by D and $\mathcal{I}$, denoted by $\langle D, \mathcal{I} \rangle$, satisfies all mapping assertions in $\mathcal{M}$. The set of models for Σ relative to an $\mathcal{S}$-database D is denoted by $Mod_D(\Sigma)$, and D is said to be consistent with Σ if $Mod_D(\Sigma) \neq \emptyset$. Note that, if $\mathcal{O}$ is expressed in *DL-Lite*$_{\mathcal{R}}^{-}$, then $Mod_D(\Sigma) \neq \emptyset$ for every D.

Chase. Given $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a CQ $q(\vec{t}) \leftarrow \phi(\vec{x}, Y)$ over $\mathcal{S}$, we denote by $\mathcal{M}(q)$ the conjunction of atoms over $\mathcal{O}$ obtained by applying the so-called *oblivious chase* (Calì, Gottlob, and Kifer 2013) to the *freezing* of the body of q (denoted as F_q) w.r.t. the set of rules repesented by $\mathcal{M}$. Briefly speaking, the oblivious chase (or, simply chase) of a set of atoms F over $\mathcal{S}$ w.r.t. $\mathcal{M}$ is a set of atoms over $\mathcal{O}$ computed as follows: (*i*) we start with an empty set of atoms $J = \emptyset$, then (*ii*) for every GLAV assertion $\phi(\vec{x}, Y) \rightarrow \varphi(\vec{x}, Z)$ in $\mathcal{M}$ and for every homomorphism h from $\phi(\vec{x}, Y)$ to F, we add to J the image of the set of atoms appearing in $\varphi(\vec{x}, Z)$ under h', that is, $J = J \cup h'(\varphi(\vec{x}, Z))$, where h' extends h by assigning to each variable $z \in Z$ a different fresh variable. We observe that each variable is taken from a denumerable infinite set of symbols denoted by Var, where $\mathsf{Const} \cap \mathsf{Var} = \emptyset$.

If we start from the $\mathcal{S}$-database D instead of q, we can compute $\mathcal{M}(D)$, and if we apply the chase to $\mathcal{M}(D)$ w.r.t. the rules corresponding to the axioms in $\mathcal{O}$, following a deterministic strategy that is fair (i.e., is such that if at some point a rule is applicable then it will be eventually applied), then we obtain a (possibly infinite) set $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ of atoms that, when restricted to the alphabet of $\mathcal{O}$, can be seen as an interpretation for $\mathcal{O}$. The structure $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ is called *canonical interpretation of* $\mathcal{O}$ *w.r.t.* Σ *and* D (Calvanese et al. 2007b).

Certain Answers. Given an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, a query $q_{\mathcal{O}}$ over $\mathcal{O}$, and an $\mathcal{S}$-database D, the *certain answers of* $q_{\mathcal{O}}$ *w.r.t.* Σ *and* D is the set of tuples $\vec{c}$ of constants in D such that $\vec{c} \in q_{\mathcal{O}}^{\mathcal{I}}$ for every $\mathcal{I} \in Mod_D(\Sigma)$, where $\mathcal{I}$ is seen as a set of facts over $\mathcal{O}$. We denote by $cert_{q_{\mathcal{O}},\Sigma}$, the query over $\mathcal{S}$ such that for every $\mathcal{S}$-database D, $cert_{q_{\mathcal{O}},\Sigma}^{D}$ coincides with the certain answers of $q_{\mathcal{O}}$ w.r.t. Σ and D. It is well known that, for every D consistent with Σ, $cert_{q_{\mathcal{O}},\Sigma}^{D} = q^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}\downarrow$, i.e., the certain answers of $q_{\mathcal{O}}$ w.r.t. Σ and D are the answers to q over the canonical interpretation of $\mathcal{O}$ w.r.t. Σ and D (Calvanese et al. 2007c), restricted to those tuples without variables (operator $\downarrow$).

For two queries q_1, q_2 over $\mathcal{O}$, we write $cert_{q_1,\Sigma} \sqsubseteq cert_{q_2,\Sigma}$ if $cert_{q_1,\Sigma}^{D} \subseteq cert_{q_2,\Sigma}^{D}$ for each $\mathcal{S}$-database D. We also write $cert_{q_1,\Sigma} \sqsubset cert_{q_2,\Sigma}$ if $cert_{q_1,\Sigma} \sqsubseteq cert_{q_2,\Sigma}$ and in addition $cert_{q_1,\Sigma}^{D} \subsetneq cert_{q_2,\Sigma}^{D}$ for at least an $\mathcal{S}$-database D. Finally, we say that q_1 and q_2 are *equivalent w.r.t.* Σ, denoted by $cert_{q_1,\Sigma} \equiv cert_{q_2,\Sigma}$, if both $cert_{q_1,\Sigma} \sqsubseteq cert_{q_2,\Sigma}$ and $cert_{q_2,\Sigma} \sqsubseteq cert_{q_1,\Sigma}$ hold, that is, $cert_{q_1,\Sigma}^{D} = cert_{q_2,\Sigma}^{D}$ for each $\mathcal{S}$-database D.

For an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, with $\mathcal{O}$ a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ a GLAV mapping, if $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$, then the UCQ $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ defined in (Cima, Lenzerini, and Poggi 2019) computes exactly the certain answers of $q_{\mathcal{O}}$ w.r.t. Σ and D, for every $\mathcal{S}$-database D, i.e., $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \equiv cert_{q_{\mathcal{O}},\Sigma}$.

***EQL*.** We recall the basis of *EQL* (Calvanese et al. 2007a), a first-order modal language with a single modal operator $\mathbf{K}$[3], used to formalize the *epistemic state* of a DL ontology $\mathcal{O}$, according to the minimal knowledge semantics.

An *epistemic interpretation* for an ontology $\mathcal{O}$ is a pair $\langle E, \mathcal{I} \rangle$, where E is a (possibly infinite) set of FOL interpretations for $\mathcal{O}$, and $\mathcal{I}$ is an interpretation in E. We inductively define when an *EQL* sentence ψ is true in an epistemic interpretation $\langle E, \mathcal{I} \rangle$, written $\langle E, \mathcal{I} \rangle \models \psi$, as follows:

$$
\begin{array}{lll}
\langle E, \mathcal{I} \rangle \models P(\vec{c}) & \text{if} & \mathcal{I} \models P(\vec{c}) \\
\langle E, \mathcal{I} \rangle \models \psi_1 \wedge \psi_2 & \text{if} & \langle E, \mathcal{I} \rangle \models \psi_1 \text{ and } \langle E, \mathcal{I} \rangle \models \psi_2 \\
\langle E, \mathcal{I} \rangle \models \neg \psi & \text{if} & \langle E, \mathcal{I} \rangle \not\models \psi \\
\langle E, \mathcal{I} \rangle \models \exists x. \psi & \text{if} & \langle E, \mathcal{I} \rangle \models \psi_c^x \text{ for some constant } c \\
\langle E, \mathcal{I} \rangle \models \mathbf{K} \psi & \text{if} & \langle E, \mathcal{I}' \rangle \models \psi \text{ for every } \mathcal{I}' \in E,
\end{array}
$$

where ψ, x, and ψ_c^x denote an arbitrary *EQL* formula, a variable, and the *EQL* formula obtained by replacing the variable x with the constant c, respectively.

[1]The logic underpinning `OWL 2 QL` (Motik et al. 2012), i.e., the `OWL 2` profile especially designed for the OBDA scenarios.

[2]Note, however, that our results can be reformulated in a setting where interpretations are sets of objects denoted by object identifiers, as usual in OBDA (Poggi et al. 2008).

[3]In fact, *EQL* includes also equality, which is not considered here.

As in knowledge base scenarios, in OBDA, among the various epistemic interpretations, one is typically interested in the specific ones representing the minimal epistemic state, i.e., the state with minimal knowledge. Formally, let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification, and let D be an $\mathcal{S}$-database. Then, a $\langle \Sigma, D \rangle$-*EQL-interpretation* $\langle E, \mathcal{I} \rangle$ is an epistemic interpretation for which $E = Mod_D(\Sigma)$ (we remind the reader that all interpretations of $Mod_\Sigma(D)$ share the same domain Const). Finally, we say that an *EQL* sentence ψ is *EQL-logically implied* by $\langle \Sigma, D \rangle$, written $\langle \Sigma, D \rangle \models_{EQL} \psi$, if for every $\langle \Sigma, D \rangle$-*EQL*-interpretation $\langle E, \mathcal{I} \rangle$ we have $\langle E, \mathcal{I} \rangle \models \psi$. Given an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, an *EQL* query $q_\mathcal{O}(\vec{t})$ over $\mathcal{O}$ (i.e., an open *EQL* formula whose free variables are the variables appearing in $\vec{t}$), and an $\mathcal{S}$-database D, $cert^D_{q_\mathcal{O},\Sigma}$ is the set of tuples $\vec{c}$ of constants in D such that $\langle \Sigma, D \rangle \models_{EQL} q_\mathcal{O}(\vec{t}/\vec{c})$.

3 Non-Monotonic Abstractions

We start by recalling the *EQL-Lite*(UCQ) query language, and show how queries in such language can be rewritten to compute certain answers over OBDA specifications. Then, we specialise the framework presented in (Cima, Lenzerini, and Poggi 2019) to *EQL-Lite*(UCQ) queries and show that there are cases in which it allows to obtain better approximated s-to-o rewritings, compared with the usual OBDA framework based on UCQs.

3.1 *EQL-Lite*(UCQ) Query Language

EQL-Lite(UCQ), introduced in (Calvanese et al. 2007a)[4], is the FOL query language whose atoms are epistemic formulas of the form $\mathbf{K}\varrho$ where ϱ is a UCQ. Formally, let $\mathcal{O}$ be a DL ontology. An *EQL-Lite*(UCQ) query over $\mathcal{O}$ is a possibly open formula built according to the following syntax:

$$\psi \ ::= \ \mathbf{K}\varrho \mid \exists x.\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \neg\psi$$

where ϱ is the body of a UCQ over $\mathcal{O}$. We call *epistemic atoms* the formulas of the form $\mathbf{K}\varrho$ occurring in queries.

Example 2. *Consider the OBDA specification of Example 1, and the following EQL-Lite(UCQ) queries:*

$q_1(x) \leftarrow \exists y.\mathbf{K}(\mathit{HadContacts}(x, y))$
$q_2(x) \leftarrow \exists y.\mathbf{K}(\mathit{HadContacts}(x, y)) \wedge \neg\mathbf{K}(\mathit{Married}(x, y))$

q_1 retrieves all people for whom at least one person is known they had contacts with, while q_2 restricts the answers to q_1 by requiring that such person is not known to be a spouse.

Answering *EQL-Lite*(UCQ) queries over OBDA systems can be achieved by exploiting a very interesting computational property of the language, i.e., that one can decouple the reasoning needed for answering the epistemic atoms, which can be delegated to the underlying OBDA service for answering UCQs, from the reasoning needed for dealing with the other operators of the whole query.

Formally, let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification, and $q_\mathcal{O}$ be an *EQL-Lite*(UCQ) query over $\mathcal{O}$, whose epistemic atoms are $\mathbf{K}\varrho_1, \ldots, \mathbf{K}\varrho_m$. We denote by $q^{FOL}_\mathcal{O}$ the FOL query over $\mathcal{S}$, obtained from $q_\mathcal{O}$ by replacing each epistemic atom $\mathbf{K}\varrho_i$ with the UCQ $\mathsf{PerfRef}_{\varrho_i,\Sigma}$, whose arity is the number of free variables in ϱ_i. From results of (Calvanese et al. 2007a), it is easy to show the following:

[4] Consistently with what we did for *EQL*, we do not include (in)equalities in *EQL-Lite* (UCQ).

Proposition 1. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification and $q_\mathcal{O}$ an EQL-Lite(UCQ) query over $\mathcal{O}$. Then for every $\mathcal{S}$-database D we have $cert^D_{q_\mathcal{O},\Sigma} = {q^{FOL}_\mathcal{O}}^D$.*

Example 3. *Consider the OBDA specification Σ presented in Example 1, and the following EQL-Lite(UCQ) queries:*

- *the queries q_1 and q_2 presented in Example 2,*
- $\varrho_1(x, y) \leftarrow \mathit{HadContacts}(x, y)$
- $\varrho_2(x, y) \leftarrow \mathit{Married}(x, y)$.

Then, the bodies of q^{FOL}_1 and q^{FOL}_2 are $\exists y.\mathsf{PerfRef}_{\varrho_1(x,y),\Sigma}$, and $\exists y.\mathsf{PerfRef}_{\varrho_1(x,y),\Sigma} \wedge \neg\mathsf{PerfRef}_{\varrho_2(x,y),\Sigma}$, respectively, and therefore they are defined as follows:

$$q^{FOL}_1(x) \leftarrow \exists y.(s_2(x, y) \vee s_3(x, y)).$$
$$q^{FOL}_2(x) \leftarrow \exists y.(s_2(x, y) \vee s_3(x, y)) \wedge \neg s_3(x, y).$$

3.2 *EQL-Lite*(UCQ) S-to-O Rewritings

In this paper, we are interested in the problem of computing abstractions of data services within an OBDA framework admitting a non-monotonic query language over the ontology. Our general goal is to find the query over $\mathcal{O}$ that precisely characterizes the data service, expressed as a query $q_\mathcal{S}$ over $\mathcal{S}$, w.r.t. the underlying OBDA specification Σ. Formally, given an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a query $q_\mathcal{S}$ over $\mathcal{S}$, the *perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$* is the query $q_\mathcal{O}$ over $\mathcal{O}$ such that $Mod_D(\Sigma) \neq \emptyset$ implies $cert^D_{q_\mathcal{O},\Sigma} = q^D_\mathcal{S}$, for every $\mathcal{S}$-database D (Cima, Lenzerini, and Poggi 2019).

As anticipated in the introduction, we next show that considering *EQL-Lite*(UCQ) queries provides more expressivity in finding s-to-o rewritings, compared to UCQs. In particular, the next example shows that there are cases where no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting exists in the class of UCQs, whereas it exists in the class of *EQL-Lite*(UCQ) queries.

Example 4. *Consider the OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ presented in Example 1, and let Σ' be the OBDA specification $\Sigma' = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}' \rangle$, where $\mathcal{M}'$ is obtained from $\mathcal{M}$ by removing the third mapping. Moreover, let $q_\mathcal{S}(x) \leftarrow \exists y.s_2(x, y)$. It is easy to see that no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ'-rewriting of $q_\mathcal{S}$ exists in the class of UCQs. Indeed, by inspecting the mapping, one can see that, since the certain answers of the query $q_\mathcal{O}(x) \leftarrow \exists y.\mathit{HadContacts}(x, y)$ include the values stored in the first component of s_2 but also in s_1 and s_5, such query is too general for exactly characterizing $q_\mathcal{S}$. On the other hand, consider again the EQL-Lite(UCQ) query q_1 discussed in Example 2. One can verify that q_1 is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ'-rewriting of $q_\mathcal{S}$.*

The next example proves that, as for the case of UCQ, there are OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and CQ $q_\mathcal{S}$ for which no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ exists in the class of *EQL-Lite*(UCQ) queries.

Example 5. *Consider again the OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of Example 1, the query $q_\mathcal{S}$ of Example 4, and* *the queries q_1 and q_2 of Example 2. One can see that, because of the presence of the third mapping, no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ exists in the class of EQL-Lite(UCQ) queries. Indeed, on the one hand, q_1 is too general because include the values stored in the first component of both s_2 and s_3, and, on the other hand, q_2 is too specific.*

Thus, following (Cima, Lenzerini, and Poggi 2019), we consider two approximations of the perfect s-to-o rewriting. First, $q_\mathcal{O}$ is a *sound* (respectively, *complete*) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ if for every $\mathcal{S}$-database D, $Mod_D(\Sigma) \neq \emptyset$ implies $cert^D_{q_\mathcal{O},\Sigma} \subseteq q^D_\mathcal{S}$ (resp., $q^D_\mathcal{S} \subseteq cert^D_{q_\mathcal{O},\Sigma}$). Second, if $\mathcal{L}_\mathcal{O}$ is a class of query, then $q_\mathcal{O} \in \mathcal{L}_\mathcal{O}$ is *an $\mathcal{L}_\mathcal{O}$-maximally sound* (respectively, *$\mathcal{L}_\mathcal{O}$-minimally complete*) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ if $q_\mathcal{O}$ is a sound (respectively, complete) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, and no $q' \in \mathcal{L}_\mathcal{O}$ exists such that (i) q' is a sound (respectively, complete) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, (ii) $cert_{q_\mathcal{O},\Sigma} \sqsubseteq cert_{q',\Sigma}$ (resp., $cert_{q',\Sigma} \sqsubseteq cert_{q_\mathcal{O},\Sigma}$), and (iii) there exists an $\mathcal{S}$-database D such that $cert^D_{q_\mathcal{O},\Sigma} \subsetneq cert^D_{q',\Sigma}$ (respectively, $cert^D_{q',\Sigma} \subsetneq cert^D_{q_\mathcal{O},\Sigma}$).

The following example illustrates these notions and shows that *EQL-Lite*(UCQ) allows to find better approximations of s-to-o rewritings w.r.t. the UCQ query language.

Example 6. *Consider the OBDA specification Σ presented in Example 1, the query $q_\mathcal{S}$ presented in Example 4, and the queries q_1 and q_2 presented in Example 2. We have that q_1 and q_2 are an EQL-Lite(UCQ)-minimally complete and an EQL-Lite(UCQ)-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_\mathcal{S}$, respectively. Moreover, one can verify that the UCQ-minimally complete and the UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_\mathcal{S}$ are the queries $q^c_\mathcal{O}(x) \leftarrow \exists y.(\mathit{HadContacts}(x,y))$ and $q^s_\mathcal{O}(x) \leftarrow \bot(x)$, respectively. It is clear that q_1 and q_2 are better approximations of $q_\mathcal{S}$ compared to $q^c_\mathcal{O}$ and $q^s_\mathcal{O}$, respectively.*

In the next section, we carry out a study on the problem of computing minimally complete and maximally sound s-to-o rewritings of data services. Taking into account the inexpressibility results reported in (Cima, Lenzerini, and Poggi 2019, Theorem 16), we will limit our attention to data services expressed as CQJFEs over $\mathcal{S}$ and OBDA specifications with the following characteristics: (i) the ontology language is *DL-Lite*$^-_\mathcal{R}$; (ii) the mapping language is GLAV; (iii) the query language $\mathcal{L}_\mathcal{O}$ used to express s-to-o rewritings is the *EQL-Lite*(UCQ) query language. Sometimes, we will also consider other classes of queries, in particular UCQ; when we omit the specification of $\mathcal{L}_\mathcal{O}$, we implicitly refer to the class of *EQL-Lite*(UCQ) queries.

Notice that, as already observed, for an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ based on *DL-Lite*$^-_\mathcal{R}$, every $\mathcal{S}$-database D is consistent with Σ. Therefore, all our results also hold under the semantics proposed in (Lutz, Marti, and Sabellek 2018), which differs from ours because it defines the perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting $q_\mathcal{O}$ of $q_\mathcal{S}$ to be such that $cert^D_{q_\mathcal{O},\Sigma} = q^D_\mathcal{S}$ for all $\mathcal{S}$-databases D, rather than for all D consistent with Σ.

4 On the Non-Existence of S-to-O Rewritings

In this section, we show that both *EQL-Lite*(UCQ)-minimally complete and *EQL-Lite*(UCQ)-maximally sound s-to-o rewritings are not guaranteed to exist, even in the case of empty ontologies and pure GAV mappings.

We start by focusing on *EQL-Lite*(UCQ)-minimally complete s-to-o rewritings.

Theorem 1. *There is an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a query $q_\mathcal{S}$ over $\mathcal{S}$ such that no EQL-Lite(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ exists.*

Proof (sketch). Consider the OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O} = \emptyset$, and $\mathcal{M}$ is the following mapping:

$$\begin{array}{lll} \{\ s(x) & \rightarrow & C(x), \\ \ s_1(x) \wedge s_2(x) & \rightarrow & C(x), \\ \ s_1(x) & \rightarrow & A(x), \\ \ s_2(x_1) \wedge s_3(x_1,x_2) & \rightarrow & R(x_1,x_2), \\ \ s_1(x_2) \wedge s_5(x_1,x_2) & \rightarrow & R(x_1,x_2), \\ \ s_2(x) \wedge s_4(x) & \rightarrow & B(x), \\ \ s(x_1) \wedge s_1(x_2) \wedge s_2(x_2) & \rightarrow & S(x_1,x_2)\ \}, \end{array}$$

and let $q_\mathcal{S}$ be the boolean CQJFE $q_\mathcal{S}() \leftarrow \exists y.s(y)$.

One can verify that each of the following query is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$:

- $q^0_\mathcal{O}() \leftarrow \mathbf{K}(\exists y.C(y))$,
- $q^1_\mathcal{O}() \leftarrow (\mathbf{K}(\exists y.C(y)) \wedge \neg\mathbf{K}(\exists y_1.A(y_1) \wedge B(y_1))) \vee (\mathbf{K}(\exists y,y'.C(y) \wedge S(y,y')))$,
- $q^n_\mathcal{O}() \leftarrow (\mathbf{K}(\exists y.C(y)) \wedge \neg\mathbf{K}(\exists y_1.A(y_1) \wedge B(y_1)) \wedge \neg\mathbf{K}(\exists y_1,y_2.A(y_1) \wedge R(y_1,y_2) \wedge B(y_2)) \wedge \ldots \wedge \neg\mathbf{K}(\exists y_1,y_2,\ldots,y_n.A(y_1) \wedge R(y_1,y_2) \wedge R(y_2,y_3) \wedge \ldots \wedge R(y_{n-1},y_n) \wedge B(y_n))) \vee (\mathbf{K}(\exists y,y'.C(y) \wedge S(y,y')))$, for every $n \geq 2$.

Moreover, for every pair $i,j \geq 0$ with $i < j$, one can verify that $cert_{q^j_\mathcal{O},\Sigma} \sqsubset cert_{q^i_\mathcal{O},\Sigma}$. It follows that, for every $n \geq 0$, $q^{n+1}_\mathcal{O}$ is a better *EQL-Lite*(UCQ) complete approximation of the $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ compared to $q^n_\mathcal{O}$, and therefore, we conclude that no finite query exists that is an *EQL-Lite*(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$. Intuitively, this is due to the ability of *EQL-Lite*(UCQ) of expressing epistemic forms of negation, which, under certain conditions, allows to exclude non-sound rewritings, while keeping the rewriting complete. $\square$

We now move to *EQL-Lite*(UCQ)-maximally sound s-to-o rewritings. Unfortunately, as for the complete case, such s-to-o rewritings are not guaranteed to exists.

Theorem 2. *There is an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a query $q_\mathcal{S}$ over $\mathcal{S}$ such that no EQL-Lite(UCQ)-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ exists.*

Proof (sketch). Consider the OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O} = \emptyset$, and $\mathcal{M}$ is the following mapping:

$$\begin{array}{lll} \{\ s'(x) & \rightarrow & C(x), \\ \ s(x) & \rightarrow & C(x), \\ \ s_1(x) & \rightarrow & A(x), \\ \ s_2(x_1) \wedge s_3(x_1,x_2) & \rightarrow & R(x_1,x_2), \\ \ s_1(x_2) \wedge s_5(x_1,x_2) & \rightarrow & R(x_1,x_2), \\ \ s_2(x) \wedge s_4(x) & \rightarrow & B(x), \\ \ s(x_1) \wedge s_1(x_2) \wedge s_2(x_2) & \rightarrow & S(x_1,x_2)\ \}, \end{array}$$

and let $q_{\mathcal{S}}$ be the boolean CQJFE $q_{\mathcal{S}}() \leftarrow \exists y.s'(y)$.

One can verify that each of the following query is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$:

- $q^0_{\mathcal{O}}() \leftarrow \mathbf{K}(\exists y.C(y)) \wedge \neg\mathbf{K}(\exists y, y'.C(y) \wedge S(y, y')) \wedge \mathbf{K}(\exists y_0.A(y_0) \wedge B(y_0))$,
- $q^n_{\mathcal{O}}() \leftarrow \mathbf{K}(\exists y.C(y)) \wedge \neg\mathbf{K}(\exists y, y'.C(y) \wedge S(y, y')) \wedge \mathbf{K}((\exists y_0.A(y_0) \wedge B(y_0)) \vee (\exists y_0, y_1.A(y_0) \wedge R(y_0, y_1) \wedge B(y_1)) \vee \ldots \vee (\exists y_0, y_1, \ldots, y_n.A(y_0) \wedge R(y_0, y_1) \wedge R(y_1, y_2) \wedge \ldots \wedge R(y_{n-1}, y_n) \wedge B(y_n)))$, for every $n \geq 1$.

Moreover, for every pair $i, j \geq 0$ with $i < j$, one can verify that $cert_{q^i_{\mathcal{O}},\Sigma} \sqsubset cert_{q^j_{\mathcal{O}},\Sigma}$. It follows that, for every $n > 0$, $q^{n+1}_{\mathcal{O}}$ is a better *EQL-Lite*(UCQ) sound approximation of the $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ compared to $q^n_{\mathcal{O}}$, and therefore, we conclude that no finite query exists that is an *EQL-Lite*(UCQ)-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

In the light of the above inexpressibility results, we now explore the following two alternative restricted scenarios. In the first one (Section 5), we consider one-to-one mapping assertions, i.e., a limitation on GLAV mapping assertions. In this case, we show that both *EQL-Lite*(UCQ)-minimally complete and *EQL-Lite*(UCQ)-maximally sound s-to-o rewritings always exist. In the second one (Section 6), we weaken the $\mathcal{L}_{\mathcal{O}}$ query language by considering s-to-o rewritings expressed in *EQL-Lite*$^-$(UCQ), i.e., a fragment of *EQL-Lite*(UCQ). In this case, while Theorem 2 already proves that *EQL-Lite*$^-$(UCQ)-maximally sound s-to-o rewritings may not exist, we prove that *EQL-Lite*$^-$(UCQ)-minimally complete s-to-o rewritings always exist.

5 The Case of One-To-One Mapping

In this section, we study the problem of computing s-to-o rewritings in *EQL-Lite*(UCQ), for OBDA specifications with one-to-one mapping.

In particular, we provide algorithms to compute *EQL-Lite*(UCQ)-minimally complete and *EQL-Lite*(UCQ)-maximally sound s-to-o rewritings, thus proving that for OBDA specifications with one-to-one mapping, they are guaranteed to exist. We now exploit the following property of one-to-one mapping, which is crucial for the technical treatment of this section.

Lemma 1. *Given a one-to-one mapping $\mathcal{M}$ and an $\mathcal{S}$-database D, the chase of D w.r.t. $\mathcal{M}$ can be computed by computing the union of the chase of each fact in D, i.e., $\mathcal{M}(D) = \bigcup_{\alpha \in D} \mathcal{M}(\alpha)$.*

Let us first focus on *EQL-Lite*(UCQ)-minimally complete rewritings, and present Algorithm 1. Roughly speaking, the algorithm computes an *EQL-Lite*(UCQ) query $q_{\mathcal{O}}$ by first chasing (the incomplete $\mathcal{S}$-database associated to) $q_{\mathcal{S}}$ w.r.t. $\mathcal{M}$, using $\top$ to bind possible distinguished variables of $q_{\mathcal{S}}$ not involved in $\mathcal{M}(q_{\mathcal{S}})$, and then exploiting the epistemic operator to bind existential variables coming from $q_{\mathcal{S}}$. Note, in particular, that the latter is achieved by pushing the subset $\mathscr{Y}$ of the existential variables Y of $q_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$ inside the $\mathbf{K}$ operator. Finally, Z denotes the set of fresh existential variables introduced by the chase.

Algorithm 1 MinimallyComplete

Input:
OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$
CQJFE $q_{\mathcal{S}}(\vec{t}) \leftarrow \phi(\vec{x}, Y)$
Output:
EQL-Lite(UCQ) query $q_{\mathcal{O}}(\vec{t})$ over $\mathcal{O}$

1: **return** $q_{\mathcal{O}}(\vec{t}) \leftarrow \exists \mathscr{Y}.\mathbf{K}(\exists Z.\mathcal{M}(q_{\mathcal{S}}) \wedge \top(\vec{x}))$, where $\mathscr{Y} \subseteq Y$ is the subset of existential variables of $q_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$, while Z are fresh variables

Observe that the running time of Algorithm 1 is independent of both the size of $\mathcal{O}$ and $\mathcal{S}$, and polynomial in the size of all inputs of the problem (indeed, for one-to-one mapping there is no query to evaluate when applying the chase). The following example illustrates the algorithm.

Example 7. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be such that $\mathcal{O} = \{ A_1 \sqsubseteq \exists P_1, A_2 \sqsubseteq \exists P_2 \}$, $\mathcal{S} = \{ s_1, s_2, s_3, s_4, s_5 \}$, and $\mathcal{M}$ is the following mapping:*

$$\begin{array}{lll} \{ s_1(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\ \exists y_1, y_2.s_2(x, y_1, y_2) & \rightarrow & A_1(x), \\ \exists y.s_3(x, y) & \rightarrow & \exists z.P_2(x, z), \\ \exists y.s_4(x, y) & \rightarrow & A_2(x), \\ \exists y.s_5(x_1, x_2, y) & \rightarrow & P_3(x_1, x_2), \\ \exists y_1, y_2.s_5(x, y_1, y_2) & \rightarrow & A_3(x) \} \end{array}$$

For the CQJFE $q_{\mathcal{S}}(x) \leftarrow \exists y_1, y_2, y_3.s_1(x, y_1) \wedge s_3(x, y_2) \wedge s_5(c_1, c_2, y_3)$, one can easily verify that Algorithm 1 returns the query $q_{\mathcal{O}}(x) \leftarrow \exists y_1.\mathbf{K}(\exists z_1.P_1(x, y_1) \wedge P_2(x, z_1) \wedge P_3(c_1, c_2) \wedge A_3(c_1))$, which is the unique (up to equivalence w.r.t. Σ) EQL-Lite(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Note that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ because of the axiom $A_2 \sqsubseteq \exists P_2$ and the mapping $s_4(x, y) \rightarrow A_2(x)$. Without at least one of them, $q_{\mathcal{O}}$ is also a sound (and therefore a perfect) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

It is easy to see that the queries of the same shape of those returned by Algorithm 1 enjoy of the following property, which holds even when considering GLAV mappings rather than one-to-one mappings.

Lemma 2. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification and D be an $\mathcal{S}$-database. For a boolean query of the form $q_{\mathcal{O}}() \leftarrow \exists \mathscr{Y}.\mathbf{K}(\phi)$, where ϕ is a CQ, we have $cert^D_{q_{\mathcal{O}},\Sigma}$ true iff there is a function h from the set of terms occurring in ϕ to the set of terms occurring in $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ (also called homomorphism from $q_{\mathcal{O}}()$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$) such that* (i) *$h(c) = c$, for each constant c;* (ii) *$h(y)$ is a constant, for each $y \in \mathscr{Y}$;* (iii) *$h(\phi) \subseteq \mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, where $h(\phi)$ is the image of the set of atoms in ϕ under h.*

By exploiting Lemmata 1 and 2, we are now ready to prove the following theorem.

Theorem 3. *Algorithm 1 terminates and computes the unique (up to equivalence w.r.t. Σ) EQL-Lite(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Termination is trivial, and therefore not discussed.

As for the correctness, observe that the possibly introduced fresh existential variables Z of $q_{\mathcal{O}}$ do not appear outside the epistemic operator $\mathbf{K}$, and therefore the query $q_{\mathcal{S}}$ corresponds to, or it is contained in, a disjunct of $q_{\mathcal{O}}^{FOL}$ (note that, since in $q_{\mathcal{O}}$ the "$\neg$" operator does not occur, its perfect ontology-to-source rewriting $q_{\mathcal{O}}^{FOL}$ can be always expressed as an equivalent UCQ). But then, $q_{\mathcal{S}} \sqsubseteq q_{\mathcal{O}}^{FOL}$ implies that $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) *EQL-Lite*(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. To this aim, we prove by contradiction that each *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Suppose that for an *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ we have $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$. So, there is an $\mathcal{S}$-database D for which $cert^{D}_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^{D}_{q'_{\mathcal{O}},\Sigma}$, i.e, $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$ and $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$ for a tuple of constants $\vec{c}$. We now exhibit an $\mathcal{S}$-database D' such that $\vec{c} \in q_{\mathcal{S}}^{D'}$ and $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$.

Since $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$, by Lemma 2 there is an homomorphism h from $q_{\mathcal{O}}(\vec{t}/\vec{c})$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ such that $h(y)$ is a constant occurring in D, for each $y \in \mathscr{Y}$. Let h' be the function extending h by assigning a fresh constant c_y (resp., c_x) to each existential variable $y \in Y \setminus \mathscr{Y}$ (resp., distinguished variable x not appearing in $\mathcal{M}(q_{\mathcal{S}})$) of $q_{\mathcal{S}}$. Let $h'(q_{\mathcal{S}})$ and $h(q_{\mathcal{O}}) \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ be the set of facts corresponding to the image of $q_{\mathcal{S}}$ under h' and the set of atoms corresponding to the image of $q_{\mathcal{O}}$ under h, respectively.

Consider each ground atom $\alpha \in h'(q_{\mathcal{S}})$ and its chase $\mathcal{M}(\alpha)$. Since the left-hand side of each mapping assertion in $\mathcal{M}$ is an atom without constants or repeated variables, and since $h(q_{\mathcal{O}}) \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ with $h(y)$ being a constant for each variable $y \in \mathscr{Y}$, it is not hard to see that there always exists a homomorphism from $\mathcal{M}(\alpha)$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$. Due to Lemma 1, moreover, it follows that there exists a homomorphism from $\mathcal{M}(h(q_{\mathcal{S}}))$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$.

Consider now the $\mathcal{S}$-database $D' = D \cup h'(q_{\mathcal{S}})$. Obviously, $\vec{c} \in q_{\mathcal{S}}^{D'}$ trivially holds. Furthermore, due to Lemma 1 and the above considerations, we have that $\mathcal{M}(D') = \mathcal{M}(D) \cup \mathcal{M}(h'(q_{\mathcal{S}}))$, in which there exists a homomorphism from $\mathcal{M}(h'(q_{\mathcal{S}}))$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$. Therefore, we easily derive that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent. Clearly, since $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$, we derive $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$. Thus, $\vec{c} \in q_{\mathcal{S}}^{D'}$ and $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ imply that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. □

We now move to the problem of computing *EQL-Lite*(UCQ)-maximally sound s-to-o rewritings, and present Algorithm 2. In a nutshell, the algorithm starts by checking whether there is some distinguished variable of $q_{\mathcal{S}}$ not appearing in $\mathcal{M}(q_{\mathcal{S}})$, and if this is the case, then it returns the query $\bot$. Otherwise, it computes the *EQL-Lite*(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and for each disjunct $q_{\mathcal{S}}^{i}$ in its reformulation $q_{\mathcal{O}}^{FOL}$ over $\mathcal{S}$ such that $q_{\mathcal{S}}^{i} \not\sqsubseteq q_{\mathcal{S}}$, it adds a conjunct that is the negation of the *EQL-Lite*(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}^{i}$. Intuitively, by doing so it prevents the rewriting to return answers that are not in $q_{\mathcal{S}}$. It is worth noting that Algorithm 2 exploits the property that, in step 5, since in the *EQL-Lite*(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ the "$\neg$" operator does not occur, its perfect ontology-to-source rewriting $q_{\mathcal{O}}^{FOL}$ can be always expressed as an equivalent UCQ over $\mathcal{S}$.

Algorithm 2 MaximallySound

Input:
OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$
CQJFE $q_{\mathcal{S}}(\vec{t}) \leftarrow \phi(\vec{x}, Y)$
Output:
EQL-Lite(UCQ) query $q_{\mathcal{O}}(\vec{t})$ over $\mathcal{O}$

1: **if** there is a variable in $\vec{t}$ not occurring in $\mathcal{M}(q_{\mathcal{S}})$ **then**
2:     **return** $q_{\mathcal{O}}(\vec{t}) \leftarrow \bot(\vec{x})$
3: **end if**
4: Let $q_{\mathcal{O}}(\vec{t}) :=$ MinimallyComplete$(\Sigma, q_{\mathcal{S}})$
5: **for** $q_{\mathcal{S}}^{i} \in q_{\mathcal{O}}^{FOL}$ such that $q_{\mathcal{S}}^{i} \not\sqsubseteq q_{\mathcal{S}}$ **do**
6:     $q_{\mathcal{O}}(\vec{t}) := q_{\mathcal{O}}(\vec{t}) \wedge \neg\exists \mathscr{Y}_i.\mathbf{K}(\exists Z_i.\mathcal{M}(q_{\mathcal{S}}^{i}))$, where $\mathscr{Y}_i$ is the subset of the existential variables of $q_{\mathcal{S}}^{i}$ occurring in $\mathcal{M}(q_{\mathcal{S}}^{i})$ and not occurring in $\mathcal{M}(q_{\mathcal{S}})$, while Z_i are fresh variables
7: **end for**
8: **return** $q_{\mathcal{O}}(\vec{t})$

As for the running time of Algorithm 2, we observe that it is independent of the size of $\mathcal{S}$, polynomial in the size of both $\mathcal{O}$ and $\mathcal{M}$, and exponential in the size of $q_{\mathcal{S}}$. This latter is due to the fact that $q_{\mathcal{O}}^{FOL}$ in step 5 is in general the union of an exponential number of CQs with respect to the number of atoms occurring in $q_{\mathcal{S}}$, and also due to the various containment check of CQs. Finally, note that the overall running time is exponential in the size of the input. The following example illustrates the algorithm.

Example 8. *Let* $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ *be such that* $\mathcal{O} = \{A_1 \sqsubseteq \exists P_1, A_2 \sqsubseteq \exists P_2\}$, $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5, s_6\ \}$, *and* $\mathcal{M}$ *is the following mapping:*

$$
\begin{array}{lll}
\{\ s_1(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\
\ \ \exists y.s_2(x, y) & \rightarrow & \exists z.P_2(x, z), \\
\ \ \exists y.s_3(x_1, x_2, y) & \rightarrow & P_2(x_1, x_2), \\
\ \ \exists y_1, y_2.s_4(x, y_1, y_2) & \rightarrow & A_2(x), \\
\ \ \exists y_1, y_2.s_4(y_1, y_2, x) & \rightarrow & \exists z.P_3(x, z), \\
\ \ s_5(x) & \rightarrow & \exists z.P_1(x, z), \\
\ \ s_6(x) & \rightarrow & A_1(x)\ \}
\end{array}
$$

and let $q_{\mathcal{S}}(x) \leftarrow \exists y_1, y_2.s_1(x, y_1) \wedge s_2(x, y_2)$. *Algorithm 2 first sets* $q_{\mathcal{O}}$ *equal to the EQL-Lite*(*UCQ*)*-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ *rewriting of* $q_{\mathcal{S}}$, *i.e.,* $q_{\mathcal{O}}(x) \leftarrow \exists y_1.\mathbf{K}(\exists z_1.P_1(x, y_1) \wedge P_2(x, z_1))$. *Then, it computes the perfect ontology-to-source rewriting* $q_{\mathcal{O}}^{FOL}$ *of* $q_{\mathcal{O}}$, *which is the union of the following CQs:*

- $q_{\mathcal{S}}^{1}(x) \leftarrow \exists y_1, y_2^1.s_1(x, y_1) \wedge s_2(x, y_2^1)$,
- $q_{\mathcal{S}}^{2}(x) \leftarrow \exists y_1, y_2^2, y_3^2.s_1(x, y_1) \wedge s_3(x, y_2^2, y_3^2)$,

- $q^3_{\mathcal{S}}(x) \leftarrow \exists y_1, y^3_2, y^3_3. s_1(x, y_1) \wedge s_4(x, y^3_2, y^3_3)$.

While $q^1_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$, it is easy to see that $q^i_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ for both $i = 2$ and $i = 3$. Thus, the algorithm returns the query:

$$\begin{aligned} q_{\mathcal{O}}(x) \leftarrow \exists y_1.\mathbf{K}(\exists z_1. P_1(x, y_1) \wedge P_2(x, z_1)) \\ \wedge\neg\exists y^2_2.\mathbf{K}(P_1(x, y_1) \wedge P_2(x, y^2_2)) \\ \wedge\neg\exists y^3_3.\mathbf{K}(\exists z_2. P_1(x, y_1) \wedge A_2(x) \wedge P_3(y^3_3, z_2)), \end{aligned}$$

which is the unique (up to equivalence w.r.t. Σ) EQL-Lite(UCQ)-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, but not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Theorem 4. *Algorithm 2 terminates and computes the unique (up to equivalence w.r.t. Σ) EQL-Lite(UCQ)-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Termination is trivial, and therefore not discussed.

As for the correctness, we first prove that the query $q_{\mathcal{O}}$ returned by the algorithm is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Let D be any $\mathcal{S}$-database and $\vec{c}$ be any tuple of constants in D such that $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$. By construction of $q_{\mathcal{O}}$ and due to Theorem 3, it follows that $\vec{c} \in cert^D_{q^c_{\mathcal{O}},\Sigma}$, where $q^c_{\mathcal{O}}(\vec{t}) \leftarrow \exists\mathscr{Y}.\mathbf{K}(\exists Z.\mathcal{M}(q_{\mathcal{S}}))$. Therefore, there is at least a disjunct $q'_{\mathcal{S}}$ of ${q^c_{\mathcal{O}}}^{FOL}$ for which $\vec{c} \in {q'_{\mathcal{S}}}^D$. Here there are two possible cases: either $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$, or $q'_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$.

In the former case, we have that $q_{\mathcal{O}}$ contains a conjunction of the form $\neg\exists\mathscr{Y}'.\mathbf{K}(\exists Z'.\mathcal{M}(q'_{\mathcal{S}}))$. Observe that, since $q'_{\mathcal{S}}$ is a disjunct of ${q^c_{\mathcal{O}}}^{FOL}$, each distinguished variable of $q'_{\mathcal{S}}$ appears in $\mathcal{M}(q'_{\mathcal{S}})$. Due to Theorem 3, it follows that $\exists\mathscr{Y}'.\mathbf{K}(\exists Z'.\mathcal{M}(q'_{\mathcal{S}}))$ is actually the body of the query corresponding to a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q'_{\mathcal{S}}$. Thus, since $\vec{c} \in {q'_{\mathcal{S}}}^D$, the conjunction $\exists\mathscr{Y}'.\mathbf{K}(\exists Z'.\mathcal{M}(q'_{\mathcal{S}}))$ is as well true for the tuple of constants $\vec{c}$, which leads to a contradiction to the fact that $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$. As for the latter case, since $\vec{c} \in {q'_{\mathcal{S}}}^D$ and $q'_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$, by definition of query containment we derive $\vec{c} \in q^D_{\mathcal{S}}$. Therefore, we conclude that $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) *EQL-Lite*(UCQ)-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. To this aim, we prove by contradiction that each *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ such that $cert_{q'_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Suppose that for an *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ we have $cert_{q'_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$. So, there is an $\mathcal{S}$-database D for which $cert^D_{q'_{\mathcal{O}},\Sigma} \not\sqsubseteq cert^D_{q_{\mathcal{O}},\Sigma}$, i.e, $\vec{c} \in cert^D_{q'_{\mathcal{O}},\Sigma}$ and $\vec{c} \notin cert^D_{q_{\mathcal{O}},\Sigma}$ for a tuple of constants $\vec{c}$. If $\vec{c} \notin q^D_{\mathcal{S}}$, then $q'_{\mathcal{O}}$ is trivially not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Therefore, we assume that $\vec{c} \in q^D_{\mathcal{S}}$ and exhibit an $\mathcal{S}$-database D' such that $\vec{c} \notin q^{D'}_{\mathcal{S}}$ and $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},\Sigma}$.

Consider each homomorphism h from $q_{\mathcal{S}}(\vec{t}/\vec{c})$ to D, and let $h(q_{\mathcal{S}})$ denote the set of facts corresponding to the image of $q_{\mathcal{S}}$ under h. Clearly, due to Theorem 3 and Lemma 2, the conjunction $\exists\mathscr{Y}.\mathbf{K}(\exists Z.\mathcal{M}(q_{\mathcal{S}}))$ of $q_{\mathcal{O}}$ is true under h with $h(y)$ being a constant for each variable $y \in \mathscr{Y}$. Since by assumption $\vec{c} \notin cert^D_{q_{\mathcal{O}},\Sigma}$, by construction of $q_{\mathcal{O}}$ there is at least a conjunction $\exists\mathscr{Y}_i.\mathbf{K}(\exists Z_i.\mathcal{M}(q^i_{\mathcal{S}}))$ in $q_{\mathcal{O}}$ true under h, where $q^i_{\mathcal{S}}$ is a disjunct of ${q^c_{\mathcal{O}}}^{FOL}$ such that $q^i_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$. Let h_i be the homomorphism from $\exists\mathscr{Y}_i.\mathbf{K}(\exists Z_i.\mathcal{M}(q^i_{\mathcal{S}}))$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ such that $h_i(y) = h(y)$ for each existential variable $y \in \mathscr{Y}$ occurring also in $\mathcal{M}(q^i_{\mathcal{S}})$, and $h_i(x) = h(x)$ for each distinguished variable $x \in \vec{t}$. We denote by h' be the function extending h by assigning a different fresh constant c_y to each existential variable $y \notin \mathscr{Y}_i \cup \mathscr{Y}$ of $q^i_{\mathcal{S}}$.

With similar arguments as those in the proof of Theorem 3, we derive that there exists a homomorphism from $\mathcal{M}(h'(q^i_{\mathcal{S}}))$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, where $h'(q^i_{\mathcal{S}})$ denotes the set of facts corresponding to the image of $q^i_{\mathcal{S}}(\vec{t}/\vec{c})$ under h'. Furthermore, since $q^i_{\mathcal{S}}$ is a disjunct of ${q^c_{\mathcal{O}}}^{FOL}$, and since $\mathcal{M}$ is a one-to-one mapping and $\mathcal{O}$ is a *DL-Lite*$^-_{\mathcal{R}}$ ontology, it is straightforward to verify that all the possible logical consequences of the set of facts $h(q_{\mathcal{S}})$ over the alphabet of $\mathcal{O}$ are a subset of the logical consequences of the set of facts $h'(q^i_{\mathcal{S}})$.

Due to the above considerations, we derive that the $\mathcal{S}$-database $D_h = (D \setminus h(q_{\mathcal{S}})) \cup h'(q^i_{\mathcal{S}})$ is such that $\mathcal{C}^{\mathcal{M}(D_h)}_{\mathcal{O}}$ and $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ are homomorphically equivalent. By iterating the above process for each homomorphism h from $q_{\mathcal{S}}(\vec{t}/\vec{c})$ to D, since $q_{\mathcal{S}}$ is a CQJFE, it is not hard to see that we obtain an $\mathcal{S}$-database D' such that (*i*) there is no more homomorphism from $q_{\mathcal{S}}(\vec{t}/\vec{c})$ to D' (thus, $\vec{c} \notin q^{D'}_{\mathcal{S}}$) and (*ii*) $\mathcal{C}^{\mathcal{M}(D')}_{\mathcal{O}}$ and $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ are homomorphically equivalent. Clearly, since $\vec{c} \in cert^D_{q'_{\mathcal{O}},\Sigma}$, we have $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ as well. Thus, $\vec{c} \notin q^{D'}_{\mathcal{S}}$ and $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ imply that $q'_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. □

6 The Case of Restricted Query Language

In this section, we explore the possibility of expressing s-to-o rewritings in a fragment of the target query language *EQL-Lite*(UCQ) considered so far, which is still non-monotonic. In particular, while the proof of Theorem 2 shows that epistemic negation already suffices to prevent the existence of s-to-o rewritings (even with empty ontologies and pure GAV mappings), the proof of Theorem 1 suggests to remove the union (i.e., the rule $\psi ::= \psi_1 \vee \psi_2$) from the syntax of *EQL-Lite*(UCQ), in order to get an $\mathcal{L}_{\mathcal{O}}$ ensuring the existence of $\mathcal{L}_{\mathcal{O}}$-minimally complete s-to-o rewritings.

Thus, based on the observation that union can be expressed by means of conjunction and nested negation, we next consider the fragment of *EQL-Lite*(UCQ) where both nested negation and union operators are disallowed. Formally, an *EQL-Lite*$^-$(UCQ) query is a possibly open formula built according to the following syntax, where ϱ is the body of a UCQ over $\mathcal{O}$:

$$\begin{aligned} \psi &::= \mathbf{K}\varrho \mid \exists y.\psi \mid \psi_1 \wedge \psi_2 \mid \neg\delta \\ \delta &::= \mathbf{K}\varrho \mid \exists y.\delta \end{aligned}$$

The following example illustrates the *EQL-Lite*$^-$(UCQ) query language.

Example 9. *The queries $q^n_{\mathcal{O}}$ for $n \geq 0$ used in the proof of Theorem 2, as well as the queries q_1 and q_2 of Example 2, are EQL-Lite$^-$(UCQ) queries. On the contrary, the query $q^n_{\mathcal{O}}() \leftarrow \mathbf{K}(\varrho) \wedge \neg\mathbf{K}(\varrho_0) \wedge \neg\mathbf{K}(\varrho_1) \wedge \ldots \wedge \neg\mathbf{K}(\varrho_n) \vee \mathbf{K}(\varrho')$, introduced in the proof of Theorem 1, is not, for any $n \geq 1$.*

Note that both Algorithms 1 and 2 return an *EQL-Lite*$^-$(UCQ) query. Therefore, Theorem 3 and Theorem 4 actually show that, in the case of one-to-one mapping, *EQL-Lite*$^-$(UCQ) is sufficient for expressing all kinds of source-to-ontology rewritings. Furthermore, since the queries involved in the proof of Theorem 2 are *EQL-Lite*$^-$(UCQ) queries, such proof in fact shows a stronger result: even *EQL-Lite*$^-$(UCQ)-maximally sound s-to-o rewritings are not guaranteed to exist.

Thus, it remains to study the case of *EQL-Lite*$^-$(UCQ)-minimally complete s-to-o rewritings. We now prove that Algorithm 1 can be used to compute *EQL-Lite*$^-$(UCQ)-minimally complete s-to-o rewritings even for OBDA specifications with GLAV mapping[5]. Before presenting the result, we illustrate its application with an example.

Example 10. *Let* $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ *be such that* $\mathcal{O} = \emptyset$, $\mathcal{S} = \{ s_1, s_2, s_3 \}$, *and* $\mathcal{M}$ *is the following:*

$$\{ \exists y. s_1(x_1, x_2) \wedge s_2(x_2, y) \rightarrow \exists z. P(x_1, z) \wedge P(z, x_2),$$
$$\exists y, y_1, y_2. s_2(x, y) \wedge s_3(y, y_1, y_2) \rightarrow \exists z. P'(x, z),$$
$$\exists y. s_3(y, x, b) \rightarrow A(x) \}$$

For the CQ $q_{\mathcal{S}}(x_1, x_2) \leftarrow \exists y_1, y_2. s_1(x_1, y_1) \wedge s_2(y_1, y_2) \wedge s_3(y_2, x_2, a)$, *one can easily verify that Algorithm 1 returns the query* $q_{\mathcal{O}}(x_1, x_2) \leftarrow \exists y_1. \mathbf{K}(\exists z_1, z_2. P(x_1, z_1) \wedge P(z_1, y_1) \wedge P'(y_1, z_2) \wedge \top(x_2))$, *which is the unique (up to equivalence w.r.t.* Σ*) EQL-Lite*$^-$*(UCQ)-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting of* $q_{\mathcal{S}}$.

Theorem 5. *Algorithm 1 terminates and computes the unique (up to equivalence w.r.t.* Σ*) EQL-Lite*$^-$*(UCQ)-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting of* $q_{\mathcal{S}}$.

Proof. Termination trivially follows from the termination of the chase w.r.t. a GLAV mapping (Fagin et al. 2005).

For the correctness, as in the proof of Theorem 3, we have $q_{\mathcal{S}} \sqsubseteq q_{\mathcal{O}}^{FOL}$, where $q_{\mathcal{O}}$ is the query returned by the algorithm. Thus, $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is the unique (up to equivalence w.r.t. Σ) *EQL-Lite*$^-$(UCQ)-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. To this aim, we prove by contradiction that each *EQL-Lite*$^-$(UCQ) query $q'_{\mathcal{O}}$ such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Suppose the existence of an *EQL-Lite*$^-$(UCQ) query $q'_{\mathcal{O}}$ such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$. So, there is an $\mathcal{S}$-database D for which $cert^D_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^D_{q'_{\mathcal{O}},\Sigma}$, i.e, $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$ and $\vec{c} \notin cert^D_{q'_{\mathcal{O}},\Sigma}$ for a tuple of constants $\vec{c}$. Here, there are two cases: either $q'_{\mathcal{O}}$ contains a negated part $\neg\delta$, i.e., $q'_{\mathcal{O}}$ is of the form $q'_{\mathcal{O}}(\vec{t'}) \leftarrow \exists Y'. \mathbf{K}(\varrho_1) \wedge \ldots \wedge \mathbf{K}(\varrho_n) \wedge \neg(\exists W'. \mathbf{K}(\varrho_{n+1}) \wedge \ldots \wedge \mathbf{K}(\varrho_m))$ with ϱ_i being a UCQ for each $i \in [1, m]$, or $q'_{\mathcal{O}}$ is a query of the form $q'_{\mathcal{O}}(\vec{t'}) \leftarrow \exists Y'. \mathbf{K}(\varrho_1) \wedge \ldots \wedge \mathbf{K}(\varrho_n)$ with ϱ_i being a UCQ for each $i \in [1, n]$.

In the former case, consider a source database $D' \supseteq D$ in which each relation $s \in \mathcal{S}$ contains all possible tuples of constants appearing in D, $q_{\mathcal{S}}$, and $q'_{\mathcal{O}}$ of the same arity of s. It is straightforward to verify that $\vec{c} \in q_{\mathcal{S}}^{D'}$ and $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$, and thus $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

In the latter case, consider $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$. Since $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$, by Lemma 2 there is an homomorphism h from $q_{\mathcal{O}}(\vec{t}/\vec{c})$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ such that $h(y)$ is a constant occurring in D, for each $y \in \mathscr{Y}$. Consider now $\vec{t} = (t_1, \ldots, t_n)$ and the $\mathcal{S}$-database D' corresponding to the *freezing* of $q_{\mathcal{S}}(\vec{t}) \leftarrow \phi(\vec{x}, Y)$, i.e., the set of facts over $\mathcal{S}$ appearing in $\phi(\vec{x}, Y)$ where each existential variable y (resp., each distinguished variable x) is replaced by a different fresh constant c_y (resp., c_x). Let $\vec{c'} = (c'_1, \ldots, c'_n)$ be the tuple of constants where $c'_i = c_x$ if $t_i = x$, and $c'_i = c$ if $t_i = c$, for $i \in [1, n]$. Obviously, $\vec{c'} \in q_{\mathcal{S}}^{D'}$ trivially holds. Moreover, by construction of $q_{\mathcal{O}}$ and the fact that $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$, there is a function f from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ such that each constant of $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ is mapped to a constant of $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$. More specifically, there is a function f such that $f(c_y) = h(y)$ (resp., $f(c_x) = h(x)$, $f(c) = h(c) = c$) for each existential variable $y \in \mathscr{Y}$ (resp., distinguished variable x, constant c) of $q_{\mathcal{S}}$. Notice that $f(\vec{c'}) = \vec{c}$. Due to the above function f, and the fact that from Lemma 2 there is no homomorphism h from $q'_{\mathcal{O}}(\vec{t'}/\vec{c})$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ such that $h(y')$ is a constant for each $y' \in Y'$ (recall that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},\Sigma}$ and Y' are the set of existential variables of $q'_{\mathcal{O}}$ appearing outside the epistemic operator $\mathbf{K}$), we easily derive that there is no homomorphism h' from $q'_{\mathcal{O}}(\vec{t'}/\vec{c'})$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ such that $h'(y')$ is a constant for each $y' \in Y'$ (otherwise, the function $f \circ h'$ would be a homomorphism from $q'_{\mathcal{O}}(\vec{t'}/\vec{c})$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ with $f(h'(y'))$ being a constant for each $y' \in Y'$, and, due to Lemma 2, this would be a contradiction to the fact that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},\Sigma}$). It follows that $\vec{c'} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$. Thus, $\vec{c'} \in q_{\mathcal{S}}^{D'}$ and $\vec{c'} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ imply that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

7 Conclusions

We have presented the first work on using non-monotonic languages for expressing abstractions of data services in OBDA. The concepts and the results presented in this paper show that non-monotonicity is indeed an interesting feature to consider when computing the s-to-o rewritings of a source query. There are many interesting directions for continuing the work presented here. For instance, we aim at singling out more interesting cases where abstractions expressed in *EQL-Lite*(UCQ) can be actually computed. We observe that the proofs reported in Sections 5 and 6 already show that the algorithm for computing minimally complete rewritings in the two restricted settings also work when $q_{\mathcal{S}}$ is a CQ (rather than a CQJFE). On the other hand, it is still open whether a maximally sound rewriting for a CQ always exists in the case of one-to-one mapping. Another notable direction is to study the existence problem in the general setting: check whether an *EQL-Lite*(UCQ) s-to-o rewriting (perfect, or approximated) exists for a given OBDA specification and a given source query expressed as a CQ. Finally, we believe that the notion of abstraction studied here can be of interest in other data interoperation architectures, such as peer-to-peer data integration (Calvanese et al. 2003).

[5]In this case the running time of the algorithm becomes exponential in the size of $\mathcal{M}$ but remains polynomial in the size of $q_{\mathcal{S}}$.

Acknowlegments

This work has been supported by Sapienza under the project "PRE-O-PRE" and by MIUR, under the SIR project "MODEUS" - grant n. RBSI14TQHQ, and under the PRIN 2017 project "HOPE" (prot. 2017MMJJRE).

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison Wesley Publ. Co.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Bienvenu, M. 2016. Ontology-mediated query answering: Harnessing knowledge to get more from data. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 4058–4061.

Calì, A.; Gottlob, G.; and Kifer, M. 2013. Taming the infinite chase: Query answering under expressive relational constraints. *Journal of Artificial Intelligence Research* 48:115–174.

Calvanese, D.; Damaggio, E.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2003. Semantic data integration in P2P systems. In *Proc. of the Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2003)*.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Rosati, R.; and Vetere, G. 2004. *DL-Lite*: Practical reasoning for rich DLs. In *Proceedings of the Seventeenth International Workshop on Description Logic (DL 2004)*, volume 104 of *CEUR Electronic Workshop Proceedings,* http://ceur-ws.org/.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007a. EQL-Lite: Effective first-order query processing in description logics. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 274–279.

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

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Vardi, M. Y. 2007c. View-based query processing: On the relationship between rewriting, answering and losslessness. *Theoretical Computer Science* 371(3):169–182.

Cima, G.; Lenzerini, M.; and Poggi, A. 2019. Semantic characterization of data services through ontologies. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, 1647–1653.

Cima, G. 2017. Preliminary results on ontology-based open data publishing. In *Proceedings of the Thirtieth International Workshop on Description Logics (DL 2017)*, volume 1879 of *CEUR Electronic Workshop Proceedings,* http://ceur-ws.org/.

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

Fagin, R.; Kolaitis, P. G.; Miller, R. J.; and Popa, L. 2005. Data exchange: Semantics and query answering. *Theoretical Computer Science* 336(1):89–124.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proceedings of the Twenty-First ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2002)*, 233–246.

Lenzerini, M. 2018. Managing data through the lens of an ontology. *AI Magazine* 39(2):65–74.

Levesque, H. J., and Lakemeyer, G. 2001. *The Logic of Knowledge Bases*. The MIT Press.

Levesque, H. J. 1984. Foundations of a functional approach to knowledge representation. *Artificial Intelligence* 23:155–212.

Lutz, C.; Marti, J.; and Sabellek, L. 2018. Query expressibility and verification in ontology-based data access. In *Proceedings of the Sixteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2018)*, 389–398.

Motik, B.; Cuenca Grau, B.; Horrocks, I.; Wu, Z.; Fokoue, A.; and Lutz, C. 2012. OWL 2 Web Ontology Language profiles (second edition). W3C Recommendation, World Wide Web Consortium. Available at http://www.w3.org/TR/owl2-profiles/.

Ortiz, M. 2018. Improving data management using domain knowledge. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 5709–5713.

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

Reiter, R. 1990. What should a database know? *Journal of Logic Programming* 14:127–153.

Xiao, G.; Calvanese, D.; Kontchakov, R.; Lembo, D.; Poggi, A.; Rosati, R.; and Zakharyaschev, M. 2018. Ontology-based data access: A survey. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, 5511–5519.