Semantic characterization of data services through ontologies*

Gianluca Cima[1] , **Maurizio Lenzerini**[1] , **Antonella Poggi**[1,2]

[1]Dipartimento di Ingegneria Informatica, Automatica e Gestionale
[2]Dipartimento di Lettere e Culture Moderne
Sapienza Università di Roma

{cima, lenzerini, poggi}@diag.uniroma1.it

Abstract

We study the problem of associating formal semantic descriptions to data services. We base our proposal on the Ontology-based Data Access paradigm, where a domain ontology is used to provide a semantic layer mapped to the data sources of an organization. The basic idea is to explain the semantics of a data service in terms of a query over the ontology. We illustrate a formal framework for this problem, based on the notion of source-to-ontology (s-to-o) rewriting, which comes in three variants, called sound, complete and perfect, respectively. We present a thorough complexity analysis of two computational problems, namely verification (checking whether a query is an s-to-o rewriting of a given data service), and computation (computing an s-to-o rewriting of a data service).

1 Introduction

The architecture of many modern Information Systems is based on data services [Zheng *et al.*, 2013], i.e., services deployed on top of data stores, other services, and/or applications to encapsulate a wide range of data-centric operations. Data services are also used to handle the programming logic for data virtualization in a cloud-hosted data storage infrastructure, so as to delegate most administrative tasks to the cloud infrastructure, and effectively realizing the idea of Data-As-A-Service. Furthermore, since big data, which is now imperative in many contexts, may be obtuse, disorganized, and may not make much sense to most potential users, in order to get value from them, it is reasonable to resort to data services built on top of massive amount of raw data.

In order to realize the promises of data services, in particular to foster their reuse, it is of vital importance to well document and clearly specify their semantics. While most current techniques manually associate APIs (Application Programming Interface) to data services, and describe their intended meaning with ad-hoc methods, often using natural language or complex metadata [Carey *et al.*, 2012], we propose a new approach, whose goal is to automatically associate formal semantic descriptions to data services. We base our proposal on the *Ontology-Based Data Access* (OBDA) paradigm [Poggi *et al.*, 2008]. An OBDA specification consists of an ontology expressed in Description Logic (DL) [Baader *et al.*, 2003], the schema of the data sources forming the information system, and a mapping between the source schema and the ontology. The ontology is a formal representation of the underlying domain, and the mapping specifies the relationship between the data at the sources and the elements in the ontology. The semantics of data services can be thus expressed using the elements of the domain ontology, which is assumed to be familiar to the consumer of data services.

But how can we automatically produce a semantic characterization of a data service, having an OBDA specification available? The idea is to exploit a new reasoning task over the OBDA specification, that works as follows: we express the data service in terms of a query over the sources, and we aim at automatically deriving the query over the ontology that best describes the data service, given the mapping. The following example illustrates this idea.

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

$\mathcal{O}$ = { $\mathsf{ErasmusStudent} \sqsubseteq \mathsf{Student}$, $\mathsf{MathStudent} \sqsubseteq \mathsf{Student}$, $\mathsf{Student} \sqsubseteq \neg\mathsf{Professor}$ }

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

$\mathcal{M}$ = { $\{(x) \mid s_1(x)\} \rightarrow \{(x) \mid \mathsf{Student}(x)\}$,
$\{(x) \mid s_2(x)\} \rightarrow \{(x) \mid \mathsf{Student}(x)\}$,
$\{(x) \mid s_3(x)\} \rightarrow \{(x) \mid \mathsf{Professor}(x)\}$,
$\{(x) \mid s_1(x), s_4(x,y)\} \rightarrow \{(x) \mid \mathsf{ErasmusStudent}(x)\}$,
$\{(x) \mid s_1(x), s_5(x,y)\} \rightarrow \{(x) \mid \mathsf{MathStudent}(x)\}$ }

One can verify that the query over $\mathcal{O}$ that best describes the data service $q_{\mathcal{S}} = \{(x) \mid s_1(x)\} \vee \{(x) \mid s_2(x)\}$ in terms of $\mathcal{O}$ is $q_{\mathcal{O}} = \{(x) \mid \mathsf{Student}(x)\}$.

Most of (if not all) the literature about managing data sources through an ontology [Lenzerini, 2018; Xiao *et al.*, 2018; Ortiz, 2018; Bienvenu, 2016] deals with user queries expressed over the ontology, and studies the problem of finding an *ontology-to-source rewriting*, i.e., a query over the source schema that, once executed over the data, provides the answers to the original query. Here, the problem is reversed, because we start with a source query and we aim at deriving a corresponding query over the ontology, called a *source-to-ontology rewriting* (*s-to-o rewriting* for short). Thus, we deal with a sort of reverse engineering problem, which is novel in

*Work supported by MIUR under the SIR project "MODEUS" – grant n. RBSI14TQHQ, and by Sapienza under the research project "PRE-O-PRE".

the investigation of both OBDA and data integration.

The notions introduced in this paper are relevant in a plethora of scenarios. For the sake of brevity, we mention only three of them. Following the ideas in [Cima, 2017], it can be shown that our notions of s-to-o rewriting can be used to provide the semantics of open datasets and open APIs published by organizations, which is a key aspect for unchaining all the potentials of open data. In [Lutz *et al.*, 2018], the concept of realization of source queries, similar to one of the notions studied here, is used for checking whether the mapping provides the right coverage for expressing the relevant data services at the ontology level. Our notions are also useful for a semantic-based approach to source profiling [Abedjan *et al.*, 2017], in particular for describing the structure and the content of a data source in terms of the business vocabulary.

The contributions provided by this paper can be summarized as follows. We propose a formal framework for the problem of semantically characterizing a data service through an ontology (Section 3). We introduce the notions of *perfect*, *sound*, and *complete* s-to-o rewritings, and we define two basic reasoning tasks, namely *verification* and *computation*. The former checks whether a given query is an s-to-o rewriting of a data service, whereas the latter computes one such rewriting. We show that, although the ideal notion is the one of perfect s-to-o rewriting, there are cases where, with the given mapping, no query over the ontology can precisely characterize the data service at hand. Thus, we introduce *maximally sound* and *minimally complete* s-to-o rewritings, which intuitively aim at approximating the perfect s-to-o rewriting of a data service at best, with the goal of either precision (sound rewriting), or recall (complete rewriting).

We study both the verification, and the computation problem for complete (Section 4), sound (Section 5) and perfect (Section 6) s-to-o rewritings in one of the most popular OBDA setting considered in the literature, namely where the ontology language is *DL-Lite*$_{\mathcal{R}}$, each mapping assertion maps a conjunctive query (CQ) over the source to a CQ over the ontology, and both the data service and the s-to-o rewriting are expressed as unions of CQs. For perfect and complete s-to-o rewritings we present algorithms for verification and computation, and characterize the complexity of both tasks. For the case of sound s-to-o rewritings, we do the same for verification, and then we precisely determine the cases where a maximally sound rewriting is not guaranteed to exist.

We single out a restricted setting for OBDA specifications that is still meaningful from the point of view of expressive power, and guarantees the existence of maximally sound s-to-o rewritings (Section 7). For such restricted setting, we provide algorithms and complexity results for verification and computation of maximally sound s-to-o rewritings. Preliminaries (Section 2), and conclusions (Section 8) complete the paper.

To the best of our knowledge, the problem studied in this paper has been (partially) addressed only in [Cima, 2017; Lutz *et al.*, 2018]. The former provides complexity upper bounds for complete s-to-o rewritings, and the latter focuses on both *DL-Lite*$_{\mathcal{R}}$ and the $\mathcal{EL}$ family of ontology languages, and studies perfect s-to-o rewritings only, under a slightly different semantics with respect to the one proposed here.

2 Preliminaries

We assume basic knowledge about databases [Abiteboul *et al.*, 1995] and DLs [Baader *et al.*, 2003]. In what follows, we use $\sigma(x)$ to denotes the size of x.

Databases and queries. A *database schema* is a set of predicate symbols, each with a specific arity, and a set of integrity constraints. Given a schema $\mathcal{S}$, an *$\mathcal{S}$-database* D is a set of *facts* $P(\vec{t})$ satisfying all integrity constraints in $\mathcal{S}$, where P is a predicate in $\mathcal{S}$ of arity n, and $\vec{t}$ is an n-uple of constants, each taken from a denumerable infinite set of symbols, where each such symbol is called an *$\mathcal{S}$-constant*, or simply constant.

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 a set of *answers* q^D, each answer being a tuple of constants. A *conjunctive query (CQ)* q over a schema $\mathcal{S}$ is an expression of the form $\{\vec{t} \mid \phi(\vec{t}, \vec{y})\}$, also denoted $q(\vec{t})$, where $\vec{t}$ is a tuple of *terms*, each term being either a constant or a variable, $\vec{y}$ is a tuple of variables not appearing in $\vec{t}$, called the *existential* variables of q, and $\phi(\vec{t}, \vec{y})$ is either $\bot$ (in this case we also say that the whole q is $\bot$), or a finite conjunction of atoms of the form $P(t_1, ..., t_n)$, where P is an n-ary predicate symbol of $\mathcal{S}$ and each t_j is either a constant, or a variable in $\vec{t} \cup \vec{y}$. We call ϕ and $\vec{t}$ the *body* and *target list of* q, respectively, and we sanction that every variable in $\vec{t}$ appears in ϕ. If $\vec{t}$ is empty, then the query is a boolean query. A *union of CQs (UCQ)* is a union of a finite set of conjunctive queries (called its disjuncts) with same arity. If not otherwise stated, we implicitly assume that all CQs of a UCQ have the same target list[1]. If q_1, q_2 are two queries with the same arity over $\mathcal{S}$, q_1 is contained in q_2, denoted as $q_1 \sqsubseteq q_2$ if for every $\mathcal{S}$-database D, $q_1^D \subseteq q_2^D$. Containment of CQs and UCQs is characterized in terms of homomorphism [Chandra and Merlin, 1977; Sagiv and Yannakakis, 1980]. In what follows, we also consider CQs with no existential variables occurring more than once (CQJFE), and unions thereof (UCQJFE).

***DL-Lite*$_{\mathcal{R}}$ ontologies.** We consider ontologies expressed in *DL-Lite*$_{\mathcal{R}}$, the member of the *DL-Lite* family [Calvanese *et al.*, 2007] that underpins OWL 2 QL [Motik *et al.*, 2012], i.e., the profile of OWL 2 especially designed for the OBDA scenarios. In *DL-Lite*$_{\mathcal{R}}$ axioms have the following forms:

$B_1 \sqsubseteq B_2$	$R_1 \sqsubseteq R_2$	(concept/role inclusion)
$B_1 \sqsubseteq \neg B_2$	$R_1 \sqsubseteq \neg R_2$	(concept/role disjointness)

where B_1, B_2 are *basic concepts*, i.e., expressions of the form A, $\exists P$, or $\exists P^-$, with A and P atomic concept (atomic concepts include the universal concept $\top$) and atomic role, respectively, and R_1 and R_2 *basic roles*, i.e., expressions of the form P, or P^-. We will also consider one sublanguage of *DL-Lite*$_{\mathcal{R}}$, namely *DL-Lite*$_{\mathsf{RDFS}}$, where both disjointness axioms, and concepts of the forms $\exists P$ or $\exists P^-$ in the right-hand side of the inclusion axioms, are ruled out.

OBDA. An OBDA specification Σ is a triple $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ [Poggi *et al.*, 2008], where $\mathcal{O}$ is a DL ontology, $\mathcal{S}$ is a database schema, called the *source schema*, and $\mathcal{M}$ is a set of *mapping assertions* (or simply mappings)

[1] All the results of this paper can be generalized to the case of UCQs whose disjuncts may have different target lists.

relating $\mathcal{S}$ to $\mathcal{O}$, i.e., assertions of the form $q_\mathcal{S} \rightarrow q_\mathcal{O}$, where $q_\mathcal{S}$ and $q_\mathcal{O}$ are CQs of the same arity over $\mathcal{S}$ and $\mathcal{O}$, respectively.

Mappings of the above form are called GLAV mappings. Special cases are GAV (Global-as-View) and LAV (Local-as-Views) mappings [Doan *et al.*, 2012]: in a GAV (resp., LAV) mapping, $q_\mathcal{O}$ (resp., $q_\mathcal{S}$) is simply an atom without existential variables. A GAV mapping is called *pure* if $q_\mathcal{O}$ does not have constants or repeated variables, i.e., it is either of the form $A(x)$, or $R(x, y)$, with x, y different variables.

An interpretation B for $\mathcal{O}$ is a *model for* Σ *relative to an* $\mathcal{S}$*-database* D if (i) it is a model of $\mathcal{O}$, and (ii) for every mapping $q_\mathcal{S} \rightarrow q_\mathcal{O}$ in $\mathcal{M}$, we have $q_\mathcal{S}^D \subseteq q_\mathcal{O}^B$, where $q_\mathcal{O}^B$ denotes the answers of $q_\mathcal{O}$ over the interpretation B (seen as a database). The set of models for Σ relative to D is denoted by $Mod_D(\Sigma)$, and D is said to be *consistent* with Σ if $Mod_D(\Sigma) \neq \emptyset$.

If $q_\mathcal{S}$ is a CQ, we denote by $\mathcal{M}(q_\mathcal{S})$ the query obtained by applying the chase [Maier *et al.*, 1979] w.r.t. $\mathcal{M}$ to the so-called freezing of $q_\mathcal{S}$, with the proviso that $\mathcal{M}(q_\mathcal{S})$ is $\bot$ if $q_\mathcal{S}$ is $\bot$, and $\top$ if its chase is empty. Note that the freezing of $q_\mathcal{S}$ is the database obtained by seeing the atoms of $q_\mathcal{S}$ as facts.

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 are the set of tuples $\vec{t}$ of $\mathcal{S}$-constants such that $\vec{t} \in q_\mathcal{O}^B$, for every $B \in Mod_D(\Sigma)$. An $\mathcal{O}$*-to-*$\mathcal{S}$ Σ*-rewriting* of a query $q_\mathcal{O}$ over $\mathcal{O}$ is a query $q_\mathcal{S}$ over $\mathcal{S}$ of the same arity as $q_\mathcal{O}$ such that for every $\mathcal{S}$-database D, $q_\mathcal{S}^D$ is a subset of the certain answers of $q_\mathcal{O}$ w.r.t. Σ and D. A *perfect* $\mathcal{O}$*-to-*$\mathcal{S}$ Σ*-rewriting* of $q_\mathcal{O}$, denoted by $cert_{q_\mathcal{O},\Sigma}$, is a query over $\mathcal{S}$ of the same arity as $q_\mathcal{O}$ 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. We say that query q_1 over $\mathcal{O}$ is *equivalent w.r.t.* Σ to query q_2 over $\mathcal{O}$ if $cert_{q_1,\Sigma} \equiv cert_{q_2,\Sigma}$. It is easy to see that, with a slight modification for taking care of the presence of $\top$ and $\bot$ in queries, we can use the results in [Levy *et al.*, 1995] to show that, if $\mathcal{O}$ has no axiom, and $q_\mathcal{O}$ is a UCQ over $\mathcal{O}$, one can compute a UCQ over $\mathcal{S}$, denoted by $\mathsf{REW}_\mathcal{M}(q_\mathcal{O})$, that is equivalent to $cert_{q_\mathcal{O},\Sigma}$.

We conclude the section with some observations about the case where the ontology $\mathcal{O}$ in Σ is a *DL-Lite*$_\mathcal{R}$ ontology. In this case, it is well-known (see, e.g., [Poggi *et al.*, 2008]) that an $\mathcal{S}$-database D is consistent with Σ if and only if $cert_{\mathcal{V}_\mathcal{O},\Sigma^p}^D = \emptyset$, where Σ^p is obtained from Σ by eliminating the disjointness axioms from $\mathcal{O}$, and $\mathcal{V}_\mathcal{O}$ is the $\mathcal{O}$-violation query, i.e., the boolean UCQ obtained by including a CQ q_d for each disjointness axiom, where q_d has the form $\{() | B_1(x) \wedge B_2(x)\}$ (resp., $\{() | R_1(x, y) \wedge R_2(x, y)\}$) for the axiom $B_1 \sqsubseteq \neg B_2$ (resp., $R_1 \sqsubseteq \neg R_2$). Also, we denote by $\mathcal{V}_\mathcal{O}^{t_1,\ldots,t_n}$ the UCQ over $\mathcal{O}$ with target list $(t_1, \ldots, t_n)$ obtained by adding $\top(t_1) \wedge \ldots \wedge \top(t_n)$ (written also as $\top(t_1, \ldots, t_n)$) to each of the disjuncts of $\mathcal{V}_\mathcal{O}$.

If $q_\mathcal{O}$ is a (U)CQ-query over $\mathcal{O}$, we denote by $\mathsf{PerfRef}_{q_\mathcal{O},\mathcal{O}}$ the UCQ computed by executing the algorithm PerfectRef described in [Calvanese *et al.*, 2007] on $q_\mathcal{O}$ and $\mathcal{O}$ (again, slightly modified to take care of $\top$ and $\bot$), and by $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma}$ the UCQ $\mathsf{REW}_\mathcal{M}(\mathsf{PerfRef}_{q_\mathcal{O},\mathcal{O}})$. Note that PerfectRef ignores the disjointness axioms in $\mathcal{O}$, and if D is consistent with Σ, then $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma}^D$ computes exactly $cert_{q_\mathcal{O},\Sigma}^D$. From these observations, and exploiting the results in [Calvanese *et al.*, 2012; Levy *et al.*, 1995], it is possible to prove that for an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, if $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology, and $q_\mathcal{O}(\vec{t})$ is a (U)CQ-query over $\mathcal{O}$, then $cert_{q_\mathcal{O},\Sigma} \equiv \mathsf{PerfRef}_{q_\mathcal{O},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}_\mathcal{O}^{\vec{t}},\Sigma}$.

3 Framework

We implicitly refer to an OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$. Intuitively, given a data service expressed as a query $q_\mathcal{S}$ over $\mathcal{S}$, we aim at finding the query over $\mathcal{O}$ that precisely characterizes $q_\mathcal{S}$ w.r.t. Σ. Since the evaluation of queries over $\mathcal{O}$ is based on certain answers, this means that we aim at finding a query over $\mathcal{O}$ whose certain answers w.r.t. Σ and D exactly capture the answers of $q_\mathcal{S}$ w.r.t. D, for every $\mathcal{S}$-database D. So, we are naturally led to the notion of perfect s-to-o rewriting. In what follows, $q_\mathcal{S}$ refers to a query over $\mathcal{S}$, and $q_\mathcal{O}$ to a query over $\mathcal{O}$ of the same arity.

Definition 2. $q_\mathcal{O}$ is a *perfect* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting* of $q_\mathcal{S}$ if for every $\mathcal{S}$-database D, $Mod_D(\Sigma) \neq \emptyset$ implies $q_\mathcal{S}^D = cert_{q_\mathcal{O},\Sigma}^D$.

The above notion is similar, but not equivalent, to the notion of realization in [Lutz *et al.*, 2018]. Indeed, while the latter sanctions that $q_\mathcal{S}^D = cert_{q_\mathcal{O},\Sigma}^D$ for *all* $\mathcal{S}$-databases, in our notion the condition is limited to the $\mathcal{S}$-databases that are consistent with Σ. The difference between the two notions is highlighted by the following example.

Example 3. Refer to Example 1, and consider again the query $q_\mathcal{S} = \{(x) \mid s_1(x)\} \vee \{(x) \mid s_2(x)\}$ over $\mathcal{S}$, and the query $q_\mathcal{O} = \{(x) \mid \mathsf{Student}(x)\}$ over $\mathcal{O}$. For the $\mathcal{S}$-database $D = \{s_1(a), s_3(a), s_4(a, b)\}$, we have that $q_\mathcal{S}^D = \{\langle a \rangle\}$, while, since D is inconsistent with Σ, $cert_{q_\mathcal{O},\Sigma}^D$ contains all $\mathcal{S}$-constants of D (including, for example, the tuple $\langle b \rangle$). It follows that $q_\mathcal{O}$ is not a realization of $q_\mathcal{S}$ in Σ, whereas, since $q_\mathcal{S}^D = cert_{q_\mathcal{O},\Sigma}^D$ for every $\mathcal{S}$-database D consistent with Σ, $q_\mathcal{O}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$.

As noted in [Cima, 2017; Lutz *et al.*, 2018] and illustrated in the next example, perfect s-to-o rewritings may not exist.

Example 4. Refer again to Example 1, and consider the data service expressed as the source query $q_\mathcal{S} = \{(x) \mid s_1(x)\}$. By inspecting the mappings, one can see that, since the certain answers of Student include the values stored both in s_1 and in s_2, such concept is too general for exactly characterizing $q_\mathcal{S}$. On the other hand, both ErasmusStudent and MathStudent are too specific, and therefore we can conclude that no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ exists.

In order to cope with the situations illustrated in the example, we introduce the notions of sound and complete s-to-o rewritings, which, intuitively, provide sound and complete approximations of perfect rewritings, respectively.

Definition 5. $q_\mathcal{O}$ 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 $cert_{q_\mathcal{O},\Sigma}^D \subseteq q_\mathcal{S}^D$ (resp., $q_\mathcal{S}^D \subseteq cert_{q_\mathcal{O},\Sigma}^D$).

Example 6. We refer to Example 4, and observe that $\{(x) \mid \mathsf{ErasmusStudent}(x) \wedge \mathsf{MathStudent}(x)\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S} = \{(x) \mid s_1(x)\}$, whereas $\{(x) \mid \mathsf{Student}(x)\}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$.

Obviously, $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{O}}$ is a sound and complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. There are also interesting relationships between the notions of $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings introduced here and the usual notions of rewritings studied in OBDA.

Proposition 7. *$q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}}$ is an $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$. If $q_{\mathcal{S}}$ is a perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$, then $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

It is easy to see that different sound or complete s-to-o rewritings of $q_{\mathcal{S}}$ may exist, and therefore it is reasonable to look for the "best" approximations of $q_{\mathcal{S}}$, at least relative to a certain class of queries.

Definition 8. $q_{\mathcal{O}} \in \mathcal{L}$ is *an $\mathcal{L}$-maximally sound* (respectively, *$\mathcal{L}$-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 $q' \in \mathcal{L}$ exists such that (i) q' is a sound (resp., 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 s.t. $cert^D_{q_{\mathcal{O}},\Sigma} \subset cert^D_{q',\Sigma}$ (resp., $cert^D_{q',\Sigma} \subset cert^D_{q_{\mathcal{O}},\Sigma}$).

Example 9. We refer again to Example 4, and observe that while $\{(x) \mid \mathsf{Student}(x)\}$ is a minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}} = \{(x) \mid s_1(x)\}$ in the class of UCQs, both $\{(x) \mid \mathsf{ErasmusStudent}(x)\}$, and $\{(x) \mid \mathsf{MathStudent}(x)\}$ are maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$ in the class of CQs, while $q_{\mathcal{O}} = \{(x) \mid \mathsf{ErasmusStudent}(x)\} \vee \{(x) \mid \mathsf{MathStudent}(x)\}$ is so in the class of UCQs.

Given the general framework presented so far, it is natural to consider the following two basic computational problems, for classes $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{L}_{\mathcal{O}}$ of queries:

- *Verification*: given $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{S}}$ over $\mathcal{S}$ and $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ over $\mathcal{O}$ of the same arity as $q_{\mathcal{S}}$, verify whether $q_{\mathcal{O}}$ is a sound (resp., complete, perfect) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$.
- *Computation*: given $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, and $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{S}}$ over $\mathcal{S}$ compute any $\mathcal{L}_{\mathcal{O}}$-maximally sound (resp., $\mathcal{L}_{\mathcal{O}}$-minimally complete, perfect) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if it exists.

In the rest of this paper, if not otherwise stated, we refer to the most common setting studied in OBDA, i.e., where (i) the ontology is expressed in *DL-Lite*$_{\mathcal{R}}$, (ii) $\mathcal{S}$ is a relational database schema without integrity constraints, and (iii) both $\mathcal{L}_{\mathcal{O}}$ and $\mathcal{L}_{\mathcal{S}}$ denote the class of UCQs. Interestingly, in this case, we have the following.

Proposition 10. *If q_1 and q_2 are UCQ-minimally complete (resp., UCQ-maximally sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, then they are equivalent w.r.t. Σ.*

4 Complete source-to-ontology rewritings

In this section, we study both the verification and the computation problem for complete s-to-o rewritings.

Verification. Suppose we want to check whether $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Obviously, if $q_{\mathcal{S}}$ is contained in $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$, then for every $\mathcal{S}$-database D consistent with Σ, we have that $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},\Sigma}$ and therefore the answer is positive. However, if $q_{\mathcal{S}}$ is not contained in $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$, it might be that $q_{\mathcal{O}}$ is still a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, in particular in the case where the non-emptiness of $q_{\mathcal{S}}$ in D reveals the presence of inconsistencies. From this observation, we derive the following characterization.

Proposition 11. *$q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}(\vec{t})$ if and only if $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_{\mathcal{O}},\Sigma}$.*

The following theorem characterizes the complexity of verification for complete s-to-o rewritings.

Theorem 12. *The verification problem for complete s-to-o rewritings is NP-complete.*

Proof sketch. As for the upper bound, we show how to check the containment $q_{\mathcal{S}}(\vec{t}) \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_{\mathcal{O}},\Sigma}$ in NP. For every disjunct q of $q_{\mathcal{S}}$, (i) we guess a query q' over $\mathcal{O}$ with the same arity of $q_{\mathcal{O}}$ and size at most the maximum between $\sigma(q_{\mathcal{O}})$ and $\sigma(\mathcal{V}^{\vec{t}}_{\mathcal{O}})$, a sequence ρ of ontology axioms, a query q'' over $\mathcal{S}$ of size $\sigma(\mathcal{M}) \times \sigma(q')$ and a function ϕ from the variables of q'' to the variables q, and (ii) we check in PTIME whether we can rewrite either $q_{\mathcal{O}}$ or $\mathcal{V}^{\vec{t}}_{\mathcal{O}}$ into q' through ρ, q'' is in $\mathsf{REW}_{\mathcal{M}}(q')$, and ϕ is a homomorphism from q'' to q.

As for the lower bound, the proof of NP-hardness is by a LOGSPACE reduction from the 3-COLOURABILITY problem, which is NP-complete [Garey *et al.*, 1976]. □

We point out that the result of NP-hardness holds even when $\mathcal{O}$ is empty, $\mathcal{M}$ is both a pure GAV mapping and a LAV mapping, and $q_{\mathcal{S}}$, $q_{\mathcal{O}}$ are boolean CQs.

Computation. Our algorithm for the computation of minimally complete s-to-o rewritings is below.

Algorithm 1
Input: $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}(\vec{t}) = q^1_{\mathcal{S}}(\vec{t}) \vee \ldots \vee q^n_{\mathcal{S}}(\vec{t})$ over $\mathcal{S}$
Output: $q_{\mathcal{O}}(\vec{t})$ over $\mathcal{O}$
begin
  return $q_{\mathcal{O}} = \bigvee_{i=1}^{n} \{\vec{t} \mid \mathcal{M}(q^i_{\mathcal{S}}) \wedge \top(\vec{t})\}$
end

Intuitively, the algorithm computes the output query as union of CQs obtained by simply applying the mapping $\mathcal{M}$ to each CQ $q^i_{\mathcal{S}}$ in $q_{\mathcal{S}}$, using $\top$ to bind the variables that are not involved in the application of $\mathcal{M}$ to $q^i_{\mathcal{S}}$.

Theorem 13. *Algorithm 1 computes the UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

The algorithm shows that the UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ always exists. Moreover, if $q_{\mathcal{S}}$ is a CQ, then it is a CQ. Finally, we observe that the complexity of Algorithm 1 does not depend on $\mathcal{O}$ and is in PTIME in $\sigma(q_{\mathcal{S}})$. Moreover, it is in EXPTIME in $\sigma(\mathcal{M})$, since it essentially applies the chase using the queries in the mapping. It can be shown that an algorithm for computing the UCQ-minimally complete s-to-o rewritings that is PTIME in the size of all inputs would imply a PTIME algorithm for CQ containment. So, assuming PTIME $\neq$ NP, the computation problem cannot be solved in PTIME.

5 Sound source-to-ontology rewritings

We now turn to both the verification and the computation problem for sound s-to-o rewritings.

Verification. We remind the reader that, for an $\mathcal{S}$-database D consistent with Σ, $\mathsf{PerfRef}^D_{q_\mathcal{O},\Sigma}$ computes exactly $cert^D_{q_\mathcal{O},\Sigma}$. So, intuitively, checking whether $q_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ means checking whether for all $\mathcal{S}$-databases D, either $Mod_D(\Sigma) = \emptyset$ or $\mathsf{PerfRef}^D_{q_\mathcal{O},\Sigma} \subseteq q^D_\mathcal{S}$. This observation leads to the following characterization.

Proposition 14. *$q_\mathcal{O}(\vec{t})$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ if and only if $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma} \sqsubseteq q_\mathcal{S} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_\mathcal{O},\Sigma}$.*

The following theorem characterizes the complexity of the verification problem for sound s-to-o rewritings.

Theorem 15. *The verification problem for sound s-to-o rewritings is Π^p_2-complete.*

Proof sketch. As for the upper bound, we show that checking $\mathsf{PerfRef}_{q_\mathcal{O}(\vec{t}),\Sigma} \not\sqsubseteq q_\mathcal{S} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_\mathcal{O},\Sigma}$ can be done in Σ^p_2: we guess a CQ q_1 over $\mathcal{S}$ whose size is at most $\sigma(\mathcal{M}) \times \sigma(q_\mathcal{O})$, we check in PTIME whether q_1 is a disjunct of $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma}$, similarly to what described in Theorem 12, and then we use an NP oracle to check $q_1 \not\sqsubseteq q_\mathcal{S} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_\mathcal{O},\Sigma}$, again using the method mentioned in Theorem 12.

As for the lower bound, the proof of Π^p_2-hardness is by a LOGSPACE reduction from the $\forall\exists$-CNF problem, which is Π^p_2-complete [Stockmeyer, 1976]. □

We point out that the result of Π^p_2-hardness holds even when $\mathcal{O}$ is empty, $\mathcal{M}$ is both a GAV mapping and a LAV mapping, and $q_\mathcal{S}$, $q_\mathcal{O}$ are boolean CQs.

Computation. We address the problem of computing UCQ-maximally sound s-to-o rewritings. Our main result is that there are many cases where a UCQ-maximally sound s-to-o rewriting of a query is not guaranteed to exist. To illustrate the result, we introduce a specific setting for OBDA specifications, that we call *restricted*, obtained from the general one by: (i) limiting the ontology language to *DL-Lite*$_\mathsf{RDFS}$, (ii) limiting the mapping to pure GAV, and (iii) limiting $q_\mathcal{S}$ to UCQJFEs. We now show that, surprisingly, as soon as we try to extend such setting, we lose the guarantee of the existence of s-to-o rewritings that are maximally sound in the class of UCQs.

Theorem 16. *UCQ-maximally sound s-to-o rewritings of a query $q_\mathcal{S}$ may not exist if we extend the restricted setting with each of the following features:*

1. *disjointness axioms in the ontology;*
2. *inclusion axioms with $\exists R$ as right-hand side in the ontology;*
3. *LAV mapping assertions, even without joins involving existential variables in the right-hand side;*
4. *non-pure GAV mapping assertions;*
5. *$q_\mathcal{S}$ in a fragment of CQs going beyond CQJFEs.*

Proof sketch. We present the proof for case 5. Consider the OBDA specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O}$ has no axiom, and $\mathcal{M}$ consists of the following pure GAV mappings:

$$\{(x,y) \mid s_1(y,y) \wedge s_3(x)\} \rightarrow \{(x,y) \mid R(x,y)\}$$
$$\{(x,y) \mid s_1(x,y)\} \rightarrow \{(x,y) \mid R(x,y)\}$$

and let $q_\mathcal{S}$ be the query $\{() \mid s_1(x,y) \wedge s_1(y,z)\}$. Observe that $q'_\mathcal{O} = \{() \mid R(x,y) \wedge R(y,z)\}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, but is not sound, because the query $q'_\mathcal{S} = \{() \mid s_1(x,y) \wedge s_1(z,z) \wedge s_3(y)\}$ is a disjunct of $\mathsf{PerfRef}_{q'_\mathcal{O},\Sigma}$ such that $q'_\mathcal{S} \not\sqsubseteq q_\mathcal{S}$. Conversely, one can verify that each of the following queries is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$:

- $q_0 = \{() \mid R(x,y) \wedge R(y,y)\}$,
- $q_1 = \{() \mid R(x,y) \wedge R(y,z_1) \wedge R(z_1,y)\}$,
- $q_2 = \{() \mid R(x,y) \wedge R(y,z_1) \wedge R(z_1,z_2) \wedge R(z_2,y)\}$,
- ...

More precisely, if we define q_n to be $\{() \mid R(x,y) \wedge R(y,z_1) \wedge R(z_1,z_2) \wedge \ldots \wedge R(z_{n-1},z_n) \wedge R(z_n,y)\}$, for $n \geq 2$,, then it can be shown that every q_n is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, and for no pair (i,j), with $i \neq j$, $i,j \geq 0$, $cert_{q_i,\Sigma} \sqsubseteq cert_{q_j,\Sigma}$. It follows that the infinite union of q_0, q_1, and all q_n's can be shown to be the maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ in the class of positive queries. □

It remains to study sound s-to-o rewritings in the restricted setting. We do so in Section 7.

6 Perfect source-to-ontology rewritings

Both the verification and the computation problem for perfect s-to-o rewritings can be addressed by combining the techniques illustrated in the previous sections. As for verification, we can check whether $q_\mathcal{O}(\vec{t})$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}(\vec{t'})$ by checking both $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma} \sqsubseteq q_\mathcal{S} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_\mathcal{O},\Sigma}$ and $q_\mathcal{S} \sqsubseteq \mathsf{PerfRef}_{q_\mathcal{O},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_\mathcal{O},\Sigma}$. As for computation, we can first compute the query $q_\mathcal{O}$ that is the UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, and then check whether $q_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$. If the answer is positive, we return $q_\mathcal{O}$, otherwise we report that no perfect rewriting exists. From the above observation we derive the following: (*i*) all complexity results illustrated for the case of sound s-to-o rewritings hold for perfect rewritings as well, and (*ii*) if $q_\mathcal{S}$ is a CQ, then either its perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting does not exist, or it is a CQ as well.

Finally, we briefly discuss the case of perfect rewritings under the semantics used in [Lutz *et al.*, 2018], that imposes the condition $q^D_\mathcal{S} = cert^D_{q_\mathcal{O},\Sigma}$ for *all* $\mathcal{S}$-databases. From the results presented in the previous sections, it follows that $q_\mathcal{O}(\vec{t})$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}(\vec{t'})$ under such semantics if and only if $q_\mathcal{S} \equiv \mathsf{PerfRef}_{q_\mathcal{O},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^{\vec{t}}_\mathcal{O},\Sigma}$. This allows us to easily derive algorithms and complexity bounds for both verification and computation in this case, too.

7 Restricted setting

We now deal with the restricted setting mentioned at the end of Section 5. Before delving into the technical part, we observe that, despite its limitations, the expressive power of this setting is sufficient for several meaningful applications. Indeed, several popular ontologies are expressible in *DL-Lite*$_\mathsf{RDFS}$ (e.g., `Dublin Core` [Weibel *et al.*, 1998] and `SKOS` [Miles and Bechhofer, 2009]), and the form of pure GAV mapping is exactly the one originally defined in the literature of data integration. Moreover, UCQJFEs captures data services expressible in the famous USPJ (Union, Select,

Project, Join) fragment of Relational Algebra [Codd, 1970], with the only limitation that joining variables cannot be projected out. Note that such fragment is the one needed for all tasks related to source profiling [Abedjan *et al.*, 2017].

Verification. Let us start with the following crucial definition.

Definition 17. Let $q_1(\vec{t})$ and $q_2(\vec{t})$ be two CQs, and let $F_1 = S(t_{1,1}, \ldots, t_{1,n})$ and $F_2 = S(t_{2,1}, \ldots, t_{2,n})$ be two atoms of q_1 and q_2, respectively. We say that F_1 *instantiates* F_2, if for every $i = 1, \ldots, n$, we have that if $t_{2,i}$ is a term of $\vec{t}$ or a constant, then $t_{2,i} = t_{1,i}$.

Clearly, given atoms F_1 in q_1 and F_2 in q_2, checking whether F_1 instantiates F_2 can be done in PTIME. Based on this observation, the following lemma shows that checking whether a UCQ is contained in a UCQJFE is tractable.

Lemma 18. *Given a UCQ q_1 and a UCQJFE q_2 of the same arity, checking whether $q_1 \sqsubseteq q_2$ can be done in* PTIME.

We are now ready to characterize the complexity of verification in the restricted setting.

Theorem 19. *The verification problem for sound s-to-o rewritings in the restricted setting is coNP-complete, and can be solved in* PTIME *when $q_{\mathcal{S}}$ is a CQJFE.*

Proof sketch. The coNP upper bound is obtained by noticing that we have to guess a disjunct q_1 of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ and then check whether q_1 is not contained in $q_{\mathcal{S}}$, which, by virtue of Lemma 18, can be done in PTIME. The coNP lower bound is shown by reduction from VALIDITY. To show that verification is in PTIME when $q_{\mathcal{S}}$ is a CQJFE, we notice that, for the characteristics on the OBDA setting, for every $q(\vec{t})$ in $q_{\mathcal{O}}$, $\mathsf{PerfRef}_{q,\Sigma} = \bigwedge_{\alpha \in q} \mathsf{PerfRef}_{q_\alpha,\Sigma}$, where q_α is the query with body α and target list the tuple of variables that occur both in $\vec{t}$ and α. Hence, verification can be solved by checking, for every atom F in $q_{\mathcal{S}}$ and every query q in $q_{\mathcal{O}}$, whether there is an atom G in q such that all disjuncts of $\mathsf{PerfRef}_{q_G,\Sigma}$ contain at least one atom that instantiates F. Clearly, this can be done in PTIME w.r.t. $\sigma(q_{\mathcal{S}})$, $\sigma(\mathcal{M})$, and $\sigma(q_{\mathcal{O}})$. □

We observe that, as long as $q_{\mathcal{S}}$ is a UCQJFE, the coNP upper bound holds even when $\mathcal{O}$ is expressed in the fragment of *DL-Lite*$_{\mathcal{R}}$ that does not admit disjointness axioms, and $\mathcal{M}$ is GLAV, while the coNP-hardness holds even when $\mathcal{O}$ is empty, $\mathcal{M}$ is a set of both pure GAV and LAV mappings, $q_{\mathcal{O}}$ is a CQ, and both $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ have no existential variables.

Computation. We provide an algorithm to compute the maximally sound s-to-o rewritings, thus proving that in the restricted setting, for each query $q_{\mathcal{S}}$, the maximally sound s-to-o rewriting of $q_{\mathcal{S}}$ always exists.

Let $\gamma(\mathcal{M})$ be the number of mapping assertions in $\mathcal{M}$ and $\eta(q_{\mathcal{S}})$ the number of distinct atoms appearing in $q_{\mathcal{S}}$. Moreover, let $bound(q_{\mathcal{S}}) = 1+\gamma(\mathcal{M})+\gamma(\mathcal{M})^2+\ldots+\gamma(\mathcal{M})^{\eta(q_{\mathcal{S}})}$ if $q_{\mathcal{S}}$ is a UCQJFE, and $bound(q_{\mathcal{S}}) = \eta(q_{\mathcal{S}})$ if $q_{\mathcal{S}}$ is a CQJFE. The following lemma shows that we can limit our attention to queries with at most $bound(q_{\mathcal{S}})$ atoms when we search for the maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Lemma 20. *If a CQ $q_{\mathcal{O}}(\vec{t})$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, then there exists a CQ $q'_{\mathcal{O}}(\vec{t})$ which is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ whose body is the conjunction of m atoms appearing in $q_{\mathcal{O}}$, where $m \leq bound(q_{\mathcal{S}})$.*

The following algorithm derives immediately.

Algorithm 2
Input: $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}$ (U)CQJFE over $\mathcal{S}$
Output: $q_{\mathcal{O}}$ over $\mathcal{O}$
begin
  $q_{\mathcal{O}} := \bot$
  for each query q over $\mathcal{O}$ with at most $bound(q_{\mathcal{S}})$ atoms, involving only constants from $q_{\mathcal{S}}$ and $\mathcal{M}$
    if q is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ rewriting of $q_{\mathcal{S}}$ **then** $q_{\mathcal{O}} := q_{\mathcal{O}} \vee q$
  return $q_{\mathcal{O}}$
end

Note that the disjuncts of query $q_{\mathcal{O}}$ computed by Algorithm 2 do not have necessarily the same target list.

Theorem 21. *Algorithm 2 computes the UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof sketch. It is immediate to verify that the query returned by the algorithm is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. To show that it is the maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, we proceed by contradiction, i.e., by assuming that there exists a CQ q' such that $\mathsf{PerfRef}_{q',\Sigma} \sqsubseteq q_{\mathcal{S}}$ and $\mathsf{PerfRef}_{q',\Sigma} \not\sqsubseteq \mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$, for every disjunct $q'_{\mathcal{O}}$ of $q_{\mathcal{O}}$, where $q_{\mathcal{O}}$ is the query computed by Algorithm 2. Let q'' be obtained by substituting in q' each constant neither in $q_{\mathcal{S}}$ nor in $\mathcal{M}$ (if any) with a new fresh existential variable. It can be shown that in the restricted case, q'' would be such that $\mathsf{PerfRef}_{q'',\Sigma} \sqsubseteq q_{\mathcal{S}}$. Also, let m be the number of atoms of q''. If $m > bound(q_{\mathcal{S}})$, by Lemma 20, there exists a query $\bar{q}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ rewriting of $q_{\mathcal{S}}$ whose body is the conjunction of $\bar{m}$ atoms appearing in q'', where $\bar{m} \leq bound(q_{\mathcal{S}})$. But then, since $\bar{q}$ possibly contains only constants in $q_{\mathcal{S}}$ or $\mathcal{M}$ and since $\bar{m} \leq bound(q_{\mathcal{S}})$, by construction, $\bar{q}$ would be a disjunct of $q_{\mathcal{O}}$ and we get a contradiction. Finally, if $m \leq bound(q_{\mathcal{S}})$, we obtain a contradiction, by a similar argument. □

It can be shown that Algorithm 2 (i) computes the unique (up to equivalence w.r.t. Σ) maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of monotone queries, (ii) is PTIME in $\sigma(\mathcal{O})$ and $\sigma(\mathcal{M})$, and EXPTIME in $\eta(q_{\mathcal{S}})$. Finally, we can show that (i) assuming PTIME $\neq$ NP, the computation problem cannot be solved in PTIME, and (ii) there are cases where the number of atoms of the maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is necessarily exponential w.r.t. $\eta(q_{\mathcal{S}})$.

8 Conclusion

We have presented a framework for semantically characterizing data services through ontologies, and carried out a comprehensive analysis for the most common OBDA setting, including a restricted setting, still useful in practice. We plan to continue this work along several directions. For example, in the unrestricted setting, it would be interesting to study the problem of checking for the existence of a UCQ-maximally sound source-to-ontology rewriting of a query, and computing it in case it exists. Also, still for the unrestricted setting, we aim at singling out the minimal class $\mathcal{L}_{\mathcal{O}}$ of queries that guarantees the existence of an $\mathcal{L}_{\mathcal{O}}$-maximally sound source-to-ontology rewriting of a query $q_{\mathcal{S}}$. Furthermore, we will extend our analysis to OBDA settings going beyond the one based on *DL-Lite*$_{\mathcal{R}}$, for example by considering *DL-Lite*$_{\mathcal{A}}$, the $\mathcal{EL}$ family, or other DLs as ontology languages.

References

[Abedjan *et al.*, 2017] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. Data profiling: A tutorial. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 1747–1751, 2017.

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Bienvenu, 2016] Meghyn Bienvenu. Ontology-mediated query answering: Harnessing knowledge to get more from data. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 4058–4061, 2016.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2012] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. View-based query answering in description logics: Semantics and complexity. *J. of Computer and System Sciences*, 78:26–46, 2012.

[Carey *et al.*, 2012] Michael J. Carey, Nicola Onose, and Michalis Petropoulos. Data services. *Communications of the ACM*, 55(6):86–97, 2012.

[Chandra and Merlin, 1977] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proc. of the 9th ACM Symp. on Theory of Computing (STOC)*, pages 77–90, 1977.

[Cima, 2017] Gianluca Cima. Preliminary results on ontology-based open data publishing. In *Proc. of the 30th Int. Workshop on Description Logic (DL)*, volume 1879 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2017.

[Codd, 1970] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[Doan *et al.*, 2012] AnHai Doan, Alon Y. Halevy, and Zachary G. Ives. *Principles of Data Integration*. Morgan Kaufmann, 2012.

[Garey *et al.*, 1976] Michael R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

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

[Levy *et al.*, 1995] Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering queries using views. In *Proc. of the 14th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS)*, pages 95–104, 1995.

[Lutz *et al.*, 2018] Carsten Lutz, Johannes Marti, and Leif Sabellek. Query expressibility and verification in ontology-based data access. In *Proc. of the 16th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*, pages 389–398, 2018.

[Maier *et al.*, 1979] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *ACM Trans. on Database Systems*, 4(4):455–469, 1979.

[Miles and Bechhofer, 2009] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System. W3C Recommendation, World Wide Web Consortium, 2009. Available at `http://www.w3.org/TR/skos-reference`.

[Motik *et al.*, 2012] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. OWL 2 Web Ontology Language profiles (second edition). W3C Recommendation, World Wide Web Consortium, 2012. Available at `http://www.w3.org/TR/owl2-profiles/`.

[Ortiz, 2018] Magdalena Ortiz. Improving data management using domain knowledge. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 5709–5713, 2018.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.

[Sagiv and Yannakakis, 1980] Yehoshua Sagiv and Mihalis Yannakakis. Equivalences among relational expressions with the union and difference operators. *J. of the ACM*, 27(4):633–655, 1980.

[Stockmeyer, 1976] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Weibel *et al.*, 1998] Stuart Weibel, John A. Kunze, Carl Lagoze, and Misha Wolf. Dublin core metadata for resource discovery. *Request for Comments*, 2413:1–8, 1998.

[Xiao *et al.*, 2018] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 5511–5519, 2018.

[Zheng *et al.*, 2013] Zibin Zheng, Jieming Zhu, and Michael R. Lyu. Service-generated big data and big data-as-a-service: An overview. In *Proc. of the 2013 IEEE Int. Conf. on Big Data*, pages 403–410, 2013.