Towards Mapping Analysis in Ontology-based Data Access

Domenico Lembo[1], Jose Mora[1], Riccardo Rosati[1], Domenico Fabio Savo[1], Evgenij Thorstensen[2]

[1] Sapienza Università di Roma
lastname@dis.uniroma1.it

[2] University of Oslo
evgenit@ifi.uio.no

Abstract. In this paper we study mapping analysis in ontology-based data access (OBDA), providing an initial set of foundational results for this problem. We start by defining general, language-independent notions of mapping inconsistency, mapping subsumption, and mapping redundancy in OBDA. Then, we focus on specific mapping languages for OBDA and illustrate techniques for verifying the above properties of mappings.

1 Introduction

Ontology-based data access (OBDA) is a data integration paradigm that relies on a three-level architecture, constituted by the ontology, the data sources, and the mapping between the two [18]. The ontology is the specification of a conceptual view of the domain, and it is the system interface towards the user, whereas the mapping relates the elements of the ontology with the data at the sources.

In the past years, studies on OBDA have mainly concentrated on query answering, and various algorithms for it have been devised, as well as tools implementing them [6, 20, 7, 16, 4, 27, 22]. Intensional reasoning has been instead so far limited to the ontology level only. This means that currenlty available services of this kind in OBDA are exactly as for stand-alone ontologies (e.g., concept/role subsumption, classification, logical implication, etc.). As a consequence, in the specification of an OBDA system, a designer can rely only on classical off-the-shelf ontology reasoners (e.g., [26, 25, 24, 12]), but she cannot find tools supporting the modeling of the other crucial component of the OBDA architecture, i.e., the mapping.

Both industrial and research OBDA projects (see, e.g., [13, 1]) have experienced that mapping specification is a very complex activity, which requires a profound understanding of both the ontology and the data sources. Indeed, data sources are in general autonomous and pre-existing the OBDA application, and thus the way in which they are structured typically does not reflect the ontology, which is instead an independent representation of the domain of interest, rather than of the underlying data sources. To reconcile this "cognitive distance" between the sources and the ontology, the mapping usually assumes a complex form, and it is expressed in terms of assertions that relate queries over the ontology to queries over the data sources.

This form of mappings has been widely studied in data integration and data exchange [17, 9, 2]. In these contexts, the research on mappings has been mainly focused on mapping composition or inversion, whereas very few efforts have been made towards the analysis of the specification, to verify, e.g., whether it is redundant or inconsistent per se (i.e., independently from the source data). In fact, in data integration and exchange, the integrated (a.k.a. global or target) schema is in general not as expressive as an ontology, and thus analysis checks are to some extent easier or less crucial than in OBDA.

In this paper we study mapping analysis in OBDA, with the aim of providing the designer with services that are useful to devise a well-founded OBDA specification. We introduce our novel definitions that formalize properties of interest for the mapping. In particular, we define when a mapping $\mathcal{M}$ is inconsistent w.r.t. an ontology $\mathcal{O}$ and a source schema $\mathcal{S}$, which intuitively means that retrieving data through all the assertions in $\mathcal{M}$ always leads to an inconsistent OBDA specification composed by $\mathcal{O}$, $\mathcal{M}$ and $\mathcal{S}$, whatever non-empty source instance is assigned to $\mathcal{S}$. Also, we define when a mapping $\mathcal{M}$ subsumes a mapping $\mathcal{M}'$ under $\mathcal{O}$ and $\mathcal{S}$, which intuitively means that the systems composed by $\mathcal{O}$, $\mathcal{S}$, and either $\mathcal{M}$ or $\mathcal{M} \cup \mathcal{M}'$ are equivalent (and thus $\mathcal{M}'$ is redundant in the specification). We point out that verifying such properties is indeed of crucial importance in real-life OBDA projects, when hundreds of mapping assertions are usually needed, and it is very likely that mappings are redundant or even inconsistent in the sense we have described above.

After defining the mapping analysis tasks we are interested in, we discuss techniques for verifying consistency, redundancy, and subsumption for (some generalizations of) the so-called GAV mapping [17], when some specific languages for querying the sources and the ontology are used in mapping assertions.

We organize our paper as follows. In Section 2 we give some preliminary definitions on OBDA and mapping languages. In Section 3 we provide our notions of consistency, subsumption, and redundancy for mappings. In Section 4 we study decidability of verification tasks for some mapping languages under the GAV paradigm. In Section 5 we add some preliminary discussions on mappings that go beyond GAV, and in Section 6 we conclude the paper.

2 Definitions

OBDA specifications. An OBDA specification is a triple $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is an ontology, $\mathcal{S}$ is a source schema, and $\mathcal{M}$ is a mapping between the two. $\mathcal{O}$ typically (although not necessarily) represents intensional knowledge and is specified in a language $\mathcal{L}_{\mathcal{O}}$, whereas $\mathcal{S}$ is specified in a language $\mathcal{L}_{\mathcal{S}}$. We denote with $\Sigma_{\mathcal{O}}$ and $\Sigma_{\mathcal{S}}$ the signature of $\mathcal{O}$ and $\mathcal{S}$, respectively, and we assume that both $\mathcal{L}_{\mathcal{O}}$ and $\mathcal{L}_{\mathcal{S}}$ are (fragments of) first-order logic (FOL). For instance, in a typical OBDA setting, $\mathcal{O}$ is a Description Logic TBox and $\mathcal{S}$ is a relational schema, possibly with integrity constraints [20]. Finally, the mapping $\mathcal{M}$ is a set of assertions of the form

$$\phi(\boldsymbol{x}) \rightsquigarrow \psi(\boldsymbol{x}) \tag{1}$$

where $\phi(\boldsymbol{x})$ is a query over $\Sigma_{\mathcal{S}}$ and $\psi(\boldsymbol{x})$ is a query over $\Sigma_{\mathcal{O}}$, both with free variables $\boldsymbol{x}$, which are called the *frontier variables*. The number of variables in $\boldsymbol{x}$ is the *arity* of the mapping assertion. Given a mapping assertion m of the form (1), we also use $FR(m)$ do denote the frontier variables $\boldsymbol{x}$, $head(m)$ to denote the query $\psi(\boldsymbol{x})$, and $body(m)$ to denote the query $\phi(\boldsymbol{x})$, and we assume that both such queries are specified in (some fragment of) FOL.

Example 1. We give here an example of OBDA specification that we will use as ongoing example throughout the paper. We refer to a setting in which the source schema is relational and the ontology is expressed in a basic Description Logic language [3], which actually corresponds to *DL-Lite*$_{core}$ [6]. Since $\mathcal{S}$ is relational, queries in the body of mapping assertions are encoded in SQL.

Then, consider the following schema $\mathcal{S}$ of the database used in a zoo for handling information about the animals and the area of the zoo they live. In the schema, the underlined attributes represent the keys of the tables, and we also assume that a foreing key is specified between the attribute `AREA` of ANM_TAB and the table AREA_TAB.

$\mathsf{ANM\_TAB}(\underline{\texttt{ANM\_CODE}},\texttt{NAME},\texttt{BREED},\texttt{AREA})$
$\mathsf{AREA\_TAB}(\underline{\texttt{AREA\_CODE}},\texttt{SIZE})$

An ontology $\mathcal{O}$ modeling a very small portion of the zoo domain is as follow.

$$\begin{aligned}\mathcal{O} = \{\ & \mathsf{Lion} \sqsubseteq \mathsf{Animal}, \mathsf{Monkey} \sqsubseteq \mathsf{Animal}, \mathsf{Lion} \sqsubseteq \neg\mathsf{Monkey}, \mathsf{Animal} \sqsubseteq \exists\mathsf{name},\\ & \mathsf{Animal} \sqsubseteq \exists\mathsf{locatedIn}, \exists\mathsf{locatedIn}^- \sqsubseteq \mathsf{Area}, \mathsf{Area} \sqsubseteq \exists\mathsf{size}\}\end{aligned}$$

In words, $\mathcal{O}$ specifies that both lions (Lion) and monkeys (Monkey) are animals (Animal), a lion cannot be a monkey, and every animal has a name (name) and is located in (locatedIn) an area (Area). Moreover, every area has a size (size).

An example of mapping $\mathcal{M}$ between $\mathcal{O}$ and $\mathcal{S}$ is as follows:

$m1$: `SELECT ANM_CODE AS X, NAME AS Y FROM ANM_TAB` $\rightsquigarrow$ $\mathsf{Animal}(X) \land \mathsf{name}(X,Y)$

$m2$: `SELECT ANM_CODE AS X, AREA AS Y FROM ANM_TAB WHERE BREED = 'Lion'` $\rightsquigarrow$ $\mathsf{Lion}(X) \land \mathsf{locatedIn}(X,Y)$

$m3$: `SELECT ANM_CODE AS X, AREA AS Y FROM ANM_TAB WHERE BREED = 'Monkey'` $\rightsquigarrow$ $\mathsf{Monkey}(X) \land \mathsf{locatedIn}(X,Y)$

$m4$: `SELECT ANM_CODE AS X, AREA AS Y FROM ANM_TAB` $\rightsquigarrow$ $\mathsf{locatedIn}(X,Y)$

$m5$: `SELECT AREA_CODE AS X, SIZE AS Y FROM AREA_TAB` $\rightsquigarrow$ $\mathsf{Area}(X) \land \mathsf{size}(X,Y)$

□

The semantics of an OBDA specification $\mathcal{J}$ is defined with respect to a source instance that is legal for $\mathcal{S}$. More precisely, a source instance D is a set of facts over $\Sigma_{\mathcal{S}}$. Given such a D, we denote by $\mathcal{I}_D$ the interpretation over $\Sigma_{\mathcal{S}}$ that is isomorphic to D. Then, we say that D *is legal for* $\mathcal{S}$ if $\mathcal{I}_D \models \mathcal{S}$. For example, if

$\mathcal{S}$ is relational, we consider as legal only the instances that satisfy the integrity constraints on $\mathcal{S}$. We assume that for each $\mathcal{S}$ a legal instance always exists. Then, for each mapping assertion $m \in \mathcal{M}$ we denote with $\pi(m)$ the FOL formula

$$\forall \boldsymbol{x}.\phi(\boldsymbol{x}) \rightarrow \exists \boldsymbol{z}.\psi(\boldsymbol{y}, \boldsymbol{z})$$

where $\boldsymbol{z}$ denotes the existential variables in $head(m)$, and we pose $\pi(\mathcal{M}) = \{\pi(m) \mid m \in \mathcal{M}\}$. Then, the *models of* $\mathcal{J}$ *w.r.t.* D are the models of the FOL theory $\mathcal{O} \cup \pi(\mathcal{M}) \cup D$ that are isomorphic to D on the interpretation of the predicates in $\mathcal{S}$. We denote with $Models(\mathcal{J}, D)$ the set of models of $\mathcal{J}$ w.r.t. D.

Mapping languages. In this paper we study specific cases of OBDA specifications where we fix the fragment of FOL adopted for the queries in the head and in the body of mapping assertions. In particular, we mainly focus on the following mapping languages:

- *FO2DCQ*, where, for each $m \in \mathcal{M}$, $body(m)$ is a FOL query over $\mathcal{S}$ and $head(m)$ is a conjunctive query over $\mathcal{O}$ without existential variables;
- *CQ2DCQ*, where, for each $m \in \mathcal{M}$, $body(m)$ is a conjunctive query over $\mathcal{S}$ and $head(m)$ is a conjunctive query over $\mathcal{O}$ without existential variables.

Obviously, *FO2DCQ* subsumes *CQ2DCQ*, and thus all defintions we give in the following for *FO2DCQ* mappings also apply to *CQ2DCQ* mappings.

Both languages above are extended forms of the so-called GAV mapping, which, differently from the LAV mapping, does not allow for non-free variables in the head of assertions [17, 9]. On the other-hand, GAV is the only kind of mapping that has been used in practical OBDA and data integration applications [13, 1]. An example of *CQ2DCQ* language is given in Example 1.

We notice that classical GAV mapping only allows for single atom queries in the head of assertions (instead of conjunctions of atoms). However, it is easy to see that each *FO2DCQ* assertion can be rephrased into a logically equivalent set of classic GAV assertions. More precisely, let

$$m: \ \exists \boldsymbol{w}.\phi(\boldsymbol{x}, \boldsymbol{w}) \rightsquigarrow \psi(\boldsymbol{x})$$

be one such assertion, where we have explicited the existential variables in the body query, then, we can rephrase m into the following set of mapping assertions

$$\{\exists \overline{\boldsymbol{x}}_i, \boldsymbol{w}.\phi(\boldsymbol{x}_i, \overline{\boldsymbol{x}}_i, \boldsymbol{w}) \rightsquigarrow \psi_i(\boldsymbol{x}_i) \mid \text{ for each atom } \psi_i(\boldsymbol{x}_i) \text{ in } body(m)\}$$

where $\overline{\boldsymbol{x}}_i$ denotes the free variables of $\boldsymbol{x}$ that do not occur in $\boldsymbol{x}_i$. Given a *FO2DCQ* mapping $\mathcal{M}$, we denote with $\mathsf{Split}(\mathcal{M})$ the above set of mappings.

Example 2. Consider the $m1$ mapping assertion of Example 1. The set $\mathsf{Split}(m1)$ contains the following mapping assertions:

$m1'$: `SELECT ANM_CODE AS X FROM ANM_TAB` $\rightsquigarrow$ $\mathsf{Animal}(X)$
$m1''$: `SELECT ANM_CODE AS X, NAME AS Y FROM ANM_TAB` $\rightsquigarrow$ $\mathsf{name}(X, Y)$

□

Let m be a *FO2DCQ* mapping assertion of arity n and let t be an n-tuple of constants. We denote by $m(t)$ the mapping assertion obtained from m by replacing the frontier variables of m with the constants in t. Then, let D be a source instance, we define the *facts retrieved by* m *on* D, denoted by $Retr(m, D)$, as the set of ground atoms

$$\{\alpha \mid t \text{ is a tuple of constants and } \mathcal{I}_D \models body(m(t)) \text{ and } \alpha \text{ occurs in } head(m(t))\}$$

Moreover, given a *FO2DCQ* mapping $\mathcal{M}$ and a source instance D, we define the *facts retrieved by* $\mathcal{M}$ *on* D, denoted by $Retr(\mathcal{M}, D)$, as the set of ground atoms

$$\bigcup_{m \in \mathcal{M}} Retr(m, D)$$

Finally, given an ontology predicate A, we define the *extension of* A *retrieved by* $\mathcal{M}$ *on* D, denoted by $Retr(A, \mathcal{M}, D)$, as the set $\{t \mid A(t) \in Retr(\mathcal{M}, D)\}$.

From now on, without loss of generality we assume that different mapping assertions use different sets of variable symbols.

3 Mapping Analysis Tasks

In this section we provide the formal definitions that constitute the basis of the mapping analysis functionalities that will be studied in Section 4. We first deal with mapping consistency, then we turn our attention to mapping redundancy and subsumption. If not otherwise specified, definitions and properties given in this section apply to mappings that contain general assertions of the form (1).

3.1 Consistency

We start by providing some notions of inconsistency relative to a single mapping assertion. Informally, with such notions we characterize the anomalous situations in which either the query in the head of an assertion has certainly an empty evaluation in every model for the ontology $\mathcal{O}$ (we call this situation head-inconsistency), or the query in the body of an assertion has certainly an empty evaluation in every model for the source schema $\mathcal{S}$ (we call this situation body-inconsistency).

Definition 1. *(mapping head-inconsistency) Let* $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ *be an OBDA specification and* $m : \phi(\boldsymbol{x}) \rightsquigarrow \psi(\boldsymbol{x})$ *be a mapping assertion in* $\mathcal{M}$*. We say that* m is head-inconsistent for $\langle \mathcal{O}, \mathcal{S} \rangle$ *if* $\mathcal{O} \models \forall \boldsymbol{x}.(\neg \psi(\boldsymbol{x}))$.

Example 3. Let $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification where $\mathcal{O}$ and $\mathcal{S}$ are as in Example 1. Suppose that the mapping $\mathcal{M}$ contains the following assertion:

$$m : \texttt{SELECT ANM\_CODE AS X FROM ANM\_TAB} \rightsquigarrow \mathsf{Lion}(X) \wedge \mathsf{Monkey}(X)$$

Then, m is head-inconsistent for $\langle \mathcal{O}, \mathcal{S} \rangle$, since we have that $\mathcal{O} \models \mathsf{Lion} \sqsubseteq \neg\mathsf{Monkey}$. □

Definition 2. *(mapping body-inconsistency) Let $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification and $m : \phi(\boldsymbol{x}) \rightsquigarrow \psi(\boldsymbol{x})$ be a mapping assertion in $\mathcal{M}$. We say that m* is body-inconsistent for *$\langle \mathcal{O}, \mathcal{S} \rangle$ if $\mathcal{S} \models \forall \boldsymbol{x}.(\neg \phi(\boldsymbol{x}))$.*

Example 4. Let $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification where $\mathcal{O}$ and $\mathcal{S}$ are as in Example 1. Suppose that the mapping $\mathcal{M}$ contains the following mapping assertion:

$$m : \texttt{SELECT ANM\_CODE AS X FROM ANM\_TAB WHERE BREED = 'Lion' AND BREED = 'Monkey'} \rightsquigarrow \mathsf{Animal}(X)$$

Since, obviously, for every tuple in ANM_TAB the attribute BREED can assume only a single value, we can easily conclude that m is body-inconsistent for $\langle \mathcal{O}, \mathcal{S} \rangle$. □

We compose the above two notions into the following notion of inconsistency of a single mapping assertion.

Definition 3. *(mapping inconsistency) Let $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification and m be a mapping assertion in $\mathcal{M}$. We say that m* is inconsistent for *$\langle \mathcal{O}, \mathcal{S} \rangle$ if m is head-inconsistent or body-inconsistent for $\langle \mathcal{O}, \mathcal{S} \rangle$.*

Then, we provide a "global" notion of inconsistency, that is, inconsistency relative to a whole mapping specification. To this aim, we first need to define when a mapping is active on a source instance.

We say that a mapping $\mathcal{M}$ *is active on* a source instance D if, for every mapping assertion $m : \phi(\boldsymbol{x}) \rightsquigarrow \psi(\boldsymbol{x})$ in $\mathcal{M}$, $\mathcal{I}_D \models \exists \boldsymbol{x}.\phi(\boldsymbol{x})$ (in other words, every mapping assertion is "activated" by D and retrieves at least one tuple from D).

Definition 4. *(global mapping inconsistency) Let $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification. We say that $\mathcal{M}$* is globally inconsistent for *$\langle \mathcal{O}, \mathcal{S} \rangle$ if there does not exist a source instance D legal for $\mathcal{S}$ such that $\mathcal{M}$ is active on D and $Models(\mathcal{J}, D) \neq \emptyset$.*

Intuitively, if a mapping is globally inconsistent, then it is not possible to simultaneously activate all its mapping assertions without causing inconsistency of the whole specification. This is certainly an anomalous situation, as shown by the following example.

Example 5. Let $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification where $\mathcal{O}$ and $\mathcal{S}$ are as in Example 1. Suppose that $\mathcal{M}$ contains the following mapping assertions:

$$m1 : \texttt{SELECT ANM\_CODE AS X FROM ANM\_TAB} \rightsquigarrow \mathsf{Lion}(X)$$
$$m2 : \texttt{SELECT ANM\_CODE AS X FROM ANM\_TAB} \rightsquigarrow \mathsf{Monkey}(X)$$

It is easy to see that $\mathcal{M}$ is globally inconsistent for $\langle \mathcal{O}, \mathcal{S} \rangle$. □

The following property relates the two notions of mapping inconsistency and global mapping inconsistency.

Proposition 1. *Let $\langle \mathcal{O}, \mathcal{S}, \{m\}\rangle$ be a OBDA specification. If the mapping assertion m is inconsistent for $\langle \mathcal{O}, \mathcal{S}\rangle$, then every mapping $\mathcal{M}$ that contains m is globally inconsistent for $\langle \mathcal{O}, \mathcal{S}\rangle$.*

Note that a mapping $\mathcal{M}$ that is globally inconsistent for some $\langle \mathcal{O}, \mathcal{S}\rangle$ may not contain any mapping assertion m that is inconsistent for $\langle \mathcal{O}, \mathcal{S}\rangle$, which is actually the case shown in Example 5. In other terms, inconsistency of a mapping assertion is a sufficient but not necessary condition for global inconsistency.

3.2 Redundancy and Subsumption

We now deal with mapping redundancy and subsumption. First, given an ODBA specification $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ where $\mathcal{M} = \{m\}$, we consider a mapping assertion m' to be redundant for m, if adding m' to $\mathcal{M}$ produces a specification equivalent to $\mathcal{J}$. This is formalized below.

Definition 5. *(mapping redundancy) Let $\mathcal{O}$ be an ontology, let $\mathcal{S}$ be a source schema, and let m, m' be mapping assertions of the same arity. We say that m'* is redundant for m under $\langle \mathcal{O}, \mathcal{S}\rangle$ *if, for every source instance D that is legal for $\mathcal{S}$, $\mathit{Models}(\langle \mathcal{O}, \mathcal{S}, \{m\}\rangle, D) = \mathit{Models}(\langle \mathcal{O}, \mathcal{S}, \{m, m'\}\rangle, D)$.*

Our aim now is to characterize the above notion of redundancy in terms of composition of separate entailment checks on the source schema level and the ontology level of the OBDA specification. We thus define the notions of head-subsumption and body-subsumption for a pair of mapping assertions.

Definition 6. *(mapping body-subsumption, mapping head-subsumption) Let $\mathcal{S}$ be a source schema, let m_1, m_2 be mapping assertions of the same arity, let $\mathit{FR}(m_2) = \{x_1, \ldots, x_n\}$, and let μ be a bijective mapping from $\mathit{FR}(m_1)$ to $\mathit{FR}(m_2)$. We say that m_1* body-subsumes m_2 under $\mathcal{S}$ and μ *if the schema $\mathcal{S}$ entails the sentence $\forall x_1, \ldots, x_n(\mathit{body}(m_2) \rightarrow \mu(\mathit{body}(m_1)))$. Moreover, we say that m_1* head-subsumes m_2 under $\mathcal{O}$ and μ *if the ontology $\mathcal{O}$ entails the sentence $\forall x_1, \ldots, x_n(\mathit{head}(m_2) \rightarrow \mu(\mathit{head}(m_1)))$.*

Informally, body-subsumption characterizes the case when the body of the mapping assertion m_2 entails the body of m_1 under the schema $\mathcal{S}$ and under a mapping μ of the frontier variables of m_1 and m_2. Head-subsumption is defined in an analogous way.

Example 6. Consider again the ontology $\mathcal{O}$ and the schema $\mathcal{S}$ of Example 1 and the following mapping assertions:

$$
\begin{array}{lll}
m1: \texttt{SELECT AREA\_CODE AS X, SIZE AS Y} & \rightsquigarrow & \mathsf{size}(X,Y) \\
\quad\;\; \texttt{FROM AREA\_TAB} & & \\
m2: \texttt{SELECT AREA\_CODE AS X, SIZE AS Y} & \rightsquigarrow & \mathsf{Area}(X) \wedge \mathsf{size}(X,Y) \\
\quad\;\; \texttt{FROM AREA\_TAB WHERE SIZE > 10} & & \\
m3: \texttt{SELECT ANM\_CODE AS X} & \rightsquigarrow & \mathsf{Animal}(X) \wedge \mathsf{name}(X,Y) \\
\quad\;\; \texttt{FROM ANM\_TAB WHERE BREED = 'Monkey'} & & \\
m4: \texttt{SELECT ANM\_CODE AS X, NAME AS Y} & \rightsquigarrow & \mathsf{Lion}(X) \wedge \mathsf{name}(X,Y) \\
\quad\;\; \texttt{FROM ANM\_TAB WHERE BREED = 'Lion'} & &
\end{array}
$$

It is easy to see that $m1$ body-subsumes $m2$. Moreover, since the ontology $\mathcal{O}$ entails that a lion is an animal, we have that $m3$ head-subsumes $m4$. □

The relationship between the notion of redundancy and the notions of head- and body-subsumption is stated by the following proposition.

Proposition 2. *Let $\mathcal{O}$ be an ontology, let $\mathcal{S}$ be a source schema, and let m, m' be mapping assertions of the same arity. Then, m' is redundant for m under $\langle \mathcal{O}, \mathcal{S} \rangle$ iff there exists a bijective mapping $\mu : FR(m) \to FR(m')$ such that m body-subsumes m' under $\mathcal{S}$ and μ and m' head-subsumes m under $\mathcal{O}$ and μ.*

Notice that, for m' to be redundant for m, we require that (under the same bijective mapping of the frontier variables) m body-subsumes m', whereas m' head-subsumes m. This indeed reflects the "semantic flow" of the data: m' is redundant since it retrieves from the sources less data than m, and at the same time the instantiation of ontology predicates that m' realizes with these data is less specific than the instantiation due to m, but implies it under $\mathcal{O}$.

Example 7. Let $\mathcal{O}$ and $\mathcal{S}$ be respectively the ontology and the source schema of Example 1. Consider the following mapping assertions:

$$
\begin{array}{lll}
m1: \texttt{SELECT ANM\_CODE AS X, NAME AS Y} & \rightsquigarrow & \mathsf{name}(X,Y) \\
\quad\;\; \texttt{FROM ANM\_TAB WHERE BREED = 'Monkey'} & & \\
m2: \texttt{SELECT ANM\_CODE AS X, NAME AS Y} & \rightsquigarrow & \mathsf{Animal}(X) \wedge \mathsf{name}(X,Y) \\
\quad\;\; \texttt{FROM ANM\_TAB} & &
\end{array}
$$

We have that $m1$ is redundant for $m2$ under $\langle \mathcal{O}, \mathcal{S} \rangle$. Indeed, it easy to see that $m2$ body-subsumes $m1$ under $\mathcal{S}$ and that $m1$ head-subsumes $m2$ under $\mathcal{O}$. Notice that, if we add the atom $\mathsf{Monkey}(X)$ in the head of $m1$, the redundancy does no longer hold, since in that case $m2$ head-subsumes $m1$. □

Then, we define a more general, global notion of mapping redundancy which is relative to a whole mapping specification.

Definition 7. *(global mapping redundancy) Let $\mathcal{O}$ be an ontology, let $\mathcal{S}$ be a source schema, and let $\mathcal{M}, \mathcal{M}'$ be mappings. We say that $\mathcal{M}'$* is globally redundant for $\mathcal{M}$ under $\langle \mathcal{O}, \mathcal{S} \rangle$ *if, for every source instance D that is legal for $\mathcal{S}$, $Models(\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle, D) = Models(\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \cup \mathcal{M}' \rangle, D)$.*

Notice that global redundancy of a mapping $\mathcal{M}'$ for a mapping $\mathcal{M}$ under $\langle \mathcal{O}, \mathcal{S} \rangle$ does not imply that there exists an assertion m' in $\mathcal{M}'$ and an assertion m in $\mathcal{M}$ such that m' is redundant for m under $\langle \mathcal{O}, \mathcal{S} \rangle$, as shown below.

Example 8. Consider the ontology $\mathcal{O} = \{A_1 \sqsubseteq A, B_1 \sqsubseteq B\}$, the source schema composed by the only unary predicate Q, and the following mapping assertions:

$$\begin{array}{lcl} m_1 : Q(X) & \rightsquigarrow & A_1(X) \\ m_2 : Q(X) & \rightsquigarrow & B_1(X) \\ m_3 : Q(X) & \rightsquigarrow & A(X) \wedge B(X) \end{array}$$

Then, $\mathcal{M}' = \{m_3\}$ is globally redundant for $\mathcal{M} = \{m_1, m_2\}$ under $\langle \mathcal{O}, \mathcal{S} \rangle$, but m_3 is not redundant under $\langle \mathcal{O}, \mathcal{S} \rangle$ for any mapping assertion in $\mathcal{M}$. $\square$

Conversely, it is easy to see the if a mapping $\mathcal{M}'$ contains only assertions that, taken one by one, are redundant under $\langle \mathcal{O}, \mathcal{S} \rangle$ for some assertion contained in a mapping $\mathcal{M}$, then $\mathcal{M}'$ is globally redundant for $\mathcal{M}$ under $\langle \mathcal{O}, \mathcal{S} \rangle$.

Finally, we define extensional predicate subsumption, a mapping-based notion of subsumption between ontology predicates. Differently from all the other definitions and propositions given in this section, such notion applies only to GAV mappings, and thus we give it for *FO2DCQ* mappings, which subsume all GAV mappings considered in this paper.

Definition 8. *(extensional predicate subsumption and emptiness) Let $\mathcal{S}$ be a source schema, let $\mathcal{M}$ be a FO2DCQ mapping, and let A, A' be ontology predicates having the same arity. We say that A* extensionally subsumes *A'* under *$\langle \mathcal{S}, \mathcal{M} \rangle$ if, for every source instance D that is legal for $\mathcal{S}$, $Retr(A, \mathcal{M}, D) \supseteq Retr(A', \mathcal{M}, D)$. Moreover, given a predicate A, we say that A* is extensionally empty under *$\langle \mathcal{S}, \mathcal{M} \rangle$ if, for every source instance D that is legal for $\mathcal{S}$, $Retr(A, \mathcal{M}, D) = \emptyset$.*

Informally, the above notion of extensional predicate subsumption checks containment of the instances of the predicates retrieved by the mapping on every legal source instance.

4 Verification

In this section we study the problem of decidability of the verification of the formal properties of mappings defined in Section 3. It can be immediately observed that verification for mappings expressed in the language *FO2DCQ* poses a serious decidability issue independently of the ontology language $\mathcal{L}_{\mathcal{O}}$ and the source schema language $\mathcal{L}_{\mathcal{S}}$, since arbitrary FOL expressions can appear in the body of such mapping assertions. Therefore, our first analysis focuses on identifying sufficient conditions for the decidability of the verification of the properties under examination.

For ease of exposition, in the rest of this section we assume that the ontology language $\mathcal{L}_{\mathcal{O}}$ has a predefined empty predicate $\perp$. More precisely, we assume the existence of an ontology predicate $\perp$ of arity 0 that is false in every interpretation.

4.1 Head-Subsumption and Head-Inconsistency

Let us consider mapping head-subsumption. In the following, let m_1, m_2 be either *FO2DCQ* or *CQ2DCQ* mapping assertions of the same arity, let μ be a bijective mapping from $FR(m_1)$ to $FR(m_2)$, and let $q_1 = head(m_1)$, $q_2 = head(m_2)$. The following algorithm checks whether m_2 head-subsumes m_1 under $\mathcal{S}$ and μ:

1. freeze query $\mu(q_1)$, i.e., generate a source instance (set of ground atoms) $D_{\mu(q_1)}$ from q_1 by replacing every occurrence of a variable x with a constant symbol c_x;
2. let q_2' be the formula obtained from q_2 by replacing every occurrence of a variable x with a constant symbol c_x. Notice that q_2' is a conjunction of ground atoms;
3. if, for every ground atom α in q_2', $\mathcal{O} \cup D_{\mu(q_1)} \models \alpha$ (ground atom entailment problem in $\mathcal{L}_{\mathcal{O}}$), then return true, otherwise return false.

It can be shown that the above algorithm is correct. This implies that mapping head-subsumption is decidable as soon as ground atom entailment in $\mathcal{L}_{\mathcal{O}}$ is decidable. Conversely, undecidability of head-subsumption when ground atom entailment in $\mathcal{L}_{\mathcal{O}}$ is undecidable can be shown by an easy reduction of ground atom entailment in $\mathcal{L}_{\mathcal{O}}$ to mapping head-subsumption. Consequently, the following property holds.

Theorem 1. *For both FO2DCQ mappings and CQ2DCQ mappings, mapping head-subsumption is decidable iff ground atom entailment in $\mathcal{L}_{\mathcal{O}}$ is decidable.*

Mapping head-inconsistency can be immediately reduced to mapping head-subsumption, since $\mathcal{L}_{\mathcal{O}}$ allows for the empty predicate $\perp$. Then, m is head-inconsistent for $\langle \mathcal{O}, \mathcal{S} \rangle$ iff m is head-subsumed by m' under $\langle \mathcal{O}, \mathcal{S} \rangle$ and μ, where m' is the mapping obtained from m by adding the atom $\perp$ in the head of m, and μ is the identity mapping on $FR(m)$.

Moreover, it can be shown that ground atom entailment can be reduced to mapping head-inconsistency, under some assumptions on the ontology language $\mathcal{L}_{\mathcal{O}}$. In particular, we say that $\mathcal{L}_{\mathcal{O}}$ allows for binary denial formulas if, for every pair of predicate names p, p' in $\Sigma_{\mathcal{O}}$ of the same arity, the formula $\forall \boldsymbol{x}\, (p(\boldsymbol{x}) \wedge p'(\boldsymbol{x}) \rightarrow \perp)$ belongs to $\mathcal{L}_{\mathcal{O}}$. The above assumption is a sufficient condition for reducing ground atom entailment to head-inconsistency.

Theorem 2. *For both FO2DCQ mappings and CQ2DCQ mappings, mapping head-inconsistency is decidable if ground atom entailment in $\mathcal{L}_{\mathcal{O}}$ is decidable. Moreover, if $\mathcal{L}_{\mathcal{O}}$ allows for binary denial formulas, then ground atom entailment in $\mathcal{L}_{\mathcal{O}}$ is decidable if mapping head-inconsistency is decidable.*

4.2 Body-Subsumption and Body-Inconsistency

Body-subsumption and body-inconsistency are undecidable for *FO2DCQ* mappings (due to the undecidability of the validity problem in FOL).

Concerning *CQ2DCQ* mappings, the following property immediately follows from the definitions of mapping body-subsumption.

Theorem 3. *For CQ2DCQ mappings, mapping body-subsumption is decidable iff conjunctive query containment is decidable in $\mathcal{L}_{\mathcal{S}}$.*

Notice that several schema languages are known to satisfy the hypothesis of the above theorem. E.g., conjunctive query containment is decidable in the language of non-key-conflicting keys and inclusion dependencies studied in [5], as well as in several classes of TGDs [4, 15].

For mapping body-inconsistency, a similar property holds under some sufficient assumptions on the language $\mathcal{L}_{\mathcal{S}}$. In particular, we say that $\mathcal{L}_{\mathcal{S}}$ allows for CQ-denial formulas if, for every conjunctive query $q(\boldsymbol{x})$ over $\Sigma_{\mathcal{S}}$, the formula $\forall \boldsymbol{x}(q(\boldsymbol{x}) \rightarrow \bot)$ belongs to $\mathcal{L}_{\mathcal{S}}$.

Theorem 4. *For CQ2DCQ mappings, mapping body-inconsistency is decidable iff conjunctive query containment is decidable in $\mathcal{L}_{\mathcal{S}}$. Moreover, if $\mathcal{L}_{\mathcal{S}}$ allows for CQ-denial formulas, then conjunctive query containment in $\mathcal{L}_{\mathcal{S}}$ is decidable if mapping body-inconsistency is decidable.*

4.3 Redundancy and Inconsistency

Given the above undecidability results for head- and body-subsumption, it obviously follows that both redundancy and inconsistency of mapping assertions are undecidable properties for *FO2DCQ* mappings.

However, the situation is different for *CQ2DCQ* mappings, In fact, it is immediate to see that Proposition 2, Definition 3, Theorem 1, and Theorem 3, imply the following properties.

Theorem 5. *For CQ2DCQ mappings, mapping redundancy is decidable iff ground atom entailment is decidable in $\mathcal{L}_{\mathcal{O}}$ and conjunctive query containment is decidable in $\mathcal{L}_{\mathcal{S}}$.*

Theorem 6. *For CQ2DCQ mappings, mapping inconsistency is decidable if ground atom entailment is decidable in $\mathcal{L}_{\mathcal{O}}$ and conjunctive query containment is decidable in $\mathcal{L}_{\mathcal{S}}$. Moreover, if $\mathcal{L}_{\mathcal{S}}$ allows for CQ-denial formulas and $\mathcal{L}_{\mathcal{O}}$ allows for binary denial formulas, then ground atom entailment is decidable in $\mathcal{L}_{\mathcal{O}}$ and conjunctive query containment is decidable in $\mathcal{L}_{\mathcal{S}}$ if mapping inconsistency is decidable.*

4.4 Extensional Subsumption and Emptiness

We start by relating the notion of extensional predicate subsumption with the notion of global mapping redundancy.

Let $\mathcal{M}$ be a mapping and let A be a predicate name. We define $\mathcal{M}_A$ as the mapping obtained from $\mathsf{Split}(\mathcal{M})$ by considering only the mapping assertions in which A occurs in the head. Moreover, given such a mapping $\mathcal{M}_A$ and a predicate name B of the same arity as A, we define $\mathcal{M}_A(B)$ as the mapping obtained from $\mathcal{M}_A$ by replacing every occurrence of the predicate A with B.

The relationship between global mapping redundancy and extensional predicate subsumption is stated by the following property.

Theorem 7. *Let A, A' be ontology predicates of the same arity. Then, A extensionally subsumes A' under $\langle \mathcal{S}, \mathcal{M} \rangle$ iff $\mathcal{M}_{A'}(A)$ is redundant for $\mathcal{M}_A$ under $\langle \emptyset, \mathcal{S} \rangle$.*

From the above theorem, it can be easily verified that extensional predicate subsumption is a generalization of the the notion of concept (and role) inclusion in *extensional constraints* (also known as *ABox dependencies*) studied in Description Logics [21, 23, 8].

As shown by the previous theorem, extensional predicate subsumption reduces to a special case of global mapping redundancy, in which the ontology is empty. Under this simplification, it can be easily verified that this task can be reduced to a containment check between two unions of conjunctive queries (UCQs) in the language $\mathcal{L}_{\mathcal{S}}$. Consequently, the following property holds.

Theorem 8. *For CQ2DCQ mappings, extensional subsumption is decidable iff UCQ containment is decidable in $\mathcal{L}_{\mathcal{S}}$.*

Furthermore, it is immediate to verify that, for *FO2DCQ* mappings, extensional subsumption (as well as extensional emptiness) is undecidable.

4.5 Global Mapping Inconsistency

Given the above undecidability results, it immediately follows that, for *FO2DCQ* mappings, verifying global mapping inconsistency is undecidable.

For *CQ2DCQ* mappings, we present a technique that is able to decide global inconsistency in the case when the source schema $\mathcal{S}$ is empty.

Let $\mathcal{M}$ be a *CQ2DCQ* mapping and let $\mathcal{C}_{\mathcal{M}}$ be any set of constant symbols whose arity is the same as the number of variable symbols occurring in the bodies of the mapping assertions in $\mathcal{M}$. We call *grounding of $\mathcal{M}$ over $\mathcal{C}_{\mathcal{M}}$* any mapping obtained from $\mathcal{M}$ by replacing, in every mapping assertion, every variable symbol with a constant from $\mathcal{C}_{\mathcal{M}}$.

Given such a grounding $\mathcal{M}_G$ of $\mathcal{M}$, let $D(\mathcal{M}_G)$ be the source instance containing all the ground atoms that occur in the bodies of the mapping assertions of $\mathcal{M}_G$.

Theorem 9. *Given an OBDA specification $\langle \mathcal{O}, \emptyset, \mathcal{M} \rangle$ where $\mathcal{M}$ is a CQ2DCQ mapping, $\mathcal{M}$ is globally inconsistent for $\langle \mathcal{O}, \emptyset \rangle$ iff there exists no grounding $\mathcal{M}_G$ of $\mathcal{M}$ over $\mathcal{C}_{\mathcal{M}}$ such that $\mathcal{O} \cup \{Retr(\mathcal{M}, D(\mathcal{M}_G))\}$ is satisfiable.*

Notice that checking satisfiability of $\mathcal{O} \cup \{Retr(\mathcal{M}, D(\mathcal{M}_G))\}$ can be reduced to ground atom entailment, in particular, entailment of the ground atom $\perp$ with respect to the theory $\mathcal{O} \cup \{Retr(\mathcal{M}, D(\mathcal{M}_G))\}$. Therefore, from Theorem 9 it follows that decidability of global inconsistency is implied by decidability of ground atom entailment in $\mathcal{L}_{\mathcal{O}}$. For the other direction, the proof easily follows from Theorem 6 and from the fact that mapping inconsistency can be obviously reduced to global mapping inconsistency.

	$\mathcal{L}_\mathcal{S} \in$ *UCQ-dec*, arbitrary $\mathcal{L}_\mathcal{O}$	arbitrary $\mathcal{L}_\mathcal{S}$, $\mathcal{L}_\mathcal{O} \in$ *GAE-dec*	$\mathcal{L}_\mathcal{S} \in$ *UCQ-dec*, $\mathcal{L}_\mathcal{O} \in$ *GAE-dec*
head-subsumption/inconsistency	U	D	D
body-subsumption/inconsistency	U	U	U
redundancy/inconsistency	U	U	U
ext. subsumption/emptiness	U	U	U
global inconsistency	U	U	U

Results for *FO2DCQ* mappings (D=decidable, U=undecidable).

	$\mathcal{L}_\mathcal{S} \in$ *UCQ-dec*, arbitrary $\mathcal{L}_\mathcal{O}$	arbitrary $\mathcal{L}_\mathcal{S}$, $\mathcal{L}_\mathcal{O} \in$ *GAE-dec*	$\mathcal{L}_\mathcal{S} \in$ *UCQ-dec*, $\mathcal{L}_\mathcal{O} \in$ *GAE-dec*
head-subsumption/inconsistency	U	D	D
body-subsumption/inconsistency	D	U	D
redundancy/inconsistency	U	U	D
ext. subsumption	D	U	D
global inconsistency*	U	D	D

Results for *CQ2DCQ* mappings
(D=decidable, U=undecidable, *=The result holds only when $\mathcal{S}$ is empty).

Fig. 1. Summary of decidability/undecidabilty results.

Theorem 10. *For CQ2DCQ mappings and empty source schemas, global mapping inconsistency is decidable if ground atom entailment is decidable in $\mathcal{L}_\mathcal{O}$. Moreover, if $\mathcal{L}_\mathcal{S}$ allows for CQ-denial formulas and $\mathcal{L}_\mathcal{O}$ allows for binary denial formulas, then ground atom entailment is decidable in $\mathcal{L}_\mathcal{O}$ and conjunctive query containment is decidable in $\mathcal{L}_\mathcal{S}$ if global mapping inconsistency is decidable.*

The results shown in this section are summarized in Figure 1. The figure reports two tables: the first one is relative to the *FO2DCQ* mappping language, while the second one is relative to the *CQ2DCQ* mapping language. In the two tables, we denote by *UCQ-dec* the class of FO languages for which UCQ containment is decidable, and denote by *GAE-dec* the class of FO languages for which entailment of ground atoms is decidable. We remark that the undecidability results for head-inconsistency hold under the assumption that $\mathcal{L}_\mathcal{O}$ allows for binary denial formulas (Theorem 2); moreover, for *CQ2DCQ* mappings, the undecidability results for body-inconsistency hold under the assumption that $\mathcal{L}_\mathcal{S}$ allows for CQ-denial formulas (Theorem 4), and the undecidability results for mapping inconsistency and global mapping inconsistency hold under the assumption that $\mathcal{L}_\mathcal{O}$ allows for binary denial formulas and $\mathcal{L}_\mathcal{S}$ allows for CQ-denial formulas (Theorem 6 and Theorem 9).

5 Beyond GAV Mappings

In this section we draw some initial considerations on extending our analysis towards mapping languages beyond GAV. As we have seen, GAV-like mappings enjoy useful properties when it comes to mappings analysis. However, there are cases in OBDA systems where GAV-like mappings are insufficient. For example, consider the simple case of relating the answers of a database query Q to the existential restriction of a role R in a Description Logic ontology. With GAV mappings, the only way to do so is to map Q to a new concept A in the ontology, and add to the ontology the concept inclusion axiom $A \sqsubseteq \exists R$. This may not be desirable, as it clutters the ontology with concepts that may have little relation to the domain being described.

In this section, we therefore consider the languages obtained from *CQ2DCQ* and *FO2DCQ* by allowing existential variables to occur in the heads of mapping assertions. Doing so without restriction gives us the languages *FO2CQ* and *CQ2CQ*, that is, the head of a mapping assertion is simply a conjunctive query over $\mathcal{O}$.

Unfortunately, such an increased expressiveness causes computational complications: for instance, for these two mapping languages the task of query unfolding is equivalent to query answering using views [17], which is much harder than unfolding with GAV-like mappings [14]. Furthermore, given such a mapping $\mathcal{M}$, it cannot be rephrased into a set of mapping assertions with single atoms in the head, as existentially quantified variables may occur in multiple atoms. Thus, $\mathsf{Split}(\mathcal{M})$ does not yield an equivalent mapping. To address these issues, we consider the languages *FO2CQE* and *CQ2CQE*, where for each $m \in \mathcal{M}$, $\mathit{head}(m)$ is a conjunctive query over O, and every existential variable in $\mathit{head}(m)$ occurs in exactly one atom. These languages allow us to map queries to existential restrictions of roles, but avoid the difficulties discussed. For example, it is easy to verify that for a mapping $\mathcal{M}$ in either of these two languages, $\mathsf{Split}(\mathcal{M})$ produces an equivalent mapping.

For all four languages, all the definitions of inconsistency, subsumption and redundancy (with the exception of extensional predicate subsumption) provided by Section 3 apply, as well as Proposition 1 and Proposition 2. For *FO2CQ* and *CQ2CQ*, we have the following analogue of Theorem 1.

Theorem 11. *For both FO2CQ and CQ2CQ mappings, mapping head-subsumption is decidable iff conjunctive query containment is decidable in* $\mathcal{L}_{\mathcal{O}}$.

For *FO2CQE* and *CQ2CQE* we can do better. Since $\mathsf{Split}(\mathcal{M})$ produces an equivalent mapping in these languages, checking head-subsumption can be done atom by atom. As such, for these languages mapping head-subsumption is decidable iff containment of positive single-atom queries is decidable.

By a similar argument, it is possible to show that, in order to check global mapping inconsistency for these languages over empty source schemas, we likewise need entailment of ground atoms in $\mathcal{L}_{\mathcal{O}}$.

6 Conclusions

In this paper we have formally defined some properties of interest for the mapping component of an OBDA specification, and we have provided several (un)decidability results concerning the task of verifying such properties for some typical mapping languages.

Our study is still in its initial stage, and several further issues need to be investigated. In particular, we left as future work the study of global redundancy and of global inconsistency for OBDA specifications with a non-empty source schema. Furthermore, we intend to extend our analysis to forms of mapping that go beyond the GAV setting (e.g., consider LAV and GLAV [17]), for which we have only provided some preliminary discussion and results in Section 5. Also, we want to study verification of the various forms of subsumption, redundancy, and inconsistency introduced in this paper for concrete instantiations of both the $\mathcal{L}_\mathcal{O}$ and the $\mathcal{L}_\mathcal{S}$ languages, and characterize its computational complexity.

Finally, we notice that the analysis conducted in this work is based on a "classical" notion of equivalence and subsumption, i.e., equality/containment between the sets of models of two specifications. Recent work in the data exchange area [10, 11, 19] has studied alternative notions of equivalence for schema mappings. One possible extension of the present work is applying these alternative semantic approaches to the case of OBDA mappings.

Acknowledgments. We thank the anonymous reviewers for precious suggestions. This research has been partially supported by the EU under FP7 project Optique (grant n. FP7-318338).

References

1. N. Antonioli, F. Castanò, C. Civili, S. Coletta, S. Grossi, D. Lembo, M. Lenzerini, A. Poggi, D. F. Savo, and E. Virardi. Ontology-based data access: the experience at the Italian Department of Treasury. In *Proc. of the Industrial Track of CAiSE*, volume 1017 of *CEUR,* `ceur-ws.org`, pages 9–16, 2013.
2. M. Arenas, P. Barceló, L. Libkin, and F. Murlak. *Foundations of Data Exchange*. Cambridge University Press, 2014.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, 2007.
4. A. Calì, G. Gottlob, and T. Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. of Web Semantics*, 14:57–83, 2012.
5. A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of PODS*, pages 260–271, 2003.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *JAR*, 39(3):385–429, 2007.

7. C. Civili, M. Console, G. De Giacomo, D. Lembo, M. Lenzerini, L. Lepore, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, V. Santarelli, and D. F. Savo. MASTRO STUDIO: Managing ontology-based data access applications. *PVLDB*, 6:1314–1317, 2013.
8. M. Console, M. Lenzerini, R. Mancini, R. Rosati, and M. Ruzzi. Synthesizing extensional constraints in Ontology-Based Data Access. In *Proc. of DL*, volume 1014 of *CEUR,* `ceur-ws.org`, pages 628–639, 2013.
9. A. Doan, A. Y. Halevy, and Z. G. Ives. *Principles of Data Integration.* Morgan Kaufmann, 2012.
10. R. Fagin, P. G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In *Proc. of PODS*, pages 33–42, 2008.
11. G. Gottlob, R. Pichler, and V. Savenkov. Normalization and optimization of schema mappings. *VLDBJ*, 20(2):277–302, 2011.
12. V. Haarslev, K. Hidde, R. Möller, and M. Wessel. The RacerPro knowledge representation and reasoning system. *Semantic Web J.*, 3(3):267–277, 2012.
13. P. Haase, I. Horrocks, D. Hovland, T. Hubauer, E. Jiménez-Ruiz, E. Kharlamov, J. W. Klüwer, C. Pinkel, R. Rosati, V. Santarelli, A. Soylu, and D. Zheleznyakov. Optique system: Towards ontology and mapping management in OBDA solutions. In *Proc. of WoDOOM*, pages 21–32, 2013.
14. A. Y. Halevy. Answering queries using views: A survey. *VLDBJ*, 10(4):270–294, 2001.
15. M. König, M. Leclère, M.-L. Mugnier, and M. Thomazo. On the exploration of the query rewriting space with existential rules. In *Proc. of RR*, pages 123–137, 2013.
16. R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to ontology-based data access. In *Proc. of IJCAI*, pages 2656–2661, 2011.
17. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS*, pages 233–246, 2002.
18. M. Lenzerini. Ontology-based data management. In *Proc. of CIKM*, pages 5–6, 2011.
19. R. Pichler, E. Sallinger, and V. Savenkov. Relaxed notions of schema mapping equivalence revisited. *Theory of Computing Systems*, 52(3):483–541, 2013.
20. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
21. M. Rodriguez-Muro and D. Calvanese. High performance query answering over *DL-Lite* ontologies. In *Proc. of KR*, pages 308–318, 2012.
22. M. Rodriguez-Muro, R. Kontchakov, and M. Zakharyaschev. Ontology-based data access: Ontop of databases. In *Proc. of ISWC*, volume 8218 of *LNCS*, pages 558–573. Springer, 2013.
23. R. Rosati. Prexto: Query rewriting under extensional constraints in *DL-Lite.* In *Proc. of ESWC*, volume 7295 of *LNCS*, pages 360–374. Springer, 2012.
24. R. Shearer, B. Motik, and I. Horrocks. HermiT: A highly-efficient OWL reasoner. In *Proc. of OWLED*, volume 432 of *CEUR,* `ceur-ws.org`, 2008.
25. E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *J. of Web Semantics*, 5(2):51–53, 2007.
26. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of IJCAR*, pages 292–297, 2006.
27. T. Venetis, G. Stoilos, and G. B. Stamou. Query extensions and incremental query rewriting for OWL 2 QL ontologies. *J. on Data Semantics*, 3(1):1–23, 2014.