Semantic Analysis of R2RML Mappings for Ontology-Based Data Access

Cristina Civili[1], Jose Mora[2], Riccardo Rosati[2](✉), Marco Ruzzi[2], and Valerio Santarelli[2]

[1] School of Informatics, University of Edinburgh, Edinburgh, UK
[2] DIAG, Sapienza Università di Roma, Rome, Italy
rosati@dis.uniroma1.it

Abstract. Ontology-based data access (OBDA) deals with the problem of accessing autonomous data sources through a shared, virtual ontology, and declarative mappings connecting the data sources to the ontology. The W3C standard R2RML allows for mapping relational data sources to RDFS/OWL ontologies. In this paper, we present algorithms for the semantic analysis of R2RML mappings in the OBDA setting, when the ontology is expressed in OWL 2 QL. The focus of such algorithms is to identify the main semantical anomalies (inconsistency and redundancy) of a mapping specification with respect to the ontology and/or the data sources. Such algorithms have been implemented in the mapping analysis tool developed within the Optique European project. We also report on the experiments conducted within the Optique project use cases.

1 Introduction

Ontology-based data access (OBDA) [12] is an approach to the access of multiple, heterogeneous *data sources* through an *ontology* that acts as a shared, abstract model of the data, and a declarative *mapping* that provides the semantic relationship between the data and the ontology.

An *OBDA specification* is the intensional specification of an OBDA setting, i.e., a triple $\langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{T}$ is the ontology, $\mathcal{S}$ is the schema of the data sources and $\mathcal{M}$ is the mapping. In this paper, we focus on the case when $\mathcal{S}$ is a single relational database schema.

Our purpose is to identify algorithms for developing semantic mapping analysis functionalities in an OBDA platform. More precisely, we aim at developing functionalities that help in the construction and maintenance of the OBDA specification. In particular, the present work is motivated by the Optique European project[1] [6], whose aim is to apply OBDA technology in big data scenarios. The issue of creating, debugging and maintaining a mapping specification is a central one in this project, and tools for supporting the design and analysis of mappings are being developed within the project.

Indeed, the specification of the mapping is the most challenging and complex design activity in an OBDA project, since the mapping has to fill the semantic

[1] http://www.optique-project.eu/.

M. Ortiz and S. Schlobach (Eds.): RR 2016, LNCS 9898, pp. 25–38, 2016.
DOI: 10.1007/978-3-319-45276-0_3

distance between the ontology and the data sources, which is often very large. So, the declarative assertions constituting the mapping are very complex statements. Moreover, in the Optique use cases, as well as in other practical applications of the OBDA framework (see, e.g., [2]), the number of mapping assertions constituting the mapping is large (hundreds of assertions), and it is extremely difficult to manually handle and debug such a specification.

In this paper we present the mapping analysis component developed within the Optique project, to provide automated support to the specification and debugging of mappings in OBDA. We base our work (Sect. 3) on recent formal notions of *anomalous* mappings in the OBDA context [10,11]: in particular, notions of *inconsistent* and *redundant* mappings, defined both in a *local* and in a *global* version. The local notions refer to single mapping assertions, while the global ones are relative to a whole mapping collection (set of mapping assertions).

We remark that defining an appropriate notion of inconsistency for mappings in OBDA is already challenging, since the "classical" notion of inconsistency is not meaningful. We thus provide a notion of inconsistency for mappings (called *global mapping inconsistency*) that is based on the idea of checking whether the mapping can be "activated" by the data source without creating contradictions with the ontology. On the other hand, a "classical" notion of redundancy (that is, the one that naturally follows from the semantics of an OBDA system) appears appropriate for our purposes.

This formal framework allows us (Sect. 4) to attack the problem of defining concrete algorithms for semantic mapping analysis in OBDA. However, differently from [11], here we consider the W3C standard R2RML [5] as the mapping language. Such a language allows for expressing arbitrary SQL queries over the database source. This immediately makes almost every significant semantic check over R2RML mappings undecidable, independently of the ontology language (or equivalently, even if the ontology is empty). Nevertheless, we are able to define approximated techniques for semantic mapping analysis based on: (i) the translation of SQL into first-order logic; (ii) the usage of a first-order theorem prover to solve reasoning problems that encode the additional expressiveness of R2RML with respect to GAV and GLAV.

Finally, in Sect. 5 we present the experimental results obtained by our mapping analysis algorithms in the Optique project use cases.

2 Preliminaries

In the following, we assume to have four pairwise disjoint, countably infinite alphabets: an alphabet $\Gamma_{\mathcal{T}}$ of ontology predicates, an alphabet $\Gamma_{\mathcal{S}}$ of source schema predicates, an alphabet $\Gamma_{\mathcal{C}}$ of constants, and an alphabet $\Gamma_{\mathcal{F}}$ of functions.

Source schemas. A source schema $\mathcal{S}$ is a relational schema containing relations in $\Gamma_{\mathcal{S}}$, possibly equipped with integrity constraints (ICs). A *legal instance D for* $\mathcal{S}$ is a database for $\mathcal{S}$ (i.e., a finite set of ground atoms over $\mathcal{S}$ and the constants in $\Gamma_{\mathcal{C}}$) that satisfies the ICs of $\mathcal{S}$. We denote by $Const(D)$ the set of constants occurring in D.

We consider integrity constraints corresponding to first-order sentences. Given a source schema $\mathcal{S}$, we denote by $\Psi(\mathcal{S})$ the first-order sentence constituted by the conjunction of the sentences corresponding to its integrity constraints.

Given a first-order sentence α, we write $\mathcal{S} \models \alpha$ if for each database D legal for $\mathcal{S}$, $\mathcal{I}_D \models \alpha$, where $\mathcal{I}_D$ is the interpretation induced by D.

We call *simple schema* a source schema without ICs. We adopt standard notions for first-order (FO) queries and conjunctive queries (CQs) over relational schemas [1]. By a FO query over a source schema $\mathcal{S}$ we mean a FO query over the alphabet of $\mathcal{S}$. With $\phi(\boldsymbol{x})$ we denote a FO query with free variables $\boldsymbol{x}$. The number of variables in $\boldsymbol{x}$ is the arity of the query. A Boolean FO query is a FO query without free variables. Given a FO q over $\mathcal{S}$ and a legal instance D for $\mathcal{S}$, $\mathit{eval}(q, D)$ denotes the evaluation of q over D. In what follows, we will always denote a source schema with $\mathcal{S}$.

Ontologies. We consider ontologies expressed in the description logic *DL-Lite*$_R$ [4], the logic underlying the OWL 2 QL standard profile.[2] In particular, a *DL-Lite*$_R$ ontology $\mathcal{O}$ is a pair $\langle \mathcal{T}, A\rangle$, where $\mathcal{T}$ is the TBox and $\mathcal{A}$ is the ABox. In what follows, $\mathcal{O}$, $\mathcal{T}$, and $\mathcal{A}$, respectively, will always have the same meaning. As in the W3C standard OWL, we do not interpret ontologies under the Unique Name Assumption. We denote with $\mathit{Models}(\mathcal{O})$ the set of models of $\mathcal{O}$, and with $\mathcal{O} \models \alpha$ the fact that $\mathcal{O}$ entails a sentence α. Also, by *ontology inconsistency* we mean the task of deciding whether $\mathit{Models}(\mathcal{O}) = \emptyset$, and by *instance checking* the task of deciding whether $\mathcal{O} \models \beta$, where β is a ground atom. By *CQs over* $\mathcal{O}$ we mean CQs over the alphabet of the TBox of $\mathcal{O}$, and by *CQ entailment* the task of checking whether $\mathcal{O} \models q$, where q is a Boolean CQ.

Mappings. A *mapping assertion* m from a source schema $\mathcal{S}$ to a TBox $\mathcal{T}$ has the form

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

where $\phi(\boldsymbol{x})$ is a function-free first-order query with free variables $\boldsymbol{x}$ (and, possibly, existentially quantified variables) over the predicates of $\mathcal{S}$, and $\psi(\boldsymbol{x})$ is a conjunctive query with function symbols, i.e., a conjunction of atoms whose predicates are concepts and roles from $\mathcal{T}$ and whose arguments may be variables from $\boldsymbol{x}$, constants, or terms of the form $f(t_1, \ldots, t_n)$ where $n \geq 1$, $f \in \Gamma_{\mathcal{F}}$ and every t_i is either a variable from $\boldsymbol{x}$ or a constant. The free variables $\boldsymbol{x}$ are called the *frontier variables* of m, and denoted by $\mathit{FR}(m)$. Moreover, $\phi(\boldsymbol{x})$ is called the *body of* m (denoted by $\mathit{body}(m)$), and $\psi(\boldsymbol{x})$ is called the *head of* m (denoted by $\mathit{head}(m)$). The number of variables in $\boldsymbol{x}$ is the *arity* of the mapping assertion. A mapping $\mathcal{M}$ from $\mathcal{S}$ to $\mathcal{T}$ is a finite set of mapping assertions from $\mathcal{S}$ to $\mathcal{T}$. Hereinafter $\mathcal{M}$ will always denote a mapping.

The above defined mapping language is the one typically considered in OBDA [3,12], and captures almost all the R2RML W3C standard mapping language [5].

[2] http://www.w3.org/TR/owl2-profiles/.

We say that a mapping assertion m *is active on a source instance* D if $\mathit{eval}(\mathit{body}(m), D)$ is a non-empty set of tuples of constants. A mapping $\mathcal{M}$ is active on D if all its mapping assertions $m \in \mathcal{M}$ are active on D.

Without loss of generality, we assume that different mapping assertions use different variable symbols. A *freeze* of a set of atoms Γ is a set of ground atoms obtained from Γ by replacing every variable with a *fresh* distinct constant. In this paper, the freeze is always used in the context of a mapping $\mathcal{M}$, so it suffices to assume that fresh constants do not appear in $\mathcal{M}$. Different freezes of the same set of atoms are equal up to renaming of constants. Thus, in the following we assume, without loss of generality, that the freeze of a set of atoms Γ is unique and is obtained by replacing each variable occurrence x with a fresh constant c_x, and we denote it by $\mathit{Freeze}(\Gamma)$.

Given a mapping assertion m of arity n and an n-tuple of constants $\boldsymbol{t}$, we denote by $m(\boldsymbol{t})$ the mapping assertion obtained by replacing $\mathit{FR}(m)$ in m with the constants in $\boldsymbol{t}$.

OBDA Specifications. An OBDA specification is a triple $\mathcal{J} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$. The semantics of $\mathcal{J}$ is given with respect to a database instance D legal for $\mathcal{S}$: a model for $\mathcal{J}$ w.r.t. D is a FOL interpretation $\mathcal{I}$ over the alphabet $\Gamma_{\mathcal{T}} \cup \Gamma_{\mathcal{C}} \cup \Gamma_{\mathcal{F}}$ that satisfies both $\mathcal{T}$ and $\mathcal{M}$. Formally, we say that $\mathcal{I}$ satisfies the mapping $\mathcal{M}$ if for each assertion $m \in \mathcal{M}$ and each tuple of constants $\boldsymbol{t}$ such that $\boldsymbol{t} \in \mathit{eval}(\mathit{body}(m), D)$ we have that $\mathcal{I} \models \mathit{head}(m(\boldsymbol{t}))$. The set of models of $\mathcal{J}$ w.r.t. D is denoted with $\mathit{Models}(\mathcal{J}, D)$. Also, we use $(\mathcal{J}, D)$ to denote $\mathcal{J}$ with source instance D. We say that $(\mathcal{J}, D)$ *is inconsistent* if $\mathit{Models}(\mathcal{J}, D) = \emptyset$, and denote with $(\mathcal{J}, D) \models \alpha$ the entailment of a sentence α by $(\mathcal{J}, D)$.

Example 1. As an example of an OBDA specification, we consider a source schema $\mathcal{S}$ where the `plants` relation contains data on extraction facilities, while the `eZones` relation contains data on the areas used for oil and gas extraction. Below, the underlined attributes represent the keys of the relations.

$$\texttt{plants}(\underline{\texttt{id\_pl}},\texttt{pl\_typ},\texttt{id\_zn}) \qquad \texttt{eZones}(\underline{\texttt{id\_zn}},\texttt{zn\_typ})$$

The formula $\Psi(\mathcal{S})$ expressing the source schema $\mathcal{S}$ is the following:

$$(\forall x, y, z, y', z'.\, \texttt{plants}(x, y, z) \wedge \texttt{plants}(x, y', z') \rightarrow (y = y' \wedge z = z')) \wedge$$
$$(\forall x, y, y'.\, \texttt{eZones}(x, y) \wedge \texttt{eZones}(x, y') \rightarrow y = y')$$

The following *DL-Lite*$_R$ TBox models a very small portion of the domain of oil and gas production extracted from an ontology developed within the Optique project. In particular, the TBox focuses on the facilities (concept $\mathsf{Facility}$) used in the oil and gas extraction and on the geographical areas (concept Area) in which they are located (role $\mathsf{locatedIn}$). A marine area (concept $\mathsf{MarArea}$) is a subconcept of the concept Area.

$$\begin{aligned} \mathcal{T} = \{\ & \mathsf{Platform} \sqsubseteq \mathsf{Facility}, && \mathsf{MarArea} \sqsubseteq \mathsf{Area}, && \exists \mathsf{locatedIn} \sqsubseteq \mathsf{Facility}, \\ & \exists \mathsf{locatedIn}^- \sqsubseteq \mathsf{Area} && \mathsf{Facility} \sqcap \mathsf{Area} \sqsubseteq \bot\ \} \end{aligned}$$

The following is an example of a mapping $\mathcal{M}$ from $\mathcal{S}$ to $\mathcal{T}$:

$$\begin{array}{lll} m_1: & (\exists y.\, \mathtt{plants}(x,y,z)) & \rightsquigarrow\ \mathsf{Facility}(f(x)) \wedge \mathsf{locatedIn}(f(x),z) \\ m_2: & \mathtt{plants}(x',\text{`pl'},y') & \rightsquigarrow\ \mathsf{Platform}(p(x')) \\ m_3: & \mathtt{eZones}(z',\text{`mz'}) & \rightsquigarrow\ \mathsf{MarArea}(m(z')). \end{array}$$

□

3 Formal Notions of Mapping Anomalies

In this section we recall the formal framework of [10,11] that constitutes the basis of the mapping analysis functionalities that will be studied in the next section. We first deal with mapping consistency, then we turn our attention to mapping redundancy and subsumption. All the definitions of this section are taken from [11], with the exception of Definition 2.

3.1 Mapping Inconsistency

We start by providing a "global" notion of inconsistency, that is, inconsistency relative to a whole mapping specification.

Definition 1 (Global Mapping Inconsistency). *Let $\mathcal{J} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification. We say that $\mathcal{M}$* is globally inconsistent for $\langle \mathcal{T}, \mathcal{S} \rangle$ *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 2. Let $\mathcal{J} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification where $\mathcal{T}$ and $\mathcal{S}$ are as in Example 1. Suppose that the mapping $\mathcal{M}$ contains the following mapping assertions:

$$\begin{array}{lll} m_1: & (\exists y,z.\, \mathtt{plants}(x,y,z)) & \rightsquigarrow\ \mathsf{Area}(x) \\ m_2: & \mathtt{plants}(x',\text{`pl'},z') & \rightsquigarrow\ \mathsf{Platform}(x') \wedge \mathsf{locatedIn}(x',z') \end{array}$$

It is easy to see that $\mathcal{M}$ is globally inconsistent for $\langle \mathcal{T}, \mathcal{S} \rangle$, because $\mathcal{T} \models \mathsf{Platform} \sqcap \mathsf{Area} \sqsubseteq \bot$ and every activation of m_2 also activates m_1, thus implying $\mathsf{Platform}(x)$ and $\mathsf{Area}(x)$ for the same individual x. □

Then, we provide a novel notion of *strong* local mapping inconsistency.[3]

Definition 2 (Strong Local Mapping Inconsistency). *Let $\mathcal{T}$ be a TBox and let $\mathcal{S}$ be a source schema. We say that a mapping assertion m is* strongly locally inconsistent for $\langle \mathcal{T}, \mathcal{S} \rangle$ *if there does not exist a source instance D legal for $\mathcal{S}$ such that $\{m\}$ is active on D and Models$(\langle \mathcal{T}, \mathcal{S}, \{m\} \rangle, D) \neq \emptyset$.*

[3] This notion of *strong local inconsistency* is slightly different from the notion of *local inconsistency* presented in [11]: in particular, it can be shown that strong local consistency implies local consistency, while the converse in general does not hold.

In practice, the notion of strong local inconsistency corresponds to check the inconsistency of a single mapping assertion with respect to $\langle \mathcal{T}, \mathcal{S} \rangle$.

Note that the strong local mapping inconsistency of $m \in \mathcal{M}$ for $\langle \mathcal{T}, \mathcal{S} \rangle$ implies the global mapping inconsistency of $\mathcal{M}$ for $\langle \mathcal{T}, \mathcal{S} \rangle$. On the other hand, a mapping $\mathcal{M}$ that is globally inconsistent for some $\langle \mathcal{T}, \mathcal{S} \rangle$ may not contain any mapping assertion m that is inconsistent for $\langle \mathcal{T}, \mathcal{S} \rangle$. That is, the strong local inconsistency of a mapping assertion is a sufficient but not necessary condition for global inconsistency.

3.2 Mapping Redundancy

We now deal with mapping redundancy. First, given an ODBA specification $\mathcal{J} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ 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 3 (Local Mapping Redundancy). *Let $\mathcal{T}$ be a TBox, let $\mathcal{S}$ be a source schema, and let m, m' be mapping assertions of the same arity from $\mathcal{S}$ to $\mathcal{T}$. We say that m'* is redundant for m under *$\langle \mathcal{T}, \mathcal{S} \rangle$ if, for every source instance D that is legal for $\mathcal{S}$, $Models(\langle \mathcal{T}, \mathcal{S}, \{m\} \rangle, D) = Models(\langle \mathcal{T}, \mathcal{S}, \{m, m'\} \rangle, D)$.*

Example 3. Let $\mathcal{T}$ and $\mathcal{S}$ be as in Example 1. Consider the following mapping assertions:

$$
\begin{array}{lll}
m_1: & \texttt{plants}(x, \text{'pl'}, z) & \rightsquigarrow \ \mathsf{locatedIn}(x, z) \\
m_2: & (\exists y.\, \texttt{plants}(x, y, z)) & \rightsquigarrow \ \mathsf{Facility}(x) \wedge \mathsf{locatedIn}(x, z)
\end{array}
$$

It is easy to see that the m_1 mapping assertion is locally redundant for m_2 under $\langle \mathcal{T}, \mathcal{S} \rangle$. □

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

Definition 4 (Global Mapping Redundancy). *Let $\mathcal{J} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification and let $\mathcal{M}'$ be a mapping from $\mathcal{S}$ to $\mathcal{T}$. We say that $\mathcal{M}'$* is globally redundant for *$\mathcal{J}$ if, for every source instance D that is legal for $\mathcal{S}$, $Models(\langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle, D) = Models(\langle \mathcal{T}, \mathcal{S}, \mathcal{M} \cup \mathcal{M}' \rangle, D)$.*

Example 4. Let $\langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDA specification, where $\mathcal{T}$ and $\mathcal{S}$ are as in Example 1, and $\mathcal{M}$ is as follows:

$$
\begin{array}{lll}
m_1: & (\exists y.\, \texttt{plants}(x, y, z) \wedge \texttt{eZones}(z, \text{'mz'})) & \rightsquigarrow \ \mathsf{locatedIn}(x, z) \\
m_2: & \texttt{eZones}(x', \text{'mz'}) & \rightsquigarrow \ \mathsf{MarArea}(x') \\
m_3: & \texttt{plants}(y', \text{'pl'}, z') \wedge \texttt{eZones}(z', \text{'mz'}) & \rightsquigarrow \ \mathsf{locatedIn}(y', z') \wedge \mathsf{Area}(z')
\end{array}
$$

Then, $\{m_3\}$ is globally redundant for $\langle \mathcal{T}, \mathcal{S}, \{m_1, m_2\} \rangle$. □

Notice that global redundancy of a mapping $\mathcal{M}'$ for a mapping $\mathcal{M}$ under $\langle \mathcal{T}, \mathcal{S} \rangle$ 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 $\langle \mathcal{T}, \mathcal{S} \rangle$, as shown below.

Example 5. Consider the ontology $\mathcal{T} = \{A_1 \sqsubseteq A, B_1 \sqsubseteq B\}$, the source schema composed by the only unary predicate Q, and the following mapping assertions:

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

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

Conversely, it is easy to see that if a mapping $\mathcal{M}'$ contains only assertions that, taken one by one, are redundant under $\langle \mathcal{T}, \mathcal{S}\rangle$ for some assertion contained in a mapping $\mathcal{M}$, then $\mathcal{M}'$ is globally redundant for $\mathcal{M}$ under $\langle \mathcal{T}, \mathcal{S}\rangle$.

Finally, we observe that local mapping redundancy is a special case of global mapping redundancy in which the mapping $\mathcal{M}$ and $\mathcal{M}'$ are both singleton.

4 Algorithms for the Optique System

The techniques for mapping analysis implemented within the Optique system are based on the construction of a matrix of ABox assertions. More precisely, given a mapping $\mathcal{M}$ relative to a source schema $\mathcal{S}$, we define the *ABox matrix for* $\mathcal{M}$ under source schema $\mathcal{S}$, and denote it by $AM(\mathcal{M}, \mathcal{S})$. We will then show that such an ABox matrix can be used to reduce all the mapping consistency and redundancy tasks defined in the previous section to standard DL ontology reasoning tasks (ontology consistency and instance checking).

First, we introduce some preliminary definitions.

The *(partial) grounding* g of the frontier variables of a mapping assertion m is a partial function from $FR(m)$ to a set of constants.

Given two groundings g_1 and g_2 for m, if g_1 is equal to g_2 on all the variables mapped by g_2 and there exists $x \in FR(m)$ that is mapped by g_1 and is not mapped by g_2, then we say that g_1 is preferred to g_2 for m.

Given a mapping assertion m, we denote by $\mathit{Freeze}_{FR}(m)$ the mapping assertion obtained from m by freezing of the frontier variables of m: more precisely, in $\mathit{Freeze}_{FR}(m)$ every occurrence of the frontier variable x is replaced by the constant c_x (w.l.o.g., we assume that different mapping assertions use different variable symbols, and that none of the c_x's appears in $\mathcal{M}$).

Finally, given a mapping assertion m, we denote by $\mathit{Const}_{FR}(m)$ the set of constants $\{c_x \mid x \in FR(m)\}$.

4.1 The Algorithm BuildABoxMatrix

We are now ready to present the algorithm that builds the ABox matrix $AM(\mathcal{M}, \mathcal{S})$:

```
Algorithm. BuildABoxMatrix($\mathcal{M}, \mathcal{S}$)
Input: mapping $\mathcal{M} = \{m_1, \dots, m_n\}$, source schema $\mathcal{S}$
Output: $AM(\mathcal{M}, \mathcal{S})$
begin
  for i:=1 to $n$ do
    for j:=1 to $n$ do begin
      $M[i,j] = \emptyset$;
      for each grounding $g : FR(m_j) \rightarrow Const_{FR}(m_i)$ such that
        (i) $\Psi(\mathcal{S}) \models body(Freeze_{FR}(m_j)) \rightarrow g(body(m_i))$
      and
        (ii) there exists no grounding $g' : FR(m_j) \rightarrow Const_{FR}(m_i)$
             such that $\Psi(\mathcal{S}) \models body(Freeze_{FR}(m_j)) \rightarrow g'(body(m_i))$
             and $g'$ is preferred to $g$ for $m$
      do $M[i,j] := M[i,j] \cup Freeze(head(g(m_i)))$
    end;
  return $M$
end
```

Informally, the ABox matrix M computed by the above algorithm BuildABoxMatrix($\mathcal{M}, \mathcal{S}$) is such that every cell $M[i,j]$ represents, through ABox assertions, *how m_i is activated by* m_j: every cell $M[i,j]$ is a set of ABox assertions that represent (using "frozen" individual names) the concept and role instances retrieved by the mapping assertion m_i when the mapping assertion m_j is active on any database instance D. More precisely, if (a projection of) the query in the body of assertion m_j is contained in (a projection of) the query in the body of assertion m_i (condition (i) in the algorithm), then any activation of m_j implies the activation of m_i: this is a crucial property both for mapping inconsistency and for mapping redundancy. The ABox matrix represents such semantic dependencies through ABox assertions that use the same individuals.

Example 6. Consider the following mapping $\mathcal{M}$ (on a simple source schema $\mathcal{S}$):

$$\begin{array}{l} m_1 : (\exists z.T_1(x,y,z) \wedge T_2(z,y) \wedge T_3(y,x)) \rightsquigarrow C(x) \wedge R(x,y) \\ m_2 : (\exists y', z'.T_1(x',y',z')) \rightsquigarrow D(x') \\ m_3 : (\exists z''.T_2(z'',y'') \wedge T_3(y'',x'')) \rightsquigarrow S(x'',y'') \end{array}$$

The ABox matrix M returned by the algorithm BuildABoxMatrix($\mathcal{M}, \mathcal{S}$) is as follows:

	1	2	3
1	$\{C(c_x), R(c_x, c_y)\}$		
2	$\{D(c_x)\}$	$\{D(c_{x'})\}$	
3	$\{S(c_x, c_y)\}$		$\{S(c_{x''}, c_{y''})\}$

In particular, the presence of the $D(c_x)$ in $M[2,1]$ encodes the fact that any activation of the mapping assertion m_1 implies that the mapping assertion m_2 is also activated (because the body query of m_1 is contained into the body query of m_2). Similarly, the presence of the $S(c_x, c_y)$ in $M[3,1]$ encodes the fact that any activation of the mapping assertion m_1 also implies the activation of mapping m_3 (because the body query of m_1 is contained into the body query of m_3).

4.2 Limits of the Algorithm

The algorithm BuildABoxMatrix and its implementation have two main limitations.

First, both check (i) and check (ii) in the above algorithm require to decide the validity of an arbitrary first-order sentence. This of course is an undecidable problem, so the above checks can only be approximated by our implementation of the algorithm. In particular, we have used the E theorem prover[4] to solve the above mentioned validity checks, using a time-out (which we configured in a range from 5 to 30 s) for every task.

In the case when no answer is provided within the time-out, our implementation assumes a "no" answer (i.e., no dependency between the two body queries). Therefore, some dependency between mapping assertions may be not represented by the ABox matrix returned by the algorithm. We believe that, given the goal of providing semantic support in the debugging phase of the mapping, this choice is better than assuming a "yes" answer in the cases not decided by the E prover, since in this case "false positives" would be produced then by the inconsistency and redundancy checks that make use of the ABox matrix.

Second, while the ABox matrix "materializes" (through concept and role instances) in a correct way the semantic relationship between two mapping assertions, there are more complex dependencies that are not captured by the matrix. For instance, consider the following mapping $\mathcal{M}$ (on a simple source schema $\mathcal{S}$):

$$\begin{aligned}
&m_1 : T_1(x, y) \rightsquigarrow R(x, y)\\
&m_2 : T_2(x', y') \rightsquigarrow S(x', y')\\
&m_3 : T_1(x'', y'') \wedge T_2(z'', w'') \rightsquigarrow P(x'', w'')
\end{aligned}$$

Here, the activation of a single mapping assertion does not imply the activation of any other mapping assertion. However, it is immediate to see that the activation of both m_1 and m_2 implies the activation of the assertion m_3. This is not captured by the ABox matrix, which only considers dependencies between single mapping assertions.

To overcome such an incompleteness, the algorithm should consider simultaneous activations of arbitrary subsets of mapping assertions: however, this would have a dramatic impact on the performance of the algorithm, since it would require an exponential number of iterations rather than a quadratic one.

We believe that such an incompleteness is in practice not problematic, since in real cases the probability of dealing with situations in which the analysis of simultaneous activations of multiple mapping assertions is required is very low.

[4] http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/E.html.

Therefore, due to both the above described limitations, the ABox matrix actually represents only a partial picture of the semantic dependencies among the mapping assertions. Despite such a limitation, we can still provide a significant semantic analysis of mappings.

4.3 Checking Mapping Inconsistency and Redundancy Through the ABox Matrix

We now show how the ABox matrix can be used to solve the mapping consistency and redundancy problems introduced in the previous section.

Strong Local Consistency. Let $\mathcal{T}$ be a TBox, let $\mathcal{S}$ be a source schema, let $\mathcal{M}$ be a mapping, let $|\mathcal{M}| = n$, let M be the matrix returned by the algorithm BuildABoxMatrix($\mathcal{M}, \mathcal{S}$), let i be an integer such that $1 \leq i \leq n$, and let $\mathcal{A}$ be the ABox defined as follows:

$$\mathcal{A} = M[i,i]$$

If the ontology $\langle \mathcal{T}, \mathcal{A} \rangle$ is inconsistent, then m_i is strongly inconsistent for $\langle \mathcal{T}, \mathcal{S} \rangle$.

Global Consistency. Let $\mathcal{T}$ be a TBox, let $\mathcal{S}$ be a source schema, let $\mathcal{M}$ be a mapping, let $|\mathcal{M}| = n$, let M be the matrix returned by the algorithm BuildABoxMatrix($\mathcal{M}, \mathcal{S}$) and let $\mathcal{A}$ be the ABox defined as follows:

$$\mathcal{A} = \bigcup_{i=1}^{n} \bigcup_{j=1}^{n} M[i,j]$$

If the ontology $\langle \mathcal{T}, \mathcal{A} \rangle$ is inconsistent, then $\mathcal{M}$ is globally inconsistent for $\langle \mathcal{T}, \mathcal{S} \rangle$.

Local Redundancy. Let $\mathcal{T}$ be a TBox, let $\mathcal{S}$ be a source schema, let $\mathcal{M}$ be a mapping, let $|\mathcal{M}| = n$, let M be the matrix returned by the algorithm BuildABoxMatrix($\mathcal{M}, \mathcal{S}$), and let i, j be integers such that $1 \leq i \leq n$ and $1 \leq j \leq n$. Now let $\mathcal{A}$ be the ABox defined as follows:

$$\mathcal{A} = M[j,i]$$

If $\langle \mathcal{T}, \mathcal{A} \rangle \models M[i,i]$, then m_i is redundant for m_j under $\langle \mathcal{T}, \mathcal{S} \rangle$.

Global Redundancy. Let $\mathcal{T}$ be a TBox, let $\mathcal{S}$ be a source schema, let $\mathcal{M}$ be a mapping, let $|\mathcal{M}| = n$, let M be the matrix returned by the algorithm BuildABoxMatrix($\mathcal{M}, \mathcal{S}$), let i be an integer such that $1 \leq i \leq n$, and let $\mathcal{M}' = \mathcal{M} \setminus \{m_i\}$. Now let $\mathcal{A}$ be the ABox defined as follows:

$$\mathcal{A} = \bigcup_{j \in \{1,\dots,i-1,i+1,\dots,n\}} M[j,i]$$

If $\langle \mathcal{T}, \mathcal{A} \rangle \models M[i,i]$, then m_i is globally redundant for $\langle \mathcal{T}, \mathcal{S}, \mathcal{M}' \rangle$.

Using the above properties, we have implemented algorithms based on the ABox matrix for both local and global mapping inconsistency and for both local and global mapping redundancy.

5 Experiments

The algorithms presented in this paper have been implemented as a novel mapping analysis component within the Optique European project, and, as all other components and APIs developed by the project partners, integrated on the Optique platform through the Information Workbench (IWB) [8]. IWB is a semantic data management and integration platform which provides a shared triple store for managing OBDA system assets, i.e., ontologies, mappings, database metadata, and queries.

Implementation of the mapping analysis component consists both in the addition of new features to the IWB mapping component and in integration with already existing mapping editing features. Namely, the latter allows a combination of mapping editing and analysis through automatic execution of syntactic checks on new or edited mapping rules. The former instead enriches the mapping component with the following capabilities.

1. Syntactic, local and global checks (for both inconsistency and redundancy) on any mapping available in the Optique IWB repository.
2. *Explanation* of the mapping analysis results. Noticeably, for inconsistency checks, the explanation or, potentially, the explanations in the case of global inconsistency, are provided in terms of the combination of the set of TBox axioms and the single ABox axiom, among the ones generated by algorithm `makeABox`, that together determine an inconsistency. This set of axioms is produced by using the HermiT reasoner [7] and the OWL API BlackBoxGenerator and HSTExplanationGenerator classes. Furthermore, *provenance* of the ABox axiom in each inconsistency explanation is provided. In other words, for each such axiom, the set of mapping assertions whose activation in algorithm `makeAbox` concurs either directly or indirectly to produce the axiom is returned.
3. Materialization of the mapping analysis results in the shared Optique repository hosted by the IWB platform. All mapping analysis results are translated into RDF triples and stored in the repository for future querying. In case of addition, deletion, or modification of one or more mapping assertions in a mapping, mapping analysis is automatically reset by the system, and all mapping analysis results are deleted from the repository.

The IWB provides the user with a *Semantic Wiki*, whose template pages are automatically instantiated for resources of some fixed type. The wiki features a table-centric interface, in which information is provided mostly in table form. Such tables are populated by the RDF resources in the IWB's repository that are the result of pre-defined structured SPARQL queries.

The interface of the mapping analysis component inside IWB is provided through extensions of the TriplesMapCollection and MappingCollection templates, which show, respectively, the available mappings in the repository, and information about a single mapping. A "Mapping Analysis Report" section has been added to the TriplesMapCollection template, showing, for each mapping in the repository, the ontology referenced by the mappings, the status of the mapping analysis, i.e., whether it has been performed or not, and, if so, whether or not there are local or global inconsistencies, and if the explanations have been computed. Instead, the MappingCollection template has been extended with an "Analysis Results" section, detailing the anomalies identified for each performed check: for each global inconsistency, a reference to its explanations; for each syntactically incorrect or locally inconsistent mapping assertion in the mapping, the reference to the mapping and a message detailing the anomaly; for each local subsumption, the subsumer and subsumee mapping assertions and a message detailing the type of subsumption, e.g., a head or body subsumption; finally, for global redundancies, the redundant mapping assertion and a message explaining the redundancy. Furthermore, custom templates have been produced for mapping inconsistency explanations and for explanation provenance, showing, for each, the relevant information described above, i.e., for explanations, the set of ontology axioms involved in the explanation and a reference to the provenance of the ABox axiom, and for provenance, the mappings responsible for producing the axiom.

The performance of the mapping analysis component was evaluated on one of the two large-scale use cases of the Optique project from the energy sector, namely the Statoil use case [9].

In this scenario, expert geologists develop stratigraphic models of unexplored areas on the basis of data acquired from previous operations at nearby geographical locations through advanced visual analytics tools that access more than one thousand terabytes of data. The ontology developed for the Statoil use case describes wellbores that are drilled for the extraction of natural resources such as gas or oil, and stratigraphic columns of rock layers in the geographical areas interested by these wellbores. It also describes the different kinds of measurements that can be performed in wellbores. The ontology consists of about 150 concepts and 100 roles and attributes.

The Statoil use case features two different data sources: the Exploration and Production Data Store (EPDS), and the NPD FactPages (NPD FP). EPDS is Statoil's corporate data store for exploration and production data and their own interpretations of this data, while NPD FP is a publicly available dataset that is published and maintained by the Norwegian authorities, containing reference data for many aspects of the Norwegian petroleum industry, and is often used as a data source by geologists in combination with EPDS. The mapping used in the mapping analysis evaluation relates to the EPDS data store, which currently has about 3,000 tables with about 37,000 columns, and contains about 700 gigabytes of data.

The evaluation was performed on a version of the EPDS mappings from February 2015, which is formed by 81 mapping assertions. The syntactic, consistency and redundancy tests were conducted incrementally, to account for the fact that a local inconsistency of a mapping assertion entails the global inconsistency of the mapping (hence, to find a global inconsistency that does not depend on local inconsistencies, there must be none of the latter in the mapping), and

Table 1. Results of the evaluation of the mapping analysis on the February 2015 version of the Statoil mappings for the EPDS data source.

Mapping check	Anomalies found	Time (sec)
Syntactic	5	.12
Local consistency	7	2.32
Global consistency	1	26.76
Local redundancy	2	55.33
Global redundancy	3	84.28

that redundancy checks must be performed on a consistent ontology. Therefore, evaluation was performed in the following steps.

1. Identification of syntactically incorrect and locally inconsistent mapping assertions.
2. Removal of locally inconsistent mapping assertions.
3. Identification of global inconsistencies in the mapping and production of explanations for each global inconsistency.
4. Removal of the mapping assertions, highlighted in the explanations, responsible for the global inconsistencies.
5. Identification of local and global redundancies.

The results of the evaluation are provided in Table 1. Syntactic correctness, local consistency and global redundancy were checked for each mapping assertion in the mapping, local redundancy was checked for each pair of mapping assertions in the mapping, and global consistency was checked for the whole mapping. In the case of global consistency, a value of “1” in the table indicates that the mapping was globally inconsistent, and for such an inconsistency 18 different explanations were produced. All execution times are expressed in seconds, and the complete mapping analysis procedure took roughly 3 min.

6 Conclusions

In this paper we have presented algorithm for the semantic analysis of R2RML mappings in the context of ontology-based data access. In particular, we have focused on the OWL 2 QL ontology language. We have also presented an experimental evaluation of our algorithms in the Optique project use cases.

We believe that supporting the design and maintenance of OBDA specifications, and the semantic analysis of mappings in particular, is a crucial aspect towards the successful depolyment of the OBDA technology in the real world. Within the Optique system, we are currently further expanding the mapping analysis component. First, we are developing new functionalities that make use of the ABox matrix presented in this paper. In particular, we are implementing a mapping evolution and repair functionality. In addition, we are defining a new *instance-level* mapping debugging technique, which expolits information about wrong or missing concept and role instances to identify the subset of mapping assertions that need to be repaired.

Acknowledgments. This research has been partially supported by the EU under FP7 project Optique (grant n. FP7-318338).

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison Wesley Publ. Co., Reading (1995)
2. Antonioli, N., Castanò, F., Civili, C., Coletta, S., Grossi, S., Lembo, D., Lenzerini, M., Poggi, A., Savo, D.F., Virardi, E.: Ontology-based data access: the experience at the Italian Department of treasury. In: Proceedings of the Industrial Track of the 25th International Conference on Advanced Information Systems Engineering (CAiSE), vol. 1017 of CEUR Electronic Workshop Proceedings, pp. 9–16 (2013). http://ceur-ws.org/
3. Bagosi, T., et al.: The ontop framework for ontology based data access. In: Zhao, D., Du, J., Wang, H., Wang, P., Ji, D., Pan, J.Z. (eds.) CSWS 2014. CCIS, vol. 480, pp. 67–77. Springer, Heidelberg (2014)
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. J. Autom. Reason. **39**(3), 385–429 (2007)
5. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. W3C RDB2RDF Working Group, W3C recommendation, September 2012
6. Giese, M., Soylu, A., Vega-Gorgojo, G., Waaler, A., Haase, P., Jiménez-Ruiz, E., Lanti, D., Rezk, M., Xiao, G., Özçep, Ö.L., Rosati, R.: Optique: zooming in on big data. IEEE Comput. **48**(3), 60–67 (2015)
7. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: an OWL 2 reasoner. J. Autom. Reason. **53**(3), 245–269 (2014)
8. Haase, P., Schmidt, M., Schwarte, A.: The information workbench as a self-service platform for linked data applications. In: Hartig, O., Harth, A., Sequeda, J. (eds.) Proceedings of the Second International Workshop on Consuming Linked Data (COLD2011), Bonn, Germany, October 23, 2011, vol. 782 of CEUR Workshop Proceedings. CEUR-WS.org (2011)
9. Kharlamov, E., et al.: Ontology based access to exploration data at Statoil. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9367, pp. 93–112. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25010-6_6
10. Lembo, D., Mora, J., Rosati, R., Savo, D.F., Thorstensen, E.: Towards mapping analysis in ontology-based data access. In: Kontchakov, R., Mugnier, M.-L. (eds.) RR 2014. LNCS, vol. 8741, pp. 108–123. Springer, Heidelberg (2014)
11. Lembo, D., Mora, J., Rosati, R., Savo, D.F., Thorstensen, E.: Mapping analysis in ontology-based data access: algorithms and complexity. In: Proceedings of the 14th International Semantic Web Conference (ISWC) (2015)
12. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 133–173. Springer, Heidelberg (2008)