Preliminary results on Ontology-based Open Data Publishing

Gianluca Cima

Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Sapienza Università di Roma
cima@diag.uniroma1.it

Abstract. Despite the current interest in Open Data publishing, a formal and comprehensive methodology supporting an organization in deciding which data to publish and carrying out precise procedures for publishing high-quality data, is still missing. In this paper we argue that the Ontology-based Data Management paradigm can provide a formal basis for a principled approach to publish high-quality, semantically annotated Open Data. We describe two main approaches to using an ontology for this endeavor, and then we present some technical results on one of the approaches, called bottom-up, where the specification of the data to be published is given in terms of the sources, and specific techniques allow deriving suitable annotations for interpreting the published data under the light of the ontology.

1 Introduction

In many aspects of our society there is growing awareness and consent on the need for data-driven approaches that are resilient, transparent and fully accountable. But to achieve a data-driven society, it is necessary that the data needed for public goods are readily available. Thus, it is no surprising that in recent years, both public and private organizations have been faced with the issue of publishing Open Data, in particular with the goal of providing data consumers with suitable information to capture the semantics of the data they publish. Significant efforts have been devoted to defining guidelines concerning the management and publication of Open Data. Notably, the W3C[1] has formed a working group, whose objective is the release of a first draft on Open Data Standards[2]. The focus of the document are areas such as metadata, data formats, data licenses, data quality, etc., which are treated in very general terms, with no reference to any specifical technical methodology. More generally, although there are several works on platforms and architectures for publishing Open Data, there is still no formal and comprehensive methodology supporting an organization in (*i*) deciding which data to publish, and (*ii*) carrying out precise procedures for publishing and documenting high-quality data. One of the reasons of this lack of formal methods is that the problem of Open Data Publishing is strictly related to the problem of managing the data

[1] World Wide Web Consortium: `https://www.w3.org/`
[2] Data on the Web Best Practice: `https://www.w3.org/TR/dwbp/`

within an organization. Indeed, a necessary prerequisite for an organization for publishing relevant and meaningful data is to be able to manage, maintain and document its own information system. The recent paradigm of Ontology-based Data Management (OBDM) [16] (used and experimented in practice in the last years, see, e.g., [3]) is an attempt to provide the principles and the techniques for addressing this challenge. An OBDM system is constituted by an ontology, the data sources forming the information system, and the mapping between the ontology and the sources. The ontology is a formal representation of the domain underlying the information system, and the mapping is a precise specification of the relationship between the data at the sources and the concepts in the ontology.

In this paper we argue that the OBDM paradigm can provide a formal basis for a principled approach to publish high-quality, semantically annotated Open Data. The most basic task in Open Data is the extraction of the *correct content* for the dataset(s) to be published, where by "content" we mean both the extensional information (i.e., facts about the domain of interest) conveyed by the dataset, and the intensional knowledge relevant to document such facts (e.g., concepts that intensionally describe facts), and "correct" means that the aspect of the domain captured by the dataset is coherent with a requirement formally expressed in the organization.

Current practices for publishing Open Data focus essentially on providing extensional information (often in very simple forms, such as CSV files), and they carry out the task of documenting data mostly by using metadata expressed in natural languages, or in terms of record structures. As a consequence, the semantics of datasets is not formally expressed in a machine-readable form. Conversely, OBDM opens up the possibility of a new way of publishing data, with the idea of annotating data items with the ontology elements that describe them in terms of the concepts in the domain of the organization. When an OBDM is available in an organization, an obvious way to proceed to Open Data publication is as follows: (*i*) express the dataset to be published in terms of a `SPARQL` query over the ontology, (*ii*) compute the certain answers to the query, and (*iii*) publish the result of the certain answer computation, using the query expression and the ontology as a basis for annotating the dataset with suitable metadata expressing its semantics. We call such method *top-down*. Using this method, the ontology is the heart of the task: it is used for expressing the content of the dataset to be published (in terms of a query), and it is used, together with the query, for annotating the published data.

Unfortunately, in many organizations (for example, in Public Administrations) it may be the case that people are not ready yet to manage their information systems through the OBDM paradigm. In these cases, the bottom-up approach could be more appropriate. For example, in the Italian Public Administration system, it is very unlikely that local administration people are able to express their queries over the ontology using `SPARQL`. Typically, the ontology and the mapping have been designed by third parties, with no or little involvement with IT people responsible of the local administration information system. In other words, these people probably cannot follow the top-down approach, and they are more confident to express the specification of the dataset to be published directly in terms of the source structures (i.e., the relational tables in their databases), or, more generally, in terms of a view over the sources. But how can we au-

tomatically publish both the content and the semantics of the dataset if its specification is given in terms of the data sources? We argue that we can achieve this goal by following what we call the *bottom-up* approach: the organization expresses its publishing requirement as a query over the sources, and, by using the ontology and the mapping, a suitable algorithm computes the corresponding query over the ontology. With such query at hand, we have reduced the problem in such a way that the top-down approach can now be followed, and the required data can be published according to the method described above. So, at the heart of the bottom-up approach there is a conceptual issue to address:

> *"Given a query Q over the sources, which is the query over the ontology that characterizes Q at best (independently from the current source database)?"*

Note that the answer to this question is relevant also for other tasks related to the management of the information system, e.g., the task of explaining the semantics of the various data sources within the organization. The question implicitly refers to a sort of reverse engineering problem, which is a novel aspect in the investigation of both OBDM and data integration. Indeed, most of (if not all) the literature about managing data sources through an ontology (see, e.g., [5, 18]), or more generally, about data integration [15] assume that the user query is expressed over the global schema, and the goal is to find a *rewriting* (i.e., a query over the source schema) that captures the original query in the best way, independently from the current source database. Here, the problem is reversed, because we start with a source query and we aim at deriving a corresponding query over the ontology, called a source-to-target rewriting.

In this paper we study the above described bottom-up approach, and provide the following contributions.

- We introduce the concept of *source-to-target rewriting* (see Section 3), the main technical notion underlying the bottom-up approach, and we describe two computation problems related to it, namely the recognition problem, and the finding problem. The former aims at checking whether a query over the ontology is a source-to-target rewriting of a given query over the sources, taking into account the mapping between the sources and the ontology. The latter aims at computing a suitable source-to-target rewriting of a given source query, with respect to the mapping.
- We discuss two different semantics for source-to-target rewritings, one based on the logical models of the OBDM specification, and one based on certain answers. The former is somehow the natural choice, given the first-order semantics behind OBDM. The latter is a significant alternative, that may better capture the intuition of a user who is accustomed to think of query semantics in terms of certain answers.
- We show that, although the ideal notion is the one of "exact" source-to-target rewriting, it is important to resort to approximations to exact rewriting when exactness cannot be achieved. For this reason, we introduce the notion of sound and complete source-to-target rewritings.
- For the case of complete source-to-target rewritings, we present algorithms both for the recognition (Section 4), and for the finding (Section 5) problem, in particular for the setting where the ontology is expressed in *DL-Lite*$_{\mathcal{A},id}$, and the queries involved in the specification are conjunctive queries.

2 Preliminaries

We assume familiarity with classical databases [1], Description Logics [4], and the OBDM paradigm. In this section, we (*i*) review the most basic notions of non-ground instances, and their correlation with conjunctive queries; (*ii*) briefly discuss the chase of a possible non-ground instance; (*iii*) discuss the relevant aspects of notation we use in the following regarding the OBDM paradigm.

For a possible non-ground instance $\mathtt{D}$, we assume that each value in $dom(\mathtt{D})$, i.e., the set of values occurring in $\mathtt{D}$, comes from the union of two fixed disjoint infinite sets: the set *Const* of all *constants*, and the set $Null_{\mathtt{D}}$ of all *labeled nulls*. We also let $null(\mathtt{D}) := dom(\mathtt{D}) \cap Null_{\mathtt{D}}$. In particular, each labeled null in a non-ground instance is treated as an unknown value (and hence, an *incomplete information*), rather than to a non-existent value [20]. Thus, a non-ground instance represents a number of ground instances obtained by assigning constants to each labeled null. More precisely, let $\mathtt{D}$ be a non-ground instance, and v be a mapping $v : \mathit{null}(\mathtt{D}) \rightarrow \mathit{Const}$. Then, v is called a *valuation* of $\mathtt{D}$, and we indicate, with $v(\mathtt{D})$, the ground instance obtained from $\mathtt{D}$ by replacing elsewhere each labeled null $x \in \mathtt{D}$ with $v(x)$. We also extend this to tuples, that is, given a tuple $\overline{u} = (u_1, ..., u_n)$ of both constants and labeled nulls, with $v(\overline{u})$ we indicate the tuple $(v'(u_1), ..., v'(u_n))$, where $v'(u_i) = u_i$ if u_i is a constant; otherwise (u_i is a labeled null), $v'(u_i) = v(u_i)$. Given an instance $\mathtt{D}$ it is possible to construct in linear time a boolean CQ $q_{\mathtt{D}}$ that fully captures it, and vice versa. We also let $q_{\mathtt{D}}(\overline{x})$ denoting the transformation of $q_{\mathtt{D}}$ by removing the existential quantification of the variables $\overline{x}$ in $q_{\mathtt{D}}$. Moreover, given a non-boolean CQ q (with $\overline{x}$ as distinguished variables), we associate to it the instance $\mathtt{D}_q$ by considering the variables in $\overline{x}$ as if they were existentially quantified. For ease of presentation, we extend CQs to allow also queries of the form $\{\overline{x} \mid \bot(\overline{x})\}$ and $\{\overline{x} \mid \top(\overline{x})\}$, with their usual meaning. We also denote with $tup(q)$ the tuple composed by the terms in *head* of q.

Given a source schema $\mathcal{S}$; a target schema T; a set $\mathcal{M}$ of *st-tgds* (i.e., assertions of the form $\forall \overline{x}, \overline{y}(\phi(\overline{x}, \overline{y}) \rightarrow \exists \overline{z}\varphi(\overline{x}, \overline{z}))$, where ϕ is a CQ over $\mathcal{S}$, and φ is a CQ over $\mathtt{T}$); and a set Σ_t of *egds* (i.e., assertions of the form $\forall \overline{x}(\phi_{\mathtt{T}}(\overline{x}) \rightarrow (x_1 = x_2))$, where $\phi_{\mathtt{T}}$ is a CQ over $\mathtt{T}$, and x_1, x_2 are among the variables in $\overline{x}$), the *chase* procedure of a possibly non-ground source instance $\mathtt{D}$ consists in: (*i*) the chase of $\mathtt{D}$ w.r.t. $\mathcal{M}$, where, for every st-tgd $\phi(\overline{x}, \overline{y}) \rightarrow \exists \overline{z}\varphi(\overline{x}, \overline{z})$ in $\mathcal{M}$ and for every pair of tuples $(\overline{a}, \overline{b})$ such that $\mathtt{D} \models \phi(\overline{a}, \overline{b})$, there is the introduction of new facts in the instance J of the target schema T so that $\varphi(\overline{a}, \overline{u})$ holds, where $\overline{u}$ consists in a fresh tuple of distinct labeled nulls coming from an infinite set $Null_{\Sigma}$ disjoint from $Null_{\mathtt{D}}$; (*ii*) the chase of J w.r.t. Σ_t, where, for every egd $\forall \overline{x}(\phi_{\mathtt{T}}(\overline{x}) \rightarrow (x_1 = x_2))$ and for every tuple $\overline{a}$ such that $J \models \phi_{\mathtt{T}}(\overline{a})$ and $a_1 \neq a_2$, we equate the two terms. Equating a_1 with a_2 means choosing one of the two so that the other is replaced elsewhere in J by the one chosen. In particular, if one is a labeled null and the other is a constant, then the chase choose the constant; if both are labeled nulls, one coming from $Null_{\mathtt{D}}$ and the other from $Null_{\Sigma}$, it always choose the one coming from $Null_{\mathtt{D}}$; if both are constants, then the chase fail. Moreover, with ψ we denote the set of equalities applied by the chase of J w.r.t. a set of egds on variables coming from $Null_{\mathtt{D}}$. This can be done by keeping track of the substitution applied by the chase. For example, if the chase equates the variable $y \in Null_{\mathtt{D}}$ with the variable $x \in Null_{\mathtt{D}}$, and then equates the variable $z \in Null_{\mathtt{D}}$ with the variable $w \in Null_{\mathtt{D}}$, and

then w with the constant a, given the tuple (x, y, z, w), $\psi(x, y, z, w)$ indicates the tuple (x, x, a, a). Note that, we can compute the certain answers of a boolean union of CQs (UCQ) q with at most one inequality per disjunct by splitting q as a boolean UCQ $q^{1\neq}$ with exactly one inequality per disjunct, and a boolean UCQ $q^{0\neq}$ with no inequality per disjunct. The key idea is that the negation of $q^{1\neq}$ consists in a set of egds, hence, the certain answers of q can be computed by applying the chase procedure over the instance J (i.e., the instance produced by the chase of $\mathcal{C}$ w.r.t. $\mathcal{M}$ and Σ_t) w.r.t. $\neg q^{1\neq}$, where, if the chase fail then the answer is $true$; otherwise, if the instance J' produced satisfy one of the conjunctive query in $q^{0\neq}$, then the answer is $true$, else the answer is $false$. We refer to [10] for more details.
Given an OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$, where $\mathcal{O}$ is a TBox, and $\mathcal{M}$ is a set of st-tgds, and given a non-ground source instance $\mathtt{D}$ for $\mathcal{S}$, and a set of egds Σ_t, we denote with $\mathcal{A}_{\mathtt{D},\Sigma}$, where $\Sigma = \mathcal{M} \cup \Sigma_t$, the ABox computed as follows: (*i*) chase the non-ground source instance $\mathtt{D}$ w.r.t. Σ; (*ii*) *freeze* the instance (or equivalently, the ABox with variables) obtained, i.e., variables in this instance are now considered as constant. Note that, such ABox $\mathcal{A}_{\mathtt{D},\Sigma}$ may also not exists due to the failing of the chase, in this case, we denote $\mathcal{A}_{\mathtt{D},\Sigma}$ with the symbol $\perp$.

For an OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$, and for a source database $\mathcal{C}$ for $\mathcal{I}$ (i.e., a ground instance over the schema $\mathcal{S}$), we denote by $sem^{\mathcal{C}}(\mathcal{I})$ the set of *models* $\mathcal{B}$ for $\mathcal{I}$ relative to $\mathcal{C}$ such that: (*i*) $\mathcal{B} \models \mathcal{O}$; (*ii*) $(\mathcal{B}, \mathcal{C}) \models \mathcal{M}$. Given a query q over $\mathcal{O}$, we denote by $cert(q, \mathcal{I}, \mathcal{C})$ the set of *certain answers* to q in $\mathcal{I}$ relative to $\mathcal{C}$. It is defined as: $cert(q, \mathcal{I}, \mathcal{C}) = \bigcap\{q^{\mathcal{B}} \mid \mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})\}$ if $sem^{\mathcal{C}}(\mathcal{I}) \neq \emptyset$; otherwise, $cert(q, \mathcal{I}, \mathcal{C}) = AllTup(q, \mathcal{C})$, where $AllTup(q, \mathcal{C})$ is the set of all possible tuples of constants in $\mathcal{C}$ whose arity is the one of the query q. Furthermore, given a *DL-Lite*$_{\mathcal{A},id}$ [5] TBox $\mathcal{O}$ and a *DL-Lite*$_{\mathcal{A},id}$ ABox $\mathcal{A}$ we are able to: (*i*) check whether $\langle \mathcal{O}, \mathcal{A} \rangle$ is satisfiable by computing the answers of a suitable boolean query Q_{sat} (a UCQ with at most one inequality per disjunct) over the ABox $\mathcal{A}$ considered as a relational database. We see Q_{sat} as the union of $Q_{sat}^{0\neq}$ (the UCQ containing every disjunct not comprising inequalities in Q_{sat}) and $Q_{sat}^{1\neq}$ (the UCQ containing every disjunct comprising inequalities in Q_{sat}); (*ii*) compute the *certain answers* to a UCQ Q_g over a satisfiable $\langle \mathcal{O}, \mathcal{A} \rangle$, denoted with $cert(Q_g, \langle \mathcal{O}, \mathcal{A} \rangle)$, by producing a *perfect reformulation* (denoted as a function $pr(\cdot)$) of such query, and then computing the answers of $pr(Q_g)$ over the ABox $\mathcal{A}$ considered as a relational database. See [6] for more details.

3 The notion of source-to-target rewriting

In what follows, we implicitly refer to (*i*) an OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$; (*ii*) a query Q_s over the source schema $\mathcal{S}$; (*iii*) a query Q_g over the ontology $\mathcal{O}$.

As we said in the introduction, there are at least two different ways to formally define a source-to-target rewriting (*s-to-t rewriting* in the following) for each of the three variants, namely "exact", "complete", and "sound". The first one is captured by the following definition.

Definition 1. Q_g is a *complete* (resp., *sound*, *exact*) *s-to-t rewriting* of Q_s with respect to $\mathcal{I}$ under the *model-based semantics*, if for each source database $\mathcal{C}$ and for each model $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$, we have that $Q_s^{\mathcal{C}} \subseteq Q_g^{\mathcal{B}}$ (resp., $Q_g^{\mathcal{B}} \subseteq Q_s^{\mathcal{C}}$, $Q_g^{\mathcal{B}} = Q_s^{\mathcal{C}}$).

Intuitively, a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under the model-based semantics is a query over $\mathcal{O}$ that, when evaluated over a model $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$ for a source database $\mathcal{C}$, returns all the answers of the evaluation of Q_s over $\mathcal{C}$. In other words, for every source database $\mathcal{C}$, the query Q_g over $\mathcal{O}$ captures all the semantics that Q_s expresses over $\mathcal{C}$. Similar arguments hold for the notions of sound and exact s-to-t rewriting under this semantics. Moreover, from the formal definition of source-to-target rewriting and the usual definition of target-to-source rewriting (simply called rewriting) used in data integration, it is easy to see that Q_g is a complete (resp., sound) source-to-target rewriting of Q_s w.r.t. $\mathcal{I}$ under the model-based semantics, if and only if Q_s is a sound (resp., complete) rewriting of Q_s w.r.t. $\mathcal{I}$, implying that, Q_g is an exact source-to-target rewriting of Q_s w.r.t. $\mathcal{I}$ under the model-based semantics, if and only if Q_s is an exact rewriting of Q_g w.r.t. $\mathcal{I}$.

The second possible way to formally define a source-to-target rewriting is as follows.

Definition 2. Q_g is a *complete* (resp., *sound*, *exact*) *s-to-t rewriting* of Q_s with respect to $\mathcal{I}$ under the *certain answers-based semantics*, if for each source database $\mathcal{C}$ such that $sem^{\mathcal{C}}(\mathcal{I}) \neq \emptyset$, we have that $Q_s^{\mathcal{C}} \subseteq cert(Q_g, \mathcal{I}, \mathcal{C})$ (resp., $cert(Q_g, \mathcal{I}, \mathcal{C}) \subseteq Q_s^{\mathcal{C}}$, $Q_s^{\mathcal{C}} = cert(Q_g, \mathcal{I}, \mathcal{C})$).

In this new semantics, in order to capture a query Q_s over $\mathcal{S}$, we resort to the notion of certain answers. Indeed, a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under the certain answers-based semantics is a query over $\mathcal{O}$ such that, when we compute its certain answers for a source database $\mathcal{C}$, we get all the answers of the evaluation of Q_s over $\mathcal{C}$. As before, similar arguments hold for the notions of sound and exact s-to-t rewriting under this semantics. Note also the strong correspondence between the exact s-to-t rewriting under the certain answers-based semantics and the notion of *perfect rewriting*. We remind that a perfect rewriting of Q_g w.r.t. $\mathcal{I}$ is a query Q_s over $\mathcal{S}$ that computes $cert(Q_g, \mathcal{I}, \mathcal{C})$ for every source database $\mathcal{C}$ such that $sem^{\mathcal{C}}(\mathcal{I}) \neq \emptyset$ [8]. Indeed, we have that Q_g is an exact s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under the certain answers-based semantics if and only if Q_s is a perfect rewriting of Q_g w.r.t. $\mathcal{I}$. Note that the above observations imply that the two semantics are indeed different, since it is well-known that the two notions of exact rewriting and perfect rewriting of Q_g w.r.t. $\mathcal{I}$ are different. The difference between the two semantics is confirmed by the following example.

Example 1. $\mathcal{O} := \emptyset$ (i.e., no TBox assertions in $\mathcal{O}$); $\mathcal{S}$ contains a binary relation r_1 and a unary relation r_2; $\mathcal{M} := \{\forall x \forall y(r_1(x, y) \rightarrow \mathsf{G}(x, y)), \forall x(r_2(x) \rightarrow \exists Y.\mathsf{G}(x, Y))\}$; $Q_s := \{(w) \mid \exists Z.r_1(Z, w)\}$; $Q_g := \{(w) \mid \exists Z.\mathsf{G}(Z, w)\}$.

It is easy to see that Q_g is a sound s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under the certain answers-based semantics (more precisely, it is an exact s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under such semantics), while it is not sound under the model-based semantics. In fact, for the source database $\mathcal{C}$ with $r_1^{\mathcal{C}} = \{(a, b)\}$ and $r_2^{\mathcal{C}} = \{(c)\}$, and for the model $\mathcal{B}$ with $\mathsf{G}^{\mathcal{B}} = \{(a, b), (c, d)\}$, we have $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$, and $Q_g^{\mathcal{B}} \not\subseteq Q_s^{\mathcal{C}}$. □

Intuitively, for the sound case, the model-based semantics is too strong, in the sense that under such semantics, a model $\mathcal{B}$ may contain not only facts depending on how data in the source $\mathcal{C}$ are linked to $\mathcal{O}$ through $\mathcal{M}$, but additionally arbitrary facts, with the only constraint of satisfying $\mathcal{O}$. One might think that, in order to address this issue, it is

sufficient to resort to a sort of minimizations of the models of $\mathcal{O}$. Actually, the above example shows that, even if we restrict the set of models to the set of minimal models (i.e., models $\mathcal{B}$ such that (*i*) $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$ and (*ii*) there is no model $\mathcal{B}' \in sem^{\mathcal{C}}(\mathcal{I})$ such that $\mathcal{B}' \subset \mathcal{B}$), and adopt a semantics like the model-based one but restricted to the set of minimal models, Q_g is still not a sound s-to-t rewriting (this can be seen considering that the target database $\mathcal{B}$ defined earlier is a minimal model).

Observe that the above considerations show the difference in the two semantics by referring to sound and exact s-to-t rewritings. It is interesting to ask whether the difference shows up when restricting our attention to complete rewritings. The following proposition deals with this question.

Proposition 1. *Q_g is a complete s-to-t rewriting of Q_s with respect to $\mathcal{I}$ under the model-based semantics if and only if it is so under the certain answers-based semantics.*

Proof (Sketch). One direction is trivial. Indeed, when Q_g is a complete s-to-t rewriting of Q_s with respect to $\mathcal{I}$ under the model-based semantics, by definition of certain answers, for each source database $\mathcal{C}$ such that $sem^{\mathcal{C}}(\mathcal{I}) \neq \emptyset$ we have that $Q_s^{\mathcal{C}} \subseteq cert(Q_g, \mathcal{I}, \mathcal{C})$. For the other direction, suppose that Q_g is not a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under the model-based semantics. It follows that, there exists a source database $\mathcal{C}$ and a model $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$ such that $Q_s^{\mathcal{C}} \not\subseteq Q_g^{\mathcal{B}}$, implying that, $Q_s^{\mathcal{C}} \not\subseteq cert(Q_g, \mathcal{I}, \mathcal{C})$, which, in turn, implies that Q_g is not a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under the certain answers-based semantics. □

Obviously, the query over the ontology which captures at best a given query q over the source schema is the exact s-to-t rewriting of q. However, the following example shows that even for very simple OBDM specifications, an exact s-to-t rewriting of even trivial queries, may not exist.

Example 2. $\mathcal{O} := \emptyset$ (i.e., no TBox assertions in $\mathcal{O}$); $\mathcal{S}$ contains two unary relations man and $woman$; $\mathcal{M} := \{\forall x(man(x) \rightarrow \mathsf{Person}(x)), \forall x(woman(x) \rightarrow \mathsf{Person}(x))\}$; $Q_s := \{(x) \mid woman(x)\}$.

It is possible to show that the only sound s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ under both semantics is the query $Q_g := \{(x) \mid \bot(x)\}$, which is obviously not a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ neither under the model-based semantics, nor under the certain answers-based semantics. On the other hand, the most immediate and intuitive complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ is the query $Q'_g := \{(x) \mid \mathsf{Person}(x)\}$. Furthermore, as we will see in Section 5, this query is an "optimal" complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$, where the term *optimal* will be precisely defined. □

As we said in the introduction, in the rest of this paper we focus on complete s-to-t-rewritings. In particular, we will address both the recognition problem (see Section 4), and the finding problem (see Section 5) in a specific setting, characterized as follows:

- The ontology $\mathcal{O}$ in an OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$ is expressed as a TBox in *DL-Lite*$_{\mathcal{A},id}$.
- The mapping $\mathcal{M}$ in $\mathcal{I}$ is a set of GLAV mapping assertions (or, *st-tgds*), where each assertion expresses a correspondence between a conjunctive query over the source schema and a conjunctive query over the ontology.

– In the recognition problem, both the query over the source schema and the query over the ontology are conjunctive queries. Similarly, in the finding problem, the query over the source schema is a conjunctive query.

4 The recognition problem for complete s-to-t rewritings

We implicitly refer to the setting described at the end of the previous section. The *recognition problem* associated to the complete s-to-t rewriting is the following decision problem: Given an OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$, a query Q_s over the source schema $\mathcal{S}$, and a query Q_g over the ontology $\mathcal{O}$, check whether Q_g is a complete s-to-t rewriting of Q_s with respect to $\mathcal{I}$. The next lemma is the starting point of our solution.

Lemma 1. *Q_g is not a complete s-to-t rewriting of Q_s with respect to $\mathcal{I} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$ if and only if there is a valuation v of D_{Q_s} and a model $\mathcal{B} \in sem^{v(\mathtt{D}_{Q_s})}(\mathcal{I})$ such that $v(tup(Q_s)) \notin Q_g^{\mathcal{B}}$.*

Proof. "$\Longleftarrow$" Suppose that there exists a valuation v of $\mathtt{D}_{Q_s}$ and a model $\mathcal{B} \in sem^{v(\mathtt{D}_{Q_s})}(\mathcal{I})$ such that $v(tup(Q_s)) \notin Q_g^{\mathcal{B}}$. Obviously, $v(tup(Q_s)) \in Q_s^{v(\mathtt{D}_{Q_s})}$. It follows that, there exist a source database $v(\mathtt{D}_{Q_s})$, a model $\mathcal{B} \in sem^{v(\mathtt{D}_{Q_s})}(\mathcal{I})$, and a tuple $v(tup(Q_s))$ such that $v(tup(Q_s)) \notin Q_g^{\mathcal{B}}$ and $v(tup(Q_s)) \in Q_s^{v(\mathtt{D}_{Q_s})}$.

"$\Longrightarrow$" Suppose that Q_g is not a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$, i.e., there is a source database $\mathcal{C}$, a model $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$, and a tuple t such that $t \in Q_s^{\mathcal{C}}$ and $t \notin Q_g^{\mathcal{B}}$. The fact that $t \in Q_s^{\mathcal{C}}$ implies the existence of a homomorphism $v : \mathtt{D}_{Q_s} \rightarrow \mathcal{C}$ such that $v(tup(Q_s)) = t$. Note also, that since $\mathcal{C}$ is a *ground* instance, v is a valuation of $\mathtt{D}_{Q_s}$ such that $v(\mathtt{D}_{Q_s}) \subseteq \mathcal{C}$. Obviously, $\mathcal{B} \in sem^{v(D_{Q_s})}(\mathcal{I})$, this can be seen by considering that (*i*) $\mathcal{B} \models \mathcal{O}$ is true from the supposition that $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$; and (*ii*) $(\mathcal{B}, v(\mathtt{D}_{Q_s})) \models \mathcal{M}$ is true by considering that, $(\mathcal{B}, \mathcal{C}) \models \mathcal{M}$ (which holds from the supposition that $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$), $v(D_{Q_s}) \subseteq \mathcal{C}$, and the queries in $\mathcal{M}$ are *monotone* queries. It follows that, there is a valuation v of $\mathtt{D}_{Q_s}$ and a model $\mathcal{B} \in sem^{v(\mathtt{D}_{Q_s})}(\mathcal{I})$ such that $v(tup(Q_s)) \notin Q_g^{\mathcal{B}}$. □

Relying on the above lemma, we are now ready to present the algorithm CheckComplete for the recognition problem.

Algorithm 1: CheckComplete($\mathcal{I}$, Q_s, Q_g)

Input: OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, query Q_s over $\mathcal{S}$, query Q_g over $\mathcal{O}$.
Output: *true* or *false*.

1: Compute $\mathtt{D}_{Q_s}$ from Q_s (i.e., the instance, possibly with incomplete information, associated to the query Q_s), and denote it with $\mathtt{D}$.
2: Compute $\mathcal{A}_{\mathtt{D},\Sigma}$, where $\Sigma = \mathcal{M} \cup \neg Q_{sat}^{1\neq}$, and let ψ be the set of equality applied to the variables in $\mathtt{D}$ by the chase.
3: If $\mathcal{A}_{\mathtt{D},\Sigma} = \bot$, then return *true*.
4: If the evaluation of $Q_{sat}^{0\neq}$ over $\mathcal{A}_{\mathtt{D},\Sigma}$ considered as a relational database is $\{()\}$ (i.e., $\mathcal{A}_{\mathtt{D},\Sigma} \models Q_{sat}^{0\neq}$), then return *true*.
5: If $\psi(tup(Q_s)) \in cert(Q_g, \langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma} \rangle)$ then return *true*, else return *false*.

The next theorem establishes the correctness of the above algorithm.

Theorem 1. *CheckComplete($\mathcal{I}$, Q_s, Q_g) terminates, and returns* true *if and only if Q_g is a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$.*

Proof (Sketch). Termination of the algorithm easily follows by the termination of the chase procedure, and by the obvious termination of computing the certain answers of a CQ over $\langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma} \rangle$.

For the "$\Longrightarrow$" direction, suppose that the algorithm returns false, i.e., $\mathcal{A}_{\mathtt{D},\Sigma} \not\models Q^{0\neq}_{sat}$, and $\psi(tup(Q_s)) \notin cert(Q_g, \langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma} \rangle)$. Now, if we extend $\psi(\mathtt{D})$ by considering the freezing of this instance (i.e., variables are now considered as constants), it is easy to see that we obtain a valuation v of $\mathtt{D}$ such that $(v(\mathtt{D}), \mathcal{A}_{\mathtt{D},\Sigma}) \models \mathcal{M}$, and such that $sem^{v(\mathtt{D})}(\mathcal{I}) \neq \emptyset$. Moreover, the fact that $\psi(tup(Q_s)) \notin cert(Q_g, \langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma} \rangle)$, implies, by the property of certain answers, that there is at least one model $\mathcal{B} \models \langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma} \rangle$, and hence $\mathcal{B} \in sem^{v(\mathtt{D})}(\mathcal{I})$ (because $(v(\mathtt{D}), \mathcal{A}_{\mathtt{D},\Sigma}) \models \mathcal{M}$) such that $v(tup(Q_s)) \notin Q^{\mathcal{B}}_g$. It follows, from Lemma 1, that Q_g is not a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$.

For the "$\Longleftarrow$" direction, in the cases that $\mathcal{A}_{\mathtt{D},\Sigma} = \bot$ or $\mathcal{A}_{\mathtt{D},\Sigma} \models Q^{0\neq}_{sat}$, it is easy to see that for every valuation v of $\mathtt{D}$, either the chase of $v(\mathtt{D})$ will fail, or every ABox $\mathcal{A}$ such that $(v(\mathtt{D}), \mathcal{A}) \models \mathcal{M}$ and $\mathcal{A} \models \neg Q^{1\neq}_{sat}$ will be such that $\mathcal{A} \models Q^{0\neq}_{sat}$, implying that, for every valuation v of $\mathtt{D}$, $sem^{v(\mathtt{D})} = \emptyset$. It follows, from Lemma 1, that in this case Q_g is a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$. While, in the cases that $\psi(tup(Q_s)) \in cert(Q_g, \langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma} \rangle)$, it is easy to see that, for every valuation v of $\mathtt{D}$ either $sem^{v(\mathtt{D})} = \emptyset$, or if we compute $\mathcal{A}_{v(\mathtt{D}),\Sigma}$, we have that $v(tup(Q_s)) \in cert(Q_g, \langle \mathcal{O}, \mathcal{A}_{v(\mathtt{D}),\Sigma} \rangle)$. More generally, every $\mathcal{A}$ obtained by chasing $v(\mathtt{D})$ w.r.t. $\mathcal{M}$ and $\neg Q^{1\neq}_{sat}$, and then choosing arbitrary constants for the possible remaining variables, is such that $v(tup(Q_s)) \in cert(Q_g, \langle \mathcal{O}, \mathcal{A} \rangle)$. Hence, for every model $\mathcal{B}$ such that $\mathcal{B} \models \langle \mathcal{O}, \mathcal{A} \rangle$, we have that $v(tup(Q_s)) \in Q^{\mathcal{B}}_g$. Also, we observe that the set of models $sem^{v(\mathtt{D})}$ coincides with the set of all models $\mathcal{B}$ such that $\mathcal{B} \models \langle \mathcal{O}, \mathcal{A} \rangle$ for all the possible ABox $\mathcal{A}$ obtained using the above procedure. It follows that, for every possible valuation v of $\mathtt{D}$ and for every possible $\mathcal{B} \in sem^{v(\mathtt{D})}(\mathcal{I})$, we have that $v(tup(Q_s)) \in Q^{\mathcal{B}}_g$, implying, from Lemma 1, that also in this case Q_g is a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$. □

As for complexity issues of the algorithm, we observe: (*i*) it runs in PTIME in the size of Q_s. Indeed, computing $\mathtt{D}$ (the instance associated to the query Q_s) can be done in linear time, and chasing an instance in the presence of a weakly acyclic set of tgds (as in our case) is PTIME in the size of $\mathtt{D}$ ($\mathcal{M}$ and Σ are considered fixed); (*ii*) it runs in PTIME in the size of $\mathcal{O}$. Indeed, Q_{sat} and the evaluation of the certain answers of Q_g can be both computed in PTIME in the size of $\mathcal{O}$; (*iii*) it runs in EXPTIME in the size of $\mathcal{M}$. This can be seen from the obvious EXPTIME process of transferring data from $\mathtt{D}$ to $\mathcal{A}_{\mathtt{D},\Sigma}$; (*iv*) the problem is NP-complete in the size of Q_g because computing the certain answers of a UCQ query is NP-complete in the size of the query (*query complexity*).

5 Finding optimal complete s-to-t rewritings

In this section we study the problem of finding optimal complete s-to-t rewritings. The first question to ask is which rewriting we chose in the case where several complete

rewritings exist. The obvious choice is to define the notion of "optimal" complete s-to-t rewriting: one such rewriting r is optimal if there is no complete s-to-t rewriting that is contained in r. In order to formalize this notion, we introduce the following definitions (where $MOD(\mathcal{O})$ denotes the set of models of $\mathcal{O}$).

Definition 3. Q_g is *contained* in Q'_g with respect to $\mathcal{O}$, denoted $Q_g \subseteq_{\mathcal{O}} Q'_g$, if for every model $\mathcal{B} \in MOD(\mathcal{O})$ we have that ${Q_g}^{\mathcal{B}} \subseteq {Q'_g}^{\mathcal{B}}$. Q_g is *proper contained* in Q'_g with respect to $\mathcal{O}$, denoted $Q_g \subset_{\mathcal{O}} Q'_g$, if $Q_g \subseteq_{\mathcal{O}} Q'_g$ and for at least one model $\mathcal{B} \in MOD(\mathcal{O})$ we have that ${Q_g}^{\mathcal{B}} \subset {Q'_g}^{\mathcal{B}}$.

Definition 4. Q_g is an *optimal complete s-to-t rewriting* of Q_s with respect to $\mathcal{I}$, if Q_g is a complete s-to-t rewriting of Q_s with respect to $\mathcal{I}$, and there exists no query Q'_g such that Q'_g is a complete s-to-t rewriting of Q_s with respect to $\mathcal{I}$ and $Q'_g \subset_{\mathcal{O}} Q_g$.

We are ready to present an algorithm for computing an optimal complete s-to-t rewriting of a query over the source schema. For the termination and the complexity of this algo-

Algorithm 2: FindOptimalComplete($\mathcal{I}$, Q_s)

Input: OBDM specification $\mathcal{I} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, CQ Q_s over $\mathcal{S}$.
Output: query Q_g over $\mathcal{O}$.

1: Compute $\mathtt{D}_{Q_s}$ from Q_s (i.e., the instance, possibly with incomplete information, associated to the query Q_s), and denote it with $\mathtt{D}$.
2: Chase $\mathtt{D}$ w.r.t. $\mathcal{M}$ to produce an instance J'.
3: Chase J' w.r.t. $\neg Q_{sat}^{1\neq}$; if the chase fails, then *stop* and return the query $\{tup(Q_s) \mid \bot(tup(Q_s))\}$; otherwise, let J be the instance produced, and let ψ be the set of equality applied to the variables in $\mathtt{D}$ by the chase.
4: Evaluate $Q_{sat}^{0\neq}$ over J; if the answer is $\{()\}$ (i.e., $J \models Q_{sat}^{0\neq}$), then *stop* and return the query $\{tup(Q_s) \mid \bot(tup(Q_s))\}$.
5: If $J = \emptyset$ (i.e., no atoms in the instance J), then *stop* and return the query $\{tup(Q_s) \mid \top(tup(Q_s))\}$; otherwise, let Q_J be the boolean conjunctive query associated to the instance J.
6: Let $\overline{w}$ be the tuple composed by all terms in $\psi(tup(Q_s))$ not appearing in J. If such tuple is not empty, then return $\{\psi(tup(Q_s)) \mid Q_J(\psi(tup(Q_s)) \wedge \top(\overline{w})\}$; otherwise, return $\{\psi(tup(Q_s)) \mid Q_J(\psi(tup(Q_s)))\}$.

rithm hold the same considerations done for the termination and the complexity of the CheckComplete algorithm. In particular, FindOptimalComplete($\mathcal{I}$,Q_s) terminates, and it runs in (*i*) PTIME in the size of Q_s; (*ii*) PTIME in the size of $\mathcal{O}$; (*iii*) EXPTIME in the size of $\mathcal{M}$. Whereas, the correctness is established by the next theorem.

Theorem 2. *FindOptimalComplete($\mathcal{I}$, Q_s) returns an optimal complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$.*

Proof (Sketch). When the algorithm returns the query $\{tup(Q_s) \mid \bot(tup(Q_s))\}$, it is easy to see that, regardless of which is the query Q_g, if we run the algorithm CheckComplete($\mathcal{I}$,Q_s,Q_g) it returns *true* (also in this case, either the chase will fail, or the

ABox $\mathcal{A}_{\mathtt{D},\Sigma}$ produced will satisfy $Q_{sat}^{0\neq}$), and hence, by Theorem 1, Q_g is a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$. It follows that, also $\{tup(Q_s) \mid \bot(tup(Q_s))\}$ is a complete s-to-t rewriting, and, by definition of such query, it is an optimal complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$.

When the algorithm returns the query $Q_g = \{\psi(tup(Q_s)) \mid Q_J(\psi(tup(Q_s)) \wedge \top(\overline{w})\}$ (or $\{tup(Q_s) \mid \top(tup(Q_s))\}$, in the case $J = \emptyset$), if we run the algorithm CheckComplete($\mathcal{I}$,Q_s,Q_g), it computes the ABox $\mathcal{A}_{\mathtt{D},\Sigma}$, where $tup(Q_s) \in cert(Q_g, \langle \mathcal{O}, \mathcal{A}_{\mathtt{D},\Sigma}\rangle)$ holds because Q_g corresponds exactly to $\mathcal{A}_{\mathtt{D},\Sigma}$ (before to be freezed) extended with $\top(\overline{w})$ for all terms $\overline{w}$ in $\psi(tup(Q_s))$ not appearing in $\mathcal{A}_{\mathtt{D},\Sigma}$. It follows that, also in this case, CheckComplete($\mathcal{I}$,Q_s,Q_g) returns *true*, implying, from Theorem 1, that Q_g is a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$.

We now prove that the query $Q_g = \{\psi(tup(Q_s)) \mid Q_J(\psi(tup(Q_s)) \wedge \top(\overline{w})\}$ (or $\{tup(Q_s) \mid \top(tup(Q_s))\}$, in the case $J = \emptyset$) is also an optimal complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$. In particular, suppose that there exist a query Q'_g such that $Q'_g \subset_{\mathcal{O}} Q_g$, i.e., $Q'_g \subseteq_{\mathcal{O}} Q_g$, and there is a model $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$ and a tuple t such that $t \in Q_g^{\mathcal{B}}$ and $t \notin {Q'_g}^{\mathcal{B}}$. The fact that $t \in Q_g^{\mathcal{B}}$ implies the existence of a valuation v to all the variables in Q_g that makes Q_g true in $\mathcal{B}$. Note that, we can extend the valuation v by assigning a new fresh constant to every variable appearing in $\mathtt{D}$ and not appearing in Q_g. The valuation v obtained is now a valuation for $\mathtt{D}$, and obviously $t \in Q_s^{v(\mathtt{D})}$. Moreover, if we apply the same valuation v to the instance J, it is easy to see that we obtain a ground instance J' such that $(v(\mathtt{D}), J') \models \mathcal{M}$ (we recall that Q_g is the CQ associated to the instance J). Obviously, $J' \subseteq \mathcal{B}$, and hence, $(v(\mathtt{D}), \mathcal{B}) \models \mathcal{M}$ holds because queries in the mapping $\mathcal{M}$ are *monotone* queries. Moreover, we also have that $\mathcal{B} \in sem^{v(\mathtt{D})}(\mathcal{I})$ (the fact that $\mathcal{B} \models \mathcal{O}$ holds from the initial supposition). Hence, for the source database $v(\mathtt{D})$ there is a model $\mathcal{B} \in sem^{v(\mathtt{D})}(\mathcal{I})$ and a tuple t such that $t \in Q_s^{v(\mathtt{D})}$ and $t \notin {Q'_g}^{\mathcal{B}}$, implying that, Q'_g is not a complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$. □

It is easy to prove that the query returned by the algorithm is not only an optimal complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$, but it is also the unique (up to equivalence) optimal complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$. Furthermore, the above result implies that an optimal complete s-to-t rewriting of Q_s w.r.t. $\mathcal{I}$ can always be expressed as a CQ.

6 Conclusion

We have introduced the notion of Ontology-based Open Data Publishing, whose idea is to use an OBDM specification as a basis for carrying out the task of publishing high-quality open data.

In this paper, we have focused on the bottom-up approach to ontology-based open data publishing, we have introduced the notion of source-to-target rewriting, and we have developed algorithms for two problems related to complete source-to-target rewritings, namely the recognition and the finding problem. We plan to continue our work on several directions. In particular, we plan to investigate the notion of sound rewriting under different semantics. Also, we want to study the top-down approach, especially with the goal of devising techniques for deriving which intensional knowledge to associate to datasets in order to document their content in a suitable way.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. 1995.
2. F. N. Afrati and P. G. Kolaitis. Answering aggregate queries in data exchange. pages 129–138, 2008.
3. N. Antonioli, F. Castanò, S. Coletta, S. Grossi, D. Lembo, M. Lenzerini, A. Poggi, E. Virardi, and P. Castracane. Ontology-based data management for the italian public debt. pages 372–385, 2014.
4. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. 2003.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, and R. Rosati. Ontologies and databases: The *DL-Lite* approach. volume 5689, pages 255–356. 2009.
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. 39(3):385–429, 2007.
7. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Path-based identification constraints in description logics. pages 231–241, 2008.
8. D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query processing: On the relationship between rewriting, answering and losslessness. volume 3363, pages 321–336, 2005.
9. A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. pages 77–90, 1977.
10. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. pages 207–224, 2003.
11. R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, mar 2005.
12. A. Hernich. Answering non-monotonic queries in relational data exchange. pages 143–154, 2010.
13. A. Hernich, L. Libkin, and N. Schweikardt. Closed world data exchange. *ACM Trans. Database Syst.*, 36(2):14:1–14:40, 2011.
14. T. Imielinski and W. Lipski, Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.
15. M. Lenzerini. Data integration: A theoretical perspective. pages 233–246, 2002.
16. M. Lenzerini. Ontology-based data management. pages 5–6, 2011.
17. L. Libkin and C. Sirangelo. Data exchange and schema mappings in open and closed worlds. pages 139–148, 2008.
18. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. X:133–173, 2008.
19. M. Y. Vardi. The complexity of relational query languages. pages 137–146, 1982.
20. C. Zaniolo. Database relations with null values. In *Proc. of PODS*, pages 27–33, 1982.