# Rewriting-Based Query Answering for Semantic Data Integration Systems

Maxime Buron, François Goasdoué, Ioana Manolescu, Marie-Laure Mugnier

# Rewriting-Based Query Answering for Semantic Data Integration Systems

Maxime Buron
Inria, France
maxime.buron@inria.fr

François Goasdoué
U. Rennes & Inria, France
fg@irisa.fr

Ioana Manolescu
Inria, France
ioana.manolescu@inria.fr

Marie-Laure Mugnier
U. Montpellier & Inria, France
marie-laure.mugnier@lirmm.fr

## ABSTRACT

We consider the integration of *heterogeneous data* under a *global RDF graph data model*, based on the knowledge expressed in an *ontology* describing the concepts relevant for an application, and a set of *entailment rules* which characterize the logical relationships between these concepts. We consider both standard RDF Schema rules and user-specified ones.

We propose *a data integration architecture* to compute certain query answers in this setting. Existing approaches to query answering in the presence of knowledge (expressed here in the ontology and the entailment rules) involve either the materialization of inferences in the data or the reformulation of the query. Both approaches have well-known drawbacks. We introduce a *new approach to query answering*, based on a reduction to view-based query answering. This approach avoids both materialization in the data and query reformulation. We define restrictions of our general architecture under which our method is correct, and formally prove its correctness.

## 1. INTRODUCTION

The development of data management systems has quickly lead to the need to *integrate* several databases under a single schema, or unified view. Such an integrated architecture simplifies application development and isolates it from possible changes in the underlying databases and data management systems. Two classical architectures have been used for data integration: $(i)$ data warehousing [24] relies on ingesting all data sources in a single system; $(ii)$ instead, mediation [35] leaves the data sources unchanged, and builds the integrated view of the data as a (virtual) layer above them.

An important question from which the design of a data integration system starts concerns the data models $(i)$ of the underlying data sources, and $(ii)$ of the integrated view of the data. Given the historical pre-eminence of relational data management systems, the early data integration systems as well as many follow-up works considered relational sources and a relational integrated schema. Quite early on, though, other data models have started being used to integrate heterogeneous data sources, e.g., object-oriented [16], Datalog-oriented [17], description logic-oriented [3], and Datalog and description logic-oriented [30, 19, 21] data models.

Two important trends influence the design of data integration architectures today.

1. First, **RDF**, the W3C's standard graph model for representing heterogeneous data, has gained wide acceptance for *modeling data* from a variety of domains, from life sciences to social networks, bibliography data, cultural resources etc.

2. Second, it has been noted that **application semantics**, usually expressed as an *ontology*, can be fruitfully used at the top (integrated) level, allowing users and applications to express their query and processing needs in the terms they are familiar with.

Part of this second trend, the term "Ontology-Based Data Access" (OBDA, in short) has been coined [33, 29] to designate data integration architectures where data is modeled using a set of unary predicates (or classes) and/or binary predicates (or properties), and described by known relationships between these predicates, e.g., any *Student* is a *Person*, and anyone having got a *Grade* in a *Class* is a *Student*). In OBDA settings, data is typically assumed stored in a relational database management system (RDBMS); *mappings* are then used to specify which parts of the stored data populate the ontology concepts and relations. OBDA has attracted significant attention in the research community, e.g., [18, 28, 27], and has been applied in several large real-life data integration settings, e.g. [25, 23].

The above examples illustrate the fact that an important part of domain knowledge can be encapsulated by *entailment* (or inference) *rules*, of the general form $\forall \bar{x}(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$, where $\bar{x}$ designates the free variables of the $\varphi$ and $\psi$ formulae, which may also use other existentially quantified variables. For instance, the sample rules above can be written as:

(1) $\forall x(S(x) \rightarrow P(x))$
(2) $\forall x(\exists y \exists z(GS(y,x) \wedge GC(y,z) \wedge C(z)) \rightarrow S(x))$

where the unary predicates $S$, $P$ and $C$ state that the respective variables are of *Student*, *Person*, respectively *Course* type, while the binary predicate $GS$ associates grades (first attribute) to students (second attribute), while $GC$ associates grades to courses. Some inference rules, such as (1) above, can be expressed between a a fixed number of concepts using a standardized vocabulary; for instance, the

RDF Schema ontology standard provides a keyword for relating exactly two classes $S, P$ by stating that the first is a subclass of the second. Other rules, such as (2) above, can take a more general form; they allow the human expert greater flexibility in describing the semantics of an application domain.

In this work, we consider the integration of *heterogeneous data of any data models*, under a *global RDF graph data model*; further, applications are offered the possibility to query the data with the help of $(i)$ an *ontology* describing its semantics and $(ii)$ *entailment rules* (both of *standard form* as per the RDF standard, and *user-specified* within a dialect that we describe). Figure 1 gives a broad view of this (focus on the yellow-background boxes at the top and the bottom now; the other components will be explained in due time). From a set of local databases $D_1, \ldots, D_n$, some data $\mathcal{E}$ of which begin accessible via mapppings $\mathcal{M}$ through an ontology $O$, associated with entailment rules $\mathcal{R}$, our goal is to compute the *certain answers* of the query $q$, on the data integration system thus composed.

Our contributions are as follows.

(1) We propose the first data integration architecture for computing certain query answers in this setting. It goes beyond comparable ones from the literature, focused either on relational data, or on a single non-relational one, e.g., JSON, by its support of data sources of heterogeneous data models, and (especially) by the ability to take into account an ontology and, separately, a set of entailment rules based on which certain answers are computed.

(2) To take into account domain knowledge (i.e., the ontology and entailment rules in our setting), classical query answering approaches would either materialize the consequences of the data and the knowledge (in warehouse data integration style), or, reformulate the query to integrate the relevant part of the knowledge. We propose a third, *original query answering approach*, based on a reduction to *view-based query answering*. We characterize precisely a class of problems for which this approach holds, present a new method for computing certain answers under these hypotheses, and formally establish the correctness of this method.

The rest of the paper is organized as follows. Section 2 introduces preliminary notions and Section 3 presents our problem statement. Section 4 outlines our query answering approach in relationship with the main existing approaches. Then, Section 5 defines a restricted setting in which our approach applies, and Section 6 proves the correctness of our method in this setting. We end with related work.

Proofs of our technical results are available in the appendix (Section 7).

# 2. PRELIMINARIES

We present the basics of the RDF graph data model (Section 2.1), of RDF reasoning used to make explicit the implicit information they encode (Section 2.2), as well as how they can be queried using the widely-considered SPARQL Basic Graph Pattern queries (Section 2.3). Finally, we recall the principles of query rewriting using views in an information integration context (Section 2.4), on which our solution to the problem tackled in this paper is built.

## 2.1 RDF Graph

| **RDF assertions** | Triple notation |
|---|---|
| Class assertion | $(\mathtt{s}, \tau, c)$ |
| Property assertion | $(\mathtt{s}, p, \mathtt{o})$ with $p \neq \tau$ |
| **RDFS constraints** | Triple notation |
| Subclass | $(\mathtt{s}, \prec_{sc}, \mathtt{o})$ |
| Subproperty | $(\mathtt{s}, \prec_{sp}, \mathtt{o})$ |
| Domain typing | $(\mathtt{s}, \hookleftarrow_{d}, \mathtt{o})$ |
| Range typing | $(\mathtt{s}, \hookrightarrow_{r}, \mathtt{o})$ |

**Table 1: RDF statements.**

RDF graphs build on three pairwise disjoint sets of values: $\mathscr{I}$ of IRIs (keys), $\mathscr{B}$ of blank nodes (labelled null modeling incomplete information [5]), and $\mathscr{L}$ of literals (constants).

An *RDF graph* $G$ is a set of well-formed triples $(\mathtt{s}, \mathtt{p}, \mathtt{o})$ from $(\mathscr{I} \cup \mathscr{B}) \times \mathscr{I} \times (\mathscr{L} \cup \mathscr{I} \cup \mathscr{B})$. A triple $(\mathtt{s}, \mathtt{p}, \mathtt{o})$ states that its *subject* $\mathtt{s}$ has the property $\mathtt{p}$ with the *object* value $\mathtt{o}$ [1]. We denote the set of all values (IRIs, blank nodes and literals) occurring in an RDF graph $G$ by $\mathrm{Val}(G)$, and $\mathrm{Bl}(G)$ its set of blank nodes.

Within an RDF graph, triples model either an *assertions* for unary relations called *classes* and for binary relations called properties *properties*, or *RDFS ontological constraints* between classes and properties. The RDFS constraints that can be used in an RDF graph $G$, which we denote $\mathrm{RDFS}(G)$, are of four flavours: subclass constraints, subproperty constraints, typing of the domain (first attribute) or of the range (second attribute) of a property. The triple notations we adopt for RDF graph's assertions and constraints are shown in Table 1. Further, within triples, we use $\_{:}b$ possibly with indices to denote blank nodes and strings between quotes to denote literals.

For instance, consider the following sample RDF graph:

$$G_{\text{ex}} = \{(:\text{Professor}, \prec_{sc}, :\text{Person}), (:\text{teaches}, \hookrightarrow_{r}, :\text{Course}) \\ (:\text{Fabian}, \tau, :\text{Professor}), (:\text{Fabian}, :\text{teaches}, \_{:}b), \\ (\_{:}b, :\text{label}, \text{"}\textit{Dance}\text{"}) \}$$

$G_{\text{ex}}$ models with triples, in this order, that professors are persons, that what is taught is a course, and that Fabian is a professor, who teaches somethings (identified with the blank node - hence unknown - value $\_{:}b$), whose label is "Dance". Further, this graph implicitly models that Fabian is a person, because professors are persons, and $\_{:}b$ is a course, because Fabian teaches it.

Finally, the notion of *homomorphism between RDF graphs* allows characterizing whether an RDF graph *simply entails*, i.e., is more specific than or subsumed by, another based on their explicit triples only.

Definition 1 (RDF graph homomorphism). *Let $G$ and $G'$ be two RDF graphs. A* homomorphism *from $G$ to $G'$ is a substitution $\varphi$ of* $\mathrm{Bl}(G)$ *by* $\mathrm{Val}(G)$*, and is the identity for the other $G$ values (IRIs and literals), such that $\varphi(G) \subseteq G'$, where $\varphi(G) = \{(\varphi(s), \varphi(p), \varphi(o)) \mid (s, p, o) \in G\}$.*

From now, we write $G' \models^{\varphi} G$ to state that $\varphi$ is a homomorphism from $G$ to $G'$, i.e., $G'$ simply entails $G$ due to $\varphi$.

## 2.2 RDF Entailment Rules

The semantics of an RDF graph consists of the explicit triples it contains, and of the implicit triples that can be derived using *RDF entailment rules*.

Definition 2 (RDF entailment rule). *An* RDF entailment rule *$r$ has the form $body(r) \rightarrow head(r)$, where $body(r)$*

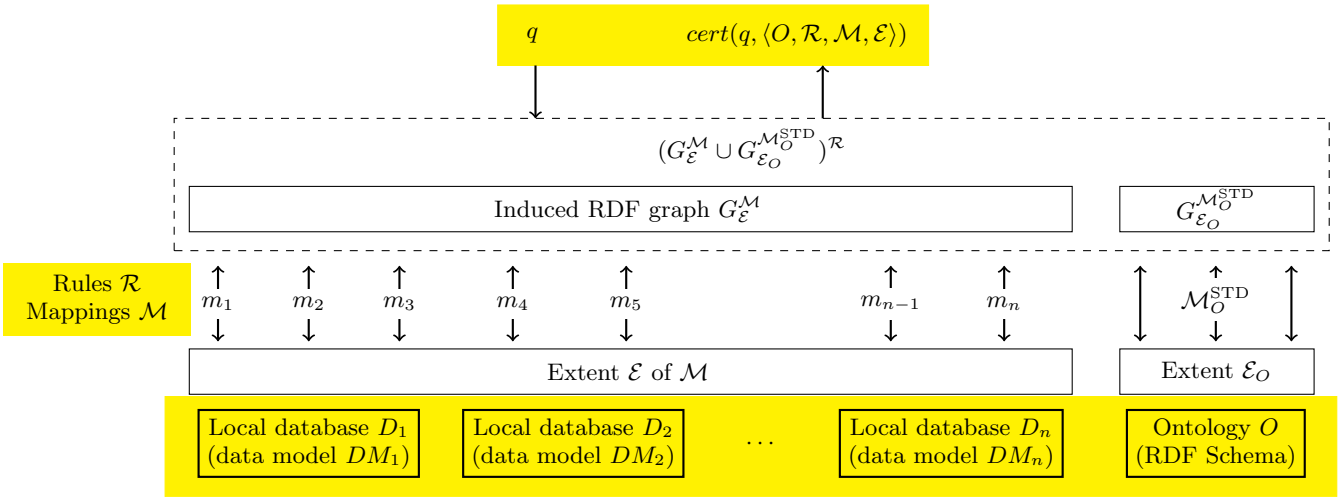


**Figure 1: Outline of an $O$-system architecture.**

| **Rule** [2] | Entailment rule |
|---|---|
| rdfs2 | $(\mathtt{p}, \hookleftarrow_d, \mathtt{o}), (\mathtt{s}_1, \mathtt{p}, \mathtt{o}_1) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$ |
| rdfs3 | $(\mathtt{p}, \hookrightarrow_r, \mathtt{o}), (\mathtt{s}_1, \mathtt{p}, \mathtt{o}_1) \rightarrow (\mathtt{o}_1, \tau, \mathtt{o})$ |
| rdfs5 | $(\mathtt{p}_1, \prec_{sp}, \mathtt{p}_2), (\mathtt{p}_2, \prec_{sp}, \mathtt{p}_3) \rightarrow (\mathtt{p}_1, \prec_{sp}, \mathtt{p}_3)$ |
| rdfs7 | $(\mathtt{p}_1, \prec_{sp}, \mathtt{p}_2), (\mathtt{s}, \mathtt{p}_1, \mathtt{o}) \rightarrow (\mathtt{s}, \mathtt{p}_2, \mathtt{o})$ |
| rdfs9 | $(\mathtt{s}, \prec_{sc}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$ |
| rdfs11 | $(\mathtt{s}, \prec_{sc}, \mathtt{o}), (\mathtt{o}, \prec_{sc}, \mathtt{o}_1) \rightarrow (\mathtt{s}, \prec_{sc}, \mathtt{o}_1)$ |
| ext1 | $(\mathtt{p}, \hookleftarrow_d, \mathtt{o}), (\mathtt{o}, \prec_{sc}, \mathtt{o}_1) \rightarrow (\mathtt{p}, \hookleftarrow_d, \mathtt{o}_1)$ |
| ext2 | $(\mathtt{p}, \hookrightarrow_r, \mathtt{o}), (\mathtt{o}, \prec_{sc}, \mathtt{o}_1) \rightarrow (\mathtt{p}, \hookrightarrow_r, \mathtt{o}_1)$ |
| ext3 | $(\mathtt{p}, \prec_{sp}, \mathtt{p}_1), (\mathtt{p}_1, \hookleftarrow_d, \mathtt{o}) \rightarrow (\mathtt{p}, \hookleftarrow_d, \mathtt{o})$ |
| ext4 | $(\mathtt{p}, \prec_{sp}, \mathtt{p}_1), (\mathtt{p}_1, \hookrightarrow_r, \mathtt{o}) \rightarrow (\mathtt{p}, \hookrightarrow_r, \mathtt{o})$ |

**Table 2: Sample RDF entailment rules.**

*and head($r$) are RDF graphs, respectively called* body *and* head *of the rule $r$.*

Built-in RDF entailment rules are defined in [2]. They produce implicit triples by exploiting the RDFS ontological constraints of an RDF graph. In this work, we consider the rule set defined in Table 2, denoted by $\mathcal{R}_{\text{RDFS}}$; all values except built-in properties denote blank nodes. For example, for Rule `rdfs9` used to propagate values from subclasses to their superclasses:

- body(`rdfs9`) $= \{(\mathtt{s}, \prec_{sc}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s})\}$
- head(`rdfs9`) $= \{(\mathtt{s}_1, \tau, \mathtt{o})\}$

where $\mathtt{s}, \mathtt{o}, \mathtt{s}_1$ are blank nodes.

The *direct entailment* of an RDF graph $G$ with a set of RDF entailment rules $\mathcal{R}$, denoted by $C_{G,\mathcal{R}}$, characterizes the set of implicit triples resulting from triggering (a.k.a. firing) the rules in $\mathcal{R}$ using the explicit triples of $G$ only. It is defined as:

$$C_{G,\mathcal{R}} = \{\varphi(\text{head}(r))^{\text{safe}} \mid \exists r \in \mathcal{R}, G \models^{\varphi} \text{body}(r) \text{ and there is no } \varphi' \text{ extension of } \varphi \text{ s.t. } G \models^{\varphi'} \text{body}(r) \cup \text{head}(r)\}$$

where $\varphi(\text{head}(r))^{\text{safe}}$ is $\varphi(\text{head}(r))$, where each blank node in $\text{Bl}(\text{head}(r)) \setminus \text{Bl}(\text{body}(r))$ is replaced by a fresh blank node. Note that the condition "there is no $\varphi'$ extension of $\varphi$ s.t. $G \models^{\varphi'} \text{body}(r) \cup \text{head}(r)$" prevents the production of obviously redundant triples.

Without loss of generality, as in the RDF standard, we only consider well-formed entailed triples, i.e., from $(\mathscr{I} \cup \mathscr{B}) \times \mathscr{I} \times (\mathscr{L} \cup \mathscr{I} \cup \mathscr{B})$.

For instance, the rule `rdfs9` applies to the RDF graph $G_{\text{ex}}$: $G_{\text{ex}} \models^{\varphi}$ body(`rdfs9`) through the homomorphism $\varphi$ defined as $\{s \mapsto$ :Professor, $o \mapsto$ :Person, $s_1 \mapsto$ :Fabian$\}$. The rule `rdfs3` also applies; the direct entailment of $G_{\text{ex}}$ with $\mathcal{R}_{\text{RDFS}}$ contains exactly the triples (:Fabian, $\tau$, :Person) and (_:$b$, $\tau$, :Course).

The *saturation* of an RDF graph allows materializing the semantics of an RDF graph, by iteratively augmenting this graph with the triples it directly entails using a set $\mathcal{R}$ of RDF entailment rules, till a fixpoint is reached.

We formalize this as the sequence $(G_i^{\mathcal{R}})_{i \in \mathbb{N}}$ of RDF graphs recursively defined as follows:

- $G_0^{\mathcal{R}} = G$, and
- $G_{i+1}^{\mathcal{R}} = G_i^{\mathcal{R}} \cup C_{G_i^{\mathcal{R}},\mathcal{R}}$ for $0 \leq i$.

Definition 3 (Saturation of RDF graph). *Let $G$ be an RDF graph, and $\mathcal{R}$ be a set of entailment rules. The* saturation *of $G$ w.r.t $\mathcal{R}$, denoted by $G^{\mathcal{R}}$, is defined by:*

$$G^{\mathcal{R}} = \cup_{i \in \mathbb{N}} G_i^{\mathcal{R}}.$$

The saturation of an RDF graph by any subset of $\mathcal{R}_{\text{RDFS}}$ is finite [2]. In the preceding example, the saturation of $G_{\text{ex}}$ w.r.t. $\mathcal{R}_{\text{RDFS}}$ is completed by the first direct entailment, hence $(G_{\text{ex}})^{\mathcal{R}_{\text{RDFS}}} = G_{\text{ex}} \cup C_{G_{\text{ex}},\mathcal{R}_{\text{RDFS}}}$.

Finally, the notion of a homomorphism between RDF graphs is also used to characterize whether an RDF graph *entails* another w.r.t. a set of RDF entailment rules, i.e., in the presence of implicit triples. An RDF graph $G$ entails an RDF graph $G'$ w.r.t. a set $\mathcal{R}$ of RDF entailment rules, noted $G \models_{\mathcal{R}}^{\varphi} G'$, whenever there is a homomorphism $\varphi$ from $G'$ to $G^{\mathcal{R}}$. From now, we will just write $G \models_{\mathcal{R}} G'$ when a particular $\varphi$ is not relevant to the discussion.

## 2.3 Basic Graph Pattern Queries

A popular fragment of the SPARQL query language for RDF graphs is that of basic graph pattern queries, i.e., the

SPARQL conjunctive queries. It builds on the notion of basic graph pattern, which generalizes RDF graphs with variables.

We assume given a set of variables $\mathscr{V}$ disjoint from $\mathscr{I} \cup \mathscr{B} \cup \mathscr{L}$. A *basic graph pattern* (BGP) is a set of *triple patterns* belonging to $(\mathscr{I} \cup \mathscr{B} \cup \mathscr{V}) \times (\mathscr{I} \cup \mathscr{V}) \times (\mathscr{I} \cup \mathscr{B} \cup \mathscr{L} \cup \mathscr{V})$.

For a BGP $P$, we note Var(P) the set of variables occurring in $P$ and by Bl($P$) its set of blank nodes.

A basic graph pattern query is defined as follows:

Definition 4 (BGP query). *A* BGP query *(BGPQ) $q$ is of the form $q(\bar{x}) \leftarrow P$, where $P$ is a BGP also denoted by body($q$) and $\bar{x} \subseteq$ Var(P). The* arity *of $q$ is $|\bar{x}|$.*

The semantics of a BGPQ is defined in terms of the homomorphisms that exist between its BGP body and the interrogated RDF graph, i.e., in terms of the possible matches of the BGP onto the RDF graph explicit and implicit triples.

Definition 5 (BGP to RDF graph homomorphism). *A* homomorphism *from a BGP $P$ to an RDF graph $G$ is a substitution $\varphi$ of* Bl($P$)∪Var(P) *by* Val($G$) *and is the identity elsewhere such that $\varphi(P) \subseteq G$ with $\varphi(P) = \{(\varphi(s), \varphi(p), \varphi(o)) \mid (s, p, o) \in P\}$. We write $G \models^{\varphi} P$ to state that $\varphi$ is a homomorphism from $P$ to $G$.*

Definition 6 (BGPQ answers). *The* answer set to a BGPQ *$q$ on an RDF graph $G$ w.r.t. a set $\mathcal{R}$ of RDF entailment rules is:*

$$q(G, \mathcal{R}) = \{\varphi(\bar{x}) \mid G \models^{\varphi}_{\mathcal{R}} \textit{body}(q)\}$$

*If $\bar{x} = \emptyset$, $q$ is a* Boolean *query and the answer to $q$ is false when $q(G) = \emptyset$ and true when $q(G) = \{\langle\rangle\}$.*

We notice that the answers of a BGPQ on an RDF graph may be composed by blank nodes.

In the following, we consider without loss of generality that BGPQs do not contain blank nodes, as a blank node appearing in a query can be equivalently replaced with a fresh variable.

For example, consider the BGPQ $q(x) \leftarrow (x, \tau, :\text{Course})$ asking for all courses in the RDF graph $G_{ex}$, i.e., for all the resources that are explicitly or implicitly of type Courses in $G_{ex}$. There is one homomorphism from the BGP body($q$) to saturated RDF graph $(G_{\text{ex}})^{\mathcal{R}_{\text{RDFS}}}$ defined by $\varphi = \{x \mapsto \_{:}b\}$, hence the answer to $q$ on $G_{\text{ex}}$ w.r.t. $\mathcal{R}_{\text{RDFS}}$ is $\_{:}b$. We remark that the answers set would be empty for $q$ on $G_{\text{ex}}$ w.r.t. an empty set of RDF entailment rules.

Further, we will rely on the *saturation of a BGP w.r.t. an RDFS ontology* by a set of RDF entailment rules, defined in [15]. Just like for RDF graphs, we have to define a homomorphism from an RDF graph (in particular a rule body) to a BGP. Then, the definition of BGP saturation is the same as for RDF graphs, up to replacing variables by blank nodes in the definition of a homomorphism.

The saturation of a BGPQ contains in its body all the triples entailed from the BGPQ body and a given ontology, but not those entailed by the ontology alone as illustrated by Figure 7:

Definition 7 (BGPQ saturation [15]). *Let $\mathcal{R}$ be a set of RDF entailment rules, $O$ a set of RDFS statements (the ontology), and $q$ a BGPQ. The* saturation *of $q$ w.r.t. $O$, denoted by $q^{\mathcal{R},O}$, is the BGPQ with the same answer variables as $q$ and whose body, denoted by body($q^{\mathcal{R},O}$), is the maximal subset of $(\textit{body}(q) \cup O)^{\mathcal{R}}$ such that for any of its subsets $S$: if $O \models_{\mathcal{R}} S$ holds, then body($q$) $\models_{\mathcal{R}} S$ holds.*

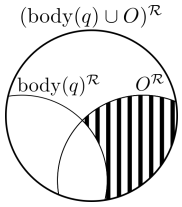


**Figure 2: The cercle represents $(\textbf{body}(q) \cup O)^{\mathcal{R}}$, the hatched area is removed from $q^{\mathcal{R},O}$, because it is consequence of $O$ only, hence not relevant to $q$.**

Consider the RDFS ontology $O_{\text{ex}}$ from $G_{\text{ex}}$, i.e., RDFS($G_{\text{ex}}$) equals to $\{(:\text{Professor}, \prec_{sc}, :\text{Person}), (:\text{teaches}, \hookrightarrow_r, :\text{Course})\}$ and the BPGQ $q(x) \leftarrow (x, :\text{teaches}, y)$ asking for professor teaching at least one course. The saturation $(q \cup O_{\text{ex}})^{\mathcal{R}_{\text{RDFS}}}$ contains exactly $(x, :\text{teaches}, y)$, $(y, \tau, :\text{Course})$ and the triples of $O_{\text{ex}}$. By pruning from this set the triples only entailed by $O_{\text{ex}}$ w.r.t. $\mathcal{R}_{\text{RDFS}}$, i.e., those in $O_{\text{ex}}$ here, the saturation of $q$ w.r.t. $O_{\text{ex}}$ and $\mathcal{R}_{\text{RDFS}}$ is $q^{\mathcal{R}_{\text{RDFS}}, O_{\text{ex}}}(x) \leftarrow (x, :\text{teaches}, y)$, $(y, \tau, :\text{Course})$.

Finally, similarly to RDF graphs, the saturation of a query w.r.t. an ontology and a subset of RDFS entailment rules is finite.

## 2.4 Query Rewriting using Views

The second scientific area (beyond OBDA) on which we base our approach, is view-based query rewriting. Since it has mainly been studied in a relational setting, we recall its main concepts as laid out e.g., in [22].

An integration system $\mathcal{I}$ is made of a *global schema* $S$, that is, a set of relations, and of a set of *views*. Each such view $V$ specifies one way in which the data from a source $D$ is *connected to* the schema $S$. The view binds together two components, one referring to $D$ and one to $S$, the global schema: $(i)$ $V^D$ is a query over the schema of $D$. It specifies *which $D$ data is exposed* by $V$ to the integration system. Its result $V^D(D)$, called *the extension of $V$*, is the maximum amount of data that one can get from $D$ through $V$. In the typical relational setting, $V^D(D)$ is a set of tuples. $(ii)$ $V(\bar{x}) \leftarrow \varphi(\bar{x})$ is a query over the global schema $S$; it specifies *how the data exposed by $V$ fits in the global integration schema*, i.e., how one can use it to answer queries over the global $S$. $\varphi(\bar{x})$ is typically a Datalog or SQL query.

We use $V(\mathcal{I})$ to designate *all the data available through the integration system $\mathcal{I}$* (including data that may be supplied by other views) *as an answer to the query $V(\bar{x}) \leftarrow \varphi(\bar{x})$*. For instance, if $V_1(\bar{x}) \leftarrow \varphi(\bar{x})$ is "all conference publications", $V_1(\mathcal{I})$ comprises all conference publications one could obtain through the integration system (whether through view $V_1$ or any other). In general, in an integration setting, it is assumed that the views are *sound*, that is $V^D(D) \subseteq V(\mathcal{I})$. This is because in general, several sources (i.e., views) may bring useful information of the same kind, e.g., no single view comprises all conference papers, and when querying $\mathcal{I}$, one typically wants to get all the papers known in the system. This vision corresponds to the Open World Assumption (OWA) [5].

The so-called *certain answers* to a query on $\mathcal{I}$ are defined based on all the instances of $S$ consistent with the views

and their extensions, i.e., for which $V^D(D) \subseteq V(\mathcal{I})$ holds for each view $V$.

DEFINITION 8 (CERTAIN ANSWERS UNDER OWA [22]). *Let $\mathcal{V}$ be a set of view definitions on a global schema $S$, and $\mathcal{E}$ the set of extensions for the the views in $\mathcal{V}$. Let $q$ be a conjunctive query on $S$. The tuple $\bar{t}$ is a* certain answer *to $q$ w.r.t. $(\mathcal{V}, \mathcal{E})$, if $\bar{t} \in q(I)$, for each instance $I$ of $S$ consistent with $(\mathcal{V}, \mathcal{E})$.*

Note that the above views integrate data sources in a global schema following the so-called *local-as-view* approach. An alternative *global-as-view* approach exists [22], but is not relevant to the purpose of this work, hence is not further discussed.

From a practical viewpoint, the certain answers (answers for short) can be computed by rewriting a query using the available views, so as to obtain a rewritten query in terms of the view relations, called a *view-based rewriting* (or *rewriting*, in short), directly evaluable on the extensions of the views. From a theoretical viewpoint, a rewriting must be contained in the original query in order to provide correct answers to it.

DEFINITION 9 (QUERY CONTAINMENT). *For two queries $q_1, q_2$ on a global schema $S$ or on a set of views $\mathcal{V}$, we say that $q_1$* is contained in *$q_2$, if for all set $\mathcal{E}$ of extensions of $\mathcal{V}$ and for all instance of $S$ consistent with $(\mathcal{V}, \mathcal{E})$, the answer set of $q_1$ is included in that of $q_2$. The two queries are said* equivalent*, if $q_1$ is contained in $q_2$* and *$q_2$ is contained in $q_1$.*

An equivalent rewriting for a query may not always exist, depending on the views and the query. In such cases, *maximally contained rewritings* are guaranteed to produce all the answers that the system $\mathcal{I}$ may bring to the query:

DEFINITION 10 (MAXIMALLY CONTAINED REWRITING [22]). *Let $q$ be a query on $S$, $\mathcal{V}$ a set of views on $S$ and $\mathscr{L}$ a query language. A query $q_r$ is a* maximally contained rewriting *of $q$ using $\mathcal{V}$ w.r.t. $\mathscr{L}$ if:*

- *$q_r$ is query in $\mathscr{L}$ on views $\mathcal{V}$ only;*
- *$q_r$ is contained in $q$;*
- *there does not exist a query $q'_r \in \mathscr{L}$ on $\mathcal{V}$ such that $q_r$ is contained in $q'_r$, $q'_r$ is contained in $q$ and $q'_r$ is not equivalent to $q_r$.*

The following known result shows that for conjunctive views, a maximally contained rewriting in the language of unions of conjunctions of views compute exactly the certain answers. This theorem is inspired by the Theorem 3.2 of [4]:

THEOREM 1. *Let $\mathcal{V}$ be a set of conjunctive views on $S$ and $q$ be a conjunctive query on $S$. If a query $q_r$ is a maximally contained rewriting of $q$ using $\mathcal{V}$ w.r.t. the language of unions of conjunctive queries, then for each set $\mathcal{E}$ of extensions of $\mathcal{V}$, the set of answers of $q_r$ on $\mathcal{E}$ is exactly the set of certain answers of $q$ on $(\mathcal{V}, \mathcal{E})$.*

# 3. PROBLEM STATEMENT

In this section, we first discuss how heterogeneous data sources can be integrated in and queried through an RDF graph using the notion of *mappings* (Section 3.1). Then, we extend this approach to the use of an *RDFS ontology* allowing to interpret the data integrated from the sources (Section 3.2).

## 3.1 RDF integration of data sources

We consider a set of *heterogeneous data sources* $D_1, \ldots, D_n$ with respective data models $\mathrm{DM}_1, \ldots, \mathrm{DM}_n$. For each data model, we assume available a query language, and assume queries expressed in this language return *tuples of bindings*, which is the usual case e.g. for SQL, XPath (and any tree pattern language), and BGPQs. Our first step toward handling this data model heterogeneity is to assume that data from each source is exposed to the integration system through a set of *mappings*:

DEFINITION 11 (MAPPING AND MAPPING EXTENSION). *A* mapping *$m$ is of the form $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$ where $q_1$ and $q_2$ are two queries with the same arity. The* body *of the mapping $m$ is $q_1$ and its* head *is $q_2$. An* extension *of $m$, denoted* $\mathrm{ext}(m)$*, is the set of answers to $q_1$.*

Intuitively, a mapping specifies how to translate the tuples returned by a query $q_1$ on a data source into answers of a query $q_2$ on a global schema. Keep in mind that a mapping is a *specification only*, i.e., it does not always lead to executing $q_1$ on that dataset, especially when no user query is interested in that data.

Given a set of mappings $\mathcal{M}$, we call an *extent of $\mathcal{M}$* a set $\mathcal{E}$ of extensions defined by $\mathcal{E} = \{\mathrm{ext}(m) \mid m \in \mathcal{M}\}$, i.e., the data made available by the sources to the integration system.

In our setting, we rely on mappings which integrate source data into RDF graphs:

DEFINITION 12 (RDF MAPPING). *An* RDF mapping *is a mapping $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$, where $q_2$ is a BGPQ.*

Importantly, the above definition makes no assumption on the query language of $q_1$, so that RDF mappings can be used to integrate heterogeneous data sources, e.g., relational, XML, RDF ones, into an RDF graph as follows:

DEFINITION 13 (INDUCED RDF GRAPH). *Given a set $\mathcal{M}$ of RDF mappings and an extent $\mathcal{E}$ of $\mathcal{M}$, the RDF graph induced by $\mathcal{M}$ and $\mathcal{E}$ is:*

$$G_{\mathcal{E}}^{\mathcal{M}} = \bigcup_{m \in \mathcal{M}} \{(body(q_2)(\bar{t}))^{safe} \mid m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x}), \bar{t} \in \mathrm{ext}(m)\}$$

*where*

- *$body(q_2)(\bar{t})$ is the set of triples of $body(q_2)$ in which we replace the $\bar{x}$ tuple of answer variables by the tuple $\bar{t}$.*
- *$(body(q_2)(\bar{t}))^{safe}$ is $body(q_2)(\bar{t})$ in which we replace each variable by a fresh blank node (recall we consider w.l.o.g. that $body(q_2)$ contains no blank nodes).*

From now, to simplify the notation, we may also specify an RDF mapping $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$ simply as:

$$m = q_1(\bar{x}) \rightsquigarrow \mathrm{body}(q_2).$$

EXAMPLE 1. *Consider the following data sources of a university department. A first relational source $S_p$ pairs professors with the courses they teach. Let $q_p(professor, course)$ be the query on $S_p$ returning such pairs. A second JSON source $S_s$ provides pairs students with the courses they register for. Let $q_s(student, course)$ be the query on $S_s$ that retrieves such pairs. Finally, let $q_l(professor, courseLabel)$ be a BGPQ asking for labels of courses appearing in each professor's page on the RDF website of the university department.*

*Consider now that these sources are integrated using the following set $\mathcal{M}_{\mathrm{ex}}$ of RDF mappings:*

$m_p = q_p(professor, course) \rightsquigarrow (professor, :\text{teaches}, course)$
$m_s = q_s(student, course) \rightsquigarrow (student, :\text{registeredFor}, course)$
$m_l = q_l(professor, courseLabel) \rightsquigarrow \begin{matrix}(professor, :\text{teaches}, x),\\ (x, :\text{label}, courseLabel)\end{matrix}$

*where* :teaches, :registeredFor, :courseLabel *are IRIs. We note that, in mapping* $m_l$*, the variable* $x$ *represents the existence of an unknown course.*

*Finally, assume that the extent* $\mathcal{E}_{\text{ex}}$ *of* $\mathcal{M}_{\text{ex}}$*, defined by that answers to* $q_p, q_s, q_l$ *is:*

$\text{ext}(m_p) = \{(:\text{Fabian}, :\text{SemWeb})\}$
$\text{ext}(m_s) = \{(:\text{Alice}, :\text{SemWeb}), (:\text{Alice}, :\text{RelDB})\}$
$\text{ext}(m_l) = \{(:\text{Fabian}, \text{"Dance"})\}$

*In this setting, the induced RDF graph* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *is*

$$\begin{aligned} G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}} \quad = \quad & \{(:\text{Fabian}, :\text{teaches}, :\text{SemWeb}),\\ & (:\text{Alice}, :\text{registeredFor}, :\text{SemWeb}),\\ & (:\text{Alice}, :\text{registeredFor}, :\text{RelDB}),\\ & (:\text{Fabian}, :\text{teaches}, \_:b), (\_:b, :\text{label}, \text{"Dance"})\} \end{aligned}$$

*in which, for instance, the last two triples* $(:\text{Fabian}, :\text{teaches}, \_:b)$, $(\_:b, :\text{label}, \text{"Dance"})$ *results from instanciating the body of* $m_l$ *with the results of* $q_l$*, i.e.,* $\text{ext}(m_l)$*, which also instanciate the head of* $m_l$ *to* $(:\text{Fabian}, :\text{teaches}, x)$, $(x, :\text{label}, \text{"Dance"})$. *The triples* $(:\text{Fabian}, :\text{teaches}, \_:b)$, $(\_:b, :\text{label}, \text{"Dance"})$ *are therefore added to* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$*, after having replaced the* $x$ *variable to a fresh blank node (recall the safe operation in Definition 13).*

We define an *RDF system* as a triple $S = \langle \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$, where $\mathcal{R}$ is the set of entailment rules under consideration defining the reasoning power of the system, $\mathcal{M}$ the set of mappings that integrate data sources into the RDF system and $\mathcal{E}$ the extent thereof.

For such systems, we recast the well-known notion of answers to a BGPQ as follows:

Definition 14 (Certain answer set). *Given a set* $\mathcal{R}$ *of RDF entailment rules, a set* $\mathcal{M}$ *of mappings and an extent* $\mathcal{E}$ *of* $\mathcal{M}$*, the* certain answer set *of a BGPQ* $q$ *against the RDF system* $S = \langle \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$ *is:*

$$\text{cert}(q, S) = \{\varphi(\bar{x}) \mid G^{\mathcal{M}}_{\mathcal{E}} \models^{\varphi}_{\mathcal{R}} q(\bar{x})\}$$

*where* $\varphi(\bar{x})$ *is made of IRIs and literals only.*

Example 2 (continued). *Consider the RDF system* $S = \langle \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$*, where* $\mathcal{M}$ *and* $\mathcal{E}$ *are these of the preceding example, and* $\mathcal{R}$ *only contains the entailment rule* $r_{\text{ex}} = (s, :\text{registeredFor}, c) \rightarrow (z, :\text{teaches}, c)$*, stating that if some student is registered for some course, then this course is taught by some teacher.*

*Suppose that one asks the query* $q(x) = (y, :\text{teaches}, x)$ *retrieving the courses taught by some teacher. The answers to this query against* $S$ *are those that can be obtained from the saturation w.r.t.* $\mathcal{R}$ *of the RDF graph induced by* $\mathcal{M}, \mathcal{E}$*, i.e., the saturation of* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *from Example 1.*

*The first triple of* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *leads to the answer* :SemWeb *to* $q$*. We remark that* $\_:b$ *is not an answer to* $q$ *though there is an obvious homomorphism from* $q$ *to the fourth triple in* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$*; this is because blank nodes (i.e., unknown values) are forbidden in query answers.*

*The saturation of* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *with* $\mathcal{R}$ *only adds the implicit triple* $(\_:a, :\text{teaches}, :\text{RelDB})$*, hence a second answer* :RelDB *found using reasoning. We stress here that the saturation of* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *with* $\mathcal{R}$ *does not add a triple* $(\_:c, :\text{teaches}, :\text{SemWeb})$ *because of the presence of the second* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *triple; indeed, such a triple would be redundant w.r.t. the first* $G^{\mathcal{M}_{\text{ex}}}_{\mathcal{E}_{\text{ex}}}$ *triple and saturation avoids this (recall Section 2.2).*

*The certain answers* $\text{cert}(q, S)$ *to* $q$ *on* $S$ *are therefore equals to* $\{:\text{SemWeb}, :\text{RelDB}\}$.

## 3.2 Ontology-based RDF integration of data sources

We now extend RDF systems with the ability to use an ontology, hence domain knowledge, when integrating data sources.

To this aim, we introduce specific mappings whose goal is to *feed an RDF system with the RDFS ontological constraints* it must consider:

Definition 15 (Standard RDFS mappings of $O$). *Given an RDFS ontology* $O$ *and* $\mathcal{R}$ *a set of RDF entailment rules, the* standard RDFS mappings of $O$*, denoted* $\mathcal{M}^{STD}_{O}$*, are as follows:*

$m_{subClassOf} = q_{subClassOf}(\mathtt{s}, \mathtt{o}) \rightsquigarrow (\mathtt{s}, \prec_{sc}, \mathtt{o})$
$m_{subPropertyOf} = q_{subPropertyOf}(\mathtt{s}, \mathtt{o}) \rightsquigarrow (\mathtt{s}, \prec_{sp}, \mathtt{o})$
$m_{domain} = q_{domain}(\mathtt{s}, \mathtt{o}) \rightsquigarrow (\mathtt{s}, \hookleftarrow_{d}, \mathtt{o})$
$m_{range} = q_{range}(\mathtt{s}, \mathtt{o}) \rightsquigarrow (\mathtt{s}, \hookrightarrow_{r}, \mathtt{o})$

*with as body mapping queries:*

$q_{subClassOf}(\mathtt{s}, \mathtt{o}) \leftarrow (\mathtt{s}, \prec_{sc}, \mathtt{o})$
$q_{subPropertyOf}(\mathtt{s}, \mathtt{o}) \leftarrow (\mathtt{s}, \prec_{sp}, \mathtt{o})$
$q_{domain}(\mathtt{s}, \mathtt{o}) \leftarrow (\mathtt{s}, \hookleftarrow_{d}, \mathtt{o})$
$q_{range}(\mathtt{s}, \mathtt{o}) \leftarrow (\mathtt{s}, \hookrightarrow_{r}, \mathtt{o})$

*Given* $\mathcal{R}$ *a set of RDF entailment rules, the* extension of $O$'s standard mappings*, denoted* $\mathcal{E}_O$*, contains the following extensions:*

$\text{ext}(m_{subClassOf}) = q_{subClassOf}(O, \mathcal{R})$
$\text{ext}(m_{subPropertyOf}) = q_{subPropertyOf}(O, \mathcal{R})$
$\text{ext}(m_{domain}) = q_{domain}(O, \mathcal{R})$
$\text{ext}(m_{range}) = q_{range}(O, \mathcal{R})$

Above, $q_{\text{x}}(O, \mathcal{R})$ denotes the answer set of the BGPQ $q_{\text{x}}$ evaluated on $O$ as an RDF graph with $\mathcal{R}$ as entailment rules (recall BGPQ answers from Definition 6).

Example 3. *Consider the following extension of the ontology* $O_{\text{ex}}$ *introduced in Section 2:*

$$\begin{aligned} O_{\text{ex}} \quad = \quad & \{(:\text{Professor}, \prec_{sc}, :\text{Person}), (:\text{Student}, \prec_{sc}, :\text{Person}),\\ & (:\text{teaches}, \hookleftarrow_{d}, :\text{Professor}), (:\text{teaches}, \hookrightarrow_{r}, :\text{Course})\\ & (:\text{registeredFor}, \hookleftarrow_{d}, :\text{Student}),\\ & (:\text{registeredFor}, \hookrightarrow_{r}, :\text{Course})\} \end{aligned}$$

*Assume a standard integration of* $O$ *as per Definition 15, with the set* $\mathcal{R}_{\text{RDFS}}$ *of RDF entailment rules (Table 2). The extension of* $m_{\text{domain}}$ *in* $\mathcal{M}^{\text{STD}}_{O_{\text{ex}}}$ *is* $q_{\text{domain}}(O_{\text{ex}}, \mathcal{R}_{\text{RDFS}})$*, i.e.,* $\{(:\text{teaches}, \hookleftarrow_{d}, :\text{Professor}), (:\text{registeredFor}, \hookleftarrow_{d}, :\text{Student}),$ $(:\text{teaches}, \hookleftarrow_{d}, :\text{Person}), (:\text{registeredFor}, \hookleftarrow_{d}, :\text{Person})\}$. *The first two triples come from* $O$*, which the two others are all generalizations thereof, here obtained using the RDF entailment rule* `ext1` *together with the fact that both students and*

*professors are persons. This is similar for the other three mappings. As a result, all $O$ RDFS triples, both explicit and implicit, are part of the induced RDF graph $G^{\mathcal{M}^{\text{STD}}_{O_{\text{ex}}}}_{\mathcal{E}_{O_{\text{ex}}}}$.*

The standard mappings $\mathcal{M}^{\text{STD}}_O$ and their extent $\mathcal{E}_O$ are defined so that **the induced RDF graph contains the knowledge of the saturation of $O$ by $\mathcal{R}$.** We will see later (Property 8), that this holds under some restrictions on entailment rules and ontology.

We are now able to define an ontology-based RDF system, as depicted in Figure 1:

DEFINITION 16 ($O$-SYSTEM). *Given an RDFS ontology $O$, a set $\mathcal{R}$ of RDF entailments rules, a set $\mathcal{M}$ of RDF mappings and an extent $\mathcal{E}$ thereof, the* RDF system with ontology *(or $O$-system, in short) $S = \langle O, \mathcal{R}, \mathcal{M}, \mathcal{E}\rangle$ is the RDF system $\langle \mathcal{R}, \mathcal{M} \cup \mathcal{M}^{STD}_O, \mathcal{E} \cup \mathcal{E}_O\rangle$.*

In this ontology-based RDF integration setting, the problem we formally study in this paper is the following:

PROBLEM 1. *Given an RDFS ontology $O$, a set $\mathcal{R}$ of RDF entailment rules, a set $\mathcal{M}$ of instance mappings and a set $\mathcal{E}$ of their extensions, compute the* certain answer set *of a BGPQ $q$ against the $O$-system $S = \langle O, \mathcal{R}, \mathcal{M}, \mathcal{E}\rangle$.*

# 4. REWRITING-BASED QUERY ANSWERING: A THIRD WAY

Below, we first review the two main existing approaches to query data in similar settings (Section 4.1), i.e., that involve mappings and ontological knowledge; these are based on instance materialization, respectively, query reformulation. Each has its own drawbacks: materialization requires time to compute and space to store, and it necessitates maintenance when the data, rules and/or mappings change, whereas reformulation may lead to very expensive query evaluation.

We then propose a new approach which does not suffer from these drawbacks (Section 4.2). The key idea is an innovative usage of the well-known technique of view-based rewriting, applied on a set of virtual views, which we obtain by *saturating the heads of the mappings* $\mathcal{M}$. This approach avoids the pitfalls of both materialization and reformulation. However, it provides complete answer sets only under some restrictions, that we will detail in due time (Section 5).

## 4.1 Existing Approaches: Materialization and Reformulation

The simplest family of methods consists in materializing both the data imported by the mappings (as in data warehousing) and inferences computed from ontological knowledge. In our setting, given an RDF system $S = \langle \mathcal{R}, \mathcal{M}, \mathcal{E}\rangle$ (including, but not limited to, $O$-systems), we would materialize the RDF graph $G^{\mathcal{M}}_{\mathcal{E}}$, then saturate it with the rules to obtain $(G^{\mathcal{M}}_{\mathcal{E}})^{\mathcal{R}}$; certain answers to queries would be computed against this materialization. The benefits and drawbacks of this approach are well-known. Query answering can be very fast as one forgets about mappings and rules at query time. On the other hand, there are several situations in which this double level of materialization is not possible, due to the volume of the obtained data or even the non-termination of the saturation by the rules (note that, whereas termination is ensured for the saturation by standard RDFS entailment rules, it is not the case for more general RDF entailment rules, see Section 5.2 for details). Moreover, this approach is not adapted to contexts where data changes frequently, as the materialization has to be recomputed after the updates, which involves triggering again both mappings and rules.

At the other end of the spectrum, one can avoid any kind of materialization as follows: at query time, a query is first reformulated using the ontological knowledge, then the obtained reformulation is rewritten into another query using the mappings. In our setting, the reformulation step would take as input an RDFS ontology $O$, a set $\mathcal{R}$ of RDF entailment rules and a BGPQ $q$, and would output a reformulation $q'$ of $q$ using $O$ and $\mathcal{R}$. We need reformulation to be sound and complete, i.e., for any RDF graph $G$ whose set of RDFS statements is $O$, the certain answers to $q'$ against $G$ alone must be the certain answers to $q$ against $G$ w.r.t. $\mathcal{R}$ (in other words, against $G^{\mathcal{R}}$). The rewriting step would then turn $q'$ into a maximally contained rewriting $q'_r$ (see Section 2.4), which can be asked against the extent $\mathcal{E}$. One could also consider a mixed approach which, on the one hand, materializes the graph obtained from the mappings (yielding $G^{\mathcal{M}}_{\mathcal{E}}$), and on the other hand reformulates queries, subsequently answered against the materialized graph.

Such an approach is typical of the OBDA setting [33, 29], in which, for the systems implemented so far (see, e.g., [11] for one of the most complete systems), the knowledge is encoded in light description logic languages (typically the OWL 2 QL dialect, which relies on the description logic DL-Lite). However, even for relational conjunctive queries and simple ontological languages that guarantee the termination of the reformulation step, the obtained reformulation can be exponentially larger than the initial query, which may question the practical usability of the technique[1]. Notably, with the aim of improving the efficiency of reformulation-based query answering, more general reformulation languages have been investigated (e.g., [34, 9, 26, 10, 27]).

Most works on query answering in the presence of an ontology make the assumption that data is described over the same vocabulary as the ontology, hence they do not study *the interaction between the ontology and mappings*. Moreover, most contributions are restricted to conjunctive queries, which, when restricted to binary predicates, correspond to specific BGPQs on triples of the form $(:s, \tau, :c)$ and $(:s, p, :o)$, where $p$ cannot be a blank node. In [20], general BGPQs were considered and a reformulation technique was introduced for a specific subset of standard RDF entailment rules (namely the DB fragment, consisting of RDF rules 2,3,7 and 9 (see Table 2). However, this technique has not been so far extended to more RDF entailment rules. Hence, to pursue this approach using RDFS rules, one should first define a sound and complete reformulation technique for BGPQs and RDF entailment rules.

## 4.2 Rewriting-Based Query Answering

We now outline our approach, which proceeds in two main steps. Offline, the mapping heads are saturated by the ontology and the entailment rules. Then, each query is rewritten

[1]Note that most papers in the Knowledge Representation area call query rewriting what we call here query reformulation to avoid confusion with view-based query rewriting.

using the saturated mappings, as if these were view definitions. As already mentioned, this approach does not provide complete answer sets for any O-system, hence we will have to put syntactic restrictions on the allowed ontologies, entailment rules and mappings (see the next section). In a nutshell, we reduce the query answering problem in restricted O-systems to a view-based query answering problem.

Note that saturating the mapping heads and saturating the data produced by the mappings have very different costs in terms of volume and robustness to data dynamicity. Indeed, the size of the mapping heads is expected to be small, and the mappings are by definition independent from changes in data. Hence, on the one hand this technique avoids the main drawbacks of materialization-based approaches, and on the other hand it also avoids those of query reformulation.

We first define the saturation of mappings.

Definition 17 (Saturation of Mapping). *Given an RDFS ontology $O$, a set $\mathcal{R}$ of RDF entailment rules and an RDF mapping $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$, the* saturation of the mapping *$m$ w.r.t. $\mathcal{R}$ and $O$ is defined as:*

$$m^{\mathcal{R},O} = q_1(\bar{x}) \rightsquigarrow q_2^{\mathcal{R},O}(\bar{x})$$

*Accordingly, the* saturation of a set of mappings *$\mathcal{M}$ w.r.t. $\mathcal{R}$ and $O$ is $\mathcal{M}^{\mathcal{R},O} = \{m^{\mathcal{R},O} \mid m \in \mathcal{M}\}$, and the* saturated mapping graph*, denoted by $G_{\mathcal{E}}^{\mathcal{M}^{\mathcal{R},O}}$, is the RDF graph induced by $\mathcal{M}^{\mathcal{R},O}$ and an extent $\mathcal{E}$.*

Example 4. *We consider the mapping $m_p$ (Example 1) whose head is the triple $(professor, \text{:teaches}, course)$, the RDF entailment rule* `rdfs2` *(about $\hookleftarrow_d$) and the ontology $O_{\text{ex}}$. To saturate $m_p$ w.r.t.* {`rdfs2`} *and $O_{\text{ex}}$, we map body(*`rdfs2`*) to body($head(m_p)$)$\cup O_{\text{ex}}$ using the following homomorphism:*

$$\varphi' = \{\mathtt{p} \mapsto \text{:teaches}, \mathtt{o} \mapsto \text{:Professor}, \mathtt{s}_1 \mapsto professor, \mathtt{o}_1 \mapsto course\}.$$

*The head of the saturated mapping $m_p^{\{\texttt{rdfs2}\},O_{\text{ex}}}$ has triples $(professor, \text{:teaches}, course)$ and $(professor, \tau, \text{:Professor})$. Hence, $m_p^{\{\texttt{rdfs2}\},O_{\text{ex}}}$ will also populate the class* :Professor *with each value taken by the variable $professor$ in $m_p$.*

The second idea is to see mappings as views in a integration system following the local-as-view approach (recall Section 2.4), which we formally define below.

Definition 18 (Views defined by mappings). *Given an RDF mapping $m = q_1 \rightsquigarrow q_2$, the* view defined by *$m$, denoted by $V_m$, is:*

$$V_m(\bar{x}) \leftarrow body(q_2).$$

*Let $\mathcal{M}$ be a set of RDF mappings. The* set of views defined by *$\mathcal{M}$, denoted by $\mathcal{V}_{\mathcal{M}}$, is:*

$$\mathcal{V}_{\mathcal{M}} = \{V_m \mid m \in \mathcal{M}\}.$$

*Furthermore, to any pair $(\mathcal{M}, \mathcal{E})$, where $\mathcal{E}$ is an extent of $\mathcal{M}$, is assigned the pair $(\mathcal{V}_{\mathcal{M}}, \mathcal{E})$, such that, for all $m \in \mathcal{M}$,* ext$(m)$ *is the extension of $V_m$.*

The following proposition shows that this translation from mappings to views preserves the certain answers to queries.

Property 1. *Let $\mathcal{M}$ be a set of RDF mappings, $\mathcal{E}$ be an extent of $\mathcal{M}$, and $\mathcal{V}_{\mathcal{M}}$ be the set of views defined by $\mathcal{M}$. For any BGPQ $q$, the certain answer set of $q$ on the RDF system (without entailment rules) $\langle \emptyset, \mathcal{M}, \mathcal{E} \rangle$ is equal to the certain answer set of $q$ on $(\mathcal{V}_{\mathcal{M}}, \mathcal{E})$.*

We are now able to outline our method. We assume given an O-system $S = \langle O, \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$. As explained in Section 3.2, we see $S$ as an RDF system $\langle \mathcal{R}, \mathcal{M} \cup \mathcal{M}_O^{\text{STD}}, \mathcal{E} \cup \mathcal{E}_O \rangle$. The different steps of our method are as follows:

1. (Offline step) Compute $\mathcal{M}^{\mathcal{R},O}$, the saturation of $\mathcal{M}$.
2. To compute the certain answer set of a BGPQ $q$ against $S$, proceed as follows:
   (a) Rewrite $q$ into a maximally contained rewriting $q_r$ using the views defined by $\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}$
   (b) Compute the answers to $q_r$ against the extent $\mathcal{E} \cup \mathcal{E}_O$.

# 5. PROBLEM RESTRICTIONS

In this section, we state the specific hypotheses under which our approach holds. We first characterize our ontologies (Section 5.1), then our entailment rules (Section 5.2), mappings (Section 5.3), and finally the resulting restricted $O$-systems (Section 5.4).

## 5.1 First-Order Ontology

The following definition of first-order ontology restricts the form of the RDFS statements that can be used in an ontology. Essentially, it is not allowed to express constraints on the RDFS built-in properties and the RDF type property, nor to use anonymous subjects and objects, i.e., unknown classes and properties.

Definition 19 (FO Ontology). *A* first-order ontology *$O$ is an RDFS ontology whose triples do not contain any of the following values:*

- *a blank node,*
- *the property type, $\tau$ (see Example 8),*
- *an RDFS IRI as subject,*
- *an RDFS IRI as object.*

This setting allows one to express schema statements similar to TBox statements in description logics. For example, the triple $(C_1, \prec_{sc}, C_2)$ in FO ontology states exactly the DL statement $C_1 \sqsubseteq C_2$, which corresponds to the FO formula $\forall x\ (C_1(x) \rightarrow C_2(x))$.

## 5.2 Restricted Rules

We consider entailment rules that comply with some restrictions. This yields two kinds of entailment rules, namely ontological rules and instance rules, which respectively allow to infer knowledge about the ontology and about individuals.

The goal of the instance rule restrictions is to allow using RDFS triples when inferring facts about individuals, which

is one of the interests of RDF, while ensuring the completeness of saturation based on mapping heads[2].

DEFINITION 20 (RESTRICTED RULES). *We call* restricted rule *an RDF entailment rule $r$ which is either an* ontological rule *or an* instance rule *as defined below:*

1. *(**Ontological rule**) body($r$) and head($r$) contain solely RDFS triples such that* $\mathrm{Bl}(head(r)) \subseteq \mathrm{Bl}(body(r))$
2. *(**Instance rule**)* $body(r) = \{t_r\} \cup \mathrm{body}_O(r)$*, where*
   - *(a)* $\mathrm{body}_O(r)$ *is a (possibly empty) set of RDFS triples*
   - *(b) $t_r$ is of one of the following forms:*
     - *i.* $(x, \mathtt{p}, y)$ *where* $x, y \in \mathscr{B} \setminus \mathrm{Bl}(\mathrm{body}_O(r))$, $x \neq y$ *and* $\mathtt{p} \in (\mathscr{I} \setminus \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r, \tau\}) \cup \mathrm{Bl}(\mathrm{body}_O(r))$,
     - *ii.* $(x, \tau, z)$ *where* $x \in \mathscr{B} \setminus \mathrm{Bl}(\mathrm{body}_O(r))$, $z \in \mathscr{I} \cup \mathrm{Bl}(\mathrm{body}_O(r))$

   *and head($r$) contains solely* $(\mathtt{s}, \mathtt{p}, \mathtt{o})$ *triples such that:*
   - *(c)* $\mathtt{p} \in \mathscr{I} \setminus \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r\}$ *or* $\mathtt{p} \in \mathrm{Bl}(\mathrm{body}_O(r))$,
   - *(d) if* $\mathtt{p} = \tau$*, then* $\mathtt{o} \in \mathscr{I}$ *or* $\mathtt{o} \in \mathrm{Bl}(\mathrm{body}_O(r))$.

We first point out below that the standard RDF entailment rules from Table 2 are specific restricted rules.

EXAMPLE 5. *Consider Table 2. Rules* `rdfs5`*,* `rdfs 11`*,* `ext1`*,* `ext2`*,* `ext2`*,* `ext3` *and* `ext4` *are ontological rules. Indeed, their body and head are composed of RDFS statements, e.g.,* `rdfs5`*:* $(\mathtt{p}_1, \prec_{sp}, \mathtt{p}_2), (\mathtt{p}_2, \prec_{sp}, \mathtt{p}_3) \rightarrow (\mathtt{p}_1, \prec_{sp}, \mathtt{p}_3)$*. All the other rules are instance rules, whose body is composed of an RDFS triple and a triple of the form $t_r$, and the head has a single triple. In Rule* `rdfs2` *defined as follows* $(\mathtt{s}, \hookleftarrow_d, \mathtt{o}), (\mathtt{s}_1, \mathtt{p}, \mathtt{o}_1) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$*, $t_r$ fulfills Restriction 2(b)i, while the head fulfills Restriction 2d. The same holds for Rule* `rdfs3`*. In Rule* `rdfs9` $= (\mathtt{s}, \prec_{sc}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$*, $t_r$ complies with 2(b)ii and the head with 2d. Finally, in Rule* `rdfs7`*:* $(\mathtt{p}_1, \prec_{sp}, \mathtt{p}_2), (\mathtt{s}, \mathtt{p}_1, \mathtt{o}) \rightarrow (\mathtt{s}, \mathtt{p}_2, \mathtt{o})$*, $t_r$ complies with 2(b)i and the head with 2c.*

The syntax of restricted rules also allows for user-specific rules **beyond standard RDF entailment rules**, see for instance the rule $r_{\text{ex}} = (s, \text{:registeredFor}, c) \rightarrow (z, \text{:teaches}, c)$ from the running example, stating that if some student is registered for some course, then this course is taught by some teacher.

Since entailment rules will be applied to saturate the mapping heads, the termination of saturation is a crucial requirement. Obviously, this requirement is fulfilled by ontological rules. Indeed, rule heads do not introduce new blank nodes (i.e., for any rule $r$, $\mathrm{Bl}(\mathrm{head}(r)) \subseteq \mathrm{Bl}(\mathrm{body}(r))$). However, termination is not ensured for instance rules: e.g., a rule of the form $(x, \mathtt{p}, y) \rightarrow (y, \mathtt{p}, z)$ (intuitively, for all $x$ and $y$, if $x$ is related to $y$ by $\mathtt{p}$, there exists $z$ such that $y$ is related to $z$ by $\mathtt{p}$) leads to infinite saturation, as each rule application produces a new individual, which leads to a new rule application. However, we prefer not to further restrict instance rules to enforce termination, because of the variety of candidate syntactic restrictions. Indeed, many acyclicity conditions for sets of rules have been defined in the literature about first-order logical rules (e.g., tuple-generating dependencies or existential rules) and can be imported in our setting. Hence, in the following, we will silently assume that the considered set of restricted rules ensures the termination of saturation, as is the case, for instance, of the set of rules $\mathcal{R}_{\text{RDFS}} \cup \{r_{\text{ex}}\}$.

[2]Essentially, the restrictions seek to ensure that, given any extensions for the mappings, the graph obtained by the saturated mappings is equal to the saturation of the graph obtained by the initial mappings, see Theorem 2.

In order to comment on the behavior of restricted rules, let us distinguish RDFS triples that can be contained in a FO ontology, from *instance triples* $(\mathtt{s}, \mathtt{p}, \mathtt{o})$ such that (1) $\mathtt{p} \notin \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r\}$, and (2) if $\mathtt{p} = \tau$, then $\mathtt{o} \in \mathscr{I}$.

We first point out that ontological rules can only be applied on triples of an FO ontology, and that any restricted rule that can be applied on an FO ontology is an ontological rule (next Property 2); second, the saturation of an FO ontology by restricted rules (hence, necessarily ontological rules) remains an FO ontology (next Property 3); third, given a graph whose set of RDFS triples is an FO ontology, all RDFS triples that can be brought by application of restricted rules come from ontological rules (next Property 4). Finally, restricted rules behave as expected when they are applied to any RDF graph $G$ composed of an FO ontology and instance triples: the ontological rules compute exactly the saturation of the ontological part of the graph (next Property 5), while the instance rules add triples about the individuals (possibly using ontological triples as well, be they initially present in the graph or inferred by the ontological rules).

PROPERTY 2. *Let $r$ be a restricted rule and $O$ be an FO ontology, if $O \models^{\varphi} body(r)$, then $r$ is an ontological rule, i.e., it fulfills Restriction 1.*

PROPERTY 3. *Let $O$ be an FO ontology and $\mathcal{R}$ be a set of restricted rules, $O^{\mathcal{R}}$ is also an FO ontology.*

PROPERTY 4. *Let $r$ be a restricted rule and $G$ be an RDF graph whose set of RDFS triples is an FO ontology. If the direct entailment of $G$ by $\{r\}$ (denoted $C_{G,\{r\}}$ in Section 2.2) contains an RDFS triple, then $r$ is an ontological rule.*

PROPERTY 5. *Let $O$ be an FO ontology, $\mathcal{R}$ be a set of restricted rules and $G$ be an RDF graph such that* $\mathrm{RDFS}(G) = O$*, it holds that:*

$$\mathrm{RDFS}(G^{\mathcal{R}}) = O^{\mathcal{R}}$$

This behavior is schematized by Figure 3: the above properties of ontological rules are summarized by the loop on the FO ontology. The body of an instance rule $r$ is composed of $t_r$ a triple and $\mathrm{body}_O(r)$ a set of RDFS triples. When $r$ is applied on $G$, the triple $t_r$ (resp. $\mathrm{body}_O(r)$) is necessarily mapped to an instance triple (resp. FO ontology triples) of $G$, furthermore the triples produced by this application are necessarily instance triples.

## 5.3 Instance Mappings

In line with the distinction between ontological and instance triples, we distinguish between standard mappings associated with an ontology and instance mappings associated with data sources.

DEFINITION 21 (INSTANCE MAPPING). *An* instance mapping *is an RDF mapping $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$, such that the body of $q_2$ contains only triples of one of the following forms:*

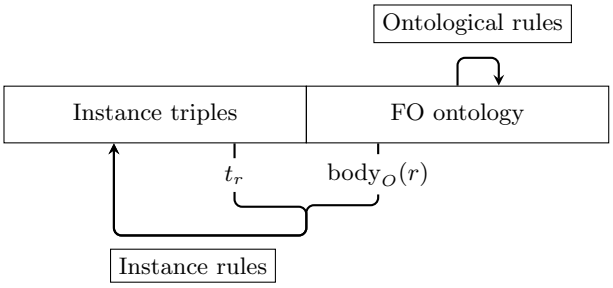


**Figure 3: Restricted rule entailments**

- $(\mathtt{s}, \mathtt{p}, \mathtt{o})$ *where* $p \in \mathscr{I} \setminus \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r, \tau\}$,
- $(\mathtt{s}, \tau, C)$ *where* $C \in \mathscr{I}$.

The triples in the head of an instance mapping are exactly instance triples where variables replace blank nodes, hence, when these are instantiated, they yield instance triples. The next example shows why the second condition is set.

Example 6. *Consider the RDF mapping $m$ defined by $q_1(x, y) \rightsquigarrow (x, \tau, y)$ and its extension $\text{ext}(m) = \{(s, C_1)\}$, as well as the RDF standard entailment rule* `rdfs9` *defined by $(s, \prec_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o)$ and the FO ontology $O = \{(C_1, \prec_{sc}, C_2)\}$. Given $\mathcal{M} = \{m\}$, $\mathcal{E} = \{\text{ext}(m)\}$ and $\mathcal{R} = \{\mathtt{rdfs9}\}$, the saturation of the induced RDF graph is:*

$$\left(G^{\mathcal{M}}_{\mathcal{E}} \cup O\right)^{\mathcal{R}} = \{(s, \tau, C_1), (s, \tau, C_2), (C_1, \prec_{sc}, C_2)\}.$$

*However, there is no homomorphism from $\text{body}(r_{\prec_{sc}})$ to $(x, \tau, y), (C_1, \prec_{sc}, C_2)$, hence $(\text{head}(m))^{\mathcal{R},O} = \text{head}(m)$. It follows that the triple $(s, \tau, C_2)$ is missing in $(G^{\mathcal{M}^{\mathcal{R},O}}_{\mathcal{E}} \cup O)$. For this reason, $m$ is not considered as an instance mapping.*

The next property partially explains why no information is lost when we locally saturate the heads of mappings instead of saturating the graph induced by them. See Figure 4: for an instance rule $r$, if $\text{body}(r)$ is mapped by a homomorphism $\varphi$ to an FO ontology $O$ and a triple $v(t)$ from the head of an instance mapping instantiated by a homomorphism $v$, then (1) there is a homomorphism $\varphi'$ that applies $r$ to $\{t\} \cup O$, and (2) the composition $\varphi' \circ v$ is exactly $\varphi$.

Property 6. *Let $O$ be an FO ontology, $t$ a triple in the head of an instance mapping, and $v$ a homomorphism from $\text{Var(t)} \rightarrow \mathscr{B} \cup \mathscr{I}$. For any restricted rule $r$ (necessarily an instance rule), if $\{v(t)\} \cup O \models^{\varphi} \text{body}(r)$ then there exists a homomorphism $\varphi'$ such that $\{t\} \cup O \models^{\varphi'} \text{body}(r)$ and $\varphi(\text{body}(r)) = v(\varphi'(\text{body}(r)))$.*

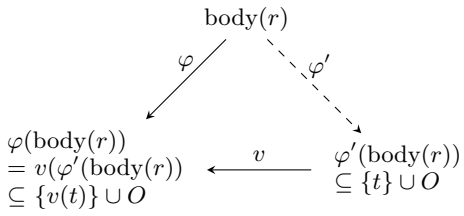


**Figure 4: Illustration of Property 6.**

The next property expresses that, when a restricted rule is applied to the head of an instance mapping, the added triples keep the property of being an instance mapping.

Property 7. *Let $O$ be an FO ontology, $t$ a triple of the head of an instance mapping $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$ and $r$ an instance rule such that $\{t\} \cup O \models^{\varphi'} \text{body}(r)$. Then the mapping $m' = q_1(\bar{x}) \rightsquigarrow q'_2(\bar{x})$ with $q'_2(\bar{x}) = \text{body}(q_2) \cup \varphi'(\text{head}(r))^{safe}$ is an instance mapping.*

Example 7. *Reusing the running example, we illustrate the two previous properties. The instance mapping $m_p$ generates the triple* (:Fabian, :teaches, :SemWeb) *by the homomorphism $v = \{professor \mapsto$ :Fabian, $course \mapsto$ :SemWeb$\}$ applied on $t = (professor,$ :teaches, $course)$ in the head of $m_p$. The instance rule* `rdfs9` *can be used on the generated triple $v(t)$ and FO ontology $O_{\text{ex}}$. Indeed, there is a homomorphism $\varphi$ from $\text{body}(r)$ to $v(t) \cup O_{\text{ex}}$ defined as:*

$$\begin{aligned} \varphi \quad = \quad & \{\mathtt{p} \mapsto \text{:teaches}, \mathtt{o} \mapsto \text{:Professor}, \\ & \mathtt{s}_1 \mapsto \text{:Fabian}, \mathtt{o}_1 \mapsto \text{:SemWeb}\}. \end{aligned}$$

*As stated by Property 6, there exists a homomorphism from $\text{body}(r)$ to $\{t\} \cup O_{\text{ex}}$, which is $\varphi'$ defined in Example 4. We check that $v(\varphi'(\text{body}(r))) = \varphi(\text{body}(r))$, which ensures that applying* `rdfs2` *on $m_p$ head with $\varphi'$ and then instantiating its saturated head with $v$ returns the same triples as applying* `rdfs2` *with $\varphi$ on $m_p$ head with $v$. Moreover, Property 7 ensures that the saturated mapping $m_p^{\{\mathtt{rdfs2}, O_{\text{ex}}\}}$ is an instance mapping like $m_p$.*

## 5.4 Restricted $O$-system

We have now defined suitable restrictions for each component of an $O$-system, which yield a restricted $O$-system. We precise that only previous properties on FO ontology, restricted rules and instance mappings are needed for the rest of the paper. It means that by using an other syntax of this elements satisfying the same properties, following results will still hold.

Definition 22 (Restricted $O$-system). *We say that an $O$-system $S = \langle O, \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$ is a* restricted $O$-system *if $O$ is an FO ontology, $\mathcal{R}$ a set of restricted rules and $\mathcal{M}$ a set of instance mappings.*

As defined in Section 3.2, the standard mappings associated with an ontology $O$ allow one to integrate the ontological statements of $O$ into an RDF graph, as follows:

Property 8. *Given an FO ontology $O$ and a set $\mathcal{R}$ of restricted rules, it holds that:*

$$G^{\mathcal{M}^{STD}_O}_{\mathcal{E}_O} = O^{\mathcal{R}}$$

Futhermore, no other RDFS triples are created by instance mappings and applications of instance rules, hence:

Property 9. *For any restricted $O$-system $S = \langle O, \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$, the set of RDFS triples in $(G^{\mathcal{M} \cup \mathcal{M}^{STD}_O}_{\mathcal{E} \cup \mathcal{E}_O})^{\mathcal{R}}$ is exactly $O^{\mathcal{R}}$.*

# 6. CORRECTNESS OF THE METHOD

In this section, we present the main arguments that prove the correctness of our method, and refer the reader to the appendix for detailed proofs.

Assume first that we adopt a classical materialization approach: starting from the extent $\mathcal{E} \cup \mathcal{E}_O$, we trigger the mappings $\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}$, then saturate the obtained graph with the entailment rules $\mathcal{R}$, and finally get the graph $\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}}$, on which we can ask BGP queries and obtain a complete certain answer set.

Now, instead of saturating the graph produced by the mappings, we proceed as follows: (1) we saturate the mappings (actually their head) with the entailment rules, then (2) we trigger the mappings. We thus obtain the graph $G_{\mathcal{E}}^{\mathcal{M}^{\mathcal{R},O}} \cup G_{\mathcal{E}_O}^{\mathcal{M}_O^{\text{STD}}}$. The next theorem states that the graphs obtained by the two ways of doing are equal (which holds up to bijective renaming of blank nodes).

THEOREM 2. *Given an FO ontology $O$, a set $\mathcal{R}$ of restricted rules, a set $\mathcal{M}$ of instance mappings, it holds that:*

$$\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{STD}}\right)^{\mathcal{R}} = G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{STD}}$$

Of course, this equality does not hold for general O-systems. In Section 5, we have illustrated by examples the role of the main restrictions we enforce and hightlighted some key properties ensured by these restrictions. An important characteristic of restricted O-systems is the distinction between standard and instance mappings, and similarly between ontological and instance entailment rules. This allows one to consider two induced graphs, whose union yields $G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$:

- $G_{\mathcal{E}_O}^{\mathcal{M}_O^{\text{STD}}}$, which is equal to the saturated ontology $O^{\mathcal{R}}$ (Property 8).
- $G_{\mathcal{E}}^{\mathcal{M}^{\mathcal{R},O}}$, which materializes exactly the instances triples of the saturated graph $\left(G_{\mathcal{E}}^{\mathcal{M}} \cup O\right)^{\mathcal{R}}$. This equality also relies on the form of the restricted instance rules, which ensures that every application of an instance rule involved in the saturation of the graph produced the mappings can be similarly performed on a mapping head (Property 6); in particular, no application of an instance rule requires instance triples coming from two different mappings.

Finally, instead of computing the answers to a query $q$ against the materialized graph $\left(G_{\mathcal{E}}^{\mathcal{M}} \cup O\right)^{\mathcal{R}}$, we rewrite $q$ into a query $q_r$, such that the answers to $q$ against the system are the answers to $q_r$ against $\mathcal{E} \cup \mathcal{E}_O$:

DEFINITION 23 (REWRITING). *Given an FO ontology $O$, a set $\mathcal{R}$ of restricted rules, and a set $\mathcal{M}$ of instance mappings, a* rewriting *$q_r$ of a BGPQ $q$ w.r.t. $O, \mathcal{R}, \mathcal{M}$ is a query such that, for any extent $\mathcal{E}$ of $\mathcal{M}$, the answer set of $q_r$ against $\mathcal{E} \cup \mathcal{E}_O$ is* $\text{cert}(q, S = \langle O, \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle)$.

We have already shown how mappings can be seen as views (Definition 18 and Property 1), more precisely the set of mappings $\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}$ is seen as the following set of views:

$$\mathcal{V}_{\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}} = \{V_m(\bar{x}) \leftarrow \text{body}(q_2) \mid m \in \mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}, \\ m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})\}$$

with, for each $m \in \mathcal{M}$, the extension of $V_m$ being set to $\text{ext}(m)$.

Based on this translation, one obtains a rewriting of $q$ w.r.t. $O, \mathcal{R}, \mathcal{M}$ by computing a maximally contained rewriting of $q$ using the views $\mathcal{V}_{\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$, as expressed by the following theorem:

THEOREM 3. *Given an FO ontology $O$, a set $\mathcal{R}$ of restricted rules, a set $\mathcal{M}$ of instance mappings, and a BGPQ $q$, $q_r$ a maximally contained rewriting of $q$ using the views $\mathcal{V}_{\mathcal{M}^{\mathcal{R},O} \cup \mathcal{M}_O^{STD}}$ w.r.t. UCQs is a rewriting of $q$ w.r.t. $O, \mathcal{R}, \mathcal{M}$.*

Grouping together Theorems 2 and 3, we obtain the wanted result: the certain answer set to a BGPQ $q$ against a restricted O-system $\langle O, \mathcal{R}, \mathcal{M}, \mathcal{E} \rangle$, is exactly the answer set of $q_r$ on the extension $\mathcal{E} \cup \mathcal{E}_O$.

# 7. RELATED WORK

As we have explained in the Introduction, our work pursues a data integration goal [35, 32], that is: providing access to a set of data sources under a single unified schema.

Ontologies have been used to integrate relational or heterogeneous data sources in mediators [35] following the local-as-view approach, using the CLASSIC description logic [30], CARIN which combines Datalog with some description logics [19, 21] or the DL-lite$_{\mathcal{R}}$ description logic [3] underpinning the OWL2QL dialect of the W3C's OWL2 semantic web standard; in particular, [3] adopts the reformulate-then-rewrite query answering approach sketched in Section 4.1. However, none of the above approaches consider graph data models, as we do with RDF.

We follow the observation, at the origin of the semistructured data management area [16], that graphs are a very convenient paradigm for integrating data from heterogeneous data sources. Adding semantics at the integration level in order to enrich its exploitation was proposed early on, for SGML [13] and then soon after for RDF [7, 8]; data is considered to be represented and stored in a flexible object-oriented model, thus no mappings are used. Reconciliation between the source and the integrated schemas is performed semi-automatically, trying to determine the best correspondences based on the available ontologies, and asking users to solve unclear situations. In contrast, our proposal considers heterogeneous sources, and, following the OBDA approach, relies on mappings to connect (in a loose coupling) the integration ontology and the source schemas.

XML trees have also been used as the integration format in systems integrating heterogeneous data sources following the local-as-view approach [31, 14, 6]. *Virtual* views were specified in a pivot relational model (enriched with integrity constraints) to describe how the content of each data source contributes to the global schema. View-based rewriting against this relational model lead to queries over the virtual views, which were then translated back into queries that can be evaluated on each individual source. Ontological knowledge was not exploited in here nor in classical relational view-based integration [17, 22], whereas we include them in our framework to enrich the set of results that a user may get out of the system, by making available the application knowledge they encapsulate.

Our work follows the OBDA vision [18, 28, 27, 12]. Compared to these works, our novelty is (*i*) to extend the typical relational setting to heterogeneous data sources (which is rather simple thanks to mappings), (*ii*) relying on RDF as

the integration model, thus in particular allowing applications to query the integrated data and the ontology, and (*iii*) proposing a novel approach for query answering, different both from materialization and reformulation, which avoids their drawbacks and is capable of computing certain query answers, under the restrictions we detailed above.

## Acknowledgements

Part of this work is supported by the Inria Project Lab grant iCoda, a collaborative project between Inria and several major French media, aiming at building a heterogeneous data integration platform for ontology-driven data journalism.

# 8. REFERENCES

[1] RDF 1.1 Concepts and Abstract Syntax.

[2] RDF 1.1 Semantics.

[3] N. Abdallah, F. Goasdoué, and M. Rousset. DL-LITER in the light of propositional logic for decentralized data management. In *IJCAI*, 2009.

[4] S. Abiteboul and O. M. Duschka. Complexity of answering queries using materialized views. ACM Press, 1998.

[5] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[6] B. Amann, C. Beeri, I. Fundulaki, and M. Scholl. Querying XML Sources Using an Ontology-Based Mediator. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 429–448. Springer Berlin Heidelberg, 2002.

[7] B. Amann and I. Fundulaki. Integrating ontologies and thesauri to build RDF schemas. In *ECDL*, 1999.

[8] B. Amann, I. Fundulaki, and M. Scholl. Integrating ontologies and thesauri for RDF schema creation and metadata querying. *Int. J. on Digital Libraries*, 3(3), 2000.

[9] D. Bursztyn, F. Goasdoué, and I. Manolescu. Optimizing reformulation-based query answering in RDF. In *EDBT*, 2015.

[10] D. Bursztyn, F. Goasdoué, and I. Manolescu. Teaching an RDBMS about ontological constraints. *PVLDB*, 9(12), 2016.

[11] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3), 2017.

[12] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, and M. Ruzzi. Using OWL in Data Integration. In *Semantic Web Information Management*, pages 397–424. Springer, 2010.

[13] V. Christophides, M. Doerr, and I. Fundulaki. A semantic network approach to semi-structured documents repositories. In *ECDL*, 1997.

[14] A. Deutsch and V. Tannen. MARS: A System for Publishing XML from Mixed and Redundant Storage. In *VLDB*, 2003.

[15] S. El Hassad, F. Goasdoué, and H. Jaudoin. Learning commonalities in SPARQL. In *ISWC*, 2017.

[16] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *J. Intell. Inf. Syst.*, 8(2), 1997.

[17] M. R. Genesereth, A. M. Keller, and O. M. Duschka. Infomaster: An information integration system. In *SIGMOD*, 1997.

[18] M. Giese, A. Soylu, G. Vega-Gorgojo, A. Waaler, P. Haase, E. Jiménez-Ruiz, D. Lanti, M. Rezk, G. Xiao, Ö. L. Özçep, and R. Rosati. Optique: Zooming in on big data. *IEEE Computer*, 48(3), 2015.

[19] F. Goasdoué, V. Lattès, and M. Rousset. The use of CARIN language and algorithms for information integration: The PICSEL system. *Int. J. Cooperative Inf. Syst.*, 9(4), 2000.

[20] F. Goasdoué, I. Manolescu, and A. Roatis. Efficient query answering against dynamic RDF databases. In *EDBT*, 2013.

[21] F. Goasdoué and M. Rousset. Answering queries using views: A KRDB perspective for the semantic web. *ACM Trans. Internet Techn.*, 4(3), 2004.

[22] A. Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4), Dec. 2001.

[23] D. Hovland, R. Kontchakov, M. G. Skjæveland, A. Waaler, and M. Zakharyaschev. Ontology-based data access to Slegge. In *ISWC*, 2017.

[24] M. Jarke. *Fundamentals of data warehouses, 2nd Edition*. Springer, 2003.

[25] E. Kharlamov, D. Hovland, M. G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, D. Lanti, M. Rezk, D. Zheleznyakov, M. Giese, H. Lie, Y. E. Ioannidis, Y. Kotidis, M. Koubarakis, and A. Waaler. Ontology based data access in Statoil. *J. Web Sem.*, 44, 2017.

[26] M. König, M. Leclère, and M. Mugnier. Query rewriting for existential rules with compiled preorder. In *IJCAI*, 2015.

[27] D. Lanti, G. Xiao, and D. Calvanese. Cost-driven ontology-based data access. In *ISWC*, 2017.

[28] D. Lembo, J. Mora, R. Rosati, D. F. Savo, and E. Thorstensen. Mapping analysis in ontology-based data access: Algorithms and complexity. In *ISWC*, 2015.

[29] M. Lenzerini. Ontology-based data management. In *CIKM*, 2011.

[30] A. Y. Levy, D. Srivastava, and T. Kirk. Data model and query evaluation in global information systems. *J. Intell. Inf. Syst.*, 5(2), 1995.

[31] I. Manolescu, D. Florescu, and D. Kossmann. Answering XML queries on heterogeneous data sources. In *VLDB*, 2001.

[32] M. T. Özsu and P. Valduriez. *Distributed and Parallel Database Systems (3rd. ed.)*. Springer, 2011.

[33] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10, 2008.

[34] M. Thomazo. Compact rewritings for existential rules. In *IJCAI*, 2013.

[35] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3), 1992.

# Appendix

## Proof of Property 1

PROOF. Let $\bar{t}$ be a certain answer of $q$ on $\langle \emptyset, \mathcal{M}, \mathcal{E} \rangle$. By definition, there exists mappings $m_1 = q_{1,1} \rightsquigarrow q_{2,1}, \ldots, m_n = q_{1,n} \rightsquigarrow q_{2,n}$ of $\mathcal{M}$ and tuples $\bar{t}_1 \in \text{ext}(m_1), \ldots, \bar{t}_n \in \text{ext}(m_n)$ such that:

$$\text{body}(q_{2,1})(\bar{t}_1)^{\text{safe}}, \ldots, \text{body}(q_{2,n})(\bar{t}_n)^{\text{safe}} \models \text{body}(q)(\bar{t})$$

Since the safe operation just replaces variables by fresh blank nodes, we have for each $1 \leq i \leq n$, we have $\text{body}(V_{m_i})(\bar{t}_i) = \text{body}(q_{2,i})(\bar{t}_i) \models \text{body}(q_{2,i})(\bar{t}_i)^{\text{safe}}$. Let $D$ be an instance of triples w.r.t. $(\mathcal{V}_\mathcal{M}, \mathcal{E})$, by definition for each $1 \leq i \leq n$, we have $\bar{t}_i \in q_{2,i}(D)$, i.e., $D \models \text{body}(V_{m_i})(\bar{t}_i)$. So finally, for each $D$ instance of triples w.r.t. $(\mathcal{V}_\mathcal{M}, \mathcal{E})$, we have $D \models \text{body}(q)(\bar{t})$. And $\bar{t}$ is a certain answer of $q$ on $(\mathcal{V}_\mathcal{M}, \mathcal{E})$.

Let's make a remark about maximally contained rewriting. Let $q_r$ be a maximally contained rewriting of $q$ using $\mathcal{V}_\mathcal{M}$ w.r.t. UCQ on views language and let $C_r$ be one conjunction component of $q_r$ with $C_r = V_{m_1}, \ldots, V_{m_n}$. Since we know that $q_r$ is contained in $q$, we can deduce that $\text{body}(V_{m_1}), \ldots, \text{body}(V_{m_n}) \models \text{body}(q)$.

Let $\bar{t}$ be a certain answer of $q$ on $(\mathcal{V}_\mathcal{M}, \mathcal{E})$. Since $q$ is a conjunctive query of triples and $\mathcal{V}_\mathcal{M}$ contains conjunctive views, we know that there exists $q_r$ a maximally contained rewriting of $q$ using $\mathcal{V}_\mathcal{M}$ w.r.t. union of conjunctive queries on $\mathcal{V}_\mathcal{M}$. Moreover, applying Theorem 1 on $q_r$ infers that $\bar{t}$ is an answer of $q_r$ on $\mathcal{E}$. So, there exists $C_r$ a conjunction component of $q_r$ with $C_r = V_{m_1}, \ldots, V_{m_n}$ and $\bar{t}_1 \in \text{ext}(m_1), \ldots, \bar{t}_n \in \text{ext}(m_n)$, such that:

$$\text{body}(V_{m_1})(\bar{t}_1), \ldots, \text{body}(V_{m_n})(\bar{t}_n) \models \text{body}(q)(\bar{t}).$$

By considering existential variables as blank nodes, we have for each $1 \leq i \leq n$, we have $\text{body}(q_{2,i})(\bar{t}_i)^{\text{safe}} \models \text{body}(V_{m_i})(\bar{t}_i)$ where $q_{2,i}$ is such that $m_i = q_{1,i} \rightsquigarrow q_{2,i}$. So finally, we have $G_\mathcal{E}^\mathcal{M} \models \text{body}(q)(\bar{t})$, i.e., $\bar{t}$ is an answer of $q$ on $G_\mathcal{E}^\mathcal{M}$. Moreover $\bar{t}$ is not composed by any blank node, so $\bar{t}$ is a certain answer on $S$. □

## Examples

The following example illustrates why the property $\tau$ is not allowed in an FO ontology, here in an object position.

EXAMPLE 8. *Consider an RDF mapping $m = q_1(x, y) \rightsquigarrow (x, \tau', y)$, its extension $\text{ext}(m) = \{(a, C_1)\}$, the RDF entailment rules* `rdfs9` *defined as $(s, \prec_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o)$ and* `rdfs7` *defined as $(p_1, \prec_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o)$, and the ontology $O = \{(C_1, \prec_{sc}, C_2), (\tau', \prec_{sp}, \tau)\}$. Let $\mathcal{M} = \{m\}$, $\mathcal{E} = \{\text{ext}(m)\}$ and $\mathcal{R} = \{$*`rdfs9`*, *`rdfs7`*$\}$; the saturation of the induced RDF graph is:*

$$\left(G_\mathcal{E}^\mathcal{M} \cup O\right)^\mathcal{R} = \{(a, \tau', C_1), (a, \tau, C_1), (a, \tau, C_2), (C_1, \prec_{sc}, C_2), (\tau', \prec_{sp}, \tau)\}$$

*The saturated mapping $m^{\mathcal{R},O}$ is $q_1(x, y) \rightsquigarrow (x, \tau', y), (x, \tau, y)$, so $(G_\mathcal{E}^{\mathcal{M}^{\mathcal{R},O}} \cup O)$ does not contain, hence misses, the triple $(a, \tau, C_2)$.*

The following example is relative to the definition of restricted rules and illustrates the importance of the condition $x \neq y$ in the triple $t_r$.

EXAMPLE 9. *Assume the rule $r = (x, p, x) \rightarrow (x, q, x)$ would be allowed, and let the mapping $m = q_1(x, y) \rightsquigarrow (x, p, y)$ and its extension $\text{ext}(m) = \{(a, a)\}$. Let $\mathcal{M} = \{m\}$, $\mathcal{E} = \{\text{ext}(m)\}$ and $\mathcal{R} = \{$*`rdfs9`*$\}$. Then the saturation of the induced RDF graph is:*

$$\left(G_\mathcal{E}^\mathcal{M}\right)^\mathcal{R} = \{(a, p, a), (a, q, a)\}.$$

*However, there is no homomorphism from body($r$) to $(x, p, y)$, hence the saturation of the mapping $m$ with $\mathcal{R}$ is exactly $m$. Therefore, the triple $(a, q, a)$ is missing in the saturated mapping graph $G_\mathcal{E}^{\mathcal{M}^{\mathcal{R},\{\}}}$. It is the reason why the condition $x \neq y$ is enforced in 2(b)i, hence $r$ is not a restricted rule.*

## Proof of Property 2

PROOF. If we assume that $r$ is an instance rule, then $\text{body}(r) = \{t_r\} \cup \text{body}_O(r)$, and the triple $t_r$ has to be of the form $(x, \mathtt{p}, y)$ with $\mathtt{p} \in \text{Bl}(\text{body}_O(r))$, otherwise $\varphi(t_r)$ could not be an RDFS triple (which would contradict the fact that $O \models^\varphi \text{body}(r)$). Then $\varphi(\mathtt{p})$ occurs as a subject or an object of a triple in $O$, because $\text{body}_O(r)$ does not have any blank node as a property. However, $\varphi(\mathtt{p})$ cannot be an RDFS IRI, because $O$ is an FO ontology. This contradicts the fact that $\varphi(\mathtt{p})$ is an RDFS IRI since $\varphi(t_r) \in O$. We conclude that $r$ is necessarily an ontological rule. □

## Proof of Property 3

PROOF. From Property 2, only ontological rules can be applied on an FO-ontology. We check that the produced RDFS triples comply with the conditions of an FO-ontology. □

## Proof of Property 4

PROOF. From the definition of the restricted rules, we know that if the direct entailment of $G$ by $\{r\}$ contains an RDFS triple, then $r$ is either an ontological rule, or an instance rule with head containing a triple $t = (\mathtt{s}, \mathtt{p}, \mathtt{o})$, case 2c. The latter case is not possible, because necessarily $\mathtt{p} \in \text{Bl}(\text{body}_O(r))$, i.e., $\mathtt{p}$ also occurs as the subject or the object of an RDFS triple $t_b$ in the body of $r$. The application of the rule $r$ maps $t_b$ to an RDFS triple of $G$. Since the set of RDFS triples of $G$ is an FO ontology, $\mathtt{p}$ cannot be mapped to an RDFS IRI, which is absurd. Therefore, $r$ is an ontological rule. □

## Proof of Property 5

PROOF. First, we prove that $O^\mathcal{R} \subseteq \text{RDFS}(G^\mathcal{R})$. Using Property 3, we know that $O^\mathcal{R}$ is an FO ontology, so at least $O^\mathcal{R}$ is a set of RDFS triples. Since $O \subseteq G$, we have $O^\mathcal{R} \subseteq \text{RDFS}(G^\mathcal{R})$.

Second, we prove that $\text{RDFS}(G^\mathcal{R}) \subseteq O^\mathcal{R}$. Let $t$ be a triple in $\text{RDFS}(G^\mathcal{R})$, so either $t$ is an RDFS triple of $G$ (and then $t \in O^\mathcal{R}$) or $t$ is a RDFS triple in entailment of $\mathcal{R}$ on $G$. In the latter case, we prove that $t \in O^\mathcal{R}$ by induction on $(G_i^\mathcal{R})_{i \in \mathbb{N}}$ the saturation sequence of $G^\mathcal{R}$.

By Property 4, if a restricted rules $r \in \mathcal{R}$ applied on $G$ directly derives at least an RDFS triples, $r$ is an ontological rule. And since, the body of an ontological rule is composed of RDFS triples, a such rule $r$ can be apply only on RDFS triples. Moreover, $\text{RDFS}(G) = O$, so each RDFS triple $t$ in $C_{G,r}$ the direct entailment of $r$ on $G$ is actually in $C_{O,r}$. Finally, we can remark that $\text{RDFS}(G_1^\mathcal{R}) \subseteq O^\mathcal{R}$.

For the initialization step of the induction, if $t$ a RDFS triple directly entails by $G$, the preceding remark proves that $t \in O^\mathcal{R}$. And the induction is assured by Property 3 which shows that $O^\mathcal{R}$ is an FO ontology, so $\text{RDFS}(G_1^\mathcal{R})$ is

an FO ontology. We can start the preceding reasoning again replacing $G$ by $G_1^{\mathcal{R}}$ and $O$ by $\text{RDFS}(G_1^{\mathcal{R}})$. □

## Proof of Property 6

PROOF. We define $\text{body}(r)$ as $t_r \wedge \text{body}_O(r)$. Since $\text{body}_O(r)$ is a set of RDFS triples and $t$ (thus also $v(t)$) is not an RDFS triple, then $\varphi(\text{body}_O(r)) \subseteq O$. We now consider the two possible forms of $t_r$.
Case *(i)*: $t_r$ has the form of $(x, \mathtt{p}, y)$, Restriction 2(b)i. As in the proof of Property 2, we know that $\varphi(\mathtt{p})$ is not an RDFS IRI and with the same reasoning, we can show that $\varphi(\mathtt{p}) \neq \tau$. So, the triple $\varphi((x, \mathtt{p}, y))$ is equal to $v(t)$ and the instance mapping triple $t = (x', \mathtt{p}', y')$ with $\mathtt{p}' \in \mathscr{I} \setminus \{\tau\}$. Since we know that $x$ and $y$ are not in $\text{Bl}(\text{body}_O(r))$, we know that $\varphi_{|\text{Bl}(\text{body}_O(r))}(t_r) = (x, \mathtt{p}', y)$. So if we choose $\varphi' = \varphi_{|\text{Bl}(\text{body}_O(r))} \cup \{x \mapsto x', y \mapsto y'\}$, which is indeed a homomorphism because $x \neq y$, then we have $\{t\} \cup O \models^{\varphi'} \text{body}(r)$ and $\varphi(\text{body}(r)) = v(\varphi'(\text{body}(r)))$.
Case *(ii)*: $t_r$ has the form $(x, \tau, z)$, with $x \notin \text{Bl}(\text{body}_O(r))$ and $z \in \mathscr{I} \cup \text{Bl}(\text{body}_O(r))$, Restriction 2(b)ii. Since $\tau$ is not an RDFS property, we know that $\varphi(t_r) = v(t)$. So $v(t)$ is equal to $(a, \tau, C)$ with $a \in \mathscr{B} \cup \mathscr{I}$ and $C \in \mathscr{I}$, and there exists $y \in \mathscr{B} \cup \mathscr{I}$ such that $t = (y, \tau, C)$. We have $\varphi_{|\text{Bl}(\text{body}_O(r))}(t_r) = (x, \tau, C)$, so if $y \in \mathscr{B}$, then $\varphi' = \varphi_{|\text{Bl}(\text{body}_O(r))} \cup \{x \mapsto y\}$ satisfies the wanted property. Otherwise, $y \in \mathscr{I}$ and $\varphi' = \varphi_{|\text{Bl}(\text{body}_O(r))}$ satisfies the wanted property as well, because $v$ is then the identity.

□

## Proof of Property 7

PROOF. Let $\text{body}(r) = \{t_r\} \cup \text{body}_O(r)$. Since $\text{body}_O(r)$ is a set of RDFS triples and $t$ is not an RDFS triple, we have $\varphi'(\text{body}_O(r)) \subseteq O$.

The result is just a consequence of the form of $\text{head}(r)$. Let $u = (\mathtt{s}, \mathtt{p}, \mathtt{o})$ be a triple in $\text{head}(r)$, we check that in each case $\varphi'(u)^{\text{safe}}$ can be a triple of the head of an instance mapping:

- if $\mathtt{p} = \tau$ then $\mathtt{o} \in \mathscr{I} \cup \text{Bl}(\text{body}_O(r))$. So $\varphi'(\mathtt{o})$ is always an IRI, since $\varphi'(\text{body}_O(r)) \subseteq O$ and $O$ is an FO ontology;
- if $\mathtt{p} \in \mathscr{I} \setminus \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r\}$, nothing more is required;
- if $\mathtt{p} \in \text{Bl}(\text{body}_O(r))$, then $\varphi'(\mathtt{p}) \in \mathscr{I} \setminus \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r\}$ and it is OK. Again since $\varphi'(\text{body}_O(r)) \subseteq O$ and $O$ is an FO ontology.

□

## Proof of Property 8

PROOF. By Property 3, $O^{\mathcal{R}}$ is an FO ontology, hence contains only RDFS triples. The proof then directly follows from the equalities:

$$
\begin{aligned}
O^{\mathcal{R}} &= \{(\mathtt{s}, \prec_{sc}, \mathtt{o}) \mid (\mathtt{s}, \prec_{sc}, \mathtt{o}) \in O^{\mathcal{R}}\} \\
&\quad \cup \{(\mathtt{s}, \prec_{sp}, \mathtt{o}) \mid (\mathtt{s}, \prec_{sp}, \mathtt{o}) \in O^{\mathcal{R}}\} \\
&\quad \cup \{(\mathtt{s}, \hookleftarrow_d, \mathtt{o}) \mid (\mathtt{s}, \hookleftarrow_d, \mathtt{o}) \in O^{\mathcal{R}}\} \\
&\quad \cup \{(\mathtt{s}, \hookrightarrow_r, \mathtt{o}) \mid (\mathtt{s}, \hookrightarrow_r, \mathtt{o}) \in O^{\mathcal{R}}\}
\end{aligned}
$$

$$
\begin{aligned}
&= \{(s, \prec_{sc}, o)^{safe} \mid m_{\text{subClassOf}} = q_{\text{subClassOf}}(\mathtt{s}, \mathtt{o}) \rightsquigarrow \\
&\quad (\mathtt{s}, \prec_{sc}, \mathtt{o}), (s, o) \in \text{ext}(m_{\text{subClassOf}}) = q_{\text{subClassOf}}(O, \mathcal{R})\} \\
&\quad \cup \{(s, \prec_{sp}, o)^{safe} \mid m_{\text{subPropertyOf}} = q_{\text{subPropertyOf}}(\mathtt{s}, \mathtt{o}) \rightsquigarrow \\
&\quad (\mathtt{s}, \prec_{sp}, \mathtt{o}), (s, o) \in \text{ext}(m_{\text{subPropertyOf}}) = q_{\text{subPropertyOf}}(O, \mathcal{R})\} \\
&\quad \cup \{(s, \hookleftarrow_d, o)^{safe} \mid m_{\text{domain}} = q_{\text{domain}}(\mathtt{s}, \mathtt{o}) \rightsquigarrow \\
&\quad (\mathtt{s}, \hookleftarrow_d, \mathtt{o}), (s, o) \in \text{ext}(m_{\text{domain}}) = q_{\text{domain}}(O, \mathcal{R})\} \\
&\quad \cup \{(s, \hookrightarrow_r, o)^{safe} \mid m_{\text{range}} = q_{\text{range}}(\mathtt{s}, \mathtt{o}) \rightsquigarrow \\
&\quad (\mathtt{s}, \hookrightarrow_r, \mathtt{o}), (s, o) \in \text{ext}(m_{\text{range}}) = q_{\text{range}}(O, \mathcal{R})\} \\
&= G_{\mathcal{E}_O}^{\mathcal{M}_O^{\text{STD}}}
\end{aligned}
$$

q □

## Corollary 1

COROLLARY 1. *The set of RDFS triples of $G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{STD}}$ is exactly $O^{\mathcal{R}}$.*

PROOF. Since no instance mapping of $\mathcal{M}$ has RDFS triples in its head, the ontology of $G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}$ is included in $G_{\mathcal{E}_O}^{\mathcal{M}_O^{\text{STD}}}$. Moreover, $G_{\mathcal{E}_O}^{\mathcal{M}_O^{\text{STD}}}$ contains only RDFS triples, hence the wanted equality holds. □

## Proof of Property 9

PROOF. By Corollary 1, the set of RDFS triples of $G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}$ is $O^{\mathcal{R}}$. By Property 3, $O^{\mathcal{R}}$ is an FO ontology, so by Property 5, the set of RDFS triples of $(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}})^{\mathcal{R}}$ is $O^{\mathcal{R}}$.

□

Below: proofs from Section 6

## Proof of Theorem 2

To prove the theorem, we will rely on the next definition and some auxilliary lemmas.

DEFINITION 24. *We define the following sequence of mappings:*

$$(\mathcal{M})_i^{\mathcal{R},O} = \left\{ q_1 \rightsquigarrow (q_2)_i^{\mathcal{R},O} \mid q_1 \rightsquigarrow q_2 \in \mathcal{M} \right\}$$

*where* $body((q)_i^{\mathcal{R},O}) = max\{S \subseteq (body(q) \cup O)_i^{\mathcal{R}} \mid \forall T \subseteq S, O \models_{\mathcal{R}} T \Rightarrow body(q) \models_{\mathcal{R}} T\}$.

Intuitively, $(q)_i^{\mathcal{R},O}$ is the saturation of $(\text{body}(q) \cup O)$ at rank $i$ from which triples entailed solely by $O$ are removed, as in Def. 7.

LEMMA 1. *Let be $q$ the head of an instance mapping, $O$ an FO ontology and $\mathcal{R}$ a set of restricted rules, we have:*

$\forall i \in \mathbb{N}$, $body((q)_i^{\mathcal{R},O}) = (body(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((body(q) \cup O)_i^{\mathcal{R}})$.

PROOF. First, let $i$ be a positive integer, we prove that $\text{body}((q)_i^{\mathcal{R},O}) \subseteq (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$. Let $t$ be a triple in $\text{body}((q)_i^{\mathcal{R},O})$, so $t \in (\text{body}(q) \cup O)_i^{\mathcal{R}}$. Hence by definition of $(q)_i^{\mathcal{R},O}$, we have either $O \not\models_{\mathcal{R}} t$ or $\text{body}(q) \models_{\mathcal{R}} t$. We will tackle the both cases separately. We recall that since $q$ is the head of an instance mapping, $\text{body}(q)$ only contains none RDFS triples. Moreover using Property 7, we know that $q^{\mathcal{R}}$ body only contains none

RDFS triples. So if $\text{body}(q) \models_{\mathcal{R}} t$, then $t \in (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$. Inspiring by the proof of Property 5, we can prove the following property: for all $u \in (\text{body}(q) \cup O)_i^{\mathcal{R}}$, if $u \in \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$ then $O \models_{\mathcal{R}} u$. Using the contraposition of this property, we know that if $O \not\models_{\mathcal{R}} t$ then $t \in (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$. So in the both case, $t \in (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$.

Secondly, let $i$ be a positive integer, we prove that $(\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}}) \subseteq \text{body}((q)_i^{\mathcal{R},O})$. Let $t \in (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$ and $S \subset (\text{body}(q) \cup O)_i^{\mathcal{R}}$, which satisfies the property $P(S) = \forall T \subseteq S,\ O \models_{\mathcal{R}} T \Rightarrow \text{body}(q) \models_{\mathcal{R}} T$. We will prove that if $t \notin S$, $S$ is not maximal for the property $P$, i.e., we will prove $P(S \cup \{t\})$. Let $T \subseteq S \cup \{t\}$, if $t \in T$ then $O \not\models_{\mathcal{R}} T$, because $t$ is not an RDFS triples and Property 3, otherwise $T \subseteq S$. In both cases, $O \models_{\mathcal{R}} T \Rightarrow \text{body}(q) \models_{\mathcal{R}} T$ holds, so $P(S \cup \{t\})$ also. Finally, $t \in \text{body}((q)_i^{\mathcal{R},O})$. □

We notice that even if this sequence of mappings is not increasing, it induces a **increasing** sequence of RDF graphs $\left(G_{\mathcal{E}}^{(\mathcal{M})_i^{\mathcal{R},O}}\right)_{i \in \mathbb{N}}$.

Lemma 2. *For any $i \in \mathbb{N}$:*

- $(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{STD}})_i^{\mathcal{R}} \subseteq G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{STD}}$
- $(\mathcal{M})_i^{\mathcal{R},O}$ *only contains instance mappings*

Proof. We start by proving that the set of RDFS triples of $\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}$ is a subset of $G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$, for each $i \in \mathbb{N}$. Using preceding results, we have the following equations for $i \in \mathbb{N}$:

$$\begin{aligned} O^{\mathcal{R}} &= \text{RDFS}(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}) \quad \text{(Corollary 1)} \\ &\subseteq \text{RDFS}(\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}) \\ &\subseteq \text{RDFS}(\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}}) \\ &= O^{\mathcal{R}} \quad \text{(Property 9)} \end{aligned}$$

We deduce that:

$$\forall i \in \mathbb{N},\ \text{RDFS}(\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}) = O^{\mathcal{R}}$$

Moreover, using Property 8, we know that:

$$\forall i \in \mathbb{N},\ O^{\mathcal{R}} \subseteq G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}} \subseteq G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}.$$

So finally, we prove that:

$$\forall i \in \mathbb{N},\ \text{RDFS}(\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}) \subseteq G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}.$$

After that, we just have to prove for $i \in \mathbb{N}$ the following statement $P(i)$:

- each non-RDFS triple of $\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}$ is in $G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$
- Let $q$ be an head of a instance mappings of $\mathcal{M}$, $\text{NR}_{q,i} = (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$ contains only correct triple for instance mapping head. So $(\mathcal{M})_i^{\mathcal{R},O}$ only contains instance mappings.

Here, we have to explain why in the second point of this list, the first sentences implies the second. It comes from the fact that if $q$ is an head of a mapping in $(\mathcal{M})_i^{\mathcal{R},O}$, then $\text{body}(q) \subset (\text{body}(q) \cup O)_i^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_i^{\mathcal{R}})$, according to Lemma 1.

We will prove $P(i)$ by induction. In the base case, we show that the statement holds for $i = 0$:

- $\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_0^{\mathcal{R}} = G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}} = G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_0^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$,
- For $q$ a head of mapping in $\mathcal{M}$, $(\text{body}(q) \cup O)_0^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_0^{\mathcal{R}}) = (\text{body}(q) \cup O) \setminus \text{RDFS}(\text{body}(q) \cup O) = (\text{body}(q) \cup O) \setminus O = \text{body}(q)$. So like previously explained we have: $(\mathcal{M})_0^{\mathcal{R},O} = \mathcal{M}$ only contains instance mappings.

In the inductive step, we assume that $P(i)$ holds for $i \in \mathbb{N}$, we will prove that $P(i+1)$ also holds. If $t'$ is a non-RDFS triple of $\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_{i+1}^{\mathcal{R}}$, then there are two cases:

- $t' \in \left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}$ so by hypothesis $t' \in G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$,
- or there exists a restricted rule $r \in \mathcal{R}$ such that $\left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}} \models^{\varphi} \text{body}(r)$ and $t' \in \varphi(\text{head}(r))^{\text{safe}}$.

Since $t' \in \varphi(\text{head}(r))^{\text{safe}}$ is a non-RDFS triple, $r$ is an instance rule. So there exists an $t \in \left(G_{\mathcal{E} \cup \mathcal{E}_O}^{\mathcal{M} \cup \mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}$ such that $\{t\} \cup O^{\mathcal{R}} \models^{\varphi} \text{body}(r)$. By the inductive hypothesis, $t$ is a triple of $G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}$. Hence there exists a mapping $m \in \mathcal{M}$ with $m = q_1 \rightsquigarrow q_2$, and a triple $t_m \in \text{body}((q_2)_i^{\mathcal{R},O})$ (defined in Definition 24) and a tuple $e \in ext(m)$ such that $v_e(t_m) = t$, where $v_e$ is the homomorphism induced by the replacement of answer variables of $(q_2)_i^{\mathcal{R},O}$ by the tuple $e$. Since $(q_2)_i^{\mathcal{R},O}$ is the head of a mapping of $(\mathcal{M})_i^{\mathcal{R},O}$, we know by induction hypothesis this mapping is actually an instance mapping. So according to Property 6, there exists a homomorphism $\varphi'$ such that $t_m \cup O^{\mathcal{R}} \models^{\varphi'} \text{body}(r)$ and $\varphi(\text{body}(r)) = v_e(\varphi'(\text{body}(r)))$. Hence, $\varphi(\text{head}(r))^{\text{safe}} = v_e(\varphi'(\text{head}(r))^{\text{safe}})$. We show that the mapping $q_1 \rightsquigarrow (q_2)_{i+1}^{\mathcal{R},O} \in (\mathcal{M})_{i+1}^{\mathcal{R},O}$ is such that $\varphi'(\text{head}(r)))^{\text{safe}} \subseteq \text{body}((q_2)_{i+1}^{\mathcal{R},O})$. Indeed, it is a consequence of Lemma 1, because we know that $\varphi'(\text{head}(r)))^{\text{safe}} \subseteq (\text{body}(q_2) \cup O)_{i+1}^{\mathcal{R}}$ and $\varphi'(\text{head}(r)))^{\text{safe}}$ contains only no RDFS triple. Finally, we have proved:

$$\begin{aligned} t' &\in \varphi(\text{head}(r))^{\text{safe}} \\ &= v_e(\varphi'(\text{head}(r))^{\text{safe}}) \\ &\subseteq v_e(\text{body}((q_2)_{i+1}^{\mathcal{R},O})) \\ &\subseteq G_{\mathcal{E} \cup \mathcal{E}_O}^{(\mathcal{M})_{i+1}^{\mathcal{R},O} \cup \mathcal{M}_O^{\text{STD}}}. \end{aligned}$$

We also have to prove that for $q$ an head of a mapping in $\mathcal{M}$, $\text{NR}_{q,i+1} = (\text{body}(q) \cup O)_{i+1}^{\mathcal{R}} \setminus \text{RDFS}((\text{body}(q) \cup O)_{i+1}^{\mathcal{R}})$ contains only valid triples for instance mapping head. By induction hypothesis, we know that $\mathcal{M}$, $\text{NR}_{q,i}$ verify the willing property. Let $t'$ a triple of $\text{NR}_{q,i+1}$, so there exists $r \in \mathcal{R}$ such that $t'$ is one directly entailed triple by $r$ on $(\text{body}(q) \cup O)_i^{\mathcal{R}}$. If the restricted rule $r$ is an ontological rule, then $t'$ is an RDFS triple. This case is absurd, because $t'$ will be in $\text{RDFS}((\text{body}(q) \cup O)_{i+1}^{\mathcal{R}})$ so $t' \notin \text{NR}_{q,i+1}$. So $r$ is an instance rule and there exists an $t \in \text{NR}_{q,i}$ such that $\{t\} \cup O^{\mathcal{R}} \models \text{body}(r)$. Using Property 7, we know that $t$ is a valid

triples for instance mapping head. Finally we can deduce from it that $(\mathcal{M})_{i+1}^{\mathcal{R},O}$ is a set of instance mappings. □

We are now able to prove the Theorem 2.

Proof Proof of Theorem 2. First, we prove the inclusion $G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}} \subseteq \left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}}$. Let $(\mathtt{s},\mathtt{p},\mathtt{o})$ be a triple from $G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}}$. Then, there exists a mapping $m \in \mathcal{M}\cup\mathcal{M}_O^{\text{STD}}$ such as $m = q_1(\bar{x}) \rightsquigarrow q_2(\bar{x})$ and there exists $\bar{t} \in \text{ext}(m)$ with $(\mathtt{s},\mathtt{p},\mathtt{o}) \in (\text{body}(q_2^{\mathcal{R},O})(\bar{t}))^{\text{safe}}$. We also know that:

- $(\text{body}(q_2)(\bar{t}))^{\text{safe}} \subseteq G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}$
- $O \subseteq G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}$, because of Property 8

We know that the saturation operation is monotonous, i.e., if $G, G'$ are RDF graphs such as $G \subseteq G'$, then $G^{\mathcal{R}} \subseteq G'^{\mathcal{R}}$. So if we put everything together, we have (considering inclusion by bijective renaming of blank nodes):

$$\begin{aligned}(\mathtt{s},\mathtt{p},\mathtt{o}) &\in (\text{body}(q_2^{\mathcal{R},O})(\bar{t}))^{\text{safe}}\\ &\subseteq ((\text{body}(q_2)(\bar{t}))^{\text{safe}} \cup O)^{\mathcal{R}}\\ &\subseteq \left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}}\end{aligned}$$

Finally, we have $(\mathtt{s},\mathtt{p},\mathtt{o}) \in \left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}}$.

Secondly, we prove that $\left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}} \subseteq G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}}$.

Let $t$ be a triple in $\left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}}$, by definition of the saturation of an RDF graph (Definition 3), there exists $i \in \mathbb{N}$ such that :

$$\begin{aligned}t &\in \left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)_i^{\mathcal{R}}\\ &\subseteq G_{\mathcal{E}\cup\mathcal{E}_O}^{(\mathcal{M})_i^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}} \quad \text{(thanks to Theorem 2)}\\ &\subseteq G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}}.\end{aligned}$$

□

## Proof of Theorem 3

Proof. For any extent $\mathcal{E}$ of $\mathcal{M}$, we prove that the certain answers of $q$ on the restricted $O$-system $\langle O,\mathcal{R},\mathcal{M},\mathcal{E}\rangle$ is equal to the certain answers of $q$ on the RDF system $\langle\emptyset,\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}},\mathcal{E}\cup\mathcal{E}_O\rangle$.

$$\begin{aligned}\text{cert}(q,\langle O,\mathcal{R},\mathcal{M},\mathcal{E}\rangle) &= \{\varphi(\bar{x}) \mid G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}} \models_{\mathcal{R}}^{\varphi} q(\bar{x})\}\\ &= \{\varphi(\bar{x}) \mid \left(G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}\cup\mathcal{M}_O^{\text{STD}}}\right)^{\mathcal{R}} \models^{\varphi} q(\bar{x})\}\\ &= \{\varphi(\bar{x}) \mid G_{\mathcal{E}\cup\mathcal{E}_O}^{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}} \models^{\varphi} q(\bar{x})\}\\ &= \text{cert}(q,\langle O,\emptyset,\mathcal{M}^{\mathcal{R},O},\mathcal{E}\rangle)\\ &= \text{cert}(q,\langle\emptyset,\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}},\mathcal{E}\cup\mathcal{E}_O\rangle)\end{aligned}$$

where $\varphi(\bar{x})$ is made of IRIs and literals only.

By Property 1, the certain answers of $q$ on the RDF system $\langle\emptyset,\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}},\mathcal{E}\cup\mathcal{E}_O\rangle$ are exactly the certain answers of $q$ on $(\mathcal{V}_{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}},\mathcal{E}\cup\mathcal{E}_O)$. By Theorem 1, since $q_r$ is a maximally contained rewriting of $q$ using $\mathcal{V}_{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}}$ w.r.t. union conjunctive queries on this views language, the certain answers of $q$ on views based integration of $\mathcal{V}_{\mathcal{M}^{\mathcal{R},O}\cup\mathcal{M}_O^{\text{STD}}}$ and $\mathcal{E}\cup\mathcal{E}_O$ are the answers of $q_r$ on the extent $\mathcal{E}\cup\mathcal{E}_O$. Because all preceding equality do not depend of the extent $\mathcal{E}$ of instance mappings $\mathcal{M}$, $q_r$ is a rewriting of $q$. □