

# Intensional and Extensional Views in DL-Lite Ontologies

Marco Console<sup>1</sup>, Giuseppe De Giacomo<sup>1</sup>, Maurizio Lenzerini<sup>1</sup> and Manuel Namicì<sup>1</sup>

<sup>1</sup>University of Rome, “La Sapienza”, Italy

{console,degiacomo,lenzerini,namici}@diag.uniroma1.it

## Abstract

The use of virtual collections of data is often essential in several data and knowledge management tasks. In the literature, the standard way to define virtual data collections is via views, i.e., virtual relations defined using queries. In data and knowledge bases, the notion of views is a staple of data access, data integration and exchange, query optimization, and data privacy. In this work, we study views in Ontology-Based Data Access (OBDA) systems. OBDA is a powerful paradigm for accessing data through an ontology, i.e., a conceptual specification of the domain of interest written using logical axioms. Intuitively, users of an OBDA system interact with the data only through the ontology’s conceptual lens. We present a novel framework to express natural and sophisticated forms of views in OBDA systems and introduce fundamental reasoning tasks for these views. We study the computational complexity of these tasks and present classes of views for which these tasks are tractable or at least decidable.

## 1 Introduction

Views have numerous applications in data and knowledge bases. For example, they can provide the user with virtual data collections derived from elementary objects, thus improving abstraction. Views can also represent a selection of the underlying data that limits the exposure of sensitive information. Indeed, users of a given class may have permission to query the knowledge base only through a set of views, effectively limiting their access to only specific parts of the data. Additionally, views may serve as a specification mechanism for data and knowledge exchange and integration. For example, a set of views can specify the portion of the source data that populates the target knowledge base in an exchange task, or the so-called global schema in data integration.

Generally speaking, the notion of view adapts naturally to ontological knowledge bases. In this paper, we focus on knowledge bases expressed in Description Logics (DL) and, specifically, in the lightweight DL *DL-Lite<sub>R</sub>* [Calvanese *et al.*, 2007b]. Intuitively, DLs are logical formalisms representing the domain of interest in terms of concepts, i.e., sets

of individuals, and roles, i.e., binary relations between objects. Surprisingly, few papers deal directly with views in DL knowledge bases. In [Buchheit *et al.*, 1998], the authors propose to split the axioms of a DL knowledge base into two sets, one for standard intensional knowledge regarding concepts and roles of interest (forming the schema) and one for defining views, i.e., new unary predicates with an associated definition. While intensional axioms may refer only to schema elements, view definitions may refer to both schema and view elements. In this context, views may be recursive, i.e., their definition may contain (either direct or indirect) references to themselves, thus beyond first-order logic. The authors advocate fixpoint semantics for dealing with cycles in the definitions. Another seminal work on views for DLs is [Beeri *et al.*, 1997], where the authors introduce the problem of rewriting queries using views. Intuitively, the latter asks for a query expression that uses only a set of views defined over a DL knowledge base and is equivalent to the input query. While query rewriting aims at computing a query over the views that is equivalent to the original one, view-based query answering has the goal of computing the answers to a query by relying only on the pre-computed extensions of a set of views. Observe that rewriting is a technique for addressing view-based query answering, but the latter is a much more general problem. Algorithms and complexity for view-based query answering in DLs, including *DL-Lite*, have been investigated in [Calvanese *et al.*, 2000; Calvanese *et al.*, 2012].

In this paper, we present a novel framework for including views in DL knowledge bases following an approach similar to the one adopted in [Buchheit *et al.*, 1998], but with the following differences. (i) View definitions in our framework consist of epistemic logic formulae rather than just DL concept expressions. Thus, they can use joins, projections, selections, and differences and the modal operator  $\mathbf{K}$  to talk about what is “known” in the knowledge base, i.e., answers for queries. (ii) We do not limit views to unary or binary predicates, and thus each view symbol has an associated arity, which corresponds to the number of distinguished variables of the formula constituting its definition. (iii) We disallow recursion in view definitions and base our semantics upon epistemic logic.

**Example 1.** Assume a DL knowledge base  $\mathcal{K}$  storing HR data.  $\mathcal{K}$  defines the following concepts and roles: Emp,

for employees,  $\text{Mng}$ , for managers,  $\text{Dep}$ , for departments,  $\text{hasDep}$ , connecting employees to their departments; and  $\text{hasMng}$ , connecting departments to their managers. The axioms of  $\mathcal{K}$  define the relationships between the concepts and roles introduced above.

A fundamental notion for the users of  $\mathcal{K}$  is the relationship between employees, their department, and their managers. This relation is ternary, and its definition requires joins, and therefore it is beyond the expressive power of DLs. However, we can define the desired relationship using a view  $V$ . However, to define  $V$  properly, we first need to clarify the users' intended meaning for this view.

For example, users may want to define the ontological concept  $\text{isMngOf}$  of being the manager of an employee in a department. The following formula  $\varphi_{\text{mng}}(x, y, z)$  defines this view:  $\text{Emp}(y) \wedge \text{hasDep}(y, z) \wedge \text{hasMng}(z, x)$ .

Also, users may be interested in restricting the concept  $\text{isMngOf}$  only to the employees, managers, and departments known to the system. That is, those obtained by the certain answers of the query  $\varphi_{\text{mng}}$  to  $\mathcal{K}$ . We call this view  $\text{qMngOf}$ . Observe that, while  $\text{isMngOf}$  is just a handy shortcut for  $\varphi_{\text{mng}}$ ,  $\text{qMngOf}$  is grounded upon an extra-logical concept, i.e., query answers. In this sense,  $\text{qMngOf}$  is closer to the standard notion of views for databases, and it requires an epistemic extension of first-order logic to be defined in a DL knowledge base. For its definition, we can use the following formula  $\varphi_{\text{qmng}}(x, y, z)$  of epistemic modal logic:  $\mathbf{K}(\varphi_{\text{mng}}(x, y, z))$ . Intuitively,  $\varphi_{\text{qmng}}(m, e, d)$  holds in some model of  $\mathcal{K}$  if only if  $\varphi_{\text{mng}}(m, e, d)$  holds in all the models of  $\mathcal{K}$ .

Finally, users may want to define complex views by means of a sophisticated use of the notion of query answers. For example, they may be interested in employees and managers of known departments, where the latter are the certain answers of  $\text{Dep}(x)$  over  $\mathcal{K}$ . The latter view, that we call  $\text{kMngOf}$ , can be defined by the formula  $\varphi_{\text{kmng}}(x, y, z)$  defined as follows:  $\varphi_{\text{mng}}(x, y, z) \wedge \mathbf{K}(\text{Dep}(z))$ .

The above example shows that views can enrich a DL knowledge base with new powerful abstractions. Such abstractions can play a crucial role in ontology exploration and query answering. The introduction of view predicates may also help overcome some intrinsic characteristics of Ontology-based Data Access (OBDA) systems. The first characteristic has to do with the nature of the ontology languages used in OBDA, which typically allow only unary and binary predicates. The use of views provides the possibility of using  $n$ -ary predicates in modelling the domain. The second characteristic is related to the restricted expressive power of DL axioms required for decidability. For example, no such languages allow the free use of joins in concepts and role expressions. Once again, views may help to mitigate such restrictions. Moreover, views can be used to introduce a lightweight mechanism for distinguishing truth from knowledge, i.e., being a given query's certain answer. The latter requires a modal mechanism, whose use through views remains controlled, avoiding intractability.

Views in knowledge bases can enhance all standard reasoning tasks of OBDA, including query answering and query containment. Moreover, there are specific reasoning tasks of interest on views, including checking consistency and con-

tainment. In this paper, we present several forms of view definitions, and we study the complexity of fundamental reasoning tasks involving them.

The rest of this paper is organized as follows. Section 2 introduces some preliminary notions; in Section 3 we present our framework; in Section 4 we study the complexity of reasoning with positive views; in Section 5 we discuss the impact of negation on view definitions; in Section 6 we conclude.

## 2 Preliminaries

We briefly review the main concepts used in the technical development of this paper.

**Logic and Complexity** We will assume basic familiarity with the standard notions of computational complexity and first-order logic, and refer the reader to [Arora and Barak, 2009] for a detailed account. Given a formula  $\phi$  and a vector of variables  $\bar{x}$ , we will sometimes use  $\phi(\bar{x})$  to say that  $\bar{x}$  are the free variables of  $\phi$ . A substitution  $\nu$  for  $\bar{x}$  is a mapping from  $\bar{x}$  to a given set of symbols, and  $\nu(\phi)$  is the formula obtained from  $\phi$  by substituting each  $x \in \bar{x}$  with  $\nu(x)$ .

**Description Logics Knowledge Bases** Let  $\mathcal{N}_I$ ,  $\mathcal{N}_C$ , and  $\mathcal{N}_R$  be countably infinite sets of individual, concept, and role names, respectively. A  $DL\text{-Lite}_{\mathcal{R}}$  knowledge base (KB), or ontology, is a pair  $\langle \mathcal{T}, \mathcal{A} \rangle$ , where  $\mathcal{T}$  and  $\mathcal{A}$  are, respectively, a  $DL\text{-Lite}_{\mathcal{R}}$  TBox and a Description Logic ABox, i.e., finite sets of logical axioms of the form

$$\begin{array}{lll} \mathcal{T} : & B_1 \sqsubseteq B_2; R_1 \sqsubseteq R_2 & (\textit{inclusion}) \\ & B_1 \sqsubseteq \neg B_2; R_1 \sqsubseteq \neg R_2 & (\textit{disjointness}) \\ \mathcal{A} : & A(a) \quad P(a, b) & (\textit{membership}) \end{array}$$

where  $a, b$  are individual names,  $A$  and  $P$  are concept and role names, respectively,  $B_1, B_2$  are *basic concepts*, i.e., expressions of the form  $A$ ,  $\exists R$ , and  $R$ ,  $R_1$ , and  $R_2$  are basic roles, i.e., expressions of the form  $P$  or  $P^-$ . Intuitively,  $\mathcal{T}$  and  $\mathcal{A}$  consist of the intensional and extensional information captured by the KB, respectively. As customary, semantics of  $DL\text{-Lite}_{\mathcal{R}}$  are defined via *Description Logic interpretations* (interpretations for short). An interpretation is a pair  $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ , where  $\Delta^{\mathcal{I}}$  is a countable set, called the domain of  $\mathcal{I}$ , and  $\cdot^{\mathcal{I}}$  is a function such that  $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ , for each  $a \in \mathcal{N}_I$ ,  $A \subseteq \Delta^{\mathcal{I}}$ , for each  $A \in \mathcal{N}_C$ , and  $P \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ , for each  $P \in \mathcal{N}_R$ . For interpretations, we assume *standard names* [Levesque and Lakemeyer, 2000], i.e., we assume that  $\Delta^{\mathcal{I}} = \mathcal{N}_I$  and  $a^{\mathcal{I}} = a$ , for every interpretation  $\mathcal{I}$  and  $a \in \mathcal{N}_I$ . In other words, we assume that constants are always interpreted into themselves, effectively blurring out the difference between constants and the individuals they represent. The standard name assumption is very common in OBDA, and it implies the *unique name assumption*. An interpretation  $\mathcal{I}$  satisfies an axiom  $\alpha \sqsubseteq \beta$  if  $\alpha^{\mathcal{I}} \subseteq \beta^{\mathcal{I}}$ ; it satisfies  $\alpha \sqsubseteq \neg\beta$  if  $\alpha^{\mathcal{I}} \cap \beta^{\mathcal{I}} = \emptyset$ ; and it satisfies  $A(a)$ , resp.  $P(a, b)$ , if  $a \in A^{\mathcal{I}}$ , resp.  $(a, b) \in P^{\mathcal{I}}$ . We say that  $\mathcal{I}$  satisfies  $\langle \mathcal{T}, \mathcal{A} \rangle$ , written  $\mathcal{I} \models \langle \mathcal{T}, \mathcal{A} \rangle$ , if  $\mathcal{I}$  satisfies all the axioms in the KB.

**Epistemic Knowledge Bases** We assume the standard epistemic logic based on the modal logic S5 and extend this concept to  $DL\text{-Lite}_{\mathcal{R}}$  ontologies in the spirit of [Calvanese et al., 2007a]. An *epistemic formula* is a standard first-order

formula over  $\mathcal{N}_I$ ,  $\mathcal{N}_C$ , and  $\mathcal{N}_R$  that can use the unary modal operator  $\mathbf{K}$  with the intuitive meaning of ‘it is known’ or ‘it is certain’. Epistemic formulae are interpreted over *epistemic interpretations*, i.e., pairs  $\langle w, W \rangle$ , where  $w$  is an interpretation and  $W$  is a set of interpretations that contains  $w$ . An epistemic interpretation  $\mathcal{W} = \langle w, W \rangle$  satisfies an epistemic formula  $\phi$  (written  $\langle w, W \rangle \models \phi$ ) if the following hold:

- $\phi = P(\bar{c})$ , with  $\bar{c} \in \mathcal{N}_I$ , and  $w \models \phi$ ;
- $\phi = \phi_1 \wedge \phi_2$  and  $\mathcal{W} \models \phi_1$  and  $\mathcal{W} \models \phi_2$ ;
- $\phi = \phi_1 \vee \phi_2$  and  $\mathcal{W} \models \phi_1$  or  $\mathcal{W} \models \phi_2$ ;
- $\phi = \neg\phi_1$  and  $\mathcal{W} \not\models \phi_1$ ;
- $\phi = \exists x.\phi_1(x)$  and  $\mathcal{W} \models \phi_1(c)$ , for some  $c \in \mathcal{N}_I$ .
- $\phi = \mathbf{K}(\phi_1)$  and  $\langle w, W \rangle \models \phi_1$ , for every  $w \in W$ .

An epistemic interpretation  $\mathcal{W} = \langle w, W \rangle$  is *compatible* with a KB  $\mathcal{K}$  if  $w \models \mathcal{K}$ . Compatible interpretations represent the possible epistemic states of  $\mathcal{K}$ , i.e., what  $\mathcal{K}$  might know for sure. Among compatible interpretations, we want those that carry *correct and minimal knowledge*. Intuitively, given an interpretation  $\mathcal{W}$  compatible with  $\mathcal{K}$ , to enforce correctness we require that every possible world of  $\mathcal{W}$  satisfies the statements of  $\mathcal{K}$ , and to enforce minimality we require that every interpretation that satisfies the statements of  $\mathcal{K}$  is a possible world of  $\mathcal{W}$ . In light of these considerations, we define the epistemic models of a *DL-Lite $\mathcal{R}$*  KB  $\mathcal{K}$  as follows. An interpretation  $\mathcal{W} = \langle w, W \rangle$  is an *epistemic model* for  $\mathcal{K}$  if  $\mathcal{W}$  is compatible with  $\mathcal{K}$  and  $w' \in W$  if and only if  $w' \models \mathcal{K}$ . With a slight abuse of notation, we write  $\mathcal{W} \models \mathcal{K}$  for  $\mathcal{W}$  is a model of  $\mathcal{K}$  and use  $Mod(\mathcal{K})$  to denote the epistemic models of  $\mathcal{K}$ . A *DL-Lite $\mathcal{R}$*  KB  $\mathcal{K}$  is satisfiable if  $Mod(\mathcal{K}) \neq \emptyset$ . Observe that the epistemic models of a *DL-Lite $\mathcal{R}$*  KB  $\mathcal{K}$  extend the standard notion of models of  $\mathcal{K}$ , i.e., the set of all interpretations that satisfy  $\mathcal{K}$ . Therefore, we can naturally extend reasoning tasks of standard *DL-Lite $\mathcal{R}$*  ontologies to the case of epistemic models. Assume an epistemic formula  $q(\bar{x})$  over the alphabet of  $\mathcal{K}$  with  $n$  free variables  $\bar{x}$ . The *certain answers* of  $\phi$  over  $\mathcal{K}$  are the set of substitutions  $\nu$  for the free variables of  $\phi$  such that  $\mathcal{W} \models \nu(\phi)$ , for every  $\mathcal{W} \in Mod(\mathcal{K})$ . Given a *DL-Lite $\mathcal{R}$*  KB  $\langle \mathcal{T}, \mathcal{A} \rangle$ , an epistemic formula  $\phi$ , and a substitution  $\nu$  for the free variables of  $\phi$ , *Query Answering* is the following decision problem: is  $\nu$  a certain answer for  $\phi$  over  $\langle \mathcal{T}, \mathcal{A} \rangle$ ? The *data complexity* of Query Answering is the complexity of checking whether  $\nu$  is a certain answer for  $\phi$  over  $\langle \mathcal{T}, \mathcal{A} \rangle$ , for fixed  $\mathcal{T}$  and  $\phi$ , and input  $\mathcal{A}$  and  $\nu$ . It is easy to see that, if  $\phi$  does not contain the operator  $\mathbf{K}$ , Certain Answers defined above coincide with standard certain answers for queries over Description Logics ontologies. Complexity of Query Answering in this context is known in the literature for many interesting cases. For example, when  $\phi$  is a *union of conjunctive queries*, i.e., the union of a set of existentially quantified conjunction of atoms without  $\mathbf{K}$ , Query Answering is *NP*-complete, and in *AC<sup>0</sup>* in data complexity ([Calvanese et al., 2007b]). Query Answering for queries expressed via epistemic formulae was first studied in [Calvanese et al., 2007a], where the authors show a broad class of epistemic formulae with *AC<sup>0</sup>* data complexity.

### 3 Views for Knowledge Bases

We proceed to introduce our framework for view-enriched KBs. Assume a countably infinite set  $\mathcal{N}_V$  of predicate symbols with an associated arity, pairwise disjoint with  $\mathcal{N}_I$ ,  $\mathcal{N}_C$ , and  $\mathcal{N}_R$ . We call  $\mathcal{N}_V$  the set of view names. In what follows, we use  $\mathcal{N}$  and  $\mathcal{N}_O$  to denote the set of all symbols, i.e.,  $\mathcal{N}_I \cup \mathcal{N}_C \cup \mathcal{N}_R \cup \mathcal{N}_V$ , and the set of ontological symbols  $\mathcal{N}_I \cup \mathcal{N}_C \cup \mathcal{N}_R$ , respectively. A *view definition*  $V$  (view for short) is a pair  $\langle v(\bar{x}), \phi(\bar{x}) \rangle$ , where  $v$  is a view name from  $\mathcal{N}_V$ , called the name of  $V$ ,  $\bar{x}$  is a tuple of distinct variables, and  $\phi(\bar{x})$  is an epistemic formula with free variables  $\bar{x}$ , called the definition of  $V$ . Observe that, since  $\phi$  is an epistemic formula, it cannot contain predicates from  $\mathcal{N}_V$ . A *View Box*  $\mathcal{V}$  (VBox for short) is a finite set of view definitions such that, for every distinct  $U, U' \in \mathcal{V}$ , the name of  $U$  is different from the name of  $U'$ . A *view-enriched DL-Lite $\mathcal{R}$  intensional specification* (intensional specification for short) is a pair  $\langle \mathcal{T}, \mathcal{V} \rangle$ , where  $\mathcal{T}$  is a *DL-Lite $\mathcal{R}$*  TBox and  $\mathcal{V}$  is a VBox. Finally, a *view-enriched DL-Lite $\mathcal{R}$  KB* (KB for short) is a triple  $\langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle$ , where  $\langle \mathcal{T}, \mathcal{V} \rangle$  is an intensional specification, and  $\mathcal{A}$  is an ABox.

**Example 2.** Assume the KB of Example 1. Relevant intensional knowledge is: (i) *hasDep* connects employees to departments; (ii) departments with a manager have at least one employee; (iii) all managers direct at least one department. The following TBox captures this information.

$$\begin{array}{ll} \mathcal{T} : & \exists \text{hasDep} \sqsubseteq \text{Emp} & \exists \text{hasDep}^- \sqsubseteq \text{Dep} \\ & \exists \text{hasMng} \sqsubseteq \exists \text{hasDep}^- & \text{Mng} \sqsubseteq \exists \text{hasMng}^- \end{array}$$

The ABox  $\mathcal{A}$  represents relevant extensional information.

$$\begin{array}{ll} \mathcal{A} : & \text{hasDep}(\text{Bob}, D1) & \text{hasMng}(D1, \text{Dylan}) \\ & \text{hasMng}(D2, \text{Cora}) & \text{Mng}(\text{Alice}) \end{array}$$

Views in the example can be defined by the following VBox

$$\mathcal{V} : \quad \langle \text{isMngOf}(\bar{x}), \varphi_{\text{mng}}(\bar{x}) \rangle \quad \langle \text{qMngOf}(\bar{x}), \varphi_{\text{qmng}}(\bar{x}) \rangle \\ \langle \text{kMngOf}(\bar{x}), \varphi_{\text{kmng}}(\bar{x}) \rangle$$

Finally, we write  $\mathcal{K}_{hr}$  for the triple  $\langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle$  in this example.

Semantics of view-enriched KBs extend the notion of epistemic models of *DL-Lite $\mathcal{R}$*  KBs. A *view-enriched interpretation* is a triple  $\mathcal{W} = \langle w, W, f \rangle$ , where  $\langle w, W \rangle$  is an epistemic interpretation over  $\mathcal{N}_O$ , and  $f$  is an interpretation for  $\mathcal{N}_V$  with domain  $\mathcal{N}_I$ . Given a view  $V = \langle v(\bar{x}), \phi(\bar{x}) \rangle$ , we say that  $\mathcal{W}$  *satisfies*  $V$  if,

$$v^f = \{ \nu(\bar{x}) \in \mathcal{N}_I^n \mid \langle w, W \rangle \models \nu(\phi(\bar{x})) \}$$

where  $\nu$  is a substitution for  $\bar{x}$ . We call  $v^f$  the *extension* of  $v$  in  $\mathcal{W}$  (sometimes denoted by  $v^{\mathcal{W}}$ ). In simple words,  $\mathcal{W}$  satisfies  $V$  if the extension of  $V$  in  $\mathcal{W}$  is defined according to  $\phi$ . Assume a view-enriched *DL-Lite $\mathcal{R}$*  KB  $\mathcal{K} = \langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle$ . We say that  $\mathcal{W}$  is a *model* for  $\mathcal{K}$  if  $\langle w, W \rangle$  satisfies  $\langle \mathcal{T}, \mathcal{A} \rangle$  and every  $V \in \mathcal{V}$ . We write  $Mod(\mathcal{K})$  for the set of all models of  $\mathcal{K}$  and  $\mathcal{K} \models \phi$  if  $\mathcal{W} \models \phi$ , for each  $\mathcal{W} \in Mod(\mathcal{K})$ . We illustrate semantics of view-enriched KBs in the following example.

**Example 3.** Assume formulae  $\psi_i(x) = \exists y \exists z R_i(x, y, z)$  with  $R_1 = \text{isMngOf}$ ,  $R_2 = \text{kMngOf}$ , and  $R_3 = \text{qMngOf}$ . Intuitively, each  $\psi_i$  asks for managers of departments with at

least one employee via the view  $R_i$ . Since every department with a manager has at least one employee, and every manager directs a department, we have  $\mathcal{K}_{HR} \models \psi_1(x)$  for  $x = Alice$ ,  $x = Cora$ , and  $x = Dylan$ . Moreover, since Dylan and Cora direct departments  $D1$  and  $D2$ , respectively, and  $D1$  and  $D2$  are known in the KB, we have  $\mathcal{K}_{HR} \models \psi_2(x)$  for  $x = Cora$ , and  $x = Dylan$ . On the other hand, while Alice is known to be a manager in the KB, the department she directs is not, and therefore  $\mathcal{K}_{HR} \not\models \psi_2(Alice)$ . Finally, since Bob works in  $D1$ , we have  $\mathcal{K}_{HR} \models \psi_3(Dylan)$ . On the other hand, Alice and Cora direct no known employee. Therefore,  $\mathcal{K}_{HR} \not\models \psi_3(x)$ , for  $x = Alice$ , and  $x = Cora$ .

### 3.1 Reasoning with Views

In a view-enriched KB, it is of interest to study several forms of reasoning involving views to enhance the conceptual analysis as well as our understanding of the data. Below we investigate these forms of reasoning.

**View Consistency** A fundamental reasoning task in DL KBs is concept and role consistency, i.e., checking whether the extension of a given concept or role is non-empty in some of the possible worlds described by the KB. Consistency allows designers to discover hidden properties of the model and automatically recognize possibly ill-defined concepts and roles. We extend consistency to view definitions with the notion of View Consistency. Given an intensional specification  $\mathcal{S}$  and a view definition  $V$  from  $\mathcal{S}$ , we say that  $V$  is consistent in  $\mathcal{S}$  if there exists an ABox  $\mathcal{A}$  and an interpretation  $\mathcal{W} \in Mod(\langle \mathcal{S}, \mathcal{A} \rangle)$  such that the extension of  $V$  in  $\mathcal{W}$  is not empty. *View Consistency* is the following decision problem: given  $\mathcal{S}$  and  $V$  as above, check whether  $V$  is consistent in  $\mathcal{S}$ .

**View Containment** Another fundamental reasoning task in DL KBs is concept and role containment. Intuitively, containment asks whether a concept or role extension is contained inside another in all the possible worlds described by  $\mathcal{K}$ . We extend containment to view definition with the notion of View Containment. Intuitively, a view  $V$  is contained in a view  $V'$  under the KB  $\mathcal{K}$  if the extension of  $V$  is all the models of  $\mathcal{K}$ . View Containment could be used, for instance, to check for redundancy in view definitions and optimize query answering. Perhaps more interestingly, view containment can be used to check whether a given view enjoys specific properties. For example, View Containment can *type* the components of a view, i.e., check whether they are subsumed by a given ontological concept.

We formalize the notion of view containment as follows. Given an intensional specification  $\mathcal{S}$  and view definitions  $V$  and  $V'$  from  $\mathcal{S}$ ,  $V$  is contained in  $V'$  under  $\mathcal{S}$ , written  $V \sqsubseteq_{\mathcal{S}} V'$ , if for every ABox  $\mathcal{A}$  and model  $\mathcal{W} \in Mod(\langle \mathcal{S}, \mathcal{A} \rangle)$ , the extension of  $V$  in  $\mathcal{W}$  is contained in the extension of  $V'$  in  $\mathcal{W}$ . *View Containment* is the following decision problem: given  $\mathcal{S}$ ,  $V$ , and  $V'$  as above, check whether  $V \sqsubseteq_{\mathcal{S}} V'$ .

**Query Answering** Given a formula  $\psi$  over  $\mathcal{N}$  and a view-enriched interpretation  $\mathcal{W}$ , we can define naturally when  $\mathcal{W}$  satisfies  $\psi$ , written  $\mathcal{W} \models \psi$ . In light of this consideration, given a KB  $\mathcal{K}$ , it is natural to talk about the Certain Answers for  $\psi$  over  $\mathcal{K}$  and therefore Query Answering. To formalize Query Answering in this context, we introduce two classes

of formulae over  $\mathcal{N}$ . A *View-Enriched Conjunctive Query* (VE-CQ) is an existentially-quantified conjunction of atoms with predicates from  $\mathcal{N}$ . A *View-Enriched Union of Conjunctive Query* (VE-UCQ) is the union of finitely-many VE-CQs. Given  $\mathcal{K}$  and a VE-UCQ  $Q(\bar{x})$ , Certain Answers for  $Q(\bar{x})$  over  $\mathcal{K}$  are defined as customary, and we use  $\text{ans}(Q, \mathcal{K})$  to denote the set of certain answers of  $Q$  over  $\mathcal{K}$ . *VE-UCQ Answering* is the following decision problem: for input a view-enriched *DL-Lite<sub>R</sub>* KB  $\langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle$ , VE-UCQ  $Q(\bar{x})$ , and substitution  $\nu$  for  $\bar{x}$ , check whether  $\nu \in \text{ans}(Q, \langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle)$ . The *data complexity of VE-UCQ Query Answering* is the complexity of checking whether  $\nu \in \text{ans}(Q, \langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle)$ , for input  $\mathcal{A}$  and  $\nu$ , and fixed  $Q$ ,  $\mathcal{T}$ , and  $\mathcal{V}$ . We call *VE-CQ Answering* the restriction of VE-UCQ Answering to VE-CQs.

**Query Containment** The basic reasoning task in query optimization is Query Containment, i.e., checking whether, given queries  $Q$  and  $Q'$ ,  $Q$  subsumes  $Q'$ . We can naturally extend Query Containment to view-enriched KBs as follows. Given VE-UCQs  $Q$  and  $Q'$  and an intensional specification  $\mathcal{S}$ , we say that  $Q$  is contained in  $Q'$  under  $\mathcal{S}$ , written  $Q \sqsubseteq_{\mathcal{S}} Q'$ , if  $\text{ans}(Q, \langle \mathcal{S}, \mathcal{A} \rangle) \subseteq \text{ans}(Q', \langle \mathcal{S}, \mathcal{A} \rangle)$ , for every ABox  $\mathcal{A}$ . *VE-UCQ Containment* is the following decision problem: for input  $Q$ ,  $Q'$ , and  $\langle \mathcal{T}, \mathcal{V} \rangle$  as above, check whether  $Q \sqsubseteq_{\mathcal{S}} Q'$ . We call *VE-CQ Containment* the restriction of VE-UCQ Containment to VE-CQs.

**Computability** Before concluding this section, we briefly discuss the decidability of the decision problems defined above.

Assume a VBox  $\mathcal{V}$  and a VE-CQ  $Q$  of the form  $\bigwedge_i R_i(\bar{z}_i) \wedge \bigwedge_j S_j(\bar{z}_j)$ , where each  $R_i$  is an ontological symbol from  $\mathcal{N}_O$ , and  $S_j$  is a view name from  $\mathcal{N}_V$  such that  $\langle S_j(\bar{x}), \phi_j(\bar{x}) \rangle \in \mathcal{V}$ . The *unfolding of  $Q$  according to  $\mathcal{V}$*  is the formula  $\text{unf}(Q, \mathcal{V}) \equiv \bigwedge_i R_i(\bar{z}_i) \wedge \bigwedge_j \psi_j(\bar{z}_j)$ , where each  $\psi_j(\bar{z}_j)$  is obtained from  $\phi_j(\bar{x})$  by

- Renaming apart the variables in  $\phi_j(\bar{x})$ ; and
- Substituting  $\bar{z}_j$  to  $\bar{x}$  in  $\phi$ .

Intuitively, the unfolding described above is a simplified version of the standard notion of mapping unfolding (see, e.g., [Poggi *et al.*, 2008]). The unfolding of a VE-UCQ  $Q$  according to  $\mathcal{V}$  is simply the union of the unfolding of all the conjunctive queries composing  $Q$ .

**Proposition 1.** *Assume a VE-UCQ  $Q$ , a VBox  $\mathcal{V}$ , and an interpretation  $\mathcal{W}$  that satisfies  $\mathcal{V}$ .  $\mathcal{W}$  satisfies  $Q$  if and only if it satisfies  $\text{unf}(Q, \mathcal{V})$ .*

Proposition 1 follows from the fact that  $Q$  and  $\text{unf}(Q, \mathcal{V})$  are logically equivalent under  $\mathcal{V}$ . Given a VE-UCQ  $Q$  and a VBox  $\mathcal{V}$ , it is clear that  $\text{unf}(Q, \mathcal{V})$  can be computed in polynomial time w.r.t.  $Q$  and  $\mathcal{V}$ . In spite of this result, all the decision problems defined above are undecidable for unrestricted view definitions due to the unrestricted use of first-order formulae. We say that an intensional specification is view-only if it is of the form  $\langle \emptyset, \mathcal{V} \rangle$ , i.e.,  $\mathcal{S}$  has no ontological axioms. It is easy to see that View Containment, View Consistency, VE-UCQ Answering, and VE-UCQ Containment are undecidable even for view-only *DL-Lite<sub>R</sub>* intensional specifications. Therefore, in order to achieve decidability, we need to restrict the language allowed in view definitions. In the

subsequent sections, we present meaningful decidable languages for view definitions and examine the complexity of relevant decision problems.

## 4 Lightweight Views

In this section, we introduce a class of view definitions that we dubbed lightweight. Lightweight views represent an interesting conceptual extension of  $DL\text{-Lite}_{\mathcal{R}}$  KBs while having a very limited impact on their computational characteristics. A *lightweight epistemic conjunction*  $\varphi(\bar{x})$  is an epistemic formula of the form

$$\exists \bar{y} \exists \bar{u}. \alpha(\bar{x}_o, \bar{y}_o, \bar{u}) \wedge \bigwedge_{i=1}^n \mathbf{K}(\exists \bar{z}_i. \beta_i(\bar{x}_i, \bar{y}_i, \bar{z}_i))$$

where  $\alpha$  and  $\beta_i$ , for each  $i$ , are conjunctions of relational atoms,  $\bar{x} = \bar{x}_o \cup \bar{x}_k$ , with  $\bar{x}_k = \bigcup_i \bar{x}_i$ , occur free in  $\varphi(\bar{x})$ ,  $\bar{y} = \bigcup_i \bar{y}_i$ , and  $\bar{y}_o \subseteq \bar{y}$ . A view definition  $\langle v(\bar{x}), \varphi(\bar{x}) \rangle$  is *lightweight* if  $\varphi(\bar{x})$  is a lightweight epistemic conjunction, and a knowledge base is lightweight if its views are. Observe that all the views in Example 2 are lightweight. We proceed to analyse the complexity of reasoning with lightweight views. First, given  $\varphi(\bar{x})$  as above, we define the following formulas.

- $\varphi^o(\bar{x}_o, \bar{y}_o) = \exists \bar{u}. \alpha(\bar{x}_o, \bar{y}_o, \bar{u})$ ;
- $\varphi^k(\bar{x}_k, \bar{y}) = \bigwedge_{i=1}^n \exists \bar{z}_i. \beta_i(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ , with  $\bar{x}_k = \bigcup_i \bar{x}_i$ ;
- $\varphi^+(\bar{x}, \bar{y}) \equiv \varphi^o(\bar{x}_o, \bar{y}_o) \wedge \varphi^k(\bar{x}_k, \bar{y})$

The canonical ABox  $\mathcal{A}$  of  $\varphi^+(\bar{x})$ , resp.,  $\varphi^k$  and  $\varphi^o$ , is an ABox obtained from  $\varphi^+(\bar{x})$ , resp.,  $\varphi^k$  and  $\varphi^o$ , by replacing each variable with a distinct individual name. When not otherwise specified, we will assume, w.l.o.g., that each variable  $z$  of  $q$  is replaced with  $c_z \in \mathcal{N}_I$  in  $\mathcal{A}$ , and denote by  $\bar{c}_x$  the image of  $\bar{x}$  in  $\mathcal{A}$ . In the remainder of this section, we assume a  $DL\text{-Lite}_{\mathcal{R}}$  intensional specification  $\mathcal{S} = \langle \mathcal{T}, \mathcal{V} \rangle$ , lightweight views  $V = \langle v(\bar{x}), \varphi(\bar{x}) \rangle$  and  $V' = \langle v'(\bar{x}), \varphi'(\bar{x}) \rangle$  in  $\mathcal{V}$ , and VE-UCQ  $Q$  and  $Q'$ .

**View Consistency** Consistency of lightweight views is tractable as the following theorem shows.

**Theorem 1.** *For input  $\mathcal{S}$  and  $V$  as above, View Consistency is in PTime.*

To prove the claim, we observe the following. For every  $\mathcal{W}$  satisfying  $\mathcal{V}$ ,  $v^{\mathcal{W}} \neq \emptyset$  if and only if  $\mathcal{W}$  contains a homomorphic copy of the canonical ABox  $\mathcal{A}$  of  $\varphi^+(\bar{x})$ . Therefore, one can prove that  $V$  is consistent in  $\langle \mathcal{T}, \mathcal{V} \rangle$  if and only if  $\langle \mathcal{T}, \mathcal{A} \rangle$  is satisfiable, being  $\mathcal{T}$  a  $DL\text{-Lite}_{\mathcal{R}}$  TBox. The desired complexity follows from the fact that satisfiability of  $DL\text{-Lite}_{\mathcal{R}}$  KBs can be checked in PTime.

**View Containment** View Containment is decidable for lightweight views as the following theorem shows.

**Theorem 2.** *For input  $\mathcal{S}$ ,  $V$  and  $V'$  as above, View Containment is NP-Complete.*

To prove the claim, we observe the following. Given an ABox  $\mathcal{A}$ , an *epistemic witness* of  $V'$  in  $\langle \mathcal{S}, \mathcal{A} \rangle$  is a substitution occurring in  $\text{ans}(\varphi'^k, \langle \mathcal{T}, \mathcal{A} \rangle)$ . Assume  $\mathcal{W} \in \text{Mod}(\mathcal{S}, \mathcal{A})$ . Intuitively, a tuple  $\nu(\bar{x})$  is in  $v^{\mathcal{W}}$  only if  $\nu$  can be extended to become an epistemic witness. Let  $\mathcal{A}_{\varphi^+}$  and  $\mathcal{A}_{\varphi^k}$  be the canonical ABox of  $\varphi^+$  and  $\varphi^k$ , respectively. One can prove

that  $V \sqsubseteq_{\mathcal{S}} V'$  if and only if there exists an epistemic witness of  $\varphi'$  in  $\langle \mathcal{S}, \mathcal{A}_{\varphi^k} \rangle$  such that  $\nu' \in \text{ans}(\langle \mathcal{T}, \mathcal{A}_{\varphi^+} \rangle)$ , for some extension  $\nu'$  of  $\nu$  such that  $\nu(\bar{x}) = \bar{c}_x$ . The desired complexity upper bound follows from the fact that we can guess, in nondeterministic polynomial time, Certain Answers for conjunctive queries over  $DL\text{-Lite}_{\mathcal{R}}$  KBs. The lower bound can be obtained with a reduction from Query Answering for  $DL\text{-Lite}_{\mathcal{R}}$  KBs.

**Query Answering** In light of Proposition 1, checking whether  $\nu \in \text{ans}(Q, \mathcal{K})$  amounts to checking whether  $\nu \in \text{ans}(\text{unf}(Q, \mathcal{V}), \langle \mathcal{T}, \mathcal{A} \rangle)$ . Therefore, we can prove decidability of VE-UCQ.

**Theorem 3.** *For input  $\mathcal{K}$  and  $Q$  as above, VE-UCQ Answering is NP-Complete and  $AC^0$  in data complexity.*

To prove the claim, we observe the following. By a slight abuse of notation, given  $\text{unf}(Q, \mathcal{V})$ , we use  $q \in \text{unf}(Q, \mathcal{V})$  to denote the disjuncts of  $\text{unf}(Q, \mathcal{V})$ . Observe that, for each such  $q$ ,  $\text{unf}(q, \mathcal{V})$  is a lightweight epistemic conjunction. In light of these considerations, one can prove that  $\nu \in \text{ans}(Q, \mathcal{K})$  if and only if there exists an extension  $\nu'$  of  $\nu$  such that  $\nu' \in \text{ans}(q^+, \mathcal{K})$ , for some  $q \in \text{unf}(Q, \mathcal{V})$ . The desired complexity follows from the fact that answering conjunctive queries over  $DL\text{-Lite}_{\mathcal{R}}$  KBs is NP-Complete and in  $AC^0$  in data complexity.

**Query Containment** Checking whether  $Q \sqsubseteq_{\mathcal{S}} Q'$  amounts to checking whether  $\text{unf}(Q, \mathcal{V})$  is contained in  $\text{unf}(Q', \mathcal{V})$  under  $\mathcal{T}$ . Once again, this is a consequence of Proposition 1. In light of these considerations, we can prove the following.

**Theorem 4.** *For input  $\mathcal{S}$ ,  $Q$  and  $Q'$  as above, VE-UCQ Containment is in NP-Complete.*

To prove the claim, we observe the following. One can prove that  $Q \sqsubseteq_{\mathcal{S}} Q'$  if and only if for every  $q_1 \in \text{unf}(Q, \mathcal{V})$  there exists  $q_2 \in \text{unf}(Q', \mathcal{V})$  such that  $\nu \in \text{ans}(q_2^+, \langle \mathcal{T}, \mathcal{A}_{q_1} \rangle)$ , where  $\mathcal{A}_{q_1}$  is the canonical ABox of  $q_1^+$  and  $\nu(\bar{x}) = \bar{c}_x$ . The desired complexity upper bound follows from the fact that one can guess an answer for  $q_2^+$  in nondeterministic polynomial time. To prove the lower bound, we can show a reduction from Query Answering for UCQ under  $DL\text{-Lite}_{\mathcal{R}}$  KBs.

## 5 Adding Negation

An important feature of ontological languages is the ability to express negative information, i.e., information about *what should not hold* for a given set of individuals. Unfortunately, several fundamental reasoning tasks become undecidable as soon as we allow unrestricted forms of negation in ontological KBs (see, e.g., [Gutiérrez-Basulto *et al.*, 2015]). A standard solution to restore decidability consists in restricting the occurrence of negation to specific syntactic positions. This is the case, e.g., of  $DL\text{-Lite}_{\mathcal{R}}$  KBs where negation can occur only in *disjointness* axioms. One may wonder whether a less restricted form of negation could be recovered using views. We explore this possibility in what follows, showing that, if we settle for *epistemic* negation, i.e., negation occurring only in front of the  $\mathbf{K}$  operator, it is possible to define expressive decidable languages of view definitions.

**Intensional Negation** We start our analysis of negation in view definitions with intensional negation, i.e., negation occurring only in front of non-epistemic atoms. An *intentional conjunction with safe negation* is a formula of the form  $\exists \bar{y}. \bigwedge_i R_i(\bar{x}_i, \bar{y}_i) \wedge \bigwedge_j \neg R_j(\bar{x}_j, \bar{y}_j)$ , where  $\bar{x} = \bigcup_i \bar{x}_i$ ,  $\bar{y} = \bigcup_i \bar{y}_i$ ,  $\bar{x}_j \subseteq \bar{x}$  and  $\bar{y}_j \subseteq \bar{y}$ , for every  $j$ , and  $R_i$  and  $R_j$  are predicates in  $\mathcal{N}_O$ . A *conjunctive view definition with safe negation* is a view definition  $\langle v(\bar{x}), \varphi(\bar{x}) \rangle$  such that  $\varphi$  is an intentional conjunction with safe negation. Conjunctive views with safe negation have a deep negative impact on decidability of query answering as the following proposition shows.

**Proposition 2.** *For conjunctive views with safe negation, VE-CQ Answering is undecidable.*

The claim follows from the fact that Query Answering for conjunctions with safe negation defined above is undecidable over *DL-Lite<sub>R</sub>* KBs ([Gutiérrez-Basulto *et al.*, 2015]).

**Epistemic Negation** A possible way to mitigate the negative effects of negation is to allow it in front of the **K** operator only. A *lightweight epistemic conjunction with negation* is an epistemic formula  $\varphi(\bar{x})$  of the form

$$\exists \bar{y}. \psi(\bar{x}, \bar{y}) \wedge \bigwedge_j \neg \mathbf{K}(\exists \bar{w}_j. \gamma_j(\bar{x}_j, \bar{y}_j, \bar{w}_j))$$

where  $\exists \bar{y}. \psi(\bar{x}, \bar{y})$  is a lightweight epistemic conjunction of the form  $\exists \bar{y} \exists \bar{u}. \alpha(\bar{x}_o, \bar{y}_o, \bar{u}) \wedge \bigwedge_{i=1}^n \mathbf{K}(\exists \bar{z}_i. \beta_i(\bar{x}_i, \bar{y}_i, \bar{z}_i))$ , and  $\bar{y}_j \subseteq \bar{y}$  and  $\bar{x}_j \subseteq \bigcup_i \bar{x}_i$ , for each  $j$ . A *lightweight view with negation* is a view definition  $\langle v(\bar{x}), \varphi(\bar{x}) \rangle$  where  $\varphi$  is a lightweight epistemic conjunction with negation. Views in this class can express meaningful properties, e.g., difference between certain answers of conjunctive queries, while preserving query answering decidability. The following theorem defines the computational complexity of query answering with this class of views.

**Theorem 5.** *For lightweight views with negation, VE-CQ answering is  $\Sigma_2^p$ -complete and  $AC^0$  in data complexity.*

To prove the claim, we simply observe that we can unfold  $Q$  according to  $\mathcal{V}$  and obtain a lightweight epistemic conjunction with negation  $\text{unf}(Q, \mathcal{V})$ . Therefore, the desired complexity upper bounds are a direct consequence of the results presented in [Calvanese *et al.*, 2007a]. To prove hardness, we can show an encoding of (the negation of) *tuple-generating dependencies* (tgds) as lightweight epistemic conjunctions with negation. The desired lower bounds follow from the fact that satisfaction of binary tgds is  $\Pi_2^p$ -hard ([Pichler and Skritek, 2011]).

While VE-CQ Answering remains decidable for lightweight views with negation, this is not the case of several other relevant reasoning tasks. A notable example is VE-UCQ Containment, as the following theorem shows.

**Theorem 6.** *VE-UCQ Containment is undecidable for intentional specifications with lightweight views with negation.*

To prove the claim, one can show a reduction from *finite tgd entailment*, i.e., the problem of checking whether  $\Sigma$  entails  $\tau$  over finite models, for input a set  $\Sigma$  of tgds and a single tgd  $\tau$ . Checking finite tgd entailment is a notoriously undecidable problem also in the case of binary tgds ([Beerli and Vardi, 1981]).

**Lightweight Epistemic Negation** Is it possible to preserve computability of relevant decision problems while still having a restricted form of negation in view definitions? Before concluding this section, we present a meaningful class of views with these characteristics. A *lightweight negative epistemic conjunction* is a lightweight epistemic conjunction with negation where  $w_j$  is empty, for each  $j$ . In simple words, lightweight negation forbids the use of existentially quantified variables inside the scope of  $\neg \mathbf{K}$ . A *lightweight negative view* is a view definition  $\langle v(\bar{x}), \varphi(\bar{x}) \rangle$  where  $\varphi$  is a lightweight negative epistemic conjunction. Lightweight negative views have a limited impact on the decidability of relevant reasoning tasks as the following theorem shows.

**Theorem 7.** *For lightweight negative views, VE-CQ Containment is in  $\Pi_3^p$ .*

To prove the claim, one can show that a counter-example of containment, i.e., an ABox  $\mathcal{A}$  such that  $\text{ans}(Q', \langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle)$  is not contained in  $\text{ans}(Q, \langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle)$ , has size at most polynomial in the input. With this upper bound in place, an algorithm can guess such  $\mathcal{A}$  in non-deterministic polynomial time, and check whether  $\mathcal{A}$  is an actual counterexample of containment simply by evaluating queries  $Q$  and  $Q'$  over  $\langle \mathcal{T}, \mathcal{V}, \mathcal{A} \rangle$ . The latter can be done using the result presented in Theorem 5. With similar arguments, one can prove decidability of View Containment.

**Theorem 8.** *For lightweight negative views, View Containment is in  $\Pi_3^p$ .*

Finally, we can show that View Consistency is decidable for lightweight views with lightweight negation. To prove this claim, we can once again show a polynomial upper bound on the size of a counter-example.

**Theorem 9.** *For lightweight negative views, View Consistency is in  $\text{coNP}$ .*

## 6 Conclusions

We presented a novel framework for enriching DL KBs with views. We defined languages for view definitions that can capture several interesting properties. For views defined in these languages, we studied the complexity of several reasoning tasks and showed that, under reasonable restrictions, views do not impact the overall complexity of reasoning with *DL-Lite<sub>R</sub>* KBs. The directions for future work are many. From the practical standpoint, it is of interest to implement our framework in real OBDA systems. From the theoretical standpoint, it is of interest to further investigate negation in view definitions, also considering relationship with the work on merging DLs and rules (e.g. [Rosati, 2008]).

## Acknowledgements

This work has been partially supported by the ERC Advanced Grant WhiteMech (No. 834228), by the EU ICT-48 2020 project TAILOR (No. 952215), by the ANR AI Chair INTENDED (ANR-19-CHIA-0014), and by the MIUR PRIN 2017 project ‘‘HOPE’’ (prot. 2017MMJJRE).

## References

- [Arora and Barak, 2009] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.
- [Beeri and Vardi, 1981] Catriel Beeri and Moshe Y. Vardi. The implication problem for data dependencies. In Shimon Even and Oded Kariv, editors, *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981, Proceedings*, volume 115 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 1981.
- [Beeri *et al.*, 1997] Catriel Beeri, Alon Y. Levy, and Marie-Christine Rousset. Rewriting queries using views in description logics. In Alberto O. Mendelzon and Z. Meral Özsoyoglu, editors, *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona, USA*, pages 99–108. ACM Press, 1997.
- [Buchheit *et al.*, 1998] Martin Buchheit, Francesco M. Donini, Werner Nutt, and Andrea Schaerf. A refined architecture for terminological systems: Terminology = schema + views. *Artif. Intell.*, 99(2):209–260, 1998.
- [Calvanese *et al.*, 2000] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Answering queries using views over description logics knowledge bases. In Henry A. Kautz and Bruce W. Porter, editors, *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, pages 386–391. AAAI Press / The MIT Press, 2000.
- [Calvanese *et al.*, 2007a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Eql-lite: Effective first-order query processing in description logics. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 274–279, 2007.
- [Calvanese *et al.*, 2007b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
- [Calvanese *et al.*, 2012] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. View-based query answering in description logics: Semantics and complexity. *J. Comput. Syst. Sci.*, 78(1):26–46, 2012.
- [Gutiérrez-Basulto *et al.*, 2015] Víctor Gutiérrez-Basulto, Yazmín Angélica Ibáñez-García, Roman Kontchakov, and Egor V. Kostylev. Queries with negation and inequalities over lightweight ontologies. *J. Web Semant.*, 35:184–202, 2015.
- [Levesque and Lakemeyer, 2000] Hector J. Levesque and Gerhard Lakemeyer. *The logic of knowledge bases*. MIT Press, 2000.
- [Pichler and Skritek, 2011] Reinhard Pichler and Sebastian Skritek. The complexity of evaluating tuple generating dependencies. In Tova Milo, editor, *Database Theory - ICDT 2011, 14th International Conference, Uppsala, Sweden, March 21-24, 2011, Proceedings*, pages 244–255. ACM, 2011.
- [Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.
- [Rosati, 2008] Riccardo Rosati. On combining description logic ontologies and nonrecursive datalog rules. In Diego Calvanese and Georg Lausen, editors, *Web Reasoning and Rule Systems, Second International Conference, RR 2008, Karlsruhe, Germany, October 31-November 1, 2008. Proceedings*, volume 5341 of *Lecture Notes in Computer Science*, pages 13–27. Springer, 2008.