Privacy Preserving Query Answering in Description Logics Through Instance Indistinguishability

(Discussion Paper)

Gianluca Cima[1], Domenico Lembo[2], Riccardo Rosati[2] and Domenico Fabio Savo[3]

[1]*University of Bordeaux, CNRS, Bordeaux INP, LaBRI*
[2]*Sapienza Università di Roma*
[3]*Università degli Studi di Bergamo*

Abstract

We study privacy-preserving query answering in Description Logics (DLs). Specifically, we consider the approach of controlled query evaluation (CQE) based on the notion of *instance indistinguishability*. We derive data complexity results for query answering over *DL-Lite*$_\mathcal{R}$ ontologies, through a comparison with an alternative, existing confidentiality-preserving approach to CQE. Finally, we identify a semantically well-founded notion of approximated query answering for CQE, and prove that, for *DL-Lite*$_\mathcal{R}$ ontologies, this form of CQE is tractable with respect to data complexity and is first-order rewritable, i.e., it is always reducible to the evaluation of a first-order query over the data instance.

1. Introduction

We consider controlled query evaluation (CQE), a declarative framework for privacy-preserving query answering investigated in the literature on knowledge representation and database theory [1, 2, 3]. The basic idea of CQE is defining a *data protection policy* through logical statements. Consider for instance an organization that wants to keep confidential the fact that it has suppliers involved in both Project A and Project B. This can be expressed over the information schema of the organization through a denial assertion of the form $\forall x.\, \mathsf{Supplier}(x) \wedge \mathsf{ProjA}(x) \wedge \mathsf{ProjB}(x) \rightarrow \bot$

In CQE, two different main approaches can be identified. The first one [4, 5, 6] models privacy preservation through the notion of *indistinguishable data instances*. In this approach, a system for CQE enforces data privacy if, for every data instance I, there exists a data instance I' that does not violate the data protection policy and is indistinguishable from I for the user, i.e., for every user query q, the system provides the same answers to q over I and over I'. We call this approach *(instance) indistinguishability-based* (IB). In continuation of the previous example,

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

gianluca.cima@u-bordeaux.fr (G. Cima); lembo@diag.uniroma1.it (D. Lembo); rosati@diag.uniroma1.it (R. Rosati); domenicofabio.savo@unibg.it (D. F. Savo)
0000-0003-1783-5605 (G. Cima); 0000-0002-0628-242X (D. Lembo); 0000-0002-7697-4958 (R. Rosati); 0000-0002-8391-8049 (D. F. Savo)

in the presence of an instance $\{\mathsf{Supplier}(c), \mathsf{ProjA}(c), \mathsf{ProjB}(c)\}$, an IB system should answer user queries as if the instance were, e.g., $\{\mathsf{Supplier}(c), \mathsf{ProjA}(c)\}$ (note that other instances not violating the policy can be considered as indistinguishable, e.g., $\{\mathsf{Supplier}(c), \mathsf{ProjB}(c)\}$).

The second approach [7, 8] models privacy preservation by considering the whole (possibly infinite) set of answers to queries that the system provides to the user. In this approach, a CQE system protects the data if, for every data instance I, the logical theory corresponding to the set of answers provided by the system to all queries over I does not entail any violation of the data protection policy. According to [8], we call this approach *confidentiality-preserving (CP)*. In our ongoing example, a CP system would entail, e.g., the queries $\mathsf{Supplier}(c) \wedge \mathsf{ProjA}(c)$ and $\exists x.\mathsf{Supplier}(x) \wedge \mathsf{ProjB}(x)$, but not also the query $\mathsf{Supplier}(c) \wedge \mathsf{ProjB}(c)$.

In both approaches, the ultimate goal is to realize *optimal* CQE systems, i.e., systems maximizing the answers returned to user queries, still respecting the data protection policy. Traditionally, this aim has been pursued through the construction of a *single optimal censor*, i.e., a specific implementation of the adopted notion of privacy-preservation, either IB or CP. Since, however, in both approaches several optimal censors typically exist, this way of proceeding requires to make a choice on how to obfuscate data, which, in the absence of additional (preference) criteria, may result discretionary. To avoid this, query answering over all optimal censors has been recently studied, limited to the CP approach [7, 9], whereas query answering over all optimal IB censors has not been investigated so far. Moreover, among the complexity results obtained and the techniques defined so far for CQE, we still miss the identification of cases that are promising towards its practical usage.

In this paper, we aim at filling some of the above mentioned gaps in the context of Description Logic (DL) ontologies. We focus on the approach to CQE based on instance indistinguishability (Section 3), and study its relationship with the CP approach (Section 4). Specifically, we prove that the IB approach to CQE in DLs corresponds to a particular instance of the CP approach to CQE [9]. Based on such a correspondence, we show that, even in the lightweight DL *DL-Lite*$_{\mathcal{R}}$, query answering in the IB approach is intractable with respect to data complexity, unless one relies on a single optimal censor chosen non-deterministically in the lack of further meta-information about the domain of the dataset.

To overcome the above problems and provide a practical, semantically well-founded solution, we define a *quasi-optimal* notion of IB censor, which corresponds to the best sound approximation of all the optimal IB censors (Section 5). We then prove that, in the case of *DL-Lite*$_{\mathcal{R}}$ ontologies, query answering based on the quasi-optimal IB censor is reducible to the evaluation of a first-order query over the data instance, i.e., it is *first-order rewritable*. We believe that this result has an important practical impact. Indeed, we have identified a setting in which privacy-preserving query answering formalized in a declarative logic-based framework as CQE, for a DL (i.e., *DL-Lite*$_{\mathcal{R}}$) specifically designed for data management, has the same data complexity as evaluating queries over a database (i.e., in AC^0). This opens the possibility of defining algorithms for CQE of practical usage, amenable to implementation on top of traditional (relational) data management systems, as done in [10] exploiting Ontology-based Data Access engines.

This paper summarizes the results of our IJCAI-20 conference publication [11].

2. Preliminaries

We use standard notions of function-free first-order (FO) logic, and in particular we consider Description Logics (DLs), which are fragments of FO using only unary and binary predicates, called concepts and roles, respectively.We assume to have the pairwise disjoint countably infinite sets $\Sigma_C, \Sigma_R, \Sigma_I$ and $\Sigma_\mathcal{V}$ for *atomic concepts*, *atomic roles*, *constants* (a.k.a. individuals), and *variables*, respectively. A DL ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ is constituted by a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, specifying intensional and extensional knowledge, respectively. The set of atomic concepts and roles occurring in $\mathcal{O}$ is the *signature* of $\mathcal{O}$. The semantics of $\mathcal{O}$ is given in terms of FO models over the signature of $\mathcal{O}$, in the standard way.In particular, we say that $\mathcal{O}$ is *consistent* if it has at least one model, *inconsistent* otherwise. $\mathcal{O}$ *entails* an FO sentence ϕ specified over the signature of $\mathcal{O}$, denoted $\mathcal{O} \models \phi$, if ϕ is true in every model of $\mathcal{O}$. In this paper, we consider ontologies expressed in *DL-Lite*$_\mathcal{R}$, the member of the *DL-Lite* family [12] which underpins OWL 2 QL, i.e., the OWL 2 profile specifically designed for efficient query answering. A TBox $\mathcal{T}$ in *DL-Lite*$_\mathcal{R}$ is a finite set of axioms of the form $B_1 \sqsubseteq B_2$ (resp., $R_1 \sqsubseteq R_2$), denoting concept (resp., role) inclusion, and $B_1 \sqsubseteq \neg B_2$ (resp., $R_1 \sqsubseteq \neg R_2$), denoting concept (resp., role) disjointness, where: R_1, R_2 are of the form P, with $P \in \Sigma_R$, or its inverse P^-, and B_1, B_2 are of the form A, with $A \in \Sigma_C$, $\exists P$, or $\exists P^-$, i.e., unqualified existential restrictions, which denote the set of objects occurring as first or second argument of P, respectively. An ABox $\mathcal{A}$ is a finite set of *ground atoms*, i.e., assertions of the form $A(a)$, $P(a, b)$, where $A \in \Sigma_C$, $P \in \Sigma_R$, and $a, b \in \Sigma_I$. As usual in query answering over DL ontologies, we focus on conjunctive queries. A Boolean conjunctive query (BCQ) q is an FO sentence of the form $\exists \vec{x}.\phi(\vec{x})$, where $\vec{x}$ are variables in $\Sigma_\mathcal{V}$, and $\phi(\vec{x})$ is a finite, non-empty conjunction of atoms of the form $\alpha(\vec{t})$, where $\alpha \in \Sigma_C \cup \Sigma_R$, and each term in $\vec{t}$ is either a constant in Σ_I or a variable in $\vec{x}$. We denote by *Eval*$(q, \mathcal{A})$ the result of the evaluation of a query q over (the model isomorphic to) an ABox $\mathcal{A}$.

A *denial assertion* (or simply a denial) is an FO sentence of the form $\forall \vec{x}.\phi(\vec{x}) \rightarrow \bot$, such that $\exists \vec{x}.\phi(\vec{x})$ is a BCQ. Given one such denial δ and an ontology $\mathcal{O}$, we say that $\mathcal{O} \cup \{\delta\}$ is consistent if $\mathcal{O} \not\models \exists \vec{x}.\phi(\vec{x})$, and is inconsistent otherwise.

In the following, with $\mathbf{FO}$, $\mathbf{CQ}$, and $\mathbf{GA}$, we denote the languages of function-free FO sentences, BCQs, and ground atoms, respectively, all specified over the alphabets $\Sigma_C, \Sigma_R, \Sigma_I$, and $\Sigma_\mathcal{V}$. Given an ontology $\mathcal{O}$ and a language $\mathcal{L}$, with $\mathcal{L}(\mathcal{O})$ we refer to the subset of $\mathcal{L}$ whose sentences are built over the signature of $\mathcal{O}$ and the variables in $\Sigma_\mathcal{V}$. For a TBox $\mathcal{T}$ and a language $\mathcal{L}$, we denote by $\mathsf{cl}^\mathcal{T}_\mathcal{L}(\cdot)$ the function that, for an ABox $\mathcal{A}$, returns all the sentences $\phi \in \mathcal{L}(\mathcal{T} \cup \mathcal{A})$ such that $\mathcal{T} \cup \mathcal{A} \models \phi$.

For the sake of presentation, we will limit our technical treatment to languages containing only closed formulas, but our results hold also for open formulas. In particular, the results on entailment of BCQs can be extended to arbitrary (i.e., non-Boolean) CQs in the standard way[1] . Our complexity results are for data complexity, i.e., are w.r.t. the size of the ABox only.

[1]We also notice that, since *DL-Lite*$_\mathcal{R}$ is insensitive to the adoption of the *unique name assumption* (UNA) for CQ answering,our results hold both with and without UNA.

3. CQE through Instance Indistinguishability

A CQE framework consists of a TBox $\mathcal{T}$ and a policy $\mathcal{P}$ over $\mathcal{T}$, i.e., a finite set of denial assertions over the signature of $\mathcal{T}$. An ABox $\mathcal{A}$ for $\mathcal{T}$ is such that $\mathcal{A}$ and $\mathcal{T}$ have the same signature. In the following, when a TBox $\mathcal{T}$ is given, we always assume that the coupled policy is specified over $\mathcal{T}$, that each considered ABox $\mathcal{A}$ is for $\mathcal{T}$, and that $\mathcal{T} \cup \mathcal{A}$ and $\mathcal{T} \cup \mathcal{P}$ are consistent. A *censor* is a function that taken an ABox for $\mathcal{T}$ as input alters standard query answering over $\mathcal{T} \cup \mathcal{A}$ so that on the basis of the answers (even a possibly infinite set thereof) and the TBox, a user can never infer a BCQ $\exists \vec{x}.\phi(\vec{x})$ such that $\forall \vec{x}.\phi(\vec{x}) \rightarrow \bot$ belongs to $\mathcal{P}$.

We here propose a notion of censor which is the natural application to our framework of the analogous definitions given in [4, 5, 6, 13, 14]. The basic idea of this approach is that for every underlying instance (an ABox in our framework) and every query, a censor returns to the user the same answers it would return on another (possibly identical) instance that does not contain confidential data, so that she cannot understand which of the two instances she is querying. This is formalized as follows.

Definition 1. [Indistinguishability-based censor] Let $\mathcal{T}$ be a DL TBox and $\mathcal{P}$ be a policy. An *indistinguishability-based (IB) censor* for $\mathcal{T}$ and $\mathcal{P}$ is a function $\mathsf{cens}(\cdot)$ that, for each ABox $\mathcal{A}$, returns a set $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathcal{A})$ such that there exists an ABox $\mathcal{A}'$ for which (i) $\mathsf{cens}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathcal{A}')$ (in this case we say that $\mathcal{A}$ and $\mathcal{A}'$ are *indistinguishable* w.r.t. cens) and (ii) $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is a consistent FO theory.

Example 1. The TBox signature consists of the atomic concepts $\mathsf{Supplier}$, ProjA, and ProjB, denoting the set of suppliers of the company, suppliers involved in Project A and those involved in Project B, respectively, and contains the axioms $\mathsf{ProjA} \sqsubseteq \mathsf{Supplier}$ and $\mathsf{ProjB} \sqsubseteq \mathsf{Supplier}$, stating that each individual instance of ProjA or ProjB is also instance of $\mathsf{Supplier}$. Data protection is specified through the policy $\mathcal{P} = \{\forall x.\mathsf{ProjA}(x) \wedge \mathsf{ProjB}(x) \rightarrow \bot\}$. The following functions are IB censors for $\mathcal{T}$ and $\mathcal{P}$:

- cens_1: given an ABox $\mathcal{A}$, $\mathsf{cens}_1(\mathcal{A})$ returns the set $\mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathcal{A}_{P_A})$ of BCQs, where $\mathcal{A}_{P_A}$ is obtained from $\mathcal{A}$ by removing the assertion $\mathsf{ProjA}(c)$, for each individual c such that both $\mathsf{ProjA}(c)$ and $\mathsf{ProjB}(c)$ are in $\mathcal{A}$ (note that for every ABox $\mathcal{A}$, $\mathcal{A}$ and $\mathcal{A}_{P_A}$ are indistinguishable w.r.t. cens_1. Similarly in the following censors).
- cens_2: given an ABox $\mathcal{A}$, $\mathsf{cens}_2(\mathcal{A})$ returns the set $\mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathcal{A}_{P_B})$ of BCQs, where $\mathcal{A}_{P_B}$ is obtained from $\mathcal{A}$ by removing the assertion $\mathsf{ProjB}(c)$, for each individual c such that both $\mathsf{ProjA}(c)$ and $\mathsf{ProjB}(c)$ are in $\mathcal{A}$.
- cens_3: given an ABox $\mathcal{A}$, $\mathsf{cens}_3(\mathcal{A})$ returns the set $\mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathcal{A}_{sup})$ of BCQs, where $\mathcal{A}_{sup}$ is obtained from $\mathcal{A}$ by adding the assertion $\mathsf{Supplier}(c)$ and removing $\mathsf{ProjA}(c)$ and $\mathsf{ProjB}(c)$, for each individual c such that both $\mathsf{ProjA}(c)$ and $\mathsf{ProjB}(c)$ are in $\mathcal{A}$. □

It is easy to see that an IB censor always exists, but, as Example 1 shows, there may be many IB censors for $\mathcal{T}$ and $\mathcal{P}$, and so it is reasonable to look for censors preserving as much information as possible. Formally, given two IB censors cens and cens' for $\mathcal{T}$ and $\mathcal{P}$, we say that cens' is *more informative* than cens if: *(i)* for every ABox $\mathcal{A}$, $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cens}'(\mathcal{A})$, and *(ii)* there exists an ABox $\mathcal{A}'$ such that $\mathsf{cens}(\mathcal{A}') \subset \mathsf{cens}'(\mathcal{A}')$. Optimal censors are then defined as follows.

Definition 2. Let $\mathcal{T}$ be a DL TBox and $\mathcal{P}$ be a policy. An IB censor cens for $\mathcal{T}$ and $\mathcal{P}$ is *optimal* if there does not exist any other IB censor for $\mathcal{T}$ and $\mathcal{P}$ that is more informative than cens. We denote by $\mathsf{OptIBCens}_{\mathcal{T},\mathcal{P}}$ the set of the optimal IB censors for $\mathcal{T}$ and $\mathcal{P}$.

Example 2. Among the censors of Example 1, $\mathsf{cens}_3 \notin \mathsf{OptIBCens}_{\mathcal{T},\mathcal{P}}$, since both cens_1 and cens_2 are more informative than cens_3. It can be then verified that cens_1 and cens_2 are the only optimal IB censors for $\mathcal{T}$ and $\mathcal{P}$. □

4. IB Censors vs. CP Censors

In [8], a different notion of censor, named *confidentiality-preserving (CP)* censor, has been proposed. Intuitively, a CP censor establishes which are the BCQs entailed by a TBox and a given ABox that can be disclosed without violating the policy. We report below the definition given in [9], which generalizes CP censors to any language $\mathcal{L} \subseteq \mathbf{FO}$, called the censor language.

Definition 3. [Confidentiality-preserving censor] Let $\mathcal{T}$ be a DL TBox, $\mathcal{P}$ be a policy, and $\mathcal{L} \subseteq \mathbf{FO}$ be a language. A *confidentiality-preserving (CP) censor* in $\mathcal{L}$ for $\mathcal{T}$ and $\mathcal{P}$ is a function $\mathsf{cens}(\cdot)$ that, for each ABox $\mathcal{A}$, returns a set $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}}(\mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is a consistent FO theory.

The notion of more informative censor previously given for IB censors can be naturally extended to CP censors, and so we can define optimal censors also in this case. We denote by $\mathcal{L}$-$\mathsf{OptCPCens}_{\mathcal{T},\mathcal{P}}$ the set of the optimal CP censors in $\mathcal{L}$ for $\mathcal{T}$ and $\mathcal{P}$.

Example 3. Consider $\mathcal{T}$ and $\mathcal{P}$ as defined in Example 1. An optimal CP censor cens_4 in $\mathbf{CQ}$ for $\mathcal{T}$ and $\mathcal{P}$ is defined as follows: given an ABox $\mathcal{A}$, $\mathsf{cens}_4(\mathcal{A})$ returns the set of BCQs obtained by removing from $\mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathcal{A})$ every query containing the atom $\mathsf{ProjA}(c)$, for each individual c such that both $\mathsf{ProjA}(c)$ and $\mathsf{ProjB}(c)$ are in $\mathcal{A}$.

We notice that the CP censor cens_4 is not an IB censor. Indeed, consider the ABox $\mathcal{A} = \{\mathsf{ProjA}(c), \mathsf{ProjB}(c)\}$. We have that $\mathsf{cens}_4(\mathcal{A}) = \{\phi \mid \phi \in \mathbf{CQ} \text{ and } \mathcal{T} \cup \mathcal{S} \models \phi\}$, where $\mathcal{S} = \{\exists x.\mathsf{ProjA}(x), \mathsf{ProjB}(c)\}$. It is not hard to see that there exists no ABox $\mathcal{A}'$ such that $\mathcal{A}'$ and $\mathcal{A}$ are indistinguishable w.r.t. cens_4 and $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is consistent. □

Let $\mathcal{A}$ be an ABox and cens be either an IB or a CP censor, the set $\mathsf{cens}(\mathcal{A})$ is called *theory of the censor* cens for $\mathcal{A}$.

The following theorems explain the relation between IB censors and CP censors.

Theorem 1. *Let $\mathcal{T}$ be a DL TBox and $\mathcal{P}$ be a policy. If* cens *is an IB censor for $\mathcal{T}$ and $\mathcal{P}$, then it is a CP censor in $\mathbf{CQ}$ for $\mathcal{T}$ and $\mathcal{P}$. The converse does not necessarily hold.*

Theorem 2. *Let $\mathcal{T}$ be a DL TBox and $\mathcal{P}$ be a policy. Then,* $\mathsf{ib\_cens} \in \mathsf{OptIBCens}_{\mathcal{T},\mathcal{P}}$ *iff there exists a CP censor* $\mathsf{cp\_cens} \in \mathbf{GA}\text{-}\mathsf{OptCPCens}_{\mathcal{T},\mathcal{P}}$ *such that, for each ABox $\mathcal{A}$,* $\mathsf{cl}^{\mathcal{T}}_{\mathbf{CQ}}(\mathsf{cp\_cens}(\mathcal{A})) = \mathsf{ib\_cens}(\mathcal{A})$.

From Theorem 2 and the results given in [9], it follows that, given a *DL-Lite*$_{\mathcal{R}}$ TBox $\mathcal{T}$, a policy $\mathcal{P}$, an ABox $\mathcal{A}$, and a BCQ q, deciding whether $q \in \mathsf{cens}(\mathcal{A})$ for every $\mathsf{cens} \in \mathsf{OptIBCens}_{\mathcal{T},\mathcal{P}}$ is coNP-complete in data complexity.

5. Approximating Optimal IB Censors

Towards a practical approach to CQE, in this section we consider a different entailment problem that approximates entailment under *each* optimal censor, and we show that its data complexity is in AC^0 (i.e., the same complexity of evaluating FO queries over a database). The approximation we propose consists in considering a non-necessarily optimal IB censor whose theory, for every ABox, is as close as possible to the theories of all the optimal IB censors. We call such censor *quasi-optimal IB (QIB) censor*.

Definition 4. [QIB censor] Let $\mathcal{T}$ be a DL TBox, let $\mathcal{P}$ be a policy, and let cens be an IB censor for $\mathcal{T}$ and $\mathcal{P}$. We say that cens is a QIB censor if: (*i*) $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cens}'(\mathcal{A})$, for each $\mathsf{cens}' \in \mathsf{OptIBCens}_{\mathcal{T},\mathcal{P}}$, and (*ii*) there exists no IB censor cens' for $\mathcal{T}$ and $\mathcal{P}$ which is more informative than cens and that satisfies condition.

Example 4. The IB censor cens_3 of Example 1 is a QIB censor for $\mathcal{T}$ and $\mathcal{P}$ (but $\mathsf{cens}_3 \notin \mathsf{OptIBCens}_{\mathcal{T},\mathcal{P}}$). □

It is not hard to see that a QIB censor for a DL TBox $\mathcal{T}$ and a policy $\mathcal{P}$ always exists and it is unique. Hereinafter, we denote with $\mathsf{qib\_cens}_{\mathcal{T},\mathcal{P}}$ the QIB censor for $\mathcal{T}$ and $\mathcal{P}$.

Definition 5. Let $\mathcal{T}$ be a DL TBox, $\mathcal{P}$ be a policy, $\mathcal{A}$ be an ABox, and q be a BCQ. $\mathsf{QIB\text{-}Entailment}(\mathcal{T}, \mathcal{P}, \mathcal{A}, q)$ is the problem of deciding whether $q \in \mathsf{qib\_cens}_{\mathcal{T},\mathcal{P}}(\mathcal{A})$.

We now focus on the case of *DL-Lite*$_\mathcal{R}$ TBoxes and prove that, in this case, entailment of BCQs under QIB censors is FO-rewritable. Formally, we say that QIB-entailment in a DL $\mathcal{L}$ is FO-rewritable, if for every TBox $\mathcal{T}$ expressed in $\mathcal{L}$, every policy $\mathcal{P}$ and every BCQ q, one can effectively compute an FO query q_r such that for every ABox $\mathcal{A}$, $\mathsf{QIB\text{-}Entailment}(\mathcal{T}, \mathcal{P}, \mathcal{A}, q)$ is true iff $\mathcal{A} \models q_r$. We call q_r the *QIB-perfect reformulation* of q w.r.t. $\mathcal{T}$ and $\mathcal{P}$.

We prove FO-rewritability of entailment of BCQs under QIB censors in *DL-Lite*$_\mathcal{R}$ by exploiting a correspondence between this problem and entailment of BCQs under IAR-semantics for DL ontologies, which is indeed FO-rewritable for *DL-Lite*$_{\mathcal{R},den}$, i.e., *DL-Lite*$_\mathcal{R}$ enriched with denial assertions [15]. We recall that the IAR-semantics is an inconsistency-tolerant semantics that allows for meaningful entailment also when the ABox contradicts the TBox of an ontology. The entailment under IAR-semantics is defined as follows: give a DL $\mathcal{T}$, an ABox $\mathcal{A}$, and BCQ q, $\mathsf{IAR\text{-}Entailment}(\mathcal{T}, \mathcal{A}, q)$ is the problem of verifying whether $\mathcal{T} \cup \mathcal{R}_{iar} \models q$, where $\mathcal{R}_{iar}$ is the intersection of all the maximal subsets $\mathcal{A}_r$ of $\mathcal{A}$ such that $\mathcal{A}_r \cup \mathcal{T}$ is consistent.

Theorem 3. *Let $\mathcal{T}$ be a DL-Lite$_\mathcal{R}$ TBox, $\mathcal{P}$ be a policy, $\mathcal{A}$ be an ABox, and q be a BCQ. QIB-Entailment$(\mathcal{T}, \mathcal{P}, \mathcal{A}, q)$ is true iff IAR-Entailment$(\mathcal{T} \cup \mathcal{P}, \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), q)$ is true.*

Theorem 3 actually states that, to solve QIB-entailment, we can resort to the query rewriting techniques used to establish IAR-entailment given in [15], provided that we compute $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$. We recall that query entailment under IAR-semantics in a DL $\mathcal{L}$ is FO-rewritable if for every TBox $\mathcal{T}$ expressed in $\mathcal{L}$ and every BCQ q, one can effectively compute an FO query q_r such

that for every ABox $\mathcal{A}$, IAR-Entailment$(\mathcal{T}, \mathcal{A}, q)$ is true iff $\mathcal{A} \models q_r$. The query q_r is called the *IAR-perfect reformulation* of q w.r.t. $\mathcal{T}$.

To establish FO-rewritability of QIB-entailment in *DL-Lite*$_\mathcal{R}$, however, we still need to address the above mentioned computation of $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$, and turn it into an additional query reformulation step. To this aim, we can exploit the fact that, for a *DL-Lite*$_{\mathcal{R},den}$ ontology $\mathcal{T} \cup \mathcal{A}$, an FO query q evaluates to true over $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ iff q' evaluates to true over $\mathcal{A}$, where q' is obtained by suitably rewriting each atom of q according to the positive inclusions of $\mathcal{T}$. Intuitively, in this way we cast into the query all the possible causes of the facts that are contained in the closure of the ABox w.r.t. the TBox.

To compute such a query q', we use the function atomRewr$(q, \mathcal{T})$, which substitutes each atom α of q with the formula $\phi(\alpha)$ defined as follows (where A, B are atomic concepts and R, S are atomic roles):

$$\phi(A(t)) = \bigvee_{\mathcal{T} \models B \sqsubseteq A} B(t) \vee \bigvee_{\mathcal{T} \models \exists R \sqsubseteq A} (\exists x.R(t,x)) \vee \bigvee_{\mathcal{T} \models \exists R^- \sqsubseteq A} (\exists x.R(x,t))$$

$$\phi(R(t_1,t_2)) = \bigvee_{\mathcal{T} \models S \sqsubseteq R} S(t_1,t_2) \vee \bigvee_{\mathcal{T} \models S^- \sqsubseteq R} S(t_2,t_1).$$

The following lemma, whose proof can be immediately obtained from the definitions of $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\cdot)$ and atomRewr$(\cdot,\cdot)$, states the property we are looking for.

Lemma 1. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R},den}$ TBox, $\mathcal{A}$ be an ABox, and q be an FO sentence. Then Eval$(q, \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})) =$ Eval(atomRewr$(q, \mathcal{T}), \mathcal{A})$.*

We are now able to extablish FO-rewritability of QIB-entailment.

Theorem 4. *Let $\mathcal{T}$ be a DL-Lite$_\mathcal{R}$ TBox, $\mathcal{P}$ be a policy, q be a BCQ, and q_r be an FO sentence that is a IAR-perfect reformulation of q w.r.t. the DL-Lite$_{\mathcal{R},den}$ TBox $\mathcal{T} \cup \mathcal{P}$. Then, the FO sentence atomRewr$(q_r, \mathcal{T})$ is a QIB-perfect reformulation of q w.r.t. $\mathcal{T}$ and $\mathcal{P}$.*

Since IAR-entailment is actually FO rewritable, as shown in [15], the above theorem proves the FO rewritability of QIB-entailment for *DL-Lite*$_\mathcal{R}$ TBoxes. Moreover, the above theorem identifies a technique for obtaining the QIB-perfect reformulation of a CQ, based on a simple combination of the IAR-perfect reformulation algorithm of [15] and the atomRewr reformulation defined above. Therefore:

Corollary 1. *Let $\mathcal{T}$ be a DL-Lite$_\mathcal{R}$ TBox, $\mathcal{P}$ be a policy, $\mathcal{A}$ be an ABox, and q be a BCQ. The problem QIB-Entailment$(\mathcal{T}, \mathcal{P}, \mathcal{A}, q)$ is in* AC^0 *in data complexity.*

6. Conclusions

In this paper, we have studied the approach to CQE based on instance indistinguishability and identified a semantically well-founded notion of CQE that enjoys first-order rewritability in the case of *DL-Lite*$_\mathcal{R}$ ontologies.

An important future research direction is a deeper study of the user model. Our framework inherits from its predecessors a relatively simple model, which assumes that the user knows (at most) the TBox and all the query answers returned by the system, and considers only the *deductive* abilities of the user over such knowledge. This user model might need to be enriched to capture more realistic data protection scenarios.

Acknowledgments

This work was partly supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014), by MIUR under the PRIN 2017 project "HOPE" (prot. 2017MMJJRE), by the EU within the H2020 Programme under the grant agreement 834228 (ERC Advanced Grant WhiteMec) and the grant agreement 825333 (MOSAICrOWN), by Regione Lombardia within the Call Hub Ricerca e Innovazione under the grant agreement 1175328 (WATCHMAN), and by Sapienza Università di Roma (2019 project CQEinOBDM).

References

[1] G. L. Sicherman, W. de Jonge, R. P. van de Riet, Answering queries without revealing secrets, ACM Trans. Database Syst. 8 (1983) 41–59.

[2] P. A. Bonatti, S. Kraus, V. S. Subrahmanian, Foundations of secure deductive databases, IEEE Trans. Knowl. Data Eng. 7 (1995) 406–422.

[3] J. Biskup, For unknown secrecies refusal is better than lying, Data and Knowl. Eng. 33 (2000) 1–23.

[4] J. Biskup, P. A. Bonatti, Controlled query evaluation for known policies by combining lying and refusal, Ann. Math. Artif. Intell. 40 (2004) 37–62.

[5] J. Biskup, T. Weibert, Keeping secrets in incomplete databases, Int. J. of Information Security 7 (2008) 199–217.

[6] P. A. Bonatti, L. Sauro, A confidentiality model for ontologies, in: Proc. of ISWC, volume 8218 of *LNCS*, 2013, pp. 17–32.

[7] B. Cuenca Grau, E. Kharlamov, E. V. Kostylev, D. Zheleznyakov, Controlled query evaluation over OWL 2 RL ontologies, in: Proc. of ISWC, volume 8218, 2013, pp. 49–65.

[8] B. Cuenca Grau, E. Kharlamov, E. V. Kostylev, D. Zheleznyakov, Controlled query evaluation for datalog and OWL 2 profile ontologies, in: Proc. of IJCAI, 2015, pp. 2883–2889.

[9] D. Lembo, R. Rosati, D. F. Savo, Revisiting controlled query evaluation in description logics, in: Proc. of IJCAI, 2019, pp. 1786–1792.

[10] G. Cima, D. Lembo, L. Marconi, R. Rosati, D. F. Savo, Controlled query evaluation in ontology-based data access, in: Proc. of ISWC, volume 12506 of *LNCS*, 2020, pp. 128–146.

[11] G. Cima, D. Lembo, R. Rosati, D. F. Savo, Controlled query evaluation in description logics through instance indistinguishability, in: Proc. of IJCAI, 2020, pp. 1791–1797.

[12] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family, J. of Automated Reasoning 39 (2007) 385–429.

[13] M. Benedikt, B. Cuenca Grau, E. V. Kostylev, Logical foundations of information disclosure in ontology-based data integration, Artif. Intell. J. 262 (2018) 52–95.

[14] M. Benedikt, P. Bourhis, L. Jachiet, M. Thomazo, Reasoning about disclosure in data integration in the presence of source constraints, in: Proc. of IJCAI, 2019, pp. 1551–1557.

[15] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant query answering in ontology-based data access, J. of Web Semantics 33 (2015) 3–29.