

A Novel Approach to Controlled Query Evaluation in *DL-Lite* (DISCUSSION PAPER)

Domenico Lembo¹, Riccardo Rosati¹, Domenico Fabio Savo²

¹ Sapienza Università di Roma
lastname@dis.uniroma1.it

² Università degli Studi di Bergamo
domenicofabio.savo@unibg.it

Abstract. In Controlled Query Evaluation (CQE) confidential data are protected through a declarative *policy* and a (optimal) *censor*, which guarantees that answers to queries are maximized without disclosing secrets. In this paper we consider CQE over Description Logic ontologies and study *query answering* over *all* optimal censors. We establish data complexity of the problem for ontologies specified in *DL-Lite_R* and for variants of the censor language, which is the language used by the censor to enforce the policy. In our investigation we also analyze the relationship between CQE and the problem of Consistent Query Answering.

1 Introduction

In Controlled Query Evaluation (CQE), a policy, i.e., a set of logical assertions, regulates the access to a database or knowledge base by specifying the information that must be kept secret, and a *censor* alters answers to queries so that confidential data cannot be inferred by the users on the basis of the queries they ask. The notion of censor traces back to [16], and since then it has been investigated for propositional closed databases [6], incomplete databases [7], and, more recently, Description Logic (DL) ontologies [10,11]. In this latter context, *optimal* censors are defined as those censors that modify query answers in a “minimal” way. Intuitively, such censors hide data to preserve confidentiality according to the policy, without restricting unnecessarily the ability of the system to return answers to users’ queries.

Previous work on CQE in DLs has mainly focused on the tasks of establishing the *existence* of an optimal censor and characterizing the complexity of computing it. However, considering only one such censor means making an arbitrary selection among several optimal censors. To avoid such a discretionary choice, in this paper we adopt a different approach and study *query answering over all optimal censors*. Intuitively, given a query q , this amounts to compute the answers to q that are returned by all optimal censors. This form of reasoning can also be considered as the application of a single

Copyright © 2019 for the individual papers by the papers’ authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors. SEBD 2019, June 16-19, 2019, Castiglione della Pescaia, Italy.

censor corresponding to an “intersection” of all the optimal censors, which is thus a semantically well-founded (i.e., sound) approximation of any optimal censor. This idea has been also previously discussed in [10].

Our approach has similarities with the work on Consistent Query Answering (CQA), a framework for inconsistency management based on the notion of repair [3,5]. Roughly speaking, in DL, a repair of a possibly inconsistent ontology \mathcal{O} is any ABox for \mathcal{O} (i.e., the extensional component of \mathcal{Q}) that is consistent with the TBox (i.e., the intensional component of \mathcal{O}), and that differs “minimally” from the original ABox. Then, computing query answers in CQA amounts to reasoning over all repairs and the TBox. The connection between CQE and CQA in DL is based on the intuition that the assertions in the policy in CQE seems to act as the class of assertions in \mathcal{T} that may be violated by the data of the ABox.

Some connections between CQA and a declarative approach for privacy preservation had already been discussed in [4]. The framework in that paper is similar to ours, with so-called secrecy views playing essentially the role of the policy. However, the setting considered there is relational and without intensional knowledge (TBox), and secrecy views are enforced through suitable virtual modifications of database values with SQL NULLs, so that this approach is incomparable with ours. Nonetheless, in this paper we elaborate on the intuition of [4] and investigate in depth the relationship between our CQE framework and CQA in DLs. We provide some general conditions ensuring that the two problems are mutually reducible, and we show cases of practical interest for which such conditions are satisfied and cases for which they are not. This also allows us to highlight the differences between the two frameworks.

The ultimate goal of this paper is to investigate data complexity of answering conjunctive queries (CQs) in CQE. In our analysis we consider ontologies specified in $DL\text{-}Lite_R$ [9]. We also consider some variants of the *censor language* \mathcal{L}_C . Intuitively, \mathcal{L}_C is the language in which the censor expresses the sentences implied by the ontology that can be disclosed to the users without violating the policy. We provide data complexity results for the cases when: (i) \mathcal{L}_C is the ABox of the ontology; (ii) \mathcal{L}_C coincides with the set of ground atoms expressed over the signature of the ontology; and (iii) \mathcal{L}_C is the language of CQs expressed over the signature of the ontology. Some of the complexity results follow from the correspondence between CQA and CQE; we provide novel techniques for the cases in which the CQE problem does not have a CQA counterpart. The complexity results proved in this paper are shown in Figure 1.

This paper is an extended abstract of [15], where also complexity results for ontologies specified in \mathcal{EL}_{\perp} [1] are provided.

2 Preliminaries

We consider a signature Σ of predicates and constants, and a countably infinite alphabet of variables \mathcal{V} . To simplify the presentation, we consider only languages containing FO sentences, i.e., formulas without free variables (our results applies to open formulas as well, modulo standard encoding of open formulas into closed ones). We use **FO** to indicate the language of all function-free FO sentences over Σ and \mathcal{V} . Every language considered in this paper is a subset of **FO**.

Given a set $\mathcal{K} \subseteq \text{FO}$, $Mod(\mathcal{K})$ indicates the set of models of \mathcal{K} , i.e., the FO interpretations \mathcal{I} such that $\phi^{\mathcal{I}}$ (i.e., the interpretation of ϕ in \mathcal{I}) evaluates to true, for each sentence $\phi \in \mathcal{K}$. If \mathcal{I} is a model of \mathcal{K} , we say that \mathcal{I} satisfies \mathcal{K} and write $\mathcal{I} \models \mathcal{K}$. \mathcal{K} is consistent if it has at least one model, i.e., if $Mod(\mathcal{K}) \neq \emptyset$, inconsistent otherwise. \mathcal{K} entails a FO sentence ϕ , denoted $\mathcal{K} \models \phi$, if $\phi^{\mathcal{I}}$ is true in every $\mathcal{I} \in Mod(\mathcal{K})$.

A Boolean conjunctive query (BCQ) q is a FO sentence of the form $\exists \mathbf{x}.conj(\mathbf{x})$, where $conj(\mathbf{x}) = \alpha_1(\mathbf{x}) \wedge \dots \wedge \alpha_n(\mathbf{x})$, \mathbf{x} is a sequence of variables, and each $\alpha_i(\mathbf{x})$ is an atom (possibly with constants) with predicate α_i and variables in \mathbf{x} . The *length* of a BCQ q is the number of its atoms, denoted by $length(q)$.

In the following, **CQ** denotes the language of BCQs over Σ and \mathcal{V} , **CQ_k** the language of BCQs from **CQ** whose maximum length is k , and **GA** the language of ground atoms. Obviously, for every integer k , **GA** \subset **CQ_k** \subset **CQ**. Verifying whether $\mathcal{K} \models \alpha$ for $\mathcal{K} \subseteq \text{FO}$ and $\alpha \in \text{GA}$ is also called *instance checking*.

A DL ontology \mathcal{O} is specified as $\mathcal{T} \cup \mathcal{A}$, where \mathcal{T} is the *TBox* and \mathcal{A} is the *ABox* [2]. Throughout the paper an ABox is always a set of ground atoms. We are interested in *DL-Lite_R* DL ontologies [9]. A *DL-Lite_R* TBox is a finite set of assertions of the form: $B_1 \sqsubseteq B_2$, $B_1 \sqsubseteq \neg B_2$, $R_1 \sqsubseteq R_2$, $R_1 \sqsubseteq \neg R_2$, where: each R_i , with $i \in \{1, 2\}$, is an *atomic role* $Q \in \Sigma$, or its inverse Q^- ; each B_i , with $i \in \{1, 2\}$, is an *atomic concept* $A \in \Sigma$, or a concept of the form $\exists Q$ (resp. $\exists Q^-$), i.e., unqualified existential restriction, which denotes the set of objects occurring as first (resp. second) argument of Q .

We also consider *denial assertions* (or simply *denials*) over concepts and roles, i.e., sentences of the form: $\forall \mathbf{x}.conj(\mathbf{x}) \rightarrow \perp$ where $conj(\mathbf{x})$ is such that $\exists \mathbf{x}.conj(\mathbf{x})$ is a BCQ whose atoms use only unary and binary predicates. The *length* of the denial is the length of such query. A denial is satisfied by an ontology \mathcal{O} if $\mathcal{O} \not\models \exists \mathbf{x}.conj(\mathbf{x})$.

In the following, given an ontology \mathcal{O} and a language $\mathcal{L} \subseteq \text{FO}$, we denote with $\mathcal{L}(\mathcal{O})$ the subset of formulas of \mathcal{L} over the predicates and constants occurring in \mathcal{O} .

All the complexity results we provide refer to *data complexity*, that in our framework is the complexity computed only with respect to the size of the ontology ABox.

3 CQE Framework

Our CQE framework is adapted from [11]. An \mathcal{L} *CQE instance* \mathcal{E} is a quadruple $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle$, where \mathcal{T} is a TBox in the DL \mathcal{L} , \mathcal{A} is an ABox such that $\mathcal{T} \cup \mathcal{A}$ is consistent, \mathcal{P} is the policy, i.e., a set of denial assertions over the signature of $\mathcal{T} \cup \mathcal{A}$, such that $\mathcal{T} \cup \mathcal{P}$ is consistent, and $\mathcal{L}_C \subseteq \text{FO}(\mathcal{T} \cup \mathcal{A})$ is the *censor language*. Intuitively: \mathcal{T} is the schema a user interacts with to pose her queries; \mathcal{A} is the dataset underlying the schema; \mathcal{P} specifies the knowledge that cannot be disclosed for confidentiality reasons, in the sense that the user will never get, through query answers, sufficient knowledge to violate the denials in \mathcal{P} ; and \mathcal{L}_C is the language with respect to which the censor is specified, that is, the censor establishes which are the sentences in \mathcal{L}_C implied by $\mathcal{T} \cup \mathcal{A}$ that can be divulged to the user while preserving the policy (cf. Definition 1). To simplify the notation, we will sometimes omit to specify that a certain censor language is limited to the signature of $\mathcal{T} \cup \mathcal{A}$ (e.g., we will use **GA** instead of **GA**($\mathcal{T} \cup \mathcal{A}$)).

We then define a censor in terms of its underlying theory.

Definition 1. The theory Th_{cens} of a censor cens for a CQE instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle$ is a subset of \mathcal{L}_C such that: (i) $\mathcal{T} \cup \mathcal{A} \models \phi$, for each $\phi \in \text{Th}_{\text{cens}}$, and (ii) $\mathcal{T} \cup \mathcal{P} \cup \text{Th}_{\text{cens}}$ is consistent.

A censor c is *optimal* if there is no censor c' such that $\text{Th}_c \subset \text{Th}_{c'} \subseteq \mathcal{L}_C$. The set of theories of all the optimal censors of a CQE instance \mathcal{E} is denoted by $\text{Th}_{\text{oc}}\text{-Set}(\mathcal{E})$.

Example 1. A CQE instance $\mathcal{E} = (\mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ})$ is used by an international humanitarian organization for regulating the access to the information about its volunteer staff. In \mathcal{E} , \mathcal{T} is an empty TBox, $\mathcal{A} = \{\text{workedIn}(v_1, c_A), \text{workedIn}(v_1, c_B), \text{workedIn}(v_2, c_A)\}$, and $\mathcal{P} = \{\forall x.\text{workedIn}(x, c_A) \wedge \text{workedIn}(x, c_B) \rightarrow \perp\}$. In words, the policy specifies as confidential the fact that a volunteer worked in both country A and country B, given that these countries are currently at war with each other.

Below we provide the definition of entailment in CQE¹.

Definition 2. (CQE-entailment) Given a CQE instance \mathcal{E} and a FO sentence ϕ , decide whether $\text{Th} \models \phi$ for every $\text{Th} \in \text{Th}_{\text{oc}}\text{-Set}(\mathcal{E})$. If this is the case, we write $\mathcal{E} \models_{\text{CQE}} \phi$.

As usual, when the language of ϕ is restricted to ground atoms (i.e., ABox assertions), CQE-entailment is called (CQE)-instance checking.

Example 2. For the CQE instance \mathcal{E} of Example 1, we have, for instance, that $\mathcal{E} \models_{\text{CQE}} \exists x.\text{workedIn}(v_1, x)$ and $\mathcal{E} \models_{\text{CQE}} \exists x.\text{workedIn}(x, c_B)$, but $\mathcal{E} \not\models_{\text{CQE}} \text{workedIn}(v_1, c_B)$.

4 Relationship between CQE and CQA

In this section we discuss the relationship between the CQE framework we have just defined and CQA. To this aim, we first provide a general definition for CQA.

A \mathcal{L} CQA instance \mathcal{J} is a triple $\langle \mathcal{T}, \mathcal{A}, \mathcal{L}_R \rangle$ where \mathcal{T} is a consistent TBox in the DL \mathcal{L} , \mathcal{A} is an ABox, and $\mathcal{L}_R \subseteq \text{FO}(\mathcal{T} \cup \mathcal{A})$ is the *repair language*. The *consistent entailment set in a language \mathcal{L}* of a (possibly inconsistent) theory $\mathcal{T} \cup \mathcal{A}$, denoted by $CES(\mathcal{T}, \mathcal{A}, \mathcal{L})$, is the set $\{\phi \mid \phi \in \mathcal{L} \text{ and there exists a } \mathcal{A}' \subseteq \mathcal{A} \text{ such that } \mathcal{T} \cup \mathcal{A}' \text{ is consistent and } \mathcal{T} \cup \mathcal{A}' \models \phi\}$. A repair for a CQA instance is defined as follows.

Definition 3. A repair \mathcal{R} for a CQA instance $\mathcal{J} = \langle \mathcal{T}, \mathcal{A}, \mathcal{L}_R \rangle$ is a subset of \mathcal{L}_R such that: (i) $\mathcal{R} \subseteq CES(\mathcal{T}, \mathcal{A}, \mathcal{L}_R)$ and; (ii) $\mathcal{T} \cup \mathcal{R}$ is consistent and; (iii) there does not exist any \mathcal{R}' such that $\mathcal{R} \subset \mathcal{R}' \subseteq CES(\mathcal{T}, \mathcal{A}, \mathcal{L}_R)$ and $\mathcal{T} \cup \mathcal{R}'$ is consistent. We denote by $\text{RepSet}(\mathcal{J})$ the set of repairs of \mathcal{J} .

Definition 3 captures some notions of repair proposed in the literature, such as the repair at the basis of the prototypical AR-semantics, or the repair adopted by the CAR-semantics [13,14]. Indeed, given an ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$, repairs in the AR-semantics aim to preserve as many facts as possible of those in \mathcal{A} . This means that, in a CQA instance adopting the AR-semantics, the language \mathcal{L}_R has to be set to \mathcal{A} . Differently, the CAR-semantics aims to preserve as many facts as possible of those implied by \mathcal{T} and

¹ Due to space limits, we give a simplified definition and refer to [15] for other formal details.

each subset of \mathcal{A} consistent with \mathcal{T} . Thus, to encode such semantics $\mathcal{L}_{\mathcal{R}}$ has to coincide with the set $\mathbf{GA}(\mathcal{O})$ of ground atoms over the predicates and constants in \mathcal{O} .

We now provide some conditions on CQE and CQA instances that allow to establish correspondences between (theories of) censors and repairs.

Definition 4. A CQE instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle$ is CQA-reducible if:

- (i) for every $\phi \in \mathcal{L}_C$ such that $\mathcal{T} \cup \mathcal{A} \models \phi$ and $\{\phi\} \cup \mathcal{T} \cup \mathcal{P}$ is consistent, there exists $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}' \cup \mathcal{P}$ is consistent and $\mathcal{T} \cup \mathcal{A}' \models \phi$;
- (ii) for every $\phi \in \mathcal{L}_C$ and every $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}' \cup \mathcal{P}$ is consistent, if $\mathcal{T} \cup \mathcal{A}' \cup \mathcal{P} \models \phi$ then $\mathcal{T} \cup \mathcal{A}' \models \phi$.

In words, condition (i) imposes that every logical consequence of $\mathcal{T} \cup \mathcal{A}$ that is consistent with the policy and the TBox belongs to $CES(\mathcal{T} \cup \mathcal{P}, \mathcal{A}, \mathcal{L}_C)$. Condition (ii) instead says that in a CQA-reducible instance the sentences in the policy act as constraints on top of $\mathcal{T} \cup \mathcal{A}$, since they never contribute to infer new formulas from \mathcal{L}_C if added to $\mathcal{T} \cup \mathcal{A}$.

Example 3. The instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle$ with $\mathcal{T} = \{A \sqsubseteq B\}$, $\mathcal{A} = \{A(d)\}$, $\mathcal{P} = \{\forall x.A(x) \rightarrow \perp\}$, and $\mathcal{L}_C = \mathbf{GA}$ is not CQA-reducible, since it does not respect condition (i), even though it satisfies condition (ii) (in a trivial way). Instead, $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}', \mathcal{P}', \mathcal{L}_C \rangle$ with $\mathcal{A}' = \{A(d), B(d)\}$, $\mathcal{P} = \{\forall x.A(x) \wedge B(x) \rightarrow \perp\}$, and \mathcal{T} and \mathcal{L}_C as before is CQA-reducible.

Theorem 1. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle$ be a CQE instance, such that \mathcal{E} is a CQA-reducible. Then $\text{Th}_{oc}\text{-Set}(\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle) = \text{RepSet}(\langle \mathcal{T} \cup \mathcal{P}, \mathcal{A}, \mathcal{L}_C \rangle)$.

Below we consider reducibility of CQA instances into CQE ones.

Definition 5. A CQA instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{L}_{\mathcal{R}} \rangle$ is CQE-reducible if there exists a partition $\mathcal{T}_P \cup \mathcal{T}_N$ of \mathcal{T} such that $\mathcal{T}_P \cup \mathcal{A}$ is consistent, \mathcal{T}_N is equivalent to a set of denials, and:

- (i) for every $\phi \in \mathcal{L}_{\mathcal{R}}$, such that $\mathcal{T}_P \cup \mathcal{A} \models \phi$ and $\{\phi\} \cup \mathcal{T}$ is consistent, there exists $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}'$ is consistent and $\mathcal{T} \cup \mathcal{A}' \models \phi$;
- (ii) for every $\phi \in \mathcal{L}_{\mathcal{R}}$ and every $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}'$ is consistent, if $\mathcal{T} \cup \mathcal{A}' \models \phi$ then $\mathcal{T}_P \cup \mathcal{A}' \models \phi$.

Intuitively, the above definition says that in a CQE-reducible instance we can identify a portion \mathcal{T}_N of \mathcal{T} such that its assertions act as constraints on the ontology $\mathcal{T}_P \cup \mathcal{A}$ (cond. (ii)), thus \mathcal{T}_N behaves as a policy in a CQE instance. At the same time, each logical consequence of $\mathcal{T}_P \cup \mathcal{A}$ consistent with \mathcal{T} must belong to $CES(\mathcal{T}, \mathcal{A}, \mathcal{L}_{\mathcal{R}})$ (cond. (i)). CQE-reducible instances have the following property.

Theorem 2. Let $\mathcal{J} = \langle \mathcal{T}, \mathcal{A}, \mathcal{L}_{\mathcal{R}} \rangle$ be a CQA instance, such that \mathcal{J} is CQE-reducible with $\mathcal{T} = \mathcal{T}_P \cup \mathcal{T}_N$. Then $\text{RepSet}(\langle \mathcal{T}_P \cup \mathcal{T}_N, \mathcal{A}, \mathcal{L}_{\mathcal{R}} \rangle) = \text{Th}_{oc}\text{-Set}(\langle \mathcal{T}_P, \mathcal{A}, \mathcal{T}_N, \mathcal{L}_{\mathcal{R}} \rangle)$.

We now rephrase entailment in CQA [14]: given a CQA instance $\mathcal{J} = \langle \mathcal{T}, \mathcal{A}, \mathcal{L}_{\mathcal{R}} \rangle$ and a FO sentence ϕ , decide whether $\mathcal{T} \cup \mathcal{R} \models \phi$ for every $\mathcal{R} \in \text{RepSet}(\mathcal{J})$. This is denoted by $\mathcal{J} \models_{CQA} \phi$. The following results follow from Theorem 1 and Theorem 2.

Corollary 1. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_C \rangle$ be a CQA-reducible CQE instance and ϕ a FO sentence. Then, $\mathcal{E} \models_{CQE} \phi$ iff $\mathcal{J} \models_{CQA} \phi$, where $\mathcal{J} = \langle \mathcal{T} \cup \mathcal{P}, \mathcal{A}, \mathcal{L}_C \rangle$. Furthermore, Let $\mathcal{J} = \langle \mathcal{T}, \mathcal{A}, \mathcal{L}_{\mathcal{R}} \rangle$ be a CQE-reducible CQA instance with $\mathcal{T} = \mathcal{T}_P \cup \mathcal{T}_N$, and ϕ a FO sentence. Then, $\mathcal{J} \models_{CQA} \phi$ iff $\mathcal{E} \models_{CQE} \phi$, where $\mathcal{E} = \langle \mathcal{T}_P, \mathcal{A}, \mathcal{T}_N, \mathcal{L}_{\mathcal{R}} \rangle$.

5 CQE under Restricted Censor Languages

In this section we establish data complexity of instance checking and CQ entailment for $DL\text{-}Lite_R$ CQE instances, when the censor language \mathcal{L}_C coincides with the ABox \mathcal{A} , and with the set of ground atoms \mathbf{GA} . For the former case, we establish our complexity results by exploiting a mutual reduction between entailment in CQE and CQA. For the latter case, the two frameworks behave in a slightly different way, and thus we also need to use techniques tailored to the CQE setting.

We start with $\mathcal{L}_C = \mathcal{A}$, and show that CQE instances are CQA-reducible, but also that CQA-instances are CQE-reducible when the repair language coincides with \mathcal{A} .

Theorem 3. *Each $DL\text{-}Lite_R$ CQE instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{A} \rangle$ is CQA-reducible, and each $DL\text{-}Lite_R$ CQA instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{A} \rangle$ is CQE-reducible.*

The following result follows from Theorem 3 and the fact that CQ entailment in $DL\text{-}Lite_{R,den}$ CQA instances under AR-semantics is coNP-complete, already for instance checking [14].

Theorem 4. *Both instance checking and CQ entailment are coNP-complete in data complexity for $DL\text{-}Lite_R$ CQE instances $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{A} \rangle$.*

We now consider $\mathcal{L}_C = \mathbf{GA}$. In this case, $DL\text{-}Lite_R$ CQE instances are *not* always CQA-reducible, as shown in Example 3. Reducibility in the other way round is also not always possible. However, we can show some weaker, but useful, properties.

Proposition 1. *Each $DL\text{-}Lite_R$ CQE instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{GA} \rangle$, such that $\mathcal{T} \cup \mathcal{P} \cup \{\alpha\}$ is satisfiable for each $\alpha \in \mathcal{A}$, is CQA-reducible. Also, each $DL\text{-}Lite_R$ CQA instance $\langle \mathcal{T}, \mathcal{A}, \mathbf{GA} \rangle$, such that $\mathcal{T} \cup \{\alpha\}$ is satisfiable for each $\alpha \in \mathcal{A}$, is CQE-reducible.*

For $DL\text{-}Lite_R$ CQE instances satisfying the conditions mentioned in Proposition 1 we can establish computational complexity of query answering by mutual reduction between CQE and CQA under CAR-semantics, similarly to what we have done to prove Theorem 4. In fact, we are able to exactly characterize the complexity for general $DL\text{-}Lite_R$ CQE instances by using tailored proofs, which are inspired to those used in CQA to establish both upper and lower complexity bounds.

Theorem 5. *Instance checking and CQ entailment are respectively in AC^0 and coNP-complete in data complexity, for $DL\text{-}Lite_R$ CQE instances $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{GA} \rangle$.*

6 CQE under Full Censor Language

In this section we study CQE-entailment when the censor language \mathcal{L}_C is \mathbf{CQ} . We start with the following property that is central for our analysis.

Theorem 6. *Let \mathcal{T} be a $DL\text{-}Lite_R$ TBox, let \mathcal{A} be an ABox such that $\mathcal{T} \cup \mathcal{A}$ is consistent, let \mathcal{P} be a policy, let q be a BCQ, and let $k = \max(h, \text{length}(q))$, where h is the maximum length of a denial assertion in \mathcal{P} . Then, $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ} \rangle \models_{CQE} q$ iff $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ}_k \rangle \models_{CQE} q$.*

	$\mathcal{L}_c = \mathcal{A}$	$\mathcal{L}_c = \mathbf{GA}$	$\mathcal{L}_c = \mathbf{CQ}$
<i>Instance Checking</i>	coNP-complete	in AC ⁰	in PTIME
<i>CQ Entailment</i>	coNP-complete	coNP-complete	in PTIME

Fig. 1: Data complexity of entailment over $DL\text{-}Lite_R$ CQE instances $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathcal{L}_c \rangle$

In the rest of the section, without loss of generality, we assume that, in every CQE instance of the form $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ}_k \rangle$, all formulas of the language \mathbf{CQ}_k (as well as the query q of the CQE-entailment problem) use the set of $2k$ variables $\{x_1, \dots, x_{2k}\}$.

We now define the *CQE-Ent-DL-Lite_R* algorithm for deciding CQE-entailment of BCQs for $DL\text{-}Lite_R$ KBs.

Algorithm *CQE-Ent-DL-Lite_R*($\mathcal{T}, \mathcal{A}, \mathcal{P}, q$)
Input: $DL\text{-}Lite_R$ TBox \mathcal{T} , ABox \mathcal{A} s. t. $\mathcal{T} \cup \mathcal{A}$ is consistent, policy \mathcal{P} , BCQ q
Output: true if $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ} \rangle \models_{CQE} q$, false otherwise
let h be the maximum length of a denial in \mathcal{P} ;
let $k = \max(h, \text{length}(q))$;
 $\Phi = CQEntailedSubset(\mathcal{T}, \mathcal{A}, k)$;
for $i = 1$ to k **do**
 remove from Φ every subset Φ' such that $|\Phi'| = i$
 and $\mathcal{T} \cup \mathcal{P} \cup \Phi'$ is inconsistent;
if $q \in \Phi$ **then** return true **else** return false

In the algorithm, *CQEntailedSubset*($\mathcal{T}, \mathcal{A}, k$) is the function returning the set of BCQs from \mathbf{CQ}_k that are entailed by $\mathcal{T} \cup \mathcal{A}$.

Informally, the algorithm first computes an integer k , based on the length of q and of the denials in \mathcal{P} ; then, it computes the set Φ that represents the intersection of the theories of the optimal censors for the CQE instance $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ}_k \rangle$: this is done by eliminating from Φ all formulas that belong to minimal subsets of Φ that are inconsistent with $\mathcal{T} \cup \mathcal{P}$; finally, it checks the presence of q among the queries in the above set Φ .

Theorem 7. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ} \rangle$ be a $DL\text{-}Lite_R$ CQE instance, and q a BCQ. Then, $\mathcal{E} \models_{CQE} q$ iff *CQE-Ent-DL-Lite_R*($\mathcal{T}, \mathcal{A}, \mathcal{P}, q$) returns true.

Algorithm *CQE-Ent-DL-Lite_R*($\mathcal{T}, \mathcal{A}, \mathcal{P}, q$) runs in PTIME in the size of the ABox, since *CQEntailedSubset*($\mathcal{T}, \mathcal{A}, k$) can be computed in polynomial time in the size of \mathcal{A} and checking the consistency of $\mathcal{T} \cup \mathcal{P} \cup \Phi'$ can be reduced to checking the consistency of a $DL\text{-}Lite_{R,den}$ ontology, which is polynomial in data complexity [14].

Theorem 8. Entailment of CQs is in PTIME in data complexity for $DL\text{-}Lite_R$ CQE instances $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \mathbf{CQ} \rangle$.

7 Conclusions

Our results (cf. Figure 1) show a surprising aspect: the complexity of CQ entailment for restricted censor languages is harder than when $\mathcal{L}_c = \mathbf{CQ}$. Indeed, in this latter case,

CQ entailment can be established through the computation of the intersection of (a finite and polynomial representation of) all the theories of optimal censors, which can be done in polynomial time. This does not hold for \mathcal{L}_C equal to \mathcal{A} or **GA**.

Our research work can be extended in many directions. First, the PTIME upper bound for CQE over $DL\text{-}Lite_R$ TBoxes and **CQ** censor language should be refined. We believe that an AC⁰ bound can be shown in this case. Then, the complexity analysis of CQE could be extended to other DLs, as well as to other policy and censor languages. Also, based on the complexity analysis presented in this paper, it would be important to look for practical techniques allowing for the implementation of CQE extensions of current DL reasoners and Ontology-based Data Access systems [8,12].

References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope. In *Proc. of IJCAI*, pages 364–369, 2005.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, 2007.
3. L. E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.
4. L. E. Bertossi and L. Li. Achieving data privacy through secrecy views and null-based virtual updates. *IEEE Trans. Knowl. Data Eng.*, 25(5):987–1000, 2013.
5. M. Bienvenu and C. Bourgau. Inconsistency-tolerant querying of description logic knowledge bases. In *RW Tutorial Lectures*, pages 156–202, 2016.
6. J. Biskup and P. A. Bonatti. Controlled query evaluation for enforcing confidentiality in complete information systems. *Int. J. Inf. Sec.*, 3(1):14–27, 2004.
7. J. Biskup and T. Weibert. Keeping secrets in incomplete databases. *Int. J. Inf. Sec.*, 7(3):199–217, 2008.
8. 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 J.*, 8(3):471–487, 2017.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The $DL\text{-}Lite$ family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
10. B. Cuenca Grau, E. Kharlamov, E. V. Kostylev, and D. Zheleznyakov. Controlled query evaluation over OWL 2 RL ontologies. In *Proc. of ISWC*, pages 49–65, 2013.
11. B. Cuenca Grau, E. Kharlamov, E. V. Kostylev, and D. Zheleznyakov. Controlled query evaluation for Datalog and OWL 2 profile ontologies. In *Proc. of IJCAI*, pages 2883–2889, 2015.
12. G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, R. Rosati, M. Ruzzi, and D. F. Savo. MASTRO: A reasoner for effective Ontology-Based Data Access. In *Proc. of ORE*, 2012.
13. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of RR*, pages 103–117, 2010.
14. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant query answering in ontology-based data access. *J. of Web Semantics*, 33:3–29, 2015.
15. D. Lembo, R. Rosati, and D. F. Savo. Revisiting controlled query evaluation in Description Logics. In *Proc. of IJCAI*, 2019. To appear.
16. G. L. Sicerman, W. de Jonge, and R. P. van de Riet. Answering queries without revealing secrets. *ACM Trans. Database Syst.*, 8(1):41–59, 1983.