Controlled Query Evaluation over Prioritized Ontologies with Expressive Data Protection Policies*

Gianluca Cima[1][0000-0003-1783-5605], Domenico Lembo[2][0000-0002-0628-242X], Lorenzo Marconi[2][0000-0001-9633-8476], Riccardo Rosati[2][0000-0002-7697-4958], and Domenico Fabio Savo[3][0000-0002-8391-8049]

[1] University of Bordeaux, CNRS, Bordeaux INP, LaBRI
gianluca.cima@u-bordeoux.fr
[2] Sapienza Università di Roma
{lembo,marconi,rosati}@diag.uniroma1.it
[3] Università degli Studi di Bergamo
domenicofabio.savo@unibg.it

Abstract. We study information disclosure in Description Logic ontologies, in the spirit of Controlled Query Evaluation, where query answering is filtered through optimal censors maximizing answers while hiding data protected by a declarative policy. Previous works have considered limited forms of policy, typically constituted by conjunctive queries (CQs), whose answer must never be inferred by a user. Also, existing implementations adopt approximated notions of censors that might result too restrictive in the practice in terms of the amount of non-protected information returned to the users. In this paper we enrich the framework, by extending CQs in the policy with comparison predicates and introducing preferences between ontology predicates, which can be exploited to decide the portion of a secret that can be disclosed to a user, thus in principle augmenting the throughput of query answers. We show that answering CQs in our framework is first-order rewritable for *DL-Lite*$_A$ ontologies and *safe* policies, and thus in AC^0 in data complexity. We also present some experiments on a popular benchmark, showing effectiveness and feasibility of our approach in a real-world scenario.

1 Introduction

In this paper, we study how to manage disclosure of sensitive information in Description Logic (DL) ontologies. This problem has been recently addressed in knowledge-based systems through Controlled Query Evaluation (CQE) [12,7,9,5,8], a declarative approach to data confidentiality preservation, originally investigated in the context of databases [14,4]. In a nutshell, CQE over ontologies involves specifying a data-protection policy as a set of queries whose answer must never be inferred by a user who

* This work was partly supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014), 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 the Italian MUR (Ministero dell'Università e della Ricerca) through the PRIN project HOPE (prot. 2017MMJJRE), and by Sapienza (project CQEinOBDM).

is able to make standard reasoning and query answering over the ontology. To enforce privacy preservation, query answering is altered by a function called censor. Intuitively, optimal censors maximize answers to queries still guaranteeing that disclosed information cannot lead to answer queries in the policy.

Among various approaches, the one proposed in [6] has been shown to be particularly interesting from the practical point of view, since it allows for an effective reduction of conjunctive query answering under censors over *DL-Lite*$_{\mathcal{R}}$ ontologies to standard processing of conjunctive queries in Ontology-based Data Access (OBDA), where mappings connecting the ontology to a source database can filter the data acting as a censor. This approach is based on the notion of *IGA censor*. Intuitively, the IGA censor protects data by disclosing to the users the intersection of all inclusion-maximal subsets of the ground facts that are inferred by the ontology and that do not violate the policy (such subsets are returned by so-called optimal GA censors [6,12]).

Example 1. Assume that an oil company wants to keep information on unproductive wildcat drilling confidential, since it does not want to disclose data about the failure of this high-risk exploration activity in new areas outside of known extraction fields[1]. Thus, no answer has to be returned to the query $\exists x.\mathsf{emptyWell}(x) \wedge \mathsf{type}(x, \text{`}\textit{wildcat}\text{'})$. Assume also that the terminological component of the ontology, i.e., the TBox, says that each empty well is a wellbore and it is maintained by someone (e.g., a sub-unit of the company), and also that everything having a type is maintained by someone (i.e., $\mathsf{emptyWell} \sqsubseteq \mathsf{wellbore}$, $\mathsf{emptyWell} \sqsubseteq \exists\mathsf{maintainedBy}$, $\exists\mathsf{type} \sqsubseteq \exists\mathsf{maintainedBy}$, in DL formulas), and consider an ontology ABox containing the facts $\mathsf{emptyWell}(e)$ and $\mathsf{type}(e, \text{`}\textit{wildcat}\text{'})$. Two optimal GA censors exist, one exposing to users the facts $\{\mathsf{emptyWell}(e), \mathsf{wellbore}(e)\}$, and the other accounting for $\{\mathsf{type}(e, \text{`}\textit{wildcat}\text{'}), \mathsf{wellbore}(e)\}$ (note that $\mathsf{wellbore}(e)$ is implied by the ontology). Thus, the IGA censor returns only the fact $\mathsf{wellbore}(e)$. □

The use of the above approach in practice is however hampered by some limitations of the proposed framework. Namely, the policy considered in [6] allows only for the specification of conjunctive queries (CQs), thus ruling out many important data protection statements typical of real-world applications. The company of our example, for instance, might want to protect only data referring to facts occurred after a certain year, and this cannot be expressed through a CQ. Moreover, IGA censors might result too restrictive with respect to the amount of non-protected data disclosed to the user. In our example, the query $\exists x.\mathsf{maintainedBy}(e, x)$ is implied under both the GA censors (i.e., inferred by each ontology we obtain by coupling the TBox with the ABox returned by a GA censor), but it is not implied under the IGA censor. Thus, confidentiality protection through IGA censors might obfuscate too much information. At the same time, answering CQs by reasoning over all GA censors is intractable, as shown in [12], and randomly selecting one single censor is arbitrary without additional metadata.

However, in practical scenarios such metadata are often available, and may lead to prefer one censor to another, so that simply taking the intersection of the results of all censors would be unsatisfactory. For instance, the company of our example might consider it preferable to disclose $\mathsf{type}(e, \text{`}\textit{wildcat}\text{'})$ over $\mathsf{emptyWell}(e)$, but not acceptable disclosing both, according to the policy. This situation calls for new modeling tools.

[1] This example is inspired by the benchmark we use in the experiments.

In this paper we contribute to fill the previous gaps, by enriching the CQE framework of [6] to support prioritized ontologies and a more expressive policy language[2], thus allowing for a more flexible management of information disclosure, still guaranteeing feasibility of the approach. Our contributions can be summarized as follows:

- We consider ontologies specified in *DL-Lite*$_A$, which is more expressive than *DL-Lite*$_\mathcal{R}$ studied in [6] and is one of the richest DLs of the *DL-Lite* family, i.e., the logical underpinning of the OWL 2 profile OWL 2 QL[3].
- We extend the policy language by allowing for CQs with atoms using comparison predicates, in a controlled way.
- We allow for the presence of priority relations between ontology predicates, such as, e.g., $\mathsf{type} \succ \mathsf{emptyWell}$, and exploit them to identify preferred optimal censors. To this aim, we first propose priority-based censor semantics for our framework, by adapting the well-known notions of Pareto and Global optimal repairs proposed in [15] in the context of Consistent Query Answering (CQA). To overcome intractability of query answering under such semantics, we provide a sound approximation of both the Pareto and Global censors, called DD censor, for which CQ answering in *DL-Lite*$_A$ is polynomial in data complexity.
- We exhibit a parametrized version of the DD censor enabling for first-order rewritable CQ answering in *DL-Lite*$_A$, which proves AC^0 data complexity.
- We show practical applicability of our approach through an experimental study over the NPD benchmark [11]. To this aim, we cable our rewriting technique in the method given in [6] that solves query answering under censors via a reduction to query processing in OBDA. Our experiments show that CQE under priorities is feasible in practice and that priorities are particularly effective in increasing the amount of data disclosed to the user, still guaranteeing confidentiality preservation.

Related work. Previous works on CQE over ontologies have considered policies expressed as ground atoms [8], ontology axioms [5], or CQs [9,12,7,6], which, as said, we extend with the presence of comparison predicates. Query answering under censors as a form of skeptical reasoning, as we do in this paper, has been first investigated in [12], from the theoretical viewpoint. In [5] and [7] censors over DL ontologies have been studied under the indistinguishability perspective, explicitly requiring that the answer returned by a censor does not allow the user to distinguish the instances containing sensitive information from the ones with no secrets. As shown in [5], this property may also protect from attacks of users with some background knowledge, thus it is important for robust privacy-preservation. We remark that, as proved in the following, the censors we consider in this paper satisfy this property. Leveraging an indistinguishability-based notion of source policy compliance, reference [2] studies information disclosure in OBDA, but does not consider query answering, as we do in this paper.

To the best of our knowledge, this is the first paper considering CQE over prioritized ontologies. The priority-based CQE semantics we propose are adapted from the literature on CQA. More in detail, our DD censor has a correspondence with the grounded

[2] For the sake of presentation, we consider here CQE over ontologies. Our extensions and results apply straightforwardly to a privacy-protected OBDA framework [6].

[3] https://www.w3.org/TR/owl2-profiles/

extension recently introduced in [3] through a transformation of the CQA problem into argumentation framework. Also, our rewritability result corresponds to an analogous finding mentioned in that paper. Besides the differences between the settings studied in the two papers, we remark that priorities considered in [3] are specified between ABox facts, whereas we here assume priorities between ontology predicates, maintaining this aspect at the intensional level, thus enriching the modeling abilities of the system designer. Furthermore, our treatment is tailored to CQE, and does not require transformation into a different problem, thus streamlining the technical aspects of the approach. Finally, the rewriting algorithm that we provide allows us to easily exploit the idea of [6] for solving CQE over ontologies through the use of off-the-shelf tools for OBDA.

Paper organization. In Section 2 we provide some preliminaries. In Section 3 we present the CQE framework for ontologies and the new policy language considered in this paper. In Section 4 we introduce priority relations between ontology predicates and define Pareto and Global censors. In Section 5 we provide sound approximations of the Pareto and Global censors and give our query-rewriting algorithm. In Section 6, we present our experiments, and in Section 7 we conclude the paper.

2 Preliminaries

Description Logics (DLs) are decidable first-order (FO) languages using unary and binary predicates [1]. Unary predicates are called concepts, corresponding to classes in OWL, which denote sets of objects, whereas binary predicates can be either roles, called object properties in OWL, denoting relations between concepts, or attributes, called data properties in OWL, denoting relations between concepts and data-types. Hereinafter we assume to have the pairwise disjoint countably infinite alphabets Σ_O, Σ_I, Σ_V, and $\Sigma_\mathcal{V}$, for ontology predicates, constants (a.k.a. individuals), values, and variables, respectively. Σ_O is in turn partitioned in three pairwise disjoint sets Σ_C, Σ_R, Σ_A, for names of concepts, a.k.a. atomic concepts, roles, a.k.a. atomic roles, and attributes, respectively. Furthermore, with $\Sigma_T = \Sigma_I \cup \Sigma_V \cup \Sigma_\mathcal{V}$ we denote the alphabet of terms.

A *DL ontology* $\mathcal{O}$ is a set $\mathcal{T} \cup \mathcal{A}$, where $\mathcal{T}$ is the *TBox*, i.e., a finite set of assertions modeling intensional knowledge, and $\mathcal{A}$ is the *ABox*, i.e., a finite set of assertions specifying extensional knowledge. For us, an ABox is always a set of assertions of the form $A(a)$, $P(a, b)$, $U(a, v)$, where $A \in \Sigma_C$, $P \in \Sigma_R$, $U \in \Sigma_A$, $a, b \in \Sigma_I$, and $v \in \Sigma_V$. The set of concept, role, and attribute names occurring in $\mathcal{O}$ is the *signature* of $\mathcal{O}$, denoted $\Sigma_O(\mathcal{O})$. The semantics of $\mathcal{O}$ is given in terms of interpretations [1]. A *model* of $\mathcal{O}$ is an interpretation that satisfies all assertions in $\mathcal{T}$ and $\mathcal{A}$. $\mathcal{O}$ is *consistent* if it has at least one model, *inconsistent* otherwise. Then, $\mathcal{O}$ *entails* an FO sentence ϕ, i.e., a closed FO formula, if ϕ is true in every model of $\mathcal{O}$. Given a TBox $\mathcal{T}$, *an ABox* $\mathcal{A}$ *for* $\mathcal{T}$ contains only assertions over $\Sigma_O(\mathcal{T})$, Σ_I and Σ_V, and $\mathcal{A}$ is such that $\mathcal{T} \cup \mathcal{A}$ is consistent. In the following, given an ABox $\mathcal{A}$ for $\mathcal{T}$, we denote by $\mathsf{cl}(\mathcal{T}, \mathcal{A})$ the set of all facts α constructible over the alphabets $\Sigma_O(\mathcal{T})$, Σ_I and Σ_V, such that $\mathcal{T} \cup \mathcal{A} \models \alpha$.

A *query* q *over a DL ontology* $\mathcal{O}$ is an FO formula $\phi(\vec{x})$ over $\Sigma_O(\mathcal{O}) \cup \Sigma_T$. The variables in $\vec{x}$ are the free variables of q, and the number of variables in $\vec{x}$ is the *arity* of q. The *evaluation of* q over a model $\mathcal{I}$ for $\mathcal{O}$ is the set of tuples of elements in the domain of $\mathcal{I}$ that assigned to the variables in $\vec{x}$ make the query true in $\mathcal{I}$.

An *atom* over $\Sigma_O \cup \Sigma_T$ is an expression of the form $A(t)$, $P(t_1, t_2)$ or $U(t_1, t_2)$ where $A \in \Sigma_C$, $P \in \Sigma_R$, $U \in \Sigma_A$ and t, t_1, t_2 are terms from Σ_T. A query q over $\mathcal{O}$ is a *conjunctive query (CQ)* if $\phi(\vec{x})$ is an expression of the form $\exists\vec{y}.S_1(\vec{x}, \vec{y}) \wedge \ldots \wedge S_n(\vec{x}, \vec{y})$, where $n \geq 1$, $\vec{y}$ are the existential variables, and each $S_i(\vec{x}, \vec{y})$ is an atom over $\Sigma_O(\mathcal{O}) \cup \Sigma_T$ with variables in $\vec{x} \cup \vec{y}$. Each variable in $\vec{x} \cup \vec{y}$ occurs in at least one atom of q. Boolean CQs (BCQs) are queries whose arity is zero (i.e., BCQs are sentences).

We will focus on the *DL-Lite*$_A$ language, whose constructs are formed as follows:

$$B \longrightarrow A \mid \exists R \mid \exists U \qquad R \longrightarrow P \mid P^-$$

where $A \in \Sigma_C$, $P \in \Sigma_R$, P^- is the *inverse* of $P \in \Sigma_R$, and $U \in \Sigma_A$. B and R denote a *basic concept* and a *basic role*, respectively. The concept $\exists R$ and $\exists U$ are the domain of R and U, respectively. *DL-Lite*$_A$ TBox assertions assume the following form:

$$\begin{array}{llll} B_1 \sqsubseteq B_2 & R_1 \sqsubseteq R_2 & U_1 \sqsubseteq U_2 & \rho(U) \sqsubseteq F \\ B_1 \sqsubseteq \neg B_2 & R_1 \sqsubseteq \neg R_2 & U_1 \sqsubseteq \neg U_2 & (\mathsf{funct}\ R) \qquad (\mathsf{funct}\ U) \end{array}$$

where $\rho(U)$ denotes the range of an attribute U, i.e., the set of values to which U relates some object, and $F \subseteq \Sigma_V$ is a value-domain (e.g., integers, strings, etc.). A *DL-Lite*$_A$ *TBox* $\mathcal{T}$ is a finite set of assertions of the above kind, such that each basic role R or attribute U that is functional in $\mathcal{T}$, i.e., $(\mathsf{funct}\ R) \in \mathcal{T}$ or $(\mathsf{funct}\ U) \in \mathcal{T}$, is never specialized, i.e., it (or its inverse, in the case of role) does not occur in assertions of the form $R' \sqsubseteq R$ or $U' \sqsubseteq U$. For the semantics of *DL-Lite*$_A$, we refer the reader to [13].

All our complexity results are given with respect to the size of the ABox only, i.e., they refer to data complexity. For the sake of exposition, in the following we deal with entailment of BCQs from DL ontologies. Our results can be straightforwardly extended to non-Boolean CQs, which we indeed consider in the experiments.

3 Framework for CQE in DLs

We now define the framework for CQE over DL ontologies. In this section we do not consider priorities between ontology predicates, which will be introduced in the next section. We start with the definition of *Boolean CQs with inequalities* (*BCQ*$_{ineq}$). To this aim, we first define *inequality atoms* over Σ_T as expressions of the form $t_1\, op\ t_2$ where $t_1, t_2 \in \Sigma_T$ and $op \in \{\neq, <, \leq, >, \geq\}$. Then, a *BCQ*$_{ineq}$ q over an ontology $\mathcal{O}$, is a sentence of the form: $\exists\vec{y}.\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \rho_1 \wedge \ldots \wedge \rho_m$, where $n \geq 1$, $m \geq 0$, every α_i is an atom over $\Sigma_O(\mathcal{O}) \cup \Sigma_T$, with variables in $\vec{y}$, and every ρ_i is an inequality atom over Σ_T with variables in $\vec{y}$. We denote as *Ineq*(q) the set of inequality atoms occurring in q, and as *Pos*(q) the Boolean CQ obtained from q by eliminating all the inequality atoms. We also assume that every variable x occurring in *Ineq*(q) occurs at least once in *Pos*(q). The evaluation of q over an interpretation is given in the standard way, by assuming that $\{\neq, <, \leq, >, \geq\}$ are interpreted in the same way in every interpretation.

Given a DL TBox $\mathcal{T}$, *a policy* $\mathcal{P}$ *for* $\mathcal{T}$ is a set of denial assertions (or simply *denials*), i.e., formulas of the form $\forall\vec{x}.\phi(\vec{x}) \rightarrow \bot$ such that $\exists\vec{x}.\phi(\vec{x})$ is a *BCQ*$_{ineq}$ over $\mathcal{T}$. We always assume that $\mathcal{T} \cup \mathcal{P}$ is consistent, i.e., there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that all the formulas in $\mathcal{P}$ are satisfied. We point out that queries used in the previous definition

are more expressive than formulas used in policies in previous works on CQE over ontologies (e.g., [9,12,7]). We also notice that reasoning over $\mathcal{T} \cup \mathcal{P}$ may be problematic from a computational viewpoint, even for a TBox expressed in a light DL. At the end of this section we will give a syntactic restriction on the interaction between $\mathcal{T}$ and $\mathcal{P}$ for the case in which $\mathcal{T}$ is a *DL-Lite*$_A$ TBox. As we will show in the rest of the paper, this restriction is enough to obtain a setting with well-founded CQE semantics and efficient reasoning (namely query answering), amenable to implementation.

An $\mathcal{L}$ *CQE specification* $\mathcal{E}$ is a pair $\langle \mathcal{T}, \mathcal{P} \rangle$, where $\mathcal{T}$ is a TBox in the DL $\mathcal{L}$ and $\mathcal{P}$ a policy for $\mathcal{T}$ (we will omit $\mathcal{L}$ for definitions and results applying to any DL language).

Example 2. Consider the *DL-Lite*$_A$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, where:

$\mathcal{T} = \{\exists \mathsf{doc}^- \sqsubseteq \mathsf{wellbore}\}$
$\mathcal{P} = \{\forall w, y, d.\mathsf{wellbore}(w) \wedge \mathsf{type}(w, \text{`}\textit{wildcat}\text{'}) \wedge \mathsf{year}(w, y) \wedge \mathsf{doc}(d, w) \wedge y > 1980 \rightarrow \bot,$
$\forall w, y.\mathsf{wellbore}(w) \wedge \mathsf{year}(w, y) \wedge \mathsf{doc}(d, w) \wedge y > 1992 \rightarrow \bot,$
$\forall w, d.\mathsf{wellbore}(w) \wedge \mathsf{doc}(d, w) \wedge \mathsf{age}(w, \text{`}\textit{Eocene}\text{'}) \rightarrow \bot\}$

In words, the TBox $\mathcal{T}$ sanctions that the documents are always about wellbores. The first denial in $\mathcal{P}$ declares confidential documents about wildcat wellbores that have been drilled after 1980. The second denial asserts that documents about wellbores drilled after 1992 have not to be disclosed. Finally, the last denial specifies that no document about wellbores that extract hydrocarbons from lithostratigraphic unit of Eocene era have to be divulged. □

A censor for $\mathcal{E}$ is a function disclosing only information that does not lead to violations of the policy $\mathcal{P}$. Below we provide a notion studied in [12,7,6].

Definition 1 (GA censor). *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification. A* Ground Atom (GA) censor *for $\mathcal{E}$ is a function $\mathsf{cens}(\cdot)$ such that for each ABox $\mathcal{A}$ for $\mathcal{T}$, returns a set $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}(\mathcal{T}, \mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is consistent*

Given two GA censors $\mathsf{cens}(\cdot)$ and $\mathsf{cens}'(\cdot)$ for a CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, we say that $\mathsf{cens}'(\cdot)$ is *more informative* than $\mathsf{cens}(\cdot)$ if: (i) $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cens}'(\mathcal{A})$, for every ABox $\mathcal{A}$ for $\mathcal{T}$, and (ii) there exists an ABox $\mathcal{A}'$ for $\mathcal{T}$ such that $\mathsf{cens}(\mathcal{A}') \subset \mathsf{cens}'(\mathcal{A}')$.

We say that a GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ is *optimal* if there does not exist a GA censor $\mathsf{cens}'(\cdot)$ for $\mathcal{E}$ such that $\mathsf{cens}'(\cdot)$ is more informative than $\mathsf{cens}(\cdot)$. We denote by $\mathsf{optGACens}(\mathcal{E})$ the set of all optimal GA censors for a CQE specification $\mathcal{E}$.

Example 3. Let $\mathcal{E}$ be as in Example 2, and let $\mathsf{cens}(\cdot)$ be the function such that, given an ABox $\mathcal{A}$ for $\mathcal{T}$, $\mathsf{cens}(\mathcal{A}) = \mathsf{cl}(\mathcal{T}, \mathcal{A}')$, where $\mathcal{A}'$ is the ABox obtained from $\mathcal{A}$ by adding the atom $\mathsf{wellbore}(w)$ and removing the atom $\mathsf{doc}(d, w)$ for each pair of individuals (d, w) such that $\mathcal{T} \cup \mathcal{A} \models \exists y.(\mathsf{wellbore}(w) \wedge \mathsf{type}(w, \text{`}\textit{wildcat}\text{'}) \wedge \mathsf{year}(w, y) \wedge \mathsf{doc}(d, w) \wedge y > 1980) \vee (\mathsf{wellbore}(w) \wedge \mathsf{year}(w, y) \wedge \mathsf{doc}(d, w) \wedge y > 1992) \vee (\mathsf{wellbore}(w) \wedge \mathsf{doc}(d, w) \wedge \mathsf{age}(w, \text{`}\textit{Eocene}\text{'}))$. One can easily verify that $\mathsf{cens} \in \mathsf{optGACens}(\mathcal{E})$. □

We now define query entailment over GA censors.

Definition 2. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification, q be a BCQ, and $\mathcal{A}$ be an ABox for $\mathcal{T}$.* GA-Cens-Ent *is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \models q$ for each $\mathsf{cens} \in \mathsf{optGACens}(\mathcal{E})$.*

As shown in [12], the above problem is intractable even for light DLs such as *DL-Lite*$_{\mathcal{R}}$ and $\mathcal{EL}_{\bot}$, and for a policy language less expressive than the one we consider in this paper. Towards the identification of a practical setting, in [6] the authors have proposed a sound approximation of GA censors, for which entailment of BCQs in *DL-Lite*$_{\mathcal{R}}$ CQE specifications (with a policy denying CQs) has been shown to be reducible to standard BCQ entailment in OBDA.

Definition 3 (IGA censor). *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification, the* intersection GA (IGA) censor *for $\mathcal{E}$ is the function $\mathsf{cens}_{IGA}(\cdot)$ such that, for every ABox $\mathcal{A}$ for $\mathcal{T}$, $\mathsf{cens}_{IGA}(\mathcal{A}) = \bigcap_{\mathsf{cens} \in \mathsf{optGACens}(\mathcal{E})} \mathsf{cens}(\mathcal{A})$.*

The IGA censor for a CQE specification $\mathcal{E}$ always exists [6]. Given a BCQ q and an ABox $\mathcal{A}$ for $\mathcal{T}$, IGA-Cens-Ent amounts to decide whether $\mathcal{T} \cup \mathsf{cens}_{IGA}(\mathcal{A}) \models q$. Obviously, IGA-Cens-Ent implies (i.e., it is a sound approximation of) GA-Cens-Ent.

Example 4. Consider the CQE specification $\mathcal{E}$ of Example 2 and Example 3, and the ABox $\mathcal{A} = \{\mathsf{type}(o, \text{`}\textit{wildcat}\text{'}), \mathsf{year}(o, 1985), \mathsf{doc}(d, o), \mathsf{age}(o, \text{`}\textit{Eocene}\text{'})\}$. One can verify that $\mathsf{cens}_{IGA}(\mathcal{A}) = \{\mathsf{wellbore}(o)\}$. □

As said in the introduction, to increase robustness of censors, literature on CQE has often looked at censors satisfying a property of instance indistinguishability [4,5,2,7]. Intuitively, a censor fulfilling such a property masks confidential information in such a way that a user cannot distinguish an instance actually containing data protected by the policy from an instance without such data, so that the incompleteness of the information of a possible attacker is increased. In our framework, this is formalized as follows.

Definition 4 (indistinguishability). *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification and $\mathsf{cens}(\cdot)$ be a censor for $\mathcal{E}$. We say that $\mathsf{cens}(\cdot)$* satisfies the indistinguishability property *if for every ABox $\mathcal{A}$ for $\mathcal{T}$, there exists an ABox $\mathcal{A}'$ for $\mathcal{T}$ (not necessarily distinct from $\mathcal{A}$) such that: (i) $\mathsf{cens}(\mathcal{A}) = \mathsf{cens}(\mathcal{A}')$, and (ii) $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is consistent.*

It is not difficult to see that the following proposition holds.

Proposition 1. *For every CQE specification $\mathcal{E}$, both optimal GA censors and the IGA censor for $\mathcal{E}$ satisfy the indistinguishability property.*

We next provide with two definitions that will be useful in the following. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification and $\mathcal{A}$ be an ABox for $\mathcal{T}$. We say that a set of ABox assertions $\mathcal{S} \subseteq \mathsf{cl}(\mathcal{T}, \mathcal{A})$ is a *secret* in $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$, if $\mathcal{T} \cup \mathcal{P} \cup \mathcal{S}$ is inconsistent and for each assertion $\sigma \in \mathcal{S}$ we have that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{S} \setminus \{\sigma\}$ is consistent. We denote with $secrets(\mathcal{T}, \mathcal{P}, \mathcal{A})$ the set of all secrets in $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$, and, given an ABox assertion γ, with $inSecrets(\mathcal{T}, \mathcal{P}, \mathcal{A}, \gamma)$ the set of secrets $\mathcal{S} \in secrets(\mathcal{T}, \mathcal{P}, \mathcal{A})$ such that $\gamma \in \mathcal{S}$.

As announced, we conclude this section by discussing the case of *DL-Lite*$_A$ CQE specifications to provide a practical syntactic condition that we will exploit to obtain our main computational results. We say that a denial $\forall \vec{x}.\phi(\vec{x}) \rightarrow \bot$ is *safe* w.r.t. a *DL-Lite*$_A$ TBox $\mathcal{T}$ if every variable x in *Ineq*$(\exists \vec{x}.\phi(\vec{x}))$ occurs in *Pos*$(\exists \vec{x}.\phi(\vec{x}))$ only in safe attribute range positions, i.e., in atoms of the form $U(t, x)$ such that U is an attribute and there exists no basic concept $B \neq \exists U$ such that $\mathcal{T} \models B \sqsubseteq \exists U$. Then, a policy $\mathcal{P}$ is safe w.r.t. $\mathcal{T}$, if $\mathcal{P}$ contains only denials that are safe w.r.t. $\mathcal{T}$, and a *DL-Lite*$_A$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ is *safe* if $\mathcal{P}$ is safe w.r.t. $\mathcal{T}$. It is easy to see that the *DL-Lite*$_A$ CQE specification of Example 2 (and throughout all examples of this paper) is safe.

4 Prioritized CQE Framework

Given a TBox $\mathcal{T}$, a priority relation $\succ$ over $\mathcal{T}$ is an acyclic binary relation over the signature of $\mathcal{T}$, i.e., $\succ \subseteq \Sigma_O(\mathcal{T}) \times \Sigma_O(\mathcal{T})$. A *prioritized $\mathcal{L}$ CQE specification* $\mathcal{E}_\succ$ is a triple $\langle \mathcal{T}, \mathcal{P}, \succ \rangle$, such that $\langle \mathcal{T}, \mathcal{P} \rangle$ is an $\mathcal{L}$ CQE specification.

Example 5. $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, where $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ is as in Example 2 and $\succ$ specifies that $\mathsf{type} \succ \mathsf{doc}$ and $\mathsf{year} \succ \mathsf{doc}$, is a (safe) prioritized *DL-Lite*$_A$ CQE specification. □

The definitions of GA censor, optimal GA censor, IGA censor, GA-Cens-Ent, and IGA-Cens-Ent apply also to a prioritized CQE specification (e.g., given one such specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, $\mathsf{cens}(\cdot)$ is a GA censor for $\mathcal{E}_\succ$ if it is a GA censor for the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$). We also use for prioritized CQE specifications the same notations introduced in Section 3 for CQE specifications, with the same meaning.

We now exploit the priority relation to define a preference criterion over censors. We consider two optimality notions introduced by [15] in the context of consistent query answering over databases, and recently adopted in [3] for repairing inconsistent prioritized DL ontologies. Whereas the priority relations considered in this paper are intentional, i.e., between ontology predicates, priorities considered in [15,3] are between (conflicting) facts. Intentional priorities however straightforwardly induce priorities over facts: given a TBox $\mathcal{T}$, a priority relation $\succ$ over $\mathcal{T}$, an ABox $\mathcal{A}$ for $\mathcal{T}$, and two assertions $S_1(\vec{n})$ and $S_2(\vec{m})$ in $\mathcal{A}$, we have that $S_1(\vec{n}) \succ S_2(\vec{m})$ if $S_1 \succ S_2$. Below we take the definitions of Pareto- and Global-optimal repair from [3] and adapt them to our framework.

Definition 5 (Pareto/Global censor). *Let $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a prioritized CQE specification, $\mathcal{A}$ be an ABox for $\mathcal{T}$, and $\mathsf{cens}(\cdot) \in \mathsf{optGACens}(\mathcal{E}_\succ)$. We say that an ABox $\mathcal{A}' \subseteq \mathsf{cl}(\mathcal{T}, \mathcal{A})$, such that $\mathcal{A}' \neq \mathsf{cens}(\mathcal{A})$ and $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is consistent, is:*

- *a* Pareto improvement *of $\mathsf{cens}(\mathcal{A})$ w.r.t. $\mathcal{E}_\succ$ if there exists $\gamma' \in \mathcal{A}' \setminus \mathsf{cens}(\mathcal{A})$ such that $\gamma' \succ \gamma$ for every $\gamma \in \mathsf{cens}(\mathcal{A}) \setminus \mathcal{A}'$ and $\{\gamma, \gamma'\} \subseteq \mathcal{S}$ for some $\mathcal{S} \in secrets(\mathcal{T}, \mathcal{P}, \mathcal{A})$;*
- *a* Global improvement *of $\mathsf{cens}(\mathcal{A})$ w.r.t. $\mathcal{E}_\succ$ if for each $\gamma \in \mathsf{cens}(\mathcal{A}) \setminus \mathcal{A}'$ there exists $\gamma' \in \mathcal{A}' \setminus \mathsf{cens}(\mathcal{A})$ such that $\gamma' \succ \gamma$ and $\{\gamma, \gamma'\} \subseteq \mathcal{S}$ for some $\mathcal{S} \in secrets(\mathcal{T}, \mathcal{P}, \mathcal{A})$.*

Then, $\mathsf{cens}(\cdot)$ is a Pareto (resp. Global) censor *for $\mathcal{E}_\succ$ if there exists no other GA censor $\mathsf{cens}'(\cdot)$ for $\mathcal{E}_\succ$ such that, for each ABox $\mathcal{A}$ for $\mathcal{T}$, either $\mathsf{cens}'(\mathcal{A}) = \mathsf{cens}(\mathcal{A})$ or $\mathsf{cens}'(\mathcal{A})$ is a Pareto (resp. Global) improvement of $\mathsf{cens}(\mathcal{A})$ w.r.t. $\mathcal{E}_\succ$.*

We denote with $\mathsf{PCens}(\mathcal{E}_\succ)$ (resp. $\mathsf{GCens}(\mathcal{E}_\succ)$) the set of all Pareto (resp. Global) censors for $\mathcal{E}_\succ$. It is easy to see that $\mathsf{GCens}(\mathcal{E}_\succ) \subseteq \mathsf{PCens}(\mathcal{E}_\succ) \subseteq \mathsf{optGACens}(\mathcal{E}_\succ)$ for every $\mathcal{E}_\succ$, analogous to the containment between Global and Pareto repairs given in [15]. Also, if $\succ$ is empty, then $\mathsf{PCens}(\mathcal{E}_\succ) = \mathsf{GCens}(\mathcal{E}_\succ) = \mathsf{optGACens}(\mathcal{E}_\succ)$. As done for GA censors, we define intersection-based versions of Pareto and Global censors. Namely, we call *Intersection Pareto (IP) censor* for $\mathcal{E}_\succ$ the function $\mathsf{cens}_{IP}(\cdot)$ such that, for every ABox $\mathcal{A}$ for $\mathcal{T}$, $\mathsf{cens}_{IP}(\mathcal{A}) = \bigcap_{\mathsf{cens} \in \mathsf{PCens}} \mathsf{cens}(\mathcal{A})$, and *Intersection Global (IG) censor* for $\mathcal{E}_\succ$ the function $\mathsf{cens}_{IG}(\cdot)$ such that, for every ABox $\mathcal{A}$ for $\mathcal{T}$, $\mathsf{cens}_{IG}(\mathcal{A}) = \bigcap_{\mathsf{cens} \in \mathsf{GCens}} \mathsf{cens}(\mathcal{A})$. Obviously, $\mathsf{cens}_{IP}(\mathcal{A}) \subseteq \mathsf{cens}_{IG}(\mathcal{A})$ for each ABox $\mathcal{A}$ for $\mathcal{T}$. Also, if $\succ$ is empty, then, since $\mathsf{PCens}(\mathcal{E}_\succ) = \mathsf{GCens}(\mathcal{E}_\succ) = \mathsf{optGACens}(\mathcal{E}_\succ)$, we have that $\mathsf{cens}_{IP}(\cdot) = \mathsf{cens}_{IG}(\cdot) = \mathsf{cens}_{IGA}(\cdot)$.

Given an ABox $\mathcal{A}$ for $\mathcal{T}$ and a BCQ q, P-Cens-Ent (resp. G-Cens-Ent) is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \models q$ for each $\mathsf{cens}(\cdot) \in \mathsf{PCens}(\mathcal{E}_\succ)$ (resp. $\mathsf{cens}(\cdot) \in \mathsf{GCens}(\mathcal{E}_\succ)$), and IP-Cens-Ent (resp. IG-Cens-Ent) is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}_{IP}(\mathcal{A}) \models q$ (resp. $\mathcal{T} \cup \mathsf{cens}_{IG}(\mathcal{A}) \models q$). It is immediate to see that P-Cens-Ent implies G-Cens-Ent, and IP-Cens-Ent (resp. IG-Cens-Ent) implies P-Cens-Ent (resp. G-Cens-Ent). The following results immediately follow from [3].

Theorem 1. *Let $\mathcal{E}_\succ \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a safe prioritized DL-Lite$_A$ CQE specification, $\mathcal{A}$ be an ABox for $\mathcal{T}$, and q be a BCQ. P-Cens-Ent and IP-Cens-Ent are coNP-hard in data complexity, whereas G-Cens-Ent and IG-Cens-Ent are Π_2^p-hard in data complexity.*

Results in Theorem 1 represent a clear obstacle to the use of the above forms of priority-based censors over real-world, large datasets. In the next section we will see how these censors can be suitably approximated for a practical use.

5 FO-rewritable prioritized CQE in *DL-Lite*$_A$

In this section we first give a deterministic notion of priority-based censor (DD censor) and its parametrized sound approximation called k-DD censor. Then, we provide an algorithm that computes a non-redundant policy, i.e., such that the image of each policy assertion corresponds to a secret. This step is crucial in order to define our query rewriting technique, which shows that BCQ entailment under k-DD censors in *DL-Lite*$_A$ is FO rewritable. The full rewriting algorithm is given in the last part of this section.

5.1 DD censors and k-DD censors

Theorem 1 clearly says that under Pareto or Global censors, or their intersection-based versions, entailment of BCQs is inherently non-deterministic. Towards the identification of a tractable approximation, we give below the notion of *deterministically disclosed (DD)* and *deterministically censored (DC)* atoms. Hereinafter, given a priority relation $\succ$, a fact α, and a set of facts $\mathcal{S}$, we write $\alpha \succ \mathcal{S}$ if there exists $\beta \in \mathcal{S}$ such that $\alpha \succ \beta$.

Definition 6. *Given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ and an ABox $\mathcal{A}$ for $\mathcal{T}$, we denote by DD$(\mathcal{E}_\succ, \mathcal{A})$ and DC$(\mathcal{E}_\succ, \mathcal{A})$ the inclusion-minimal subsets of* $\mathsf{cl}(\mathcal{T}, \mathcal{A})$ *such that:*

$$\begin{aligned} DD(\mathcal{E}_\succ, \mathcal{A}) = \{\alpha \in \mathsf{cl}(\mathcal{T}, \mathcal{A}) \mid \forall \mathcal{S} \in inSecrets(\mathcal{T}, \mathcal{P}, \mathcal{A}, \alpha) \text{ either } \alpha \succ (\mathcal{S} \setminus \{\alpha\}) \\ \text{or } \mathcal{S} \cap DC(\mathcal{E}_\succ, \mathcal{A}) \neq \emptyset \} \\ DC(\mathcal{E}_\succ, \mathcal{A}) = \{\alpha \in \mathsf{cl}(\mathcal{T}, \mathcal{A}) \mid \exists \mathcal{S} \in inSecrets(\mathcal{T}, \mathcal{P}, \mathcal{A}, \alpha) \text{ s.t. } \mathcal{S} \setminus DD(\mathcal{E}_\succ, \mathcal{A}) = \{\alpha\}\} \end{aligned}$$

In words, a DD atom α is such that α does not occur in any secret, or, either, in each secret in which it occurs there is an atom β such that $\alpha \succ \beta$ or β is a DC atom. Instead, a DC atom is such that there is a secret where it is the only non-DD atom. It is immediate to verify that DD$(\mathcal{E}_\succ, \mathcal{A})$ and DC$(\mathcal{E}_\succ, \mathcal{A})$ are unique for a given pair $(\mathcal{E}_\succ, \mathcal{A})$.

Given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, we call *DD censor* for $\mathcal{E}_\succ$ the function $\mathsf{cens}_{DD}(\cdot)$ such that, for each ABox $\mathcal{A}$ for $\mathcal{T}$, $\mathsf{cens}_{DD}(\mathcal{A}) = \text{DD}(\mathcal{E}_\succ, \mathcal{A})$

Example 6. Consider the safe prioritized *DL-Lite*$_A$ CQE specification $\mathcal{E}_\succ$ of Example 5 and the censor cens of Example 3. We have that cens coincides with the DD censor for $\mathcal{E}_\succ$. Moreover, for the ABox $\mathcal{A}$ of Example 4, $\mathsf{cens}(\mathcal{A}) = \{\mathsf{wellbore}(o), \mathsf{type}(o, \text{'}\textit{wildcat}\text{'}), \mathsf{year}(o, 1985), \mathsf{age}(o, \text{'}\textit{Eocene}\text{'})\}$. □

The proposition below follows from the definition of DD censor[4].

Proposition 2. *Let $\mathcal{E}_\emptyset = \langle \mathcal{T}, \mathcal{P}, \emptyset \rangle$ be a prioritized CQE specification with an empty priority relation. The DD censor for $\mathcal{E}_\emptyset$ coincides with the IGA censor for $\mathcal{E}_\emptyset$.*

It is also easy to verify that the DD censor satisfies the property given in Definition 4.

Proposition 3. *For every prioritized CQE specification $\mathcal{E}_\succ$, the DD censor for $\mathcal{E}_\succ$ satisfies the indistinguishability property.*

We now establish the relationship between DD censors and the previously presented IP and IG censors.

Proposition 4. *Let $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a prioritized CQE specification, and let $\mathsf{cens}_{IP}(\cdot)$ and $\mathsf{cens}_{IG}(\cdot)$ be the Intersection Pareto and Global censor for $\mathcal{E}_\succ$. Then, DD$(\mathcal{E}_\succ, \mathcal{A}) \subseteq \mathsf{cens}_{IP}(\mathcal{A}) \subseteq \mathsf{cens}_{IG}(\mathcal{A})$, for every ABox $\mathcal{A}$ for $\mathcal{T}$.*

BCQ entailment under DD censors is defined as usual. Namely, given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, an ABox $\mathcal{A}$ for $\mathcal{T}$, and a BCQ q, DD-Cens-Ent is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}_{DD}(\mathcal{A}) \models q$. From Proposition 4, it follows that DD-Cens-Ent implies IP-Cens-Ent (and consequently IG-Cens-Ent).

Given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, and an ABox $\mathcal{A}$ for $\mathcal{T}$, it is not difficult to see that DD$(\mathcal{E}_\succ, \mathcal{A})$ and DC$(\mathcal{E}_\succ, \mathcal{A})$ correspond to the least fixpoint of the equations:

$$\begin{aligned}
\text{DD}_{i+1}(\mathcal{E}_\succ, \mathcal{A}) = \{\alpha \in \mathsf{cl}(\mathcal{T}, \mathcal{A}) \mid \forall \mathcal{S} \in \mathit{inSecrets}(\mathcal{T}, \mathcal{P}, \mathcal{A}, \alpha), \\
\alpha \succ (\mathcal{S} \setminus \{\alpha\}) \text{ or } \mathcal{S} \cap \text{DC}_i(\mathcal{E}_\succ, \mathcal{A}) \neq \emptyset \,\} \\
\text{DC}_{i+1}(\mathcal{E}_\succ, \mathcal{A}) = \{\alpha \in \mathsf{cl}(\mathcal{T}, \mathcal{A}) \mid \exists \mathcal{S} \in \mathit{inSecrets}(\mathcal{T}, \mathcal{P}, \mathcal{A}, \alpha) \text{ s.t.} \\
\mathcal{S} \setminus \text{DD}_i(\mathcal{E}_\succ, \mathcal{A}) = \{\alpha\}\}
\end{aligned}$$

where $\text{DD}_0(\mathcal{E}_\succ, \mathcal{A}) = \text{DC}_0(\mathcal{E}_\succ, \mathcal{A}) = \emptyset$. For safe prioritized *DL-Lite*$_A$ CQE specifications, computing such fixpoint is in P in the size of $\mathcal{A}$, and from the results in [3] it also follows that DD-Cens-Ent is P-hard in data complexity. By fixing a k, we can define a new censor $\mathsf{cens}DD_k(\cdot)$, which we call *k-DD censor* for $\mathcal{E}_\succ$, such that $\mathsf{cens}DD_k(\mathcal{A}) = \text{DD}_k(\mathcal{E}_\succ, \mathcal{A})$, for each ABox $\mathcal{A}$ for $\mathcal{T}$.

We next define BCQ entailment under k-DD censors, which is the problem that we study in the rest of the paper for safe prioritized *DL-Lite*$_A$ CQE specifications.

Definition 7. *Let $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a prioritized $\mathcal{L}$ CQE specification, k be a positive integer, $\mathcal{A}$ be an ABox for $\mathcal{T}$, and q be a BCQ. kDD-Cens-Ent is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}DD_k(\mathcal{A}) \models q$.*

[4] A similar result is provided in [3, Theorem 38] in the context of CQA.

Since for every prioritized CQE specification $\mathcal{E}_\succ$, positive integer k, and ABox $\mathcal{A}$, $\text{DD}_k(\mathcal{E}_\succ, \mathcal{A}) \subseteq \text{DD}(\mathcal{E}_\succ, \mathcal{A})$, the k-DD censor for $\mathcal{E}_\succ$ constitutes a sound approximation of the DD censor for $\mathcal{E}_\succ$, and thus kDD-Cens-Ent implies DD-Cens-Ent. Moreover, it is immediate to verify that the k-DD censor preserves the indistinguishability property.

Example 7. For the specification $\mathcal{E}_\succ$ of Example 5 and the ABox $\mathcal{A}$ of Example 4, we have that $\text{DD}_1(\mathcal{E}_\succ, \mathcal{A}) = \{\mathsf{wellbore}(o), \mathsf{type}(o, \text{'}\textit{wildcat}\text{'}), \mathsf{year}(o, 1985)\}$, while $\text{DD}_3(\mathcal{E}_\succ, \mathcal{A}) = \{\mathsf{wellbore}(o), \mathsf{type}(o, \text{'}\textit{wildcat}\text{'}), \mathsf{year}(o, 1985), \mathsf{age}(o, \text{'}\textit{Eocene}\text{'})\}$, which coincides with the DD-censor for $\mathcal{E}_\succ$. □

5.2 Generating a non-redundant policy specification

We now provide the algorithm PolicyRefine, which we use to produce a non-redundant policy specification. A specification of this kind enjoys the property that every image over the ABox of a $\textit{BCQ}_{ineq}$ q in a policy denial is a secret, where the image is a minimal set of facts inferring q. This property is crucial for the correctness of the query rewriting algorithm presented in Section 5.3. It is not difficult to see that in general a policy can be redundant. For example, consider the policy $\mathcal{P} = \{A(x) \wedge U(x, y) \wedge y < 20 \rightarrow \bot; U(x, y) \wedge y < 15 \rightarrow \bot\}$ and the ABox $\mathcal{A} = \{A(a), U(a, 12)\}$, the ABox $\mathcal{A}$ itself is an image of the query in the premise of the first denial, but it is not a secret, since $U(a, 12)$ alone is a secret. The technique we propose here extends the one discussed in [6], tailored to policy assertions denying CQs.

We start with some preliminary definitions. As said before, the symbol op represents a comparison operator in $\{=, \neq, >, \geq, <, \leq\}$. Given a set of sets of inequalities RC and a denial $\delta = \forall \vec{x}.\phi(\vec{x}) \rightarrow \bot$, we denote by $\tau(\delta, RC)$ the function that returns the *extended denial assertion* $\forall \vec{x}.\phi(\vec{x}) \wedge \neg(\pi(\vec{x})) \rightarrow \bot$, where $\pi(\vec{x})$ is the disjunction of conjunctions of inequalities

$$\bigvee_{Ineq \in RC} \Big(\bigwedge_{t_1\, op\, t_2 \in Ineq} t_1\, op\, t_2 \Big)$$

In the rest of this section we call *non-extended denial*, or simply *denial*, a denial as defined in Section 3. Moreover, we call *extended policy* a set of extended denials and non-extended denials.

Given two set of inequalities $Ineq$ and $Ineq'$, we write $Ineq \models Ineq'$ to denote that every inequality in $Ineq'$ is implied by the inequalities in $Ineq$.

Definition 8. *Given an extended policy $\mathcal{P}$ and a non-extended denial δ in $\mathcal{P}$, we say that a set of inequalities $Ineq$ is a* strict redundancy condition *for δ in $\mathcal{P}$ if there exists $\delta' \in \mathcal{P}$ such that: (i) $\delta' \models \delta \cup Ineq$ and $\delta \cup Ineq \not\models \delta'$; (ii) there exists no set of inequalities $Ineq'$ such that $Ineq \models Ineq'$ and $\delta \models \delta' \cup Ineq'$ and $\delta' \cup Ineq' \not\models \delta$.*

We say that a set SRC of strict redundancy conditions for δ in $\mathcal{P}'$ is *complete* if, for every extended denial δ' in $\mathcal{P}'$, if there exists a set of inequalities $Ineq$ such that conditions (i) and (ii) of Definition 8 hold, then there exists a set $Ineq' \in SRC$ such that $Ineq \models Ineq'$.

Then, we say that an extended denial δ' is a *non-redundant representation of δ in* $\mathcal{P}'$ if every minimal ABox $\mathcal{A}$ such that $\{\delta'\} \cup \mathcal{A}$ is inconsistent is also a minimal ABox such that $\mathcal{P}' \cup \mathcal{A}$ is inconsistent.

Algorithm 1 PolicyRefine

input: a policy $\mathcal{P}$;
output: an extended policy $\mathcal{P}'$ that is a non-redundant representation of $\mathcal{P}$;
1) $\mathcal{P}' \leftarrow \emptyset$;
2) **foreach** denial $\delta \in \mathcal{P}$ **do**
3) $RC \leftarrow \emptyset$;
4) **foreach** denial $\delta' \in \mathcal{P}$ such that $\delta \neq \delta'$ **do**
5) **foreach** partition $Q_1, \ldots, Q_{k+1}$ of $Atoms(\delta)$ **do**
6) **foreach** partition $Q'_1, \ldots, Q'_k$ of $PredAt(\delta')$
7) **such that**, for each i s.t. $1 \leq i \leq k$,
8) $Q_i \cup Q'_i$ is a set of unifiable atoms **do**
9) $\sigma \leftarrow \bigcup_{1 \leq i \leq k} MGU(Q_i \cup Q'_i)$;
10) **if** $\sigma(Atoms(\delta')) \not\models \sigma(Q_{k+1})$
11) **then** $RC \leftarrow RC \cup \{\sigma\} \cup \sigma(CompAt(\delta'))$;
12) $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\tau(\delta, RC)\}$;
13) **return** $\mathcal{P}'$;

Definition 9. *We say that an extended policy $\mathcal{P}'$ is a* non-redundant representation *of an extended policy $\mathcal{P}$ if: (i) $\mathcal{P}'$ is equivalent to $\mathcal{P}$; and (ii) every $\delta \in \mathcal{P}'$ is such that there exists no strict redundancy condition for δ in $\mathcal{P}'$.*

We are now able to define the algorithm PolicyRefine (Figure 1). Given a policy $\mathcal{P}$, PolicyRefine($\mathcal{P}$) returns an extended policy $\mathcal{P}'$ that is a non-redundant representation of $\mathcal{P}$. To this aim, PolicyRefine identifies, for each denial δ in $\mathcal{P}$, a set of sets of inequalities RC that is a complete set of strict redundancy conditions for δ in $\mathcal{P}$, and then represents the denial δ by the extended denial $\tau(\delta, RC)$ in $\mathcal{P}'$. In the algorithm, $Atoms(\delta)$ denotes the set of all atoms occurring in the denial δ, $PredAt(\delta)$ denotes the set of standard predicate atoms, and $CompAt(\delta)$ denotes the set of comparison atoms. Moreover, $MGU(Q)$ denotes the most general unifier of the set of atoms Q.

The correctness of the algorithm is stated by the following theorem.

Theorem 2. *Let $\mathcal{P}$ be a policy and let $\mathcal{P}'$ be the extended policy returned by* PolicyRefine($\mathcal{P}$). *Then, $\mathcal{P}'$ is a non-redundant representation of $\mathcal{P}$.*

We finally notice that, given a *DL-Lite*$_A$ TBox $\mathcal{T}$ and a policy $\mathcal{P}$ that is safe w.r.t. $\mathcal{T}$, before refining $\mathcal{P}$, in our procedure we have to reformulate it by using the algorithm PerfectRef($\mathcal{T}, \mathcal{P}$) of [13], which returns relevant policy assertions implied by $\mathcal{T}$ and $\mathcal{P}$[5].

Example 8. Let $\mathcal{T}$ and $\mathcal{P}$ be as in Example 2. One can verify that the set $\mathcal{P}' =$ PolicyRefine(PerfectRef($\mathcal{T}, \mathcal{P}$)) is constituted by the following denials:

$\mathcal{P} = \{\forall w, y, d.\mathsf{type}(w, \text{`}\textit{wildcat}\text{'}) \wedge \mathsf{year}(w, y) \wedge \mathsf{doc}(d, w) \wedge 1980 < y \leq 1992 \rightarrow \bot,$
$\forall w, y.\mathsf{year}(w, y) \wedge \mathsf{doc}(d, w) \wedge y > 1992 \rightarrow \bot,$
$\forall w, d.\mathsf{doc}(d, w) \wedge \mathsf{age}(w, \text{`}\textit{Eocene}\text{'}) \rightarrow \bot\}$

□

[5] Technically speaking, PerfectRef rewrites CQs. We here adopt a variant that rewrites the positive part of each BCQ_{ineq} in the premise of a policy assertion, which provides a correct reformulation under the safe policy assumption.

5.3 Query rewriting algorithm

We now give our query rewriting technique. In the following, without loss of generality, we assume that in each denial, the arguments of an atom are always variables different to one another (the presence of the same variable or of constants can be indeed expressed through equalities). First of all, given a *DL-Lite*$_A$ TBox $\mathcal{T}$ and a policy $\mathcal{P}$ that is safe w.r.t. $\mathcal{T}$, we reformulate $\mathcal{P}$ by using the algorithm $\mathsf{PerfectRef}(\mathcal{T}, \mathcal{P})$. Then, let α and β be two atoms. We say that β is compatible with α if there exists a mapping $\mu_{\alpha/\beta}$ of the variables occurring in β to the terms occurring in α such that $\mu(\beta) = \alpha$. Given an atom α and an FO formula Φ , we denote by $\mathsf{compSet}(\alpha, \Phi)$ the set of atoms of Φ that are compatible with α. Moreover, let α be an atom, let $\mathcal{Q}$ be a set of FO formulas, and let $\succ$ be a preference relation, we denote by $\mathsf{notPreferred}(\alpha, \mathcal{Q}, \succ)$ the set of formulas $\Phi \in \mathcal{Q}$ such that there does not exist in Φ any atom β such that $\alpha \succ \beta$.

Let $\Phi = \exists \vec{x}.\alpha \wedge \beta_1 \wedge \ldots \wedge \beta_n$ be a query, we denote by $\mathsf{allDD}_i(\Phi, \alpha)$ the FOL formula $\exists \vec{y}.\mathrm{DD}_i(\beta_1) \wedge \ldots \wedge \mathrm{DD}_i(\beta_n)$ where $\vec{y}$ are the variables in $\vec{x}$ that do not occur in α and by $\mathsf{oneDC}_i(\Phi, \alpha)$ the FO formula $\mathrm{DC}_i(\beta_1) \vee \ldots \vee \mathrm{DC}_i(\beta_n)$ (of course, if $n = 0$ then $\mathsf{allDD}_i(\Phi, \alpha) = \mathit{true}$ and $\mathsf{oneDC}_i(\Phi, \alpha) = \mathit{false}$). Also, $\mathrm{DD}_0(\alpha) = \mathrm{DC}_0(\alpha) = \mathit{false}$, for each atom α. Moreover, we denote by $\mathcal{Q}_\mathcal{P}$ the set of queries returned by $\mathsf{PolicyRefine}(\mathsf{PerfectRef}(\mathcal{T}, \mathcal{P}))$.

For an atom α and a natural number $i \geq 1$, we denote by $\mathrm{DD}_i(\alpha)$ the FO formula:

$$\alpha \wedge \left(\bigwedge_{\substack{\forall q_d \in \mathsf{notPreferred}_\succ(\alpha, \mathcal{Q}_\mathcal{P}), \\ \forall \beta \in \mathsf{compSet}(\alpha, q_d)}} \forall \vec{w}. \left(\neg \mu_{\alpha/\beta}(q_d) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_d), \alpha) \right) \right)$$

Where $\vec{w}$ contains all the variables in $\mu_{\alpha/\beta}(q_d)$ that do not occur in α.

For an atom α and a natural number $i \geq 1$, we denote by $\mathrm{DC}_i(\alpha)$ the FO formula:

$$\bigvee_{\substack{\forall q_d \in \mathsf{notPreferred}_\succ(\alpha, \mathcal{Q}_\mathcal{P}), \\ \forall \beta \in \mathsf{compSet}(\alpha, q_d)}} \exists \vec{v}.\mu_{\alpha/\beta}(q_d) \wedge \mathsf{allDD}_{i-1}(\mu_{\alpha/\beta}(q_d), \alpha)$$

Given a union of BCQs Q and a prioritized CQE specification $\mathcal{E}_\succ$, we define the FO query k-$\mathsf{DDClosed}(Q, \mathcal{E}_\succ)$ as follows:

$$k\text{-}\mathsf{DDClosed}(Q, \mathcal{E}_\succ) = \bigvee_{q \in Q} \left(\bigwedge_{\alpha \in q} \mathrm{DD}_k(\alpha) \right)$$

Given a *DL-Lite*$_A$ TBox $\mathcal{T}$ and an FO query ϕ, we define $\mathsf{expand}(\mathcal{T}, \phi)$ as the FO query obtained from ϕ by replacing every atom α occurring in ϕ with its "$\mathcal{T}$-expansion" $\mathsf{expand}(\mathcal{T}, \alpha)$, where:
(*i*) if $\alpha = C(t)$, then $\mathsf{expand}(\mathcal{T}, \alpha) = \bigvee_{\mathcal{T} \models D \sqsubseteq C} D(t) \vee \bigvee_{\mathcal{T} \models \exists R \sqsubseteq C} (\exists x.R(t, x)) \vee \bigvee_{\mathcal{T} \models \exists R^- \sqsubseteq C} (\exists x.R(x, t))$.
(*ii*) if $\alpha = R(t_1, t_2)$, then $\mathsf{expand}(\mathcal{T}, \alpha) = \bigvee_{\mathcal{T} \models S \sqsubseteq R} S(t_1, t_2) \vee \bigvee_{\mathcal{T} \models \exists S^- \sqsubseteq R} S(t_2, t_1)$.

Finally, given a safe *DL-Lite*$_A$ prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, a positive integer k, and a BCQ q we define:

$$k\text{-}\mathsf{DDRew}(\mathcal{E}_\succ, q) = \mathsf{expand}(\mathcal{T}, k\text{-}\mathsf{DDClosed}(\mathsf{PerfectRef}(\mathcal{T}, q), \mathcal{E}_\succ)).$$

Notice that, for every odd i, $\text{DD}_i(\alpha) = \text{DD}_{i+1}(\alpha)$ (by definition), and thus i-DDRew$(\mathcal{E}_\succ, q) = (i{+}1)$-DDRew$(\mathcal{E}_\succ, q)$.

It is easy to see that k-DDRew$(\mathcal{E}_\succ, q)$ is an FO query. The following theorem states that, for safe prioritized *DL-Lite*$_A$ CQE specifications, kDD-Cens-Ent can always be solved by checking whether k-DDRew$(\mathcal{E}_\succ, q)$ is entailed by the ABox, which amounts to evaluating such query over the ABox. In other terms, the problem is FO rewritable.

Theorem 3. *Let $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a safe prioritized DL-Lite$_A$ CQE specification, k be a positive integer, and* censDD_k *be the k-DD censor for $\mathcal{E}_\succ$. For every ABox $\mathcal{A}$ for $\mathcal{T}$ and BCQ q, $\mathcal{T} \cup \text{cens}DD_k(\mathcal{A}) \models q$ iff $\mathcal{A} \models k$-DDRew$(\mathcal{E}_\succ, q)$.*

Proof. The proof is based on three crucial lemmas. The first recalls a property of the PerfectRef algorithm [13].

Lemma 1. *$\mathcal{T} \cup \text{cens}DD_k(\mathcal{A}) \models q$ iff $\text{cens}DD_k(\mathcal{A}) \models \text{PerfectRef}(\mathcal{T}, q)$.*

Then, we prove the following property.

Lemma 2. *Let Q be a union of BCQs. Then, $\text{cens}DD_k(\mathcal{A}) \models Q$ iff $\text{cl}(\mathcal{T}, \mathcal{A}) \models k$-DDClosed$(Q, \mathcal{E}_\succ)$.*

Proof (sketch). First, we prove inductively the following property: For every i such that $0 \leq i \leq k$, and for every atom α, $\alpha \in \text{DD}_i(\mathcal{E}_\succ, \mathcal{A})$ iff $\text{cl}(\mathcal{T}, \mathcal{A}) \models \text{DD}_k(\alpha)$ and $\alpha \in \text{DC}_i(\mathcal{E}_\succ, \mathcal{A})$ iff $\text{cl}(\mathcal{T}, \mathcal{A}) \models \text{DC}_k(\alpha)$. The base case holds since $\text{DD}_0(\mathcal{E}_\succ, \mathcal{A}) = \text{DC}_0(\mathcal{E}_\succ, \mathcal{A}) = \emptyset$ and $\text{DD}_0(\alpha) = \text{DC}_0(\alpha) =$ *false*. The inductive case follows immediately from Theorem 2, Definition 6, and the definition of the formulas $\text{DD}_i(\alpha)$ and $\text{DC}_i(\alpha)$. Then, the thesis follows immediately from the previous property and the definition of k-DDClosed$(Q, \mathcal{E}_\succ)$. ∎

The next lemma directly follows from the definition of $\text{cl}(\mathcal{T}, \mathcal{A})$ and $\text{expand}(\mathcal{T}, \phi)$.

Lemma 3. *Let ϕ be a FO query. Then, $\text{cl}(\mathcal{T}, \mathcal{A}) \models \phi$ iff $\mathcal{A} \models \text{expand}(\mathcal{T}, \phi)$.*

Then, the theorem is an immediate consequence of the above lemmas. □

Example 9. Consider the safe prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ of Example 5, and the BCQ $q = \exists x, y, z.\text{year}(x, y) \wedge \text{age}(x, z)$. We have that:

$$\begin{aligned}
1\text{-DDRew}(\mathcal{E}_\succ, q) = &\exists x, y, z.\text{year}(x, z) \wedge \text{age}(x, y) \wedge \forall w.(\neg(\text{doc}(w, x) \wedge \text{age}(x, y) \wedge \\
&y = \text{`}\textit{Eocene}\text{'}))\\
3\text{-DDRew}(\mathcal{E}_\succ, q) = &\exists x, y, z.\text{year}(x, z) \wedge \text{age}(x, y) \wedge \forall w.(\neg(\text{doc}(w, x) \wedge \text{age}(x, y) \wedge \\
&y = \text{`}\textit{Eocene}\text{'}) \vee (\exists v, u.\text{type}(x, v) \wedge v = \text{`}\textit{wildcat}\text{'} \wedge \text{year}(x, u) \wedge \\
&\text{doc}(w, x) \wedge 1980 < u \leq 1992) \vee (\exists r.\text{year}(x, r) \wedge \text{doc}(w, x) \wedge \\
&r > 1992))
\end{aligned}$$

Now, let $\mathcal{A}$ be the ABox of Example 4. It is easy to see that $\mathcal{A} \not\models 1\text{-DDRew}(\mathcal{E}_\succ, q)$, while $\mathcal{A} \models 3\text{-DDRew}(\mathcal{E}_\succ, q)$. □

The corollary below follows from Theorem 3 and the fact that evaluating an FO query over an ABox is in AC^0 in the size of the ABox (i.e., in data complexity).

Corollary 1. *kDD-Cens-Ent for safe prioritized DL-Lite$_A$ CQE specifications is in AC^0 in data complexity.*

	q_3 [5]		q_4 [4]		q_5 [6]		q_9 [5]		q_{12} [10]		q_{13} [7]		q_{14} [5]		q_{18} [9]		q_{44} [6]	
Setting	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**
$\emptyset, \emptyset$	910	207	1558	168	17254	585	1566	320	96671	5665	22541	811	141439	2553	339	1525	5078	221
$\mathcal{P}, \emptyset$	910	278	252	295	14797	825	416	331	13028	2876	9374	2861	62255	12372	311	1804	325	153
$\mathcal{P}, \succ, 1$	910	221	252	179	17254	612	416	216	96671	5933	22541	914	125656	4145	311	1384	325	112
$\mathcal{P}, \succ, 3$	910	249	521	1445	17254	749	1252	1148	96671	5378	22541	716	131791	15873	311	1416	4630	1952
$\mathcal{P}, \succ, 5$	910	242	566	8942	17254	723	1456	7715	96671	5219	22541	732	132127	1625K	311	4733	4630	522K
$\mathcal{P}, \succ, 7$	910	472	–	t.o.	17254	993	–	t.o.	96671	7691	22541	912	–	t.o.	311	5464	–	t.o.

Table 1: k-DD censor results for the six considered settings. $\emptyset, \emptyset$: empty policy and empty preference relation; $\mathcal{P}, \emptyset$: policy $\mathcal{P}$ and empty preference relation; $\mathcal{P}, \succ, i$: policy $\mathcal{P}$, preference relation $\succ$, and $k = i$, with $i \in \{1, 3, 5, 7\}$. In the time columns, "t.o." indicates a time out (30 minutes), and nK stands for $n \cdot 10^3$.

6 Experiments

For our experiments, we used the NPD benchmark for OBDA [11], which models the Norwegian Petroleum Directorate's FactPages domain. The benchmark provides an OWL 2 QL version of the NPD TBox[6] comprising 1377 axioms (over 321 concepts, 135 roles, and 233 attributes), the NPD ABox expressed in RDF with a total of around 2 millions of instances, and a set of 30 SPARQL queries.

Following the approach of [6], we reduced query answering over prioritized CQE specifications under k-DD censors to query answering in OBDA. We recall that an OBDA instance is a pair $(\mathcal{J}, D)$, where $\mathcal{J} = \langle \mathcal{T}, \mathcal{M}, \mathcal{S} \rangle$ is an OBDA specification, with TBox $\mathcal{T}$, source schema $\mathcal{S}$, and mapping $\mathcal{M}$ between $\mathcal{T}$ and $\mathcal{S}$, and D is a database for $\mathcal{S}$ [13]. In the experiments, we proceeded as follows: we used the TBox $\mathcal{T}$ of the benchmark, generated the schema $\mathcal{S}$ comprising unary and binary tables corresponding to predicates of the signature of $\mathcal{T}$ (for a total of 689 tables), and produced a database D for $\mathcal{S}$ in which the extension of each table coincides with the extension of the corresponding predicate in the (RDF) ABox $\mathcal{A}$ of the benchmark.

For each of the settings considered in our experiments, i.e., pairs with a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ and positive integer k, we produced a mapping $\mathcal{M}^k_{\mathcal{E}_\succ}$. More precisely, for each atomic concept A in $\mathcal{T}$, $\mathcal{M}^k_{\mathcal{E}_\succ}$ contains an assertion $\Phi(x) \rightsquigarrow A(x)$, where $\Phi(x)$ is the rewriting of the query $A(x)$ returned by k-DDRew, in which ontology predicates are substituted with the corresponding table symbol in $\mathcal{S}$. Analogously for atomic roles and attributes. Under this transformation, answering CQs under k-DD censor over $(\mathcal{E}_\succ, \mathcal{A})$ is equivalent to answering CQs over the OBDA instance $(\mathcal{J}, D)$, where $\mathcal{J} = \langle \mathcal{T}, \mathcal{M}^k_{\mathcal{E}_\succ}, \mathcal{S} \rangle$.

Exactly as done in [6], we executed the *conjunctive version* of 9 queries of the benchmark, i.e., q_3, q_4, q_5, q_9, q_{12}, q_{13}, q_{14}, q_{18}, and q_{44}.[7]

We analyzed six different settings. In the first one we set an empty policy (and, consequently, an empty priority relation), which corresponds to the case of standard query answering over the ontology. For the other settings, we specified a policy $\mathcal{P}$ constituted by the following denials:

[6] http://sws.ifi.uio.no/vocab/npd-v2

[7] In [6], we have extracted the conjunctive component of each such query, which in NPD contains also aggregate operators.

d_1: $\forall w, d, i.\mathit{dateWellboreEntry}(w, d) \wedge \mathit{wellboreMaxInclation}(w, i) \wedge$
$\mathit{wellboreType}(w, \text{“}\mathit{initial}\text{”}) \wedge i \neq 6 \rightarrow \bot$
d_2: $\forall c, w, d, y.\mathit{coreForWellbore}(c, w) \wedge \mathit{wellboreCompletionYear}(w, y) \wedge$
$\mathit{documentForWellbore}(d, w) \wedge y \neq 1985 \rightarrow \bot$
d_3: $\forall w, c, t, s.\mathit{wellOperator}(w, c) \wedge \mathit{taskForCompany}(t, c) \wedge$
$\mathit{wellboreCompletionYear}(w, 1985) \wedge \mathit{oilSampleTestForWellbore}(s, w) \rightarrow \bot$
d_4: $\forall w, l, d.\mathit{explorationWellboreForLicence}(w, l) \wedge \mathit{documentForWellbore}(d, w) \rightarrow \bot$
d_5: $\forall f, p, l.\mathit{Field}(f) \wedge \mathit{currentFieldOwner}(f, p) \wedge \mathit{ProductionLicence}(p) \wedge$
$\mathit{licenseeForLicence}(l, p) \rightarrow \bot$
d_6: $\forall p, f.\mathit{productionMonth}(p, 1) \wedge \mathit{productionForField}(p, f) \rightarrow \bot$

By coupling $\mathcal{P}$ with the OWL 2 QL version of the NPD TBox we obtained a safe CQE specification. In the second setting, the prioritized CQE specification contains the policy $\mathcal{P}$ illustrated above, and an empty priority relation. Notice that, according to Proposition 2, this setting is similar to the full setting considered in [6], but with a different policy. All the other settings are intended to verify the effectiveness of providing a priority relation and filtering data with a k-DD censor. In each setting we used a different odd k with $1 \leq k \leq 7$, and considered the following priority relations, which, together with the denials in $\mathcal{P}$, generate challenging scenarios for our technique.

$$\begin{aligned}
\mathit{wellboreType} &\succ \mathit{dateWellboreEntry},\\
\mathit{coreForWellbore} &\succ \mathit{documentForWellbore},\\
\mathit{licenseeForLicence} &\succ \mathit{currentFieldOwner},\\
\mathit{wellboreCompletionYear} &\succ \mathit{documentForWellbore},\\
\mathit{wellboreCompletionYear} &\succ \mathit{wellOperator},\\
\mathit{oilSampleTestForWellbore} &\succ \mathit{wellboreCompletionYear}
\end{aligned}$$

We performed the experiments through the Java API of MASTRO system [10] for OBDA on a standard laptop with an Intel i7 @2.6Ghz processor and 16GB of RAM.

Table 1 reports the result of our experiments. The column “#” under each query q_i displays the number of tuples in its evaluation, while the column “time” indicates the evaluation time in milliseconds. Finally, the length of each query, i.e., the number of atoms occurring in it, is indicated in square brackets near the query.

The values in the second row show that the policy $\mathcal{P}$ has an effect on query answering for eight of the nine queries (query q_3 is the only one not altered by the censor), hiding several answers with respect to the setting with no policy (with the only exception of queries q_5, q_{18}, answers to queries are reduced by up to one third). By introducing the priority relation, already with $k = 1$, we recover a substantial portion of the original answers for query q_{14}, whereas for q_5, q_{12}, q_{13} the recovery is even total. Interestingly, the evaluation time slightly increases w.r.t. the setting without policy but it decreases w.r.t. the setting with the policy without a priority relation. This is due to the fact that, for each atom α, when we adopt a priority relation, $\mathrm{DD}_1(\alpha)$ contains less conditions than the case with empty priority relation.

As for $k = 3$, we have a noticeable recovery of original answers for queries q_9 and q_{44}, a further increment for query q_{14}, and a small one for query q_4. In these cases the evaluation times are only slightly affected. When $k = 5$, for some queries we notice a worsening of the evaluation time, with only a limited recovery of the original answers in queries q_4, q_9, and q_{14}. With $k = 7$, query execution was feasible only for five queries, in particular those for which we already recovered all the original answers with a smaller k. For the remaining queries, we stopped the execution after 30 minutes.

We remark that in our experiments difficulties in executing queries have been encountered only for $k = 7$. However, the largest number (arguably, a considerable one) of original query answers has been recovered for $k = 1$ and $k = 3$, for which the associated evaluation times improve and worsen slightly, respectively, with respect to the setting with the policy without a priority relation.

7 Conclusions

Our experiments show applicability in the practice of our technique, and how priorities, besides being an important modeling feature for the designer, play an important role in increasing the amount of answers disclosed to the user, while still preserving confidentiality. An interesting direction for our research, leveraging the fact that priorities are specified between ontology predicates and not on facts, is investigating the problem of establishing at the intensional level the value for k which makes the k-DD censor coincide with the DD censor. We leave this aspect for future research.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. 2nd edition, 2007.
2. M. Benedikt, B. Cuenca Grau, and E. V. Kostylev. Logical foundations of information disclosure in ontology-based data integration. *AIJ*, 262:52–95, 2018.
3. M. Bienvenu and C. Bourgaux. Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation. In *Proc. of KR*, pages 141–151, 2020.
4. J. Biskup and P. A. Bonatti. Controlled query evaluation for known policies by combining lying and refusal. *AMAI*, 40(1-2):37–62, 2004.
5. P. A. Bonatti and L. Sauro. A confidentiality model for ontologies. In *Proc. of ISWC*, volume 8218 of *LNCS*, pages 17–32, 2013.
6. G. Cima, D. Lembo, L. Marconi, R. Rosati, and D. F. Savo. Controlled query evaluation in ontology-based data access. In *Proc. of ISWC*, pages 128–146, 2020.
7. G. Cima, D. Lembo, R. Rosati, and D. F. Savo. Controlled query evaluation in description logics through instance indistinguishability. In *Proc. of IJCAI*, pages 1791–1797, 2020.
8. B. Cuenca Grau, E. Kharlamov, E. V. Kostylev, and D. Zheleznyakov. Controlled query evaluation over OWL 2 RL ontologies. In *Proc. of ISWC*, volume 8218 of *LNCS*, 2013.
9. 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*, 2015.
10. 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.
11. D. Lanti, M. Rezk, G. Xiao, and D. Calvanese. The NPD benchmark: Reality check for OBDA systems. In *Proc. of EDBT*, pages 617–628, 2015.
12. D. Lembo, R. Rosati, and D. F. Savo. Revisiting controlled query evaluation in description logics. In *Proc. of IJCAI*, pages 1786–1792, 2019.
13. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
14. G. L. Sicherman, W. de Jonge, and R. P. van de Riet. Answering queries without revealing secrets. *ACM Trans. Database Syst.*, 8(1):41–59, 1983.
15. S. Staworko, J. Chomicki, and J. Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *AMAI*, 64(2-3):209–246, 2012.