Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes*

Franz Baader,[0000−0002−4049−221X] Patrick Koopmann,[0000−0001−5999−2583]
Francesco Kriegel,[0000−0003−0219−0330] Adrian Nuradiansyah[0000−0002−9047−7624]

Theoretical Computer Science, TU Dresden, Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract. The application of automated reasoning approaches to Description Logic (DL) ontologies may produce certain consequences that either are deemed to be wrong or should be hidden for privacy reasons. The question is then how to repair the ontology such that the unwanted consequences can no longer be deduced. An optimal repair is one where the least amount of other consequences is removed. Most of the previous approaches to ontology repair are of a syntactic nature in that they remove or weaken the axioms explicitly present in the ontology, and thus cannot achieve semantic optimality. In previous work, we have addressed the problem of computing optimal repairs of (quantified) ABoxes, where the unwanted consequences are described by concept assertions of the lightweight DL $\mathcal{EL}$. In the present paper, we improve on the results achieved so far in two ways. First, we allow for the presence of terminological knowledge in the form of an $\mathcal{EL}$ TBox. This TBox is assumed to be static in the sense that it cannot be changed in the repair process. Second, the construction of optimal repairs described in our previous work is best case exponential. We introduce an optimized construction that is exponential only in the worst case. First experimental results indicate that this reduces the size of the computed optimal repairs considerably.

1 Introduction

Description Logics [3] are a well-investigated family of logic-based knowledge representation languages, which are frequently used to formalize ontologies for application domains such as biology and medicine [17]. As the size of ontologies grows, the likelihood of them containing errors increases as well. This is particularly problematic if the data, stored in the ABox, are automatically extracted from text or other sources using natural language processing or machine learning. The reasoning services of DL systems [22,12,33,15], which derive implicit consequences from the explicitly represented knowledge, are not only useful once an ontology is deployed, but can also be employed for debugging purposes by exhibiting consequences that are not supposed to hold in the application domain. Another reason why one might want to remove a consequence is that

* funded by DFG in project number 430150274 and TRR 248 (cpec, grant 389792660).

it reveals private information that is supposed to be hidden [14,5]. Once such an unwanted consequence is detected, it is often not easy to see how to repair the ontology in order to get rid of this consequence. Classical repair approaches based on axiom pinpointing [31,29,27,32,21,8] compute maximal subsets of the ontology that do not have the consequence. The obtained result thus strongly depends on the syntactic form of the axioms. For example, it is well-known that, for expressive DLs, a finite set of terminological axioms can be expressed by a single axiom. If the given terminology (TBox) is of this shape, then the only possible classical repair is the empty TBox. To alleviate this problem, repair approaches have been developed that replace certain axioms by weaker ones (in the sense that they have less consequences) instead of removing them completely [18,24,34,6]. However, these approaches usually do not produce optimal repairs. In fact, it was shown in [6] that, even for the inexpressive DL $\mathcal{EL}$, optimal repairs need not exist. The abstract example given there can be rephrased as follows. Assume that the TBox defines humans to be exactly those individuals that have a human parent, and that the ABox says that Sam is a human. After we find out that Sam is in fact not human [9], we want to get rid of the latter assertion, but keep the (correct) consequences saying that Sam has an unbounded chain of ancestors (of undetermined species). If the TBox is assumed to be fixed, then there is no optimal repair of the ABox since we can add only a finite number of parent assertions.

To avoid such problems, our previous work on computing optimal repairs (formulated in the guise of achieving compliance with privacy policies) restricted the attention to the case without TBox. In [5] the ABox was additionally restricted to be a so-called instance store [19], i.e., an ABox without role assertions. The privacy policy (specifying which consequences are to be removed) was given as $\mathcal{EL}$ instance queries. In this setting, optimal repairs always exist and can be computed in exponential time, which is optimal since there may be exponentially many optimal repairs of exponential size.

In [7] these results were extended to ABoxes with role assertions. More precisely, we considered *quantified* ABoxes in which some individuals are anonymized by viewing them as existentially quantified variables. For example, assume that the ABox contains the information that Ben has a parent, Jerry, that is both rich and famous, and we want to remove the consequence $\exists \mathit{parent}.(\mathit{Rich} \sqcap \mathit{Famous})(\mathit{BEN})$. Classical repairs can be obtained by removing one of the assertions $\mathit{Rich}(\mathit{JERRY})$, $\mathit{Famous}(\mathit{JERRY})$, and $\mathit{parent}(\mathit{BEN}, \mathit{JERRY})$. If instead we replace the first assertion with $\mathit{Rich}(x)$ and $\mathit{parent}(\mathit{BEN}, x)$ for an existentially quantified variable x, then we retain more consequences. Note that we could not have used an individual name (i.e., constant) ANNE instead of x since information like $\mathit{Rich}(\mathit{ANNE})$ about Anne does not follow from the original ABox. We show in [7] that in this setting all optimal repairs can be computed by an exponential-time algorithm with access to an NP-oracle. The oracle is needed since our algorithm first computes a superset of the set of optimal repairs, from which non-optimal ones need to be removed using the (NP-complete) entailment test between (potentially exponentially large) quantified ABoxes. We also

consider a modified version of entailment (called IQ-entailment) in [7], where quantified ABoxes are compared w.r.t. which $\mathcal{EL}$ instance relationships they imply. Using this notion, no NP-oracle is needed for computing the set of all IQ-optimal repairs since IQ-entailment can be decided in polynomial time.

In the present paper, we improve on these results in two respects. On the one hand, we allow for the presence of terminological knowledge in the form of an $\mathcal{EL}$ TBox, which is assumed to be correct, and thus is not changed by the repair. To deal with a TBox, the approach from [7] for computing optimal repairs must be extended in two ways. First, the ABox needs to be saturated w.r.t. the TBox before applying our repair approach. The saturated ABox has the same consequences as the original one has together with the TBox. In our Ben and Jerry example, assume that the assertion $\mathit{Rich}(\mathit{JERRY})$ does not belong to the original ABox, but the TBox contains the axiom $\mathit{Famous} \sqsubseteq \mathit{Rich}$. Then the ABox on its own does not have the unwanted consequence $\exists \mathit{parent}.(\mathit{Rich} \sqcap \mathit{Famous})(\mathit{BEN})$, but together with the TBox it does. Saturation adds the assertion $\mathit{Rich}(\mathit{JERRY})$ to the ABox. For arbitrary TBoxes, saturation need not terminate. We consider two ways to remedy this problem: either allow for arbitrary TBoxes, but consider IQ-entailment, or use classical entailment, but consider cycle-restricted TBoxes [1]. In both cases, saturation always terminates; in the former in polynomial and in the latter in exponential time. One might be tempted to assume that, after saturation, one can simply apply the repair approach of [7] unchanged. This is not true, however, since the TBox may re-add assertions that have been removed or replaced by the repair. In our example, where $\mathit{Rich}(\mathit{JERRY})$ is replaced, but $\mathit{Famous}(\mathit{JERRY})$ is left untouched in the repair, the repaired ABox together with the TBox would still have the unwanted consequence. Thus, the repair approach needs to be changed to take this possibility into account.

On the other hand, the construction of optimal repairs described in our previous work [5,7], and extended in this paper such that it can deal with TBoxes, is best case exponential. The second contribution of this paper is the design of a new construction, both for classical and IQ-entailment, that is exponential only in the worst case. We also report on first experimental results, which indicate that this reduces the size of the computed optimal repairs considerably.

Detailed proofs of our results can be found in [4].

2 Preliminaries

Throughout this paper, we assume that Σ is a *signature*, which is a disjoint union of sets Σ_{O}, Σ_{C}, and Σ_{R} of *object names*, *concept names*, and *role names*. We use symbols t, u, v, w to denote object names, A, B to denote concept names, and r, s to denote role names, all of them possibly with sub- or superscripts.

As in [7], a *quantified ABox (qABox)* $\exists X.\mathcal{A}$ over Σ consists of a finite subset X of Σ_{O}, the elements of which are called *variables*, and a *matrix* $\mathcal{A}$, which is a finite set of *concept assertions* $A(u)$ where $u \in \Sigma_{\mathsf{O}}$ and $A \in \Sigma_{\mathsf{C}}$, and of *role assertions* $r(u, v)$ where $u, v \in \Sigma_{\mathsf{O}}$ and $r \in \Sigma_{\mathsf{R}}$. An non-variable object name in $\exists X.\mathcal{A}$ is called an *individual name*, and the set of all these names is denoted as

$\Sigma_{\mathsf{I}}(\exists X.\mathcal{A})$. We further set $\Sigma_{\mathsf{O}}(\exists X.\mathcal{A}) := \Sigma_{\mathsf{I}}(\exists X.\mathcal{A}) \cup X$. Traditional DL ABoxes are qABoxes where $X = \emptyset$; we then write $\mathcal{A}$ instead of $\exists \emptyset.\mathcal{A}$. The matrix of a qABox is such a traditional ABox.

An *interpretation* $\mathcal{I}$ of Σ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the *domain* $\Delta^{\mathcal{I}}$ is a non-empty set and the *interpretation function* $\cdot^{\mathcal{I}}$ maps each $u \in \Sigma_{\mathsf{O}}$ to an element $u^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each $A \in \Sigma_{\mathsf{C}}$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each $r \in \Sigma_{\mathsf{R}}$ to a binary relation $r^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$. The interpretation $\mathcal{I}$ of Σ is a *model* of a qABox $\exists X.\mathcal{A}$ over Σ if there is an interpretation $\mathcal{J}$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$, the interpretation functions $\cdot^{\mathcal{I}}$ and $\cdot^{\mathcal{J}}$ coincide on $\Sigma \setminus X$, and $u^{\mathcal{J}} \in A^{\mathcal{J}}$ for each $A(u) \in \mathcal{A}$ as well as $(u^{\mathcal{J}}, v^{\mathcal{J}}) \in r^{\mathcal{J}}$ for each $r(u,v) \in \mathcal{A}$.

Following [7], we define $\mathcal{EL}$ atoms and $\mathcal{EL}$ concept descriptions over Σ by simultaneous induction as follows. An $\mathcal{EL}$ *atom* is either a concept name $A \in \Sigma_{\mathsf{C}}$ or an *existential restriction* $\exists r.C$ for some role name $r \in \Sigma_{\mathsf{R}}$ and an $\mathcal{EL}$ concept description C. An $\mathcal{EL}$ *concept description* is a *conjunction* $\bigsqcap \mathcal{C}$ where $\mathcal{C}$ is a finite set of $\mathcal{EL}$ atoms. An $\mathcal{EL}$ *concept inclusion* is of the form $C \sqsubseteq D$ for $\mathcal{EL}$ concept descriptions C and D, and an $\mathcal{EL}$ *TBox* is a finite set of such concept inclusions. An $\mathcal{EL}$ *concept assertion* is an expression $C(u)$, where C is an $\mathcal{EL}$ concept description and $u \in \Sigma_{\mathsf{O}}$.

For each interpretation $\mathcal{I}$ of Σ, we extend the interpretation function $\cdot^{\mathcal{I}}$ to $\mathcal{EL}$ atoms and $\mathcal{EL}$ concept descriptions in the following manner:

- $(\exists r.C)^{\mathcal{I}} := \{\, \delta \mid \text{there exists some } \gamma \text{ such that } (\delta,\gamma) \in r^{\mathcal{I}} \text{ and } \gamma \in C^{\mathcal{I}} \,\}$,
- $(\bigsqcap \mathcal{C})^{\mathcal{I}} := \bigcap \{\, C^{\mathcal{I}} \mid C \in \mathcal{C} \,\}$ where $\bigcap \emptyset = \Delta^{\mathcal{I}}$.

The interpretation $\mathcal{I}$ is a *model* of the concept inclusion $C \sqsubseteq D$ (the concept assertion $C(u)$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($u^{\mathcal{I}} \in C^{\mathcal{I}}$), and of the TBox $\mathcal{T}$ if it is a model of each concept inclusion in $\mathcal{T}$.

To make the syntax introduced above more akin to the one usually employed for $\mathcal{EL}$, we denote the empty conjunction $\bigsqcap \emptyset$ as $\top$ (*top concept*), singleton conjunctions $\bigsqcap\{C\}$ as C, and conjunctions $\bigsqcap \mathcal{C}$ for $|\mathcal{C}| \geq 2$ as $C_1 \sqcap \ldots \sqcap C_n$, where $C_1, \ldots, C_n$ is an enumeration of the elements of $\mathcal{C}$ in an arbitrary order. Since we do not distinguish between the singleton conjunction $\bigsqcap\{C\}$ and the atom C, each atom is also a concept description. The set $\mathsf{Sub}(C)$ of *subconcepts* of an $\mathcal{EL}$ concept description C is defined as follows: $\mathsf{Sub}(A) := \{A\}$, $\mathsf{Sub}(\exists r.C) := \{\exists r.C\} \cup \mathsf{Sub}(C)$, and $\mathsf{Sub}(\bigsqcap \mathcal{C}) := \{\bigsqcap \mathcal{C}\} \cup \bigcup\{\, \mathsf{Sub}(D) \mid D \in \mathcal{C} \,\}$. The set $\mathsf{Atoms}(C)$ consists of all atoms contained in $\mathsf{Sub}(C)$. These two notions are extended to TBoxes and sets of concept assertions in the obvious way.

Let α, β be qABoxes, concept inclusions, or concept assertions (possibly not both of the same kind), and $\mathcal{T}$ an $\mathcal{EL}$ TBox. Then we write $\mathcal{I} \models \alpha$ if the interpretation $\mathcal{I}$ is a model of α. We say that α *entails* β *w.r.t.* $\mathcal{T}$ (written $\alpha \models^{\mathcal{T}} \beta$) if every model of α and $\mathcal{T}$ is a model of β. Furthermore, α and β are *equivalent w.r.t.* $\mathcal{T}$ (written $\alpha \equiv^{\mathcal{T}} \beta$), if $\alpha \models^{\mathcal{T}} \beta$ and $\beta \models^{\mathcal{T}} \alpha$. In case $\mathcal{T} = \emptyset$, we will sometimes write $\models$ instead of $\models^{\emptyset}$. If $\exists \emptyset.\emptyset \models^{\mathcal{T}} C \sqsubseteq D$, then we also write $C \sqsubseteq^{\mathcal{T}} D$ and say that C *is subsumed by* D *w.r.t.* $\mathcal{T}$; in case $\mathcal{T} = \emptyset$ we simply say that C is subsumed by D. Two $\mathcal{EL}$ concept descriptions are *equivalent w.r.t.* $\mathcal{T}$ (written $C \equiv^{\mathcal{T}} D$) if they subsume each other w.r.t. $\mathcal{T}$. We write $C \sqsubset^{\mathcal{T}} D$ to indicate that $C \sqsubseteq^{\mathcal{T}} D$, but $C \not\equiv^{\mathcal{T}} D$. If $\exists X.\mathcal{A} \models^{\mathcal{T}} C(a)$, then

a is called an *instance of* C w.r.t. $\exists X.\mathcal{A}$ and $\mathcal{T}$. For $\mathcal{EL}$, the subsumption and the instance problem are decidable in polynomial time [2]. However, entailment between qABoxes is NP-complete even w.r.t. the empty TBox [7].

We also use the reduced form C^r of $\mathcal{EL}$ concept descriptions C [23], which is obtained by removing redundant subdescriptions (see [7] for details). Adapting the results in [23], one can show that $C \equiv^{\emptyset} C^r$ and that $C \equiv^{\emptyset} D$ implies $C^r = D^r$.

3 A Tale of Two Entailments

DL-based ontologies are usually accessed through appropriate query languages, where for the purpose of this paper it is sufficient to assume that a query language is given by a fragment of first-order logic. Instead of comparing ontologies w.r.t. the models they have, it thus makes sense to compare them w.r.t. the answers to queries they entail [25]. Given such a query language QL and an $\mathcal{EL}$ TBox $\mathcal{T}$, we say that the qABox $\exists X.\mathcal{A}$ QL-*entails* the qABox $\exists Y.\mathcal{B}$ *w.r.t.* $\mathcal{T}$ (written $\exists X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$) if for each query $\varphi(x_1,\dots,x_k) \in \mathsf{QL}$ and each tuple of individuals $(a_1,\dots,a_k)$ we have that $\mathcal{T} \wedge \exists Y.\mathcal{B} \models \varphi(a_1,\dots,a_k)$ implies $\mathcal{T} \wedge \exists X.\mathcal{A} \models \varphi(a_1,\dots,a_k)$, where we view the TBox and the ABox as first-order formulae and $\models$ is classical first-order entailment (see [25] for more details). We say that two qABox are QL-*equivalent w.r.t.* $\mathcal{T}$ if they QL-entail each other w.r.t. $\mathcal{T}$, and denote this equivalence relation as $\equiv^{\mathcal{T}}_{\mathsf{QL}}$.

For $\mathcal{EL}$ ontologies, one usually considers instance queries (IQ) or conjunctive queries (CQ). The former are given by $\mathcal{EL}$ concept descriptions, viewed as first-order formulae with one free variable. The latter are basically qABoxes of the form $\exists X.\mathcal{A}$, but with the elements of $\Sigma_{\mathsf{I}}(\exists X.\mathcal{A})$ viewed as free variables. Replacing these free variables with a tuple of individuals thus yields a qABox in the sense introduced above. In particular, this means that CQ-entailment corresponds to entailment of the same qABoxes (see [7] for more details regarding the connection between conjunctive queries and qABoxes).

3.1 Classical Entailment and CQ-Entailment

Due to the close connection between conjunctive queries and qABoxes mentioned above, it is easy to see that the classical entailment relation $\models^{\mathcal{T}}$ between qABoxes, as introduced in the previous section, actually coincides with CQ-entailment $\models^{\mathcal{T}}_{\mathsf{CQ}}$. To keep the notation more uniform and to distinguish this kind of entailment explicitly from IQ-entailment, we will usually talk about CQ-entailment and write $\models^{\mathcal{T}}_{\mathsf{CQ}}$.

Whenever we compare two qABoxes $\exists X.\mathcal{A}$ and $\exists Y.\mathcal{B}$, we assume without loss of generality that they are *renamed apart*, which means that X is disjoint with $\Sigma_{\mathsf{O}}(\exists Y.\mathcal{B})$ and Y is disjoint with $\Sigma_{\mathsf{O}}(\exists X.\mathcal{A})$, and we further assume that the two qABoxes speak about the same set of individual names $\Sigma_{\mathsf{I}} := \Sigma_{\mathsf{I}}(\exists X.\mathcal{A}) \cup \Sigma_{\mathsf{I}}(\exists Y.\mathcal{B})$. For the case of an empty TBox, it was shown in [7] that $\exists X.\mathcal{A} \models^{\emptyset}_{\mathsf{CQ}} \exists Y.\mathcal{B}$ iff there is a homomorphism from $\exists Y.\mathcal{B}$ to $\exists X.\mathcal{A}$. A *homomorphism* from $\exists Y.\mathcal{B}$ to $\exists X.\mathcal{A}$ is a mapping $h\colon \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) \to \Sigma_{\mathsf{O}}(\exists X.\mathcal{A})$ such that $h(a) = a$

$\sqcap$-rule. If $(C_1 \sqcap \cdots \sqcap C_n)(t) \in \mathcal{A}$, then remove this assertion from $\mathcal{A}$, and add the assertions $C_1(t), \cdots, C_n(t)$ to $\mathcal{A}$.

$\exists$-rule. If $(\exists r.C)(t) \in \mathcal{A}$, then remove this assertion from $\mathcal{A}$, add the two assertions $r(t,x)$ and $C(x)$ to $\mathcal{A}$, and add x to X, where x is a fresh variable not occurring in $\mathcal{A}$ or X.

$\sqsubseteq$-rule. If $t \in \Sigma_{\mathsf{O}}(\exists X.\mathcal{A})$, $C \sqsubseteq D \in \mathcal{T}$, $\mathcal{A} \models C(t)$, and $\mathcal{A} \not\models D(t)$, then add the assertion $D(t)$ to $\mathcal{A}$.

The $\sqcap$-rule has highest priority and the $\sqsubseteq$-rule has lowest priority.

Fig. 1: The CQ-saturation rules.

for each $a \in \Sigma_{\mathsf{I}}$, $A(h(u)) \in \mathcal{A}$ for each $A(u) \in \mathcal{B}$, and $r(h(u), h(v)) \in \mathcal{A}$ for each $r(u,v) \in \mathcal{B}$. In order to obtain a similar characterization of entailment for the case of a non-empty TBox $\mathcal{T}$, we need to saturate the given qABox w.r.t. $\mathcal{T}$.

Basically, this saturation performs what is called *the chase* in the database community [26,20,10]. Given an $\mathcal{EL}$ TBox $\mathcal{T}$ and a qABox $\exists X.\mathcal{A}$, it extends the ABox by new assertions that are implied by the TBox. The rules that realize this are described in Fig. 1. Their rôle is two-fold: whereas the $\sqsubseteq$-rule adds new concept assertions that are implied by the ABox together with the TBox, the other two rules break down the complex concept assertions added by this rule into smaller parts.

In general, applying these rules need not terminate; e.g., if applied to the qABox $\exists\emptyset.\{A(a)\}$ for the TBox $\{A \sqsubseteq \exists r.A\}$. There are various sufficient conditions that guarantee termination of the chase [13]. Here, we use a condition introduced in [1] in the context of unification in $\mathcal{EL}$.

Definition 1. *The $\mathcal{EL}$ TBox $\mathcal{T}$ is* cycle-restricted *if there is no non-empty sequence of role names $r_1, \ldots, r_k$ and $\mathcal{EL}$ concept description C such that $C \sqsubseteq^{\mathcal{T}} \exists r_1. \cdots \exists r_k.C$.*

As shown in [1], it can be decided in time polynomial whether a given $\mathcal{EL}$ TBox is cycle-restricted or not. For cycle-restricted TBoxes, CQ-saturation always terminates.

Theorem 2. *Let $\mathcal{T}$ be a cycle-restricted $\mathcal{EL}$ TBox and $\exists X.\mathcal{A}$ a qABox. Then exhaustive application of the CQ-saturation rules terminates in exponential time in the size of $\exists X.\mathcal{A}$ and $\mathcal{T}$, and yields a qABox $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$ such that the following statements are equivalent for all qABoxes $\exists Y.\mathcal{B}$:*

- $\exists X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{CQ}} \exists Y.\mathcal{B}$,
- $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A}) \models^{\emptyset}_{\mathsf{CQ}} \exists Y.\mathcal{B}$,
- *there is a homomorphism from $\exists Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$.*

We can show that there are examples where the CQ-saturation of a qABox w.r.t. a cycle-restricted TBox is of exponential size, and thus its computation must take exponential time. Nevertheless, the entailment relation $\models^{\mathcal{T}}_{\mathsf{CQ}}$ can still be decided within NP by adapting results for conjunctive query answering in $\mathcal{EL}$ [30].

$\sqcap$-rule. If $(C_1 \sqcap \ldots \sqcap C_n)(t) \in \mathcal{A}$, then remove this assertion from $\mathcal{A}$ and add the assertions $C_1(t), \ldots, C_n(t)$ to $\mathcal{A}$.

$\exists$-rule. If $(\exists r.C)(t) \in \mathcal{A}$, then remove this assertion from $\mathcal{A}$, add the two assertions $r(t, x_C)$ and $C(x_C)$ to $\mathcal{A}$, and add x_C to X if it is not already there.

$\sqsubseteq$-rule. If $t \in \Sigma_{\mathsf{O}}(\exists X.\mathcal{A})$, $C \sqsubseteq D \in \mathcal{T}$, $\mathcal{A} \models C(t)$, and $\mathcal{A} \not\models D(t)$, then add the assertion $D(t)$ to $\mathcal{A}$.

The $\sqcap$-rule has higher precedence than the $\exists$-rule, and the latter has higher precedence than the $\sqsubseteq$-rule.

Fig. 2: The IQ-saturation rules.

3.2 IQ-Entailment

Recall that the qABox $\exists X.\mathcal{A}$ IQ-entails the qABox $\exists Y.\mathcal{B}$ w.r.t. the $\mathcal{EL}$ TBox $\mathcal{T}$ if every concept assertion $C(a)$ entailed w.r.t. $\mathcal{T}$ by the latter is also entailed w.r.t. $\mathcal{T}$ by the former. In the following we assume again that these two qABoxes are renamed apart. For the case of an empty TBox, it was shown in [7] that $\exists X.\mathcal{A} \models^{\emptyset}_{\mathsf{IQ}} \exists Y.\mathcal{B}$ iff there is a simulation from $\exists Y.\mathcal{B}$ to $\exists X.\mathcal{A}$. A *simulation* from $\exists Y.\mathcal{B}$ to $\exists X.\mathcal{A}$ is a relation $\mathfrak{S} \subseteq \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) \times \Sigma_{\mathsf{O}}(\exists X.\mathcal{A})$ such that $(a, a) \in \mathfrak{S}$ for each $a \in \Sigma_{\mathsf{I}}$ and, for each $(u, v) \in \mathfrak{S}$, $A(u) \in \mathcal{B}$ implies $A(v) \in \mathcal{A}$ and $r(u, u') \in \mathcal{B}$ implies that there exists an object $v' \in \Sigma_{\mathsf{I}} \cup X$ such that $(u', v') \in \mathfrak{S}$ and $r(v, v') \in \mathcal{A}$. Since checking the existence of a simulation can be done in polynomial time [16], we conclude that IQ-entailment between qABoxes can be decided in polynomial time for the case of an empty TBox.

To extend these results to the case of a non-empty TBox, we again need to saturate the ABox w.r.t. the TBox. But now the saturation rules, given in Fig. 2, are more parsimonious w.r.t. the introduction of new objects. To be more precise, for each existential restriction $\exists r.C \in \mathsf{Sub}(\mathcal{T})$, we assume that x_C is a fresh variable not contained in the initial qABox $\exists X.\mathcal{A}$. When applying the $\exists$-rule to an assertion of the form $(\exists r.C)(t)$, we always use this variable for the successor object. Due to this restriction, IQ-saturation always terminates, i.e., it is not necessary to impose any restrictions on the TBox. Also note that IQ-saturation basically generates a qABox representation of what is called the *canonical model* in [25, Section 5.2].

Theorem 3. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and $\exists X.\mathcal{A}$ a qABox. Then exhaustive application of the IQ-saturation rules terminates in polynomial time in the size of $\exists X.\mathcal{A}$ and $\mathcal{T}$, and yields a qABox $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$ such that the following statements are equivalent for all qABoxes $\exists Y.\mathcal{B}$:*

- $\exists X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{IQ}} \exists Y.\mathcal{B}$,
- $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}) \models^{\emptyset}_{\mathsf{IQ}} \exists Y.\mathcal{B}$,
- *there is a simulation from $\exists Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$.*

Since $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$ can be computed in polynomial time and the existence of a simulation can be decided in polynomial time, this shows that the entailment relation $\models^{\mathcal{T}}_{\mathsf{IQ}}$ can be decided in polynomial time.

4 Canonical Repairs

We specify what is to be repaired by a finite set of $\mathcal{EL}$ concept assertions, which we call a repair request. A repair is a qABox that does not have any of these assertions as a consequence. This generalizes previous repair approaches [6] in that more than one consequence specified as unwanted is removed in one step. It also encompasses the notion of a privacy policy, as introduced in [7], which specifies forbidden concepts, with the meaning that one should not be able to derive that any of the individuals occurring in the qABox is an instance of such a concept. We assume that the TBox is static (i.e., may not be changed by the repair) and consider both CQ- and IQ-entailment for comparing qABoxes.

Definition 4. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and $\mathsf{QL} \in \{\mathsf{CQ}, \mathsf{IQ}\}$.*

- *An $\mathcal{EL}$* repair request *is a finite set of $\mathcal{EL}$ concept assertions.*
- *Given a qABox $\exists X.\mathcal{A}$ and an $\mathcal{EL}$ repair request $\mathcal{R}$, a* QL-repair of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$ *is a qABox $\exists Y.\mathcal{B}$ such that $\exists X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$ and $\exists Y.\mathcal{B} \not\models^{\mathcal{T}} C(a)$ for all $C(a) \in \mathcal{R}$.*
- *Such a repair $\exists Y.\mathcal{B}$ is* optimal *if there is no QL-repair $\exists Z.\mathcal{C}$ of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$ such that $\exists Z.\mathcal{C} \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$ and $\exists Z.\mathcal{C} \not\equiv^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$.*

Intuitively, a repair is a qABox that has no new consequences of the specified type (instance relationships or answers to conjunctive queries), and no longer has the consequences forbidden by the repair request. In an optimal repair, a minimal amount of consequences of the specified type is lost. Since there are different options for what to change when repairing a qABox, there may exist several non-equivalent optimal repairs.

In the following, let $\mathsf{QL} \in \{\mathsf{CQ}, \mathsf{IQ}\}$ and let $\mathcal{T}$ be a fixed TBox, which is assumed to be cycle-restricted if $\mathsf{QL} = \mathsf{CQ}$. In addition, let $\mathcal{R}$ be a repair request and $\exists X.\mathcal{A}$ be the qABox to be QL-repaired for $\mathcal{R}$ w.r.t. $\mathcal{T}$. We assume that $\mathcal{R}$ does not contain an assertion of the form $C(a)$ such that $\top \sqsubseteq^{\mathcal{T}} C$ since the presence of such an assertions would preclude the existence of a repair. If $\mathcal{R}$ satisfies this restriction, then the empty qABox $\exists \emptyset.\emptyset$ is always a repair. However, as mentioned in the introduction, this does not imply that there is an optimal repair. We will show that, for the case of IQ-entailment, optimal repairs always exist. For CQ-entailment, this is the case if the TBox $\mathcal{T}$ is cycle-restricted. In both cases, the set of optimal repairs covers all repairs in the sense that each repair is entailed by some optimal repair.

As mentioned in the introduction, to deal with TBoxes, the approach for computing so-called canonical repairs from [7] needs to be adapted in two ways. First, one needs to QL-saturate the given qABox w.r.t. the TBox. Second, when computing canonical repairs from $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$, the construction needs to ensure that the TBox does not reintroduce consequences that have been removed by the repair. The main idea underlying the construction of canonical repairs is to introduce variables as copies of the objects occurring in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$. Such a variable is of the form $y_{u,\mathcal{K}}$, where the first component of the subscript says that this is a copy of the object u. The second component $\mathcal{K}$ is a set of atoms, with

the intuitive meaning that $y_{u,\mathcal{K}}$ must *not* be an instance of any element of $\mathcal{K}$. To avoid introducing unnecessary copies, certain restrictions were imposed in [7] on the sets $\mathcal{K}$. We add a further restriction that takes care of the TBox.

To be more precise, let $\mathsf{Sub}(\mathcal{R},\mathcal{T})$ be the set of subconcepts of concept descriptions occurring in $\mathcal{R}$ or $\mathcal{T}$, and let $\mathsf{Atoms}(\mathcal{R},\mathcal{T})$ be the set of atoms occurring in $\mathsf{Sub}(\mathcal{R},\mathcal{T})$. The set $\mathcal{K}$ in a variable $y_{u,\mathcal{K}}$ must be a repair type for u.

Definition 5. *Let $\exists Y.\mathcal{B} \coloneqq \mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ and let u be an object name occurring in $\mathcal{B}$. A* repair type *for u is a subset $\mathcal{K}$ of $\mathsf{Atoms}(\mathcal{R},\mathcal{T})$ that satisfies the following:*

1. *$\mathcal{B} \models^{\emptyset} C(u)$ for each atom $C \in \mathcal{K}$,*
2. *if C, D are distinct atoms in $\mathcal{K}$, then $C \not\sqsubseteq^{\emptyset} D$,*
3. *$\mathcal{K}$ is* premise-saturated *w.r.t. $\mathcal{T}$, i.e., for all $C \in \mathsf{Sub}(\mathcal{R},\mathcal{T})$ with $\mathcal{B} \models^{\emptyset} C(u)$ and $C \sqsubseteq^{\mathcal{T}} D$ for some $D \in \mathcal{K}$, there is $E \in \mathcal{K}$ such that $C \sqsubseteq^{\emptyset} E$.*

The first two conditions coincide with the ones in [7]. Basically, 1. says that we only need to remove instance relationships explicitly if they are really there. Condition 2. corresponds to the fact that preventing $D(y_{u,\mathcal{K}})$ as a consequence also prevents $C(y_{u,\mathcal{K}})$ if D subsumes C, and thus $C \in \mathcal{K}$ would be redundant if $D \in \mathcal{K}$. Condition 3. ensures that instance relationships that are removed due to $\mathcal{K}$ cannot be re-introduced by the TBox. It is easy to see that the set of repair types for u can be computed in exponential time.

Similarly to the approach in [7], canonical repairs are induced by seed functions. Such a function determines, for each individual, which instance relationships should be prevented in order to obtain a repair.

Definition 6. *A* repair seed function *is a function s that maps each individual name $b \in \Sigma_{\mathsf{I}}(\exists X.\mathcal{A})$ to a repair type $s(b)$ for b that satisfies the following:*

- *if $C(b) \in \mathcal{R}$ and $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}) \models C(b)$, then $s(b)$ contains an atom D such that $C \sqsubseteq^{\emptyset} D$.*

Using our general assumption that the repair request $\mathcal{R}$ does not contain a concept assertion $C(a)$ with $\top \sqsubseteq^{\mathcal{T}} C$, we can show that there is always at least one repair seed function. Each repair seed function induces a repair as follows.

Definition 7. *Given a repair seed function s, we define the* canonical QL-repair $\mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s)$ induced by s *as the qABox $\exists Y.\mathcal{B}$ where*

1. *the set Y consists of the variables $y_{u,\mathcal{K}}$ for all object names u occurring in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ and all repair types $\mathcal{K}$ for u, except for the case where u is an individual name and $\mathcal{K} = s(u)$, and*
2. *the matrix $\mathcal{B}$ consists of the following assertions, where we use $y_{b,s(b)}$ as a synonym for the individual name b:*
   - *$A(y_{u,\mathcal{K}}) \in \mathcal{B}$ for each concept assertion $A(u)$ in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ such that $A \notin \mathcal{K}$,*
   - *$r(y_{u,\mathcal{K}}, y_{v,\mathcal{L}}) \in \mathcal{B}$ for each role assertion $r(u,v)$ in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ such that the following holds for each $\exists r.C \in \mathcal{K}$: if the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ entails $C(v)$, then the set $\mathcal{L}$ contains an atom that subsumes C.*

Our construction of canonical repairs based on seed functions is sound and complete in the following sense.

Proposition 8. *For each repair seed function s, the induced canonical repair $\mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s)$ is a QL-repair of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$. Conversely, if $\exists Y.\mathcal{B}$ is a QL-repair of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$, then there is a repair seed function s such that $\mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s) \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$.*

We define the set of all canonical QL-repairs of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$ as

$$\mathsf{Repairs}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, \mathcal{R}) := \{\, \mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s) \mid s \text{ is a repair seed function} \,\}.$$

As an easy consequence of Proposition 8 we obtain that $\mathsf{Repairs}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, \mathcal{R})$ contains all optimal repairs (up to equivalence). However, as in the case without a TBox, it may also contain non-optimal repairs [7]. To compute the set of optimal repairs, one thus needs to remove such non-optimal elements from $\mathsf{Repairs}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, \mathcal{R})$. Since the entailment test required for this is NP-complete for $\mathsf{QL} = \mathsf{CQ}$ and polynomial for $\mathsf{QL} = \mathsf{IQ}$, we obtain the following theorem.

Theorem 9. *There is a (deterministic) algorithm that computes the set of all optimal QL-repairs of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$ and runs in exponential time. If $\mathsf{QL} = \mathsf{CQ}$, then this algorithm needs access to an NP oracle, whereas no such oracle is required for $\mathsf{QL} = \mathsf{IQ}$.*

5 Optimized Repairs

The construction of the canonical repair induced by a seed function described in the previous section usually introduces an exponential number of copies for the objects occurring in the saturated qABox. The following example demonstrates that this is not always necessary to obtain an optimal repair.

Example 10. Let $\mathcal{T} := \emptyset$ and consider the repair request $\{(\exists r.(A_1 \sqcap \ldots \sqcap A_n))(a)\}$ for the qABox $\exists\{x\}.\{r(a,x), A_1(x), \ldots, A_n(x)\}$. There is only one repair seed function s, which assigns $\{\exists r.(A_1 \sqcap \ldots \sqcap A_n)\}$ to a. Both for the CQ and the IQ case, the canonical repair induced by s contains 2^n copies of x, namely all the variables $y_{x,\mathcal{K}}$ for $\mathcal{K} \subseteq \{A_1, \ldots, A_n\}$. However, most of these copies are redundant. In fact, we will see below that there are optimal repairs equivalent to the canonical one that contain only linearly many variables in n, both for the CQ and the IQ case.

The idea is now to construct, for a given seed function, a set of variables that is a (hopefully small) subset of the set Y introduced in Definition 7, which is nevertheless sufficient to obtain a repair equivalent to the canonical one. Note, however, that in general an exponential blow-up cannot be avoided, as already shown in [5] for the case of $\mathcal{EL}$ instance stores. Throughout this section, we assume that QL, $\mathcal{T}$, $\mathcal{R}$, and $\exists X.\mathcal{A}$ satisfy the properties assumed in the previous section. In addition, we assume that the repair request $\mathcal{R}$ is *reduced*, i.e., every

concept occurring in a concept assertion in $\mathcal{R}$ is reduced, and if $\mathcal{R}$ contains $C(a)$ and $D(a)$ for distinct concept descriptions C, D, then $C \not\sqsubseteq^{\emptyset} D$, and we further assume that each concept occurring in the TBox $\mathcal{T}$ is reduced. Before we can describe our construction of the set of relevant variables, we must introduce some notation and show an auxiliary result.

Given two sets of concept descriptions $\mathcal{K}$ and $\mathcal{L}$, we say that $\mathcal{L}$ *covers* $\mathcal{K}$ (written $\mathcal{K} \leq \mathcal{L}$) if each concept in $\mathcal{K}$ is subsumed by some concept in $\mathcal{L}$.

Now, let s be a repair seed function and set $\exists Y.\mathcal{B} \coloneqq \mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s)$. Recall that, according to Definition 7, a role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ belongs to the matrix $\mathcal{B}$ iff the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ contains the role assertion $r(t,u)$ and the repair type $\mathcal{L}$ covers the set $\mathsf{Succ}(\mathcal{K}, r, u) \coloneqq \{\, C \mid \exists r.C \in \mathcal{K}$ and the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ entails $C(u) \,\}$.

If $\mathcal{L}$ does not satisfy this requirement, there might be another repair type $\mathcal{L}'$ such that the canonical repair contains the assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}'})$, and thus our optimized repair needs to contain an appropriate variable to which $y_{u,\mathcal{L}'}$ can be mapped by a homomorphism or simulation. We generate such variables by looking for repair types $\mathcal{M}$ that cover both $\mathcal{L}$ and $\mathsf{Succ}(\mathcal{K}, r, u)$. The set of all such repair types can effectively be computed, though it might be empty. For our purposes, it is sufficient to use only the ones that are minimal w.r.t. the cover relation $\leq$.

Lemma 11. *The set of all $\leq$-minimal repair types for u that cover $\mathcal{L} \cup \mathsf{Succ}(\mathcal{K}, r, u)$ can be computed in exponential time.*

In general, this computation may produce exponentially many repair types, but this is not always the case. For instance, consider $a = y_{a,s(a)}$ and $y_{x,\emptyset}$ in Example 10. We have $\mathsf{Succ}(s(a), r, x) = \{A_1 \sqcap \ldots \sqcap A_n\}$ and thus the assertion $r(a, y_{x,\emptyset})$ is not in $\mathcal{B}$ since $\emptyset$ clearly does not cover $\mathsf{Succ}(s(a), r, x)$. The $\leq$-minimal repair types covering $\mathsf{Succ}(s(a), r, x)$ are exactly the sets $\{A_i\}$ for $i = 1, \ldots, n$.

In the following, we construct a sequence $Y_0, Y_1, \ldots, Y_m$ of subsets Y_i of Y such that $\exists Y.\mathcal{B}$ is QL-equivalent to its sub-qABox $\exists Y_m.\mathcal{B}_m$ where $\mathcal{B}_m$ contains only those assertions in $\mathcal{B}$ involving object names in $\Sigma_{\mathsf{I}} \cup Y_m$. Recall that we use $y_{a,s(a)}$ as synonyms for the individuals $a \in \Sigma_{\mathsf{I}}$.

We start with the set Y_0, which is empty if $\mathsf{QL} = \mathsf{IQ}$, and equal to the set $\{\, y_{t,\emptyset} \mid t$ is an object name occurring in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A}) \,\}$ if $\mathsf{QL} = \mathsf{CQ}$.

The subsequent sets are obtained by exhaustively applying one of the following rules, depending on whether $\mathsf{QL} = \mathsf{CQ}$ or $\mathsf{QL} = \mathsf{IQ}$.

CQ-construction rule. If $y_{t,\mathcal{K}}$ and $y_{u,\mathcal{L}}$ are elements of $\Sigma_{\mathsf{I}} \cup Y_i$, the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$ contains the role assertion $r(t,u)$, the repair type $\mathcal{L}$ does not cover $\mathsf{Succ}(\mathcal{K}, r, u)$, and $\mathcal{M}$ is a $\leq$-minimal repair type for u that covers $\mathcal{L} \cup \mathsf{Succ}(\mathcal{K}, r, u)$, but $y_{u,\mathcal{M}}$ is not contained in $\Sigma_{\mathsf{I}} \cup Y_i$, then set $Y_{i+1} \coloneqq Y_i \cup \{y_{u,\mathcal{M}}\}$.

IQ-construction rule. If $y_{t,\mathcal{K}}$ is an element of $\Sigma_{\mathsf{I}} \cup Y_i$, the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$ contains the role assertion $r(t,u)$, and $\mathcal{M}$ is a $\leq$-minimal repair type for u that covers $\mathsf{Succ}(\mathcal{K}, r, u)$, but $y_{u,\mathcal{M}}$ is not contained in $\Sigma_{\mathsf{I}} \cup Y_i$, then set $Y_{i+1} \coloneqq Y_i \cup \{y_{u,\mathcal{M}}\}$.

The sets Y_i are all subsets of the set Y of variables in the canonical repair. Since each rule application adds a variable, the exhaustive application of rules must terminate after finitely many steps with a set of variables $Y_m \subseteq Y$.

Let us illustrate this construction using Example 10, first for the IQ case. We have $a = y_{a,s(a)} \in \Sigma_\mathsf{I}$ and the assertion $r(a,x)$ belongs to the saturation, which is equal to the original qABox. As mentioned above, the $\leq$-minimal repair types covering $\mathsf{Succ}(s(a), r, x)$ are exactly the sets $\{A_i\}$ for $i = 1, \ldots, n$. Thus, repeated applications of the IQ-construction rule add the variables $y_{x,\{A_i\}}$, and the construction ends with $Y_m^{\mathsf{IQ}} = \{\, y_{x,\{A_i\}} \mid i = 1, \ldots, n \,\}$. In the CQ case, the initial set of variables is $Y_0^{\mathsf{CQ}} = \{y_{a,\emptyset}, y_{x,\emptyset}\}$. In this example, the CQ-construction rule then generates the same variables as the IQ rule, though this need not be the case in general. We end up with the final set $Y_m^{\mathsf{IQ}} \cup Y_0^{\mathsf{CQ}}$.

Definition 12. *Let s be a repair seed function and $Y_m \subseteq Y$ be the set of variables obtained by an exhaustive application of the QL-construction rule. The* optimized QL-repair *of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$ induced by s, denoted by $\mathsf{orep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s)$, is the qABox $\exists Y_m.\mathcal{B}_m$ where the matrix $\mathcal{B}_m$ contains all assertions in $\mathcal{B}$ involving only object names in $\Sigma_\mathsf{I} \cup Y_m$.*

Note that, to compute $\mathcal{B}_m$, we need not compute the larger matrix $\mathcal{B}$ first. Instead, we just apply the definition of the matrix in Definition 7 to the object names in $\Sigma_\mathsf{I} \cup Y_m$.

In our example, the optimized IQ-repair is the qABox $\exists Y_m^{\mathsf{IQ}}.\mathcal{B}_m$ with

$$\mathcal{B}_m = \{\, r(a, y_{x,\{A_i\}}) \mid 1 \leq i \leq n \,\} \cup \{\, A_j(y_{x,\{A_i\}}) \mid j \neq i \text{ and } 1 \leq i, j \leq n \,\}.$$

In the optimized CQ-repair, the quantifier prefix additionally contains the variables $y_{a,\emptyset}$ and $y_{x,\emptyset}$, and the matrix additionally contains the assertions $r(y_{a,\emptyset}, y_{x,\emptyset})$ and $A_i(y_{x,\emptyset})$ for $i = 1, \ldots, n$. Note that, without these assertions, the positive answer to the Boolean conjunctive query $\exists y, z.\,(r(y,z) \wedge A_1(z) \wedge \ldots \wedge A_n(z))$ would be lost.

Coming back to the general case, we first observe that the canonical QL-repair induced by s QL-entails the optimized QL-repair induced by s due to the inclusion relationship between these two qABoxes. The entailment in the other direction also holds, but this is harder to show, in particular for $\mathsf{QL} = \mathsf{CQ}$.

Proposition 13. *For each repair seed function s, the optimized QL-repair induced by s QL-entails the canonical QL-repair induced by s.*

Proof sketch. For $\mathsf{QL} = \mathsf{IQ}$, the proposition can be proved by showing that the following relation $\mathfrak{S}$ is a simulation from $\exists Y.\mathcal{B}$ to $\exists Y_m.\mathcal{B}_m$:

$$\mathfrak{S} \coloneqq \{\, (y_{t,\mathcal{K}}, y_{t,\mathcal{K}'}) \mid y_{t,\mathcal{K}} \in \Sigma_\mathsf{O}(\exists Y.\mathcal{B}),\ y_{t,\mathcal{K}'} \in \Sigma_\mathsf{O}(\exists Y_m.\mathcal{B}_m),\ \text{and } \mathcal{K}' \leq \mathcal{K} \,\}.$$

For $\mathsf{QL} = \mathsf{CQ}$, we introduce a sequence of mappings $h_0, h_1, \ldots, h_n \colon \Sigma_\mathsf{O}(\exists Y.\mathcal{B}) \to \Sigma_\mathsf{O}(\exists Y_m.\mathcal{B}_m)$, starting with $h_0(y_{t,\mathcal{K}}) = y_{t,s(t)}$ if $t \in \Sigma_\mathsf{I}$ and $s(t) \leq \mathcal{K}$ and $h_0(y_{t,\mathcal{K}}) = y_{t,\emptyset}$ otherwise. The initial mapping h_0 need not be a homomorphism

since role assertions may not be preserved. In the step-wise construction of the mappings h_i such defects are corrected, one by one. We can show that this construction always terminates after finitely many steps, yielding a homomorphism h_n from $\exists Y.\mathcal{B}$ to $\exists Y_m.\mathcal{B}_m$. □

Summing up, we have thus shown the following theorem, which implies that the optimized repairs also satisfy the properties stated in Proposition 8.

Theorem 14. *For each repair seed function s, the canonical* QL*-repair induced by s and the optimized* QL*-repair induced by s are* QL*-equivalent.*

6 Evaluation

To find out whether the repair approaches introduced in this paper are in principle viable for non-trivial ontologies, we made experiments for both IQ and CQ-repairs with a first, rather unoptimized implementation. In addition to checking how often the implementation was able to compute a repair within a certain timeout, we also compared the sizes of optimized repairs with those of canonical repairs. We considered two different repair scenarios: repairing a single unwanted consequence for a single individual (S1), and repairing a single unwanted consequence for 10% of the individuals occurring in the ABox (S2). We report here the main results—more details and discussions can be found in [4].

As corpus for our evaluation, we chose the ontologies used in the 2015 OWL Reasoner Competition for the track OWL EL Realisation [28], since they contain a substantial amount of ABox assertions. These 109 ontologies were converted into pure $\mathcal{EL}$ by applying standard transformations and afterwards filtering out unsupported axioms. From these ontologies, we kept those that had at most 100,000 axioms in total. The resulting corpus contained 80 ontologies.

We implemented our methods in Java, using the OWL-API[1] for parsing OWL ontologies, and ELK [22] for precomputing any subsumption relationships entailed with and without the TBox potentially relevant for our repair approach. The code is available online.[2] All experiments were performed on an Intel(R) Core(TM) i5-4590 CPU with 4 cores and 32 GB RAM, of which we assigned 16 GB as maximal heap space to the Java VM.

Since it is a precondition of our repair approach, we first saturated the ontologies using the IQ-saturation rules of Figure 2, and the CQ-saturation rules of Figure 1. The CQ-saturation rules were implemented using the rule engine VLog [11] through the Java facade Rulewerk.[3] As CQ-saturation only terminates for cycle-restricted TBoxes, we only considered those ontologies for the CQ-saturation whose IQ-saturation did not introduce cycles between introduced variables. We used a timeout of 60 minutes for every saturation. This way, we successfully computed IQ-saturations of every ontology, and 62 CQ-saturations.

[1] `http://owlapi.sourceforge.net`
[2] `https://github.com/de-tu-dresden-inf-lat/abox-repairs-wrt-static-tbox`
[3] `https://github.com/knowsys/rulewerk`

The size of the saturated ABox was usually not much larger than that of the original one, and always less than two orders of magnitude larger. Interestingly, the successful CQ-saturations were rarely larger than the IQ-saturations, and often even of the same size, because no variables were added.

Scenario S1 was about repairing a single faulty entailment $\mathcal{A} \models^{\mathcal{T}} C(a)$. Since we did not have information about whether any entailments from the considered ontologies are faulty, we generated such assertions randomly. For this, we looked at entailments of the form $\mathcal{A} \models^{\mathcal{T}} C(a)$, where $C \in \mathsf{Sub}(\mathcal{T})$. To make the repair requests more interesting, we furthermore required that C is not of the form A or $\exists r.\top$, where A is a concept name. This requirement already ruled out 54 of the IQ-saturated ontologies, and 44 of the CQ-saturated ontologies, as they did not have any complex entailments of the required form. For Scenario S2, we randomly selected some concept $C \in \mathsf{Sub}(\mathcal{T})$ which had at least one instance (surprisingly, although C was not required to be complex, this ruled out 12 ontologies, including 4 of the CQ-saturated ones), together with a random selection of 10% of the individuals in $\mathcal{A}$, and built the repair request consisting of all assertions $C(a)$ where a ranges over the selected individuals. For both scenarios, we selected a random seed function for the obtained repair request.

For each ontology, scenario, and $\mathsf{QL} \in \{\mathsf{IQ}, \mathsf{CQ}\}$, we attempted to compute optimised QL-repairs for 50 different repair requests. We also tried to compute the set of objects that would be included in the canonical repairs, to get an idea of the impact of our optimisation. For each such repair computation, we used a timeout of 10 minutes. Since all repair requests used only concept descriptions that were already in the input ontology, the number of objects in the canonical repair was independent of the repair request. We thus performed the latter computation only once for each ontology. The success rates were as follows:

- The objects included in the canonical IQ- and CQ-repair could be computed within the timeout and without memory exceptions for respectively only 52.9 % and 62.1 % of the ontologies.
- For S1, we could compute the optimized IQ-repair in 99.9 %, and the optimised CQ-repair in 100.0 % of all attempts.
- For S2, 98.9 % of IQ-repairs and 99.9 % of CQ-repairs were successful.

This shows that the optimizations introduced in Section 5 have a very positive impact on the viability of our repair approach.

Fig. 3 gives more information on the number of objects and assertions in the computed repairs. On the left, we consider canonical and optimised IQ-repairs for scenario S2: specifically, we look at the difference in numbers of individuals occurring in the repair compared to the input ABox. In the middle and on the right, we visualise the difference between the number of assertions in the optimized IQ- and CQ-repairs, compared to the input ABoxes, for the scenarios S1 and S2, respectively. By construction, CQ-repairs cannot contain less assertions than the input ontologies. Sometimes the CQ-repairs were smaller than the corresponding IQ-repairs, which is due to the different saturation methods: variables introduced by the IQ-saturation could be connected to more individuals than for the CQ-saturation.

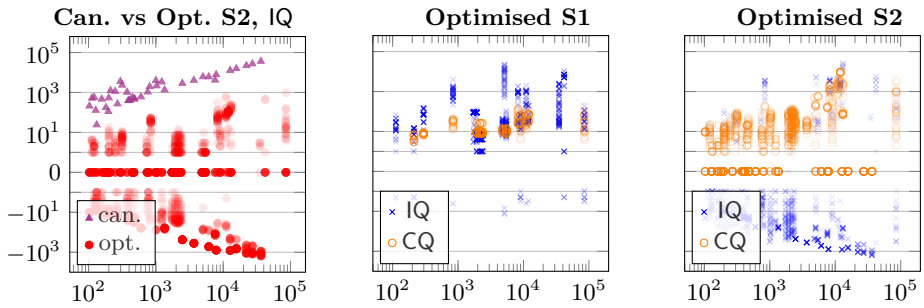

Fig. 3: Evaluation results. On the left, we show the difference of the number of object names in the canonical IQ-repairs (purple triangle) with the same difference, but restricted to objects occurring in assertions, for the optimised IQ-repairs (red circle) for S2. The other two graphs consider optimised IQ- and CQ-repairs for S1 and S2. In each graph, the x-axis shows the number of assertions in the input ontology, and the y-axis the observed difference.

7 Conclusion

This paper presents approaches for repairing DL-based ontologies, in the sense that they allow to get rid of unwanted consequences. In contrast to most of the other work on ontology repair, our goal is to compute *optimal* repairs, i.e., ones that lose the least amount of other consequences. As relevant consequences to be preserved, we consider both answers to conjunctive queries (CQ) and answers to $\mathcal{EL}$ instance queries (IQ). The presented results improve on our previous work in this direction in two respects. First, we allow for the presence of a TBox, which is assumed to be static (i.e., cannot be changed by the repair), whereas before we assumed that the TBox is empty. Second, we develop a more efficient construction of optimal repairs, which is exponential only in the worst case. Our experimental results show that this optimization makes our repair approach viable also for fairly large ontologies, at least for the IQ case.

One question for future research is how to lift the restriction to cycle-restricted TBoxes in the CQ case. Since optimal repairs need not longer exist then, one can ask whether the existence question is decidable, and how to compute optimal repairs if they exist. We have already noticed in our first attempts to tackle this problem that optimal repairs may then become larger than single-exponential.

In this and in our previous work, we have assumed that unwanted consequences are specified as $\mathcal{EL}$ instance relationships. Another interesting open question is whether our results can be generalized to a setting where unwanted consequences are specified as answers to conjunctive queries, as e.g. in [14].[4]

[4] Note that no TBox is considered in [14], and the notion of optimality used there is different from ours (see the introduction of [7] for a discussion of the differences).

References

1. Baader, F., Borgwardt, S., Morawska, B.: Extending unification in $\mathcal{EL}$ towards general TBoxes. In: Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012). pp. 568–572. AAAI Press/The MIT Press (2012)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005. pp. 364–369. Professional Book Center (2005)
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
4. Baader, F., Koopmann, P., Kriegel, F., Nuradiansyah, A.: Computing optimal repairs of quantified ABoxes w.r.t. static $\mathcal{EL}$ TBoxes (extended version). LTCS-Report 21-01, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany (2021), `https://lat.inf.tu-dresden.de/research/reports/2021/BaKoKrNu-LTCS-21-01.pdf`
5. Baader, F., Kriegel, F., Nuradiansyah, A.: Privacy-preserving ontology publishing for $\mathcal{EL}$ instance stores. In: Calimeri, F., Leone, N., Manna, M. (eds.) Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11468, pp. 323–338. Springer (2019)
6. Baader, F., Kriegel, F., Nuradiansyah, A., Peñaloza, R.: Making repairs in description logics more gentle. In: Thielscher, M., Toni, F., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018. pp. 319–328. AAAI Press (2018)
7. Baader, F., Kriegel, F., Nuradiansyah, A., Peñaloza, R.: Computing compliant anonymisations of quantified aboxes w.r.t. $\mathcal{EL}$ policies. In: Pan, J.Z., Tamma, V.A.M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12506, pp. 3–20. Springer (2020)
8. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In: Proceedings of the International Conference on Representing and Sharing Knowledge Using SNOMED (KR-MED'08). Phoenix, Arizona (2008)
9. Boyle, T.C.: Talk to Me. Bloomsbury Publishing (2021), To appear.
10. Calì, A., Lembo, D., Rosati, R.: On the decidability and complexity of query answering over inconsistent and incomplete databases. In: Neven, F., Beeri, C., Milo, T. (eds.) Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA. pp. 260–271. ACM (2003)
11. Carral, D., Dragoste, I., González, L., Jacobs, C.J.H., Krötzsch, M., Urbani, J.: Vlog: A rule engine for knowledge graphs. In: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I.F., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (eds.) The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference. Lecture Notes in Computer Science, vol. 11779, pp. 19–35. Springer (2019)
12. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: Hermit: An OWL 2 reasoner. J. Autom. Reason. **53**(3), 245–269 (2014)

13. Grau, B.C., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., Wang, Z.: Acyclicity notions for existential rules and their application to query answering in ontologies. J. Artif. Intell. Res. **47**, 741–808 (2013)
14. Grau, B.C., Kostylev, E.V.: Logical foundations of linked data anonymisation. J. Artif. Intell. Res. **64**, 253–314 (2019)
15. Haarslev, V., Hidde, K., Möller, R., Wessel, M.: The RacerPro knowledge representation and reasoning system. Semantic Web **3**(3), 267–277 (2012)
16. Henzinger, M.R., Henzinger, T.A., Kopke, P.W.: Computing simulations on finite and infinite graphs. In: 36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995. pp. 453–462. IEEE Computer Society (1995)
17. Hoehndorf, R., Schofield, P.N., Gkoutos, G.V.: The role of ontologies in biological and biomedical research: A functional perspective. Brief. Bioinform. **16**(6), 1069–1080 (2015)
18. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T.W., Thirunarayan, K. (eds.) The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5318, pp. 323–338. Springer (2008)
19. Horrocks, I., Li, L., Turi, D., Bechhofer, S.: The instance store: DL reasoning with large numbers of individuals. In: Haarslev, V., Möller, R. (eds.) Proceedings of the 2004 International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8, 2004. CEUR Workshop Proceedings, vol. 104. CEUR-WS.org (2004)
20. Johnson, D.S., Klug, A.C.: Testing containment of conjunctive queries under functional and inclusion dependencies. In: Ullman, J.D., Aho, A.V. (eds.) Proceedings of the ACM Symposium on Principles of Database Systems, March 29-31, 1982, Los Angeles, California, USA. pp. 164–169. ACM (1982)
21. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Proc. of ISWC'07. Lecture Notes in Computer Science, vol. 4825, pp. 267–280. Springer-Verlag (2007)
22. Kazakov, Y., Krötzsch, M., Simancik, F.: The incredible ELK - from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. Journal of Automed Reasoning **53**(1), 1–61 (2014)
23. Küsters, R.: Non-standard Inferences in Description Logics, Lecture Notes in Artificial Intelligence, vol. 2100. Springer-Verlag (2001)
24. Lam, J.S.C., Sleeman, D.H., Pan, J.Z., Vasconcelos, W.W.: A fine-grained approach to resolving unsatisfiable ontologies. J. Data Semant. **10**, 62–95 (2008)
25. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. J. Symb. Comput. **45**(2), 194–228 (2010)
26. Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing implications of data dependencies. ACM Trans. Database Syst. **4**(4), 455–469 (1979)
27. Meyer, T., Lee, K., Booth, R., Pan, J.Z.: Finding maximally satisfiable terminologies for the description logic $\mathcal{ALC}$. In: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006). AAAI Press/The MIT Press (2006)
28. Parsia, B., Matentzoglu, N., Gonçalves, R.S., Glimm, B., Steigmiller, A.: The OWL Reasoner Evaluation (ORE) 2015 competition report. Journal of Automed Reasoning **59**(4), 455–482 (2017)

29. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging OWL ontologies. In: Ellis, A., Hagino, T. (eds.) Proc. of the 14th International Conference on World Wide Web (WWW'05). pp. 633–640. ACM (2005)
30. Rosati, R.: On conjunctive query answering in $\mathcal{EL}$. In: Calvanese, D., Franconi, E., Haarslev, V., Lembo, D., Motik, B., Turhan, A., Tessaris, S. (eds.) Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007. CEUR Workshop Proceedings, vol. 250. CEUR-WS.org (2007)
31. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003). pp. 355–362. Morgan Kaufmann, Los Altos, Acapulco, Mexico (2003)
32. Schlobach, S., Huang, Z., Cornet, R., Harmelen, F.: Debugging incoherent terminologies. J. Automated Reasoning **39**(3), 317–349 (2007)
33. Steigmiller, A., Liebig, T., Glimm, B.: Konclude: System description. J. Web Semant. **27-28**, 78–85 (2014)
34. Troquard, N., Confalonieri, R., Galliani, P., Peñaloza, R., Porello, D., Kutz, O.: Repairing ontologies via axiom weakening. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 1981–1988. AAAI Press (2018)