Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes (Extended Version)

Franz Baader, Patrick Koopmann,
Francesco Kriegel, Adrian Nuradiansyah

LTCS-Report 21-01

Updated on May 5, 2021. This is an extended version of an article accepted at CADE 2021.

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes (Extended Version)

Franz Baader, Patrick Koopmann,
Francesco Kriegel, Adrian Nuradiansyah

May 5, 2021

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 [18]. 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 [23, 12, 34, 16], 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 it reveals private information that is supposed to be hidden [14, 4]. 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 [32, 30, 28, 33, 22, 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 [19, 25, 35, 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 [4] the ABox was additionally restricted to be a so-called instance store [20], 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 $Rich(x)$ and $parent(BEN, x)$ 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 $Rich(ANNE)$ 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 $Rich(JERRY)$ does not belong to the original ABox, but the TBox contains the axiom $Famous \sqsubseteq Rich$. Then the ABox on its own does not have the unwanted consequence $\exists parent.(Rich \sqcap Famous)(BEN)$, but together with the TBox it does. Saturation adds the assertion $Rich(JERRY)$ 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 $Rich(JERRY)$ is replaced, but $Famous(JERRY)$ 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 [4, 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.

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 $\Sigma_\mathsf{O}(\exists X.\mathcal{A}) \coloneqq \Sigma_\mathsf{I}(\exists X.\mathcal{A}) \cup X$. 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:

- $(\exists r.C)^\mathcal{I} \coloneqq \{\, \delta \mid \text{there exists some } \gamma \text{ such that } (\delta, \gamma) \in r^\mathcal{I} \text{ and } \gamma \in C^\mathcal{I} \,\}$,
- $(\bigsqcap \mathcal{C})^\mathcal{I} \coloneqq \bigcap \{\, C^\mathcal{I} \mid C \in \mathcal{C} \,\}$ 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. Given an $\mathcal{EL}$ concept description $C = \bigsqcap \mathcal{C}$, we denote its set of atoms $\mathcal{C}$ as $\mathsf{Conj}(C)$. 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].

The following two lemmas characterize subsumption and the instance problem, which both follow from the homomorphism characterization of entailment [7].

Lemma I. *Let C and D be two $\mathcal{EL}$ concept descriptions. C is subsumed by D w.r.t. $\emptyset$ if and only if $A \in \mathsf{Conj}(D)$ implies $A \in \mathsf{Conj}(C)$ for each concept name A and, for each existential restriction $\exists\, r.\, F \in \mathsf{Conj}(D)$, there is an existential restriction $\exists\, r.\, E \in \mathsf{Conj}(C)$ such that E is subsumed by F w.r.t. $\emptyset$.*

Lemma II. *Let $\mathcal{A}$ be the matrix of a qABox (seen as a usual ABox) and let $C(a)$ be an $\mathcal{EL}$ concept assertion. $\mathcal{A} \models C(a)$ if and only if $A(a) \in \mathcal{A}$ for each concept name $A \in \mathsf{Conj}(C)$ and, for each existential restriction $\exists\, r.\, D \in \mathsf{Conj}(C)$, there is some role assertion $r(a, b) \in \mathcal{A}$ such that $\mathcal{A} \models D(b)$.*

We also use the reduced form C^r of $\mathcal{EL}$ concept descriptions C [24], which in our setting is defined inductively as follows:

- For atoms, we set $A^r := A$ for $A \in \Sigma_{\mathsf{C}}$ and $(\exists\, r.\, C)^r := \exists\, r.\, C^r$.
- To obtain the reduced form of $\bigsqcap \mathcal{C}$, we first reduce the elements of $\mathcal{C}$, i.e., construct the set $\mathcal{C}^r := \{\, C^r \mid C \in \mathcal{C}\,\}$. Then we build $\mathsf{Min}(\mathcal{C}^r)$ by removing all elements D that are not subsumption minimal, i.e., for which there is an E in the set such that $E \sqsubset^{\emptyset} D$. We then set $(\bigsqcap \mathcal{C})^r := \bigsqcap \mathsf{Min}(\mathcal{C}^r)$.

Adapting the results in [24], one can show that $C \equiv^{\emptyset} C^r$ and that $C \equiv^{\emptyset} D$ implies $C^r = D^r$. In particular, this implies that, on reduced $\mathcal{EL}$ concept descriptions, subsumption is a partial order and not just a pre-order.

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 [26]. 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 $\varphi(x_1,\ldots,x_k) \in \mathsf{QL}$ and each tuple of individuals $(a_1,\ldots,a_k)$ we have that $\mathcal{T} \wedge \exists Y.\mathcal{B} \models \varphi(a_1,\ldots,a_k)$ implies $\mathcal{T} \wedge \exists X.\mathcal{A} \models \varphi(a_1,\ldots,a_k)$, where we view the TBox and the ABox as first-order formulae and $\models$ is classical first-order entailment (see [26] 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 $h(a) = a$ 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. $\mathcal{T}$.

$\sqcap$-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.

Figure 1: The CQ-saturation rules.

Basically, this saturation performs what is called *the chase* in the database community [27, 21, 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 $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$.*

Proof. By examining the three rules, we see that applying them attaches a tree to each object name occurring in the initial qABox. Each such tree is polynomially branching, since each freshly introduced successor of an object name t must be created by an application of the $\exists$-rule to an assertion $\exists r.C(t)$ for some existential restriction $\exists r.C$ occurring in the TBox. As the TBox is cycle-restricted, each such tree has polynomial depth for the following reason. If a path in such a tree contained two object names created for the same existential restriction $\exists r.C$, then the TBox would entail the concept inclusion $C \sqsubseteq \exists r_1. \cdots \exists r_k.C$ where $r_1, \ldots, r_k$ is the non-empty sequence of role names connecting the two object names—a contradiction. Finally, we conclude that each such tree has an exponential size, i.e., rule application must terminate after exponentially many steps.

Given the close connection between conjunctive queries and qABoxes as well as between the chase and the saturation, the equivalence of the first two statements easily follows from a well-known result on conjunctive query answering, namely Lemma 3.6 in [10]. The equivalence of the latter two statements is an immediate consequence of Proposition 2 in [7]. □

Below we provide an example 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 $\mathcal{EL}$ [31].

Example III. For each number $n \in \mathbb{N}$, consider the following TBox $\mathcal{T}_n$ that is defined over the signature consisting of the concept names $A_0, \ldots, A_n$ and of the role names r and s.

$$\mathcal{T}_n \coloneqq \{\, A_i \sqsubseteq \exists r.A_{i+1} \sqcap \exists s.A_{i+1} \mid 0 \leq i < n \,\}$$

The TBox obviously is cycle-restricted, and its size is polynomial in n.

Further consider the quantified ABox $\exists X.\mathcal{A} \coloneqq \exists \emptyset.\{A_0(a)\}$. A graphical representation of its CQ-saturation w.r.t. $\mathcal{T}_n$ is shown in Figure I. Essentially, it is a finite, binary tree with depth n, i.e., it has a size that is exponential in n.

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 [17], we conclude that IQ-entailment between qABoxes can be decided in polynomial time for the case of an empty TBox.

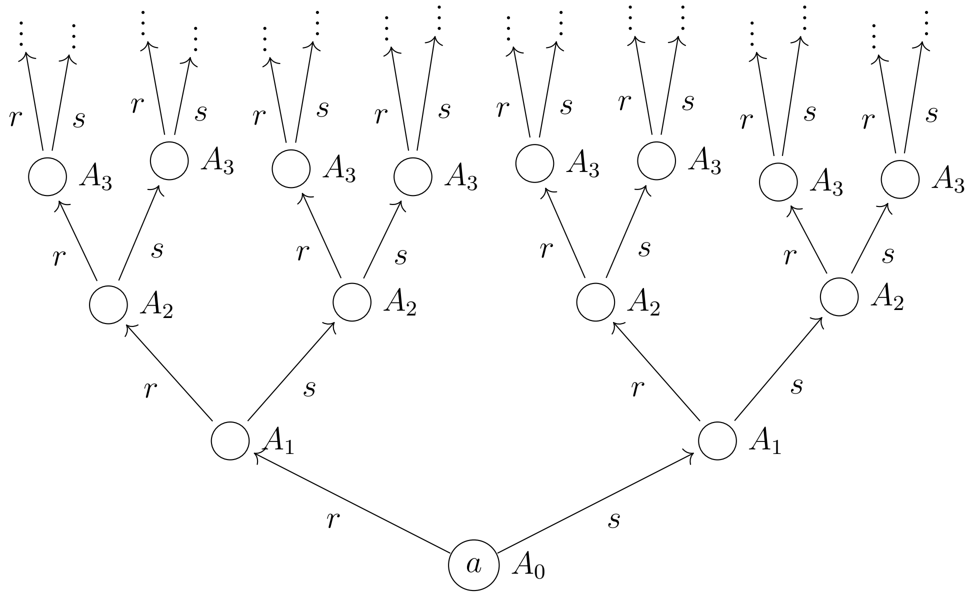


Figure I: The CQ-saturation of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}_n$

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 [26, 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 $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$.*

Before we prove the above theorem, we introduce some additional notions and prove some auxiliary results. Firstly, we define the interpretation $\mathcal{I}_{\exists X.\mathcal{A}}$ induced by a qABox $\exists X.\mathcal{A}$ as follows:

$$\Delta^{\mathcal{I}_{\exists X.\mathcal{A}}} \coloneqq \Sigma_{\mathsf{O}}(\exists X.\mathcal{A})$$

$\sqcap$-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.

Figure 2: The IQ-saturation rules.

$$\cdot^{\mathcal{I}_{\exists X.\mathcal{A}}} : \begin{cases} u \mapsto u & \text{for each object name } u \in \Sigma_{\mathsf{O}} \\ A \mapsto \{\, t \mid A(t) \in \mathcal{A} \,\} & \text{for each concept name } A \in \Sigma_{\mathsf{C}} \\ r \mapsto \{\, (t, u) \mid r(t, u) \in \mathcal{A} \,\} & \text{for each role name } r \in \Sigma_{\mathsf{R}} \end{cases}$$

It is easy to verify that $\mathcal{I}_{\exists X.\mathcal{A}}$ is a model of $\exists\, X.\mathcal{A}$. Furthermore, it is a finger exercise to show by induction on C that $\mathcal{A} \models C(t)$ if and only if $t \in C^{\mathcal{I}_{\exists X.\mathcal{A}}}$ for each $\mathcal{EL}$ concept description C and for each object name $t \in \Sigma_{\mathsf{O}}(\exists\, X.\mathcal{A})$.

Secondly, a *simulation* from an interpretation $\mathcal{I}$ to an interpretation $\mathcal{J}$ is a relation $\mathfrak{S} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ that satisfies the following conditions:

1. if $(\delta, \delta') \in \mathfrak{S}$ and $\delta \in A^{\mathcal{I}}$, then $\delta' \in A^{\mathcal{J}}$,
2. if $(\delta, \delta') \in \mathfrak{S}$ and $(\delta, \gamma) \in r^{\mathcal{I}}$, then there is some γ' such that $(\gamma, \gamma') \in \mathfrak{S}$ and $(\delta', \gamma') \in r^{\mathcal{J}}$.

It is straight-forward to prove by induction on C that $\delta \in C^{\mathcal{I}}$ implies $\delta' \in C^{\mathcal{J}}$ for each $\mathcal{EL}$ concept description C and for each simulation from $\mathcal{I}$ to $\mathcal{J}$ that contains (δ, δ'), see e.g. [26].

Proposition IV. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and consider a qABox $\exists\, X.\mathcal{A}$. For each $\mathcal{EL}$ concept assertion $C(a)$, it holds true that $\exists\, X.\mathcal{A} \models^{\mathcal{T}} C(a)$ if and only if $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\, X.\mathcal{A}) \models^{\emptyset} C(a)$.*

Proof. It is easy to see that the interpretation $\mathcal{I}_{\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})}$ is a model of both the qABox $\exists\, X.\mathcal{A}$ and the TBox $\mathcal{T}$. Thus, if $\exists\, X.\mathcal{A} \models^{\mathcal{T}} C(a)$, then $\mathcal{I}_{\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})} \models C(a)$ and so $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\, X.\mathcal{A}) \models^{\emptyset} C(a)$.

Now consider an interpretation $\mathcal{I}$ that is a model both of the qABox $\exists\, X.\mathcal{A}$ and of the TBox $\mathcal{T}$, and let

$$\exists\, X_0.\mathcal{A}_0 \to \exists\, X_1.\mathcal{A}_1 \to \cdots \to \exists\, X_n.\mathcal{A}_n$$

be the exhaustive sequence of rule applications starting with $\exists\, X_0.\mathcal{A}_0 \coloneqq \exists\, X.\mathcal{A}$ and ending with $\exists\, X_n.\mathcal{A}_n \coloneqq \mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\, X.\mathcal{A})$, i.e., no further rule application to

$\exists X_n.\mathcal{A}_n$ is possible. We show by induction on i that there is always a simulation $\mathfrak{S}_i$ from $\mathcal{I}_{\exists X_i.\mathcal{A}_i}$ to $\mathcal{I}$ that contains $(a, a^{\mathcal{I}})$ for each individual name a occurring in $\exists X.\mathcal{A}$. It then follows that there is a simulation from $\mathcal{I}_{\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})}$ to $\mathcal{I}$ containing $(a, a^{\mathcal{I}})$ for each individual name a. Thus, if $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}) \models C(a)$, then $\mathcal{I} \models C(a)$ for each interpretation $\mathcal{I}$ that is a model of $\exists X.\mathcal{A}$ and $\mathcal{T}$, i.e., $\exists X.\mathcal{A} \models^{\mathcal{T}} C(a)$.

For the induction base where $i = 0$, observe that $\mathcal{I}$ is a model of $\exists X.\mathcal{A} = \exists X_0.\mathcal{A}_0$. It follows that there exists a model $\mathcal{J}$ of the matrix $\mathcal{A}_0$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and where the interpretation functions $\cdot^{\mathcal{I}}$ and $\cdot^{\mathcal{J}}$ coincide on $\Sigma \setminus X_0$. It is easy to check that the following relation $\mathfrak{S}_0$ is a simulation from $\mathcal{I}_{\exists X_0.\mathcal{A}_0}$ to $\mathcal{I}$.

$$\mathfrak{S}_0 \coloneqq \{\, (t, t^{\mathcal{J}}) \mid t \in \Sigma_{\mathsf{O}}(\exists X.\mathcal{A}) \,\}$$

Furthermore, $\mathfrak{S}_0$ contains the pair $(a, a^{\mathcal{I}})$ for each individual name a occurring in $\exists X.\mathcal{A}$, since $a^{\mathcal{I}} = a^{\mathcal{J}}$ holds true for each individual name a.

For the induction step, we first define an invariant that we prove to be true for each ABox $\exists X_i.\mathcal{A}_i$, namely:

Invariant. If $C(t) \in \mathcal{A}_i$ and $(t, \delta) \in \mathfrak{S}_i$, then $\delta \in C^{\mathcal{I}}$.

Of course, the invariant is already satisfied for the cases where C is a concept name, cf. the definition of a simulation. Since the first ABox $\exists X_0.\mathcal{A}_0$ does not contain concept assertions with complex concept descriptions, the invariant is thus satisfied for $i = 0$.

Now assume that the invariant is true for i and that $\mathfrak{S}_i$ is a simulation from $\mathcal{I}_{\exists X_i.\mathcal{A}_i}$ to $\mathcal{I}$. We make a case distinction on the rule that is applied to $\exists X_i.\mathcal{A}_i$ and yields $\exists X_{i+1}.\mathcal{A}_{i+1}$.

$\sqcap$-rule. Assume that the $\sqcap$-rule is applied for the concept assertion $(C_1 \sqcap \cdots \sqcap C_n)(t)$ in $\mathcal{A}_i$, which removes that assertion and adds the assertions $C_1(t), \ldots, C_n(t)$. Since the invariant is true for i, we conclude that, for each $(t, \delta) \in \mathfrak{S}_i$, we have $\delta \in (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}}$, i.e., $\delta \in C_1^{\mathcal{I}}$, …, and $\delta \in C_n^{\mathcal{I}}$. Thus, we can simply define $\mathfrak{S}_{i+1} \coloneqq \mathfrak{S}_i$ to get a simulation from $\mathcal{I}_{\exists X_{i+1}.\mathcal{A}_{i+1}}$ to $\mathcal{I}$ and the invariant is satisfied for $i + 1$.

$\exists$-rule. Let the $\exists$-rule be applied for the concept assertion $\exists r.C(t)$ in $\mathcal{A}_i$, which removes that assertion and adds the assertions $r(t, x_C)$ and $C(x_C)$. For each $(t, \delta) \in \mathfrak{S}_i$, it holds true that $\delta \in (\exists r.C)^{\mathcal{I}}$, i.e., there is some γ_δ such that $(\delta, \gamma_\delta) \in r^{\mathcal{I}}$ and $\gamma_\delta \in C^{\mathcal{I}}$. Now define the relation

$$\mathfrak{S}_{i+1} \coloneqq \mathfrak{S}_i \cup \{\, (x_C, \gamma_\delta) \mid (t, \delta) \in \mathfrak{S}_i \,\}.$$

By construction, it is a simulation from $\mathcal{I}_{\exists X_{i+1}.\mathcal{A}_{i+1}}$ to $\mathcal{I}$ and the invariant is satisfied for $i + 1$.

$\sqsubseteq$-rule. Finally, assume that the $\sqsubseteq$-rule is applied for the object name t and the concept inclusion $C \sqsubseteq D$ in $\mathcal{T}$ where $\mathcal{A}_i \models C(t)$ and $\mathcal{A}_i \not\models D(t)$, which adds the concept assertion $D(t)$ to the matrix. As $\mathfrak{S}_i$ is a simulation from $\mathcal{I}_{\exists X_i.\mathcal{A}_i}$ to $\mathcal{I}$ and $\mathcal{A}_i \models C(t)$ implies that $\mathcal{I}_{\exists X_i.\mathcal{A}_i} \models C(t)$, we know that $\delta \in C^{\mathcal{I}}$ for each $(t, \delta) \in \mathfrak{S}_i$. Since $\mathcal{I}$ is a model of $\mathcal{T}$, we infer that $\delta \in D^{\mathcal{I}}$. Thus, we define $\mathfrak{S}_{i+1} \coloneqq \mathfrak{S}_i$ to obtain a simulation from $\mathcal{I}_{\exists X_{i+1}.\mathcal{A}_{i+1}}$ to $\mathcal{I}$ and the invariant is also true for $i + 1$. □

Lemma V. *Consider an $\mathcal{EL}$ TBox $\mathcal{T}$, a qABox $\exists X.\mathcal{A}$, and an $\mathcal{EL}$ concept assertion $C(a)$. It holds true that $\exists X.\mathcal{A} \models^{\mathcal{T}} C(a)$ if and only if there exists some $\mathcal{EL}$ concept description D such that $\exists X.\mathcal{A} \models^{\emptyset} D(a)$ and $D \sqsubseteq^{\mathcal{T}} C$.*

Proof Sketch. The statement was already proved as Lemma 22 in [26], but with classical ABoxes instead of qABoxes. That proof can be easily adapted. Alternatively, the statement can be proved as follows.

The *if* direction is straightforward. We can prove the *only-if* direction by utilizing *compactness of first-order logic* [15] and tree unravelings. Firstly, $\exists X.\mathcal{A}$ entails the concept assertion $C(a)$ w.r.t. $\mathcal{T}$ if and only if the pair of the tree unraveling of the ABox $\exists X.\mathcal{A}$ at a and of the TBox $\mathcal{T}$ entails $C(a)$. Secondly, this unraveling of the ABox, the TBox, and the negation of the concept assertion are translated into an infinite set of first-order sentences (using the standard translation from $\mathcal{EL}$ into first-order logic), which is unsatisfiable. Compactness now essentially yields a finite subtree of the unraveling of the ABox that together with the TBox and the negation of $C(a)$ is still unsatisfiable. Translating this back to $\mathcal{EL}$ shows the claim. □

We are now ready to prove Theorem 3.

Proof of Theorem 3. We know that, in the rule-based construction of the IQ-saturation, each new variable x_C corresponds to some atom $\exists r.C$ occurring in the TBox $\mathcal{T}$. Thus, there are always polynomially many object names in the qABox. The $\sqsubseteq$-rule can be applied at most once to some object name and some concept inclusion, i.e., there are at most polynomially many applications of $\sqsubseteq$-rule. Since each complex concept in a concept assertion is a subconcept occurring in the TBox, and the TBox contains only polynomially many such subconcepts, also the $\sqcap$-rule and the $\exists$-rule are applicable at most polynomially many times. With the results in [2] it follows that one application of the $\sqsubseteq$-rule needs polynomial time only, and it is further obvious that both the $\sqcap$-rule and the $\exists$-rule are applicable in polynomial time.

We continue with proving the equivalence of the first two statements of the theorem.

- For the only-if direction, let $\exists X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{IQ}} \exists Y.\mathcal{B}$ and consider a concept assertion $C(a)$ such that $\exists Y.\mathcal{B} \models^{\emptyset} C(a)$. It follows that $\exists Y.\mathcal{B} \models^{\mathcal{T}} C(a)$, and so the assumption implies that $\exists X.\mathcal{A} \models^{\mathcal{T}} C(a)$. An application of Proposition IV yields $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}) \models^{\emptyset} C(a)$ and we are done.

- Regarding the if direction, assume that $\mathsf{sat}_{\mathsf{IQ}}^{\mathcal{T}}(\exists X.\mathcal{A}) \models_{\mathsf{IQ}}^{\emptyset} \exists Y.\mathcal{B}$ and let $C(a)$ be a concept assertion such that $\exists Y.\mathcal{B} \models^{\mathcal{T}} C(a)$. By means of Lemma V it follows that there is some concept description D such that $\exists Y.\mathcal{B} \models^{\emptyset} D(a)$ and $D \sqsubseteq^{\mathcal{T}} C$. Now the assumption yields that $\mathsf{sat}_{\mathsf{IQ}}^{\mathcal{T}}(\exists X.\mathcal{A}) \models^{\emptyset} D(a)$, and thus we infer with Proposition IV that $\exists X.\mathcal{A} \models^{\mathcal{T}} D(a)$. Since $D \sqsubseteq^{\mathcal{T}} C$, we conclude that $\exists X.\mathcal{A} \models^{\mathcal{T}} C(a)$.

The last two statements of the theorem are equivalent according to Proposition 23 in [7]. □

Since $\mathsf{sat}_{\mathsf{IQ}}^{\mathcal{T}}(\exists X.\mathcal{A})$ can be computed in polynomial time and the existence of a simulation can be decided in polynomial time, this shows that the entailment relation $\models_{\mathsf{IQ}}^{\mathcal{T}}$ 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 $\exists X.\mathcal{A} \models_{\mathsf{QL}}^{\mathcal{T}} \exists Y.\mathcal{B}$ 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 $\exists Z.\mathcal{C} \models_{\mathsf{QL}}^{\mathcal{T}} \exists Y.\mathcal{B}$ and $\exists Z.\mathcal{C} \not\equiv_{\mathsf{QL}}^{\mathcal{T}} \exists Y.\mathcal{B}$.

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.

Recall that, in our setting where the TBox is fixed, classical entailment $\models$ coincides with CQ-entailment $\models_{\mathsf{CQ}}$. Thus, we do not need to dinguish between both, but we will mostly use $\models_{\mathsf{CQ}}$ to emphasize that not the IQ-entailment $\models_{\mathsf{IQ}}$ is meant.

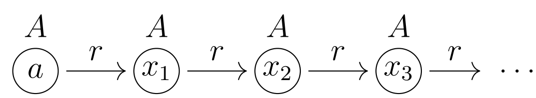
In addition, let $\mathcal{R}$ be a repair request and $\exists X.\mathcal{A}$ be the qABox to be QL-repaired for $\mathcal{R}$ w.r.t. $\mathcal{T}$. We assume that $\mathcal{R}$ does not contain an assertion of the form $C(a)$ such that $\top \sqsubseteq^{\mathcal{T}} C$ since the presence of such an 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.

Proposition VI. *Optimal repairs need not exist. More specifically, for the qABox* $\exists\,\emptyset.\,\{A(a)\}$*, the TBox* $\{A \sqsubseteq \exists r.A,\ \exists r.A \sqsubseteq A\}$*, and the repair request* $A(a)$ *that are all defined over the signature* $\Sigma \coloneqq \{a, A, r\}$*, there is no optimal* CQ*-repair, but there exists an optimal* IQ*-repair.*

Proof. Denote the above qABox by $\exists X.\mathcal{A}$ and the TBox by $\mathcal{T}$.

($\models_{\mathsf{CQ}}$) Let $\exists Y.\mathcal{B}$ be a CQ-repair of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$. In particular, $\exists X.\mathcal{A}$ CQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$ and so Theorem 2 yields that there exists a homomorphism from $\exists Y.\mathcal{B}$ to the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$. That saturation has the following graphical representation.

$\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$:



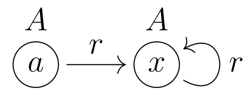
It follows that the set of all object names in $\exists Y.\mathcal{B}$ that are connected to a form a directed tree with root a in which all edges point away from a, and no object name in this tree is an instance of A.

Let $y \in Y$ be an object name that is reachable from a and has longest distance to a, choose some z not occurring in $\exists Y.\mathcal{B}$, and define the qABox $\exists\,(Y \cup \{z\}).\,(\mathcal{B} \cup \{r(y,z)\})$. Clearly, it is a repair that CQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$.

It remains to show that, in the converse direction, $\exists Y.\mathcal{B}$ does not CQ-entail $\exists\,(Y \cup \{z\}).\,(\mathcal{B} \cup \{r(y,z)\})$ w.r.t. $\mathcal{T}$, i.e., that there is no homomorphism from $\exists\,(Y \cup \{z\}).\,(\mathcal{B} \cup \{r(y,z)\})$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists Y.\mathcal{B})$. Since no object name in $\exists Y.\mathcal{B}$ reachable from a is an instance of A, the concept inclusions in $\mathcal{T}$ do not induce any new successors of nodes in the tree with root a when constructing the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists Y.\mathcal{B})$. Thus, there cannot exist a homomorphism of the tree rooted at a in $\exists\,(Y \cup \{z\}).\,(\mathcal{B} \cup \{r(y,z)\})$ to the tree rooted at a in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists Y.\mathcal{B})$, and so no homomorphism from $\exists\,(Y \cup \{z\}).\,(\mathcal{B} \cup \{r(y,z)\})$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists Y.\mathcal{B})$ can exist. We conclude that $\exists Y.\mathcal{B}$ is not optimal.

($\models_{\mathsf{IQ}}$) Now consider an IQ-repair $\exists Y.\mathcal{B}$ of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}$, i.e., there is a simulation from $\exists Y.\mathcal{B}$ to the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$, which equals $\exists\,\{x\}.\,\{A(a), r(a,x), A(x), r(x,x)\}$ and is depicted below.

$\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\, X.\, \mathcal{A})$:



We conclude that all object names in $\exists\, Y.\, \mathcal{B}$ connected to a form a tree with root a, in which the nodes can have loops (except a), and where all edges that are no loop point away from a. Furthermore, no object name in this tree can be an instance of A, since otherwise the concept inclusion $\exists\, r.\, A \sqsubseteq A$ in $\mathcal{T}$ would allow to infer that the root a is an instance of A, which would violate the assumption that $\exists\, Y.\, \mathcal{B}$ is a repair.

Consider the qABox $\exists\, \{x\}.\, \{r(a,x), r(x,x)\}$. Of course, it is a repair too. It further holds true that its IQ-saturation equals the qABox $\exists\, \{x\}.\, \{r(a,x), r(x,x)\}$. We conclude that there is a simulation from $\exists\, Y.\, \mathcal{B}$ to that saturation, namely which contains the pair (a,a) and the pair (y,x) for each object name y in the tree rooted at a. Note that we can ignore all object names not connected to a, since these are not connected to any individual name and so the definition of a simulation does not require that they occur in some pair of the simulation.

As a consequence we obtain that the qABox $\exists\, \{x\}.\, \{r(a,x), r(x,x)\}$ is, up to IQ-equivalence, the unique optimal IQ-repair of $\exists\, X.\, \mathcal{A}$. □

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.

Note that no repair type can contain $\top$, as $\top$ is no atom. The empty set is always a repair type. Obviously, not every subset of $\mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ is a repair type. We can, however, try to enlarge such a subset to a repair type by exhaustively applying the following non-deterministic rule. This will not always be possible, because such a subset of $\mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ could contain an atom that subsumes $\top$ w.r.t. $\mathcal{T}$ and so Condition 3 in Definition 5 cannot be fulfilled.

Consider a subset $\mathcal{K} \subseteq \mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ and some object name u occurring in the saturation $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$. Without loss of generality assume that $\mathcal{K}$ is a *repair pre-type* for u, i.e., the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $C(u)$ for each atom $C \in \mathcal{K}$ and $\mathcal{K}$ does not contain $\sqsubseteq^{\emptyset}$-comparable atoms—if this is not the case, then we can simply replace $\mathcal{K}$ with the set

$$\mathsf{Max}_{\sqsubseteq^{\emptyset}}(\{\, C \mid C \in \mathcal{K} \text{ and the matrix of } \mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}) \text{ entails } C(u) \,\}).$$

Premise-saturation rule. If $C \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$, the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $C(u)$, $D \in \mathcal{K}$, $C \sqsubseteq^{\mathcal{T}} D$, and $\mathcal{K}$ does not contain an atom subsuming C, then choose some atom E in $\mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ that subsumes C (i.e., where $C \sqsubseteq^{\emptyset} E$) and return $\mathsf{Max}_{\sqsubseteq^{\emptyset}}(\mathcal{K} \cup \{E\})$.

Note that the above rule is *non-deterministic*: given a repair pre-type $\mathcal{K}$, the rule constructs, for each atom E subsuming C, a successor pre-type $\mathsf{Max}_{\sqsubseteq^{\emptyset}}(\mathcal{K} \cup \{E\})$. Further note that the rule cannot return a pre-type if $\mathcal{K}$ contains an atom subsuming $\top$ w.r.t. $\mathcal{T}$, since there is no atom that subsumes $\top$ w.r.t. $\emptyset$.

If $\mathcal{L}$ is constructed from $\mathcal{K}$ by one application of the premise-saturation rule, then each atom in $\mathcal{K}$ is subsumed by some atom in $\mathcal{L}$, which we denote by $\mathcal{K} \leq \mathcal{L}$ and say that $\mathcal{K}$ is *covered by* $\mathcal{L}$. As a special case, the proof of the below proposition shows that, starting with some set of atoms, the premise-saturation rule produces all relevant repair types.

Proposition VII. *Let $\mathcal{S}$ be a subset of $\mathsf{Sub}(\mathcal{R}, \mathcal{T})$ and let u be an object name occurring in $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ such that the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $C(u)$ for* *each $C \in \mathcal{S}$. The set of all $\leq$-minimal repair types for u that cover $\mathcal{S}$ can be computed in exponential time.*

Proof. Since the set $\mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ has polynomial size, we can enumerate in exponential time all its subsets and filter out those that do not satisfy the three conditions in Definition 5 or which do not cover $\mathcal{S}$. Each check whether such a subset is to be filtered needs polynomial time. Afterwards, we need to compare the resulting repair types pairwise w.r.t. $\leq$ and keep those for which there is no smaller one w.r.t. $\leq$. Each such comparison can be done in polynomial time. Summing up, the set of all $\leq$-minimal repair types for u covering $\mathcal{S}$ can be computed in exponential time.

A more efficient way to compute all $\leq$-minimal repair types for u covering $\mathcal{S}$ is as follows. Initially, compute all sets of atoms that can be obtained from $\mathcal{S}$ by exhaustively applying the following non-deterministic rule, starting with $\mathcal{K} \coloneqq \mathcal{S}$.

Initialization rule. If $\mathcal{K}$ contains a concept description C that is no atom, then choose some atom $D \in \mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ that subsumes C w.r.t. $\emptyset$ and return $\mathsf{Max}_{\sqsubseteq^\emptyset}((\mathcal{K} \setminus \{C\}) \cup \{D\})$.

By construction, all resulting sets are repair pre-types for u that cover $\mathcal{S}$. Afterwards, exhaustively apply the premise-saturation rule to all these repair pre-types. That way, we produce a superset of all $\leq$-minimal repair types for u covering $\mathcal{S}$, which we will prove in the following.

First of all, if the premise-saturation rule is not applicable to some repair pre-type, then it must already be a repair type. Furthermore, we have already recognized that one application of the premise-saturation rule produces a repair pre-type that covers the given one. By induction, we conclude that, if $\mathcal{L}$ is constructed from some repair pre-type $\mathcal{K}$ for u by exhaustively applying the premise-saturation rule, then $\mathcal{L}$ is a repair type for u such that $\mathcal{K}$ is covered by $\mathcal{L}$, and consequently $\mathcal{L}$ covers $\mathcal{S}$ as well.

Vice versa, if $\mathcal{L}$ is a $\leq$-minimal repair type for u that covers $\mathcal{S}$, then $\mathcal{L}$ can be constructed from $\mathcal{S}$ by first exhaustively applying the initialization rule and then exhaustively applying the premise-saturation rule where we always choose an atom from $\mathcal{L}$. We will need to add all atoms from $\mathcal{L}$ since there is no repair type $\mathcal{L}'$ for u such that $\mathcal{S} \leq \mathcal{L}' < \mathcal{L}$.

We conclude that we can produce a superset of all $\leq$-minimal repair types that cover $\mathcal{S}$—in the end, we just need to filter out the non-minimal ones.

The computation tree of first applying the initialization rule and then applying the premise-saturation rule is polynomially branching and has polynomial depth, both since $\mathsf{Atoms}(\mathcal{R}, \mathcal{T})$ has polynomial size. It follows that the computation tree has exponential size, and thus contains exponentially many leafs (which are the repair types). Since one edge in this tree can be computed in polynomial time, the whole tree can be constructed in exponential time. Eventually, filtering out the non-optimal repair types can be done in exponential time, since each repair type has polynomial size and thus comparing two repair types w.r.t. $\leq$ (the covers relation) needs polynomial time. That is, computing all $\leq$-minimal repair types that cover $\mathcal{S}$ can be done in exponential time. $\square$

To illustrate that this exponential upper bound can indeed be reached, we provide the following example.

Example VIII. Consider the quantified ABox

$$\exists X_n.\mathcal{A}_n \coloneqq \exists \emptyset.\{\, A_i(a), P_i(a), Q_i(a) \mid i \in \{1,\ldots,n\}\,\}$$

and the TBox

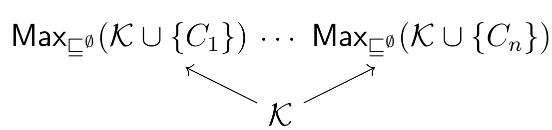
$$\mathcal{T}_n \coloneqq \{\, P_i \sqcap Q_i \sqsubseteq A_i \mid i \in \{1,\ldots,n\}\,\}.$$

The set $\mathcal{K}_n \coloneqq \{A_1,\ldots,A_n\}$ is a repair pre-type for a. Clearly, the sizes of $\exists X_n.\mathcal{A}_n$, of $\mathcal{T}_n$, and of $\mathcal{K}_n$ are all polynomial in n. Each $\leq$-minimal repair type covering $\mathcal{K}_n$ is of the form $\mathcal{K}_n \cup \{X_1,\ldots,X_n\}$ where $X_i \in \{P_i, Q_i\}$ for each index i, which can all be constructed by means of the premise-saturation rule, and their number is exponential in n.

As already noticed, the premise-saturation rule cannot be applied to some pre-type $\mathcal{K}$ if $\top$ is a subconcept occurring in $\mathcal{R}$ or in $\mathcal{T}$ and there is some atom $D \in \mathcal{K}$ such that $\top \sqsubseteq^{\mathcal{T}} D$. Then there simply does not exist an atom subsuming $\top$, which could be added to $\mathcal{K}$. Thus, there can be repair pre-types for u that are not covered by a repair type for u. The following lemma characterizes when a pre-type is covered by some repair type.

Lemma IX. *Assume that u is an object name occurring in $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$. If $\mathcal{K}$ is a repair pre-type for u and $\top \not\sqsubseteq^{\mathcal{T}} D$ for each $D \in \mathcal{K}$, then there is a repair type for u that covers $\mathcal{K}$.*

Proof. Assume that there is no repair type for u covering $\mathcal{K}$. It follows that, when exhaustively applying the above premise-saturation rule, we always reach a situation where $C = \top$, i.e., where no atom subsuming C exists. In particular, we will also always encounter a situation where $C = \top$ if, instead of choosing some atom in $\mathsf{Atoms}(\mathcal{R},\mathcal{T})$ that subsumes C and adding it to $\mathcal{K}$, we simply add some top-level conjunct of C to $\mathcal{K}$. The following picture shows the non-deterministic choice to be made in this variant of the premise-saturation rule.



where $C \in \mathsf{Sub}(\mathcal{R},\mathcal{T})$, the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ does not entail $C(u)$, $D \in \mathcal{K}$, $C \sqsubseteq^{\mathcal{T}} D$, $\mathcal{K}$ does not contain an atom subsuming C, and $\mathsf{Conj}(C) = \{C_1,\ldots,C_n\}$

Now consider the full tree of choices induced by exhaustive application of the premise-saturation rule starting with $\mathcal{K}$. By assumption, each leaf $\mathcal{L}$ in this tree contains some atom $D \in \mathcal{L}$ such that $\top \sqsubseteq^{\mathcal{T}} D$ (and $\top \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$). We will now show by induction on the tree that each node $\mathcal{M}$ contains some atom $D \in \mathcal{M}$ such that $\top \sqsubseteq^{\mathcal{T}} D$. Finally, this property will be satisfied also for the root node $\mathcal{K}$ and we are done.

We call a node $\mathcal{M}$ *unprocessed* if we have not yet shown that it contains an atom D where $\top \sqsubseteq^{\mathcal{T}} D$, and *processed* otherwise. Initially, all leafs are processed, which is the induction base. For the induction step, consider an unprocessed node $\mathcal{M}$ such that each successor is processed. By construction of the tree, the successors are of the form $\mathsf{Max}_{\sqsubseteq^\emptyset}(\mathcal{M} \cup \{C_i\})$ where $C \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$, $E \in \mathcal{M}$, $C \sqsubseteq^{\mathcal{T}} E$, and $\mathsf{Conj}(C) = \{C_1, \ldots, C_n\}$. Since each $\mathsf{Max}_{\sqsubseteq^\emptyset}(\mathcal{M} \cup \{C_i\})$ is processed, each contains some atom D_i where $\top \sqsubseteq^{\mathcal{T}} D_i$. If one of the D_i is already in $\mathcal{M}$, then $\mathcal{M}$ is processed. Otherwise, we can only have $C_i = D_i$ (because $D_i \in \mathsf{Max}_{\sqsubseteq^\emptyset}(\mathcal{M} \cup \{C_i\}) \subseteq \mathcal{M} \cup \{C_i\}$) and thus $\top \sqsubseteq^{\mathcal{T}} C_i$ for each index $i \in \{1, \ldots, n\}$—we conclude that $\top \sqsubseteq^{\mathcal{T}} C$ and thus $\top \sqsubseteq^{\mathcal{T}} E$, i.e., $\mathcal{M}$ is processed. □

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 $C \sqsubseteq^{\emptyset} D$.

Together with our initial assumption that the repair request $\mathcal{R}$ does not contain a concept assertion $C(a)$ such that $\top \sqsubseteq^{\mathcal{T}} C$, Lemma IX implies that a repair seed function exists.

Proposition X. *There is at least one repair seed function.*

Proof. Consider an individual name a. By assumption, $\top \not\sqsubseteq^{\mathcal{T}} C$ for each $C(a) \in \mathcal{R}$, i.e., there always exists some atom $D_C \in \mathsf{Conj}(C)$ such that $\top \not\sqsubseteq^{\mathcal{T}} D_C$. Now $\mathcal{K}_a := \mathsf{Max}_{\sqsubseteq^\emptyset}(\{\, D_C \mid C(a) \in \mathcal{R} \text{ and } \mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}) \models C(a) \,\})$ is a repair pre-type for a which does not contain an atom subsuming $\top$ w.r.t. $\mathcal{T}$ and which contains an atom subsuming C for each $C(a) \in \mathcal{R}$ where $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}) \models C(a)$. Lemma IX guarantees the existence of a repair type $\mathcal{L}_a$ for a that covers $\mathcal{K}_a$ and we can simply define $s(a) := \mathcal{L}_a$. The mapping s is a 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.

Before we can prove that canonical repairs are in fact repairs and further that each optimal repair is equivalent to a canonical one, we need the following three lemmas.

Lemma XI. *Let u be an object name occurring in the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$. If $C \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$ and $\mathcal{K}$ is a repair pre-type for u where $C \not\sqsubseteq^{\mathcal{T}} D$ for each $D \in \mathcal{K}$, then there is a repair type for u that covers $\mathcal{K}$ and that does not contain an atom subsuming C.*

Proof. Let $C \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$, let $\mathcal{K}$ be a repair pre-type for u, and assume that each repair type for u covering $\mathcal{K}$ contains some atom subsuming C. Consider the full tree of choices that is generated by exhaustively applying the variant of the premise-saturation rule, just like in the proof of Lemma IX. By assumption, each leaf in this tree contains some atom D such that $C \sqsubseteq^{\mathcal{T}} D$. We can show by induction on the tree that each node contains some atom D such that $C \sqsubseteq^{\mathcal{T}} D$. Finally, this property is satisfied also for the root node $\mathcal{K}$ and we are done. □

Lemma XII. *Assume that s is a repair seed function. For each subconcept $C \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$, it holds true that the matrix of $\mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s)$ entails $C(y_{u,\mathcal{K}})$ if and only if the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ entails $C(u)$ and $\mathcal{K}$ does not contain an atom that subsumes C.*

Proof. Denote by $\mathcal{B}$ the matrix of the canonical repair $\mathsf{rep}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}, s)$. We start with the only-if direction. Thus assume that $\mathcal{B}$ entails $C(y_{u,\mathcal{K}})$ for some $C \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$. Since the mapping $y_{u,\mathcal{K}} \mapsto u$ is a homomorphism from $\exists Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$, it follows that the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A})$ entails $C(u)$. We now show the following claim by induction on the role depth of D, from which it easily follows that $\mathcal{K}$ cannot contain an atom subsuming C.

Claim. If $D \in \mathcal{K}$, then $\mathcal{B} \not\models D(y_{u,\mathcal{K}})$.

The base case where D is a concept name is true by the very definition of $\mathcal{B}$. Now let D be an existential restriction $\exists r.E$, and consider a role assertion $r(y_{u,\mathcal{K}}, y_{v,\mathcal{L}})$

in $\mathcal{B}$. If the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ does not entail $E(v)$, then $\mathcal{B}$ does not entail $E(y_{v,\mathcal{L}})$. Otherwise, by definition of $\mathcal{B}$, the set $\mathcal{L}$ must contain an atom F such that $E \sqsubseteq^{\emptyset} F$. Specifically, the role depth of F is bounded by the role depth of E, and so we can apply the induction hypothesis to infer that $\mathcal{B} \not\models F(y_{v,\mathcal{L}})$. Due to $E \sqsubseteq^{\emptyset} F$, we conclude that $\mathcal{B} \not\models E(y_{v,\mathcal{L}})$. Using Lemma II, it follows that $\mathcal{B}$ does not entail $\exists r.E(y_{u,\mathcal{K}})$.

It remains to prove the if direction. We do this by induction on C. Let the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entail $C(u)$ and assume that $\mathcal{K}$ does not contain an atom subsuming C.

- The case where $C = \top$ is trivial.
- Assume that $C = A$ for a concept name A. Then $A \notin \mathcal{K}$ and so it follows from the very definition of $\mathcal{B}$ that the concept assertion $A(y_{u,\mathcal{K}})$ is contained in $\mathcal{B}$, i.e., $\mathcal{B} \models A(y_{u,\mathcal{K}})$.
- Let $C = C_1 \sqcap \cdots \sqcap C_n$ be a conjunction of atoms $C_1, \ldots, C_n$ where $n \geq 2$. Note that, by definition of $\mathsf{Sub}(\mathcal{R}, \mathcal{T})$, each conjunct C_i is an element of $\mathsf{Sub}(\mathcal{R}, \mathcal{T})$. The preconditions immediately imply that, for each index i, the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $C_i(u)$ and $\mathcal{K}$ does not contain an atom subsuming C_i. The induction hypothesis yields that $\mathcal{B} \models C_i(y_{u,\mathcal{K}})$ for each i, and thus it follows that $\mathcal{B} \models C(y_{u,\mathcal{K}})$.
- Consider the last case where $C = \exists r.D$ is an existential restriction. Of course, we have $D \in \mathsf{Sub}(\mathcal{R}, \mathcal{T})$. According to Lemma II, it follows from the preconditions that there exists some object name v such that the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ contains $r(u, v)$ and entails $D(v)$. Since $\exists r.D$ is not subsumed by an atom in $\mathcal{K}$, it follows that $\exists r.D$ is no element of $\mathcal{K}$. We further conclude that $D \not\sqsubseteq^{\mathcal{T}} E$ for each $\exists r.E \in \mathcal{K}$ (otherwise $C = \exists r.D$ would be in $\mathcal{K}$, a contradiction). Thus for each $\exists r.E \in \mathcal{K}$, there is some atom $F_E \in \mathsf{Conj}(E)$ such that $D \not\sqsubseteq^{\mathcal{T}} F_E$. According to Lemma XI there exists a repair type $\mathcal{L}$ for v that covers the repair pre-type $\mathsf{Max}_{\sqsubseteq^{\emptyset}}(\{\, F_E \mid \exists r.E \in \mathcal{K}$ and the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $E(v)\,\})$ and that does not contain an atom subsuming D. Applying the induction hypothesis then yields that $\mathcal{B} \models D(y_{v,\mathcal{L}})$. By the very construction of $\mathcal{L}$, it follows that the matrix $\mathcal{B}$ contains the role assertion $r(y_{u,\mathcal{K}}, y_{v,\mathcal{L}})$. Thus, we conclude that $\mathcal{B} \models C(y_{u,\mathcal{K}})$. □

Lemma XIII. *For each repair seed function s, the canonical repair induced by s equals its saturation, i.e.,* $\mathsf{rep}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}, s) = \mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\mathsf{rep}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}, s))$.

Proof. Let $\exists Y.\mathcal{B} \coloneqq \mathsf{rep}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$ be the canonical repair induced by s. Since, for both query languages IQ and CQ, the $\sqsubseteq$-rule employed for constructing the saturations is the same, the following argumentation applies to both choices.

We show that the $\sqsubseteq$-rule is not applicable to $\exists Y.\mathcal{B}$. It is trivial that none of the other two rules is applicable, since the matrix $\mathcal{B}$ can never contain a concept assertion involving a complex concept description.

Consider an object name $y_{u,\mathcal{K}} \in \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B})$ and a concept inclusion $C \sqsubseteq D \in \mathcal{T}$. We know that the $\sqsubseteq$-rule is not applicable to the QL-saturation $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$, which means that the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ either does not entail $C(u)$ or entails $D(u)$. We proceed with a case distinction.

- Assume that the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ does not entail $C(u)$. Lemma XII shows that the matrix $\mathcal{B}$ does not entail $C(y_{u,\mathcal{K}})$, and so we conclude that the $\sqsubseteq$-rule is not applicable to $y_{u,\mathcal{K}}$ for $C \sqsubseteq D$.
- Otherwise, the matrix of $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $D(u)$. We make a further case distinction.
  - Assume that $\mathcal{K}$ contains an atom subsuming D. Since $\mathcal{K}$ is a repair type and $C \sqsubseteq D \in \mathcal{T}$, we conclude that $\mathcal{K}$ must contain some atom subsuming C. Lemma XII yields that the matrix $\mathcal{B}$ does not entail $C(y_{u,\mathcal{K}})$, which implies that the $\sqsubseteq$-rule is not applicable to $y_{u,\mathcal{K}}$ for $C \sqsubseteq D$.
  - In the remaining case there is no atom in $\mathcal{K}$ that subsumes D. An application of Lemma XII shows that the matrix $\mathcal{B}$ entails $D(y_{u,\mathcal{K}})$, i.e., the $\sqsubseteq$-rule is not applicable to $y_{u,\mathcal{K}}$ for $C \sqsubseteq D$. □

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 $\mathsf{rep}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$ 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 $\mathsf{rep}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}, s) \models_{\mathsf{QL}}^{\mathcal{T}} \exists Y.\mathcal{B}$.*

Proof. We begin with proving the first statement, namely, that $\mathsf{rep}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$ is a QL-repair. We first consider the case where $\mathsf{QL} = \mathsf{IQ}$. Denote by $\exists Y.\mathcal{B}$ the canonical repair $\mathsf{rep}_{\mathsf{IQ}}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$. Let $C(a) \in \mathcal{R}$. Either the IQ-saturation $\mathsf{sat}_{\mathsf{IQ}}^{\mathcal{T}}(\exists X.\mathcal{A})$ does not entail $C(a)$, or the repair type $s(a)$ contains an atom that subsumes C. For both cases Lemma XII yields that $\mathcal{B} \not\models^{\emptyset} C(a)$. It follows that $\exists Y.\mathcal{B} \not\models^{\emptyset} C(a)$, and with Lemma XIII we conclude that $\mathsf{sat}_{\mathsf{QL}}^{\mathcal{T}}(\exists Y.\mathcal{B}) \not\models^{\emptyset} C(a)$. Proposition IV immediately yields that $\exists Y.\mathcal{B} \not\models^{\mathcal{T}} C(a)$.

It remains to show that $\exists X.\mathcal{A}$ IQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$. Theorem 3 shows that it suffices to find a simulation from $\exists Y.\mathcal{B}$ to $\mathsf{sat}_{\mathsf{IQ}}^{\mathcal{T}}(\exists X.\mathcal{A})$. It is easy to verify that $\{\, (y_{t,\mathcal{K}}, t) \mid y_{t,\mathcal{K}} \in \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) \,\}$ is such a simulation.

Now, we treat the case $\mathsf{QL} = \mathsf{CQ}$. Denote by $\exists Y.\mathcal{B}$ the canonical repair $\mathsf{rep}_{\mathsf{CQ}}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$. Let $C(a) \in \mathcal{R}$. Either the CQ-saturation $\mathsf{sat}_{\mathsf{CQ}}^{\mathcal{T}}(\exists X.\mathcal{A})$ does not entail $C(a)$, or the repair type $s(a)$ contains an atom that subsumes C. For both cases Lemma XII yields that $\mathcal{B} \not\models^{\emptyset} C(a)$. It follows that $\exists Y.\mathcal{B} \not\models^{\emptyset} C(a)$, and with Lemma XIII we conclude that $\mathsf{sat}_{\mathsf{CQ}}^{\mathcal{T}}(\exists Y.\mathcal{B}) \not\models^{\emptyset} C(a)$.

It is easy to see that there is a qABox which is equivalent to the concept assertion $C(a)$. For instance, such a qABox can be constructed by exhaustively applying the $\sqcap$-rule and the $\exists$-rule to $\exists\,\emptyset.\{C(a)\}$, or alternatively using the construction described in [5]. Note that the qABox translation contains only the individual name a and all other object names occurring in it are variables. In the following, we will identify $C(a)$ with its translation to an equivalent qABox.

Since classical entailment $\models^{\mathcal{T}}$ ($\models^{\emptyset}$) and CQ-entailment $\models^{\mathcal{T}}_{\mathsf{CQ}}$ ($\models^{\emptyset}_{\mathsf{CQ}}$) coincide, we infer by an application of Theorem 2 that $\exists\,Y.\mathcal{B} \not\models^{\mathcal{T}} C(a)$.

It remains to show that $\exists\,X.\mathcal{A}$ CQ-entails $\exists\,Y.\mathcal{B}$ w.r.t. $\mathcal{T}$. Theorem 2 shows that it suffices to find a homomorphism from $\exists\,Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists\,X.\mathcal{A})$. It is easy to verify that the mapping $\{\, y_{t,\mathcal{K}} \mapsto t \mid y_{t,\mathcal{K}} \in \Sigma_{\mathsf{O}}(\exists\,Y.\mathcal{B}) \,\}$ is such a homomorphism.

In the following, we will prove the second statement. We first deal with the case where $\mathsf{QL} = \mathsf{IQ}$. Consider an IQ-repair $\exists\,Y.\mathcal{B}$ of $\exists\,X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$, i.e., $\exists\,X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{IQ}} \exists\,Y.\mathcal{B}$ and $\exists\,Y.\mathcal{B} \not\models^{\mathcal{T}} C(a)$ for each $C(a) \in \mathcal{R}$. According to Theorem 3 there exists a simulation $\mathfrak{S}$ from $\exists\,Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A})$.

We define the mapping $f\colon \Sigma_{\mathsf{O}}(\exists\,Y.\mathcal{B}) \times \Sigma_{\mathsf{O}}(\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A})) \to \wp(\mathsf{Atoms}(\mathcal{R},\mathcal{T}))$ by

$$f(t,v) \coloneqq \mathsf{Max}_{\sqsubseteq^{\emptyset}}\left(\left\{\, D \;\middle|\; \begin{array}{l} D \in \mathsf{Atoms}(\mathcal{R},\mathcal{T}),\ \mathcal{B} \not\models^{\mathcal{T}} D(t), \\ \text{and the matrix of } \mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A}) \text{ entails } D(v) \end{array} \right\}\right)$$

for each object name $t \in \Sigma_{\mathsf{O}}(\exists\,Y.\mathcal{B})$ and each object name $v \in \Sigma_{\mathsf{O}}(\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A}))$. We show that each $f(t,v)$ is a repair type for v. Assume that $C \in \mathsf{Sub}(\mathcal{R},\mathcal{T})$, the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A})$ entails $C(v)$, $D \in f(t,v)$, and $C \sqsubseteq^{\mathcal{T}} D$. It follows that $\mathcal{B} \not\models^{\mathcal{T}} C(t)$, i.e., there is some atom $E \in \mathsf{Conj}(C) \subseteq \mathsf{Atoms}(\mathcal{R},\mathcal{T})$ such that $\mathcal{B} \not\models^{\mathcal{T}} E(t)$. Of course, the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A})$ entails $E(v)$ as well. Thus, $f(t,v)$ contains either E or some atom subsuming E.

The function s where $s(a) \coloneqq f(a,a)$ for each individual name $a \in \Sigma_{\mathsf{I}}(\exists\,Y.\mathcal{B})$ clearly is a repair seed function. We now show that the relation

$$\mathfrak{T} \coloneqq \{\, (t, y_{v,f(t,v)}) \mid (t,v) \in \mathfrak{S} \,\}$$

is a simulation from $\exists\,Y.\mathcal{B}$ to $\mathsf{rep}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A}, s)$.

1. Consider an individual name a. Since $(a,a) \in \mathfrak{S}$, we conclude that $(a, y_{a,f(a,a)}) = (a, y_{a,s(a)}) = (a,a)$ is in $\mathfrak{T}$.
2. Assume that $A(t) \in \mathcal{B}$ and consider $(t, y_{v,f(t,v)}) \in \mathfrak{T}$. It follows that $(t,v) \in \mathfrak{S}$, which implies that the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A})$ contains $A(v)$. Furthermore, $A(t) \in \mathcal{B}$ implies that $\mathcal{B} \models^{\mathcal{T}} A(t)$, and thus $A \notin f(t,v)$. We conclude that the matrix of $\mathsf{rep}^{\mathcal{T}}_{\mathsf{IQ}}(\exists\,X.\mathcal{A}, s)$ contains $A(y_{v,f(t,v)})$.
3. Consider a role assertion $r(t,u) \in \mathcal{B}$ and let the pair $(t, y_{v,f(t,v)})$ be in the simulation $\mathfrak{T}$. It follows that the simulation $\mathfrak{S}$ contains (t,v), and so

there is some w such that the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$ contains the role assertion $r(v,w)$ and the pair (u,w) is in $\mathfrak{S}$. Note that $(u, y_{w,f(u,w)}) \in \mathfrak{T}$. We show that the matrix of the canonical repair $\mathsf{rep}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}, s)$ contains $r(y_{v,f(t,v)}, y_{w,f(u,w)})$.

Assume that the existential restriction $\exists r.C$ is in $f(t,v)$ and the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$ entails $C(w)$. According to Definition 7 we need to find an atom in $f(u,w)$ that subsumes C. Since $\exists r.C \in f(t,v)$, it follows that $\mathcal{B} \not\models^{\mathcal{T}} \exists r.C(t)$. We conclude that, in particular, the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists Y.\mathcal{B})$ does not entail $\exists r.C(t)$ (by Proposition IV) but it contains $r(t,u)$ (by construction, cf. Page 9), which yields that $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists Y.\mathcal{B})$ cannot entail $C(u)$, i.e., $\mathcal{B} \not\models^{\mathcal{T}} C(u)$ (by Proposition IV).

It follows that there is an atom $D \in \mathsf{Conj}(C)$ such that $\mathcal{B} \not\models^{\mathcal{T}} D(u)$. Of course, the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$ also entails $D(w)$. We conclude that either D or some atom subsuming D must be contained in the repair type $f(u,w)$ and we are done.

Finally, we conclude by means of Theorem 3 and Lemma XIII that $\mathsf{rep}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}, s)$ IQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$.

For the remaining case where $\mathsf{QL} = \mathsf{CQ}$, we can prove the claim very similarly—we only need to replace simulations by homomorphisms. Assume that $\exists Y.\mathcal{B}$ is a CQ-repair of $\exists X.\mathcal{A}$ for $\mathcal{R}$ w.r.t. $\mathcal{T}$. So there is a homomorphism h from $\exists Y.\mathcal{B}$ to the CQ-saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$. Instead of the previous mapping f we now use the mapping $f\colon \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) \to \wp(\mathsf{Atoms}(\mathcal{R},\mathcal{T}))$ defined by

$$f(t) := \mathsf{Max}_{\sqsubseteq^\emptyset}\left(\left\{\, D \;\middle|\; \begin{array}{l} D \in \mathsf{Atoms}(\mathcal{R},\mathcal{T}),\ \mathcal{B} \not\models^{\mathcal{T}} D(t), \\ \text{and the matrix of } \mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A}) \text{ entails } D(h(t)) \end{array} \right\}\right)$$

for each object name t occurring in $\exists Y.\mathcal{B}$. As before, we can easily show that each such set $f(t)$ is a repair type, and thus the function s that is the restriction of f to the set Σ_{I} of all individual names is a repair seed function. Similarly as we have proven that the relation $\mathfrak{T}$ is a simulation, we can now demonstrate that the mapping k where $k(t) := y_{h(t),f(t)}$ for each $t \in \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B})$ is a homomorphism from $\exists Y.\mathcal{B}$ to $\mathsf{rep}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A}, s)$. With Theorem 2 and Lemma XIII we infer that $\mathsf{rep}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A}, s)$ CQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$. □

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 $\mathsf{Repairs}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A},\mathcal{R})$. 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* $\mathsf{QL} = \mathsf{IQ}$*.*

Proof. Proposition 8 shows that $\mathsf{Repairs}_{\mathsf{QL}}^{\mathcal{T}}(\exists X.\mathcal{A},\mathcal{R})$ contains only QL-repairs and further that it contains all optimal QL-repairs (up to QL-equivalence). Next, we show that it can be computed in exponential time.

The size of $\mathsf{Atoms}(\mathcal{R},\mathcal{T})$ is polynomial. Since each repair type is a subset of $\mathsf{Atoms}(\mathcal{R},\mathcal{T})$, there are exponentially many repair types for each object name that occurs in the saturation. Of course, we can compute in exponential time all repair types of a particular object name in the saturation by enumerating all subsets of $\mathsf{Atoms}(\mathcal{R},\mathcal{T})$ and then filtering out those not satisfying the three conditions in Definition 5.

According to Theorem 2 the CQ-saturation can be computed in exponential time, while Theorem 3 shows that IQ-saturations can always be computed in polynomial time. We conclude that the QL-saturation of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}$ contains at most exponentially many object names. As shown above, there are exponentially many repair types for each such object name. Since each object name in a canonical repair has the form $y_{t,\mathcal{K}}$ where t is an object name occurring in the saturation and where $\mathcal{K}$ is a repair type for t, we infer that each canonical repair contains at most exponentially many object names, and the set of these can be computed in exponential time.

Finally, the matrix of the canonical repair can be constructed in exponential time as follows.

- Iterate through all concept assertions in the saturation and, for each such $A(u)$ and for each copy $y_{u,\mathcal{K}}$, check whether $A \in \mathcal{K}$ and, if this is not the case, then add $A(y_{u,\mathcal{K}})$ to the matrix. There are exponentially many concept assertions $A(u)$ and exponentially many copies $y_{u,\mathcal{K}}$. Checking $A \in \mathcal{K}$ needs polynomial time, since each repair type has polynomial size. Adding a concept assertion to the matrix needs constant time. We conclude that all concept assertions can be generated in exponential time.
- Iterate through all role assertions in the saturation, and for each such $r(u, v)$ and for all copies $y_{u,\mathcal{K}}$ and $y_{v,\mathcal{L}}$, check whether $\mathcal{L}$ contains an atom subsuming C for each existential restriction $\exists r.C \in \mathcal{K}$ where the matrix of the saturation entails $C(v)$ and, if this is the case, then add $r(y_{u,\mathcal{K}}, y_{v,\mathcal{L}})$ to the matrix. There are exponentially many role assertions $r(u, v)$ and exponentially many copies $y_{u,\mathcal{K}}$ and $y_{v,\mathcal{L}}$. Checking the condition needs exponential

time, since each repair type has polynomial size, deciding subsumption can be done in polynomial time for $\mathcal{EL}$, and checking whether the saturation entails a concept assertion needs exponential time (since the saturation has exponential size). Adding a role assertion to the matrix needs constant time. We conclude that all role assertions can be generated in exponential time.

We can compute the subset of all optimal QL-repairs by filtering out non-optimal ones. In particular, we remove each canonical QL-repair that is QL-entailed by another canonical QL-repair. For the query language IQ, each such entailment text needs polynomial time in the size of the repairs, i.e., needs exponential time. For the query language CQ, we use an *NP* oracle that decides CQ-entailment. Of course, we need to conduct at most exponentially many such entailment tests and so we infer that the subset of all optimal QL-repairs can be computed in exponential time, using an *NP* oracle only for the case $\mathsf{QL} = \mathsf{CQ}$. □

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 $\mathcal{T} \coloneqq \emptyset$ 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 [4] 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.

Recall that, 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 $\mathcal{L}$, cf. Page 16. We already pointed out in [7] that, restricted to sets that contain only reduced concept descriptions and that do not contain $\sqsubseteq^{\emptyset}$-comparable concept descriptions, the cover relation $\leq$ is a partial order.

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

$$\mathsf{Succ}(\mathcal{K}, r, u) \coloneqq \{\, C \mid \exists r.C \in \mathcal{K} \text{ and the matrix of } \mathsf{sat}^{\mathcal{T}}_{\mathsf{QL}}(\exists X.\mathcal{A}) \text{ entails } C(u) \,\}.$$

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.*

Proof. The statement is a special case of Proposition VII, namely for $\mathcal{C} \coloneqq \mathcal{L} \cup \mathsf{Succ}(\mathcal{K}, r, u)$. □

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 defined as follows:

$$Y_0 \coloneqq \begin{cases} \{\, y_{t,\emptyset} \mid t \text{ is an object name occurring in } \mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A}) \,\} & \text{if } \mathsf{QL} = \mathsf{CQ} \\ \emptyset & \text{if } \mathsf{QL} = \mathsf{IQ}. \end{cases}$$

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 $Y^{\mathsf{IQ}}_m = \{\, y_{x,\{A_i\}} \mid i = 1,\ldots,n \,\}$. In the CQ case, the initial set of variables is $Y^{\mathsf{CQ}}_0 = \{y_{a,\emptyset}, y_{x,\emptyset}\}$. 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 $Y^{\mathsf{IQ}}_m \cup Y^{\mathsf{CQ}}_0$.

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 $\exists Y^{\mathsf{IQ}}_m.\mathcal{B}_m$ 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. We start with the easier case where the query language QL is IQ. Specifically, we show that the following relation $\mathfrak{S}$ is a simulation from $\exists Y.\mathcal{B}$ to $\exists Y_m.\mathcal{B}_m$.

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

1. By construction, each individual name a that occurs in $\exists Y.\mathcal{B}$ is also an individual name in $\exists Y_m.\mathcal{B}_m$. Recall that we use $y_{a,s(a)}$ as a synonym for a. It follows that the pair (a, a), which equals $(y_{a,s(a)}, y_{a,s(a)})$, is contained in $\mathfrak{S}$ for each individual name a.
2. Let the pair $(y_{t,\mathcal{K}}, y_{t,\mathcal{K}'})$ be in $\mathfrak{S}$ and consider a concept assertion $A(y_{t,\mathcal{K}})$ in $\mathcal{B}$. It follows that the saturation contains $A(t)$ and further that $A \notin \mathcal{K}$. Since $\mathcal{K}$ covers $\mathcal{K}'$, we infer that $A \notin \mathcal{K}'$. Consequently, $A(y_{t,\mathcal{K}'})$ is in $\mathcal{B}_m$.
3. Assume that $(y_{t,\mathcal{K}}, y_{t,\mathcal{K}'})$ is in $\mathfrak{S}$ and consider a role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ in $\mathcal{B}$. So the saturation contains $r(t, u)$ and we have $\mathsf{Succ}(\mathcal{K}, r, u) \leq \mathcal{L}$. Since $\mathcal{K}' \leq \mathcal{K}$, we infer that $\mathsf{Succ}(\mathcal{K}', r, u) \leq \mathsf{Succ}(\mathcal{K}, r, u)$, and thus $\mathsf{Succ}(\mathcal{K}', r, u) \leq \mathcal{L}$. It follows that there is a $\leq$-minimal repair type $\mathcal{L}'$ such that $\mathsf{Succ}(\mathcal{K}', r, u) \leq \mathcal{L}' \leq \mathcal{L}$. Since the IQ-Construction rule is not applicable to Y_m, it follows that $y_{u,\mathcal{L}'}$ is an object name that occurs in $\exists Y_m.\mathcal{B}_m$. Since $\mathcal{L}'$ covers $\mathsf{Succ}(\mathcal{K}', r, u)$, the role assertion $r(y_{t,\mathcal{K}'}, y_{u,\mathcal{L}'})$ is contained in $\mathcal{B}_m$. Since $\mathcal{L}$ covers $\mathcal{L}'$, the pair $(y_{u,\mathcal{L}}, y_{u,\mathcal{L}'})$ is contained in $\mathfrak{S}$.

Since $\exists Y_m.\mathcal{B}_m$ is a subset of the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists Y_m.\mathcal{B}_m)$, the above relation $\mathfrak{S}$ is a simulation from $\exists Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists Y_m.\mathcal{B}_m)$. An application of Theorem 3 shows that $\exists Y_m.\mathcal{B}_m$ IQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$.

It remains to consider the case where the query language QL is CQ. We will construct a sequence $(h_0, h_1, \ldots, h_n)$ of mappings $h_i \colon \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) \to \Sigma_{\mathsf{O}}(\exists Y_m.\mathcal{B}_m)$ that ends with a homomorphism h_n from $\exists Y.\mathcal{B}$ to $\exists Y_m.\mathcal{B}_m$. Initialize the first mapping

$$\begin{aligned} h_0 \colon \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) &\to \Sigma_{\mathsf{O}}(\exists Y_m.\mathcal{B}_m) \\ y_{t,\mathcal{K}} &\mapsto \begin{cases} y_{t,s(t)} & \text{if } t \in \Sigma_{\mathsf{I}} \text{ and } s(t) \leq \mathcal{K} \\ y_{t,\emptyset} & \text{otherwise} \end{cases} \end{aligned}$$

The following invariants will be satisfied for all mappings in the sequence.

Invariant 1. If $h_i(y_{t,\mathcal{K}}) = y_{u,\mathcal{L}}$, then $u = t$ and $\mathcal{L} \leq \mathcal{K}$.

Invariant 2. If $h_i(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_i}$ and $h_{i+1}(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_{i+1}}$, then $\mathcal{K}_i \leq \mathcal{K}_{i+1}$.

Of course, the first mapping h_0 satisfies Invariant 1 and has its range in $\Sigma_\mathsf{I} \cup Y_0$, which is a subset of $\Sigma_\mathsf{O}(\exists\, Y_m.\, \mathcal{B}_m)$.

A *defect* of a mapping h_i is a role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ in $\mathcal{B}$ such that its image $r(h_i(y_{t,\mathcal{K}}), h_i(y_{u,\mathcal{L}}))$ is not in $\mathcal{B}$. In the following, we will show how a successor mapping h_{i+1} can be constructed if h_i has a defect. Let h_i be the last mapping constructed so far, and assume that it has a defect, which is a role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ in $\mathcal{B}$ such that $\mathsf{Succ}(\mathcal{K}_i, r, u) \not\leq \mathcal{L}_i$ for $y_{t,\mathcal{K}_i} \coloneqq h_i(y_{t,\mathcal{K}})$ and $y_{u,\mathcal{L}_i} \coloneqq h_i(y_{u,\mathcal{L}})$.

- We first show that $\mathsf{Succ}(\mathcal{K}_i, r, u) \leq \mathcal{L}$. Consider an existential restriction $\exists\, r.\, C \in \mathcal{K}_i$ where the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists\, X.\, \mathcal{A})$ entails $C(u)$. According to Invariant 1, $\mathcal{K}$ covers $\mathcal{K}_i$. Thus, there exists an existential restriction $\exists\, r.\, D \in \mathcal{K}$ such that D subsumes C. It follows that the matrix of $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists\, X.\, \mathcal{A})$ entails $D(u)$ as well, and so $D \in \mathsf{Succ}(\mathcal{K}, r, u)$. According to Definition 7 and since the canonical repair contains the role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$, we infer that $\mathcal{L}$ covers $\mathsf{Succ}(\mathcal{K}, r, u)$. Consequently, $\mathcal{L}$ contains some atom subsuming D and thus also subsuming C.
- Due to Invariant 1, $\mathcal{L}$ covers $\mathcal{L}_i$. We have just seen that $\mathcal{L}$ also covers $\mathsf{Succ}(\mathcal{K}_i, r, u)$. In summary, it follows that $\mathcal{L}_i \cup \mathsf{Succ}(\mathcal{K}_i, r, u) \leq \mathcal{L}$.
- Since $\mathcal{L}_i$ does not cover $\mathsf{Succ}(\mathcal{K}_i, r, u)$, we infer that $\mathcal{L}_i < \mathcal{L}_i \cup \mathsf{Succ}(\mathcal{K}_i, r, u)$.
- Now choose a $\leq$-minimal repair type $\mathcal{L}_{i+1}$ for u such that $\mathcal{L}_i \cup \mathsf{Succ}(\mathcal{K}_i, r, u) \leq \mathcal{L}_{i+1} \leq \mathcal{L}$ and define $h_{i+1} \coloneqq h_i$ except that $h_{i+1}(y_{u,\mathcal{L}}) \coloneqq y_{u,\mathcal{L}_{i+1}}$. Note that $\mathcal{L}_i < \mathcal{L}_{i+1}$, and so $h_i \neq h_{i+1}$. Clearly, both invariants are satisfied.
- Since the CQ-Construction rule is not applicable to Y_m, it follows that $y_{u,\mathcal{L}_{i+1}}$ is contained in $\Sigma_\mathsf{I} \cup Y_m = \Sigma_\mathsf{O}(\exists\, Y_m.\, \mathcal{B}_m)$, i.e., the mapping h_{i+1} has its range in $\Sigma_\mathsf{O}(\exists\, Y_m.\, \mathcal{B}_m)$.

Next, we show that the sequence must be finite. For this purpose, we define a partial order $\leq$ on the mappings as follows: $h_i \leq h_j$ if, for each $y_{t,\mathcal{K}} \in \Sigma_\mathsf{O}(\exists\, Y.\, \mathcal{B})$, we have $\mathcal{K}_i \leq \mathcal{K}_j$ where $y_{t,\mathcal{K}_i} \coloneqq h_i(y_{t,\mathcal{K}})$ and $y_{t,\mathcal{K}_j} \coloneqq h_j(y_{t,\mathcal{K}})$. Note that $\leq$ is indeed a partial order, since the covers relation is a partial order on repair types. According to the above construction, $h_i < h_{i+1}$ is always satisfied, i.e., the sequence is strictly increasing. Since $\Sigma_\mathsf{O}(\exists\, Y.\, \mathcal{B})$ and its subset $\Sigma_\mathsf{O}(\exists\, Y_m.\, \mathcal{B}_m)$ are both finite, there are only finitely many mappings of type $\Sigma_\mathsf{O}(\exists\, Y.\, \mathcal{B}) \to \Sigma_\mathsf{O}(\exists\, Y_m.\, \mathcal{B}_m)$. We infer that after finitely many iterations the sequence cannot be extended by means of the above construction, i.e., it must end with a mapping h_n that is free of defects.

Claim. The last mapping h_n is a homomorphism from $\exists\, Y.\, \mathcal{B}$ to $\exists\, Y_m.\, \mathcal{B}_m$.

1. Consider an individual name a, which has the synonym $y_{a,s(a)}$ in $\mathcal{B}$. Let $y_{a,\mathcal{K}_i} \coloneqq h_i(y_{a,s(a)})$ for each index i. Due to the invariants, we have

$$s(a) = \mathcal{K}_0 \leq \mathcal{K}_1 \leq \cdots \leq \mathcal{K}_n \leq s(a)$$

and so we conclude that $\mathcal{K}_i = s(a)$ for each index i. We conclude that $h_i(y_{a,s(a)}) = y_{a,s(a)}$ for each i, i.e., and thus in particular that $h_n(a) = a$.

2. Consider a concept assertion $A(y_{t,\mathcal{K}})$ in $\mathcal{B}$ and its image $A(y_{t,\mathcal{K}_n})$ where $y_{t,\mathcal{K}_n} \coloneqq h_n(y_{t,\mathcal{K}})$. We will show that $A(y_{t,\mathcal{K}_n})$ is in $\mathcal{B}_m$. By assumption and according to Definition 7, $A(t)$ is a concept assertion in the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$ and $A \notin \mathcal{K}$. Since h_n satisfies Invariant 1, $\mathcal{K}$ covers $\mathcal{K}_n$. It follows that $A \notin \mathcal{K}_n$, and so we conclude that $A(y_{t,\mathcal{K}_n})$ is indeed in $\mathcal{B}_m$, cf. Definitions 7 and 12.
3. Consider a role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ in $\mathcal{B}$ and its image $r(y_{t,\mathcal{K}_n}, y_{u,\mathcal{L}_n})$ where $y_{t,\mathcal{K}_n} \coloneqq h_n(y_{t,\mathcal{K}})$ as well as $y_{u,\mathcal{L}_n} \coloneqq h_n(y_{u,\mathcal{L}})$. By assumption and according to Definition 7, $r(t, u)$ is a role assertion in the saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists X.\mathcal{A})$. Since h_n is free of defects, it follows that $\mathsf{Succ}(\mathcal{K}_n, r, u) \leq \mathcal{L}_n$ and thus the role assertion $r(y_{t,\mathcal{K}_n}, y_{u,\mathcal{L}_n})$ is in $\mathcal{B}_m$, cf. Definitions 7 and 12.

Since $\exists Y_m.\mathcal{B}_m$ is a sub-qABox of its saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists Y_m.\mathcal{B}_m)$, it follows that h_n is also a homomorphism from $\exists Y.\mathcal{B}$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{CQ}}(\exists Y_m.\mathcal{B}_m)$. Theorem 2 implies that $\exists Y_m.\mathcal{B}_m$ CQ-entails $\exists Y.\mathcal{B}$ w.r.t. $\mathcal{T}$. □

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.*

We have explicitly shown this theorem for QL-equivalence without a TBox, but this trivially implies that the equivalence also holds w.r.t. the TBox $\mathcal{T}$.

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).

As corpus for our evaluation, we chose the ontologies used in the 2015 OWL Reasoner Competition for the track OWL EL Realisation [29], 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

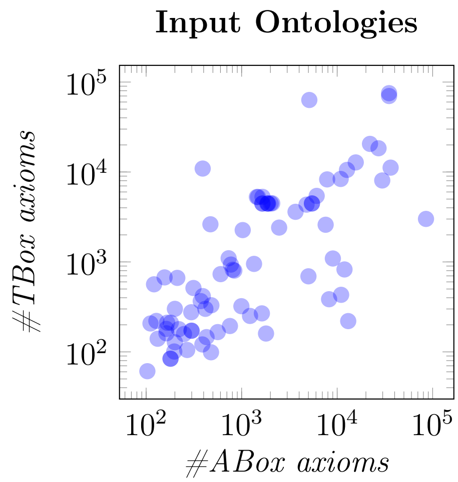


Figure II: Number of TBox and ABox axioms in the input ontologies.

unsupported axioms. From these ontologies, we kept those that had at most 100,000 axioms in total. The resulting corpus contained 80 ontologies.

In Fig. II we show how the numbers of TBox and ABox axioms in the resulting ontologies distribute. Specifically, each point in the graph corresponds to an ontology in the corpus after restricting to $\mathcal{EL}$ and flattening of the ABox, where the x-axis shows its number of ABox assertions, and the y-axis its number of TBox axioms. Note that in this graph, as well as in the plots that follow, we use logarithmic scale. We can see that the corpus used is not only balanced in terms of overall size, but also in terms of ratio between TBox and ABox axioms, including both ontologies where the TBox is small compared to the ABox, and where it is large compared to the ABox. Note however that by flattening of the ABox, we often increased the TBox size significantly depending on the ABox.

We implemented our methods in Java, using the OWL-API[1] for parsing OWL ontologies, and ELK [23] 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.

We first discuss the results of the IQ-repairs in Section 6.1, and then discuss our results for the CQ-repairs in Section 6.2.

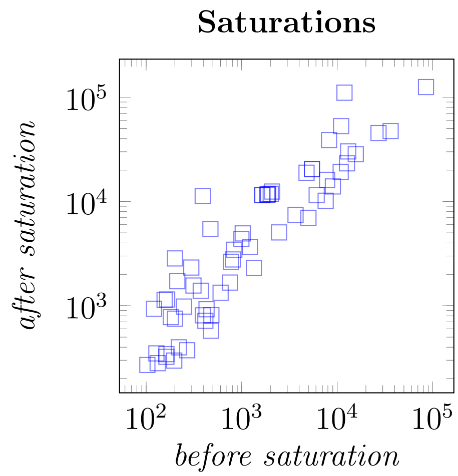


Figure III: Number of assertions before and after IQ-saturation.

6.1 IQ-Repairs

Since it is a precondition of our repair approach, we first saturated the ontologies using the IQ-saturation rules of Figure 2. We used a timeout of 60 minutes for every saturation. This way, we successfully computed IQ-saturations of every ontology. 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. In Figure III, we compare the sizes of the ABoxes before and after the saturation.

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, 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, 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. Surprisingly, although C was not required to be complex, this ruled out 12 ontologies for which (in the version restricted to pure $\mathcal{EL}$) no concept with at least one instance could be found.

For both scenarios, we selected a random seed function for the obtained repair request.

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

For each ontology and each scenario, we attempted to compute optimised IQ-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-repair could be computed within the timeout and without memory exceptions for only 52.9 % of the ontologies.
- For S1, we could compute the optimized IQ-repair in 99.9 % of all attempts.
- For S2, 98.9 % of IQ-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. IV gives more information on the number of objects and assertions in the computed repairs. On the left, we consider scenario S1, and on the right, we consider scenario S2. On the top, we show how the numbers of object names occurring in the canonical and the optimised repairs changed: The purple triangles indicate the number of object names occurring in the canonical repairs minus the number of objects in the saturated input ABox (for the ontologies for which this computation succeeded). The red dots show the difference between the number of objects in the optimized repairs and in the saturated input ABoxes. This difference can become negative since objects without assertions in the repairs are not counted. On the bottom of Fig. IV, we look at the number of assertions: the blue crosses indicate the difference between the number of assertions in the optimized repairs and in the saturated input ABoxes for the scenarios S1 and S2, respectively. It should not be surprising that, in S2, where the repair requests are larger, more assertions may be removed than are added by the copying of objects and assertions.

6.2 CQ-Repairs

We ran the same experiments for CQ-repairs. The CQ-saturation was computed using the rule engine VLog [11] through the Java facade Rulewerk.[3] Note that the CQ-saturation only terminates for cycle-restricted TBoxes. We thus excluded from our experiment all ontologies for which the IQ-saturation contained a cycle between introduced variables, as those would clearly not be cycle-restricted. This excluded 16 ontologies. For the remaining 64 ontologies, we computed CQ-

[3] https://github.com/knowsys/rulewerk

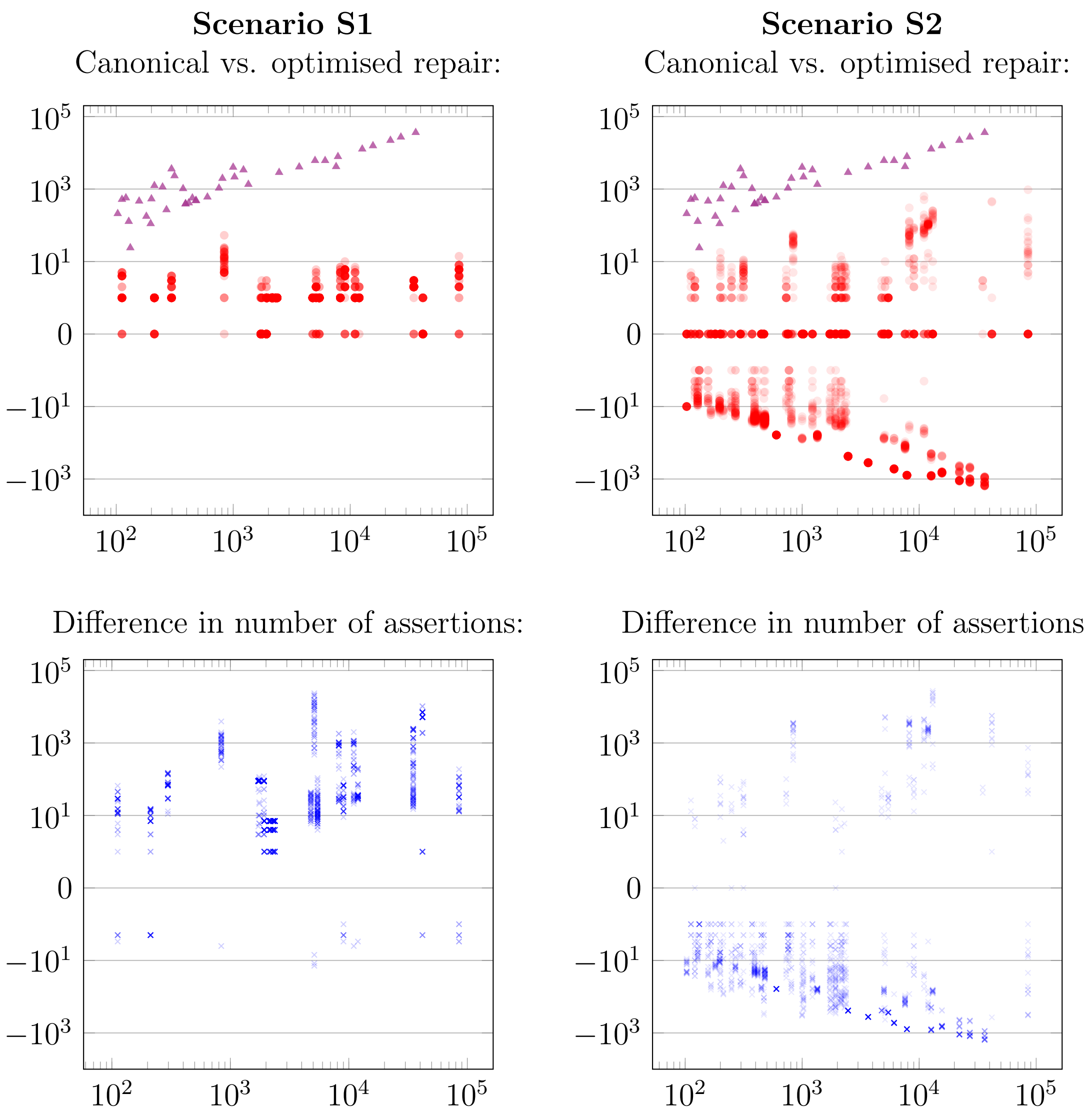


Figure IV: Evaluation results for IQ-repairs. On the top, we show the difference of the number of object names in the canonical repairs (purple triangle) with the same difference, but restricted to objects occurring in assertions, for the optimised repairs (red circle). Below, we look at the difference in the number of assertions. In each graph, the x-axis shows the number of assertions in the saturated input ontology, and the y-axis the observed difference.

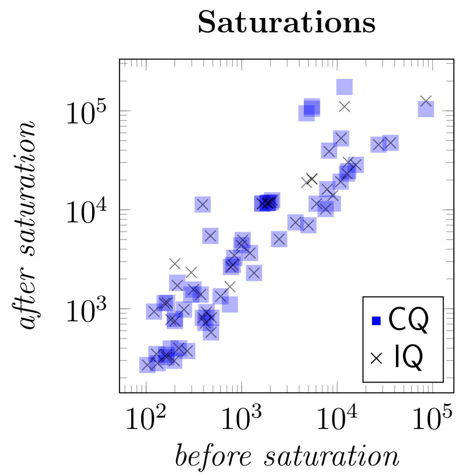


Figure V: Number of assertions before and after the CQ-saturation, with corresponding results for the IQ-saturation added for comparison.

saturations with a timeout of 60 minutes. This was successful for 62 ontologies.

In Figure V, we show the size of the CQ-saturations, where the x-axis again shows the ABox size before the saturation. For comparison, we also included the sizes of the IQ-saturations for those ontologies for which we could compute the CQ-saturation. Surprisingly, even though, in theory, the CQ-saturation can be of size exponential in the TBox size, whereas the IQ-saturations are polynomial, the CQ-saturations that could be computed without timeout were often of similar size to the IQ-saturations. In fact, in a lot of cases, they had the same size, because no variables were added during the saturation. Indeed sometimes even neither of the saturations added any assertions, since the ABox was already saturated. There were even cases where the CQ-saturation was smaller: this was due to the non-determinism of the order in which the $\sqsubseteq$-rules are applied: Note that a concept assertion of the form $(\exists r.C)(t)$ is not added if we first add $C(u)$ for an r-successor u of t. It seems that VLog does a good job at determining which order of rule-applications leads to fewer variables added, while we did not add any optimisation for this for the IQ-saturation.

For the 62 ontologies for which we could compute a saturation, 4 had no non-trivial entailment of the form $C(a)$, and 44 had no such entailment where C is not of the form $A(a)$ or $\exists r.\top(a)$ with A a concept name. Consequently, 18 ontologies could be used for scenario S1, and 58 ontologies could be used for scenario S2.

For the actual repairs performed on the CQ-saturations, we used a timeout of 10 minutes as for the IQ-repairs. The number of variables required for the canonical CQ-repairs could be computed in 62.1 % of cases. We were able to compute all optimised CQ-repairs for scenario S1, and in 99.9 % of cases, we could compute the optimised CQ-repairs for scenario S2 (in fact, only one of the experimental runs caused a timeout). The results of our experiments can be seen in Figure VI,

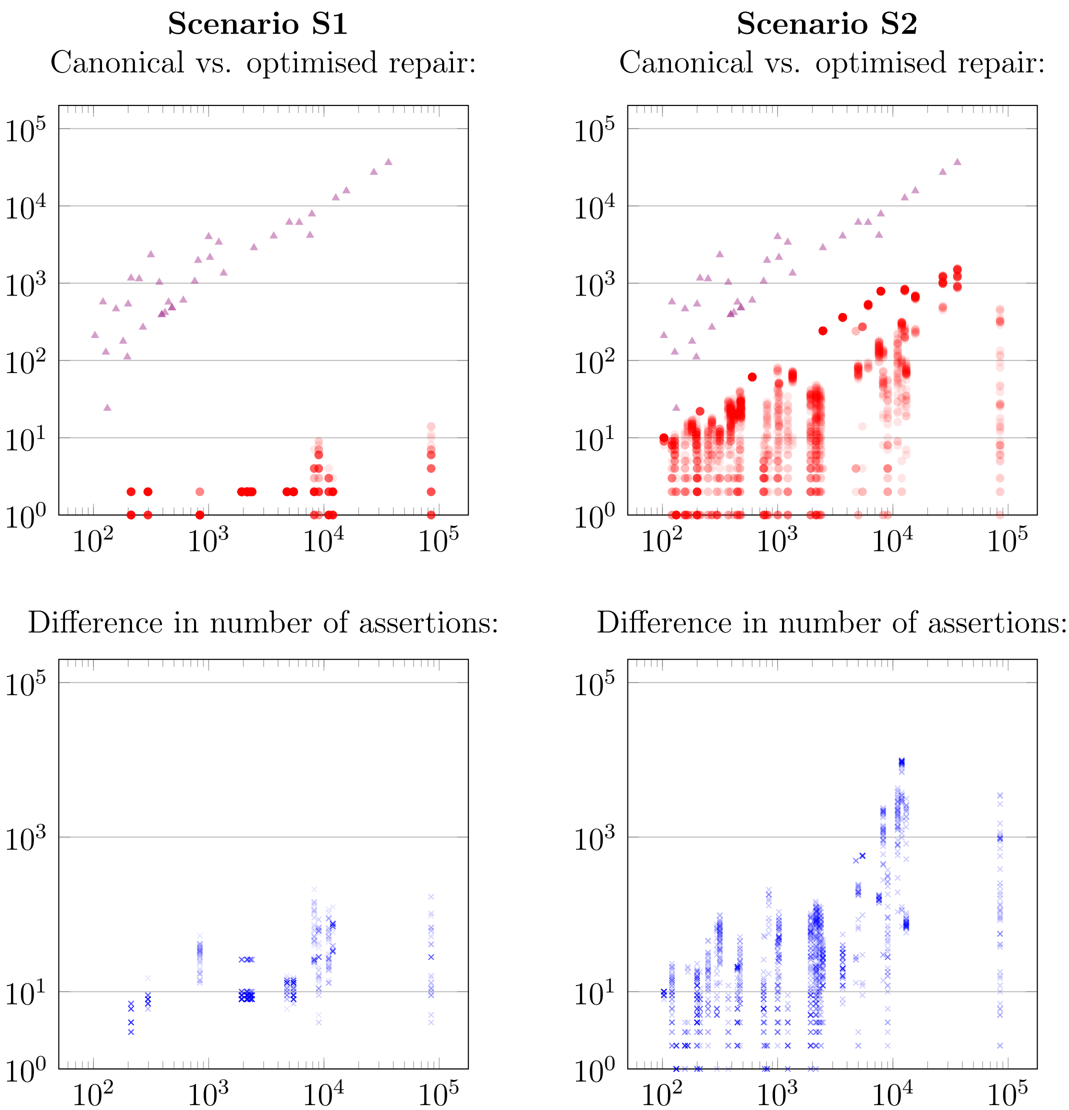
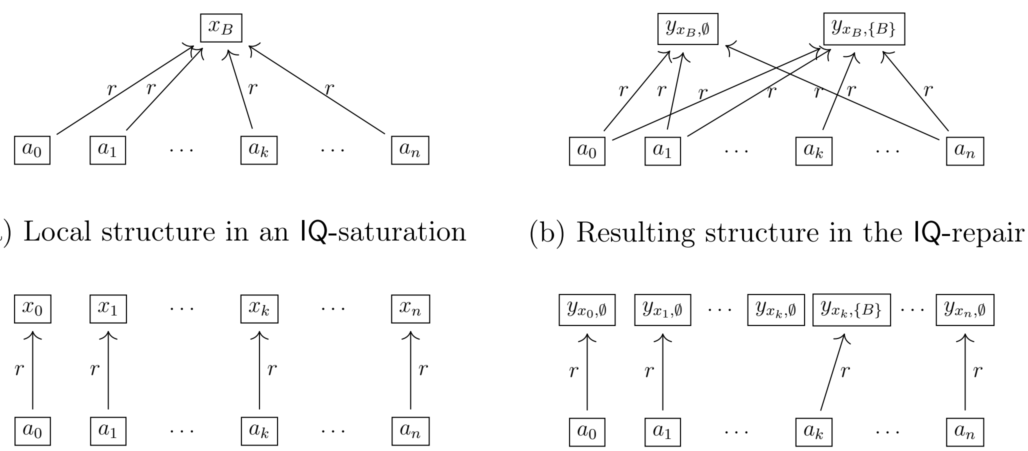


Figure VI: Evaluation results for the CQ-repairs. On the top, we show the difference of the number of object names in the canonical repairs (purple triangle) with the same difference, but restricted to objects occurring in assertions, for the optimised repairs (red circle). Below, we look at the difference in the number of assertions. In each graph, the x-axis shows the numer of assertions in the saturated input ontology, and the y-axis the observed difference.

which is similar to Figure IV, but shows the corresponding results for the CQ case. Note that the patterns of the triangles representing sizes of canonical repairs do not differ much between Figure IV and Figure VI, because the saturations, on which their computation is based, do not differ much in size, cf. Figure V. Further note that for scenario S1, the graph captures fewer ontologies than Figure IV. In contrast to the IQ-repairs, the number of assertions, and also the number of object names occurring in any assertion, always increased by at least 1. This is a direct consequence of our method, but in most cases indeed unavoidable irrespective of the approach. Note that, in order to repair an assertion $C(u)$, where u is an individual, the matrix of the repair would have to entail at least $C(x)$ for some fresh variable x. Still, we see that, in each case, the optimised repairs were significantly smaller than the canonical repairs. What is surprising is that, for scenario S1, often fewer variables and assertions were added by the repair as for IQ-repairs. This has to do with the different shape of the saturations, which we are going to explain in the following.



(a) Local structure in an IQ-saturation (b) Resulting structure in the IQ-repair

(c) Local structure in a CQ-saturation (d) Resulting structure in the CQ-repair

Figure VII: Explanation for large differences in number of role assertions between IQ-saturation and IQ-repair, which could not be observed for CQ-entailment

For instance, computing the IQ-saturation could introduce a fresh variable x_B that is an r-successor of several individual names $a_1, \ldots, a_n$, since each a_i is an instance of A and the TBox contains the concept inclusion $A \sqsubseteq \exists r.B$, cf. Figure VII (a). Further assume that only one of the individual names, say a_k, needs to be repaired in a way such that the single successor x_B is split up into multiple copies, e.g., because the seed function maps a_k to the repair type $\{A, \exists r.B\}$ and maps the others to the empty repair type. The resulting IQ-repair would contain two copies of x_B, namely one for the empty repair type and one for the repair type $\{B\}$, and both will be r-successors of each individual name a_i except a_k, cf. Figure VII (b). Consequently, $n-1$ role assertions in the IQ-saturation get duplicated during the construction of the IQ-repair.

In contrast, the CQ-saturation for the same setting would contain a separate r-successor for each of the individual names $a_1, \ldots, a_n$, cf. Figure VII (c). During the construction of the CQ-repair only the successor of a_k is split into two copies, of which one remains a successor of a_k, cf. Figure VII (d). Here, the number of role assertions would remain unchanged.

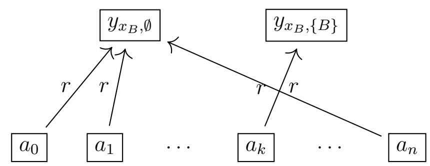


Figure VIII: Resulting structure in the modified IQ-repair

In order to keep the number of role assertions in an IQ-repair small, we could apply the following modification to our construction of the optimized IQ-repair: for each role assertion $r(t, u)$ in the saturation and for all copies $y_{t,\mathcal{K}}, y_{u,\mathcal{L}} \in \Sigma_\mathsf{I} \cup Y_m$, add the role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ to the matrix of the repair only if $\mathcal{L}$ is $\leq$-minimal.[4] In our example, this would produce a repair as depicted in Figure VIII where, as for the CQ-repair, the number of role assertions would not change compared to the saturation. Correctness of this optimization can be shown by a simple modification of the proof of Proposition 13.

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

[4] That is, add $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ to the matrix if $\mathcal{L}$ covers $\mathsf{Succ}(\mathcal{K}, r, u)$ and there is no $y_{u,\mathcal{M}} \in \Sigma_\mathsf{I} \cup Y_m$ such that $\mathcal{M} < \mathcal{L}$ and $\mathsf{Succ}(\mathcal{K}, r, u) \leq \mathcal{M}$.

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].[5]

References

[1] F. Baader, S. Borgwardt, and B. Morawska. Extending unification in $\mathcal{EL}$ towards general TBoxes. In *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 568–572. AAAI Press/The MIT Press, 2012.

[2] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In L. P. Kaelbling and A. Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 364–369. Professional Book Center, 2005.

[3] F. Baader, I. Horrocks, C. Lutz, and U. Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[4] F. Baader, F. Kriegel, and A. Nuradiansyah. Privacy-preserving ontology publishing for $\mathcal{EL}$ instance stores. In F. Calimeri, N. Leone, and M. Manna, editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 323–338. Springer, 2019.

[5] F. Baader, F. Kriegel, A. Nuradiansyah, and R. Peñaloza. Computing safe anonymisations of quantified aboxes w.r.t. $\mathcal{EL}$ policies (extended version). LTCS-Report 20-09, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2020.

[6] F. Baader, F. Kriegel, A. Nuradiansyah, and R. Peñaloza. Making repairs in description logics more gentle. In M. Thielscher, F. Toni, and F. Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, pages 319–328. AAAI Press, 2018.

[5] 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).

[7] F. Baader, F. Kriegel, A. Nuradiansyah, and R. Peñaloza. Computing compliant anonymisations of quantified aboxes w.r.t. $\mathcal{EL}$ policies. In J. Z. Pan, V. A. M. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, editors, *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I*, volume 12506 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2020.

[8] F. Baader and B. Suntisrivaraporn. 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] T. C. Boyle. *Talk to Me.* Bloomsbury Publishing, 2021. To appear.

[10] A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In F. Neven, C. Beeri, and T. Milo, editors, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 260–271. ACM, 2003.

[11] D. Carral, I. Dragoste, L. González, C. J. H. Jacobs, M. Krötzsch, and J. Urbani. Vlog: A rule engine for knowledge graphs. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, editors, *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference*, volume 11779 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2019.

[12] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang. Hermit: An OWL 2 reasoner. *J. Autom. Reason.*, 53(3):245–269, 2014.

[13] B. C. Grau, I. Horrocks, M. Krötzsch, C. Kupke, D. Magka, B. Motik, and Z. Wang. Acyclicity notions for existential rules and their application to query answering in ontologies. *J. Artif. Intell. Res.*, 47:741–808, 2013.

[14] B. C. Grau and E. V. Kostylev. Logical foundations of linked data anonymisation. *J. Artif. Intell. Res.*, 64:253–314, 2019.

[15] K. Gödel. *Über die Vollständigkeit des Logikkalküls*. Doctoral thesis, Universität Wien, Wien, Austria, 1929.

[16] V. Haarslev, K. Hidde, R. Möller, and M. Wessel. The RacerPro knowledge representation and reasoning system. *Semantic Web*, 3(3):267–277, 2012.

[17] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke. Computing simulations on finite and infinite graphs. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 453–462. IEEE Computer Society, 1995.

[18] R. Hoehndorf, P. N. Schofield, and G. V. Gkoutos. The role of ontologies in biological and biomedical research: A functional perspective. *Brief. Bioinform.*, 16(6):1069–1080, 2015.

[19] M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in OWL. In A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, pages 323–338. Springer, 2008.

[20] I. Horrocks, L. Li, D. Turi, and S. Bechhofer. The instance store: DL reasoning with large numbers of individuals. In V. Haarslev and R. Möller, editors, *Proceedings of the 2004 International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8, 2004*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[21] D. S. Johnson and A. C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. In J. D. Ullman and A. V. Aho, editors, *Proceedings of the ACM Symposium on Principles of Database Systems, March 29-31, 1982, Los Angeles, California, USA*, pages 164–169. ACM, 1982.

[22] A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. In *Proc. of ISWC'07*, volume 4825 of *Lecture Notes in Computer Science*, pages 267–280. Springer-Verlag, 2007.

[23] Y. Kazakov, M. Krötzsch, and F. Simancik. The incredible ELK - from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *Journal of Automed Reasoning*, 53(1):1–61, 2014.

[24] R. Küsters. *Non-standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.

[25] J. S. C. Lam, D. H. Sleeman, J. Z. Pan, and W. W. Vasconcelos. A fine-grained approach to resolving unsatisfiable ontologies. *J. Data Semant.*, 10:62–95, 2008.

[26] C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.*, 45(2):194–228, 2010.

[27] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979.

[28] T. Meyer, K. Lee, R. Booth, and J. Z. Pan. 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.

[29] B. Parsia, N. Matentzoglu, R. S. Gonçalves, B. Glimm, and A. Steigmiller. The OWL Reasoner Evaluation (ORE) 2015 competition report. *Journal of Automed Reasoning*, 59(4):455–482, 2017.

[30] B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In A. Ellis and T. Hagino, editors, *Proc. of the 14th International Conference on World Wide Web (WWW'05)*, pages 633–640. ACM, 2005.

[31] R. Rosati. On conjunctive query answering in $\mathcal{EL}$. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A. Turhan, and S. Tessaris, editors, *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[32] S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In G. Gottlob and T. Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 355–362, Acapulco, Mexico, 2003. Morgan Kaufmann, Los Altos.

[33] S. Schlobach, Z. Huang, R. Cornet, and F. Harmelen. Debugging incoherent terminologies. *J. Automated Reasoning*, 39(3):317–349, 2007.

[34] A. Steigmiller, T. Liebig, and B. Glimm. Konclude: System description. *J. Web Semant.*, 27-28:78–85, 2014.

[35] N. Troquard, R. Confalonieri, P. Galliani, R. Peñaloza, D. Porello, and O. Kutz. Repairing ontologies via axiom weakening. In S. A. McIlraith and K. Q. Weinberger, editors, *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*, pages 1981–1988. AAAI Press, 2018.