Beyond Optimal: Interactive Identification of Better-than-optimal Repairs

Francesco Kriegel
Theoretical Computer Science, Technische Universität Dresden
Dresden, Germany
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Germany
francesco.kriegel@tu-dresden.de

Abstract

We propose an interactive repair method for the description logic $\mathcal{EL}$ that is based on the optimal-repair framework. The obtained repair might not be optimal in the theoretical sense, i.e. more than a minimal amount of consequences might have been removed — but from a practical perspective it is superior to a theoretically optimal repair as the interaction strategy enables the users to identify further faulty consequences connected to the initially reported errors.

CCS Concepts

• **Theory of computation → Description logics**; • **Computing methodologies → Description logics**; **Ontology engineering**.

Keywords

Knowledge-base repair, Optimal repair, Disputable consequence, Interactive repair

ACM Reference Format:
Francesco Kriegel. 2025. Beyond Optimal: Interactive Identification of Better-than-optimal Repairs. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31 – April 4, 2025, Catania, Italy.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3672608.3707750

1 Introduction

Knowledge-based systems (KBS) represent complex domains in an explicit, structured manner and comprise an inference engine that reasons about the domain to draw precise conclusions and to answer queries. Many KBS additionally include an explanation component that makes transparent the derivation of a conclusion by showing a human-readable proof to the user. A modern foundation of KBS is the family of Description Logics (DLs) [3]. Knowledge is represented as a DL knowledge base (KB) consisting of an assertion box (ABox) and a DL ontology, which is further subdivided into a terminology box (TBox) and a role box (RBox). Complete domain knowledge is not required to build a KB as DLs adopt the open-world assumption and so allow for statements of which the truth cannot be determined from the represented knowledge. Factual assertions concerning specific individuals of the domain are declared in the ABox, such as classifications of the individuals into concepts and connections between the individuals by roles. The ontology governs all individuals and defines the domain's terminology in terms of concepts and their hierarchy (in the TBox), roles and their characteristics (in the RBox), as well as further rules and constraints.

The separation of instance-level and schema-level knowledge facilitates using the same ontology across multiple ABoxes. This is valuable in applications where sensitive data is processed and privacy must be protected, e.g. in the medical domain. For example, The Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) is a multilingual ontology that describes medical terms used in clinical documentation and reporting, and is the most comprehensive computer-processable clinical terminology in the world. SNOMED CT is used in clinical decision support systems to assist healthcare professionals in making accurate diagnoses, suggesting appropriate treatments, and predicting outcomes based on patient-specific information. Other domains in which DLs have been employed are e-commerce [14, 27], finance, biology, and car manufacturing and Industry 4.0 [28]. Moreover, DLs serve as the logical foundation of the Web Ontology Language (OWL) [16].

Even though DL-based KBS produce logically correct and explainable inferences, faulty conclusions might be drawn if the KB itself contains errors. Of course, the KB should then be appropriately repaired. The classical method is to pinpoint the statements in the KB from which the incorrect conclusion was drawn, and then either delete a minimal number of them such that the observed error vanishes or present these statements to knowledge engineers and domain experts for rectification. However, often only parts of statements are erroneous and thus deletion of whole statements would erase too much. On the other hand, it might be difficult for the experts to correct the statements since they first need to understand how the faulty consequence is inferred from them.

The $\mathcal{EL}$ family [1, 2, 17, 24, 25, 26] stands out from the various DLs available. Since $\mathcal{EL}$ strikes a balance between expressivity and computational complexity, it offers short reasoning latency and scalability to even large ontologies, which makes it an ideal choice for many applications. For example, SNOMED CT is formulated in $\mathcal{EL}$ and OWL comprises the profile OWL 2 EL based on $\mathcal{EL}$. Currently the fastest $\mathcal{EL}$ reasoner is ELK [18], which is a highly optimized, multi-threaded implementation of the polynomial-time completion algorithm. It can classify SNOMED CT (with more than 360,000 concepts) in a few seconds on a modern laptop.

In this article, we propose an interactive repair method for $\mathcal{EL}$ that surmounts both practicability issues of the classical method. As underlying repair framework we employ the optimal repairs [5, 8], which resolve errors not by deleting a minimal number of

SAC '25, March 31 – April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707750

statements but by modifying the KB such that only minimally many consequences are removed (including the observed faulty ones). Moreover, the best such repair is interactively identified in a top-down manner. Unlike the classical method, the experts do not need to consult, in a bottom-up manner, proofs of the unwanted consequence to appropriately correct the KB. Instead, they start with the reported error broken down into atomic statements and proceed towards logical causes of identified faulty statements, finally reaching statements in the KB. Through this guidance, the experts' workload is significantly lowered.

From a technical perspective, each optimal repair can be obtained from the input by saturation and then delete and copy operations. To ensure that no consequence is lost unnecessarily, the input ABox is initially saturated by adding statements implied by the TBox. While deletions are then necessary to remove the unwanted consequences, the copy mechanism ensures that not too many consequences get lost since each of the copies can be modified differently. The interaction process terminates in polynomial time and allows the experts to efficiently control these three operations and thus how the repair is constructed.

(1) *Control of the delete operation:* The causes of unwanted consequences are identified, which are afterwards used to compile a plan which statements need to be removed to obtain the repair.
(2) *Control of the copy operation:* When an object is split into copies that are modified in different ways, it is explored which copies actually exist in the underlying domain such that the others can be eliminated.
(3) *Control of the saturation:* A second phase is devoted to statements entailed by the input that are still undecided and of which all substantiations entail a rejected query. These disputable consequences have no substantiations in the repair anymore and should thus be investigated.

An implementation of the underlying repair construction as well as the smart interaction strategy is available at https://github.com/francesco-kriegel/interactive-optimal-repairs. It comes in form of a plug-in for the KB editor Protégé. For space restrictions, some technical details such as proofs needed to be outsourced [21].

2 Preliminaries

First, we provide general definitions regarding optimal repairs. To this end, we assume an arbitrary model-based logic consisting of a set of all *statements*, a set of all *interpretations*, and a relation $\models$ between them such that $\mathcal{I} \models \alpha$ indicates that the interpretation $\mathcal{I}$ *satisfies* the statement α. A *knowledge base (KB)* $\mathcal{K}$ is a finite set of statements, and we say that $\mathcal{I}$ is a *model* of $\mathcal{K}$ and write $\mathcal{I} \models \mathcal{K}$ if $\mathcal{I}$ satisfies every statement in $\mathcal{K}$. Moreover, $\mathcal{K}$ is *consistent* if it has a model, and $\mathcal{K}$ *entails* another KB $\mathcal{K}'$, written $\mathcal{K} \models \mathcal{K}'$, if every model of $\mathcal{K}$ is also one of $\mathcal{K}'$. We further assume that the statements are subdivided into *assertions* and *ontological statements*, and so each KB is a disjoint union $\mathcal{A} \uplus \mathcal{O}$ of an *assertion box (ABox)* $\mathcal{A}$ consisting of assertions and an *ontology* $\mathcal{O}$ with ontological statements.

Definition 2.1. A *repair request* is an assertion set $\mathcal{P} \coloneqq \mathcal{P}_+ \uplus \mathcal{P}_-$ partitioned into an *addition part* $\mathcal{P}_+$ and a *removal part* $\mathcal{P}_-$. Of a consistent KB $\mathcal{K} \coloneqq \mathcal{A} \uplus \mathcal{O}$, an *ABox repair* for $\mathcal{P}$ is an ABox $\mathcal{B}$ s.t.

- $\mathcal{B} \cup \mathcal{O}$ is consistent,
- $\mathcal{B} \cup \mathcal{O} \models \alpha$ for each $\alpha \in \mathcal{P}_+$, and
- $\mathcal{B} \cup \mathcal{O} \not\models \beta$ for each $\beta \in \mathcal{P}_-$.

If there is an ABox repair, then $\mathcal{P}$ is *feasible* w.r.t. $\mathcal{K}$.

In the above definition, we treat the ABox as refutable and the ontology as static. We may just use the denotation "repair" when no confusion can arise. Instead of $\alpha \in \mathcal{P}_+$ we may also write $+\alpha \in \mathcal{P}$, and likewise $-\beta \in \mathcal{P}$ for $\beta \in \mathcal{P}_-$. We observe that a repair request $\mathcal{P}$ is feasible iff. $\mathcal{P}_+$ is a repair iff. $\mathcal{P}_+ \cup \mathcal{O}$ is consistent and entails no statement in $\mathcal{P}_-$. Moreover, repairs as defined above have no connection to the input ABox, but the following order relation between repairs takes it into account.

Definition 2.2. Consider two repairs $\mathcal{B} \coloneqq \mathcal{B}_+ \uplus \mathcal{B}_-$ and $\mathcal{C} \coloneqq \mathcal{C}_+ \uplus \mathcal{C}_-$ of $\mathcal{K}$ for $\mathcal{P}$, where $\mathcal{B}_- \coloneqq \{\, \gamma \mid \gamma \in \mathcal{B} \text{ and } \mathcal{K} \models \gamma \,\}$ and $\mathcal{C}_-$ is defined likewise. We write $\mathcal{B} \leq \mathcal{C}$ and say that $\mathcal{C}$ is *at least as good as* $\mathcal{B}$ if $\mathcal{B} \cup \mathcal{O} \models \mathcal{C}_+$ and $\mathcal{C} \cup \mathcal{O} \models \mathcal{B}_-$. Moreover, we write $\mathcal{B} < \mathcal{C}$ and say that $\mathcal{C}$ is *better than* $\mathcal{B}$ if $\mathcal{B} \leq \mathcal{C}$ but $\mathcal{C} \nleq \mathcal{B}$, i.e. either less knowledge is added, $\mathcal{C} \cup \mathcal{O} \not\models \mathcal{B}_+$, or less knowledge is removed, $\mathcal{B} \cup \mathcal{O} \not\models \mathcal{C}_-$. We call $\mathcal{B}$ *optimal* if there is no repair better than $\mathcal{B}$, and $\mathcal{P}$ is *optimally coverable* w.r.t. $\mathcal{K}$ if every repair of $\mathcal{K}$ for $\mathcal{P}$ is at most as good as some optimal one. If $\mathcal{P}$ is optimally coverable w.r.t. $\mathcal{K}$ and there is exactly one optimal repair of $\mathcal{K}$ w.r.t. $\mathcal{P}$ up to equivalence w.r.t. $\mathcal{O}$, then $\mathcal{P}$ is *deterministic* w.r.t. $\mathcal{K}$.

In order to comply with the repair request, optimal repairs preserve as much as possible knowledge entailed by the input KB while containing as little as possible new knowledge — yet not every assertion entailed by $\mathcal{K}$ should be preserved in the concrete application. The reason is that such an entailed assertion might only make sense in the application domain as long as it is substantiated. For instance, if we repair for an assertion stating that Bob has a particular disease, then we would not want to keep the consequence that Bob is ill, unless there is knowledge that he has another disease. In contrast, if it should be repaired that Alice is a celebrity, then we would still want to retain the consequence that Alice is a human. In order to formulate this precisely, we use substantiations. In the literature, justifications of a statement γ have been defined as *subsets* of the refutable part that together with the static part entail γ. In order to eliminate dependence on the syntax, our following definition instead defines substantiations as KBs *entailed* by the refutable part. We further take the provided information in the repair request into account by treating its addition part like the ABox of the input KB, since both are the "positive knowledge" before the repair process.

Definition 2.3. W.r.t. a KB $\mathcal{K}$ and a repair request $\mathcal{P}$, a *substantiation* of an assertion γ is an ABox $\mathcal{J}$ s.t. $\mathcal{A} \cup \mathcal{P}_+ \models \mathcal{J}$ and $\mathcal{J} \cup \mathcal{O} \models \gamma$.

With that, we call a consequence of $\mathcal{K}$ disputable if a repair entails it while another not (i.e. it could be included in a repair or not), but none of its substantiations can be preserved in any repair (i.e. it is not justified anymore).

Definition 2.4. Given a consistent KB $\mathcal{K}$ and a feasible repair request $\mathcal{P}$, a *disputable consequence* of $\mathcal{K}$ w.r.t. $\mathcal{P}$ is an assertion γ s.t.

- $\mathcal{K} \cup \mathcal{P}_+ \models \gamma$,
- there is a repair $\mathcal{B}$ of $\mathcal{K}$ for $\mathcal{P}$ with $\mathcal{B} \cup \mathcal{O} \models \gamma$,
- there is a repair $\mathcal{B}$ of $\mathcal{K}$ for $\mathcal{P}$ with $\mathcal{B} \cup \mathcal{O} \not\models \gamma$, and
- for each repair $\mathcal{B}$ of $\mathcal{K}$ for $\mathcal{P}$, the KB $\mathcal{B} \cup \mathcal{O}$ does not entail any substantiation of γ w.r.t. $\mathcal{K}$ and $\mathcal{P}$.

In order to obtain an optimal repair that makes sense in the application domain, we recommend to decide each disputable consequence by hand and accordingly refine the repair request, i.e. add all accepted ones to $\mathcal{P}_+$ and all rejected ones to $\mathcal{P}_-$. Furthermore, if the repair request is non-deterministic, then it should be further refined to eventually identify an optimal repair appropriate for the application. Formally, we say that a repair request $\mathcal{P}'$ is a *refinement* of $\mathcal{P}$ if $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $\mathcal{P}_- \subseteq \mathcal{P}'_-$, and at least one of these inclusions is strict (i.e. does not hold in the converse direction). Then every repair of $\mathcal{K}$ for $\mathcal{P}'$ is also one for $\mathcal{P}$, but not vice versa. In Section 3 we will develop an interactive method for refining a given repair request to a deterministic one in the DL $\mathcal{EL}$.

The Description Logic $\mathcal{EL}$. We recall the DL $\mathcal{EL}$, on which all other DLs in the $\mathcal{EL}$ family are based. In order to structurally describe the domain of interest, we fix *individual names (INs)*, *concept names (CNs)*, and *role names (RNs)*. *Concept descriptions (CDs)* are built by $C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$ where A ranges over all CNs and r over all RNs. We call $\top$ the *top CD*, $C \sqcap D$ the *conjunction* of C and D, and $\exists r.C$ the *existential requirement* on r with *intent* C. Since nestings of and order in conjunctions are irrelevant, we also use conjunctions $\bigsqcap \Phi$ of finite sets Φ of CDs. CNs and existential requirements are *atoms* and each CD C is a conjunction of atoms, called the *top-level conjuncts* of C and gathered in the set $\mathsf{Conj}(C)$. A *terminological box (TBox)* $\mathcal{T}$ is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$ involving CDs C, D, and an $\mathcal{EL}$ *ontology* is such a TBox without anything else. An *assertion box (ABox)* $\mathcal{A}$ is a finite set of *concept assertions (CAs)* $a : C$ and *role assertions (RAs)* $(a, b) : r$ involving INs a, b, CDs C, and RNs r. A KB consists of an ABox and a TBox.

The semantics of $\mathcal{EL}$ can be defined by translation into the two-variable fragment of first-order logic (FOL): $\tau_x(\top)$ is any tautology with one free variable x, $\tau_x(A) \coloneqq A(x)$, $\tau_x(C \sqcap D) \coloneqq \tau_x(C) \land \tau_x(D)$, $\tau_x(\exists r.C) \coloneqq \exists y.\,(r(x, y) \land \tau_y(C))$ where τ_y is obtained from τ_x by swapping x and y, $\tau(C \sqsubseteq D) \coloneqq \forall x.\,(\tau_x(C) \to \tau_x(D))$, $\tau(a : C) \coloneqq \tau_x(C)[x/a]$, $\tau((a, b) : r) \coloneqq r(x, y)[x/a, y/b]$, $\tau(\mathcal{K}) \coloneqq \bigwedge\{\, \tau(\alpha) \mid \alpha \in \mathcal{K} \,\}$. Thus, $\mathcal{EL}$ inherits the model-theoretic semantics of FOL: we say that α *entails* β and write $\alpha \models \beta$ if $\tau(\alpha) \models \tau(\beta)$, i.e. if every model of $\tau(\alpha)$ is a model of $\tau(\beta)$. Unlike FOL, entailment in $\mathcal{EL}$ is decidable, viz. in polynomial time. We say that a CD C is *subsumed by* a CD D w.r.t. a TBox $\mathcal{T}$, written $C \sqsubseteq^{\mathcal{T}} D$, if $\mathcal{T}$ entails $C \sqsubseteq D$.

Quantified ABoxes. A *quantified ABox (qABox)* $\exists X.\,\mathcal{A}$ consists of a finite set X of *variables* and a finite set $\mathcal{A}$, called *matrix*, that contains assertions $u : A$ and $(u, v) : r$ involving INs or variables u, v, CNs A, and RNs r. Each variable in X and each IN is an *object* of $\exists X.\,\mathcal{A}$. A KB can now also consist of a qABox and a TBox. QABoxes are ABoxes in which variables can be additionally used in place of INs. Since these variables are existentially quantified, they are "anonymous individuals" whose names are not exposed. Moreover, only CNs are allowed within qABoxes, but complex CDs can be represented by the use of variables — e.g. the ABox $\{a : (A \sqcap \exists r.B)\}$ is equivalent to the qABox $\exists\{x\}.\,\{a : A,\ (a, x) : r,\ x : B\}$. Conjunctive queries (CQs), primitive positive formulas (pp-formulas) in FOL, and qABoxes are syntactic variants of each other. The *union* of two qABoxes is $\exists X.\,\mathcal{A} \cup \exists Y.\,\mathcal{B} \coloneqq \exists (X \cup Y).\,(\mathcal{A} \cup \mathcal{B})$ where w.l.o.g. $X \cap Y = \emptyset$ (otherwise variables need to be renamed).

A qABox $\exists X.\,\mathcal{A}$ can be translated into FOL by $\tau(\exists X.\,\mathcal{A}) \coloneqq \exists x_1.\cdots\exists x_n.\,\bigwedge\{\, \tau(\alpha) \mid \alpha \in \mathcal{A} \,\}$ where $X = \{x_1, \ldots, x_n\}$ is an arbitrary enumeration, and a KB $\mathcal{K}$ consisting of $\exists X.\,\mathcal{A}$ and a TBox $\mathcal{T}$ is translated to $\tau(\mathcal{K}) \coloneqq \tau(\exists X.\,\mathcal{A}) \land \tau(\mathcal{T})$. If $\mathcal{K} \models \beta$, then we also say that $\exists X.\,\mathcal{A}$ entails β w.r.t. $\mathcal{T}$ and write $\exists X.\,\mathcal{A} \models^{\mathcal{T}} \beta$.

Entailment between two qABoxes is an NP-complete problem, but whether a qABox entails an ABox can be decided in polynomial time. Without a TBox, $\exists X.\,\mathcal{A} \models \exists Y.\,\mathcal{B}$ iff. there is a *homomorphism* from $\exists Y.\,\mathcal{B}$ to $\exists X.\,\mathcal{A}$, which is a function h that sends each IN a to itself and each variable in Y to an object of $\exists X.\,\mathcal{A}$ such that applying h within any assertion in $\mathcal{B}$ yields an assertion in $\mathcal{A}$. With a TBox $\mathcal{T}$, entailment can be decided by first *saturating* $\exists X.\,\mathcal{A}$ by means of $\mathcal{T}$ (i.e. compute the chase or the universal model) and then checking for a homomorphism from $\exists Y.\,\mathcal{B}$ to the saturation.

For some applications model-based entailment is too strong and it suffices to compare qABoxes based on their consequences from a query language. In DL, important query languages are IQ and IRQ. The former consists of all CAs (sometimes also called *instance queries*, IQs), and the latter of all CAs and RAs. As further query languages, CQ consists of all qABoxes, and gloIRQ extends IRQ by all global IQs $\exists\{x\}.\,\{C(x)\}$ where C is a CD. Given a query language QL, we say that $\exists X.\,\mathcal{A}$ QL-*entails* $\exists Y.\,\mathcal{B}$ w.r.t. $\mathcal{T}$ and write $\exists X.\,\mathcal{A} \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\,\mathcal{B}$ if $\exists Y.\,\mathcal{B} \models^{\mathcal{T}} \gamma$ only if $\exists X.\,\mathcal{A} \models^{\mathcal{T}} \gamma$ for each query $\gamma \in \mathsf{QL}$. Since the TBox is fixed, CQ-entailment coincides with model-based entailment. In contrast, IQ- and IRQ-entailment are decidable in polynomial time. Without a TBox, $\exists X.\,\mathcal{A} \models_{\mathsf{IQ}} \exists Y.\,\mathcal{B}$ iff. there is a *simulation* from $\exists Y.\,\mathcal{B}$ to $\exists X.\,\mathcal{A}$, which is a "non-functional homomorphism" that can relate each object of $\exists Y.\,\mathcal{B}$ to multiple objects of $\exists X.\,\mathcal{A}$. With a TBox $\mathcal{T}$, we check for a simulation from $\exists Y.\,\mathcal{B}$ to the IQ-saturation of $\exists X.\,\mathcal{A}$ w.r.t. $\mathcal{T}$, which is obtained by materializing all CAs implied by the TBox: while there is an object u and a CI $C \sqsubseteq D$ such that the matrix of the current qABox entails $u : C$ but not $u : D$, we extend the current qABox with $u : D$ but represented by the use of variables if D is complex. Unlike the above saturations, we can here reuse variables that have already been introduced for the same CD (w.l.o.g. denoted as x_E where E is the subconcept to be represented) and thus the IQ-saturation can be computed in polynomial time. Alternatively, the canonical model of the input KB computed by an IQ-complete calculus (such as the completion procedure [1, 18]) can be treated as a qABox to obtain the IQ-saturation, though might be larger than necessary. Furthermore, $\exists X.\,\mathcal{A} \models^{\mathcal{T}}_{\mathsf{IRQ}} \exists Y.\,\mathcal{B}$ iff. $\exists X.\,\mathcal{A} \models^{\mathcal{T}}_{\mathsf{IQ}} \exists Y.\,\mathcal{B}$ and each RA in $\mathcal{B}$ involving only INs is also contained in $\mathcal{A}$.

Optimal Repairs in $\mathcal{EL}$. There is no general approach to computing optimal repairs, and we will now recall results obtained so far. Previous research focused on repair requests without an addition part. It seems that abduction methods [15, 19, 20] could be used to treat the addition part but it is still unclear how optimality could be achieved, and thus we leave this for future research. Instead, we just assume that the addition part $\mathcal{P}_+$ is already entailed by the input KB (which could be achieved by simply adding all statements in $\mathcal{P}_+$ to the KB), and so $\mathcal{P}_+$ is only to be preserved by every repair.

Optimal repairs need not exist in every setting [13]. $\mathcal{EL}$ TBoxes can be optimally repaired when the left-hand sides of CIs are fixed [22, 23]. Moreover, we can compute optimal repairs of KBs consisting of a refutable qABox and a static TBox [4, 5, 7, 8, 10, 11, 12]. In the following, we will reformulate definitions regarding optimal repairs for some of these settings and recall main results.

Assume that the input KB $\mathcal{K}$ consists of a refutable qABox $\exists X.\mathcal{A}$ and a static $\mathcal{EL}$ TBox $\mathcal{T}$, and further let QL be the query language used to access this KB. Since such KBs are always consistent, we do not need to explicitly require consistency. Now a *repair request* is a finite subset $\mathcal{P}$ of QL and, as explained above, such that the given KB $\mathcal{K}$ entails its addition part $\mathcal{P}_+$. Thus the optimal repairs are qABoxes entailed by $\mathcal{K}$ since no new knowledge must be added to make $\mathcal{P}_+$ entailed. In this sense, a QL-*repair* of $\mathcal{K}$ for $\mathcal{P}$ (also called a QL-*repair* of $\exists X.\mathcal{A}$ for $\mathcal{P}$ w.r.t. $\mathcal{T}$) is a qABox $\exists Y.\mathcal{B}$ such that

(R1) $\exists X.\mathcal{A} \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$,

(R2) $\exists Y.\mathcal{B} \models^{\mathcal{T}} \alpha$ for each $+\alpha \in \mathcal{P}$, and

(R3) $\exists Y.\mathcal{B} \not\models^{\mathcal{T}} \beta$ for each $-\beta \in \mathcal{P}$.

$\exists Y.\mathcal{B}$ is *optimal* if there is no QL-repair $\exists Z.C$ that strictly QL-entails $\exists Y.\mathcal{B}$ (i.e. $\exists Z.C \models^{\mathcal{T}}_{\mathsf{QL}} \exists Y.\mathcal{B}$ but $\exists Y.\mathcal{B} \not\models^{\mathcal{T}}_{\mathsf{QL}} \exists Z.C$).

The set of optimal QL-repairs can effectively be computed and every QL-repair is QL-entailed by an optimal one (i.e. each repair request is optimally coverable) in the following situations: in $\mathcal{EL}$ with QL = IQ and arbitrary TBoxes and with QL = CQ but $\mathcal{P} \subseteq$ IQ and cycle-restricted TBoxes [5, 12], in $\mathcal{EL}$ with QL = CQ but $\mathcal{P} \subseteq$ IQ and arbitrary TBoxes but infinite repairs [4], in Horn-$\mathcal{ALCROI}$ with QL = CQ but $\mathcal{P} \subseteq$ gloIRQ and terminating TBoxes [8], in $\mathcal{EL}$ with QL = IRQ and arbitrary TBoxes [7, 11], and in $\mathcal{EL}^{\bot}$ with QL = gloIRQ and arbitrary TBoxes [10]. In all aforementioned cases, a construction of canonical repairs is provided such that every optimal repair is entailed by a canonical one. We will recall the construction for $\mathcal{EL}$ and QL = IRQ, which is especially interesting since between $\mathcal{EL}$ ABoxes model-based entailment coincides with IRQ-entailment [7] and qABoxes can represent $\mathcal{EL}$ ABoxes.

Let $\mathcal{P} \subseteq$ IRQ be a repair request. We use nominals to express RAs $(a,b):r$ and replace them with CAs $a:\exists r.\{b\}$ [8]. Keep in mind that each of these CAs contributes the atoms $\exists r.\{b\}$ and $\{b\}$. Since entailment between qABoxes is characterized by a rewrite system that can copy objects into fresh variables and delete assertions [4], we can construct repairs by copying and deleting too. Moreover, since optimal repairs retain as many consequences as possible, we construct them from saturations. The canonical repairs have a closed-form representation involving copies of the form $\langle\!\langle u, \Phi \rangle\!\rangle$ where u is an object in the saturation and Φ is a repair type, which specifies what must be deleted for this copy.

More specifically, let $\exists X^{\mathcal{T}}.\mathcal{A}^{\mathcal{T}}$ be the IQ-saturation of the input KB. Each *repair type* Φ for u consists of atoms occurring in $\mathcal{P}$ or $\mathcal{T}$ and must satisfy the following three conditions:

(RT1) $\mathcal{A}^{\mathcal{T}} \models u : C$ for each atom C in Φ.

(RT2) $C \not\sqsubseteq^{\emptyset} D$ for each two atoms C, D in Φ.

(RT3) For each atom C in Φ and each CI $E \sqsubseteq F$ in $\mathcal{T}$ with $\mathcal{A}^{\mathcal{T}} \models u : E$ and $F \sqsubseteq^{\mathcal{T}} C$, there is an atom D in Φ with $E \sqsubseteq^{\emptyset} D$.

In order to ensure that each copy $\langle\!\langle u, \Phi \rangle\!\rangle$ is no instance of any atom in Φ, the matrix of each canonical IRQ-repair consists of the following assertions:

(CR1) $\langle\!\langle u, \Phi \rangle\!\rangle : A$ if $u : A \in \mathcal{A}^{\mathcal{T}}$ and $A \notin \Phi$, and

(CR2) $(\langle\!\langle u, \Phi \rangle\!\rangle, \langle\!\langle v, \Psi \rangle\!\rangle) : r$ if $(u,v) : r \in \mathcal{A}^{\mathcal{T}}$ and, for each $\exists r.C \in \Phi$ with $\mathcal{A}^{\mathcal{T}} \models v : C$, there is an atom $D \in \Psi$ with $C \sqsubseteq^{\emptyset} D$.

We finally need to select which of the copies $\langle\!\langle a, \Phi \rangle\!\rangle$ is used as the IN a. For this purpose, a *repair seed* $\mathcal{S}$ maps each IN a to a repair type $\mathcal{S}_a$ for a such that:

(RS1) For each $+\,a : C$ in $\mathcal{P}$, there is no atom $D \in \mathcal{S}_a$ with $C \sqsubseteq^{\mathcal{T}} D$.

(RS2) For each $+\,(a,b) : r$ in $\mathcal{P}$ and for each $\exists r.C \in \mathcal{S}_a$ with $\exists X.\mathcal{A} \models^{\mathcal{T}} b : C$, there is an atom $D \in \mathcal{S}_b$ with $C \sqsubseteq^{\emptyset} D$.

(RS3) For each $-\,a : C$ in $\mathcal{P}$ with $\exists X.\mathcal{A} \models^{\mathcal{T}} a : C$, there is an atom $D \in \mathcal{S}_a$ with $C \sqsubseteq^{\emptyset} D$.

(RS4) For each $-\,(a,b) : r$ in $\mathcal{P}$ with $\exists X.\mathcal{A} \models^{\mathcal{T}} (a,b) : r$, we have $\exists r.\{b\} \in \mathcal{S}_a$.

(RS5) $\{a\} \notin \mathcal{S}_a$

(RS1) is new since previous work on optimal repairs in $\mathcal{EL}$ has assumed $\mathcal{P}_+ = \emptyset$. Support can be added either through the static part of the input KB [8] or by means of Lemma 3 in [9], from which (RS1) is derived. Also Condition (RS2) is new and together with Instruction (CR2) ensures that every $+(a,b):r$ is entailed, as observed in Lemma 4.5 in [11]. Moreover, the interplay of Conditions (RS4) and (RS5) and Instruction (CR2) makes sure that no $-\,(a,b) : r$ is entailed. Only (RS3) is from the original definition and guarantees together with Instructions (CR1) and (CR2) that no $-\,a : C$ is entailed [4, 5]. In the end, the *canonical* IRQ-*repair* induced by $\mathcal{S}$ is denoted as $\mathsf{rep}^{\mathcal{T}}_{\mathsf{IRQ}}(\exists X.\mathcal{A}, \mathcal{S})$ and its variable set consists of all copies $\langle\!\langle u, \Phi \rangle\!\rangle$ except those of the form $\langle\!\langle a, \mathcal{S}_a \rangle\!\rangle$, which rather are synonyms of the INs a. This repair is saturated, i.e. it entails a query $\gamma \in$ IRQ w.r.t. $\mathcal{T}$ iff. it entails γ w.r.t. the empty TBox.

Evidently, canonical repairs are computable in polynomial time w.r.t. data complexity, i.e. computation time is dominated by $\mathcal{T}$ and $\mathcal{P}$. In practice we should not compute the whole exponentially large canonical repair but only an equivalent sub-qABox, called *optimized repair* [5, 8]. Experiments have shown that such optimized repairs of real-world KBs can indeed be computed in practice [5]. With the revised definition of repair types used here a further speed-up is expected, but this would still need to be verified empirically.

Not every canonical repair is optimal, but every optimal repair is equivalent to a canonical one. Thus in order to get all optimal repairs in exponential time, we can enumerate all repair seeds, compute the induced repairs, and then filter out the non-optimal ones. Alternatively to filtering the repairs, we could also filter the repair seeds since it is decidable in polynomial time whether a repair seed induces an optimal repair [9, 11].

3 The Smart Interaction Strategy

When a KB consisting of a qABox $\exists X.\mathcal{A}$ and a static $\mathcal{EL}$ TBox $\mathcal{T}$ should be repaired for a feasible repair request $\mathcal{P}$, it is not useful to compute all optimal repairs by enumerating all repair seeds and then let the experts choose among them. With that approach the workload of the experts would be too high since in the worst case there are exponentially many optimal repairs and each of them might be of exponential size (even their optimized variant).

From a practical perspective, not every optimal repair makes sense in the domain of interest. This is because the unwanted consequences in $\mathcal{P}_-$ as well as their logical causes w.r.t. $\mathcal{T}$ may contain conjunctions, and in order to prevent entailment of a conjunction it suffices to choose one of its conjuncts and make it not entailed. The construction of an optimal repair is thus non-deterministic and there are multiple repair seeds in general. Instead of making random choices, we recommend to interact with the experts to refine the given repair request $\mathcal{P}$ to a deterministic one and so identify a useful optimal repair. With our interactive approach the experts need to answer at most polynomially many questions, i.e. their workload is significantly lower.

This section presents the *smart interaction strategy*, which runs in two phases. By means of backward chaining, Phase 1 computes causes of identified errors and interacts with the experts when choices must be made. Phase 1 terminates with a deterministic refinement of the initially provided repair request $\mathcal{P}$. Already with Phase 1 every optimal repair can be reached. Phase 2 is concerned with disputable consequences: they first need to be decided by the experts, and then Phase 2 proceeds with the further refined repair request like Phase 1. In the end, a repair that is optimal w.r.t. the given repair request $\mathcal{P}$ and all experts answers has been identified.

3.1 Fundamentals of the Strategy and Phase 1

In the following, let $\mathcal{P}^0 \coloneqq \mathcal{P}$ be the initial repair request. The strategy maintains three sets. Undecided queries are held by the set Q, which is initially empty, and all queries currently in Q are displayed to the experts for the purpose of decision making, e.g. in form of a list with action buttons for each entry (accept and reject). Queries accepted by the experts are added to $\mathcal{P}_+$, whereas rejected queries are added to $\mathcal{P}_-$. Initially all queries not entailed by the input KB are removed from $\mathcal{P}_-$ since for these no repair is necessary. The strategy evolves as follows, always ensuring that $\mathcal{P}_+ \uplus \mathcal{P}_-$ refines $\mathcal{P}^0$ and is feasible (i.e. $\mathcal{P}_+ \not\models^\mathcal{T} \beta$ for each $\beta \in \mathcal{P}_-$).

(SIS1) Whenever a non-atomic query α has been added to $\mathcal{P}_-$, it needs to be inspected to find out why it should not hold.
- If α has the form $a : C$ where C is a conjunction, then for each top-level conjunct $D \in \mathsf{Conj}(C)$, the query $a : D$ is added to Q.
- If α has the form $a : \exists r. C$, then for each IN b where $\exists X. \mathcal{A}$ entails $(a, b) : r$ and $b : C$ w.r.t. $\mathcal{T}$, the two queries $(a, b) : r$ and $b : C$ are added to Q.

Nothing needs to be done for atomic queries, i.e. CAs $a : A$ where A is a CN or RAs $(a, b) : r$, since these can be directly deleted from the qABox.

(SIS2) Furthermore, we need to ensure that inference with the TBox cannot restore a rejected query: for each rejected query α added to $\mathcal{P}_-$, all implicant queries β with $\exists X. \mathcal{A} \models^\mathcal{T} \beta$ and $\beta \models^\mathcal{T} \alpha$ must be rejected too.
- RAs have no implicants w.r.t. an $\mathcal{EL}$ TBox and thus need no further treatment.
- For CAs, it suffices to restrict attention to the implicants as in Condition (RT3): when a CA $a : C$ has been added to $\mathcal{P}_-$, we add to $\mathcal{P}_-$ all CAs $a : D$ where $D \sqsubseteq E$ is a CI in $\mathcal{T}$ with $\exists X. \mathcal{A} \models^\mathcal{T} a : D$ and $E \sqsubseteq^\mathcal{T} C$.

(SIS3) Last, inherited answers are computed after every answer received from the experts. To this end, each currently undecided query α in Q is checked.
- If $\mathcal{P}_+$ entails α w.r.t. $\mathcal{T}$, then α inherits acceptance and is moved from Q to $\mathcal{P}_+$.
- If $\mathcal{P}_+ \cup \{\alpha\}$ entails w.r.t. $\mathcal{T}$ any assertion in $\mathcal{P}_-$, then α inherits rejection and is moved from Q to $\mathcal{P}_-$.

These instructions enable control of the delete operation. Phase 1 ends as soon as no undecided queries are contained in Q anymore. As all queries are built from the polynomially many sub-CDs of the input and $\mathcal{EL}$ allows for polynomial-time reasoning, Phase 1 terminates in polynomial time.

3.2 Induced Repairs after Phase 1

We denote by $\mathcal{P}^1_+$ and $\mathcal{P}^1_-$ the sets of accepted and, respectively, rejected queries at the end of Phase 1. Then $\mathcal{P}^1 \coloneqq \mathcal{P}^1_+ \uplus \mathcal{P}^1_-$ is a repair request that refines the initial repair request $\mathcal{P}^0$. Furthermore, we define the mapping $\mathcal{S}^1$ that sends each IN a to the set

$$\mathcal{S}^1_a \coloneqq \mathsf{Max}\{\, C \mid a : C \in \mathcal{P}^1_- \,\} \cup \{\, \exists r. \{b\} \mid (a, b) : r \in \mathcal{P}^1_- \,\},$$

where the operator Max selects the CDs that are maximal w.r.t. subsumption $\sqsubseteq^\emptyset$. We first show that this definition yields a repair seed for $\mathcal{P}^1$. It follows that its induced repair entails each accepted query but no rejected query, i.e. the identified repair actually reflects all decisions made by the experts. In particular, it is a repair for $\mathcal{P}^0$ since $\mathcal{P}^1$ refines $\mathcal{P}^0$.

Lemma 3.1. *$\mathcal{S}^1$ is a repair seed for $\mathcal{P}^1$ and thus also for $\mathcal{P}^0$.*

Next, we show that the strategy is fine-grained enough in the sense that a unique repair is identified.

Proposition 3.2. *$\mathcal{P}^1$ is a deterministic repair request, for which the only optimal* IRQ*-repair is the one induced by $\mathcal{S}^1$.*

As first main result, we verify that every optimal repair can really be reached with the strategy.

Theorem 3.3. *For every optimal* IRQ*-repair for $\mathcal{P}^0$, an inducing repair seed can be identified with Phase 1.*

3.3 Control of the Copy Operation

So far, Phase 1 only allows to control the delete operation, i.e. the repair might contain undesired copies. On the one hand, such a situation can only occur when the input is insufficiently specified: the repair request does not preclude these copies and thus an optimal repair will contain them. On the other hand, these copies might not be immediately problematic since they could only later be revealed when the repair is queried — then one can simply repair again.

In order to also control the copy operation and thus the creation of copies, we can employ the following additional instruction.

(SIS4) When a query $a : \exists r_1. \cdots \exists r_n. C$ has been added to $\mathcal{P}_-$, the experts needs to specify which copies of objects linked to a by an $r_1 \cdots r_n$-chain exist.
- If C is a conjunction, then the query $a : \exists r_1. \cdots \exists r_n. C \setminus D$ is added to Q for each top-level conjunct $D \in \mathsf{Conj}(C)$, where $C \setminus D \coloneqq \bigsqcap \mathsf{Conj}(C) \setminus \{D\}$.
- The query $a : \exists r_1. \cdots \exists r_n. E$ is added to Q for each CI $E \sqsubseteq F \in \mathcal{T}$ with $\exists X. \mathcal{A} \models^\mathcal{T} a : \exists r_1. \cdots \exists r_n. E$ and $F \sqsubseteq^\mathcal{T} C$.

The number of queries to be decided by the experts is polynomial with Instructions (SIS1), (SIS2), and (SIS3), but can be worst-case exponential when the additional Instruction (SIS4) is employed.

Another alternative would be to wait until Phase 1 without the above additional instruction has finished, and then determine superfluous copies as follows. For each object u reachable from an IN a by a role path $r_1, \ldots, r_n$ in the input KB, determine all copies $\langle\!\langle u, \Phi_1 \rangle\!\rangle, \ldots, \langle\!\langle u, \Phi_n \rangle\!\rangle$ in the repair induced by $\mathcal{S}^1$ that are still reachable from a by the role path $r_1, \ldots, r_n$. Then compute characteristic CDs $C_1, \ldots, C_n$ such that C_i only has copy $\langle\!\langle u, \Phi_i \rangle\!\rangle$ as instance and present the queries $a : \exists r_1. \cdots \exists r_n. C_i$ to the experts. If such a query is rejected, then the corresponding copy $\langle\langle u, \Phi_i \rangle\rangle$ is deleted. With that, still all causes need to be considered by means of Instruction (SIS2), viz. to avoid re-introduction of the unwanted copies through the TBox. Since there might be exponentially many copies in the worst case, this alternative approach could also need exponentially many additional queries (at least one for each reachable copy).

3.4 Disputable Consequences and Phase 2

We now consider the disputable consequences of the input KB $\mathcal{T} \uplus \exists X. \mathcal{A}$ w.r.t. the refined repair request $\mathcal{P}^1$ obtained from Phase 1, which we also call *disputable consequence at the end of Phase 1*. In this specific setting, Definitions 2.3 and 2.4 read as follows.

Definition 3.4. A *substantiation* of a query $\gamma \in \mathsf{IRQ}$ is a qABox $\exists Y. \mathcal{B}$ with $\exists X. \mathcal{A} \cup \mathcal{P}^1_+ \models \exists Y. \mathcal{B}$ and $\exists Y. \mathcal{B} \models^{\mathcal{T}} \gamma$.

Corollary 3.5. *A query $\gamma \in \mathsf{IRQ}$ is a disputable consequence at the end of Phase 1 iff. it fulfills all of the following conditions:*

(DC1) *γ is entailed by $\exists X. \mathcal{A}$ w.r.t. $\mathcal{T}$.*
(DC2) *γ has not been decided by the experts or by inheritance, i.e.*
- *$\mathcal{P}^1_+ \cup \{\gamma\}$ does not entail w.r.t. $\mathcal{T}$ any query in $\mathcal{P}^1_-$, and*
- *$\mathcal{P}^1_+$ does not entail γ w.r.t. $\mathcal{T}$.*

(DC3) *For every substantiation $\mathcal{J}$ of γ, we have that $\mathcal{J}$ does not entail w.r.t. $\mathcal{T}$ all accepted queries in $\mathcal{P}^1_+$, or $\mathcal{J}$ entails w.r.t. $\mathcal{T}$ some rejected query in $\mathcal{P}^1_-$.*

Recall from Lemma 3.1 that the repair induced by the repair seed identified in Phase 1 does not entail any rejected query in $\mathcal{P}^1_-$. No substantiations for a disputable consequence γ can thus be retained in the repair, i.e. all bare causes for γ in the data must be removed. Since validity of γ cannot be decided by the information collected from the experts in Phase 1, and keeping γ in the repair might not be reasonable, such disputable consequences γ are rather shown to the experts for examination, viz. right in the beginning of Phase 2.

Phase 2 controls the saturation and runs as follows. First we compute all disputable consequences. Since we consider the query language IRQ, we restrict attention to disputable RAs and disputable CAs involving sub-CDs of the input, i.e. of the form $a : C$ where C occurs in $\mathcal{P}^0$ or $\mathcal{T}$. All these queries are then presented to the experts by adding them to $\mathcal{Q}$. Afterwards, the strategy proceeds as in Phase 1. Like the first phase, Phase 2 ends when $\mathcal{Q}$ contains no undecided query anymore. Then all disputable consequences have been processed, and the final repair is computed from the so identified repair seed $\mathcal{S}^2$. Lemma 3.1 and Proposition 3.2 hold analogously.

4 Computing Disputable Consequences

Next, we will develop a practical method for computing the disputable consequences at the end of Phase 1. Conditions (DC1) and (DC2) in Corollary 3.5 can be checked in polynomial time, but it is not obvious how Condition (DC3) can be checked in an efficient way. Naïvely following the very definition will not work in practice since we would need to go through all queries in IRQ and all substantiations. In order to find a more efficient approach, we transform Condition (DC3) into the following equivalent condition:

- For every qABox $\exists Y. \mathcal{B}$, if $\exists X. \mathcal{A} \cup \mathcal{P}^1_+ \models \exists Y. \mathcal{B}$, and $\exists Y. \mathcal{B} \models^{\mathcal{T}} \alpha$ for every $\alpha \in \mathcal{P}^1_+$, and $\exists Y. \mathcal{B} \not\models^{\mathcal{T}} \beta$ for each $\beta \in \mathcal{P}^1_-$, then $\exists Y. \mathcal{B} \not\models^{\mathcal{T}} \gamma$. (see proof of Proposition 4.8)

Now the qABoxes $\exists Y. \mathcal{B}$ almost satisfy the definition of a repair of $\exists X. \mathcal{A}$ for $\mathcal{P}^1$, the only exceptions are that the entailment of $\exists Y. \mathcal{B}$ by the input KB does not involve the TBox $\mathcal{T}$ but additionally $\mathcal{P}^1_+$. Such repairs are obtained directly from $\exists X. \mathcal{A}$ and not from its saturation, and are thus called "unsaturated."

Definition 4.1. An *unsaturated repair* of a qABox $\exists X. \mathcal{A}$ for a repair request $\mathcal{P}$ w.r.t. a TBox $\mathcal{T}$ is a qABox $\exists Y. \mathcal{B}$ such that

(UR1) $\exists X. \mathcal{A} \cup \mathcal{P}_+ \models \exists Y. \mathcal{B}$,
(UR2) $\exists Y. \mathcal{B} \models^{\mathcal{T}} \alpha$ for each $\alpha \in \mathcal{P}_+$, and
(UR3) $\exists Y. \mathcal{B} \not\models^{\mathcal{T}} \beta$ for each $\beta \in \mathcal{P}_-$.

Moreover, we call $\exists Y. \mathcal{B}$ QL-*optimal* if it is not strictly QL-entailed by another unsaturated repair.

With this definition in place, Condition (DC3) can be rewritten to:

- no unsat. repair of $\exists X. \mathcal{A}$ for $\mathcal{P}^1$ w.r.t. $\mathcal{T}$ entails γ w.r.t. $\mathcal{T}$

4.1 Unsaturated Repairs

Unsaturated repairs can be constructed from the input qABox and not from its saturation, i.e. they are independent from the employed query language. Only optimality must refer to a particular entailment relation. In order to define a canonical form of unsaturated repairs, we keep the notion of repair types as it is and only include copies of objects of the input qABox, whereas objects and assertions produced by the saturation are ignored.

Since we also need to take the addition part $\mathcal{P}_+$ into account, we first define the qABox $\exists X'. \mathcal{A}'$ as the union of $\exists X. \mathcal{A}$ and a qABox equivalent to $\mathcal{P}_+$.

Lemma 4.2. *There is a homomorphism h from $\exists X'. \mathcal{A}'$ to the* IQ-*saturation $\exists X^{\mathcal{T}}. \mathcal{A}^{\mathcal{T}}$ such that $h(u) = u$ for each object u of $\exists X. \mathcal{A}$.*

Definition 4.3. The *canonical unsaturated repair* induced by a repair seed $\mathcal{S}$ is the qABox $\mathsf{rep}^{\mathcal{T}}_{\mathsf{unsat}}(\exists X. \mathcal{A}, \mathcal{S})$ of which the objects are all pairs $\langle\langle u, \Phi \rangle\rangle$ with an object u of $\exists X'. \mathcal{A}'$ and a repair type Φ for $h(u)$, and the matrix consists of the following assertions:

(CUR1) $\langle\langle u, \Phi \rangle\rangle : A$ if $u : A \in \mathcal{A}'$ and $A \notin \Phi$, and
(CUR2) $(\langle\langle u, \Phi \rangle\rangle, \langle\langle v, \Psi \rangle\rangle) : r$ if $(u, v) : r \in \mathcal{A}'$ and, for each $\exists r. C \in \Phi$ with $\mathcal{A}^{\mathcal{T}} \models h(v) : C$, there is an atom $D \in \Psi$ with $C \sqsubseteq^{\emptyset} D$.

Moreover, an object $\langle\langle u, \Phi \rangle\rangle$ is a variable if u is a variable (i.e. $u \in X$) or $\Phi \neq \mathcal{S}_u$, and each IN a and $\langle\langle a, \mathcal{S}_a \rangle\rangle$ are treated as synonyms.

Proposition 4.4. *$\mathsf{rep}^{\mathcal{T}}_{\mathsf{unsat}}(\exists X. \mathcal{A}, \mathcal{S})$ is an unsaturated repair of $\exists X. \mathcal{A}$ for $\mathcal{P}$ w.r.t. $\mathcal{T}$.*

The canonical unsaturated repairs are complete in the sense that they cover all unsaturated repairs, similarly to the saturated repairs.

Proposition 4.5. *Each unsat. repair is entailed by a canonical one.*

As next step, we determine the computational complexity of query answering w.r.t. an unsaturated repair.

Theorem 4.6. *W.r.t. the size of the input KB and repair request only, deciding entailment of queries in* IRQ *by canonical unsaturated repairs is in* P, *but is* NP-*complete if the TBox is taken into account in the entailment. Hardness already holds without an addition part and even if the TBox is cycle-restricted.*

Proof. See [21] for the first statement. Regarding the second statement, we show that satisfiability of a 3-CNF can be reduced to entailment of a CA by an unsaturated repair and a TBox in polynomial time. NP-hardness is then inherited from the former decision problem.

Let α be a propositional formula in 3-CNF, i.e. α is a conjunction of disjunctions over clauses that consist of three literals each, say $\alpha = \beta_1 \land \cdots \land \beta_n$ where $\beta_i = \lambda_{i,1} \lor \lambda_{i,2} \lor \lambda_{i,3}$ for each index i and each $\lambda_{i,j}$ is either a propositional variable or a negation of a propositional variable. We define the TBox $\mathcal{T}_\alpha$ consisting of the following CIs:

- $T_{\beta_1} \sqcap \cdots \sqcap T_{\beta_n} \sqsubseteq T_\alpha$
- $T_{\lambda_{i,j}} \sqsubseteq T_{\beta_i}$ for all i, j
- $F_p \sqsubseteq T_{\neg p}$ for each propositional variable p occurring in α
- $F_p \sqcap T_p \sqsubseteq E$ for each propositional variable p occurring in α.

We further take the qABox $\exists\{x\}.\{\,(a,x) : r,\ x : T_p,\ x : F_p \mid p \in \mathsf{Var}(\alpha)\,\}$, the repair request $\{a{:}\exists r.E\}$ without addition part, and the query $a{:}\exists r.T_\alpha$. For this input, there is only the repair seed $\mathcal{S}$ where $\mathcal{S}_a \coloneqq \{\exists r.E\}$. We will show that the induced unsaturated repair entails the query w.r.t. the TBox iff. the formula α is satisfiable.

First let α be satisfiable, certified by the variable assignment $g\colon \mathsf{Var}(\alpha) \to \{T, F\}$ under which α evaluates to T. By flipping the values we obtain a repair type for x, namely $\{\,T_p \mid g(p) = F\,\} \cup \{\,F_p \mid g(p) = T\,\} \cup \{E\}$. In the unsaturated repair the copy of x annotated with this repair type is an instance of T_p iff. $g(p) = T$ and of F_p iff. $g(p) = F$, and this copy is an r-successor of a since the repair type contains E. Inference with the TBox yields that this copy is an instance of T_α, and thus the query is entailed.

It remains to prove the only-if direction. To this end, assume that the unsaturated repair induced by $\mathcal{S}$ entails $a : \exists r.T_\alpha$ w.r.t. $\mathcal{T}$ and is denoted by $\exists Y.\mathcal{B}$. Since the TBox does not introduce new r-successors, the matrix $\mathcal{B}$ must contain a RA $(a, y) : r$ for which y is an instance of T_α w.r.t. $\mathcal{T}$. The repair type that annotates y must contain E since $\mathcal{S}_a$ contains $\exists r.E$, and thus this repair type must contain F_p or T_p for each variable $p \in \mathsf{Var}(\alpha)$. It follows that $\mathcal{B}$ does not contain both $y{:}F_p$ and $y{:}T_p$ for any variable p. We define a variable assignment $g\colon \mathsf{Var}(\alpha) \to \{F, T\}$ by $g(p) \coloneqq F$ if $\mathcal{B}$ contains $y : F_p$, and otherwise by $g(p) \coloneqq T$. We show that α evaluates to T under g, which shows that α is satisfiable.

Recall that α is a conjunction of clauses $\beta_i = \lambda_{i,1} \lor \lambda_{i,2} \lor \lambda_{i,3}$. By the very definition of the TBox $\mathcal{T}_\alpha$, and since $\mathcal{B} \models^{\mathcal{T}} y : T_\alpha$, at least one of the assertions $y{:}T_{\lambda_{i,j}}$ with $j \in \{1, 2, 3\}$ must be entailed by $\mathcal{B}$ for each clause β_i.

- If the literal $\lambda_{i,j}$ is a variable p, then $\mathcal{B} \models^{\mathcal{T}} y : T_p$ requires that the matrix $\mathcal{B}$ already contains this assertion $y{:}T_p$. Since $\mathcal{B} \not\models^{\mathcal{T}} y{:}E$, it cannot contain the assertion $y{:}F_p$ as well, and thus $g(p) = T$.
- Otherwise, the literal $\lambda_{i,j}$ is a negated variable $\neg p$. Since the matrix $\mathcal{B}$ can only contain CAs involving the CNs F_q and T_q for variables $q \in \mathsf{Var}(\alpha)$, the consequence $y : T_{\neg p}$ can only be produced by means of the CI $F_p \sqsubseteq T_{\neg p}$ and therefore the matrix $\mathcal{B}$ must contain the assertion $y : F_p$. It follows that $g(p) = F$, and so $\neg p$ evaluates to T under g.

It now easily follows that each clause β_i evaluates to T under g, and thus the whole formula α as well.

Last, we show that IRQ-query answering is in NP. According to Theorem 2 in [6], the given query is entailed iff. there is a homomorphism from the query (seen as qABox) to the saturation of the unsaturated repair. Since the unsaturated repair has exponential size in the worst case, it should not be fully computed. Instead, such a homomorphism has polynomial size since it sends the polynomially many objects in the qABox representation of the query to the objects in the saturation, which are either of the form $\langle\!\langle u, \Phi \rangle\!\rangle$ (in the unsaturated repair) or of the form x_C for a CD C that occurs in the TBox $\mathcal{T}$ (added by saturation), and thus these objects need polynomial size only. It thus suffices to guess a polynomial-size mapping from the objects in the qABox representation of the query to such objects, and then check whether it is a homomorphism, which would certify the entailment. For the latter, note that to check if an assertion $(\langle\!\langle u, \Phi \rangle\!\rangle, x_C) : r$ is present boils down to checking if the saturation rule can be applied at $\langle\!\langle u, \Phi \rangle\!\rangle$ in order to generate this successor x_C. To this end, it is enough to guess polynomially many assertions in the neighborhood of $\langle\!\langle u, \Phi \rangle\!\rangle$ since the TBox consists of polynomially many CIs, each of which has a polynomial-size premise. Checking for presence of the other assertions can be done in polynomial time in the obvious way. ◻

4.2 Computing Disputable Consequences from Unsaturated Repairs

As last step we show how the disputable consequences can be efficiently computed with one particular unsaturated repair.

Lemma 4.7. *The canonical unsaturated repair induced by $\mathcal{S}^1$ is the only* IRQ*-optimal unsaturated repair for $\mathcal{P}^1$.*

The above lemma now allows us to formulate a more efficient variant of Condition (DC3).

Proposition 4.8. *For each query $\gamma \in$ IRQ, Condition (DC3) in Corollary 3.5 is equivalent to the following:*

(DC3*) *γ is not entailed w.r.t. $\mathcal{T}$ by the unsat. repair induced by $\mathcal{S}^1$.*

This Condition (DC3*) is much easier to check than (DC3) as we only need to compute (the optimized variant of) a single unsaturated repair and then determine which queries it entails. Unfortunately, this cannot be done by only looking at the repair seed as for saturated repairs, and we will prove that deciding disputable consequences is co NP-complete. Even though such a complexity result seems to prohibit a practical use, our implementation works surprisingly fast even with large TBoxes such as (the $\mathcal{EL}$ fragment of) SNOMED CT. Since saturated repairs can be computed in practice [5], the same holds for unsaturated repairs as they are constructed similarly but directly from the input qABox. Therefore disputable consequences can be computed in practice as well.

Theorem 4.9. *Deciding disputable consequences at the end of Phase 1 is* co NP*-complete.*

Proof. We use the same input as in the proof of Theorem 4.6, without any modifications. No queries need to be decided by the experts in Phase 1 and we obtain $\mathcal{P}^1_+ = \emptyset$ and $\mathcal{P}^1_- = \{a : \exists r.E\}$, which is the given repair request. Further recall that there is only one repair seed, namely with $\mathcal{S}_a = \{\exists r.E\}$, and that the formula α is satisfiable iff. the induced unsat. repair entails the query $a{:}\exists r.T_\alpha$.

Now, since the query $a : \exists r.T_\alpha$ is entailed by the input qABox and TBox and since it has not been decided by the experts nor by inheritance within Phase 1, we conclude by Proposition 4.8 that the formula α is satisfiable iff. the query $a : \exists r. T_\alpha$ is not disputable. It follows that recognizing disputable consequences is co NP-hard.

It remains to prove containment in co NP. Conditions (DC1) and (DC2) can be checked in polynomial time since, more generally, entailment of queries in IRQ by ABoxes and qABoxes has this complexity. Furthermore, Conditions (DC3) and (DC3*) are equivalent by Proposition 4.8 and the negation of the latter can be decided in non-deterministic polynomial time, see Theorem 4.6. □

Surprisingly, it suffices to restrict attention to disputable CAs.

Proposition 4.10. *There are no disputable RAs at the end of Phase 1.*

The disputable consequences are exactly the CAs in the "entailment difference" between the saturated and the unsaturated repair induced by $\mathcal{S}^1$, i.e. those CAs entailed by the saturated repair but not by the unsaturated repair. Our implementation is based on this.

Lemma 4.11. *γ is a disputable consequence at the end of Phase 1 iff.*

(DC4) $\mathsf{rep}^{\mathcal{T}}_{\mathsf{IRQ}}(\exists X. \mathcal{A}, \mathcal{S}^1) \models^{\mathcal{T}} \gamma$ *and*

(DC3*) $\mathsf{rep}^{\mathcal{T}}_{\mathsf{unsat}}(\exists X. \mathcal{A}, \mathcal{S}^1) \not\models^{\mathcal{T}} \gamma$.

5 Summary and Future Prospects

We have delved into the topic of identifying and computing a practically relevant repair of a given knowledge base consisting of a quantified ABox and a static $\mathcal{EL}$ TBox, where the consequences repaired for are concept and role assertions. To this end, we introduced the *smart interaction strategy* to the optimal-repair framework, with which experts can interactively determine a suitable repair in polynomial time. Each such repair can be constructed by means of three operations (copying, deleting, and saturation) and the strategy allows us to efficiently control them. Moreover, we considered *disputable consequences*. Since the optimal repairs retain all disputable consequences but none of their substantiations within the input knowledge base, it might not be desirable to keep each of them but these decisions need to be made by the experts. An implementation in form of a plug-in for the knowledge-base editor Protégé is provided.

As future work, we want to extend the strategy to optimal repairs in more expressive DLs, e.g. Horn-$\mathcal{ALCROI}$ [8]. In order to increase support for wanted consequences in the repair request, we also want to combine it with interaction strategies for existing or novel abduction methods.

Acknowledgements. This work was supported by Deutsche Forschungsgemeinschaft (DFG) in Projects 430150274 (Repairing Description Logic Ontologies) and 389792660 (TRR 248: Foundations of Perspicuous Software Systems), and has further been supported by the Saxon State Ministry for Science, Culture, and Tourism (SMWK) by funding the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI).

References

[1] Franz Baader, Sebastian Brandt, and Carsten Lutz. 2005. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI*. Professional Book Center, 364–369. http://ijcai.org/Proceedings/05/Papers/0372.pdf.
[2] Franz Baader, Sebastian Brandt, and Carsten Lutz. 2008. Pushing the $\mathcal{EL}$ envelope further. In *Proc. of OWLED*. Vol. 496. CEUR-WS.org. http://ceur-ws.org/Vol-496/owled2008dc%5C_paper%5C_3.pdf.
[3] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic*. Cambridge University Press. doi: 10.1017/9781139025355.
[4] Franz Baader, Patrick Koopmann, and Francesco Kriegel. 2023. Optimal repairs in the description logic $\mathcal{EL}$ revisited. In *Proc. of JELIA* (LNCS). Vol. 14281. Springer, 11–34. doi: 10.1007/978-3-031-43619-2_2. See the erratum https://doi.org/10.5281/zenodo.10276270.
[5] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. 2021. Computing optimal repairs of quantified ABoxes w.r.t. static $\mathcal{EL}$ TBoxes. In *Proc. of CADE* (LNCS). Vol. 12699. Springer, 309–326. doi: 10.1007/978-3-030-79876-5_18.
[6] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. 2021. Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes (Extended Version). LTCS-Report 21-01. TUD. doi: 10.25368/2022.64.
[7] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. 2022. Optimal ABox repair w.r.t. static $\mathcal{EL}$ TBoxes: from quantified ABoxes back to ABoxes. In *Proc. of ESWC* (LNCS). Vol. 13261. Springer, 130–146. doi: 10.1007/978-3-031-06981-9_8.
[8] Franz Baader and Francesco Kriegel. 2022. Pushing optimal ABox repair from $\mathcal{EL}$ towards more expressive Horn-DLs. In *Proc. of KR*, 22–32. doi: 10.24963/kr.2022/3. See the addendum https://doi.org/10.5281/zenodo.8060198.
[9] Franz Baader, Francesco Kriegel, and Adrian Nuradiansyah. 2022. Error-tolerant reasoning in the description logic $\mathcal{EL}$ based on optimal repairs. In *Proc. of RuleML+RR* (LNCS). Vol. 13752. Springer, 227–243. doi: 10.1007/978-3-031-21541-4_15.
[10] Franz Baader, Francesco Kriegel, and Adrian Nuradiansyah. 2024. Inconsistency- and error-tolerant reasoning w.r.t. optimal repairs of $\mathcal{EL}^\bot$ ontologies. In *Proc. of FoIKS* (LNCS). Vol. 14589. Springer, 3–22. doi: 10.1007/978-3-031-56940-1_1.
[11] Franz Baader, Francesco Kriegel, and Adrian Nuradiansyah. 2023. Treating role assertions as first-class citizens in repair and error-tolerant reasoning. In *Proc. of SAC*. ACM, 974–982. doi: 10.1145/3555776.3577630.
[12] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. 2020. Computing compliant anonymisations of quantified ABoxes w.r.t. $\mathcal{EL}$ policies. In *Proc. of ISWC* (LNCS). Vol. 12506. Springer, 3–20. doi: 10.1007/978-3-030-62419-4_1.
[13] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. 2018. Making repairs in description logics more gentle. In *Proc. of KR*. AAAI Press, 319–328. https://aaai.org/ocs/index.php/KR/KR18/paper/view/18056.
[14] E. Di Sciascio, F.M. Donini, M. Mongiello, and G. Piscitelli. 2002. Meeting in the agora: a description logic approach to peer-to-peer e-commerce. In *Proc. of ITI*, 319–324 vol.1. doi: 10.1109/ITI.2002.1024693.
[15] Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. 2022. Connection-minimal abduction in $\mathcal{EL}$ via translation to FOL. In *Proc. of IJCAR* (LNCS). Vol. 13385. Springer, 188–207. doi: 10.1007/978-3-031-10769-6_12.
[16] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. 2010. *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC Press. http://www.semantic-web-book.org/.
[17] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. 2012. Practical reasoning with nominals in the $\mathcal{EL}$ family of description logics. In *Proc. of KR*. AAAI Press. http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4540.
[18] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. 2014. The incredible ELK – from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *J. Autom. Reason.*, 53, 1, 1–61. doi: 10.1007/s10817-013-9296-3.
[19] Patrick Koopmann. 2021. Signature-based abduction with fresh individuals and complex concepts for description logics. In *Proc. of IJCAI*. ijcai.org, 1929–1935. doi: 10.24963/IJCAI.2021/266.
[20] Patrick Koopmann, Warren Del-Pinto, Sophie Tourret, and Renate A. Schmidt. 2020. Signature-based abduction for expressive description logics. In *Proc. of KR*, 592–602. doi: 10.24963/KR.2020/59.
[21] Francesco Kriegel. 2024. Beyond Optimal: Interactive Identification of Better-than-optimal Repairs (Extended Version). LTCS-Report 24-05. TUD. doi: 10.25368/2024.319.
[22] Francesco Kriegel. 2022. Optimal fixed-premise repairs of $\mathcal{EL}$ TBoxes. In *Proc. of KI* (LNCS). Vol. 13404. Springer, 115–130. doi: 10.1007/978-3-031-15791-2_11.
[23] Francesco Kriegel. 2023. Optimal fixed-premise repairs of $\mathcal{EL}$ TBoxes (extended abstract). In *Proc. of KR, RPR track*. doi: 10.5281/zenodo.8341194.
[24] Markus Krötzsch. 2011. Efficient rule-based inferencing for OWL EL. In *Proc. of IJCAI*. IJCAI/AAAI, 2668–2673. doi: 10.5591/978-1-57735-516-8/IJCAI11-444.
[25] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. 2008. All elephants are bigger than all mice. In *Proc. of DL*. Vol. 353. CEUR-WS.org. https://ceur-ws.org/Vol-353/RudolphKraetzschHitzler.pdf.
[26] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. 2008. Cheap boolean role constructors for description logics. In *Proc. of JELIA* (LNCS). Vol. 5293. Springer, 362–374. doi: 10.1007/978-3-540-87803-2_30.
[27] Jingchuan Shi, Jiaoyan Chen, Hang Dong, Ishita Khan, Lizzie Liang, Qunzhi Zhou, Zhe Wu, and Ian Horrocks. 2023. Subsumption prediction for e-commerce taxonomies. In *Proc. of ESWC* (LNCS). Vol. 13870. Springer, 244–261. doi: 10.1007/978-3-031-33455-9_15.
[28] Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Egor V. Kostylev, Gong Cheng, Ernesto Jiménez-Ruiz, Ahmet Soylu, and Evgeny Kharlamov. 2022. Enhancing knowledge graph generation with ontology reshaping - Bosch case. In *Proc. of ESWC Satellite Events* (LNCS). Vol. 13384. Springer, 299–302. doi: 10.1007/978-3-031-11609-4_45.