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

Extended Abstract*

Francesco Kriegel[1,2]

[1] Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany
[2] Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden and Leipzig, Germany

Repairs of Knowledge Bases. Even though logic-based systems produce logically correct and explainable inferences, faulty conclusions might be drawn if the knowledge base (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.

In order to surmount both practicability issues of the classical method, we propose an interactive repair method which employs the optimal repair framework. Errors are not resolved by deleting a minimal number of statements but instead by modifying the KB such that only minimally many consequences are removed (including the observed faulty ones). 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 errors and proceed towards logical causes of identified faulty statements. Through this interactive top-down manner, the experts' workload is significantly lowered.

Optimal Repairs in General. The general setting is as follows, which covers both unwanted consequences as well as missing consequences (whereas previous definitions in the literature considered only unwanted consequences). A *repair request* is an assertion set $\mathcal{P} := \mathcal{P}_+ \uplus \mathcal{P}_-$ partitioned into an *addition part* $\mathcal{P}_+$ and a *removal part* $\mathcal{P}_-$. Of a consistent KB $\mathcal{K} := \mathcal{A} \uplus \mathcal{O}$, an *ABox repair* for $\mathcal{P}$ is an ABox $\mathcal{B}$ such that the KB $\mathcal{B} \cup \mathcal{O}$ is consistent, entails all assertions in $\mathcal{P}_+$, and does not entail any assertion in $\mathcal{P}_-$. That is, we treat the ABox $\mathcal{A}$ as refutable and

*This is an extended abstract of an article [1] published in the proceedings of the 40th ACM/SIGAPP Symposium On Applied Computing (SAC 2025) and of its extended version [2].

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

francesco.kriegel@tu-dresden.de (Francesco Kriegel)
https://tu-dresden.de/inf/lat/francesco-kriegel (Francesco Kriegel)
0000-0003-0219-0330 (Francesco Kriegel)

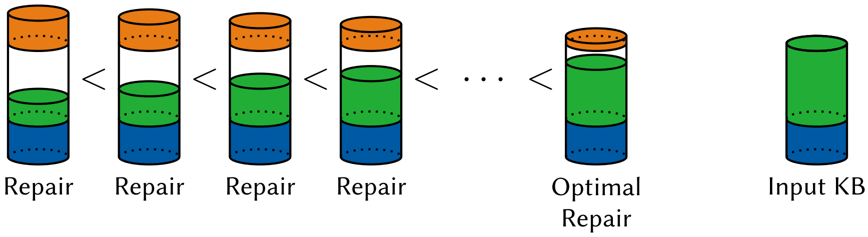

Figure 1: Illustration of an optimal repair

the ontology $\mathcal{O}$ as static — a reasonable assumption since ABoxes often contain vast amounts of collected observations whereas ontologies are usually more carefully maintained.

Repairs as defined above have no connection to the input ABox, but the following order relation $\geq$ between repairs takes it into account. We write $\mathcal{B} \geq \mathcal{C}$ and say that $\mathcal{B}$ is *at least as good as* another repair $\mathcal{C}$ if $\mathcal{B} \cup \mathcal{O}$ entails all retained knowledge in $\mathcal{C}$ (i.e. $\mathcal{B} \cup \mathcal{O} \models \gamma$ for each $\gamma \in \mathcal{C}$ with $\mathcal{K} \models \gamma$) and $\mathcal{C} \cup \mathcal{O}$ entails all additional knowledge in $\mathcal{B}$ (i.e. $\mathcal{C} \cup \mathcal{O} \models \gamma$ for each $\gamma \in \mathcal{B}$ with $\mathcal{K} \not\models \gamma$). Moreover, we write $\mathcal{B} > \mathcal{C}$ and say that $\mathcal{B}$ is *better than* $\mathcal{C}$ if $\mathcal{B} \geq \mathcal{C}$ but $\mathcal{C} \not\geq \mathcal{B}$, i.e. either less knowledge is added or less knowledge is removed. In order to emphasize that these two relations depend on the input KB $\mathcal{K}$ to be repaired, the symbols $\geq_{\mathcal{K}}$ and $>_{\mathcal{K}}$ could be used instead. A repair $\mathcal{B}$ is *optimal* if there is no repair better than $\mathcal{B}$. Figure 1 illustrates an optimal repair: between the optimal repair and the input KB no further repair exists.

Previous Work. There is no general approach to computing optimal repairs, and previous research focused on repair requests without an addition part. It seems that abduction methods [3–5] could be used to treat the addition part but it is still unclear how optimality could be achieved. 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.

Interestingly enough, optimal repairs need not exist even if the input KB can be repaired [6]. The reason is that every repair might be entailed by an even better repair, somewhat similar to the fact that irrational numbers have no best approximation by rational numbers. We say that the repair request $\mathcal{P}$ 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. W.r.t. KBs consisting of a refutable quantified ABox $\exists X.\mathcal{A}$ (i.e. an ABox with existentially quantified variables) and a static, terminating Horn-$\mathcal{ALCROI}$ ontology $\mathcal{O}$, all repair requests are optimally coverable and we can effectively compute all their optimal repairs [7–13]. With consequence-based entailment relations (weaker than the usual model-based entailment relation) also non-terminating ontologies are supported. For instance IRQ-entailment takes into account all $\mathcal{EL}$ concept assertions $a : C$ and all role assertions $(a, b) : r$ but no other queries (such as Boolean conjunctive queries). Examples 1, 6, 7, 26, 30 in the technical report [14] illustrate how optimal repairs in $\mathcal{EL}$ are built. In addition, the prototypical implementation[L1] developed for the underlying conference article [1] can be used with arbitrary $\mathcal{EL}$ ontologies to see the repair construction in action. Moreover, $\mathcal{EL}$ TBoxes can be optimally repaired when the left-hand sides of concept inclusions are fixed [15, 16].

Disputable Consequences. 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 chose another denotation to avoid confusion.) 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. That is, a *substantiation* of γ is an ABox $\mathcal{J}$ such that $\mathcal{A} \cup \mathcal{P}_+ \models \mathcal{J}$ 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 found in any repair (i.e. it is not justified anymore). More formally: given a consistent KB $\mathcal{K}$ and a repair request $\mathcal{P}$, a *disputable consequence* is an assertion γ such that $\mathcal{K} \cup \mathcal{P}_+ \models \gamma$, there is a repair that entails γ, there is another repair that does not entail γ, and for each repair $\mathcal{B}$, the KB $\mathcal{B} \cup \mathcal{O}$ does not entail any substantiation of γ. In order to obtain an optimal repair that makes sense in the application domain, we recommend to initially decide all disputable consequences and accordingly refine the repair request, i.e. add all accepted ones to $\mathcal{P}_+$ and all rejected ones to $\mathcal{P}_-$.

Deterministic Repair Requests. Even if repair requests are optimally coverable in the considered setting, they might be under-specified in the sense that there is no unique optimal repair. From a practical perspective, it then makes no sense to compute a random optimal repair. To see this, consider the KB consisting of the ABox $\{\mathsf{charlie} : \mathsf{Horse},\ \mathsf{charlie} : \mathsf{Male}\}$ and the TBox $\{\mathsf{Horse} \sqcap \mathsf{Male} \sqsubseteq \mathsf{Stallion}\}$. For the repair request $\mathcal{P}$ with removal part $\{\mathsf{charlie} : \mathsf{Stallion}\}$, there are two optimal ABox repairs: $\{\mathsf{charlie} : \mathsf{Horse}\}$ and $\{\mathsf{charlie} : \mathsf{Male}\}$. Now, if in real world Charlie is a horse, then the second, formally optimal repair does not make sense.

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 $\mathcal{P}$ is *deterministic* w.r.t. $\mathcal{K}$ 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. $\mathcal{O}$.

$$\mathcal{P} = \mathcal{P}_0 \xrightarrow{\text{refine}} \mathcal{P}_1 \xrightarrow{\text{refine}} \mathcal{P}_2 \xrightarrow{\text{refine}} \dots \xrightarrow{\text{refine}} \mathcal{P}_n \text{ deterministic}$$

Apart from computing disputable consequences, the question addressed in the conference article [1] is: how can we efficiently find a deterministic refinement of a given repair request?

Interaction Strategy. To this end, we consider the setting where the KB $\mathcal{K}$ consists of a quantified ABox $\exists X.\mathcal{A}$ and an $\mathcal{EL}$ ontology $\mathcal{O}$, the repair request $\mathcal{P}$ consists of concept assertions

$a : C$ and role assertions $(a, b) : r$, and IRQ-entailment is used. According to previous results, these repair requests are always optimally coverable, and there is a canonical form of optimal repairs that can be computed in exponential time [7–13]. From a technical perspective, each optimal repair can be obtained from the input by saturation and then delete and copy operations [9, 14]. 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.

We show that deciding disputable consequences in this setting is coNP-complete w.r.t. combined complexity — nevertheless, they can be computed rather efficiently for real-world KBs. W.r.t. data complexity, Lemma 4.11 implies that it can be decided in polynomial time only whether an assertion is a disputable consequence. We further present an interaction strategy with which a deterministic refinement can be identified in polynomially many steps, i.e. the number of questions that need to be answered by the users/experts is polynomial. Every (theoretically) optimal repair can be found with the strategy, and it runs in two phases:

- Phase 1 is devoted to identifying the causes of the initially reported errors in $\mathcal{P}$.
- Phase 2 first computes all disputable consequences that are concept assertions built from the subconcepts in the input or role assertions, and then proceeds with these as Phase 1.

In both phases, the questions are not fixed but evolve according to the answers received. If a question inherits its answer already by logical reasoning from previous answers, then it is not presented to the users/experts.

An implementation of the underlying repair construction as well as the interaction strategy is available.[L1] It comes in form of a plug-in for the KB editor Protégé.[L2] Since it is implemented in the programming language Scala,[L3] it can be used on any operating system for which a Java virtual machine (JVM) is available. It is recommended to use a modern JVM like GraalVM[L4] since it offers faster code execution (often twice as fast as a standard JVM). Furthermore, the implementation employs the currently fastest $\mathcal{EL}$ reasoner: ELK.[L5] On MacOS, the easiest way to try out the implementation is to use the installer script.[L6] For other operating systems this script can be easily adapted. More details are explained on the start page of the GitHub repository.[L1]

Even though coNP-hardness of disputable consequences seems to indicate that Phase 2 would be intractable, the implementation works sufficiently fast even with large TBoxes such as (the $\mathcal{EL}$ fragment of) SNOMED CT,[L7] which contains more than 360,000 concept names. Interactively identifying a repair of an ABox representing data on a patient having a common cold and then computing this repair completes within about four to five minutes. More specifically, in this experiment we used an ABox containing a single assertion stating that a particular person has a common cold; the user interaction then amounted to about 30 questions only (as the interaction process is local to the statements to be repaired for). This computation time can surely be further improved with optimizations or faster programming languages such as C++.

Future Work. As future work, we want to extend the strategy to optimal repairs in more expressive DLs such as Horn-$\mathcal{ALCROI}$ [10]. 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.

Acknowledgments

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) in Project 389792660 (TRR 248: Foundations of Perspicuous Software Systems) and in Project 558917076 (Construction and Repair of Description-logic Knowledge Bases) as well as 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] Francesco Kriegel. Beyond Optimal: Interactive Identification of Better-than-optimal Repairs. In: *Proceedings of the 40th ACM/SIGAPP Symposium On Applied Computing (SAC 2025), March 31 – April 4, 2025, Sicily, Italy*. 2025, pp. 1019–1026. DOI: 10.1145/3672608.3707750.

[2] Francesco Kriegel. *Beyond Optimal: Interactive Identification of Better-than-optimal Repairs (Extended Version)*. LTCS-Report 24-05. TUD, 2024. DOI: 10.25368/2024.319.

[3] Patrick Koopmann, Warren Del-Pinto, Sophie Tourret, Renate A. Schmidt. Signature-Based Abduction for Expressive Description Logics. In: *Proc. of KR*. 2020, pp. 592–602. DOI: 10.24963/KR.2020/59.

[4] Patrick Koopmann. Signature-Based Abduction with Fresh Individuals and Complex Concepts for Description Logics. In: *Proc. of IJCAI*. 2021, pp. 1929–1935. DOI: 10.24963/IJCAI.2021/266.

[5] Fajar Haifani, Patrick Koopmann, Sophie Tourret, Christoph Weidenbach. Connection-Minimal Abduction in $\mathcal{EL}$ via Translation to FOL. In: *Proc. of IJCAR*. 2022, pp. 188–207. DOI: 10.1007/978-3-031-10769-6_12.

[6] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, Rafael Peñaloza. Making Repairs in Description Logics More Gentle. In: *Proc. of KR*. 2018, pp. 319–328. URL: https://aaai.org/ocs/index.php/KR/KR18/paper/view/18056.

[7] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, Rafael Peñaloza. Computing Compliant Anonymisations of Quantified ABoxes w.r.t. $\mathcal{EL}$ Policies. In: *Proc. of ISWC*. 2020, pp. 3–20. URL: https://doi.org/10.1007/978-3-030-62419-4_1.

[8] Franz Baader, Patrick Koopmann, Francesco Kriegel, Adrian Nuradiansyah. Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes. In: *Proc. of CADE*. 2021, pp. 309–326. URL: https://doi.org/10.1007/978-3-030-79876-5_18.

[9] Franz Baader, Patrick Koopmann, Francesco Kriegel. Optimal Repairs in the Description Logic $\mathcal{EL}$ Revisited. In: *Proc. of JELIA*. See the erratum https://doi.org/10.5281/zenodo.10276270. 2023, pp. 11–34. URL: https://doi.org/10.1007/978-3-031-43619-2_2.

[10] Franz Baader, Francesco Kriegel. Pushing Optimal ABox Repair from $\mathcal{EL}$ Towards More Expressive Horn-DLs. In: *Proc. of KR*. See the addendum https://doi.org/10.5281/zenodo.8060198. 2022, pp. 22–32. URL: https://doi.org/10.24963/kr.2022/3.

[11] Franz Baader, Patrick Koopmann, Francesco Kriegel, Adrian Nuradiansyah. Optimal ABox Repair w.r.t. Static $\mathcal{EL}$ TBoxes: from Quantified ABoxes back to ABoxes. In: *Proc. of ESWC*. 2022, pp. 130–146. URL: https://doi.org/10.1007/978-3-031-06981-9_8.

[12] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah. Treating Role Assertions as First-class Citizens in Repair and Error-tolerant Reasoning. In: *Proc. of SAC*. 2023, pp. 974–982. URL: https://doi.org/10.1145/3555776.3577630.

[13] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah. Inconsistency- and Error-Tolerant Reasoning w.r.t. Optimal Repairs of $\mathcal{EL}^{\bot}$ Ontologies. In: *Proc. of FoIKS*. 2024, pp. 3–22. DOI: 10.1007/978-3-031-56940-1_1.

[14] Franz Baader, Patrick Koopmann, Francesco Kriegel. *Optimal Repairs in the Description Logic $\mathcal{EL}$ Revisited (Extended Version)*. LTCS-Report 23-03. TUD, 2023. URL: https://doi.org/10.25368/2023.121.

[15] Francesco Kriegel. Optimal Fixed-Premise Repairs of $\mathcal{EL}$ TBoxes (Extended Abstract). In: *Proc. of KR, RPR track*. 2023. DOI: 10.5281/zenodo.8341194.

[16] Francesco Kriegel. Optimal Fixed-Premise Repairs of $\mathcal{EL}$ TBoxes. In: *Proc. of KI*. 2022, pp. 115–130. DOI: 10.1007/978-3-031-15791-2_11.

Links

L1 https://github.com/francesco-kriegel/interactive-optimal-repairs

L2 https://protege.stanford.edu

L3 https://www.scala-lang.org

L4 https://www.graalvm.org

L5 https://github.com/liveontologies/elk-reasoner

L6 https://raw.githubusercontent.com/francesco-kriegel/interactive-optimal-repairs/main/install-macos.sh

L7 https://www.snomed.org/