Repairing $\mathcal{EL}$ Ontologies Using Weakening and Completing

Ying Li[1,2][0000−0002−1367−9679] and Patrick Lambrix[1,2,3][0000−0002−9084−0470]

[1] Linköping University, Linköping, Sweden
[2] The Swedish e-Science Research Centre, Linköping, Sweden
[3] University of Gävle, Sweden

Abstract. The quality of ontologies in terms of their correctness and completeness is crucial for developing high-quality ontology-based applications. Traditional debugging techniques repair ontologies by removing unwanted axioms, but may thereby remove consequences that are correct in the domain of the ontology. In this paper we propose an interactive approach to mitigate this for $\mathcal{EL}$ ontologies by axiom weakening and completing. We present the first approach for repairing that takes into account removing, weakening and completing. We show different combination strategies, discuss the influence on the final ontologies and show experimental results. We show that previous work has only considered special cases and that there is a trade-off, and how to deal with it, involving the amount of validation work for a domain expert and the quality of the ontology in terms of correctness and completeness. We also present new algorithms for weakening and completing.

1 Introduction

Debugging ontologies aims to remove unwanted knowledge in the ontology. This can be knowledge that leads to logical problems such as inconsistency or incoherence (semantic defects) or statements that are not correct in the domain of the ontology (modeling defects) (e.g., [16]). The workflow consists of several steps including the detection and localization of the defects and the repairing. In this paper we assume we have detected and localized the defects, e.g., using traditional debugging techniques as in, e.g., [29,28,30,22,16,15,14,20,13,23,32,12,1,26,27], and we now need to repair the ontology. In the classical approaches for debugging the end result is a set of axioms to remove from the ontology that is obtained after detection and localization, and the repairing consists solely of removing the suggested axioms. However, first, these approaches are usually purely logic-based and therefore may remove correct axioms (e.g., [25]). Therefore, it is argued that a domain expert should validate the results of such systems. Furthermore, removing an axiom may remove more knowledge than necessary. Correct knowledge that is derivable with the help of the wrong axioms may not be derivable in the new ontology. In this paper we mitigate these effects of removing wrong axioms by, in addition to removing those axioms, also adding correct knowledge. Two approaches could be used. A first approach is to replace a wrong axiom with a weakened version of the axiom (e.g., [17,9,5,33]). Another approach is to complete[4] an ontology (e.g., [35]) which adds previously unknown correct axioms that allow to derive existing axioms, and that could be used on the results of weakening. These approaches have, however, not been studied together.

In this paper we focus on $\mathcal{EL}$ ontologies. $\mathcal{EL}$ is a description logic for which subsumption checking remains tractable and that is used (as is or with small extensions) by well-known ontologies such as SNOMED or Gene Ontology [2]. Further, we assume that we are given a set of wrong axioms W that we want to remove from the ontology and that when removing these axioms, they cannot be derived from the ontology anymore.

Our main contribution (i) (Sect. 5) is a framework for weakening and completing ontologies. It is the first work that combines removing with weakening and completing. For this framework we give a formal definition of the repairing problem, and introduce different operations for combining removing, weakening and completing approaches, and their relationships. Using the relationships between these operations, we show that different solutions to the repairing problem exist even using the same basic weakening and completing algorithms (an insight that no other work has discussed), that there is a trade-off involving completeness and correctness of the resulting ontologies with more validation effort for more complete ontologies (another insight that no other work has discussed), and we show how basic algorithms can be combined according to a preference for the level of completeness. Earlier work on weakening and earlier work on completing can be represented using our operators and their particular weakening and completing algorithms. Using the framework we can show that earlier work on weakening used one particular combination strategy (although with different weakening algorithms by different authors). Similarly, work on completing used one particular combination strategy. Our work shows thus that there are different variants of the earlier work by combining their basic algorithms in different ways, with trade-offs involving completeness, correctness and validation work.

In addition to the formal framework there are also other contributions. (ii) We show the trade-offs for 13 different combination strategies for 6 ontologies in experiments (Sect. 6). Further, in Sect. 4 (iii) we develop a new algorithm for weakening and a new algorithm for completing. For efficiency reasons, weakening algorithms restrict the search space, and we propose a new heuristic for this restriction. Our algorithm for completing is an extension of the approach in [35]. Finally, (iv) we provide two implemented systems, a Protégé plugin and a stand-alone system (Sect. 8).

[4] This term has been used with different meanings. In this paper we refer to completing as the dual task of weakening. The term has been used with other meanings in, e.g., [4,31]. Related terms are, e.g., ontology extension [21], ontology learning [6], ontology enrichment [11], and ontology revision [24].

2 Preliminaries

In this paper we assume that ontologies are represented using a description logic TBox. Description logics [3] are knowledge representation languages where concept descriptions are constructed inductively from a set N_C of atomic concepts and a set N_R of atomic roles and (possibly) a set N_I of individual names. Different description logics allow for different constructors for defining complex concepts and roles. An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ which assigns to each atomic concept $P \in N_C$ a subset $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to each atomic role $r \in N_R$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each individual name[5] $i \in N_I$ an element $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is straightforwardly extended to complex concepts. A TBox is a finite set of axioms which in $\mathcal{EL}$ are *general concept inclusions* (GCIs). The syntax and semantics for $\mathcal{EL}$ are shown in Table 1.

Table 1. $\mathcal{EL}$ syntax and semantics. (Note that P and Q are arbitrary concepts. In the remainder we often use P and Q for atomic concepts.)

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
conjunction	$P \sqcap Q$	$P^{\mathcal{I}} \cap Q^{\mathcal{I}}$
existential restriction	$\exists r.P$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} :$ $(x, y) \in r^{\mathcal{I}} \wedge y \in P^{\mathcal{I}}\}$
GCI	$P \sqsubseteq Q$	$P^{\mathcal{I}} \subseteq Q^{\mathcal{I}}$

An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if for each GCI in $\mathcal{T}$, the semantic conditions are satisfied.[6] One of the main reasoning tasks for description logics is subsumption checking in which the problem is to decide for a TBox $\mathcal{T}$ and concepts P and Q whether $\mathcal{T} \models P \sqsubseteq Q$, i.e., whether $P^{\mathcal{I}} \subseteq Q^{\mathcal{I}}$ for every model of TBox $\mathcal{T}$. In this paper we update the TBox during the repairing and we always use subsumption with respect to the current TBox.

3 Problem formulation

We can now formally define the repairing problem that we want to solve (Definition 1). We are given a set of wrong axioms W that we want to remove[7] from

[5] As we do not deal with individuals in this paper, we do not use individuals in the later sections.

[6] We do not take up consistency of TBoxes, i.e., whether a model exists or not, in this paper as every $\mathcal{EL}$ TBox is consistent.

[7] We note that in this paper we deal with removing and not the full debugging problem, i.e., we assume that the axioms to be removed are already found. Removing can be seen as a simple kind of debugging, or as the second step of the debugging process.

the ontology and that when they are removed, they cannot be derived from the TBox representing the ontology anymore. Further, to guarantee a high level of quality of the ontology (i.e., so that no correct information is removed or no incorrect information is added), domain expert validation is a necessity (e.g., [25]). Therefore, we assume an oracle (representing a domain expert) that, when given an axiom, can answer whether this axiom is correct or wrong in the domain of interest of the ontology. We have not required specific properties regarding the performance of the oracle. For instance, we did not require that an oracle always answers correctly or that the oracle gives consistent answers. As a first step we have chosen this way as it reflects reality. According to our long experience working with domain experts in ontology engineering, domain experts make mistakes. However, this does not necessarily mean that domain expert validation is not useful. In experiments in ontology alignment, it was shown that oracles making up to 30% mistakes were still beneficial (e.g., [8]). Further, requiring consistent answers seems to be a tough requirement for domain experts. This would require the ability to reason with long proof chains, while humans usually do well for chains of limited length. It is also not clear how to check that a particular domain expert would fulfil the required properties. Therefore, in this work we do not require such properties, but provide user support in our systems by providing warnings when incompatible validations are made and then allow the domain expert to revise the validations. We do acknowledge, however, that requiring such properties and thereby classifying types of domain experts, may allow us to guarantee certain properties regarding correctness and completeness and allow us to reduce the search space of possible repairs.

A repair for the ontology given the TBox $\mathcal{T}$, oracle Or, and a set of wrong axioms W, is a set of correct axioms that when added to the TBox where the axioms in W are removed will not allow deriving the axioms in W.

Definition 1. (Repair) Let $\mathcal{T}$ be a TBox. Let Or be an oracle that given a TBox axiom returns true or false. Let W be a finite set of TBox axioms in $\mathcal{T}$ such that $\forall\ \psi \in W$: $Or(\psi)$ = false. Then, a repair for Debug-Problem DP$(\mathcal{T}, Or, W)$ is a finite set of TBox axioms A such that
(i) $\forall\ \psi \in A$: $Or(\psi)$ = true;
(ii) $\forall\ \psi \in W$: $(\mathcal{T} \cup A) \setminus W \not\models \psi$.

Our aim is to find repairs that remove as much wrong knowledge and add as much correct knowledge to our ontology as possible. Therefore, we introduce the preference relations *less incorrect* and *more complete* between ontologies (Definition 2) that formalize these intuitions, respectively.

Definition 2. (less incorrect/more complete - ontologies) Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two ontologies represented by TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively.
Then, $\mathcal{O}_1$ is *less incorrect* than $\mathcal{O}_2$ ($\mathcal{O}_2$ is *more incorrect* than $\mathcal{O}_1$) iff $(\forall \psi : (\mathcal{T}_1 \models \psi \wedge Or(\psi) = false) \rightarrow \mathcal{T}_2 \models \psi)) \wedge (\exists \psi : Or(\psi) = false\ \wedge \mathcal{T}_1 \not\models \psi \wedge \mathcal{T}_2 \models \psi)$.
$\mathcal{O}_1$ and $\mathcal{O}_2$ are *equally incorrect* iff $\forall \psi : Or(\psi) = false \rightarrow (\mathcal{T}_1 \models \psi \leftrightarrow \mathcal{T}_2 \models \psi)$
Further, $\mathcal{O}_1$ is *more complete* than $\mathcal{O}_2$ (or $\mathcal{O}_2$ is *less complete* than $\mathcal{O}_1$) iff $(\forall \psi :$

$(\mathcal{T}_2 \models \psi \wedge Or(\psi) = true) \rightarrow \mathcal{T}_1 \models \psi)) \wedge (\exists \psi : Or(\psi) = true \wedge \mathcal{T}_1 \models \psi \wedge \mathcal{T}_2 \not\models \psi)$.
$\mathcal{O}_1$ and $\mathcal{O}_2$ are *equally complete* iff $\forall \psi : Or(\psi) = true \rightarrow (\mathcal{T}_1 \models \psi \leftrightarrow \mathcal{T}_2 \models \psi)$

4 Weakening and completing algorithms

We now define algorithms for weakening and completing that we use in our experiments in Sect. 6. Completing is used on the results of weakening.

Basics. We assume that ontologies are represented by *normalized* $\mathcal{EL}$ TBoxes. A normalized $\mathcal{EL}$ TBox $\mathcal{T}$ contains only axioms of the forms $P \sqsubseteq Q$, $P \sqcap Q \sqsubseteq R$, $\exists r.P \sqsubseteq Q$ and $P \sqsubseteq \exists r.Q$ where P, Q, $R \in N_C$ and $r \in N_R$. Every $\mathcal{EL}$ TBox can in linear time be transformed into a normalized TBox that is a conservative extension, i.e., every model of the normalized TBox is also a model of the original TBox and every model of the original TBox can be extended to a model of the normalized TBox [2]. Further, we define the *simple complex concept set for a TBox* $\mathcal{T}$, which contains all atomic concepts in the ontology as well as the concepts that can be constructed by using one constructor ($\sqcap$ or $\exists$) and only atomic concepts and roles in the ontology (Definition 3). Note that $\top$ is not in SCC($\mathcal{T}$). Further, if the number of concepts in $N_C^{\mathcal{T}}$ is n and the number of roles in $N_R^{\mathcal{T}}$ is t, then the number of concepts in SCC($\mathcal{T}$) is $(n^2 + n)/2 + tn$.

Definition 3. For a normalized $\mathcal{EL}$ TBox $\mathcal{T}$ with $N_C^{\mathcal{T}}$ the set of atomic concepts occurring in $\mathcal{T}$ and $N_R^{\mathcal{T}}$ the set of atomic roles occurring in $\mathcal{T}$, we define the *simple complex concept set for* $\mathcal{T}$, denoted by SCC($\mathcal{T}$), as the set containing all the concepts of the forms P, $P \sqcap Q$, and $\exists r.P$ where P, $Q \in N_C^{\mathcal{T}}$ and $r \in N_R^{\mathcal{T}}$.

In our algorithms we use two basic operations which remove and add axioms to a TBox. The result of *Remove-axioms*($\mathcal{T}$,D) for a TBox $\mathcal{T}$ and a set of axioms D is the TBox $\mathcal{T} \setminus D$. If D contains only wrong axioms (such as W), then the ontology represented by Remove-axioms($\mathcal{T}$,D) is less (if at least one of the removed axioms cannot be derived anymore) or equally incorrect (if all removed axioms can still be derived), as well as less (if some correct axioms cannot be derived anymore by removing the wrong ones) or equally complete (if all correct axioms can still be derived), than the ontology represented by $\mathcal{T}$. The result of *Add-axioms*($\mathcal{T}$,A) for a TBox $\mathcal{T}$ and a set of axioms A is the TBox $\mathcal{T} \cup A$. If A contains only correct axioms then the ontology represented by Add-axioms($\mathcal{T}$,A) is more (if some added axiom was not derivable from the ontology) or equally complete (if all added axioms were derivable from the ontology), as well as more (if some wrong axioms can now be derived by adding the new ones) or equally incorrect (if no new wrong axioms can now be derived by adding the new ones), than the ontology represented by $\mathcal{T}$.

We also need to compute sub-concepts and super-concepts of concepts. However, to reduce the infinite search space of possible axioms to add during weakening and completing, we limit the use of nesting operators while computing sub- and super-concepts.[8] This we do by only considering sub- and super-concepts in

[8] Weaker limitations are possible, but the weaker the restriction, the larger the solution search space and the higher the probability of a less usable practical system.

the SCC of a TBox (Definition 4). As subsumption checking in $\mathcal{EL}$ is tractable, finding these sub- and super-concepts is tractable.

Definition 4. (super- and sub-concepts in SCC)
$sup(P, \mathcal{T}) \leftarrow \{\ sp \mid \mathcal{T} \models P \sqsubseteq sp \wedge sp \in \mathrm{SCC}(\mathcal{T})\}$
$sub(P, \mathcal{T}) \leftarrow \{\ sb \mid \mathcal{T} \models sb \sqsubseteq P \wedge sb \in \mathrm{SCC}(\mathcal{T})\}$

Finally, as we work on normalized $\mathcal{EL}$ TBoxes, we need to make sure that when adding axioms, these are of one of the forms $P \sqsubseteq Q$, $P \sqcap Q \sqsubseteq R$, $\exists r.P \sqsubseteq Q$ and $P \sqsubseteq \exists r.Q$ where P, Q, $R \in N_C^{\mathcal{T}}$ and $r \in N_R^{\mathcal{T}}$. We note that new atomic concepts, not originally in the ontology, may be introduced.

Algorithm 1 Weakened axiom set

Input: TBox $\mathcal{T}$, Oracle Or, unwanted axiom $\alpha \sqsubseteq \beta$
Output: Weakened axiom set of $\alpha \sqsubseteq \beta$

```
wt_{α⊑β} ←{sb ⊑ sp | sb ∈ sub(α,T) ∧ sp ∈ sup(β,T) ∧ Or(sb ⊑ sp) = True ∧ ¬∃
   sb′ ∈ sub(α,T), sp′ ∈ sup(β,T): (Or(sb′ ⊑ sp′) = True ∧ ((sb ⊑ sb′ ∧ sp′ ⊏ sp) ∨
   (sb ⊏ sb′ ∧ sp′ ⊑ sp)))}
w_{α⊑β} ← ∅
for each sb ⊑ sp ∈ wt_{α⊑β} do
    w_{α⊑β} ← w_{α⊑β} ∪ Normalize(sb ⊑ sp)
end for
return w_{α⊑β}
```

Algorithm 2 Completed axiom set

Input: TBox $\mathcal{T}$, Oracle Or, a wanted axiom $\alpha \sqsubseteq \beta$
Output: Completed axiom set of $\alpha \sqsubseteq \beta$

```
ct_{α⊑β} ←{sp ⊑ sb | sp ∈ sup(α,T) ∧ sb ∈ sub(β,T) ∧ Or(sp ⊑ sb) = True ∧ ¬∃
   sp′ ∈ sup(α), sb′ ∈ sub(β): (Or(sp′ ⊑ sb′) = True ∧ (sp ⊑ sp′ ∧ sb′ ⊏ sb) ∨ (sp ⊏
   sp′ ∧ sb′ ⊑ sb)}
c_{α⊑β} ← ∅
for each sb ⊑ sp ∈ ct_{α⊑β} do
    c_{α⊑β} ← c_{α⊑β} ∪ Normalize(sb ⊑ sp)
end for
return c_{α⊑β}
```

Weakening and completing. Given an axiom, *weakening* aims to find other axioms that are weaker than the given axiom, i.e., the given axiom logically implies the other axioms. For an axiom $\alpha \sqsubseteq \beta$, this is often done by replacing α by a more specific concept or replacing β by a more general concept. For the repairing this means that a wrong axiom $\alpha \sqsubseteq \beta$ can be replaced by a correct weaker axiom, thereby mitigating the effect of removing the wrong axiom (Fig. 1). Algorithm 1 presents a tractable weakening algorithm for normalized $\mathcal{EL}$ TBoxes. For a given axiom $\alpha \sqsubseteq \beta$, it finds correct axioms $sb \sqsubseteq sp$ such that sb is a sub-concept in SCC($\mathcal{T}$) of α and sp is a super-concept in SCC($\mathcal{T}$) of β. Further, there should not be another correct axiom under these conditions that would add more correct knowledge to the ontology than $sb \sqsubseteq sp$. As we work

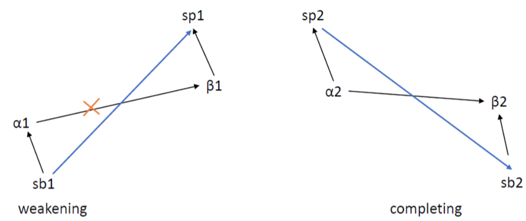


Fig. 1. Examples. Weakening: unwanted axiom $\alpha 1 \sqsubseteq \beta 1$ is replaced by correct axiom $sb1 \sqsubseteq sp1$; assumed that $\alpha 1 \sqsubseteq sp1$ is not correct; formerly derivable correct axiom $sb1 \sqsubseteq sp1$ still entailed by repaired ontology. Completion: wanted axiom $\alpha 2 \sqsubseteq \beta 2$ is replaced by correct axiom $sp2 \sqsubseteq sb2$; $\alpha 2 \sqsubseteq \beta 2$ is still derivable and additional correct axiom $sp2 \sqsubseteq sb2$ in the repaired ontology.

with normalized $\mathcal{EL}$ TBoxes, the new axioms are normalized. The existence of such weaker axioms is not guaranteed.

Completing aims to find correct axioms that are not derivable from the ontology yet and that would make a given axiom derivable. It was introduced to aid domain experts when adding axioms to the ontology to find additional knowledge to add. While weakening is usually performed on unwanted axioms, completing is usually performed on wanted axioms. Algorithm 2 presents a tractable completion algorithm for normalized $\mathcal{EL}$ TBoxes. For a given axiom $\alpha \sqsubseteq \beta$, it finds correct axioms $sp \sqsubseteq sb$ such that sp is a super-concept in SCC($\mathcal{T}$) of α and sb is a sub-concept in SCC($\mathcal{T}$) of β (Fig. 1). This means that if $sp \sqsubseteq sb$ is added to $\mathcal{T}$, then $\alpha \sqsubseteq \beta$ would be derivable. Further, there should not be another correct axiom under these conditions that would add more correct knowledge to the ontology than $sp \sqsubseteq sb$. Similarly as for weakening, the new axioms are normalized. The completed axiom set is guaranteed to be not empty for a correct axiom $\alpha \sqsubseteq \beta$. It contains $\alpha \sqsubseteq \beta$ or other axioms that lead to the derivation of $\alpha \sqsubseteq \beta$.

Note that weakening and completing are dual operations where the former finds weaker axioms and the latter stronger axioms. This is reflected in the mirroring of the sub- and super-concepts of α and β in Algorithms 1 and 2.

5 Combination strategies

Given a set of wrong axioms, there are different ways to repair the ontology using the removing, weakening and completing operations. There are choices to be made regarding the use of wrong axioms in the weakening and completing steps, regarding removing, weakening and completing all axioms at once or one at a time and in the latter case regarding the order the axioms are processed, as well as regarding when to update the ontology. Each of these choices may have an influence on the completeness and correctness of the repaired ontology. In general, using as much (possibly wrong) information as possible may lead to more complete ontologies, but also requires a larger validation effort. We have experimented with 13 different kinds of combinations. In this paper we show 4 representative algorithms that we use in the discussion as examples for general statements. (We note that all proposed algorithms are tractable and find repairs as defined in Definition 1.

To show the trade-off between the choices regarding completeness and validation effort between the different algorithms, we define operators in Table 2 that can be used as building blocks in the design of algorithms. The operations represent choices regarding the use of wrong axioms by removing them (R) and adding them back (AB), regarding weakening (W) and completing (C) one at a time or all at once. Furthermore, update (U) is always used in combination with weakening or completing, and relates to when changes to the ontology are performed. For instance, when weakening an axiom, the weakened axioms could be added immediately to the ontology (and thus influence the ontology before weakening other axioms) or can be added after having weakened all axioms (and thus weakening one axiom does not influence weakening the next axiom). In the algorithms this is represented by the use of $\mathcal{T}$ or $\mathcal{T}_r$ as TBox. The operations have different effects on the completeness of the final ontology and validation effort. This is represented in the Hasse diagrams in Fig. 3 where the partial order represents more or equally complete final ontologies. For instance, Fig. 3b shows that weakening one axiom at a time and immediately updating the TBox (W-one,U-now) leads to a more complete ontology (and more validation effort) than the other choices. Fig. 3c, shows that ontologies repaired by algorithms using one axiom at a time completing and immediate updates (C-one,U-now) are more complete than ontologies repaired using one axiom at a time completing and updating the ontology after each weakened axiom set for a wrong axiom (C-one,U-end_one). These ontologies are in turn more complete than for the other choices. Similar observations regarding removing are in Fig. 3a.

The combination algorithms can be defined by which of these building blocks are used and in which order. For instance, Algorithm C9 uses weaken one at a time, remove all wrong, complete one at a time, then add completed axiom sets at the end, while Algorithm C10 uses weaken one at a time, remove all wrong, add completed axiom sets one at a time. We can then compare algorithms using the Hasse diagrams. If the sequence of operators for one algorithm can be transformed to the sequence of operators of a second algorithm, by replacing some operators of the first algorithm using operators higher up in the lattices in Fig. 3, then the ontologies repaired using the second algorithm are more (or equally) complete than the ontologies repaired using the first algorithm. For instance, the sequence of Algorithm C9 can be rewritten into the sequence of Algorithm C10 by replacing the completion operator to a higher-level completion operator. Thus, repairing an ontology using Algorithm C10 leads to a more (or equally) complete ontology than repairing using Algorithm C9.

6 Experiments

In order to compare the use of the different combinations of strategies, we run experiments on several ontologies: Mini-GALEN (used as our running example), PACO, NCI, OFSMR, EKAW and Pizza ontology (Table 3). We have used the

Algorithm C2 Remove/weaken/add weakened axiom sets one at a time

Input: TBox $\mathcal{T}$, Oracle Or, set of unwanted axioms W
Output: A repaired TBox

$\mathcal{T}_r \leftarrow \mathcal{T}$
for each $\alpha \sqsubseteq \beta \in W$ **do**
  $\mathcal{T}_r \leftarrow$ Remove-axioms($\mathcal{T}_r$, $\{\alpha \sqsubseteq \beta\}$)
  $w_{\alpha \sqsubseteq \beta} \leftarrow$ weakened-axiom-set($\alpha \sqsubseteq \beta$, $\mathcal{T}_r, Or$)
  $\mathcal{T}_r \leftarrow$ Add-axioms($\mathcal{T}_r$,$w_{\alpha \sqsubseteq \beta}$)
end for
return $\mathcal{T}_r$

Algorithm C4 Remove all wrong, weaken/add weakened axiom sets one at a time

Input: TBox $\mathcal{T}$, Oracle Or, set of unwanted axioms W
Output: A repaired TBox

$\mathcal{T}_r \leftarrow$ Remove-axioms($\mathcal{T}$, W)
for each $\alpha \sqsubseteq \beta \in W$ **do**
  $w_{\alpha \sqsubseteq \beta} \leftarrow$ weakened-axiom-set($\alpha \sqsubseteq \beta$, $\mathcal{T}_r, Or$)
  $\mathcal{T}_r \leftarrow$ Add-axioms($\mathcal{T}_r$,$w_{\alpha \sqsubseteq \beta}$)
end for
return $\mathcal{T}_r$

parts of these ontologies that are expressible in $\mathcal{EL}$ in the sense that we removed the parts of axioms that used constructors not in $\mathcal{EL}$. We introduced new axioms in the ontologies by replacing existing axioms with axioms where the left-hand or right-hand side concepts of the existing axioms were changed. Further, we also flagged axioms as wrong in our full experiment set (e.g., in PACO). All axioms were validated manually.

For subsumption checking in the algorithms we used HermiT (`http://www.hermit-reasoner.com/`). We give results for Mini-GALEN (Fig. 2) which are representative for all experiments. Table 4 shows results for Algorithm C2 vs C4 regarding the number of sub-concepts of α and super-concepts of β for each wrong axiom $\alpha \sqsubseteq \beta$ when choosing to remove one wrong axiom at a time or all at once (while both update using the weakening result at once). In the table for each algorithm there is one sub and one sup set for each of the wrong axioms (e.g., for C2 for the first wrong axiom there are 3 concepts in the sup set and 2 in the sub set, resulting in 6 candidate weakened axioms). Further, the weakened axioms are shown. Table 5 shows the sizes of the sub and sup sets of the axioms (PPr$\sqsubseteq$IPr, IPr$\sqsubseteq$GPr, $\sqsubseteq$PPr), and the axioms to add using different orders of computing weakened axioms sets and adding them as soon as they are found for Algorithm C4. In Table 6 we show the sizes of the sub and sup sets for the completing step as well as the completed axioms for Algorithms C9 and C10.

Algorithm C9 Weaken one at a time, remove all wrong, complete one at a time, add completed axiom sets at end

Input: TBox $\mathcal{T}$, Oracle Or, set of unwanted axioms W
Output: A repaired TBox

for each $\alpha \sqsubseteq \beta \in W$ **do**
  $\mathcal{T}_r \leftarrow$ Remove-axioms($\mathcal{T}$, $\{\alpha \sqsubseteq \beta\}$)
  $w_{\alpha \sqsubseteq \beta} \leftarrow$ weakened-axiom-set($\alpha \sqsubseteq \beta$, $\mathcal{T}_r, Or$)
end for
$\mathcal{T}_r \leftarrow$ Remove-axioms($\mathcal{T}_r$,W)
for each $\alpha \sqsubseteq \beta \in W$ **do**
  $c_{\alpha \sqsubseteq \beta} \leftarrow \emptyset$
  for each $sb \sqsubseteq sp \in w_{\alpha \sqsubseteq \beta}$ **do**
    $c_{sb \sqsubseteq sp} \leftarrow$ completed-axiom-set($sb \sqsubseteq sp$,$\mathcal{T}_r, Or$)
    $c_{\alpha \sqsubseteq \beta} \leftarrow c_{\alpha \sqsubseteq \beta} \cup c_{sb \sqsubseteq sp}$
  end for
end for
$\mathcal{T}_r \leftarrow$ Add-axioms($\mathcal{T}_r$,$\bigcup_{\alpha \sqsubseteq \beta} c_{\alpha \sqsubseteq \beta}$)
return $\mathcal{T}_r$

7 Discussion

Choosing an algorithm. The most preferred repair for an ontology with wrong axioms would lead to a more complete and less incorrect ontology than the original ontology. In general, however, this cannot be guaranteed unless we use a brute-force method that checks all axioms in an ontology. Although some optimizations are possible, this is in general not feasible. On the positive side, removing axioms does not introduce more incorrect knowledge and adding axioms does not remove correct knowledge. Unfortunately, removing wrong axioms may make the ontology less complete. For instance, when removing W from Mini-GALEN, the correct axiom PPr $\sqsubseteq$ NPr cannot be derived anymore. The weakening and completing alleviate this problem, but do not solve it completely. Adding correct axioms may make the ontology more incorrect in the case where some defects in the ontology were not yet detected or repaired and these lead to the derivation of new defects.

There is also a trade-off between using as much, but possibly wrong, knowledge as possible in the ontology and removing as much wrong knowledge as possible, when computing weakened and completed axiom sets (Fig. 3a). In the former case, more axioms (including more wrong axioms) are generated and need to be validated than in the latter case, but the final ontology in the former case may be more complete than in the latter case. For instance, Table 4 shows that sizes of the sup and sub sets for removing one axiom at a time are larger than or equal to the sizes of the sets for removing all at once (Algorithms C2 vs C4). When removing one at a time, the other wrong axioms can lead to more sub- and super-concepts and thus larger weakened axiom sets. This entails a higher validation effort by the domain expert, but it also leads to a more complete ontology as the axiom $PPr \sqsubseteq NPr$ is not always found by the approaches that

Algorithm C10 Weaken one at a time, remove all wrong, complete/add completed axiom sets one at a time

Input: TBox $\mathcal{T}$, Oracle Or, set of unwanted axioms W
Output: A repaired TBox

for each $\alpha \sqsubseteq \beta \in W$ **do**
  $\mathcal{T}_r \leftarrow$ Remove-axioms($\mathcal{T}$, $\{\alpha \sqsubseteq \beta\}$)
  $w_{\alpha \sqsubseteq \beta} \leftarrow$ weakened-axiom-set($\alpha \sqsubseteq \beta$, $\mathcal{T}_r, Or$)
end for
$\mathcal{T}_r \leftarrow$ Remove-axioms($\mathcal{T}$, W)
for each $\alpha \sqsubseteq \beta \in W$ **do**
  for each $sb \sqsubseteq sp \in w_{\alpha \sqsubseteq \beta}$ **do**
    $c_{sb \sqsubseteq sp} \leftarrow$ completed-axiom-set($sb \sqsubseteq sp$,$\mathcal{T}_r, Or$)
    $\mathcal{T}_r \leftarrow$ Add-axioms($\mathcal{T}_r$,$c_{sb \sqsubseteq sp}$)
  end for
end for
return $\mathcal{T}_r$

remove all at once. Another choice is to add new correct axioms as soon as they are found or wait until the end (Fig. 3b). In the former case they may be used to find additional information, but the end result may depend on the order that the axioms are handled. Also the order in which the axioms are processed, has an influence on the result as seen from Table 5 for Algorithm C4.

Similar observations can be made when completing is added to the removing and weakening (Fig. 3c). When the completed axioms are added one at a time during each iteration, the sizes of the sub and sup sets for each weakened axiom are larger than or equal to the sizes of the sets generated when adding them all at the end. In Table 6 we show this for Algorithms C9 and C10 that differ from each other in this aspect. Also here it entails a higher validation effort when adding one at a time, but it also leads to a more complete ontology.

In general, there is a trade-off between validation effort and the level of completeness and thus the choice of algorithm depends on the user's priority between these. For instance, earlier work on weakening discussed one combination strategy ((R-one, AB-none) with (W-one, U-now)) and did not show there were different options. In this work we show this trade-off. Further, by providing the Hasse diagrams we help deciding which features to use. Using features higher up in the diagrams means more validation work and more complete ontologies.

Domain expert validation in practice. Introducing new concepts may make it hard for a domain expert to validate the axioms [5]. In our implemented systems we alleviate this problem by using a naming convention that reflects the logical description of the new concepts as they would be in a non-normalized TBox. For instance, we use names such as 'S-SOME-Q' and 'Q-AND-R'. This convention also allows the nesting of operators. In future work we will investigate the technique of 'forgetting' (e.g., [34]) to further alleviate this problem.

In [5] an ontology consists of a static part considered to be correct and a refutable part. If we would follow this approach, then in our setting wrong

Table 2. Removing, weakening and completing - operations.

Operations	Description
R-all	Remove all the wrong axioms at once.
R-one	Remove the wrong axioms one at a time.
R-none	Remove nothing.
W-all	Weaken all wrong axioms at once.
W-one	Weaken the wrong axioms one at a time
C-all	Complete all weakened axioms at once.
C-one	Complete the weakened axioms one at a time.
AB-one	Add one wrong axiom back.
AB-all	Add all wrong axioms back.
AB-none	Add nothing back.
U-now	Update the changes immediately.
U-end_one	Update the changes after the iteration of each wrong axiom.
U-end_all	Update the changes after iterations of all wrong axioms.

Table 3. Ontologies

	Mini-GALEN	Pizza	EKAW	OFSMR	PACO	NCI
Concepts	9	74	100	159	224	3304
Roles	1	33	8	2	23	1
Axioms	20	341	801	1517	1153	30364

Table 4. Weakening for Mini-GALEN using C2 and C4. Three wrong axioms give 3 sup/sub-sets per algorithm.

	C2	C4
Sup(β,$\mathcal{T}$)	3 2 2	1 2 1
Sub(α,$\mathcal{T}$)	2 1 1	1 1 1
Weakened	PPr $\sqsubseteq$ NPr IPr $\sqsubseteq$ NPr	IPr $\sqsubseteq$ NPr

axioms in W can only be from the refutable part. Axioms from the static part do not need to be validated and should never be removed. Adding correct axioms should then grow the static part. We note that, in practise, it is not so clear how to divide an ontology in a static and a refutable part as, as mentioned before, according to our experience in assisting the development of ontologies in different domains, domain experts make mistakes even in the parts they think are correct.

8 Implemented systems

We implemented two systems (see supplemental material). As Protégé is a well-known ontology development tool, we implemented a plugin for repairing based on Algorithm C9. Using this algorithm the user can repair all wrong axioms at once. However, by iteratively invoking this plugin the user can also repair the

N_C = {GPr (GranulomaProcess), NPr (NonNormalProcess),
PPh (PathologicalPhenomenon), F(Fracture), E (Endocarditis),
IPr (InflammationProcess), PPr (PathologicalProcess),
C (Carditis), CVD (CardioVascularDisease)};
N_R = { hAPr (hasAssociatedProcess) }
$\mathcal{T}$ = { CVD $\sqsubseteq$ PPh, F $\sqsubseteq$ PPh, $\exists$hAPr.PPr $\sqsubseteq$ PPh, E $\sqsubseteq$ C,
E $\sqsubseteq$ $\exists$hAPr.IPr, GPr $\sqsubseteq$ NPr, PPr $\sqsubseteq$ IPr, IPr $\sqsubseteq$ GPr, E $\sqsubseteq$ PPr };
W = { E $\sqsubseteq$ PPr, PPr $\sqsubseteq$ IPr, IPr $\sqsubseteq$ GPr }
Or returns *true* for:
GPr $\sqsubseteq$ IPr, GPr $\sqsubseteq$ PPr, GPr $\sqsubseteq$ NPr, IPr $\sqsubseteq$ PPr, IPr $\sqsubseteq$ NPr,
PPr $\sqsubseteq$ NPr, CVD $\sqsubseteq$ PPh, F $\sqsubseteq$ PPh, E $\sqsubseteq$ PPh, E $\sqsubseteq$ C,
E $\sqsubseteq$ CVD, C $\sqsubseteq$ PPh, C $\sqsubseteq$ CVD, $\exists$hAPr.PPr $\sqsubseteq$ PPh,
$\exists$hAPr.IPr $\sqsubseteq$ PPh, E $\sqsubseteq$ $\exists$hAPr.IPr, E $\sqsubseteq$ $\exists$hAPr.PPh.
Note that for an oracle that does not make mistakes,
if Or(P $\sqsubseteq$ Q) = true, then also Or($\exists$r.P $\sqsubseteq$ $\exists$r.Q)=true and
Or(P $\sqcap$ O $\sqsubseteq$ Q)=true.
For other axioms P $\sqsubseteq$ Q with P, Q $\in N_C$, Or(P $\sqsubseteq$ Q) = false.

Fig. 2. Mini-GALEN. (Visualized in supplemental material.)

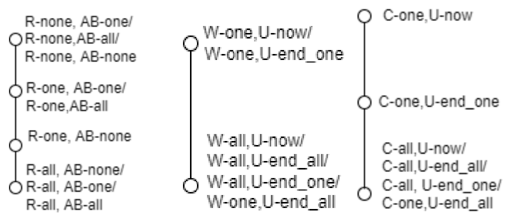
Table 5. Adding weakened axioms in different order for Mini-GALEN by C4. Wrong axioms: ①PPr$\sqsubseteq$IPr, ②IPr$\sqsubseteq$GPr, ③E$\sqsubseteq$PPr.

Wrong	①→②→③	①→③→②	②→①→③	②→③→①	③→①→②	③→②→①
Sup(β,$\mathcal{T}$) Sub(α,$\mathcal{T}$)	1 2 1 1 1 1	1 2 1 1 1 1	2 2 2 1 1 1	2 2 1 1 1 1	1 2 1 1 1 1	3 2 1 1 1 1
Weakened	IPr $\sqsubseteq$ NPr	IPr $\sqsubseteq$ NPr	IPr $\sqsubseteq$ NPr PPr $\sqsubseteq$ NPr	IPr $\sqsubseteq$ NPr PPr $\sqsubseteq$ NPr	IPr $\sqsubseteq$ NPr	IPr $\sqsubseteq$ NPr PPr $\sqsubseteq$ NPr

Table 6. Completing Mini-GALEN using C9 and C10.

	C9	C10
Sup(α,$\mathcal{T}$) Sub(β,$\mathcal{T}$)	1 1 2 2	1 1 2 3
Completed	PPr$\sqsubseteq$NPr, IPr$\sqsubseteq$NPr	PPr$\sqsubseteq$NPr, IPr$\sqsubseteq$PPr

wrong axioms one at a time. Further, we extended the $\mathcal{EL}$ version of the RepOSE system [35,19]. We allow the user to choose different combinations, thereby giving a choice in the trade-off between validation work and completeness. In the system, candidate weakened and completing axioms are shown in lists and also visualized using two sets of concepts. The axioms $\alpha \sqsubseteq \beta$ to be validated are the ones that can be constructed by choosing α from the first set and β from the second set. By showing them together, context of the solutions in the form of sub- and super-concepts is available. The domain expert can choose to validate such axioms by clicking in the different panes representing the sets of concepts.



(a) Removing (b) Weakening (c) Completing

Fig. 3. Hasse diagrams. (a) remove and add back wrong axioms; (b) weakening and update; (c) completing and update. Combinations of operations higher up in the lattices lead to more validation work and more complete ontologies.

9 Related work

We briefly discuss previous work on weakening and on completing. We are not aware of work that combines these.

Regarding *weakening*, previous work looks at the combination of debugging and weakening. Justifications for wrong axioms and a hitting set are computed. Then, instead of removing, weakened axioms are computed. In our approach we assume that the axioms to remove are given (e.g., by having computed a hitting set) and that when removing them they cannot be derived anymore. When this assumption is not made then, as pointed out in [5] (and ignored by older approaches) the weakening needs to be iterated to obtain a repair. We also note that none of the approaches explicitly state the use of a domain expert/oracle and they are purely logic-based. In practice, however, a domain expert/oracle is needed as otherwise axioms that are wrong in the domain of the ontology could be added. Regarding the *weakening algorithm*, in contrast to our approach, the other approaches work on non-normalized TBoxes. This means that they may find better solutions for the weakening, but the search space for solutions also becomes infinite. In [33] algorithms for weakening for $\mathcal{EL}$ and $\mathcal{ALC}$ are given with tractable and exponential complexity, respectively. They are based on refinement operators that are applied on the concepts of GCIs. The approach is extended in [7] for $\mathcal{SROIQ}$ TBoxes with an algorithm with almost-sure termination. Also in [9] an approach based on refinement operators is presented for $\mathcal{ALC}$. The nesting of operators is restricted based on the size of a concept. In [5] the right-hand side of axioms is generalized, but the left-hand side is not specialized to obtain a well-founded weakening relation (i.e., there is no infinite chain of weakenings). Essentially, our use of $sup(P, \mathcal{T})$ and $sub(P, \mathcal{T})$ in the weakening is a similar approach. As we have restricted the $sup(P, \mathcal{T})$ and $sub(P, \mathcal{T})$ to contain only concepts in SCC($\mathcal{T}$), we only have a finite number of possible axioms. Regarding the *strategy to combine removing with weakening*, in all these other approaches usually one-at-a-time removing (R-one, AB-none) and weakening (W-one, U-now) is used. From our Hasse diagrams we can see that this means the most complete ontologies and most validation work for weakening, but neither the most nor the least complete ontologies for removing. We note that using our Hasse diagrams, new variants of these other approaches can be created with another trade-off involving correctness and completeness. Further, the issue of the influence of the order is not addressed. In [17] parts of axioms to remove are pinpointed and harmful and helpful changes are defined.

Regarding *completing*, previous work with validation by a domain expert (e.g., [35] for the $\mathcal{EL}$ family, [18] for $\mathcal{ALC}$) allowed only axioms of the form $P \sqsubseteq Q$ where P and Q are atomic concepts in the completed axioms set while Algorithm 2 allows P and Q to be in SCC($\mathcal{T}$) (and then normalizes). That work used one particular combination strategy, i.e., (C-one, U-end-all). A non-interactive solution that is independent of the constructors of the description logic is proposed in [10]. This approach introduces justification patterns that can be instantiated with existing concepts or new concepts.

10 Conclusion

In this paper we proposed an interactive approach using weakening and completing to mitigate the negative effects of removing wrong axioms in $\mathcal{EL}$ ontologies. We presented a framework (and the first approach) for combining removing with weakening and completing. We showed that there are different combination strategies and that there is a trade-off involving correctness and completeness. We also introduced a way to compare combination strategies and showed that earlier work covered one type of combination strategy. Further, we presented new algorithms for weakening and completion and using these, showed the influence of 13 combination strategies on the completeness for 6 ontologies in experiments.

For future work we will investigate integrating the full debugging with weakening and completing. It is clear that, when for debugging we also add the step of finding which axioms to remove, that also this new step can be combined with removing, weakening and completing and thus leads to new combination strategies with different trade-offs. Further, for completing, we will look into other strategies for reducing the search space while still maintaining a practically feasible validation work for the domain expert. It is also interesting to investigate the problem for more expressive description logics.

Supplemental Material Statement. More details regarding the other algorithms, the full results of the experiments, diagram derivation and a short discussion on the difference of the combination strategies can be found in the supplemental material, which is available at `https://www.ida.liu.se/~patla00/publications/ESWC2023/`.

Acknowledgements: We thank Olaf Hartig for discussions leading to the Hasse diagrams. This work is financially supported by the Swedish e-Science Research Centre (SeRC) and the Swedish Research Council (Vetenskapsrådet, dnr 2018-04147).

References

1. Arif, M.F., Mencía, C., Ignatiev, A., Manthey, N., Peñaloza, R., Marques-Silva, J.: BEACON: An Efficient SAT-Based Tool for Debugging $\mathcal{E}L+$ Ontologies. In: Theory and Applications of Satisfiability Testing. LNCS, vol. 9710, pp. 521–530 (2016). https://doi.org/10.1007/978-3-319-40970-2_32
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{E}L$ Envelope. In: IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence. pp. 364–369 (2005)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
4. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In: IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence. p. 230–235 (2007)
5. Baader, F., Kriegel, F., Nuradiansyah, A., Peñaloza, R.: Making repairs in description logics more gentle. In: KR 2018: 16th International Conference on Principles of Knowledge Representation and Reasoning. pp. 319–328 (2018)
6. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press (2005)
7. Confalonieri, R., Galliani, P., Kutz, O., Porello, D., Righetti, G., Troquard, N.: Towards even more irresistible axiom weakening. In: Proceedings of the 33rd International Workshop on Description Logics. CEUR, vol. 2663 (2020)
8. Dragisic, Z., Ivanova, V., Li, H., Lambrix, P.: Experiences from the anatomy track in the ontology alignment evaluation initiative. Journal of Biomedical Semantics **8**(1), 56:1–56:28 (2017). https://doi.org/10.1186/s13326-017-0166-5
9. Du, J., Qi, G., Fu, X.: A practical fine-grained approach to resolving incoherent OWL 2 DL terminologies. In: CIKM '14: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 919–928 (2014). https://doi.org/10.1145/2661829.2662046
10. Du, J., Wan, H., Ma, H.: Practical TBox abduction based on justification patterns. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI17). pp. 1100–1106 (2017)
11. Ferré, S., Rudolph, S.: Advocatus diaboli – exploratory enrichment of ontologies with negative constraints. In: EKAW: Knowledge Engineering and Knowledge Management. LNAI, vol. 7603, pp. 42–56 (2012). https://doi.org/10.1007/978-3-642-33876-2$_7$
12. Fleischhacker, D., Meilicke, C., Völker, J., Niepert, M.: Computing incoherence explanations for learned ontologies. In: Web Reasoning and Rule Systems - RR13. LNCS, vol. 7994, pp. 80–94 (2013). https://doi.org/10.1007/978-3-642-39666-3_7
13. Ji, Q., Gao, Z., Huang, Z., Zhu, M.: An efficient approach to debugging ontologies based on patterns. In: JIST 2011: The Semantic Web. pp. 425–433 (2011). https://doi.org/10.1007/978-3-642-29923-0_33
14. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding All Justifications of OWL DL Entailments. In: ISWC 2007, ASWC 2007: The Semantic Web. pp. 267–280 (2007). https://doi.org/10.1007/978-3-540-76298-0_20
15. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing Unsatisfiable Concepts in OWL Ontologies. In: ESWC 2006: The Semantic Web: Research and Applications. pp. 170–184 (2006). https://doi.org/10.1007/11762256_15

16. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging Unsatisfiable Classes in OWL Ontologies. Journal of Web Semantics **3**(4), 268–293 (2005). https://doi.org/10.1016/j.websem.2005.09.005
17. Lam, J.S.C., Sleeman, D.H., Pan, J.Z., Vasconcelos, W.W.: A fine-grained approach to resolving unsatisfiable ontologies. Journal on Data Semantics **10**, 62–95 (2008). https://doi.org/10.1007/978-3-540-77688-8_3
18. Lambrix, P., Dragisic, Z., Ivanova, V.: Get My Pizza Right: Repairing Missing is-a Relations in $\mathcal{ALC}$ Ontologies. In: JIST 2012: Semantic Technology. pp. 17–32 (2012). https://doi.org/10.1007/978-3-642-37996-3_2
19. Lambrix, P., Wei-Kleiner, F., Dragisic, Z.: Completing the is-a structure in light-weight ontologies. Journal of Biomedical Semantics **6** (2015). https://doi.org/10.1186/s13326-015-0002-8
20. Lehmann, J., Bühmann, L.: ORE - A tool for repairing and enriching knowledge bases. In: ISWC 2010: The Semantic Web. LNCS, vol. 6497, pp. 177–193 (2010). https://doi.org/10.1007/978-3-642-17749-1_12
21. Li, H., Armiento, R., Lambrix, P.: A method for extending ontologies with application to the materials science domain. Data Science Journal **18**(1) (2019). https://doi.org/10.5334/dsj-2019-050
22. Meyer, T., Lee, K., Booth, R., Pan, J.: Finding maximally satisfiable terminologies for the description logic ALC. In: Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference (AAAI06). pp. 269–274 (2006)
23. Moodley, K., Meyer, T., Varzinczak, I.J.: Root justifications for ontology repair. In: RR 2011: Web Reasoning and Rule Systems. LNCS, vol. 6902, pp. 275–280 (2011). https://doi.org/10.1007/978-3-642-23580-1_24
24. Nikitina, N., Rudolph, S., Glimm, B.: Interactive ontology revision. Journal of Web Semantics **12-13**, 118–130 (2012). https://doi.org/https://doi.org/10.1016/j.websem.2011.12.002
25. Pesquita, C., Faria, D., Santos, E., Couto, F.M.: To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In: Proceedings of the 8th International Workshop on Ontology Matching. CEUR, vol. 1111, pp. 13–24 (2013)
26. Rodler, P., Schmid, W.: On the impact and proper use of heuristics in test-driven ontology debugging. In: RuleML+RR 2018: Rules and Reasoning. LNCS, vol. 11092, pp. 164–184 (2018). https://doi.org/10.1007/978-3-319-99906-7_11
27. Schekotihin, K., Rodler, P., Schmid, W.: OntoDebug: Interactive Ontology Debugging Plug-in for Protégé. In: Ferrarotti, F., Woltran, S. (eds.) FoIKS 2018: Foundations of Information and Knowledge Systems. LNCS, vol. 10833, pp. 340–359 (2018). https://doi.org/10.1007/978-3-319-90050-6_19
28. Schlobach, S.: Debugging and Semantic Clarification by Pinpointing. In: ESWC 2005: The Semantic Web: Research and Applications. pp. 226–240 (2005). https://doi.org/10.1007/11431053_16
29. Schlobach, S., Cornet, R.: Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In: IJCAI. pp. 355–360 (2003)
30. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging Incoherent Terminologies. Journal of Automated Reasoning **39**(3), 317–349 (2007). https://doi.org/10.1007/s10817-007-9076-z
31. Sertkaya, B.: Ontocomp: A protégé plugin for completing owl ontologies. In: ISWC 2009: The Semantic Web: Research and Applications. LNCS, vol. 5554, pp. 898–902 (2009). https://doi.org/10.1007/978-3-642-02121-3$_7$8

32. Shchekotykhin, K.M., Friedrich, G., Fleiss, P., Rodler, P.: Interactive ontology debugging: Two query strategies for efficient fault localization. Journal of Web Semantics **12**, 88–103 (2012). https://doi.org/10.1016/j.websem.2011.12.006
33. Troquard, N., Confalonieri, R., Galliani, P., Peñaloza, R., Porello, D., Kutz, O.: Repairing ontologies via axiom weakening. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI18). pp. 1981–1988 (2018). https://doi.org/10.1609/aaai.v32i1.11567
34. Wang, K., Wang, Z., Topor, R.W., Pan, J.Z., Antoniou, G.: Concept and role forgetting in $\mathcal{ALC}$ ontologies. In: ISWC 2009: The Semantic Web. pp. 666–681 (2009). https://doi.org/10.1007/978-3-642-04930-$9_4$2
35. Wei-Kleiner, F., Dragisic, Z., Lambrix, P.: Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI14. pp. 1120–1127 (2014). https://doi.org/10.1609/aaai.v28i1.8858