Repairing $\mathcal{EL}_\bot$ Ontologies using Debugging, Weakening and Completing (Extended Abstract)

Ying Li[1], Patrick Lambrix[1]

[1] *Linköping University and The Swedish e-Science Research Centre, Linköping, 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 $\mathcal{EL}_\bot$ ontologies by combining debugging with axiom weakening and completing. We show different combination strategies, discuss the influence on the final ontologies and show experimental results. We show that there is a trade-off between the amount of validation work for a domain expert and the quality of the ontology in terms of correctness and completeness.

Keywords

Ontology engineering, Ontology debugging, Ontology repair

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., [1]). The workflow consists of several steps including the detection and localization of the defects and the repairing. 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., [2]). 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., [3, 4, 5, 6]). Another approach is to complete an ontology (e.g., [7]) which adds previously unknown correct axioms that allow to derive existing axioms, and that could be used on the results of weakening.

DL 2023: 36th International Workshop on Description Logics, September 2–4, 2023, Rhodes, Greece

ying.li@liu.se (Y. Li); patrick.lambrix@liu.se (P. Lambrix)

0000-0002-1367-9679 (Y. Li); 0000-0002-9084-0470 (P. Lambrix)

In [8] we proposed a framework for repairing $\mathcal{EL}$ ontologies, where, given a set of wrong asserted axioms, these axioms are removed but the effects of removing are mitigated by using weakening and completing to add (back) correct knowledge. That work is the first that combines these operations and it was shown that combining these in different ways has an influence on the amount of validation work by a domain expert and the completeness of the final ontology. It was also shown that earlier work on weakening only considered one of the possible combinations. In this extended abstract of [9], we extend the framework to include full debugging. In this case a set of (not necessarily asserted) wrong axioms is given and in the final repaired ontology these should not be derivable. The weakening and completing mitigate the effect of removing axioms. While the approach is general, our experiments and proofs are currently for $\mathcal{EL}_\bot$ ontologies.

2. Preliminaries

Problem formulation. Assume that the ontology to be repaired is represented by a TBox $\mathcal{T}$, the axioms to be removed are collected in set W and a domain expert (or a team of experts) is represented by oracle Or. Then the repairing problem can be formulated as follows.

Definition 1. *(Repair)*[1] *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 tuple (A, D) where A and D are a finite sets of TBox axioms such that (i) $\forall\ \psi \in A$: $Or(\psi)$ = true; (ii) D is a finite set of asserted axioms in $\mathcal{T}$; (iii) $\forall\ \psi \in D$: $Or(\psi)$ = false; (iv) $\forall\ \psi \in W$: $(\mathcal{T} \cup A) \setminus D \not\models \psi$

Further, 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 use the preference relations *less incorrect* and *more complete* between ontologies that formalize these intuitions [10].

Definition 2. *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 $\mathcal{O}_2$ iff* $(\forall\psi : (\mathcal{T}_1 \models \psi \wedge Or(\psi) = false) \rightarrow \mathcal{T}_2 \models \psi)) \wedge (\exists\psi : Or(\psi) = false\ \wedge \mathcal{T}_1 \not\models \psi \wedge \mathcal{T}_2 \models \psi)$. *Further, $\mathcal{O}_1$ is* more complete *than $\mathcal{O}_2$ iff* $(\forall\psi : (\mathcal{T}_2 \models \psi \wedge\ Or(\psi) = true) \rightarrow \mathcal{T}_1 \models \psi)) \wedge (\exists\psi : Or(\psi) = true\ \wedge \mathcal{T}_1 \models \psi \wedge \mathcal{T}_2 \not\models \psi)$.

Basic operations. Given a set of wrong axioms W, *debugging* aims to find a set of wrong asserted axioms D that when the axioms are removed from the ontologies, the axioms in W can not be derived anymore. Many approaches have been proposed (e.g., [11, 12, 13, 14, 1, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]). A basic approach is based on the computation of justifications for the wrong axioms and then computing a Hitting set over the justifications, where a justification for axiom ψ in $\mathcal{T}$ is a set of axioms $\mathcal{T}' \subseteq \mathcal{T}$ if $\mathcal{T}' \models \psi$ and $\forall \mathcal{T}'' \subsetneq \mathcal{T}' : \mathcal{T}'' \not\models \psi$. In this paper we call *Removing* the operation of deleting an asserted axiom from the TBox. 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 the repairing this means that a wrong axiom $\alpha \sqsubseteq \beta$ can be replaced by a correct weaker axiom $sb \sqsubseteq sp$ such that sb is a sub-concept of α and sp

[1]This is a simplified version of the definition of repair in [10] where, in addition to a set of wrong axioms to remove, also a set of correct axioms to add is given.

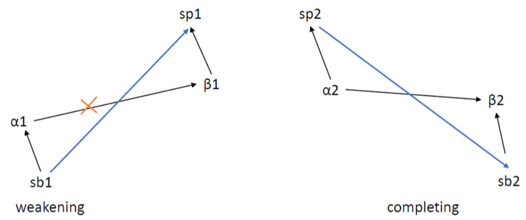
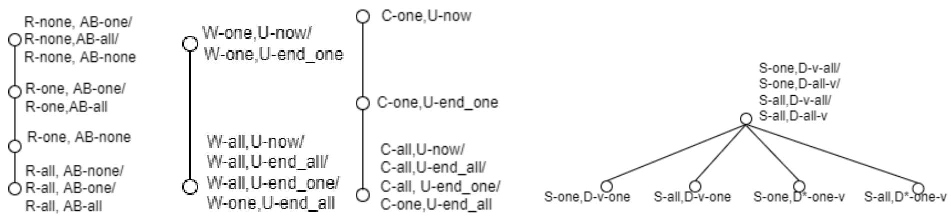


Figure 1: Weakening and Completing

is a super-concept of β, thereby mitigating the effect of removing the wrong axiom (Fig. 1).[2] *Completing* aims to find correct axioms that are not derivable from the ontology yet and that would make a given axiom derivable. For a given axiom $\alpha \sqsubseteq \beta$, it finds correct axioms $sp \sqsubseteq sb$ such that sp is a super-concept of α and sb is a sub-concept of β (Fig. 1). This means that if $sp \sqsubseteq sb$ is added to $\mathcal{T}$, then $\alpha \sqsubseteq \beta$ would be derivable. Note that weakening and completing are dual operations where the former finds weaker axioms and the latter stronger axioms.

3. Combination strategies

Removing, weakening and completing. In [8] we introduced different ways to combine removing (R), weakening (W) and completing (C). These different ways take into account the different choices that can be made in terms of, e.g., whether axioms are dealt with one at a time or all at once (one, all), whether wrong axioms can be used to find solutions (AB-none, AB-one, AB-all) and when the TBox is updated (U-now, U-end_one, U-end_all). The different combinations were classified in Hasse diagrams (Fig. 2(a-c)) where combinations higher up in the diagram lead to more validation work for the domain expert, but also more complete ontologies, which is the aim of weakening and completing.



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

Figure 2: Hasse diagrams (a-c from [8]). (a) remove and add back wrong axioms; (b) weakening and update; (c) completing and update; (d) selecting and debugging

Adding debugging. During debugging justifications are generated. There are different choices to be made regarding generating the justifications of all wrong axioms at once or one at a time, validating all the axioms in the justifications or just a valid Hitting set, and when to validate. Therefore, we introduce different combination operators. *S-one* and *S-all* represent the

[2]Different ranges can be used for sb and sp, e.g., concept names in the TBox or restricted concept definitions.

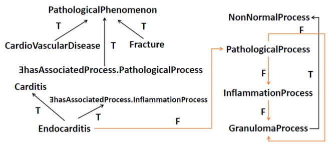


Figure 3: Mini-GALEN. The domain expert's knowledge about the subsumption axioms is marked with T (true) or F (false).

	D-v-all	D-v-one (1)	D-v-one (2)	D-v-one (3)
Asserted Wrong Axioms	IPr $\sqsubseteq$ GPr PPr $\sqsubseteq$ IPr PPr $\sqsubseteq$ GPr E $\sqsubseteq$ PPr	IPr $\sqsubseteq$ GPr PPr $\sqsubseteq$ IPr PPr $\sqsubseteq$ GPr	PPr $\sqsubseteq$ IPr PPr $\sqsubseteq$ GPr E $\sqsubseteq$ PPr	IPr $\sqsubseteq$ GPr PPr $\sqsubseteq$ GPr E $\sqsubseteq$ PPr

Table 1: Debugging for Mini-GALEN. Concept names from Fig 3 are abbreviated. W = { PPr $\sqsubseteq$ GPr, E $\sqsubseteq$ GPr }. For D-v-one there are different solutions based on the chosen Hitting set.

choice to calculate the justifications for one wrong axiom at the time or for all at once. The operations with name starting with *D* concern choices regarding using the justifications to generate asserted wrong axioms to remove from the ontology. The *one/all* choice concerns whether to validate one Hitting set or all axioms in the justifications. The validation of these axioms can be done during (such that we always have a Hitting set with wrong asserted axioms) or after (which may lead to not repairing) the generation of the Hitting set, represented by the order of *v* (for validation) and *one* (while for the *all* the order does not matter).

Also these combinations can be represented in a Hasse diagram (Fig. 2(d))[3]. Combinations higher up in the diagram lead to more validation work for the domain expert, but also to less incorrect ontologies, which is the aim of debugging. The Hasse diagrams can be used to compare different combination strategies. For instance, assume two algorithms that perform the same weakening and completing operations, but the first uses *S-all,D-v-all* and the second uses *S-one,D-v-one*, then by using the Hasse diagram in Fig. 2(d), we know that the ontology repaired by the first algorithm is less incorrect than the ontology repaired by the second algorithm.

4. Experiments

In order to compare the use of the different combinations of strategies, we run experiments on several ontologies as in [8]: Mini-GALEN, PACO, NCI, OFSMR, EKAW and Pizza ontology. 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).

As an example of the influence of different strategies, we show in Table 1 how the choice of selecting wrong axioms, when to compute Hitting sets and when to validate axioms, influences which asserted axioms are retained for weakening and completing. For more examples and full experimental results including weakening and completing, we refer to [9].

Acknowledgments

This work is financially supported by the Swedish e-Science Research Centre (SeRC).

[3]In the Hasse diagram (Fig. 2(d)), D* means the final output is a valid Hitting set.

References

[1] A. Kalyanpur, B. Parsia, E. Sirin, J. Hendler, Debugging Unsatisfiable Classes in OWL Ontologies, J Web Semantics 3 (2005) 268–293. doi:10.1016/j.websem.2005.09.005.

[2] C. Pesquita, D. Faria, E. Santos, F. M. Couto, To repair or not to repair: reconciling correctness and coherence in ontology reference alignments, in: 8th International Workshop on Ontology Matching, volume 1111 of *CEUR*, 2013, pp. 13–24.

[3] J. S. C. Lam, D. H. Sleeman, J. Z. Pan, W. W. Vasconcelos, A fine-grained approach to resolving unsatisfiable ontologies, J Data Semantics 10 (2008) 62–95. doi:10.1007/978-3-540-77688-8_3.

[4] J. Du, G. Qi, X. Fu, A practical fine-grained approach to resolving incoherent OWL 2 DL terminologies, in: 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 919–928. doi:10.1145/2661829.2662046.

[5] F. Baader, F. Kriegel, A. Nuradiansyah, R. Peñaloza, Making repairs in description logics more gentle, in: 16th International Conference on Principles of Knowledge Representation and Reasoning, 2018, pp. 319–328.

[6] N. Troquard, R. Confalonieri, P. Galliani, R. Peñaloza, D. Porello, O. Kutz, Repairing ontologies via axiom weakening, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 1981–1988.

[7] F. Wei-Kleiner, Z. Dragisic, P. Lambrix, Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms, in: Twenty-Eighth AAAI Conference On Artificial Intelligence, 2014, pp. 1120–1127.

[8] Y. Li, P. Lambrix, Repairing $\mathcal{EL}$ ontologies using weakening and completing, in: The Semantic Web - 20th International Conference, ESWC 2023, volume 13870 of *LNCS*, 2023, pp. 298–315. doi:10.1007/978-3-031-33455-9_18.

[9] Y. Li, P. Lambrix, Repairing $\mathcal{EL}_\bot$ ontologies using debugging, weakening and completing, In preparation (2023).

[10] P. Lambrix, Completing and debugging ontologies: State of the art and challenges in repairing ontologies, ACM Journal of Data and Information Quality (2023). doi:10.1145/3597304.

[11] S. Schlobach, R. Cornet, Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies, in: Eighteenth International Joint Conference on Artificial Intelligence, 2003, pp. 355–360.

[12] S. Schlobach, Debugging and Semantic Clarification by Pinpointing, in: The Semantic Web: Research and Applications. ESWC 2005, volume 3532 of *LNCS*, 2005, pp. 226–240. doi:10.1007/11431053_16.

[13] S. Schlobach, Z. Huang, R. Cornet, F. van Harmelen, Debugging Incoherent Terminologies, J Automated Reasoning 39 (2007) 317–349. doi:10.1007/s10817-007-9076-z.

[14] T. Meyer, K. Lee, R. Booth, J. Pan, Finding maximally satisfiable terminologies for the description logic $\mathcal{ALC}$, in: Twenty-First AAAI Conference on Artificial Intelligence, 2006, pp. 269–274.

[15] A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca-Grau, Repairing Unsatisfiable Concepts in OWL Ontologies, in: The Semantic Web: Research and Applications. ESWC 2006, volume 4011 of *LNCS*, 2006, pp. 170–184. doi:10.1007/11762256_15.

[16] A. Kalyanpur, B. Parsia, M. Horridge, E. Sirin, Finding All Justifications of OWL DL Entailments, in: The Semantic Web. ISWC ASWC 2007, volume 4825 of *LNCS*, 2007, pp. 267–280. doi:10.1007/978-3-540-76298-0_20.

[17] J. Lehmann, L. Bühmann, ORE - A tool for repairing and enriching knowledge bases, in: The Semantic Web – ISWC 2010, volume 6497 of *LNCS*, 2010, pp. 177–193. doi:10.1007/978-3-642-17749-1_12.

[18] Q. Ji, Z. Gao, Z. Huang, M. Zhu, An efficient approach to debugging ontologies based on patterns, in: The Semantic Web. JIST 2011, volume 7185 of *LNCS*, 2011. doi:10.1007/978-3-642-29923-0_33.

[19] K. Moodley, T. Meyer, I. J. Varzinczak, Root justifications for ontology repair, in: Web Reasoning and Rule Systems. RR 2011, volume 6902 of *LNCS*, 2011, pp. 275–280. doi:10.1007/978-3-642-23580-1_24.

[20] K. M. Shchekotykhin, G. Friedrich, P. Fleiss, P. Rodler, Interactive ontology debugging: Two query strategies for efficient fault localization, J Web Semantics 12 (2012) 88–103. doi:10.1016/j.websem.2011.12.006.

[21] D. Fleischhacker, C. Meilicke, J. Völker, M. Niepert, Computing incoherence explanations for learned ontologies, in: Web Reasoning and Rule Systems. RR 2013, volume 7994 of *LNCS*, 2013, pp. 80–94. doi:10.1007/978-3-642-39666-3_7.

[22] M. F. Arif, C. Mencía, A. Ignatiev, N. Manthey, R. Peñaloza, J. Marques-Silva, BEACON: An Efficient SAT-Based Tool for Debugging $\mathcal{EL}$+ Ontologies, in: Theory and Applications of Satisfiability Testing – SAT 2016, volume 9710 of *LNCS*, 2016, pp. 521–530. doi:10.1007/978-3-319-40970-2_32.

[23] P. Rodler, W. Schmid, On the impact and proper use of heuristics in test-driven ontology debugging, in: Rules and Reasoning. RuleML+RR 2018, volume 11092 of *LNCS*, 2018, pp. 164–184. doi:10.1007/978-3-319-99906-7_11.

[24] K. Schekotihin, P. Rodler, W. Schmid, OntoDebug: Interactive Ontology Debugging Plug-in for Protégé, in: F. Ferrarotti, S. Woltran (Eds.), Foundations of Information and Knowledge Systems. FoIKS 2018, volume 10833 of *LNCS*, 2018, pp. 340–359. doi:10.1007/978-3-319-90050-6_19.