Position paper: Issues in Logic-based Repairing of Knowledge Graphs

Patrick Lambrix[1,2], Ying Li[1,2]

[1]Department of Computer and Information Science, Linköping University, Linköping, Sweden
[2]Swedish e-Science Research Centre, Linköping, Sweden

Abstract

In this paper we define a problem of logic-based repairing of knowledge graphs and discuss different issues related to correctness and completeness that need to be solved for creating repairing systems. We summarize and extend recent ideas for solving these issues.

Keywords

Knowledge graph quality, Knowledge graph repair

1. Introduction

Knowledge graphs (KGs) are core components in application scenarios that involve searching, integrating, managing and extracting value from diverse sources of data at large scale. They are used by major data and database providers and consumers such as Google, Amazon, Meta, and Neo4j. The quality of KGs is crucial for developing high-quality semantically enabled applications. However, ensuring their quality, in particular regarding completeness (or coverage, all relevant information is modeled) and correctness (no incorrect information is modeled), is a major challenge [1, 2, 3, 4, 5].

There are different types of defects concerning completeness and correctness in KGs, including semantic defects such as inconsistency or incoherence, and modeling defects where axioms in the KG are incorrectly stated or inappropriately reflect the intended semantics of the domain. A process of dealing with defects has different steps including detection, localization and repair. Many approaches exist for the first two steps [1]. For instance, semantic defects are usually detected by standard reasoning techniques. Inspection and competency questions can be used to find modeling defects. This paper focuses on the repairing step. In contrast to traditional debugging approaches, we discuss ways to repair KGs with incorrect information such that as much correct information as possible is maintained. Further, we discuss ways to add missing correct knowledge that are more advanced than just adding this knowledge. These approaches improve the quality of KGs in terms of correctness and completeness.

In this paper, we address the problem of repairing KGs represented by description logic knowledge bases. As our discussion does not look into particular implementations of repairing systems, but rather provides a framework that can be instantiated with different algorithms, the discussion is agnostic with respect to which description logic is used for the representation of the KG.

2. Preliminaries

Description logics [6] are knowledge representation languages where concept descriptions are constructed inductively from a set N_C of concept names, a set N_R of role names and 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 concept name $C \in N_C$ a subset $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to each role name $r \in N_R$ a relation

QKG - Workshop on Quality of Knowledge Graphs
patrick.lambrix@liu.se (P. Lambrix); ying.li@liu.se (Y. Li)
0000-0002-9084-0470 (P. Lambrix); 0000-0002-1367-9679 (Y. Li)

$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each individual name $i \in N_I$ an element $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is straightforwardly extended to complex concepts.

A knowledge base is a tuple $\langle T, A \rangle$ where T is a TBox and A is an ABox. A TBox is a finite set of axioms which include general concept inclusions (GCIs) and role inclusions (RIs). An ABox is a finite set of facts, which consists of concept membership assertions (CMAs) and role membership assertions (RMAs). An interpretation I is a model of a knowledge base K if for each axiom in K the semantic conditions are satisfied. We say that a knowledge base K is *inconsistent* if there is no model for K. Further, a concept C in a TBox $\mathcal{T}$ is *unsatisfiable* if for all models $\mathcal{I}$ of $\mathcal{T}$: $C^{\mathcal{I}} = \emptyset$. A TBox is *incoherent* if it contains an unsatisfiable concept name.

In general, a KG may contain ontologies represented by description logic TBoxes, alignments (i.e., sets of mappings) between ontologies represented by TBoxes, and facts regarding data represented by ABoxes. Similar to KGs and knowledge bases, in the remainder, we use the terms ontology and alignment also for their representations as TBoxes.

Definition 1. *Let $\mathcal{T}_1$, ..., $\mathcal{T}_n$ be TBoxes representing ontologies. For $i, j \in [1..n]$ with $i < j$, let $\mathcal{A}_{ij}$ be a TBox representing an alignment between the ontologies represented by $\mathcal{T}_i$ and $\mathcal{T}_j$. Let $\mathcal{A}_1$, ..., $\mathcal{A}_k$ be ABoxes representing facts regarding data. Then, the KG containing the ontologies, their alignments, and the data is represented by the knowledge base $(\bigcup_{i=1..n} \mathcal{T}_i) \cup (\bigcup_{i,j=1..n,i<j} \mathcal{A}_{ij}) \cup (\bigcup_{i=1..k} \mathcal{A}_i)$.*

Note that this definition also covers the simpler cases of an ontology (one ontology, no alignments and no data), an ontology network (no data), and a single ontology with data (no alignments). In the remainder we use the term 'KG' for the KG and the knowledge base representing the KG interchangeably.

In Sect. 4 we deal with repairing KGs in general, while in Sect. 5 we make use of the fact that axioms in KGs can be of different types.

3. Repairing KGs

3.1. KG quality

In this paper we discuss KG quality in terms of correctness and completeness. We assume an oracle (representing a domain expert or a set of doamin experts) that, when given an axiom, can answer whether this axiom is correct or not in the domain of interest. As deciding whether a KG is correct or complete is not feasible in practice, we instead use preference relations to compare KGs with respect to correctness and completeness. Essentially, KG1 is more complete/incorrect than KG2, if all correct/incorrect information in KG2 can be derived in KG1, but there is additional correct/incorrect information in KG1 that is not derivable in KG2.[1]

Definition 2. *(less incorrect) Let $\mathcal{KG}_1$ and $\mathcal{KG}_2$ be two KGs. Then, we say that $\mathcal{KG}_1$ is* less incorrect *than $\mathcal{KG}_2$ ($\mathcal{KG}_2$ is* more incorrect *than $\mathcal{KG}_1$) iff $(\forall \psi : (\mathcal{KG}_1 \models \psi \wedge Or(\psi) = false) \rightarrow \mathcal{KG}_2 \models \psi)) \wedge (\exists \psi : Or(\psi) = false \wedge \mathcal{KG}_1 \not\models \psi \wedge \mathcal{KG}_2 \models \psi)$.*

Definition 3. *(more complete) Let $\mathcal{KG}_1$ and $\mathcal{KG}_2$ be two KGs. Then, we say that $\mathcal{KG}_1$ is* more complete *than $\mathcal{KG}_2$ (or $\mathcal{KG}_2$ is* less complete *than $\mathcal{KG}_1$) iff $(\forall \psi : (\mathcal{KG}_2 \models \psi \wedge Or(\psi) = true) \rightarrow \mathcal{KG}_1 \models \psi)) \wedge (\exists \psi : Or(\psi) = true \wedge \mathcal{KG}_1 \models \psi \wedge \mathcal{KG}_2 \not\models \psi)$.*

3.2. Repair

We assume that, for a given KG, after the detection and localization steps, we are given two sets of axioms. The first set W is a set of incorrect axioms derivable[2] from the KG. The second set M is a set of (missing) correct axioms that are not derivable from the KG.

[1] Based on [5].

[2] Note that W can contain asserted axioms, i.e., axioms explicitly stated in the KG, as they are trivially derivable from the KG.

For the axioms in W, we need to find the *asserted* axioms in the KG that cause the derivation of the axioms in W and remove the incorrect asserted axioms, such that the axioms in W cannot be derived anymore. We may also want to add correct axioms to make the KG more complete. For the axioms in M, we need to find correct axioms that when added to the KG, would make the axioms in M derivable. It is well-known that for obtaining high-quality repairs, domain expert involvement is needed.

Thus, a repair for $\mathcal{KG}$, given the oracle Or, a set of incorrect axioms W[3] and a set of correct axioms M, is a set of correct axioms A and a set of asserted incorrect axioms W_a such that when the axioms in W_a are removed and the axioms in A are added to $\mathcal{KG}$, then the axioms in W cannot be derived anymore and the axioms in M are derivable.

Definition 4. *(Repair) Let $\mathcal{KG}$ be a knowledge graph. Let Or be an oracle that, given an axiom, returns true or false. Let W be a finite set of axioms derivable from $\mathcal{KG}$ such that $\forall\ \psi \in W$: $Or(\psi)$ = false. Let M be a finite set of axioms such that $\forall\ \psi \in M$: $Or(\psi)$ = true. Then, a repair for $(\mathcal{KG}, Or, W, M)$ is a tuple (A, W_a) where A and W_a are finite sets of axioms such that*
(i) $\forall\ \psi \in A$: $Or(\psi)$ = true;
(ii) W_a is a finite set of asserted axioms in $\mathcal{KG}$;
(iii) $\forall\ \psi \in W_a$: $Or(\psi)$ = false;
(iv) $\forall\ \psi \in W$: $(\mathcal{KG} \cup A) \setminus W_a \not\models \psi$;
(v) $\forall\ \psi \in M$: $(\mathcal{KG} \cup A) \setminus W_a \models \psi$.

3.3. Repair quality

To compare the quality of repairs of a given KG, we can make use of Defs 2 and 3.

Definition 5. *(more complete and less incorrect repairs) Let (A_1, W_{a1}) and (A_2, W_{a2}) be two repairs for $(\mathcal{KG}, Or, W, M)$. Let $\mathcal{KG}_1$ = $((\mathcal{KG} \cup A_1) \setminus W_{a1})$ and $\mathcal{KG}_2$ = $((\mathcal{KG} \cup A_2) \setminus W_{a2})$. Then, repair (A_1, W_{a1}) is* more complete *than repair (A_2, W_{a2}) iff $\mathcal{KG}_1$ is* more complete *than $\mathcal{KG}_2$. Further, (A_1, W_{a1}) is* less incorrect *than repair (A_2, W_{a2}) iff $\mathcal{KG}_1$ is* less incorrect *than $\mathcal{KG}_2$.*

3.4. Oracle quality

As described in [5], there are different interesting cases for Or. The *all-knowing oracle* always answers correctly. This is the ideal case guaranteeing that repairs always add correct information and remove incorrect information. However, this requires that domain experts can do reasoning with long chains of derivations and is not achievable in general. For the *limited all-knowing oracle*, if Or answers, then the answer is correct, but it may not know the answer to all questions. This case represents a domain expert who knows a part of the domain well. An approximation of this case is when there are several domain experts and only if all domain experts give the same answer regarding the correctness of an axiom, the answer is considered. The most common case is that the oracle makes mistakes. Axioms that are not correct in the domain may be validated as correct and vice versa. Although this does not give the same guarantees as the previous cases, oracles with error rates can still be beneficial to the systems (e.g., error rates up to 30% for ontology alignment systems [7]). There is also the situation where no domain expert is available and there is no validation of axioms, such as in fully automated systems.

In practice, when using domain experts, it is not possible to know which kind of domain expert is used. When only one domain expert is available, it is reasonable for the systems to assume that an all-knowing expert is used, although we should be aware that mistakes can occur and warning systems could be implemented when contradictory validations are made by the domain expert. When more domain experts are available, a skeptical approach or a voting approach may be used to increase the quality of the KG in terms of correctness and completeness.

[3]This covers the case of incoherent ontologies, as the fact that a concept C is unsatisfiable can be represented by the axiom $C \sqsubseteq \bot$ and thus W consists of such axioms. It also covers the case of inconsistent KGs as in that case $W = \{\top \sqsubseteq \bot\}$.

4. Framework for building KG repair systems

In this section we show a framework for building KG repair systems. The framework uses the basic operations debugging, weakening and strengthening (in addition to just adding and removing axioms). We define the operations in Sect. 4.1. Note that we define the operations in a declarative way, but do not provide implementations. Indeed, the operations can be implemented in different ways and different algorithms may be needed for different description logics. Our discussion is agnostic with respect to the chosen description logic as well as of the chosen implementation of the basic operations.[4]

Further, the framework includes combination operators (Sect. 4.2) that represent different choices for how the basic operations are combined. The choice of these combination operators has an influence on the quality of the repaired KG in terms of correctness and completeness.

The framework provides a blueprint for designing KG repair systems. A system with particular properties regarding the correctness and completeness of the repaired KG can be developed by choosing implementations for the basic operations, by choosing combination operators, and by deciding how these can be interleaved.

4.1. Basic operations

Debugging aims to find, given a set of incorrect axioms W, a set of incorrect asserted axioms W_a such that when the axioms in W_a are removed from the KGs, the axioms in W cannot be derived anymore (Fig. 1). Removing these incorrect asserted axioms leads to a less incorrect KG than the original KG. Many approaches have been proposed (e.g., [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28], overview in [5]). A basic approach is based on the computation of justifications for the incorrect axioms and then computing a Hitting set over the set of justifications. A justification for axiom ψ in $\mathcal{T}$ is a set of axioms $\mathcal{T}' \subseteq \mathcal{T}$ such that $\mathcal{T}' \models \psi$ and $\forall \mathcal{T}'' \subsetneq \mathcal{T}' : \mathcal{T}'' \not\models \psi$. A Hitting set for a collection of sets $\mathcal{S}$ is a set $H \subseteq \cup_{S \in \mathcal{S}} S$ such that $\forall S \in \mathcal{S} : H \cap S \neq \emptyset$. In general, if we assume that the oracle makes no mistakes and that W is validated to include only incorrect axioms, then there is a solution for W_a.

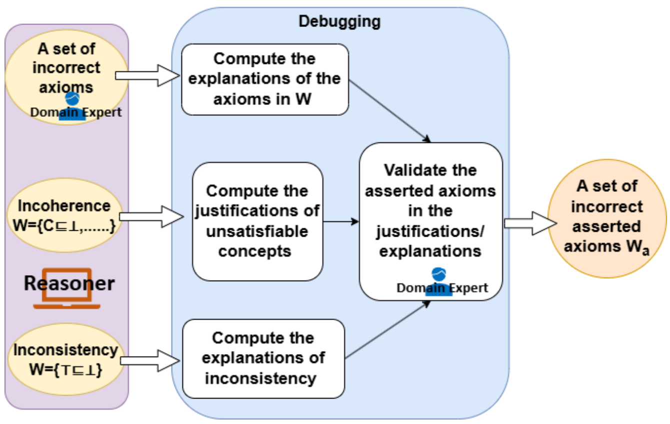


Figure 1: The input to the debugging phase is a KG together with the identified defects, represented as a set of incorrect axioms W. The output is a set of incorrect asserted axioms W_a such that when these asserted axioms in W_a are removed from the KG, the defects in W cannot be derived anymore.

Weakening aims to, given an axiom, find axioms that are weaker than the given axiom, i.e., the given axiom logically implies the weaker axioms within the context of the KG (Fig. 2). Given an axiom $P \sqsubseteq Q$, weakening can be done by replacing P by a more specific concept or replacing Q by a more general concept. For an axiom $P(a)$, P could be replaced by a more general concept. In the context of repair, when removing incorrect asserted axioms, unfortunately, correct knowledge may also be removed. To

[4] Most of the work on KG or ontology repair systems has focused on developing algorithms for particular basic operations and languages, which can be used as instantiations for the basic operations in the framework.

alleviate this problem, we may try to replace the incorrect asserted axioms with *correct* weaker versions instead of just removing them. The correct weakened axioms represent correct knowledge that would be removed by removing the axioms in W_a and that should be kept in the KG. They contribute to A in Def. 4. Thus, weakening leads to a more complete KG than the KG obtained by just removing the incorrect axioms from the original KG. Note, however, that there may not be any correct weakened axioms for a given incorrect axiom. Weakening algorithms are proposed in, e.g., [29, 30, 31, 32, 33, 34, 35].

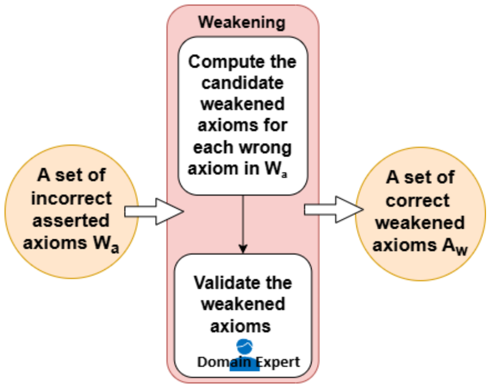


Figure 2: The input to the weakening phase is a set of incorrect asserted axioms W_a. After weakening, the output is a set of axioms A_w, which are correct weaker versions of these axioms in W_a.

Strengthening aims to find axioms that would make a given axiom derivable within the context of the KG (Fig. 3). Given an axiom $P \sqsubseteq Q$, strengthening can be done by replacing P by a more general concept or replacing Q by a more specific concept. For an axiom $P(a)$, P could be replaced by a more specific concept. In the context of repair, strengthening is applied to correct weakened axioms as well as to missing correct axioms which can be replaced by *correct* stronger axioms, thereby possibly adding additional knowledge to the KG and thus leading to a more complete KG. They contribute to A in Def. 4. As the strenghtening is applied to correct axioms, these axioms are correct strenghtened axioms of themselves. Strengthening algorithms are proposed in, e.g., [36, 37, 38, 39, 34, 35].

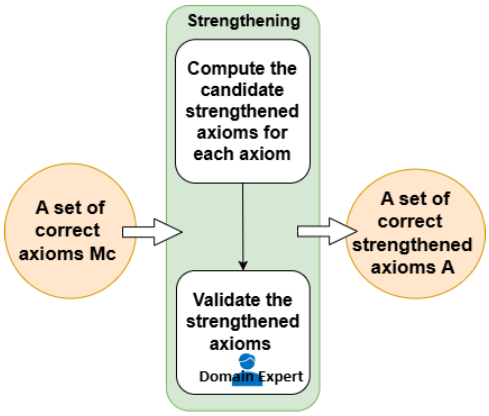


Figure 3: The input to the strengthening phase is a set of correct (missing) axioms M_c. The output is a set of axioms A, which are correct stronger versions of these correct axioms in M_c.

4.2. Combination operators

In [34, 35] combination operators were introduced to combine the basic operators for repairing ontologies. They reflect choices regarding, e.g., processing axioms all at once or one at the time, when to validate generated solutions, and when to update the ontology. These operators can also be used to repair KGs and are shown in Table 1.[5]

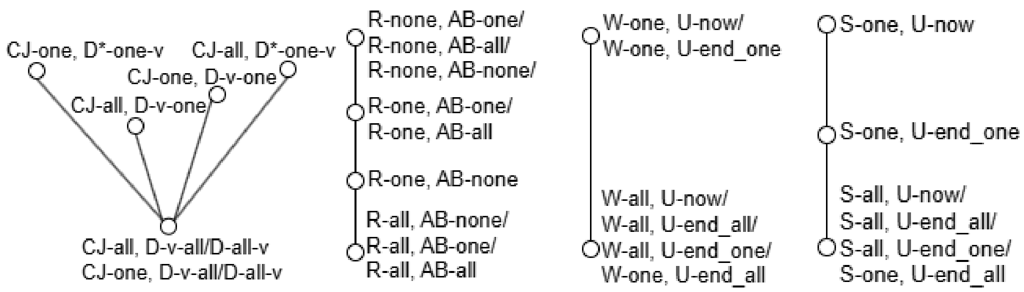
Table 1
Debugging, removing, weakening and strengthening - operations. (Updated from [34, 35].)

Operations	Description
CJ-one	Compute the justifications for one incorrect axiom at the time.
CJ-all	Compute the justifications for all incorrect axioms at once.
D-one-v	Generate one Hitting set from the justifications, then validate the asserted axioms in the generated Hitting set.
D-v-one	Validate one valid Hitting set in the justifications.
D-all-v/D-v-all	Validate all asserted axioms from the justifications.
R-all	Remove all the incorrect axioms at once.
R-one	Remove the incorrect axioms one at a time.
R-none	Remove nothing.
W-all	Weaken all incorrect axioms at once.
W-one	Weaken the incorrect axioms one at a time
S-all	Strengthen all weakened/missing axioms at once.
S-one	Strengthen the weakened/missing axioms one at a time.
AB-one	Add one incorrect axiom back.
AB-all	Add all incorrect axioms back.
AB-none	Add nothing back.
U-now	Update the changes immediately.
U-end_one	Update the changes after the iteration of each incorrect axiom.
U-end_all	Update the changes after iterations of all incorrect axioms.

The choice of combination operators influences the completeness and correctness of the repaired KG as well as the amount of validation work by a domain expert. This can be represented in Hasse diagrams (Figure 4).[6,7] Operations higher up in the diagrams use more information[8] during the computations and lead to more complete and more incorrect KGs.[9] For instance, Figure 4c shows that weakening one axiom at a time and immediately updating the KG (*W-one*, *U-now*) leads to a more complete KG than the other choices. Figure 4d, shows that using one-axiom-at-a-time strengthening and immediate updates (*S-one*, *U-now*) leads to more complete KGs than using one-axiom-at-a-time strengthening and updating the KG after each weakened axiom set for an incorrect axiom (*S-one*, *U-end_one*). The latter leads in turn to more complete KGs than for the other choices. These statements were proven for ontologies in [35], but the proofs can be straightforwardly extended for KGs [40].

For weakening and strengthening, operations higher up in the diagrams lead to more validation work for the domain expert, while for debugging, they lead to less validation work. In [34, 35] it was shown that most systems for repairing ontologies used one particular fixed set of combination operators, thus not allowing choices regarding the level of correctness and completeness.

[5]We note that the operators in Table 1 that deal with computations essentially compute solutions or deal with axioms one at a time or all at once. In general, they could also deal with a subset of all cases at a time. Thus, the X-one and X-all versions of an operator X can be seen as the extreme cases of a spectrum of cases.
[6]As discussed in the previous footnote, instead of X-one and X-all for an operator X, one could also work with a subset, and such operators would fit in between the extreme cases in the Hasse diagrams.
[7]In the Hasse diagram (Figure 4(a)), D* means the final output is a valid Hitting set. However, note that this operator does **not** guarantee a repair (even in the case where an oracle makes no mistakes).
[8]For instance, by adding new (correct) axioms to the KG as soon as they are computed and validated which can then be used when computing repairing solutions for other axioms.
[9]We note that adding more correct knowledge to an ontology makes the ontology more complete, but also possibly more incorrect. This is because the newly added correct knowledge may, together with existing incorrect knowledge in the KG, allow the derivation of new incorrect knowledge. However, the root cause is the existing incorrect knowledge and once that knowledge would be removed, also the newly derived incorrect knowledge would be removed. Therefore, this should not be a reason for not adding correct knowledge to the KG.



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

Figure 4: Hasse diagrams. (a) selecting and debugging; (b) removing and adding back incorrect axioms; (c) weakening and updating; (d) strengthening and updating. (Updated from [34, 35].)

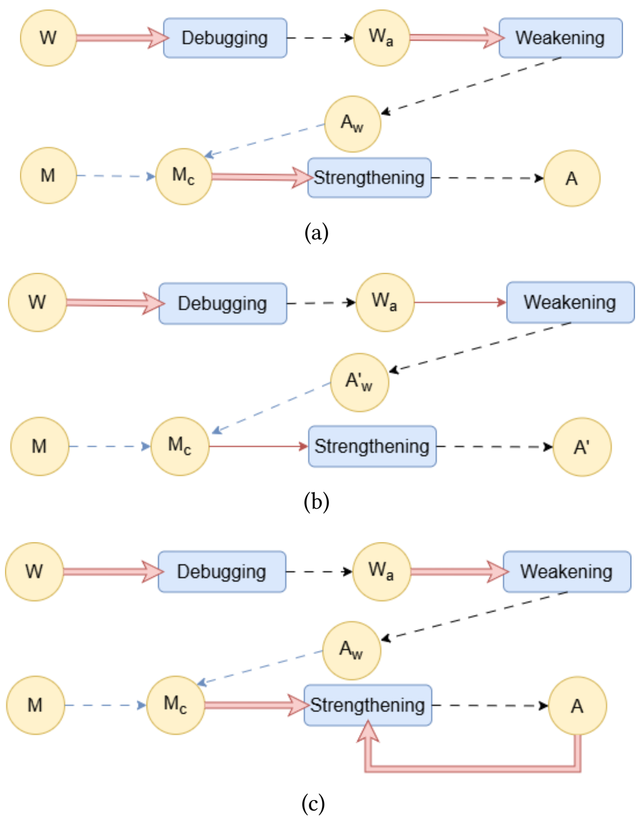


Figure 5: Examples of different repairing workflows. Circles represent input and output data, while rounded rectangles represent different operations (X). Thick red arrows indicate X-all operators (the operation applies to all axioms at once) whereas thin red arrows indicate X-one operators (the operation applies to axioms one at a time). Black dashed arrows represent the output of an operation X. Blue dashed arrows represent combining axiom sets.

4.3. Developing repair systems

To develop a KG repair system, we need to (i) decide which basic operations to use and develop implementations for these that deal with KGs in the chosen description logic, (ii) decide on which combination operators to use, as well as (iii) decide whether the steps can be interleaved and in that case how. The Hasse diagrams can then be used during decision making by showing the influence of the different choices on the correctness and completeness of the repaired KG as well as on the amount of validation work needed.

Fig. 5 shows examples of how the basic operations and combination operators can be used to define a repairing workflow where the basic operations then still need to be instantiated with specific algorithms. Fig. 5a shows one possible variation of a linear workflow of KG repair. It begins with debugging all the incorrect axioms in W at once, followed by weakening all incorrect asserted axioms in W_a at once and finally all correct weakened axioms in A_w together with the missing axioms M are strengthened all at once. Fig. 5b shows another possible variation. Like the previous workflow, it begins with debugging all incorrect axioms in W at once. However, instead of weakening and strengthening all axioms at once, this workflow applies the weakening and strengthening one axiom at a time. Using the Hasse diagrams, we can then conclude that the workflow in Fig. 5b removes the same asserted incorrect axioms, but leads to more complete KG than the workflow in Fig. 5a. It also requires more validation work for the domain expert.

Figure 5c shows a variant of the workflow in Fig. 5a where the result of the strengthening step is used as input for additional strengthening steps to find additional correct missing axioms and make the KG more complete. For ontologies, such a mechanism was implemented in, e.g., [41].

5. Choices regarding ownership and refutability

Ownership. Up to now, we have treated all axioms in the KG in the same way. However, in some cases the situation is more complex. For instance, in a KG with several networked ontologies, the ontology and alignment owners may want to retain different levels of autonomy and not necessarily allow others to change or propose changes to their ontologies and alignments. Similarly, when data is owned by different organizations, others may not be allowed to change data or use that data for computations. Also, in the case that resources (e.g., computation time and domain experts) are scarce, it may be advantageous or necessary to focus on a part of the KG (e.g., one ontology or one data set). In [35] autonomy levels for ontology networks were introduced that can be used as a basis for autonomy levels for KGs (Table 2).[10]

A first choice relates to the knowledge that can be used during the computation of repair solutions (KB choices in Table 2). At one extreme, we use the whole KG to compute solutions. At the other extreme, we only use a single ontology or alignment. A second choice relates to which axioms to retain for the final repair (AS choices in Table 2,[11] i.e. A in Def. 4. In the most general case, axioms belonging to all parts of the KG are retained, i.e., incorrect axioms in the KG may lead to removing and adding axioms in any ontology, alignment or data set in the KG. At the other end of the spectrum, we only retain axioms within a specific ontology, alignment or data set.

Also these operators can be organized in a Hasse diagram where the 'KG' operators lead to the most complete KGs but with most validation work for the domain experts, and the other operators ('O', 'M', 'D') to less complete KGs, but less validation work than the respective 'KG' operators. This means that, for example, if all KG part owners cooperate and make their part of the KG available, the final repaired KG will be more complete. When an ontology owner uses KB_{KG} and AS_O, the whole KG would be used to repair the ontology, but only axioms related to the ontology would be retained (and some axioms related to other parts may be discarded). Further, when an ontology owner uses KB_O and AS_O, only the ontology itself would be used to find solutions and repair the ontology, but fewer solutions may be found than in the case of using the whole KG.

Refutability. We note that the autonomy levels and operators can also be used to deal with cases in which a belief is stated about which axioms are deemed to be correct and should never be removed, and which axioms are refutable. In many KG repair systems, ontologies are deemed correct and only ABox statements are refutable (e.g., [42, 43, 44, 45]). In this case the computation often uses KB_{KG} while the final repair uses AS_D. Further, in the ontology alignment field, most systems that provide alignment repair, assume that the ontologies are correct, and require that the repair only contains

[10] Similarly to the combination operators, the operators in Table 2 can be seen as extreme cases.

[11] We note that, for AS_O, AS_M, AS_D, if the repair solutions only contain axioms in other parts of the KG, then these choices do not guarantee a repair for the KG as a whole.

mappings [46, 47, 48, 49, 50], thus using KB_{KG} and AS_M.

Table 2
Choices for background knowledge (KB) and answer sets (AS). (Updated from [35].)

Choices	Description
KB_{KG}	Use the whole KG for computing repairs.
KB_O	Disconnect each ontology from the ontology network. Use the axioms within the respective ontologies to compute repairs.
KB_M	Disconnect each alignment from the ontology network. Use the mappings within the respective alignments to compute repairs.
AS_{KG}	Add and remove all types of axioms.
AS_O	Add and remove only axioms within the respective ontologies.
AS_M	Add and remove only mappings between the ontologies.
AS_D	Add and remove only axioms within the respective data sets.

6. Conclusion

In this paper we discussed issues related to logic-based KG repair and the influence of different choices on the quality of repaired KGs in terms of correctness and completeness. The choices pertained to how to combine the basic operators debugging, weakening and strengthening, and to choices regarding autonomy for owners of parts of a KG. The ideas presented in this paper show how different repair solutions can be compared, and allow for decision making when developing repair systems. They also show that existing KG repair systems could be extended to give users a choice regarding the desired level of correctness and completeness of repaired KGs.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic Web 8 (2016) 489–508. doi:10.3233/SW-160218.

[2] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: Lessons and challenges, Communications of the ACM 62 (2019) 36–43. doi:10.1145/3331166.

[3] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2022) 71:1–37. doi:10.1145/3447772.

[4] J. Chen, H. Dong, J. Hastings, E. Jiménez-Ruiz, V. López, P. Monnin, C. Pesquita, P. Škoda, V. Tamma, Knowledge Graphs for the Life Sciences: Recent Developments, Challenges and Opportunities, Transactions on Graph Data and Knowledge 1 (2023) 5:1–33. doi:10.4230/TGDK.1.1.5.

[5] P. Lambrix, Completing and debugging ontologies: State of the art and challenges in repairing ontologies, ACM Journal of Data and Information Quality 15 (2023) 41:1–38. doi:10.1145/3597304.

[6] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.

[7] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, C. Pesquita, User validation in ontology alignment, in: P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck, Y. Gil (Eds.), The Semantic Web - ISWC 2016, 15th International Semantic Web Conference, Kobe,

Japan, October 17-21, 2016, Proceedings, Part I, volume 9981 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 200–217. doi:10.1007/978-3-319-46523-4_13.

[8] S. Schlobach, R. Cornet, Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies, in: G. Gottlob, T. Walsh (Eds.), Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003, pp. 355–360. URL: https://www.ijcai.org/Proceedings/03/Papers/053.pdf.

[9] S. Schlobach, Debugging and Semantic Clarification by Pinpointing, in: A. Gomez-Perez, J. Euzenat (Eds.), The Semantic Web: Research and Applications - Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29-June 1, 2005. Proceedings, volume 3532 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 226–240. doi:10.1007/11431053_16.

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

[11] T. Meyer, K. Lee, R. Booth, J. Pan, Finding maximally satisfiable terminologies for the description logic ALC, in: A. Cohn (Ed.), AAAI'06 - Proceedings of the 21st national conference on Artificial intelligence, AAAI Press, 2006, pp. 269–274. URL: https://www.aaai.org/Papers/AAAI/2006/AAAI06-043.pdf.

[12] A. Kalyanpur, B. Parsia, E. Sirin, J. A. Hendler, Debugging unsatisfiable classes in OWL ontologies, Journal of Web Semantics 3 (2005) 268–293. doi:10.1016/J.WEBSEM.2005.09.005.

[13] A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca Grau, Repairing unsatisfiable concepts in OWL ontologies, in: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings, volume 4011 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 170–184. doi:10.1007/11762256_15.

[14] A. Kalyanpur, B. Parsia, M. Horridge, E. Sirin, Finding all justifications of OWL DL entailments, in: K. Aberer, K. Choi, N. F. Noy, D. Allemang, K. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Eds.), The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007, volume 4825 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 267–280. doi:10.1007/978-3-540-76298-0_20.

[15] J. Lehmann, L. Bühmann, ORE - A tool for repairing and enriching knowledge bases, in: P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, B. Glimm (Eds.), The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part II, volume 6497 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 177–193. doi:10.1007/978-3-642-17749-1_12.

[16] Q. Ji, Z. Gao, Z. Huang, M. Zhu, An efficient approach to debugging ontologies based on patterns, in: J. Z. Pan, H. Chen, H. Kim, J. Li, Z. Wu, I. Horrocks, R. Mizoguchi, Z. Wu (Eds.), The Semantic Web - Joint International Semantic Technology Conference, JIST 2011, Hangzhou, China, December 4-7, 2011. Proceedings, volume 7185 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 425–433. doi:10.1007/978-3-642-29923-0_33.

[17] K. Moodley, T. Meyer, I. J. Varzinczak, Root justifications for ontology repair, in: S. Rudolph, C. Gutiérrez (Eds.), Web Reasoning and Rule Systems - 5th International Conference, RR 2011, Galway, Ireland, August 29-30, 2011. Proceedings, volume 6902 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 275–280. doi:10.1007/978-3-642-23580-1_24.

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

[19] D. Fleischhacker, C. Meilicke, J. Völker, M. Niepert, Computing incoherence explanations for learned ontologies, in: W. Faber, D. Lembo (Eds.), Web Reasoning and Rule Systems - 7th International Conference, RR 2013, Mannheim, Germany, July 27-29, 2013. Proceedings, volume 7994 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 80–94. doi:10.1007/978-3-642-39666-3_7.

[20] M. F. Arif, C. Mencía, A. Ignatiev, N. Manthey, R. Peñaloza, J. Marques-Silva, BEACON: an efficient sat-based tool for debugging $^{+}$ ontologies, in: N. Creignou, D. L. Berre (Eds.), Theory

and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings, volume 9710 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 521–530. doi:10.1007/978-3-319-40970-2_32.

[21] P. Rodler, W. Schmid, On the impact and proper use of heuristics in test-driven ontology debugging, in: C. Benzmüller, F. Ricca, X. Parent, D. Roman (Eds.), Rules and Reasoning - Second International Joint Conference, RuleML+RR 2018, Luxembourg, September 18-21, 2018, Proceedings, volume 11092 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 164–184. doi:10.1007/978-3-319-99906-7_11.

[22] 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 - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings, volume 10833 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 340–359. doi:10.1007/978-3-319-90050-6_19.

[23] J. Méndez, C. Alrabbaa, P. Koopmann, R. Langner, F. Baader, R. Dachselt, Evonne: A visual tool for explaining reasoning with OWL ontologies and supporting interactive debugging, Computer Graphics forum 42 (2023) e14730:1–15. doi:10.1111/cgf.14730.

[24] A. Nentidis, C. Akasiadis, A. Charalambidis, A. Artikis, Dealing with inconsistency for reasoning over knowledge graphs: A survey, ArXiv (2025). doi:10.48550/arXiv.2502.19023.

[25] T.-K. Tran, M. H. Gad-Elrab, D. Stepanova, E. Kharlamov, J. Strotgen, Fast computation of explanations for inconsistency in large-scale knowledge graphs, Proceedings of The Web Conference 2020 (2020) 2613 – 2619. doi:10.1145/3366423.3380014.

[26] T. de Groot, J. Raad, S. Schlobach, Analysing large inconsistent knowledge graphs using anti-patterns, in: R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski, M. Alam (Eds.), The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings, volume 12731 of *Lecture Notes in Computer Science*, Springer, 2022, p. 40–56. doi:10.1007/978-3-030-77385-4_3.

[27] C. Meilicke, D. Ruffinelli, A. Nolle, H. Paulheim, H. Stuckenschmidt, Fast ABox consistency checking using incomplete reasoning and caching, in: S. Costantini, E. Franconi, W. V. Woensel, R. Kontchakov, F. Sadri, D. Roman (Eds.), Rules and Reasoning: International Joint Conference, RuleML+RR 2017, London, UK, July 12–15, 2017, Proceedings, volume 10364 of *Lecture Notes in Computer Science*, Springer, 2017, p. 168–183. doi:10.1007/978-3-319-61252-2_12.

[28] H. Paulheim, H. Stuckenschmidt, Fast approximate a-box consistency checking using machine learning, in: H. Sack, E. Blomqvist, M. d'Aquin, C. Ghidini, S. P. Ponzetto, C. Lange (Eds.), The Semantic Web. Latest Advances and New Domains, Springer International Publishing, 2016, pp. 135–150.

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

[30] J. Du, G. Qi, X. Fu, A practical fine-grained approach to resolving incoherent OWL 2 DL terminologies, in: J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, M. Wang (Eds.), Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, ACM, 2014, pp. 919–928. doi:10.1145/2661829.2662046.

[31] F. Baader, F. Kriegel, A. Nuradiansyah, R. Peñaloza, Making repairs in description logics more gentle, in: M. Thielscher, F. Toni, F. Wolter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018., 2018, pp. 319–328. URL: https://aaai.org/ocs/index.php/KR/KR18/paper/view/18056.

[32] N. Troquard, R. Confalonieri, P. Galliani, R. Peñaloza, D. Porello, O. Kutz, Repairing ontologies via axiom weakening, in: S. A. McIlraith, K. Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in

Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 1981–1988. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17189.
[33] R. Confalonieri, P. Galliani, O. Kutz, D. Porello, G. Righetti, N. Troquard, Towards even more irresistible axiom weakening, in: S. Borgwardt, T. Meyer (Eds.), Proceedings of the 33rd International Workshop on Description Logics (DL 2020), volume 2663 of *CEUR Workshop Proceedings*, 2020. URL: https://ceur-ws.org/Vol-2663/paper-8.pdf.
[34] Y. Li, P. Lambrix, Repairing $\mathcal{EL}$ ontologies using weakening and completing, in: C. Pesquita, E. Jimenez-Ruiz, J. McCusker, D. Faria, M. Dragoni, A. Dimou, R. Troncy, S. Hertling (Eds.), The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings, volume 13870 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 298–315. doi:10.1007/978-3-031-33455-9_18.
[35] Y. Li, P. Lambrix, Repairing networks of $\mathcal{EL}_\bot$ ontologies using weakening and completing, in: G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández, A. Hogan (Eds.), The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part I, volume 15231 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 107–125. doi:10.1007/978-3-031-77844-5_6.
[36] P. Lambrix, Q. Liu, Debugging the missing is-a structure within taxonomies networked by partial reference alignments, Data & Knowledge Engineering 86 (2013) 179–205. doi:10.1016/J.DATAK.2013.03.003.
[37] F. Wei-Kleiner, Z. Dragisic, P. Lambrix, Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms, in: C. E. Brodley, P. Stone (Eds.), Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada, AAAI Press, 2014, pp. 1120–1127. doi:10.1609/AAAI.V28I1.8858.
[38] J. Du, H. Wan, H. Ma, Practical tbox abduction based on justification patterns, in: S. P. Singh, S. Markovitch (Eds.), Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 1100–1106. doi:10.1609/aaai.v31i1.10683.
[39] F. Haifani, P. Koopmann, S. Tourret, C. Weidenbach, Connection-minimal abduction in $\mathcal{EL}$ via translation to FOL, in: J. Blanchette, L. Kovács, D. Pattinson (Eds.), Automated Reasoning, IJCAR 2022, volume 13385 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 188–207. doi:10.1007/978-3-031-10769-6_12.
[40] Y. Li, Logic-based repairing of ontologies and knowledge graphs, Ph.D. thesis, Linköping University, 2026.
[41] P. Lambrix, F. Wei-Kleiner, Z. Dragisic, Completing the is-a structure in light-weight ontologies, Journal of Biomedical Semantics 6 (2015) 12:1–26. doi:10.1186/S13326-015-0002-8.
[42] F. Kriegel, Beyond optimal: Interactive identification of better-than-optimal repairs, in: Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing, 2025, pp. 1019–1026. doi:10.1145/3672608.3707750.
[43] C. Akasiadis, A. Nentidis, A. Charalambidis, A. Artikis, Detecting and fixing inconsistency of large knowledge graphs, in: Proceedings of the 13th Hellenic Conference on Artificial Intelligence, SETN '24, Association for Computing Machinery, 2024. doi:10.1145/3688671.3688766.
[44] M. Bienvenu, C. Bourgaux, F. Goasdoué, Query-driven repairing of inconsistent DL-Lite knowledge bases, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, AAAI Press, 2016, p. 957–964. URL: https://www.ijcai.org/Proceedings/16/Papers/140.pdf.
[45] J. Chen, E. Jiménez-Ruiz, I. Horrocks, X. Chen, E. B. Myklebust, An assertion and alignment correction framework for large scale knowledge bases, Semantic Web 14 (2021) 29–53. doi:10.3233/SW-210448.
[46] Q. Ji, P. Haase, G. Qi, P. Hitzler, S. Stadtmüller, Radon : Repair and diagnosis in ontology networks, in: The SemanticWeb : Research and Applications, Proceedings of the 6th European SemanticWeb Conference (ESWC2009), Heraklion, Crete, Greece, 31st May - 4th June 2009. Ed.: L. Aroyo, volume 5554 of *Lecture Notes in Computer Science*, Springer-Verlag, 2009. doi:10.1007/978-3-642-02121-3_71.

[47] C. Meilicke, Alignment Incoherence in Ontology Matching, Ph.D. thesis, University of Mannheim, 2011.
[48] E. Jiménez-Ruiz, B. C. Grau, Logmap: logic-based and scalable ontology matching, in: Proceedings of the 10th International Conference on The Semantic Web - Volume Part I, ISWC'11, Springer-Verlag, Berlin, Heidelberg, 2011, p. 273–288. doi:10.1007/978-3-642-25073-6_18.
[49] E. Jiménez-Ruiz, B. C. Grau, Y. Zhou, I. Horrocks, Large-scale interactive ontology matching: algorithms and implementation, in: Proceedings of the 20th European Conference on Artificial Intelligence, ECAI'12, IOS Press, NLD, 2012, p. 444–449. doi:10.3233/978-1-61499-098-7-444.
[50] E. Santos, D. Faria, C. Pesquita, F. M. Couto, Ontology alignment repair through modularization and confidence-based heuristics, PLoS ONE 10(12):e0144807) (2015) 1–19. doi:10.1371/journal.pone.0144807.