

Making Axiom Weakening Work in *SRIOQ*

Roland Bernard¹, Oliver Kutz¹ and Nicolas Troquard¹

¹ Free University of Bozen-Bolzano, Italy

Abstract

Axiom weakening is a technique that allows for a fine-grained repair of inconsistent ontologies. Its main advantage is that it repairs ontologies by making axioms less restrictive rather than by deleting them, employing refinement operators. In this paper, we build on previously introduced axiom weakening for *ALC*, and show how it can be extended to deal with *SRIOQ*, the expressive and decidable description logic underlying OWL 2 DL. The main problem here is to ensure that the regularity conditions of *SRIOQ* are preserved in the weakening process, as not every weaker axiom can be inserted into an ontology without compromising regularity. We present a basic regularity-preserving weakening approach for *SRIOQ*, describe briefly a prototype implementation realising it as well as an accompanying Protégé plugin, and perform and discuss basic evaluations of the approach.


Keywords

Description Logic, Knowledge Refinement, Axiom Weakening, Ontology Debugging, *SRIOQ*, Protégé

1. Introduction

Many approaches to repairing inconsistent ontologies amount to identifying problematic axioms and then removing them (e.g., [1, 2, 3, 4]). Whilst this approach is obviously sufficient to guarantee that the obtained ontology is consistent, it tends to lead to information loss as a secondary effect, as we outlined in detail in previous work [5, 6], where also a more extensive discussion of related work can be consulted. Further, this information loss can be reduced by weakening the inferential power of an axiom rather than by deleting it [7, 8, 9, 5, 6]. In [5], axiom weakening using refinement operators has been described for *ALC* and experimentally evaluated, showing that axiom weakening is able to retain more information than deletion. In [6], the axiom weakening has been extended to include many aspects of *SRIOQ*, notably omitting, however, the weakening of RBox axioms. The authors of [6] further show that the proposed repair by iterated weakening almost surely terminates.


In this paper, we extend the previous work on axiom weakening in DLs by extending the underlying basic principles to the logic *SRIOQ*, including also the weakening of RIAs. We discuss a number of scenarios where weakening can impact regularity of *SRIOQ* RBoxes, and provide a framework where this is avoided. Additionally, by implementing the proposed refinement and weakening operators we are able to perform experimental evaluation, also on ontologies using the more expressive features of *SRIOQ*. The results reaffirm the results of [5] for the case of *SRIOQ*, namely that weakening may significantly outperform deletion.

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

 roland.bernard@student.unibz.it (R. Bernard); oliver.kutz@unibz.it (O. Kutz); nicolas.troquard@unibz.it (N. Troquard)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

2. Axiom Weakening for $SR\mathcal{OIQ}$

We first give a brief description of the DL $SR\mathcal{OIQ}$; for full details, see [10, 11, 12].

2.1. Defining $SR\mathcal{OIQ}$

The syntax of $SR\mathcal{OIQ}$ is based on a vocabulary of three disjoint sets N_C, N_R, N_I of, respectively, *concept names*, *role names*, and *individual names*. The sets of, respectively, $SR\mathcal{OIQ}$ *roles* and $SR\mathcal{OIQ}$ *concepts* are generated by the following grammar.

$$\begin{aligned} R, S &::= U \mid r \mid r^- , \\ C &::= \perp \mid \top \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C \mid \\ &\quad \geq n S.C \mid \leq n S.C \mid \exists S.Self \mid \{a\} , \end{aligned}$$

where $A \in N_C$ is a concept name, $r \in N_R$ is a role name, $a \in N_I$ is an individual name and $n \in \mathbb{N}_0$ is a non-negative integer. U is the universal role. S is a *simple role* in the RBox \mathcal{R} (see below). In the following, $\mathcal{L}(N_C, N_R, N_I)$ and $\mathcal{L}(N_R) = N_R \cup \{U\} \cup \{r^- \mid r \in N_R\}$ denote, respectively, the set of concepts and roles that can be built over N_C, N_R , and N_I in $SR\mathcal{OIQ}$.

We next define the notions of TBox, ABox, and (regular) RBox, of complex role inclusions, and of (non-)simple roles: A TBox \mathcal{T} is a finite set of concept inclusions (GCIs) of the form $C \sqsubseteq D$ where C and D are concepts. The TBox is used to store terminological knowledge concerning the relationship between concepts. An ABox \mathcal{A} is a finite set of statements of the form $C(a)$, $R(a, b)$, $\neg R(a, b)$, $a = b$, and $a \neq b$, where C is a concept, R is a role and a and b are individual names. The ABox expresses knowledge regarding individuals in the domain. An RBox \mathcal{R} is a finite set of role inclusions (RIAs) of the form $R_1 \circ \dots \circ R_n \sqsubseteq R$, and disjoint role axioms $disjoint(S_1, S_2)$ where R, R_1, \dots, R_n, S_1 , and S_2 are roles. S_1 and S_2 are simple (defined next) in the RBox \mathcal{R} . The special case of $n = 1$ is a simple role inclusion, while we call the cases where $n > 1$ *complex role inclusions*. The RBox represents knowledge about the relationships between roles.

The set of *non-simple* roles in \mathcal{R} is the smallest set such that: U is non-simple; any role R that appears as the super role of a complex RIA $R_1 \circ \dots \circ R_n \sqsubseteq R$ where $n > 1$ is non-simple; any role R that appears on the right-hand side of a simple RIA $S \sqsubseteq R$ where S is non-simple, is also non-simple; and a role r is non-simple if and only if r^- is non-simple. All other roles are *simple*.

For convenience, let us define the function $inv(R)$ such that $inv(r) = r^-$ and $inv(r^-) = r$ for all role names $r \in N_R$. An RBox \mathcal{R} is *regular* if there exists a preorder \preceq , i.e., a transitive and reflexive relation over the set of roles appearing in \mathcal{R} such that, $R \preceq S \iff inv(R) \preceq S$, and all RIAs in \mathcal{R} are of the forms: $inv(R) \sqsubseteq R$, $R \circ R \sqsubseteq R$, $S \sqsubseteq R$, $R \circ S_1 \circ \dots \circ S_n \sqsubseteq R$, $S_1 \circ \dots \circ S_n \circ R \sqsubseteq R$, or $S_1 \circ \dots \circ S_n \sqsubseteq R$, where $n > 1$ and R, S, S_1, \dots, S_n are roles such that $S \preceq R$, $S_i \preceq R$, and $R \not\preceq S_i$ for $i = 1, \dots, n$. This regularity restriction has been chosen to align with the implementation of the OWL 2 DL [13] profile checker in the OWL API [14].

Definition 1. A $SR\mathcal{OIQ}$ ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A} \cup \mathcal{R}$ consists of a TBox \mathcal{T} , an ABox \mathcal{A} , and a RBox \mathcal{R} in the language of $SR\mathcal{OIQ}$, and where the RBox \mathcal{R} is regular.

The semantics of $SR\mathcal{OIQ}$ is defined using *interpretations* $I = \langle \Delta^I, \cdot^I \rangle$, where Δ^I is a non-empty *domain* and \cdot^I is a function associating to each individual name a an element of the

domain $a^I \in \Delta^I$, to each concept C a subset of the domain $C^I \subseteq \Delta^I$, and to each role R a binary relation on the domain $R^I \subseteq \Delta^I \times \Delta^I$; see [12, 10] for further details. An interpretation I is a *model* for \mathcal{O} if it satisfies all the axioms in \mathcal{O} .

Given two concepts C and D we say that C is *subsumed* by D (or D *subsumes* C) with respect to the ontology \mathcal{O} , written $C \sqsubseteq_{\mathcal{O}} D$, if $C^I \subseteq D^I$ in every model I of \mathcal{O} . Further, C is *strictly subsumed* by D , written $C \sqsubset_{\mathcal{O}} D$, if $C \sqsubseteq_{\mathcal{O}} D$ but not $D \sqsubseteq_{\mathcal{O}} C$. Analogously, given two roles R and S , R is subsumed by S with respect to \mathcal{O} ($R \sqsubseteq_{\mathcal{O}} S$) if $R^I \subseteq S^I$ in all models I of \mathcal{O} . Again, $R \sqsubset_{\mathcal{O}} S$ holds if $R \sqsubseteq_{\mathcal{O}} S$ but not $S \sqsubseteq_{\mathcal{O}} R$.

2.2. Weakening, reference ontologies, covers, refinement operator

The case of \mathcal{ALC} . Axiom weakening, as discussed in [5], is based on refinement operators. Two types of refinement operators are used, a specialization operator and a generalization operator. They return for a given concept a set of, respectively, more specific or more general concepts. In [5] the proposed refinement operator is based on upward and downward cover sets. The upward or downward cover for a concept is the set of, respectively, the most specific generalization or most general specializations from the set of subconcepts. Given an inconsistent ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A} \cup \mathcal{R}$, to be able to compute a non-trivial upcover and downcover, we first need to find a consistent subontology \mathcal{O}^{ref} of \mathcal{O} to serve as *reference ontology*. One approach is to pick a random maximal consistent subset of \mathcal{O} and choose it as reference ontology \mathcal{O}^{ref} ; another is to choose the intersection of all maximal consistent subsets of \mathcal{O} (e.g., [15]).

Once a reference ontology \mathcal{O}^{ref} has been chosen, and as long as \mathcal{O} is inconsistent, we select a “bad axiom” and replace it with a random weakening of it with respect to \mathcal{O}^{ref} . We can randomly sample a number of (or all the) minimally inconsistent subsets of axioms $I_1, I_2, \dots, I_k \subseteq \mathcal{O}$ and return one axiom in $\mathcal{T} \cup \mathcal{A}$ from the ones occurring the most often. Weakening of GCIs is performed by either using the specialization operator to refine the left-hand side or the generalization operator for the right-hand side. For class assertions, the concept is refined using the generalization operator.

The case of \mathcal{SROIQ} . The main difficulties that arise when weakening axioms in \mathcal{SROIQ} ontologies, and especially when weakening RIAs, are related to ensuring that the constraints on the use of non-simple roles and the regularity of the RBox as a whole are maintained. Not every weaker axiom can be inserted into a valid \mathcal{SROIQ} ontology without causing a violation of these restrictions, as shown in the next example.

Example 1. Take the ontology $\mathcal{O} = \{r \circ s \circ r \sqsubseteq t, r \sqsubseteq s, \top \sqsubseteq \forall t.\perp, \exists s.\text{Self} \sqsubseteq \top\}$. Since t is empty in every model of this ontology, the axiom $r \sqsubseteq s$ could be weakened to $t \sqsubseteq s$ if we ignore the additional constraints. This would result in an ontology where s is non-simple, which is not allowed since s is used as part of a self constraint. Additionally, using this weakening would also cause a non-regular RBox, because for any preorder \preceq , $t \not\preceq s$ must hold for the complex RIA and $t \preceq s$ must hold for the new axiom. Yet, this is a contradiction.

To prevent these kinds of issues, we restrict how concepts are refined and RIAs weakened. In [6], the refinement of RIAs was not considered at all to avoid these problems. In this paper, however, we have extended the axiom weakening operator to handle also RIAs. To achieve this,

we must ensure that only simple roles are used when weakening disjoint role axioms or refining cardinality and *Self* constraints. Further, it must be guaranteed that all roles that are currently used in such contexts remain simple when adding the weakened axioms to the ontology. Finally, the addition of a weakened axiom must maintain the regularity of the RBox.

We now discuss which restrictions we applied in order to satisfy these requirements.

2.3. Saving Regularity

Firstly, the covers and refinement operators for roles operate only on roles that are simple. A similar restriction has already been applied in the refinement operator suggested in [6]. Restricting the refinement to simple roles guarantees that the new axioms created by weakening will not contain non-simple roles in axioms or concepts where they are not allowed. An important detail that was not considered in [6] is that the roles over which the covers operate must be simple in all ontologies that the weaker axioms are used in. It is therefore not generally sufficient to use the roles that are simple in the reference ontology, since the reference ontology may not contain all RBox axioms, and therefore contain simple roles that are not simple in the full ontology. For this reason, we give to the upward and downward cover as an argument not only the reference ontology \mathcal{O}^{ref} , but also the full ontology $\mathcal{O}^{\text{full}}$. Both \mathcal{O}^{ref} and $\mathcal{O}^{\text{full}}$ share the same vocabulary N_I , N_C , and N_R . We assume that $\mathcal{O}^{\text{ref}} \subseteq \mathcal{O}^{\text{full}}$. In the context of repairing inconsistent ontologies, $\mathcal{O}^{\text{full}}$ can be chosen to be the inconsistent ontology that we want to repair.

Then, to ensure further that by adding weakened axioms we do not cause a constraint violation in existing axioms and concepts, we choose the allowed weakening for RIAs such that all roles that are simple in $\mathcal{O}^{\text{full}}$, are also simple after adding to it a weakening of one of its axioms. We observe that for complex RIAs $S_1 \circ \dots \circ S_n \sqsubseteq R$ we should not refine the role R . Since all roles returned by our refinement operator are simple in $\mathcal{O}^{\text{full}}$, a replacement of R with a refinement R' would create an axiom which would make a role which was simple in $\mathcal{O}^{\text{full}}$ now non-simple. A similar argument can be made for refining R in a simple RIA $S \sqsubseteq R$ where the role S is non-simple in $\mathcal{O}^{\text{full}}$. So the only way to refine the super role during the weakening of a RIA is when it is a simple RIA and additionally the sub role of the axiom is simple in $\mathcal{O}^{\text{full}}$.

When it comes to refining the left-hand side of RIAs, we do not need any special restrictions. The main significant observation is that all roles that are returned by the refinement will be simple. This means that in a simple RIA $R \sqsubseteq S$, even if S is simple, replacing R with another simple role will not cause S to become non-simple. For a complex RIA $S_1 \circ \dots \circ S_n \sqsubseteq R$ on the other hand, the role R must already have been non-simple in $\mathcal{O}^{\text{full}}$, and replacing any S_i with a refinement has no effect on which roles are simple.

A more interesting question is whether such a weakening may still cause a non-regular RBox. The important insight is that simple roles are always allowed on the left-hand side of a RIA. While this is more directly evident in some alternative definitions of regularity (e.g., [16]) it is not so apparent from the one presented in this paper. Intuitively, the constraints given above for regularity disallow dependency cycles that contain complex RIAs. Simple roles cannot be part of such a cycle, since the cycle must contain at least one complex RIA to be a violation of the constraint, and all roles that depend in this sense on a complex RIA must be non-simple. A more formal justification for this fact is given in Lemma 4. Since all refinements of the left-hand

side of RIAs are performed using simple roles, these cannot lead to a non-regular RBox. Further, refinements of the super role of RIAs are only performed on simple RIAs $S \sqsubseteq R$ where S is a simple role. Since S is simple in this case, all refinements of R are allowed, potentially also if the refinement yielded a non-simple role.

Definition 2. Let \mathcal{O} be a *SR \mathcal{OIQ}* ontology. The set of subconcepts of \mathcal{O} is given by

$$\text{sub}(\mathcal{O}) = \{\top, \perp\} \cup \bigcup_{C(a) \in \mathcal{O}} \text{sub}(C) \cup \bigcup_{C \sqsubseteq D \in \mathcal{O}} (\text{sub}(C) \cup \text{sub}(D)) ,$$

where $\text{sub}(C)$ is the set of subconcepts in C .

We will define now the upward and downward cover sets for concepts and roles. Intuitively, for a given concept the upward cover is the set of the most specific generalizations from the (fixed) set of subconcepts or roles, while the downward cover set contains the most general specializations from the same set of subconcepts and roles. (Note that the related open search for e.g. a least common subsumer is in general a harder problem [17].) We define the upward and downward cover additionally also for non-negative integers, as they will be useful in the refinement of cardinality constraints.

Definition 3. Let $\mathcal{O}^{\text{full}}$ and $\mathcal{O}^{\text{ref}} \subseteq \mathcal{O}^{\text{full}}$ be two *SR \mathcal{OIQ}* ontologies that share the same vocabulary N_C , N_R , and N_I . The upward cover and downward cover for a concept C are given by

$$\begin{aligned} \text{UpCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(C) &= \{D \in \text{sub}(\mathcal{O}^{\text{full}}) \mid C \sqsubseteq_{\mathcal{O}^{\text{ref}}} D \text{ and} \\ &\quad \nexists D' \in \text{sub}(\mathcal{O}^{\text{full}}) \text{ with } C \sqsubseteq_{\mathcal{O}^{\text{ref}}} D' \sqsubseteq_{\mathcal{O}^{\text{ref}}} D\} , \\ \text{DownCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(C) &= \{D \in \text{sub}(\mathcal{O}^{\text{full}}) \mid D \sqsubseteq_{\mathcal{O}^{\text{ref}}} C \text{ and} \\ &\quad \nexists D' \in \text{sub}(\mathcal{O}^{\text{full}}) \text{ with } D \sqsubseteq_{\mathcal{O}^{\text{ref}}} D' \sqsubseteq_{\mathcal{O}^{\text{ref}}} C\} . \end{aligned}$$

The upward and downward covers for a role R are given by

$$\begin{aligned} \text{UpCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(R) &= \{S \in \mathcal{L}(N_R) \mid R \sqsubseteq_{\mathcal{O}^{\text{ref}}} S \text{ and} \\ &\quad \nexists S' \in \mathcal{L}(N_R) \text{ with } R \sqsubseteq_{\mathcal{O}^{\text{ref}}} S' \sqsubseteq_{\mathcal{O}^{\text{ref}}} S \text{ and} \\ &\quad S, S' \text{ are simple in } \mathcal{O}^{\text{full}}\} , \\ \text{DownCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(R) &= \{S \in \mathcal{L}(N_R) \mid S \sqsubseteq_{\mathcal{O}^{\text{ref}}} R \text{ and} \\ &\quad \nexists S' \in \mathcal{L}(N_R) \text{ with } S \sqsubseteq_{\mathcal{O}^{\text{ref}}} S' \sqsubseteq_{\mathcal{O}^{\text{ref}}} R \text{ and} \\ &\quad S, S' \text{ are simple in } \mathcal{O}^{\text{full}}\} . \end{aligned}$$

The upward and downward covers for a non-negative integer n are given by

$$\begin{aligned} \text{UpCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(n) &= \{n, n + 1\} , \\ \text{DownCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(n) &= \begin{cases} \{n\} & \text{if } n = 0 \\ \{n, n - 1\} & \text{if } n > 0 \end{cases} . \end{aligned}$$

Since they operate only over the subconcepts of $\mathcal{O}^{\text{full}}$, on their own, the upward and downward covers of concepts are missing some interesting refinements.

Example 2. Let $N_C = \{A, B, C\}$, $N_R = \{r, s\}$, and $\mathcal{O} = \{A \sqsubseteq B, r \sqsubseteq s\}$. $\text{sub}(\mathcal{O}) = \{\top, \perp, A, B\}$. The upward cover of $C \sqcup A$ is equal to $\text{UpCover}_{\mathcal{O}, \mathcal{O}}(C \sqcup A) = \{\top\}$. The potentiality refinement to $C \sqcup B$ will be missed even by iterated application of the upward cover because $C \sqcup B \notin \text{sub}(\mathcal{O})$. Similarly, $\text{UpCover}_{\mathcal{O}, \mathcal{O}}(\forall r.A) = \{\top\}$, even if $\forall r.B$ and $\forall s.A$ are reasonable generalizations.

To also capture these omissions, we define generalization and specialization operators that exploit the recursive structure of the concept being refined to generate more complex refinements. For convenience, we also define these operators for roles.

Definition 4. Let \uparrow and \downarrow be two functions with domain $\mathcal{L}(N_C, N_R, N_I) \cup \mathcal{L}(N_R) \cup \mathbb{N}_0$. They map every concept to a finite subset of $\mathcal{L}(N_C, N_R, N_I)$, every role to a subset of $\mathcal{L}(N_R)$, and every non-negative integer to a finite subset of \mathbb{N}_0 . The abstract refinement operator is defined recursively by induction on the structure of concepts as follows.

$$\begin{aligned}
\zeta_{\uparrow, \downarrow}(A) &= \uparrow(A) \quad , A \in N_C \cup \{\top, \perp\} \quad , \\
\zeta_{\uparrow, \downarrow}(\neg C) &= \uparrow(\neg C) \cup \{\neg C' \mid C' \in \zeta_{\downarrow, \uparrow}(C)\} \quad , \\
\zeta_{\uparrow, \downarrow}(C \sqcap D) &= \uparrow(C \sqcap D) \cup \{C' \sqcap D' \mid C' \in \zeta_{\uparrow, \downarrow}(C)\} \cup \{C \sqcap D' \mid D' \in \zeta_{\uparrow, \downarrow}(D)\} \quad , \\
\zeta_{\uparrow, \downarrow}(C \sqcup D) &= \uparrow(C \sqcup D) \cup \{C' \sqcup D' \mid C' \in \zeta_{\uparrow, \downarrow}(C)\} \cup \{C \sqcup D' \mid D' \in \zeta_{\uparrow, \downarrow}(D)\} \quad , \\
\zeta_{\uparrow, \downarrow}(\forall R.C) &= \uparrow(\forall R.C) \cup \{\forall R'.C \mid R' \in \downarrow(R)\} \cup \{\forall R.C' \mid C' \in \zeta_{\uparrow, \downarrow}(C)\} \quad , \\
\zeta_{\uparrow, \downarrow}(\exists R.C) &= \uparrow(\exists R.C) \cup \{\exists R'.C \mid R' \in \uparrow(R)\} \cup \{\exists R.C' \mid C' \in \zeta_{\uparrow, \downarrow}(C)\} \quad , \\
&\text{SRQIQ concepts:} \\
\zeta_{\uparrow, \downarrow}(\{a\}) &= \uparrow(\{a\}) \quad , \\
\zeta_{\uparrow, \downarrow}(\exists R.Self) &= \uparrow(\exists R.Self) \cup \{\exists R'.Self \mid R' \in \uparrow(R)\} \quad , \\
\zeta_{\uparrow, \downarrow}(\geq n R.C) &= \uparrow(\geq n R.C) \cup \{\geq n R'.C \mid R' \in \uparrow(R)\} \\
&\quad \cup \{\geq n R.C' \mid C' \in \zeta_{\uparrow, \downarrow}(C)\} \cup \{\geq n' R.C \mid n' \in \downarrow(n)\} \quad , \\
\zeta_{\uparrow, \downarrow}(\leq n R.C) &= \uparrow(\leq n R.C) \cup \{\leq n R'.C \mid R' \in \downarrow(R)\} \\
&\quad \cup \{\leq n R.C' \mid C' \in \zeta_{\downarrow, \uparrow}(C)\} \cup \{\leq n' R.C \mid n' \in \uparrow(n)\} \quad , \\
&\text{SRQIQ roles:} \\
\zeta_{\uparrow, \downarrow}(R) &= \uparrow(R) \quad .
\end{aligned}$$

From the abstract refinement operator $\zeta_{\uparrow, \downarrow}$, two concrete refinement operators, the generalization operator and specialization operator are, respectively, defined as

$$\begin{aligned}
\gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}} &= \zeta_{\text{UpCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}, \text{DownCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}} \quad \text{and} \\
\rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}} &= \zeta_{\text{DownCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}, \text{UpCover}_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}} \quad .
\end{aligned}$$

Revisiting the case in Example 2 we observe that $\gamma_{\mathcal{O}, \mathcal{O}}(C \sqcup A) = \{\top, \top \sqcup A, C \sqcup A, C \sqcup B\}$ does contain $C \sqcup B$ as a possible refinement. Similarly, $\gamma_{\mathcal{O}, \mathcal{O}}(\forall r.A) = \{\top, \forall r.A, \forall s.A, \forall r.B\}$ contains $\forall r.B$. We will show now some basic properties of $\gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}$ and $\rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}$ that will prove useful in the remainder of this paper.

Lemma 1. For every pair of SRQIQ ontologies $\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}$ and every pair of concepts or roles $X, Y \in \mathcal{L}(N_C, N_R, N_I) \cup \mathcal{L}(N_R)$:

1. **generalisation:** if $X \in \gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(Y)$ then $Y \sqsubseteq_{\mathcal{O}^{\text{ref}}} X$
specialisation: if $X \in \rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(Y)$ then $X \sqsubseteq_{\mathcal{O}^{\text{ref}}} Y$
2. **generalisation finiteness:** $\gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(X)$ is finite
specialisation finiteness: $\rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(X)$ is finite

We define now the axiom weakening operator using these generalization and specialization operators.

Definition 5. Given an axiom ϕ , the set of weakenings with respect to the reference ontology \mathcal{O}^{ref} and full ontology $\mathcal{O}^{\text{full}}$, written $g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi)$ is defined such that

$$\begin{aligned}
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(C \sqsubseteq D) &= \{C' \sqsubseteq D \mid C' \in \rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(C)\} \cup \{C \sqsubseteq D' \mid D' \in \gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(D)\} , \\
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(C(a)) &= \{C'(a) \mid C' \in \gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(C)\} , \\
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(R(a, b)) &= \{R'(a, b) \mid R' \in \gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(R)\} \cup \{R(a, b), \perp \sqsubseteq \top\} , \\
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\neg R(a, b)) &= \{\neg R'(a, b) \mid R' \in \rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(R)\} \cup \{\neg R(a, b), \perp \sqsubseteq \top\} , \\
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(a = b) &= \{a = b, \perp \sqsubseteq \top\} , \quad g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(a \neq b) = \{a \neq b, \perp \sqsubseteq \top\} , \\
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\text{disjoint}(R, S)) &= \{\text{disjoint}(R', S) \mid R' \in \rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(R)\} \\
&\quad \cup \{\text{disjoint}(R, S') \mid S' \in \rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(S)\} \\
&\quad \cup \{\text{disjoint}(R, S), \perp \sqsubseteq \top\} , \\
g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(S_1 \circ \dots \circ S_n \sqsubseteq R) &= \{S_1 \circ \dots \circ S'_i \circ \dots \circ S_n \sqsubseteq R \mid S'_i \in \rho_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(S_i) \text{ for } i = 1, \dots, n\} \\
&\quad \cup \{S_1 \sqsubseteq R' \mid R' \in \gamma_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}\text{ and } n = 1 \text{ and } S_1 \text{ is simple in } \mathcal{O}^{\text{full}}\} \\
&\quad \cup \{S_1 \circ \dots \circ S_n \sqsubseteq R, \perp \sqsubseteq \top\} .
\end{aligned}$$

The axioms in the set $g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi)$ are indeed weaker than ϕ for every axiom ϕ , in the sense that, given the reference ontology \mathcal{O}^{ref} , ϕ entails them and the opposite is not necessarily true.

Lemma 2. For every $SR\mathcal{O}IQ$ axiom ϕ , if $\phi' \in g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi)$, then $\phi \models_{\mathcal{O}^{\text{ref}}} \phi'$

Clearly, replacing an axiom in the full ontology with a weakening cannot reduce the number of models of the ontology. However, for the weakening to be useful in practice, we must show additionally that by adding the weakened axioms to the ontology will not violate any of the constraints that ensure the decidability of $SR\mathcal{O}IQ$. To do this, we show first that all roles that are simple in $\mathcal{O}^{\text{full}}$ are also simple in the ontology obtained by adding the weakening of any axiom.

Lemma 3. Let \mathcal{O} be an ontology such that all simple roles of $\mathcal{O}^{\text{full}}$ are also simple in \mathcal{O} . For every axiom $\phi \in \mathcal{O}$ and role R , if $\phi' \in g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi)$ and R simple in \mathcal{O} , then R is simple in $\mathcal{O} \cup \{\phi'\}$.

Note that removal of axioms will never cause a role that was previously simple to become non-simple, so all roles simple in \mathcal{O} are also simple in $\mathcal{O} \setminus \{\phi\} \cup \{\phi'\}$. Further, it should be noted that repeated replacement of axioms with weakened axioms keeps simple roles simple. This is an important observation, since repeated weakening is required for the proposed ontology repair algorithm.

Lemma 4. Let \mathcal{O} be an ontology such that all simple roles of $\mathcal{O}^{\text{full}}$ are also simple in \mathcal{O} . For every axiom $\phi \in \mathcal{O}$, if $\phi' \in g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi)$ and the $R\text{Box}$ of \mathcal{O} is regular, then the $R\text{Box}$ of $\mathcal{O} \cup \{\phi'\}$ is also regular.

Like for the invariant on simple roles, also regularity is maintained by repeated replacement of axioms with weakenings. With the help of Lemma 3 and Lemma 4 we can now show that adding weakened axioms to a $SR\mathcal{O}IQ$ ontology will yield another valid $SR\mathcal{O}IQ$ ontology.

Theorem 1. Given that \mathcal{O}^{ref} and $\mathcal{O}^{\text{full}}$ are valid $SR\mathcal{O}IQ$ ontologies. For every axiom $\phi \in \mathcal{O}^{\text{full}}$, if $\phi' \in g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi)$, then $\mathcal{O}^{\text{full}} \cup \{\phi'\}$ is a valid $SR\mathcal{O}IQ$ ontology.

3. Implementing and Evaluating Axiom Weakening for $SR\mathcal{OIQ}$

The refinement operators and axiom weakening have previously been implemented for \mathcal{ALC} in [5]. Based on this, we have extended the implementation to cover the full range of $SR\mathcal{OIQ}$ axioms and concepts.¹ The concept refinement and axiom weakening operators for $SR\mathcal{OIQ}$ have been implemented as discussed above. Further, we implemented a repair algorithm using the axiom weakening operator based on the procedures already proposed in [5] and [6]. The implementation performs weakening in OWL 2 DL [13] and is implemented in Java using the OWL API [14]. A plug-in for the ontology development tool Protégé has also been implemented, but will not be discussed in detail in this paper.² The plug-in allows for manually weakening axioms and executing the automatic repair algorithm.

Algorithm 1 RepairOntologyWeaken(\mathcal{O})

```

 $\mathcal{O}^{\text{full}} \leftarrow \mathcal{O}$ 
 $\mathcal{O}^{\text{ref}} \leftarrow \text{FindMaximalConsistentSubset}(\mathcal{O})$ 
while  $\mathcal{O}$  is inconsistent do
   $\phi_{\text{bad}} \leftarrow \text{FindBadAxiom}(\mathcal{O})$ 
   $\Phi_{\text{weaker}} \leftarrow g_{\mathcal{O}^{\text{ref}}, \mathcal{O}^{\text{full}}}(\phi_{\text{bad}})$ 
   $\phi_{\text{weaker}} \leftarrow \text{SelectWeakerAxiom}(\Phi_{\text{weaker}})$ 
   $\mathcal{O} \leftarrow (\mathcal{O} \setminus \{\phi_{\text{bad}}\}) \cup \{\phi_{\text{weaker}}\}$ 
end while
Return  $\mathcal{O}$ 

```

Algorithm 2 RepairOntologyRemove(\mathcal{O})

```

while  $\mathcal{O}$  is inconsistent do
   $\phi_{\text{bad}} \leftarrow \text{FindBadAxiom}(\mathcal{O})$ 
   $\mathcal{O} \leftarrow \mathcal{O} \setminus \{\phi_{\text{bad}}\}$ 
end while
Return  $\mathcal{O}$ 

```

The automatic repair by weakening is implemented as shown in Algorithm 1. The reference ontology is selected by choosing a maximal consistent subset of the inconsistent ontology. In our implementation used for the evaluation in this paper, the reference ontology was selected by randomly sampling a maximal consistent subset. The procedure $\text{FindBadAxiom}(\mathcal{O})$ may be implemented in a number of ways. Here we consider an implementation that samples some (or all) of the minimal inconsistent subsets of \mathcal{O} and selects as the bad axiom the one occurring most frequently. Then, $\text{SelectWeakerAxiom}(\Phi_{\text{weaker}})$ has been chosen such that it selects an axiom uniformly at random from Φ_{weaker} . Regarding termination, we showed that the corresponding algorithm for weakening for the \mathcal{ALC} fragment almost certainly terminates (the event in which it does not terminate has probability 0) [18], and we expect a similar result to hold for the extended weakening procedure for $SR\mathcal{OIQ}$, however, we here must leave the verification of the details to future work. For all experiments presented in this paper, the FaCT++ reasoner [19] was used to compute all entailment and consistency checks.

To experimentally evaluate the proposed axiom weakening operator in the context of its use in automatic repair of ontologies, we need some way to compare the quality of repair. As has already been discussed in [5], the problem of deciding which of two possible repaired ontologies \mathcal{O}_1 or \mathcal{O}_2 is preferable is not generally well-defined. Similar to what has been proposed in [5] we will base the evaluation of the repairs on the size of the *inferred class hierarchy*. To compare two possible repairs, we use the *inferable information content* (IIC) as defined in [5]. The IIC

¹The source code for the implementation is available at <https://github.com/rolandbernard/ontologyutils>.

²The Protégé plug-in is available at <https://github.com/rolandbernard/protege-weakening>.

Abbreviation	Name	Expressivity	Axioms	Concepts	Roles	Subconcepts
admin	Nurse Administrator	$ALCHOIF$	229	42	29	144
ahso	Animal Health Surveillance Ontology	$ALCRIF$	166	38	31	104
cdao	Comparative Data Analysis Ontology	$ALCROIQ$	437	132	68	375
cdpeo	Chronic Disease Patient Education	$ALCHF$	422	41	31	170
covid19-ibo	Covid19 Impact on Banking Ontology	$ALCH$	288	160	33	227
ecp	Electronic Care Plan	$ALCRQ$	127	33	17	99
emo	Enzyme Mechanism Ontology	$ALCHQ$	368	157	24	255
evi	Evidence Graph Ontology	$ALCRI$	143	30	38	69
falls	Falls Prevention	$ALCH$	79	30	20	35
fo	Fern Ontology	$ALCHI$	59	31	4	46
gbm	Glioblastoma	$ALCIF$	603	108	28	227
gfvo	Genomic Feature and Variation Ontology	$ALCH$	332	102	30	170
koro	Knowledge Object Reference Ontology	$ALCHI$	262	110	19	194
lico	Liver Case Ontology	$ALCHQ$	366	93	36	230
mamo	Mathematical Modelling Ontology	$ALCR$	229	107	3	154
mpio	Minimum PDDI Information Ontology	$ALCH$	38	30	14	45
provo	Provenance Ontology	$ALCRIN$	170	31	42	128
qudt	Quantities, Units, Dimensions, and Types	$SHOIQ$	293	74	73	177
trans	Nurse Transitional	$ALCROIQ$	244	44	22	123
trriage	Nurse triage	$ALCHF$	132	33	29	129
vio	Vaccine Investigation Ontology	$ALCRI$	249	81	44	235
pizza	Pizza Ontology	$SHOIN$	1131	100	8	376

Table 1

The ontologies used for evaluation. The number of axioms, concept names, role names, and subconcepts are taken after preprocessing.

of an ontology \mathcal{O}_1 with respect to a second ontology \mathcal{O}_2 , written $IIC(\mathcal{O}_1, \mathcal{O}_2)$, is a number between 0 and 1. A value closer to 1 indicates that \mathcal{O}_1 contains more “information” than \mathcal{O}_2 , while a value towards 0 indicates the opposite. Some weaknesses of this measure when it comes to evaluating repairs, like the fact that only atomic concepts are considered, have already been discussed in [5]. For the case of repairing $SRROIQ$ ontologies this is even more relevant, since the role hierarchy is entirely ignored.

Definition 6. The inferred class hierarchy of an ontology \mathcal{O} is given by

$$\text{Inf}(\mathcal{O}) = \{A \sqsubseteq B \mid A, B \in N_C \text{ and } \mathcal{O} \models A \sqsubseteq B\} .$$

The inferable information content of an ontology \mathcal{O}_1 with respect to another ontology \mathcal{O}_2 is given by

$$IIC(\mathcal{O}_1, \mathcal{O}_2) = \frac{\text{card}(\text{Inf}(\mathcal{O}_1) \setminus \text{Inf}(\mathcal{O}_2))}{\text{card}(\text{Inf}(\mathcal{O}_1) \setminus \text{Inf}(\mathcal{O}_2)) + \text{card}(\text{Inf}(\mathcal{O}_2) \setminus \text{Inf}(\mathcal{O}_1))} ,$$

where $\text{card}(X)$ is the cardinality of the set X .

For the experimental evaluation we have selected ontologies of varying size and expressivity from BioPortal³ [20]. Additionally, the pizza ontology⁴ was included in the testing. Some characteristics of the used ontologies are shown in Table 1. On average, they contain about 289 axioms, 73 concept names, 29 role names, and 168 subconcepts.

Since the ontologies use OWL 2, the axioms and concepts do not map directly to $SRROIQ$. In order to follow the definitions laid out in this paper, the OWL ontologies axioms were

³<https://biportal.bioontology.org/>

⁴Available from Protégé at <https://protege.stanford.edu/ontologies/pizza/pizza.owl>.

Abbreviation	IIC w.r.t. repair by removal	IIC w.r.t. maximal consistent subset
admin	0.53 [0.47; 0.59]	0.39 [0.31; 0.47]
ahso	0.56 [0.50; 0.62]	0.51 [0.44; 0.57]
cdao	0.53 [0.44; 0.61]	0.53 [0.45; 0.61]
cdpeo	0.50 [0.45; 0.55]	0.22 [0.16; 0.28]
covid19-ibo	0.70 [0.65; 0.75]	0.63 [0.57; 0.69]
ecp	0.74 [0.67; 0.81]	0.36 [0.28; 0.44]
emo	0.69 [0.63; 0.75]	0.60 [0.53; 0.66]
evi	0.49 [0.43; 0.55]	0.59 [0.53; 0.66]
falls	0.78 [0.71; 0.85]	0.49 [0.41; 0.58]
fo	0.50 [0.44; 0.57]	0.70 [0.62; 0.76]
gbm	0.59 [0.52; 0.66]	0.52 [0.44; 0.59]
gfvo	0.56 [0.49; 0.62]	0.54 [0.49; 0.60]
koro	0.51 [0.45; 0.57]	0.37 [0.29; 0.45]
lico	0.55 [0.48; 0.62]	0.53 [0.46; 0.60]
mamo	0.52 [0.44; 0.60]	0.68 [0.61; 0.74]
mpio	0.70 [0.63; 0.76]	0.73 [0.67; 0.78]
provo	0.50 [0.43; 0.56]	0.55 [0.49; 0.62]
qudt	0.96 [0.93; 0.99]	0.44 [0.35; 0.55]
trans	0.58 [0.52; 0.64]	0.43 [0.35; 0.52]
triage	0.53 [0.46; 0.60]	0.53 [0.46; 0.60]
vio	0.46 [0.37; 0.55]	0.49 [0.40; 0.57]
pizza	0.56 [0.49; 0.64]	0.61 [0.53; 0.68]
Overall	0.59 [0.57; 0.61]	0.52 [0.50; 0.54]

Table 2

Results of the evaluation. IIC is given as mean and 95% confidence interval in brackets.

normalized to conform with \mathcal{SROIQ} . During the preprocessing, we further removed axioms related to data properties and any axiom that caused any OWL 2 DL profile violation, as our weakening does not handle them.

The evaluation proceeds by first making the ontologies inconsistent. This was achieved by repeatedly adding axioms to the ontology until they became inconsistent. The newly added axioms were generated by strengthening randomly selected axioms of the original ontology. It was ensured that the added axioms on their own were not inconsistent. After making the ontology inconsistent, it was repaired once with the repair algorithm using the axiom weakening operator presented in Algorithm 1 and once using Algorithm 2. Additionally, the repair was also performed by selecting a randomly sampled maximal consistent subset. Note that Algorithm 2 is not guaranteed to produce a maximal consistent subset. This process, both making the ontology inconsistent and then repairing it, was repeated one hundred times for each ontology, and the IIC was computed between the repairs generated using the different approaches. The evaluation was performed using a randomly selected maximal consistent subset as the reference ontology and by sampling 16 minimal inconsistent subsets during the selection of bad axioms.

Unfortunately, even though the utilized reasoners are generally very fast to evaluate queries on the selected ontologies, they exhibit undesirable performance in some edge cases. When performance pitfalls are encountered, they make the computations required for weakening unreasonably slow. For this reason a timeout of 5 minutes was placed on the repairs execution and the outputs of these runs were discarded and replaced by new runs. The results of these experiments are listed in Table 2 and shown in Figure 1 and Figure 2.

The results of the evaluation suggest that the repair by weakening is on average about as good or better than the repair by removal of axioms. While this supports the conclusion in [5] that

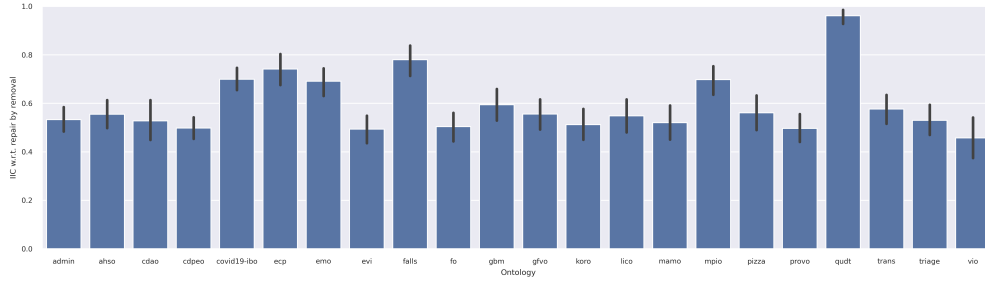


Figure 1: Mean IIC with respect to repair via removal per ontology. The error bars show the 95% confidence interval.

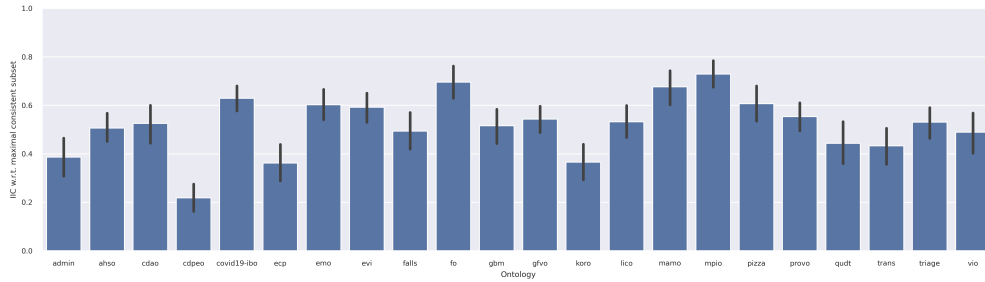


Figure 2: Mean IIC with respect to a random maximal consistent subset per ontology. The error bars show the 95% confidence interval.

weakening is able to retain more information than removal, the observed advantage was worse than what has been observed in [5]. In contrast, it can be seen that the repair using weakening is not in general better than choosing a random maximal consistent subset. There are ontologies for which the repairs by weakening are on average significantly worse when comparing using IIC. This is however a somewhat unequal comparison. An alternative repair algorithm could start with a maximal consistent subset and use weakening to add in more information from the remaining axioms. Still, this result suggests that the heuristic used for selecting bad axioms is not reliable for preserving information, at least with respect to the chosen measure.

4. Conclusions and Outlook

We have proposed refinement operators and an axiom weakening operator for all aspects of *SR_{OLQ}* and shown that for repairs of inconsistent ontologies weakening can, in some cases, significantly outperform removal. Further additions to the refinement operators may be studied, e.g., using non-simple roles in the upward and downward covers in certain contexts. Relaxing the allowed weakening for RIAs may also be considered, and to cover also extensions to regularity conditions such as those studied in [21]. We have also seen that the repair algorithm likely needs better heuristics to steer the selection of bad and weakened axioms in order to result in better repairs. Future work could further focus on finding more robust measures for comparing the quality of repairs.

References

- [1] S. Schlobach, R. Cornet, Non-standard reasoning services for the debugging of description logic terminologies, in: G. Gottlob, T. Walsh (Eds.), IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003, Morgan Kaufmann, 2003, pp. 355–362.
- [2] A. Kalyanpur, B. Parsia, E. Sirin, J. Hendler, Debugging unsatisfiable classes in OWL ontologies, *Web Semantics: Science, Services and Agents on the World Wide Web 3* (2005) 268–293.
- [3] A. Kalyanpur, B. Parsia, E. Sirin, B. C. 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.
- [4] F. Baader, R. Peñaloza, B. Suntisrivaraporn, Pinpointing in the description logic \mathcal{EL}^+ , in: J. Hertzberg, M. Beetz, R. Englert (Eds.), *KI 2007: Advances in Artificial Intelligence*, 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 10-13, 2007, Proceedings, volume 4667 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 52–67.
- [5] 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), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 1981–1988.
- [6] 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, CEUR-WS, 2020.
- [7] J. Du, G. Qi, X. Fu, A practical fine-grained approach to resolving incoherent owl 2 dl terminologies, in: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 919–928.
- [8] R. Confalonieri, M. Eppe, M. Schorlemmer, O. Kutz, R. Peñaloza, E. Plaza, Upward Refinement Operators for Conceptual Blending in the Description Logic \mathcal{EL}^{++} , *Annals of Mathematics and Artificial Intelligence* 82 (2018) 69–99.
- [9] F. Baader, F. Kriegel, A. Nuradiansyah, R. Peñaloza, Making repairs in description logics more gentle, in: *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.
- [10] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible \mathcal{SROIQ} , in: P. Doherty, J. Mylopoulos, C. A. Welty (Eds.), *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning*, Lake District of the United Kingdom, June 2-5, 2006, AAAI Press, 2006, pp. 57–67.
- [11] Y. Kazakov, \mathcal{RIQ} and \mathcal{SROIQ} Are Harder than \mathcal{SHOIQ} , in: G. Brewka, J. Lang (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008*, Sydney, Australia, September 16-19, 2008, AAAI Press, 2008, pp. 274–284.

- [12] F. Baader, I. Horrocks, C. Lutz, U. Sattler, *An Introduction to Description Logic*, Cambridge University Press, 2017.
- [13] OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition) (2012). URL: <http://www.w3.org/TR/owl2-syntax/>.
- [14] M. Horridge, S. Bechhofer, The OWL API: A Java API for OWL Ontologies, *Semantic Web 2* (2011) 11–21.
- [15] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant semantics for description logics, in: P. Hitzler, T. Lukasiewicz (Eds.), *Web Reasoning and Rule Systems - Fourth International Conference*, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010. Proceedings, volume 6333 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 103–117.
- [16] S. Rudolph, Foundations of description logics, Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures 7 (2011) 76–136.
- [17] F. Baader, Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles, in: G. Gottlob, T. Walsh (Eds.), *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 9-15, 2003, Morgan Kaufmann, 2003, pp. 319–324.
- [18] R. Confalonieri, P. Galliani, O. Kutz, D. Porello, G. Righetti, N. Troquard, Almost Certain Termination for \mathcal{ALC} Weakening, in: G. Marreiros, B. Martins, A. Paiva, B. Ribeiro, A. Sardinha (Eds.), *Progress in Artificial Intelligence: 21st EPIA Conference on Artificial Intelligence*, EPIA 2022, Lisbon, Portugal, August 31–September 2, 2022, volume 13566 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 663–675.
- [19] D. Tsarkov, I. Horrocks, Fact++ description logic reasoner: System description, in: *Automated Reasoning: Third International Joint Conference*, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006. Proceedings 3, Springer, 2006, pp. 292–297.
- [20] P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, M. A. Musen, Biportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications, *Nucleic acids research* 39 (2011) W541–W545.
- [21] Y. Kazakov, An Extension of Complex Role Inclusion Axioms in the Description Logic \mathcal{SROIQ} , in: J. Giesl, R. Hähnle (Eds.), *Automated Reasoning*, 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16-19, 2010. Proceedings, volume 6173 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 472–486.