Interactive ontology revision ☆

Nadeschda Nikitina [a,*], Sebastian Rudolph [a], Birte Glimm [b,1]

[a] Institute AIFB, Karlsruhe Institute of Technology, Building 11.40, Englerstr. 11, Karlsruhe D-76131, Germany
[b] Ulm University, Institute of Artificial Intelligence, Building O27, Room 448, James-Franck-Ring, Ulm D-89081, Germany

ARTICLE INFO

ABSTRACT

When ontological knowledge is acquired automatically, quality control is essential. Which part of the automatically acquired knowledge is appropriate for an application often depends on the context in which the knowledge base or ontology is used. In order to determine relevant and irrelevant or even wrong knowledge, we support the tightest possible quality assurance approach – an exhaustive manual inspection of the acquired data. By using automated reasoning, this process can be partially automatized: after each expert decision, axioms that are entailed by the already confirmed statements are automatically approved, whereas axioms that would lead to an inconsistency are declined.

Starting from this consideration, this paper provides theoretical foundations, heuristics, optimization strategies and comprehensive experimental results for our approach to efficient reasoning-supported interactive ontology revision.

We introduce and elaborate on the notions of revision states and revision closure as formal foundations of our method. Additionally, we propose a notion of axiom impact which is used to determine a beneficial order of axiom evaluation in order to further increase the effectiveness of ontology revision. The initial notion of impact is then further refined to take different validity ratios – the proportion of valid statements within a dataset – into account. Since the validity ratio is generally not known *a priori* – we show how one can work with an estimate that is continuously improved over the course of the inspection process.

Finally, we develop the notion of *decision spaces*, which are structures for calculating and updating the revision closure and axiom impact. We optimize the computation performance further by employing partitioning techniques and provide an implementation supporting these optimizations as well as featuring a user front-end. Our evaluation shows that our ranking functions almost achieve the maximum possible automatization and that the computation time needed for each reasoning-based, automatic decision takes less than one second on average for our test dataset of over 25000 statements.

1. Introduction

Many real-world applications in the Semantic Web make use of ontologies, also called knowledge bases, in order to enrich the semantics of the data on which the application is based. As a popular example, consider DBpedia, which consists of structured information extracted from Wikipedia. DBpedia uses a background ontology defining the meaning of and relationships between terms. For example, if two terms are related via the property *river*, the first one can be inferred to be an instance of the class *Place* and the latter one of the class *River*.

In order to ensure a very high quality, the DBpedia background ontology has been created manually. For many applications, however, the time costs of a completely manual knowledge acquisition process are too high. Thus, the additional application of (semi-) automatic knowledge acquisition methods such as ontology learning or matching techniques is often considered a reasonable way to reduce the expenses of ontology development. The results produced by such automatic methods usually need to be manually inspected either partially, in order to estimate the overall quality of the resulting data, or to the full extent, in order to keep the quality of the developed ontology under control. Even when we aim at taking an ontology that has been developed in another area or as part of another application, it might not be appropriate to simply use the ontology as it is since statements that hold in the context of one application not necessarily hold in a different context.

So far, knowledge representation (KR) research has been focusing on restoring the consistency of knowledge bases enriched with new axioms as done in various belief revision and repair

---

approaches, see, e.g., [3–7]. Thereby, new axioms not causing an inconsistency are accepted as valid facts not requiring further inspection. We aim at a more restrictive quality control process in which a domain expert inspects a set of candidate axioms and decides for each of them whether it is a desired logical consequence. We call this exhaustive manual inspection of the acquired data *ontology revision*. If we assume that the deductive closure of the confirmed statements must be disjoint from the set of declined statements, then this process can be partially automatized: based on the decisions taken by the expert, we can automatically discard or include yet unevaluated axioms depending on their logical relationships with the already evaluated axioms. On the one hand, we can automatically approve axioms that are entailed by the already confirmed statements, since declining them would violate our assumption. On the other hand, we can automatically decline axioms that would cause any of the declined axioms to become a consequence of the confirmed ones, since accepting them would again violate our assumption.

Throughout the paper, we use the following running example written in OWL's functional-style syntax [8]. We use an imaginary prefix ex to abbreviate IRIs:

Example 1. Consider the ontology in Table 1. Let us assume that we have already confirmed that the axioms in the upper part, which state subclass relations between classes, belong to the desired consequences.

We further assume that Axioms (1)–(8) in the lower part, which define several different types for the individual ex:nanotube1, are still to be evaluated.

If Axiom (8) is declined, we can immediately also decline Axioms (1)–(6) assuming OWL or RDFS reasoning since accepting the axioms would implicitly lead to the undesired consequence (8). Note that no automatic decision is possible for Axiom (7) since it is not a consequence of Axiom (8) and the already approved subsumption axioms. Similarly, if Axiom (1) is approved, Axioms (2)–(8) are implicit consequences, which can be approved automatically. If we start, however, with declining Axiom (1), no automatic evaluation can be performed.

In the previous example, we only made decisions about class assertion axioms since we assumed that all subclass axioms were already approved or part of an already established ontology. This is, however, not a restriction of the approach. The following example shows that we can also make decisions about terminological axioms in the process of revising an ontology. We characterize the formalisms and kinds of axioms to which our approach can be applied more precisely in the following section. We use imaginary prefixes a and b to abbreviate IRIs in this example:

Table 1
An example ontology from the nano technology domain.

SubClassOf(ex:AluminiumNitrideNanotube ex:AluminiumNitride)
SubClassOf(ex:AluminiumNitride ex:NonOxideCeramics)
SubClassOf(ex:NonOxideCeramics ex:Ceramics)
SubClassOf(ex:Ceramics ex:MaterialByMaterialClass)
SubClassOf(ex:MaterialByMaterialClass ex:Material)
SubClassOf(ex:Material ex:PortionOfMaterial)
SubClassOf(ex:Material ex:TangibleObject)

ClassAssertion(ex:AluminiumNitrideNanotube ex:nanotube1)	(1)
ClassAssertion(ex:AluminiumNitride ex:nanotube1)	(2)
ClassAssertion(ex:NonOxideCeramics ex:nanotube1)	(3)
ClassAssertion(ex:Ceramics ex:nanotube1)	(4)
ClassAssertion(ex:MaterialByMaterialClass ex:nanotube1)	(5)
ClassAssertion(ex:Material ex:nanotube1)	(6)
ClassAssertion(ex:PortionOfMaterial ex:nanotube1)	(7)
ClassAssertion(ex:TangibleObject ex:nanotube1)	(8)

Example 2. Let us assume that we have already approved the axioms in the upper part of Table 2, which state subclass relations between classes. We further assume that Axioms (9)–(11) in the lower part of Table 2 are still to be evaluated. If Axiom (9) is approved, we can immediately also approve Axiom (10) since it is already a consequence of the approved axioms: a:Employee is interpreted as a subset of the extension of a:Person and b:Lecture is interpreted as a subset of the extension of b:Event, but if a:Person and b:Event are disjoint due the just approved Axiom (9) then so are a:Employee and b:Lecture. Moreover, we can decline Axiom (11), since approving this axiom would implicitly lead to incoherency, again since a:Ordinary and b:Ordinary have to be interpreted as subsets of disjoint sets and can, therefore, not be equivalent.

From the above examples, it can be observed that

- a high grade of automation requires a good evaluation order and
- approval and decline of an axiom has a different impact.

Which axioms have the highest impact on decline or approval and which axioms can be automatically evaluated once a particular decision has been made can be determined with the help of algorithms for automated reasoning, e.g., for RDFS or OWL reasoning. One of the difficulties is, however, that it is not known in advance, which of the two decisions the domain expert takes. We show that, in some cases, a realistic prediction about the decision of the user can be made: if the quality is fairly high, the user is likely to approve an axiom. Hence, axioms that have a high impact on approval (approval impact) should be evaluated with higher priority. For low quality data, the situation is reversed, i.e., axioms that have a high impact on decline (decline impact) should be considered first. We measure the quality by means of the *validity ratio*, i.e., the proportion of (manually and automatically) accepted axioms, and show that, depending on the validity ratio of a dataset, different impact measures used for axiom ranking are beneficial. While approval and decline impact measures yield fairly good results for validity ratios close to 100% or 0%, the optimality of results is left to chance in case of validity ratios close to 50%. To close this gap, we introduce an advanced ranking function based on these simple impact measures but parametrized by an estimated validity ratio. In our evaluation, we show that the revision based on the novel ranking function almost achieves the maximum possible automation. In particular the parametrized ranking functions achieve very good results for arbitrary validity ratios.

Further, since the expected validity ratio is usually not known in advance, we suggest a ranking function where the validity ratio is learned on-the-fly during the revision. We show that, even for small datasets (50–100 axioms), it is worthwhile to rank axioms based on this learned validity ratio instead of evaluating them in a random order. Furthermore, we show that, in case of larger datasets (e.g., 5000 axioms and more) with an unknown validity ratio, learning the validity ratio is particularly effective (with only 0.3% loss of effectiveness) due to the law of large numbers, thereby making the assumption of a known or expected validity ratio unnecessary. For such datasets, our experiments show that the proportion of automatically evaluated axioms when learning the

Table 2
An example ontology from the enterprise domain.

SubClassOf(a:Ordinary a:Employee)
SubClassOf(a:Employee a:Person)
SubClassOf(b:Ordinary b:Lecture)
SubClassOf(b:Lecture b:Event)

DisjointClasses(a:Person b:Event)	(9)
DisjointClasses(a:Employee b:Lecture)	(10)
EquivalentClasses(a:Ordinary b:Ordinary)	(11)

validity ratio is nearly the same (difference of 0.3%) as in case where the validity ratio is known in advance.

Even for light-weight knowledge representation formalisms, reasoning is often expensive and in an interactive setting it is crucial to minimize the number of reasoning tasks while still maximizing the number of automated decisions. Inspired by the techniques used to optimize ontology classification [9], we reduce the number of reasoning tasks by introducing the notion of *decision spaces* – auxiliary data structures that allow for storing the results of reasoning and reading-off the impact that an axiom will have upon approval or decline. Decision spaces exploit the characteristics of the logical entailment relation between axioms to maximize the amount of information gained by reasoning, and, therefore, in particular in case of logics for which entailment checking is not tractable, decision spaces reduce the computational effort. In addition to the performance gain achieved by using decision spaces, we show that partitioning – dividing the datasets under revision into logically independent subsets – further decreases the number of required reasoning calls.

We implemented the proposed techniques in the tool *revision helper*, which even for expressive OWL reasoning and our dataset of 25,000 axioms requires on average only 0.84 s (7.4 reasoning calls) per expert decision, where the automatic evaluation significantly reduces the number of expert decisions.

From our evaluation, it can be observed that, on the one hand, a considerable proportion (up to 80%) of axioms can be evaluated automatically by our revision support, and, on the other hand, an application of decision spaces and partitioning significantly reduces the number of required reasoning operations, resulting in a considerable performance gain – 83% of reasoning calls could be avoided.

The paper is organized as follows: In Section 2, we formalize the basic notions of reasoning-supported ontology revision. In Section 3, we define decision spaces and show how they can be updated during the revision. Section 4 describes the proposed parameterized ranking function. Section 5 introduces the partitioning optimization and we evaluate the approach in Section 6. We then present the user front-end *revision helper* in Section 7 and discuss existing related approaches in Section 8 before we conclude in Section 9.

The paper combines and extends previous work [1,2] with full proofs.

2. Revision of ontologies

The approach proposed here is not specific to a particular KR formalism. The only requirement for the chosen formalism is that it is logic-based, there is a procedure for deciding whether an ontology entails an axiom, and that taking all consequences is a closure operation. The latter means that, for $\mathcal{O}$, $\mathcal{O}'$ ontologies, the underlying entailment relation, denoted $\models$, has the following properties[2]:

1. it is *extensive*, i.e., any statement logically follows from itself: $\{\alpha\} \models \alpha$,
2. it is *monotone*, i.e., adding further statements does not invalidate previous consequences: $\mathcal{O} \models \alpha$ implies $\mathcal{O} \cup \mathcal{O}' \models \alpha$, and
3. it is *idempotent*, i.e., extending an ontology with an entailed axiom does not yield new consequences: $\mathcal{O} \models \alpha$ and $\mathcal{O} \cup \{\alpha\} \models \beta$ imply $\mathcal{O} \models \beta$.

Most commonly used ontology languages for the Semantic Web such as RDFS or OWL fulfill these requirements. For OWL ontologies we are, however, restricted to the Description Logic based OWL Direct Semantics [10] since we require the existence of a decision procedure for logical entailment. The OWL 2 RL profile [11] allows for

decidable reasoning under OWL's RDF-Based Semantics, but it is defined in terms of rules that materialize only *certain* inferences, namely assertions about individuals in the ontology such as class memberships or property relations between individuals. Other entailed consequences are not necessarily derived by the OWL 2 RL rules. Thus, completeness can only be guaranteed when the entailment checks are only for assertions about individuals in the ontology and our approach is only applicable for revising assertions, whereas, in general, any kind of axiom can be used in the revision.

The revision of an ontology $\mathcal{O}$ aims at a separation of its axioms (i.e., logical statements) into two disjoint sets: the set of wanted consequences $\mathcal{O}^{\models}$ and the set of unwanted consequences $\mathcal{O}^{\not\models}$. This motivates the following definitions.

Definition 1 (*Revision State*). A *revision state* is defined as a tuple $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ of ontologies with $\mathcal{O}^{\models} \subseteq \mathcal{O}$, $\mathcal{O}^{\not\models} \subseteq \mathcal{O}$, and $\mathcal{O}^{\models} \cap \mathcal{O}^{\not\models} = \emptyset$. Given two revision states $(\mathcal{O}, \mathcal{O}_1^{\models}, \mathcal{O}_2^{\not\models})$ and $(\mathcal{O}, \mathcal{O}_2^{\models}, \mathcal{O}_2^{\not\models})$, we call $(\mathcal{O}, \mathcal{O}_2^{\models}, \mathcal{O}_2^{\not\models})$ a *refinement* of $(\mathcal{O}, \mathcal{O}_1^{\models}, \mathcal{O}_2^{\not\models})$, if $\mathcal{O}_1^{\models} \subseteq \mathcal{O}_2^{\models}$ and $\mathcal{O}_1^{\not\models} \subseteq \mathcal{O}_2^{\not\models}$. A revision state is *complete*, if $\mathcal{O} = \mathcal{O}^{\models} \cup \mathcal{O}^{\not\models}$, and *incomplete* otherwise. An incomplete revision state $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ can be refined by evaluating a further axiom $\alpha \in \mathcal{O} \setminus (\mathcal{O}^{\models} \cup \mathcal{O}^{\not\models})$, obtaining $(\mathcal{O}, \mathcal{O}^{\models} \cup \{\alpha\}, \mathcal{O}^{\not\models})$ or $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models} \cup \{\alpha\})$. We call the resulting revision state an *elementary refinement* of $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$.

We introduce the notion of *consistency* of revision states to express the condition that the deductive closure of the wanted consequences in $\mathcal{O}^{\models}$ must not contain unwanted consequences. If we want to maintain consistency, a single evaluation decision can predetermine the decision for several yet unevaluated axioms. These implicit consequences of a refinement are captured in the *revision closure*.

Definition 2 (*Revision State Consistency & Closure*). A (complete or incomplete) revision state $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is *consistent* if there is no $\alpha \in \mathcal{O}^{\not\models}$ such that $\mathcal{O}^{\models} \models \alpha$. The *revision closure* $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ of $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is $(\mathcal{O}, \mathcal{O}_c^{\models}, \mathcal{O}_c^{\not\models})$ with $\mathcal{O}_c^{\models} := \{\alpha \in \mathcal{O} | \mathcal{O}^{\models} \models \alpha\}$ and $\mathcal{O}_c^{\not\models} := \{\alpha \in \mathcal{O} | \mathcal{O}^{\models} \cup \{\alpha\} \models \beta \text{ for some } \beta \in \mathcal{O}^{\not\models}\}$.

Note that, in order to be able to maintain the consistency of a revision state, $\mathcal{O}_c^{\not\models}$ must contain all axioms that, in case of an accept, would lead to an entailment of any unwanted consequences. We can show the following useful properties of the closure of consistent revision states:

Lemma 1. *For $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ a consistent revision state,*

1. $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is consistent,*
2. *every elementary refinement of* $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is consistent,*
3. *every consistent and complete refinement of* $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ *is a refinement of* $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$.

Proof. The first claim is immediate by the definition of consistency and closures of revisions. For the second claim, $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is consistent by assumption and $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is then consistent (by the first claim). Since $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is a closure of $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$, we have $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models}) = (\mathcal{O}, \{\alpha \in \mathcal{O} | \mathcal{O}^{\models} \models \alpha\}, \{\alpha \in \mathcal{O} | \mathcal{O}^{\models} \cup \{\alpha\} \models \beta \text{ for some } \beta \in \mathcal{O}^{\not\models}\})$. Since an elementary revision of $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ has to be for an axiom $\alpha \in \mathcal{O} \setminus (\{\beta | \mathcal{O}^{\models} \models \beta\} \cup \{\beta | \mathcal{O}^{\not\models} \cup \beta \models \gamma \text{ for some } \gamma \in \mathcal{O}^{\not\models}\})$, we immediately get that the elementary refinement is consistent. For the last claim, if $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is already complete, the claim trivially holds. Otherwise, since $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is consistent, we cannot make elementary refinements that add an axiom $\alpha \in \{\beta | \mathcal{O}^{\models} \models \beta\}$ to $\mathcal{O}^{\not\models}$ since this would result in an inconsistent refinement, neither can we add an axiom $\alpha \in \{\beta | \mathcal{O}^{\models} \cup \beta \models \gamma \text{ for some } \gamma \in \mathcal{O}^{\not\models}\}$ to $\mathcal{O}^{\models}$. Thus, a complete and consistent refinement of $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ is a refinement of $\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$. $\square$

---

[2] Throughout this paper, we assume that an ontology is a set of axioms, and will use α, β, and γ to denote axioms.

Algorithm 1 employs the above properties to implement a general methodology for interactive ontology revision.

Algorithm 1. Interactive Ontology Revision

Input: $(\mathcal{O}, \mathcal{O}_0^{\vDash}, \mathcal{O}_0^{\nvDash})$ a consistent revision state
Output: $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ a complete and consistent revision state
1: $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) \leftarrow \mathrm{clos}(\mathcal{O}, \mathcal{O}_0^{\vDash}, \mathcal{O}_0^{\nvDash})$
2: **while** $\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash} \neq \mathcal{O}$ **do**
3:     choose $\alpha \in \mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash})$
4:     **if** expert confirms α **then**
5:         $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) \leftarrow \mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash})$
6:     **else**
7:         $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) \leftarrow \mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\})$
8:     **end if**
9: **end while**

Instead of starting with empty $\mathcal{O}_0^{\vDash}$ and $\mathcal{O}_0^{\nvDash}$, we can initialize these sets with approved and declined axioms from a previous revision or add axioms of the ontology that is being developed to $\mathcal{O}_0^{\vDash}$. We can further initialize $\mathcal{O}_0^{\nvDash}$ with axioms expressing inconsistency and unsatisfiability of predicates (i.e. of classes or relations) in $\mathcal{O}$, which we assume to be unwanted.

In line 3, an axiom is chosen that is evaluated next. As mentioned earlier, choosing randomly can have a detrimental effect on the number of manual decisions needed. Ideally, we want to rank the axioms and choose one that allows for a high number of consequential automatic decisions. The notion of *axiom impact* refers to the number of axioms that can be automatically evaluated upon approval or decline of an axiom. Note that after an approval, the closure might extend both $\mathcal{O}^{\vDash}$ and $\mathcal{O}^{\nvDash}$, whereas after a decline only $\mathcal{O}^{\nvDash}$ can be extended. We further define $?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ as the number of yet unevaluated axioms and write $|S|$ to denote the cardinality of a set S:

Definition 3 (*Impact*). Let $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ be a consistent revision state with $\alpha \in \mathcal{O}$ and let $?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) := |\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash})|$. For an axiom α, we define its *approval impact*, $impact^+(\alpha)$, its *decline impact*, $impact^-(\alpha)$, and its *guaranteed impact, guaranteed*(α) :

$$impact^+(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$$
$$\qquad - ?(\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash})),$$
$$impact^-(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$$
$$\qquad - ?(\mathrm{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\})),$$
$$guaranteed(\alpha) = \min(impact^+(\alpha), impact^-(\alpha)).$$

We further separate $impact^+(\alpha)$ into the number of *automatic approvals*, $impact^{+a}(\alpha)$, and the number of *automatic declines*, $impact^{+d}(\alpha)$:

$$impact^{+a}(\alpha) = |\{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta\}|,$$
$$impact^{+d}(\alpha) = |\{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \vDash \gamma, \quad \text{for some } \gamma \in \mathcal{O}^{\nvDash}\}|.$$

Note that $impact^+(\alpha) = impact^{+a}(\alpha) + impact^{+d}(\alpha)$. Ranking axioms by $impact^+$ privileges axioms for which the number of automatically evaluated axioms in case of an accept is high. Going back to our running example, Axiom (1), which yields 7 automatically accepted axioms in case it is approved, will be ranked highest. The situation is the opposite for $impact^-$, which privileges axioms for which the number of automatically evaluated axioms in case of a decline is high (Axioms (7) and (8)). Ranking by *guaranteed* privileges axioms with the highest guaranteed impact, i.e., axioms with the highest number of automatically evaluated axioms in the worst-case (Axioms (4) and (5)).

Table 3 lists the values for all ranking functions for the axioms from Example 1.

Since computing such an impact as well as computing the closure after each evaluation (lines 1, 5, and 7) can be considered very expensive, we next introduce *decision spaces*, auxiliary data structures which significantly reduce the cost of computing the closure upon elementary revisions and provide an elegant way of determining high impact axioms.

3. Decision spaces

Intuitively, the purpose of decision spaces is to keep track of the dependencies between the axioms in such a way, that we can read-off the consequences of revision state refinements upon an approval or a decline of an axiom, thereby reducing the required reasoning operations. Furthermore, we will show how we can update these structures after a refinement step avoiding many costly recomputations.

Definition 4 (*Decision Space*). Given a revision state $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ with $\mathcal{O}^{\nvDash} \neq \emptyset$, the according *decision space* $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})} = (\mathcal{O}^?, E, C)$ contains the set
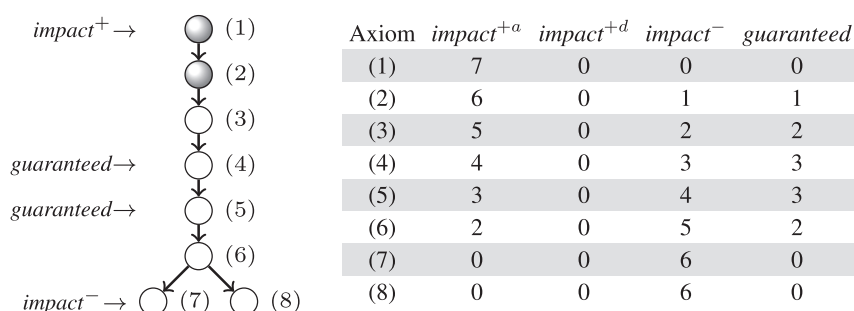
$$\mathcal{O}^? := \mathcal{O} \setminus (\{\alpha | \mathcal{O}^{\vDash} \vDash \alpha\} \cup \{\alpha | \mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta, \beta \in \mathcal{O}^{\nvDash}\})$$

of unevaluated axioms and two binary relations, E (entails) and C (conflicts) defined by

$\alpha E \beta$ iff $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta$ and
$\alpha C \beta$ iff $\mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \vDash \gamma$ for some $\gamma \in \mathcal{O}^{\nvDash}$.

The requirement that $\mathcal{O}^{\nvDash} \neq \emptyset$ is without loss of generality since we can always add an axiom that expresses an inconsistency,

Table 3
Example axiom dependency graph and the corresponding ranking values.

Axiom	$impact^{+a}$	$impact^{+d}$	$impact^-$	$guaranteed$
(1)	7	0	0	0
(2)	6	0	1	1
(3)	5	0	2	2
(4)	4	0	3	3
(5)	3	0	4	3
(6)	2	0	5	2
(7)	0	0	6	0
(8)	0	0	6	0

which is clearly undesired. On the other hand, the non-emptiness condition ensures that two axioms which together lead to an inconsistency are indeed recognized as conflicting. For example, consider the following two axioms:

$$SameIndividual(ex : a\ ex : b) \tag{12}$$

$$DifferentIndividuals(ex : a\ ex : b) \tag{13}$$

We assume that Axiom (12) has just been approved and belongs, therefore, to $\mathcal{O}^{\vDash}$, whereas Axiom (13) is a not yet evaluated axiom. Clearly, Axiom (12) and (13) cannot be true at the same time and, consequently, the inconsistent ontology $\mathcal{O}^{\vDash} \cup \{(13)\}$ entails any axiom, but, unless we have some axiom β in $\mathcal{O}^{\nvDash}$, this will not be recognized.

As a direct consequence of this definition, we have $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})} = \mathbb{D}_{\text{clos}(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})}$. The following properties follow immediately from the above definition:

Lemma 2. *For any decision space* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})} = (\mathcal{O}^?, E, C)$, *the following hold:*

P1 $(\mathcal{O}^?, E)$ *is a quasi-order (i.e., reflexive and transitive),*
P2 C *is symmetric,*
P3 $\alpha E\beta$ *and* $\beta C\gamma$ *imply* $\alpha C\gamma$ *for all* α, β, $\gamma \in \mathcal{O}^?$, *and*
P4 *if* $\alpha E\beta$ *then* $\alpha C\beta$ *does not hold.*

Proof. For P1, due to the required properties of the underlying logic we have $\{\alpha\} \vDash \alpha$ (extensivity) and $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \alpha$ (monotonicity) and it follows that E is reflexive. Given $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta$ and $\mathcal{O}^{\vDash} \cup \{\beta\} \vDash \gamma$, idempotency ensures $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \gamma$, hence E is transitive. For P2, symmetry of C is an immediate consequence from its definition. For showing P3, suppose $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta$ and $\mathcal{O}^{\vDash} \cup \{\beta, \gamma\} \vDash \delta$ for some $\delta \in \mathcal{O}^{\nvDash}$. Monotonicity allows to get $\mathcal{O}^{\vDash} \cup \{\alpha, \gamma\} \vDash \beta$ from the former and $\mathcal{O}^{\vDash} \cup \{\alpha, \beta, \gamma\} \vDash \delta$ from the latter, whence $\mathcal{O}^{\vDash} \cup \{\alpha, \beta, \gamma\} \vDash \delta$ follows via idempotency. To see that E and C are mutually exclusive (P4), assume the contrary, i.e., $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta$ and $\mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \vDash \gamma$ for some $\gamma \in \mathcal{O}^{\nvDash}$ hold simultaneously. Yet, idempotency allows to conclude $\mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \delta$. However then α cannot be contained in $\mathcal{O}^?$ by definition, which gives a contradiction and proves the claim. $\square$

In fact, the properties established in Lemma 2 are characteristic. This means that no other than the above established properties hold for decision spaces. We show this by proving that any structure satisfying these properties can be seen as the decision space for an appropriate revision state[3]:

Lemma 3. *Let V be finite set and let $\underline{E}$, $\underline{C} \subseteq V \times V$ be relations for which $(V, \underline{E})$ is a quasi-order, $\underline{C} = \underline{C}^-$, $\underline{E} \circ \underline{C} \subseteq \underline{C}$ and $\underline{E} \cap \underline{C} = \emptyset$. Then there is a decision space $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})}$ isomorphic to $(V, \underline{E}, \underline{C})$.*

Proof. As a very basic formalism, we choose propositional logic as KR language. Let $\mathcal{O}$ contain one atomic proposition p_v for every $v \in V$, let $\mathcal{O}^{\vDash} = \{p_v \rightarrow p_{v'} | v\underline{E}v'\} \cup \{\neg p_v \vee \neg p_{v'} | v\underline{C}v'\}$ and let $\mathcal{O}^{\nvDash} = \{false\}$. First observe that $\mathcal{O}^? = \mathcal{O}$. Next, we claim that the function $f : V \rightarrow \mathcal{O}$ with $v \mapsto p_v$ is an isomorphism between $(V, \underline{E}, \underline{C})$ and $?D_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})}$. Clearly, f is a bijection. Moreover, $v\underline{E}v'$ implies $p_v E p_{v'}$ by modus ponens since $p_v \rightarrow p_{v'} \in \mathcal{O}^{\vDash}$. Likewise, $v\underline{C}v'$ implies $p_v C p_{v'}$ due to $\neg p_v \vee \neg p_{v'} \in \mathcal{O}^{\vDash}$. The two other directions are shown indirectly.

To show that $p_v E p_{v'}$ implies $v\underline{E}v'$ assume there are p_v, $p_{v'}$ with $p_v E p_{v'}$, but $v\underline{E}v'$ does not hold. Now, consider the propositional interpretation mapping $p_{\tilde{v}}$ to *true* whenever $\tilde{v} \in \uparrow v$ and to *false* otherwise. It can be verified that this interpretation is a model of $\mathcal{O}^{\vDash}$ and satisfies p_v as well as $\neg p_{v'}$, hence $\mathcal{O}^{\vDash} \cup \{p_v\} \nvDash p_{v'}$ and consequently $p_v E p_{v'}$ cannot hold, so we have a contradiction.

[3] As usual, we let $R^- = \{(y, x) | (x, y) \in R\}$ as well as $R \circ S = \{(x, z) | (x, y) \in R, (y, z) \in S$ for some $y\}$.

To show that $p_v C p_{v'}$ implies $v\underline{C}v'$ assume there are p_v, $p_{v'}$ with $p_v C p_{v'}$, but $v\underline{C}v'$ does not hold. Now, consider the propositional interpretation mapping $p_{\tilde{v}}$ to *true* whenever $\tilde{v} \in \uparrow v \cup \uparrow v'$ and to *false* otherwise. It can be verified that this interpretation is a model of $\mathcal{O}^{\vDash}$ and satisfies p_v as well as $p_{v'}$, hence $\mathcal{O}^{\vDash} \cup \{p_v, p_{v'}\} \nvDash false$ and consequently $p_v C p_{v'}$ cannot hold, so we have a contradiction. $\square$

The following lemma shows how decision spaces can be used for calculating closures of updated revision states and impacts of axioms. As usual for (quasi)orders, we define $\uparrow\alpha = \{\beta | \alpha E\beta\}$ and $\downarrow\alpha = \{\beta | \beta E\alpha\}$. Moreover, we let $\wr\alpha = \{\beta | \alpha C\beta\}$.

Lemma 4. *Given* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})} = (\mathcal{O}^?, E, C)$ *for a revision state* $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ *such that* $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) = \text{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ *with* $\mathcal{O}^{\nvDash} \neq \emptyset$ *and* $\alpha \in \mathcal{O}^?$, *then*

1. $\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}) = (\mathcal{O}, \mathcal{O}^{\vDash} \cup \uparrow\alpha, \mathcal{O}^{\nvDash} \cup \wr\alpha)$,
2. $\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\}) = (\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \downarrow\alpha)$,
3. $impact^+(\alpha) = |\uparrow\alpha| + |\wr\alpha|$, *and*
4. $impact^-(\alpha) = |\downarrow\alpha|$.

Proof

1. By definition of closures, we have that $\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash})$ is $(\mathcal{O}, \mathcal{O}_c^{\vDash}, \mathcal{O}_c^{\nvDash})$ for $\mathcal{O}_c^{\vDash} = \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta\}$ and $\mathcal{O}_c^{\nvDash} = \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \vDash \gamma, \gamma \in \mathcal{O}^{\nvDash}\}$.
   By definition of the entails and conflicts relation we obtain $\mathcal{O}_c^{\vDash} = \mathcal{O}^{\vDash} \cup \{\beta \in \mathcal{O}^? | \alpha E\beta\}$ and $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^? | \alpha C\beta\}$.
   By definition of $\uparrow\alpha$ and $\wr\alpha$ follows $\mathcal{O}_c^{\vDash} = \mathcal{O}^{\vDash} \cup \uparrow\alpha$ and $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \wr\alpha$.
   Thus we obtain $\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}) = (\mathcal{O}, \mathcal{O}^{\vDash} \cup \uparrow\alpha, \mathcal{O}^{\nvDash} \cup \wr\alpha)$ as claimed.
2. Since $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash})$ is already closed, $\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\})$ is $(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}_c^{\nvDash})$ with $\mathcal{O}_c^{\nvDash} = \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\beta\} \vDash \gamma$ for some $\gamma \in (\mathcal{O}^{\nvDash} \cup \{\alpha\})\}$. Due to the prior closedness, α is the only possibly γ that will yield some β, hence $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^? | \mathcal{O}^{\vDash} \cup \{\beta\} \vDash \alpha\}$. By definition of the conflicts relation, this implies $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^? | \alpha C\beta\}$, whence by definition of $\downarrow\alpha$ follows $\mathcal{O}_c^{\nvDash} = \mathcal{O}^{\nvDash} \cup \downarrow\alpha$. Therefore $\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\}) = (\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \downarrow\alpha)$.
3. By Definition 3, $impact^+(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) - ?(\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}))$. By Definition 2, $?(\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash} \cup \{\alpha\}, \mathcal{O}^{\nvDash}))$ equals

$$?(\mathcal{O}, \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta\}, \qquad \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \vDash \gamma, \gamma \in \mathcal{O}^{\nvDash}\}).$$

By the definition of $?(\cdot)$ (Definition 3), $impact^+(\alpha) = |\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash})| - z|\mathcal{O} \setminus (\mathcal{O}_\alpha^{\vDash} \cup \mathcal{O}_\alpha^{\nvDash})|$ where $\mathcal{O}_\alpha^{\vDash} = \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha\} \vDash \beta\}$ and $\mathcal{O}_\alpha^{\nvDash} = \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\alpha, \beta\} \vDash \gamma, \gamma \in \mathcal{O}^{\nvDash}\}$.
   By definition of the entails and conflicts relations the line above equals

$$|\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \{\beta \in \mathcal{O}^? | \alpha E\beta\} \cup \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^? | \alpha C\beta\})|,$$

which, by definition of $\uparrow$ and $\wr$, is $|\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \uparrow\alpha \cup \mathcal{O}^{\nvDash} \cup \wr\alpha)|$. Overall we then have $impact^+(\alpha) = |\mathcal{O}| - (|\mathcal{O}^{\vDash}| + |\mathcal{O}^{\nvDash}|) - (|\mathcal{O}| - (|\mathcal{O}^{\vDash}| + |\uparrow\alpha| + |\mathcal{O}^{\nvDash}| + |\wr\alpha|))$, which is $|\uparrow\alpha| + |\wr\alpha|$.
4. By Definition 3, $impact^-(\alpha) = ?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash}) - ?(\text{clos}(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\alpha\}))$. By Definition 2, the latter is $?(\mathcal{O}, \mathcal{O}^{\vDash}, \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\beta\} \vDash \alpha\})$. Using Definition 3, $impact^-(\alpha)$ then is:

$$|\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash})| - |\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O} | \mathcal{O}^{\vDash} \cup \{\beta\} \vDash \alpha\})|$$

By definition of the entails relation the latter is $|\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash} \cup \{\beta \in \mathcal{O}^? | \beta E\alpha\})|$, which, by definition of $\downarrow$, is $|\mathcal{O} \setminus (\mathcal{O}^{\vDash} \cup \mathcal{O}^{\nvDash} \cup \downarrow\alpha)|$.
   Thus $impact^-(\alpha) = |\mathcal{O}| - (|\mathcal{O}^{\vDash}| + |\mathcal{O}^{\nvDash}|) - (|\mathcal{O}| - (|\mathcal{O}^{\vDash}| + |\mathcal{O}^{\nvDash}| + |\downarrow\alpha|)) = |\downarrow\alpha|$. $\square$

Hence, the computation of the revision closure (lines 5 and 7) and axiom impacts does not require any entailment checks if the according decision space is available. For the computation of deci-

sion spaces, we exploit the structural properties established in Lemmas 2 and 3 in order to reduce the number of required entailment checks in cases where the relations E and C are partially known. For this purpose, we define the rules R0–R9 displayed in Table 4, which describe the interplay between the relations E and C and their complements $\overline{E}$ and $\overline{C}$. The rules can serve as production rules to derive new instances of these relations thereby minimizing calls to costly reasoning procedures. By virtue of Lemma 3, we also have the guarantee that no further rules of this kind can be created, i.e., the rule set is complete for decision spaces.

An analysis of the dependencies between the rules R0–R9 reveals an acyclic structure (indicated by the order of the rules). Therefore E, C, $\overline{C}$, and $\overline{E}$ can be saturated one after another. Moreover, the exhaustive application of the rules R0–R9 can be condensed into the following operations:

$$E \leftarrow E^*,$$
$$C \leftarrow E \circ (C \cup C^-) \circ E^-,$$
$$\overline{C} \leftarrow E^- \circ (\overline{C} \cup Id \cup \overline{C}^-) \circ E,$$
$$\overline{E} \leftarrow E^- \circ (\overline{C} \circ C \cup \overline{E}) \circ E^-.$$

The correctness of the first operation (where $(\cdot)^*$ denotes the reflexive and transitive closure) is a direct consequence of R0 and R1. For the second operation, we exploit the relationships

$$E \circ C \circ E^- \overset{R2}{\subseteq} C \circ E^- \overset{R3}{\subseteq} C^- \circ E^- \overset{R2}{\subseteq} C^- \overset{R3}{\subseteq} C,$$
$$E \circ C^- \circ E^- \overset{R2}{\subseteq} E \circ C^- \overset{R3}{\subseteq} E \circ C \overset{R2}{\subseteq} C$$

that can be further composed into

$$E \circ C \circ E^- \cup E \circ C^- \circ E^- = E \circ (C \cup C^-) \circ E^- \subseteq C.$$

Conversely, iterated backward chaining for C w.r.t. R2 and R3 yields $E \circ (C \cup C^-) \circ E^-$ as a fixpoint, under the assumption $E = E^*$. The correctness of the last two operations can be shown accordingly.

Algorithm 2 realizes the cost-saving identification of the complete entailment and conflict relations of a decision space. Maintaining sets of known entailments (E), non-entailments ($\overline{E}$), conflicts (C) and non-conflicts ($\overline{C}$), the algorithm always closes these sets under the above operations before it cautiously executes expensive deduction checks to clarify missing cases. First, the initially known (non-)entailments and (non-)conflicts are closed in the aforementioned way (lines 1–7). There and in the subsequent lines, we split computations into several ones where appropriate in order to minimize the size of sets subject to the join operation ($\circ$). Lines 8–26 describe the successive clarification of the entailment relation (for cases where neither entailment nor non-entailment is known yet) via deduction checks. After each such clarification step, the sets E, $\overline{E}$, C, and $\overline{C}$ are closed. Thereby, we exploit known properties of intermediate results such as already being transitive or symmetric to avoid redoing the according closure operations unnecessarily (`transupdatediff` computes, for a relation R and a pair of elements (α, β), the difference between the reflexive transitive closure of R extended with (α, β) and R^*, i.e., $(R \cup \{(\alpha, \beta)\})^* \setminus R^*)$. Likewise, we also avoid redundant computa-

tions and reduce the size of the input sets for the join operations by explicitly bookkeeping sets E', C', $\overline{C}'$, and $\overline{E}'$ containing only the instances newly added in the current step. Lines 27–38 proceed in the analog way for stepwise clarification of the conflicts relation.

Algorithm 2. Decision Space Completion

Input: $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ a consistent revision state; E, $\overline{E}$, C, $\overline{C}$ subsets of the entailment and conflict relations and their complements

Output: $(\mathcal{O}^?, E, C)$ the corresponding decision space

1: $E \leftarrow E^*$
2: $C \leftarrow E \circ C \circ E^-$
3: $C \leftarrow C \cup C^-$
4: $\overline{C} \leftarrow E^- \circ \overline{C} \cup Id_{\mathcal{O}^?} \circ E$
5: $\overline{C} \leftarrow \overline{C} \cup \overline{C}^-$
6: $\overline{E} \leftarrow (\overline{C} \circ C) \cup \overline{E}$
7: $\overline{E} \leftarrow E^- \circ \overline{E} \circ E^-$
8: **while** $E \cup \overline{E} \neq \mathcal{O}^? \times \mathcal{O}^?$ **do**
9:     pick one $(\alpha, \beta) \in \mathcal{O}^? \times \mathcal{O}^? \setminus (E \cup \overline{E})$
10:    **if** $\mathcal{O}^{\models} \cup \{\alpha\} \models \beta$ **then**
11:        $E' \leftarrow \texttt{transupdatediff}(E, (\alpha, \beta))$
12:        $E \leftarrow E \cup E'$
13:        $C' \leftarrow (E' \circ C) \setminus C$
14:        $C' \leftarrow C' \cup (C' \circ E'^-) \setminus C$
15:        $C \leftarrow C \cup C'$
16:        $\overline{C}' \leftarrow (E'^- \circ \overline{C}) \setminus \overline{C}$
17:        $\overline{C}' \leftarrow \overline{C}' \cup (\overline{C}' \circ E') \setminus \overline{C}$
18:        $\overline{C} \leftarrow \overline{C} \cup \overline{C}'$
19:        $\overline{E}' \leftarrow ((\overline{C}' \circ C) \cup (\overline{C} \circ C')) \setminus \overline{E}$
20:        $\overline{E} \leftarrow \overline{E} \cup \overline{E}'$
21:        $\overline{E}' \leftarrow ((E'^- \circ \overline{E}) \cup (E^- \circ \overline{E}')) \setminus \overline{E}$
22:        $\overline{E} \leftarrow \overline{E} \cup \overline{E}' \cup (\overline{E}' \circ E^-) \cup (\overline{E} \circ E'^-)$
23:    **else**
24:        $\overline{E} \leftarrow \overline{E} \cup (E^- \circ \{(\alpha, \beta)\} \circ E^-)$
25:    **end if**
26: **end while**
27: **while** $C \cup \overline{C} \neq \mathcal{O}^? \times \mathcal{O}^?$ **do**
28:    pick one $(\alpha, \beta) \in \mathcal{O}^? \times \mathcal{O}^? \setminus (C \cup \overline{C})$
29:    **if** $\mathcal{O}^{\models} \cup \{\alpha, \beta\} \models \gamma$ for some $\gamma \in \mathcal{O}^{\not\models}$ **then**
30:        $C' \leftarrow E \circ \{(\alpha, \beta), (\beta, \alpha)\} \circ E^-$
31:        $C \leftarrow C \cup C'$
32:        $\overline{E} \leftarrow \overline{E} \cup (E^- \circ \overline{C} \circ C' \circ E^-)$
33:    **else**
34:        $\overline{C}' \leftarrow (E^- \circ \{(\alpha, \beta), (\beta, \alpha)\} \circ E) \setminus \overline{C}$
35:        $\overline{C} \leftarrow \overline{C} \cup \overline{C}'$
36:        $\overline{E} \leftarrow \overline{E} \cup (E^- \circ \overline{C}' \circ C \circ E^-)$
37:    **end if**
38: **end while**

Since the complexity of entailment checking will almost always outweigh the complexity of the other operations in Algorithm 2, we first analyze the complexity of the algorithm under the assumption that entailment checking is done by a constant time oracle. We then show how entailment checking can be factored in.

Lemma 5. *Let $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ be a revision state with $\mathcal{O}^{\not\models} \neq \emptyset$ and E, $\overline{E}$, C, $\overline{C}$ (possibly empty) subsets of the entailment and conflicts relations. We denote the size $|\mathcal{O}|$ of $\mathcal{O}$ with n. Given $(\mathcal{O}, \mathcal{O}^{\models}, \mathcal{O}^{\not\models})$ and E, $\overline{E}$, C, $\overline{C}$ as input, Algorithm 2 runs in time bounded by $O(n^5)$ and space bounded by $O(n^2)$ if we assume that entailment checking is a constant time operation.*

Table 4
Completion rules for partially known decision spaces.

R0		$\rightarrow$	$E(x,x)$	Reflexivity of E
R1	$E(x,y) \wedge E(y,z)$	$\rightarrow$	$E(x,z)$	Transitivity of E
R2	$E(x,y) \wedge C(y,z)$	$\rightarrow$	$C(x,z)$	(P3)
R3	$C(x,y)$	$\rightarrow$	$C(y,x)$	Symmetry of C
R4	$E(x,y)$	$\rightarrow$	$\overline{C}(x,y)$	Disjointness of E and C
R5	$\overline{C}(x,y)$	$\rightarrow$	$\overline{C}(y,x)$	Symmetry of C
R6	$E(x,y) \wedge \overline{C}(x,z)$	$\rightarrow$	$\overline{C}(y,z)$	(P3)
R7	$C(x,y)$	$\rightarrow$	$\overline{E}(x,y)$	Disjointness of E and C
R8	$\overline{C}(x,y) \wedge C(y,z)$	$\rightarrow$	$\overline{E}(x,z)$	(P3)
R9	$E(x,y) \wedge \overline{E}(x,z)$	$\rightarrow$	$\overline{E}(y,z)$	Transitivity of E

Proof. We first note that $\mathcal{O}^?$ is bounded by n since $|\mathcal{O}^?| = |\mathcal{O}| - (|\mathcal{O}^\vDash| + |\mathcal{O}^{\nvDash}|)$. Similarly, the size of each relation E, $\overline{E}$, C, and $\overline{C}$ is bounded by n^2 since the relations are binary relations over axioms in $\mathcal{O}$. We first analyze the individual operations. Computing the transitive reflexive closure of a relation can be done in cubic time, i.e., for E^* with E a relation over at most n axioms, we get a bound of n^3. The computation of `transupdatediff` is in the worst case the same as computing the reflexive transitive closure. For a binary join operation ($\circ$), the output is again a binary relation over $\mathcal{O}$ of size bounded by n^2. Each binary join can be computed in at most n^3 steps. Note that multiple joins can be seen as several binary joins since each intermediate for formalisms where entailment checking is harder than PTime relation is again over axioms from $\mathcal{O}$ and is of size at most n^2. The union operation ($\cup$) corresponds to the addition of axioms. Each of the while loops is executed at most n^2 times and requires a fixed number of join operations and possibly in one case the computation of `transupdatediff`, which gives an upper bound of $O(n^2 \cdot n^3) = O(n^5)$ for both while loops. Together with the reflexive transitive closure and the fixed number of join operations before the while loops, we have that the time complexity of Algorithm 2 is $O(n^5)$ and its space complexity is $O(n^2)$ assuming that entailment checking is a constant time operation. $\square$

Note that the approach is targeted towards logics where the entailment checking problem is not tractable, i.e., harder than PTime. Otherwise, the performance benefit is only due to avoiding applying the same entailment checks multiple times.

Lemma 6. *Let $(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})$ be a revision state with $\mathcal{O}^{\nvDash} \neq \emptyset$, $|\mathcal{O}| := n$ and the axioms in $\mathcal{O}$ expressed in a logic $\mathcal{L}$ in which taking all consequences is a closure operation and for which there is a decision procedure for logical entailment of complexity $c(n)$ where n is the size of the input to the procedure. Let E, $\overline{E}$, C, $\overline{C}$ be (possibly empty) subsets of the according entailment and conflicts relations. Then there is a polynomial p such that the runtime of Algorithm 2, given $(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})$ and E, $\overline{E}$, C, $\overline{C}$ as input, is bounded by $p(n) \cdot c(n)$.*

Proof. The input to the entailment checking algorithm is in all cases of size n. Both while loops perform at most n^2 entailment checks, which together with the analysis from Lemma 5 give the desired result. $\square$

3.1. Updating decision spaces

We proceed by formally describing the change of the decision space as a consequence of approving or declining one axiom with the objective of again minimizing the required number of entailment checks. We first consider the case that an expert approves an axiom $\alpha \in \mathcal{O}^?$, and hence α is added to the set $\mathcal{O}^\vDash$ of wanted consequences.

Lemma 7. *Let $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})} = (\mathcal{O}^?, E, C)$, $\alpha \in \mathcal{O}^?$, and $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash \cup \{\alpha\}, \mathcal{O}^{\nvDash})} = (\mathcal{O}^?_{\text{new}}, E', C')$. Then*

- *$\mathcal{O}^?_{\text{new}} = \mathcal{O}^? \setminus (\uparrow \alpha \cup \wr \alpha)$,*
- *$\beta E \gamma$ implies $\beta E' \gamma$ for β, $\gamma \in \mathcal{O}^?_{\text{new}}$, and*
- *$\beta C \gamma$ implies $\beta C' \gamma$ for β, $\gamma \in \mathcal{O}^?_{\text{new}}$.*

Essentially, the lemma states that all axioms entailed by α (as witnessed by E) as well as all axioms conflicting with α (indicated by C) will be removed from the decision space if α is approved. Moreover due to monotonicity, all positive information about entailments and conflicts remains valid. Algorithm 3 runs in time bounded by

$O(n^5)$ and space bounded by $O(n^2)$ and takes advantage of these correspondences when fully determining the updated decision space.

Lemma 8. *Let $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})}$ be a decision space, $\alpha \in \mathcal{O}^?$ an axiom. We denote the size $|\mathcal{O}|$ of $\mathcal{O}$ with n. Given $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})}$ and α as input, Algorithm 3 runs in time bounded by $O(n^5)$ and space bounded by $O(n^2)$ if we assume that entailment checking is a constant time operation.*

Proof. Lines 1–5 of Algorithm 3 can be executed in cubic time and quadratic space using the same arguments as in Lemma 5. By Lemma 5, executing lines 8–38 from Algorithm 2 under the assumption that entailment checking is a constant time operation can be done in time $O(n^5)$, which proves the claim.

Algorithm 3. Update of Decision Space $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})}$ on Approving α

Input: $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})}, \alpha \in \mathcal{O}^?$
Output: $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash \cup \{\alpha\}, \mathcal{O}^{\nvDash})}$ updated decision space
1: $\mathcal{O}^? \leftarrow \mathcal{O}^? \setminus (\uparrow \alpha \cup \wr \alpha)$
2: $E \leftarrow E \cap (\mathcal{O}^? \times \mathcal{O}^?)$
3: $C \leftarrow C \cap (\mathcal{O}^? \times \mathcal{O}^?)$
4: $\overline{C} \leftarrow E^- \circ E$
5: $\overline{E} \leftarrow E^- \circ \overline{C} \circ C \circ E^-$
6: execute lines 8–38 from Algorithm 2

Algorithm 4. Update of Decision Space $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})}$ on Declining α

Input: $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})}, \alpha \in \mathcal{O}^?$
Output: $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash} \cup \{\alpha\})}$ updated decision space
 1: $\mathcal{O}^? \leftarrow \mathcal{O}^? \setminus \downarrow \alpha,$
 2: $E \leftarrow E \cap (\mathcal{O}^? \times \mathcal{O}^?)$
 3: $\overline{E} \leftarrow \overline{E} \cap (\mathcal{O}^? \times \mathcal{O}^?)$
 4: $C \leftarrow C \cap (\mathcal{O}^? \times \mathcal{O}^?)$
 5: $\overline{C} \leftarrow E^- \circ E$
 6: **while** $C \cup \overline{C} \neq \mathcal{O}^? \times \mathcal{O}^?$ **do**
 7:     pick one $(\beta, \gamma) \in \mathcal{O}^? \times \mathcal{O}^? \setminus (C \cup \overline{C})$
 8:     **if** $\mathcal{O}^\vDash \cup \{\beta, \gamma\} \vDash \alpha$ **then**
 9:        $C \leftarrow C \cup (E \circ \{(\beta, \gamma), (\gamma, \beta)\} \circ E^-)$
10:     **else**
11:        $\overline{C} \leftarrow \overline{C} \cup (E^- \circ \{(\beta, \gamma), (\gamma, \beta)\} \circ E)$
12:     **end if**
13: **end while**

The next lemma considers changes to be made to the decision space on the denial of an axiom α by characterizing it as unwanted consequence.

Lemma 9. *Let $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash})} = (\mathcal{O}^?, E, C)$, $\alpha \in \mathcal{O}^?$, and $\mathbb{D}_{(\mathcal{O}, \mathcal{O}^\vDash, \mathcal{O}^{\nvDash} \cup \{\alpha\})} = (\mathcal{O}^?_{\text{new}}, E', C')$. Then*

- *$\mathcal{O}^?_{\text{new}} = \mathcal{O}^? \setminus \downarrow \alpha,$*
- *$\beta E \gamma$ exactly if $\beta E' \gamma$ for β, $\gamma \in \mathcal{O}^?_{\text{new}}$, and*
- *$\beta C \gamma$ implies $\beta C' \gamma$ for β, $\gamma \in \mathcal{O}^?_{\text{new}}$.*

The lemma shows that the updated decision space can be obtained by removing all axioms that entail α. Furthermore entailments between remaining axioms remain unaltered whereas the set of conflicts may increase. Algorithm 4 implements the respective decision space update, additionally exploiting that new conflicts can only arise from derivability of the newly declined axiom α. Algorithms 3 and 4 have to be called in Algorithm 1 after the accept (line 5) or decline revision step (line 7), respectively.

For n the number of involved axioms, Algorithms 2–4 run in time bounded by $O(n^5)$ and space bounded by $O(n^2)$ if we treat entailment checking as a constant time operation. Without the latter assumption, the complexity of reasoning usually dominates. For example, if the axioms use all features of OWL 2 DL, entailment checking is *N2ExpTime*-complete [12], which then also applies to our algorithm.

Lemma 10. *Let* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})}$ *be a decision space,* $\alpha \in \mathcal{O}^?$ *an axiom. We denote the size* $|\mathcal{O}|$ *of* $\mathcal{O}$ *with* n. *Given* $\mathbb{D}_{(\mathcal{O},\mathcal{O}^{\vDash},\mathcal{O}^{\nvDash})}$ *and* α *as input, Algorithm 4 runs in time bounded by* $O(n^5)$ *and space bounded by* $O(n^2)$ *if we assume that entailment checking is a constant time operation.*

Proof. The execution lines 1–5 of Algorithm 4 can be performed in quadratic space and cubic time using the same arguments as in Lemma 5. We execute the operations within the while loop at most n^2 times, and under the assumption that entailment checking is a constant time operation, we find that the operations can again be performed in cubic time and quadratic space resulting in an overall bound for the time complexity of $O(n^5)$ and $O(n^2)$ space complexity. □

4. Parametrized ranking

Let us look again at our running example, but assume that Axioms (1) and (2) are incorrect, i.e., the validity ratio is 75%. The so far introduced ranking functions do not take this into account: if we use $impact^+$, which shows the highest value of 7 for Axiom (1), then the user would decline the axiom and no automatic decisions are possible. Next, Axiom (2) is highest ranked, but again declined. Finally, after presenting Axiom (3) to the user, the axiom is approved and all remaining axioms can be approved automatically. The ranking function $impact^-$ even takes 7 steps, whereas *guaranteed* performs slightly better with (theoretically) 2.8 expert decisions. This is an average for the different possible choices among the highest ranked axioms assuming that these have the same probability of being chosen. If we look at $impact^+$ again, the problem is that the function ignores the validity ratio, i.e., Axiom (1) is presented to the user in the hope of an approval knowing that this could only happen if the validity ratio was 100% (due to the automatic approval of all remaining axioms). To address this issue, we now present a ranking function that takes the validity ratio into account.

4.1. The ranking function norm

We now define the ranking function $norm_R$, which minimizes the deviation of the fraction of accepted and declined axioms from the expected overall ratios of desired and undesired consequences. To determine this deviation for each axiom α, we first have to compute the fraction of accepted and declined axioms by normalizing impacts of α to values between 0 and 1. For this purpose, we define functions $impact_N^+$ and $impact_N^-$. Since in the case of an approval, we can possibly both accept and decline axioms automatically, an approval influences both, the ratio of accepted and declined axioms. To take both influences into account, in Definition 3 we split the approval impact accordingly into $impact^{+a}$ and $impact^{+d}$. Along the same lines, we obtain $impact_N^{+a}$ and $impact_N^{+a}$ by normalizing these two components with respect to the expected validity ratio. On the contrary, in the case of a decline, we can only decline axioms automatically. Therefore, we do not split $impact^-$.

Definition 5. Let $\mathcal{O}^?$ be a connected component of the decision space and R the expected validity ratio. The *normalized impact functions* are:

$$impact_N^{+a}(\alpha) = \frac{1 + impact^{+a}(\alpha)}{|\mathcal{O}^?|},$$

$$impact_N^{+d}(\alpha) = \frac{impact^{+d}(\alpha)}{|\mathcal{O}^?|},$$

$$impact_N^-(\alpha) = \frac{1 + impact^-(\alpha)}{|\mathcal{O}^?|}.$$

The ranking functions $norm_R^{+a}$, $norm_R^{+d}$ and $norm_R^-$ are then defined by

$$norm_R^{+a}(\alpha) = -|R - impact_N^{+a}(\alpha)|,$$

$$norm_R^{+d}(\alpha) = -|1 - R - impact_N^{+d}(\alpha)|,$$

$$norm_R^-(\alpha) = -|1 - R - impact_N^-(\alpha)|.$$

Finally, the ranking function $norm_R$ for an axiom α is defined as

$$\max\left(norm_R^{+a}(\alpha), norm_R^{+d}(\alpha), norm_R^-(\alpha)\right).$$

When computing $impact_N^{+a}$, we increment it by 1, since we are interested in the overall fraction of accepted axioms, and, α itself is one of the accepted axioms. For the same reason, we also increment $impact_N^-$ by 1, but not $impact_N^{+d}$, where α itself is accepted and does not increment the number of declined axioms.

Table 5 shows the computation of $norm_{0.75}$ for Example 1. The function $norm_R^{+a}$ captures how the fraction of automatically accepted axioms deviates from the expected overall ratio of wanted consequences, e.g., accepting Axiom (2) or (4) deviates by 12.5%: for the former axiom we have automatically accepted too many axioms, while for the latter we do not yet have accepted enough under the premise that the validity ratio is indeed 75%. Since Example 1 does not allow for automatic declines after an approval, the function $norm_R^{+d}$ shows that for each accept, we still deviate 25% from the expected ratio of *invalid* axioms, which is $1 - R$. The function $norm_R^-$ works analogously for declines. Hence, $norm_R$ is defined in a way that it takes the greatest value if the chance that all wanted (unwanted) axioms are accepted (declined) at once becomes maximal.

Note that the expected validity ratio within the corresponding connected decision space component needs to be adjusted after each expert decision, to reflect the expected validity ratio of the remaining unevaluated axioms. For instance, after Axiom (2) has been declined, $norm_{1.00}$ needs to be applied to rank the remaining axioms. If, however, Axiom (3) has been accepted, $norm_{0.00}$ is required.

Further, it is interesting to observe that employing $norm_{0.00}$ for ranking yields the same behavior as $impact^-$. On the other hand, $norm_{1.00}$ corresponds to $impact^+$ in case no conflicting axioms are involved, which is in fact very probable if R is close to 100%. Therefore, $norm$ represents a generalization of the earlier introduced impact functions $impact^+$ and $impact^-$.

4.2. Learning the validity ratio

Users might only have a rough idea or even no idea at all of the validity ratio of a dataset in advance of the revision. Hence, it might be difficult or impossible to decide upfront which R to use for $norm_R$. To address this problem, we investigate how efficiently we can "learn" the validity ratio on the fly. In this setting, the user gives a prior estimate for R (or we use 50% as default) and with each revision of another connected decision space component, R is adjusted to reflect exactly the actual validity ratio at the current stage– the proportion of (manually and automatically) approved axioms within the total set of the evaluated axioms so far. Thus, the algorithm tunes itself towards an optimal ranking function, which relieves the user from choosing a validity ratio. We call

Table 5
The values for $norm_{0.75}$ and the intermediate functions (shown in percentage).

Axiom	$impact_N^{+a}$ (%)	$impact_N^{+d}$ (%)	$impact_N^-$ (%)	$norm_{0.75}^{+a}$ (%)	$norm_{0.75}^{+d}$ (%)	$norm_{0.75}^-$ (%)	$norm_{0.75}$ (%)
(1)	100.0	0.0	12.5	−25.0	−25.0	−12.5	−12.5
(2)	87.5	0.0	25.0	−12.5	−25.0	0.0	0.0
(3)	75.0	0.0	37.5	0.0	−25.0	−12.5	0.0
(4)	62.5	0.0	50.0	−12.5	−25.0	−25.0	−12.5
(5)	50.0	0.0	62.5	−25.0	−25.0	−37.5	−25.0
(6)	37.5	0.0	75.0	−37.5	−25.0	−50.0	−25.0
(7)	12.5	0.0	87.5	−62.5	−25.0	−62.5	−25.0
(8)	12.5	0.0	87.5	−62.5	−25.0	−62.5	−25.0

the according ranking function *dynnorm* as it dynamically adapts the estimated validity ratio over the course of the revision.

In our experiments, we show that, already for small datasets, *dynnorm* outperforms random ordering and, in case of sufficiently large datasets, the estimate converges towards the actual validity ratio, thereby making the assumption of a known validity ratio obsolete.

5. Partitioning

Since reasoning operations, in particular for expressive formalisms such as OWL, are very expensive (the reasoner methods take 99.2% of the computation time in our experiments according to our profiling measurements), we combine the optimization using decision spaces with a straight-forward partitioning approach that is applicable for OWL ontologies and splits ABox axioms (i.e., class and property assertions) into disjoint subsets. Thus, the subsequent discussion is specific to OWL reasoning.

Definition 6. Let $\mathcal{A}$ be a set of ABox axioms, $ind(\mathcal{A})$ the set of individual names used in $\mathcal{A}$, then $\mathcal{A}$ is *connected* if, for all pairs of individuals $a, a' \in ind(\mathcal{A})$, there exists a sequence $a_1, \ldots, a_n$ such that $a = a_1$, $a' = a_n$, and, for all $1 \leqslant i < n$, there exists a property assertion in A containing a_i and a_{i+1}. A collection of ABoxes $\mathcal{A}_1, \ldots, \mathcal{A}_k$ is a *partitioning* of $\mathcal{A}$ if $\mathcal{A} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$,

$ind(\mathcal{A}_i) \cap ind(\mathcal{A}_j) = \emptyset$ for $1 \leqslant i < j \leqslant k$, and each $\mathcal{A}_i$ is connected.
The proposed partitioning process can be done in linear time, since it is more or less straight-forward product of the computation of the connected components of the ABox graph. In the absence of nominals (OWL's oneOf constructor), the above described partitions or clusters of an ABox are indeed independent. Thus, the above partitioning does not split any connected decision space components, and, therefore, we obtain an equivalent complete revision state and an equivalent evaluation order within each connected decision space component. We apply partitioning once at the beginning of the revision to the whole set of unevaluated axioms and then perform the revision for each partition separately by joining the partition with the remaining terminological axioms. So far, we abstract from the possibility to update the partitioning to a more fine-grained one over the course of revision.

In order to also partition non-Abox axioms or to properly take axioms with nominals into account, other partitioning techniques can be applied, e.g., the signature decomposition approach by Konev et al. [13] that partitions the vocabulary of an ontology into subsets that are independent regarding their meaning. The resulting independent subsets of the ontology can then be reviewed independently from each other analogously to the clusters of ABox axioms used in our evaluation. We made the following general observations:

- In particular in case of large datasets containing several partitions, the additional partitioning-based optimization significantly reduces the computational effort.

- Partitioning intensifies the effectiveness of decision spaces, since the "density" of entailment and contradiction relations are significantly higher within each partition than the density within a set of independent partitions.

6. Experimental results

We evaluated our revision support methodology within the project *NanOn* aiming at ontology-supported literature search. During this project, a hand-crafted ontology modeling the scientific domain of nano technology has been developed, capturing substances, structures, and procedures used in that domain. The ontology, denoted here with $\mathcal{O}$, is specified in the Web Ontology Language OWL 2 DL [14] and comprises 2289 logical axioms. This ontology is used as the core resource to automatically analyze scientific documents for the occurrence of NanOn classes and properties by the means of lexical patterns. When such classes and properties are found, the document is automatically annotated with them to facilitate topic-specific information retrieval on a fine-grained level. Note that these annotations are modeled as ABox facts and must not be confused with OWL annotations, which do not carry any semantics in OWL 2 DL. In this way, one of the project outputs is a large amount of class and property assertions associated with the *NanOn* ontology. In order to estimate the accuracy of such automatically added annotations, they need to be inspected by human experts, which provides a natural application scenario for our approach. The manual inspection of annotations provided us with sets of valid and invalid annotation assertions (denoted by $\mathcal{A}^+$ and $\mathcal{A}^-$, respectively). To investigate how the quality and the size of each axiom set influences the results, we created several distinct annotation sets with different *validity ratios* $|\mathcal{A}^+|/(|\mathcal{A}^+| + |\mathcal{A}^-|)$. As the annotation tools provided rather reliable data, we had to manually create additional wrong frequently occurring patterns and apply them for annotating texts to obtain datasets with a lower validity ratio.

For each set, we applied our methodology starting from the revision state $(\mathcal{O} \cup \mathcal{O}^- \cup \mathcal{A}^+ \cup \mathcal{A}^-, \mathcal{O}, \mathcal{O}^-)$ with $\mathcal{O}$ containing the axioms of the NanOn ontology and with $\mathcal{O}^-$ containing axioms expressing inconsistency and class unsatisfiability. We obtained a complete revision state $(\mathcal{O} \cup \mathcal{O}^- \cup \mathcal{A}^+ \cup \mathcal{A}^-, \mathcal{O} \cup \mathcal{A}^+, \mathcal{O}^- \cup \mathcal{A}^-)$ where on-the-fly expert decisions about approval or decline were simulated according to the membership in $\mathcal{A}^+$ or $\mathcal{A}^-$. For computing the entailments, we used the OWL reasoner HermiT.[4]

For each set, our baseline is the reduction of expert decisions when axioms are evaluated in random order, i.e., no ranking is applied and only the revision closure is used to automatically evaluate axioms. The upper bound for the in principle possible reduction of expert decisions is called the *optimal* ranking, obtained by applying the "impact oracle" for each axiom α that is to be evaluated:

---

	validity ratio	*optimal*	*norm*	*best unparameterized*	*random*
L_1	90%	65.6%	65.4%	(*impact$^+$*) 65.4%	41.7%
L_2	76%	59.8%	59.8%	(*impact$^+$*) 55.8%	35.8%
L_3	50%	47.8%	47.6%	(*guaranteed*) 36.5%	24.4%
L_4	25%	59.9%	59.8%	(*impact$^-$*) 54.9%	37.6%
L_5	10%	63.9%	63.9%	(*impact$^-$*) 63.9%	40.3%
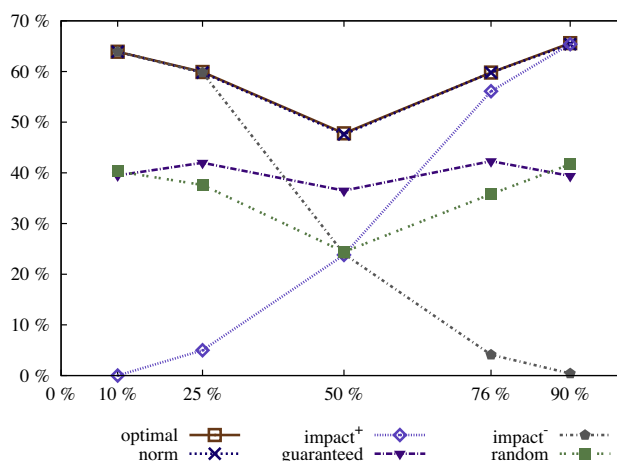
Fig. 1. Revision results of *norm* in comparison with other ranking functions for the sets L_1–L_5.

$$KnownImpact(\alpha) = \begin{cases} impact^+(\alpha) & \text{if } \alpha \in \mathcal{A}^+, \\ impact^-(\alpha) & \text{if } \alpha \in \mathcal{A}^-. \end{cases}$$

6.1. Evaluation of norm

To compare the effectiveness of *impact$^+$*, *impact$^-$*, and *guaranteed* with the parametrized ranking *norm*, we created five sets of annotations L_1 to L_5, each comprising 5000 axioms with validity ratios varying from 10% to 90%.

The table in Fig. 1 shows the results for the different ranking techniques: the column *optimal* shows the upper bound achieved by using the impact oracle, *norm* shows the results for *norm* parameterized with the actual validity ratio, *best unparameterized* shows the best possible value achievable with the unparameterized functions, and, finally, the column *random* states the effort reduction already achieved by presenting the axioms in random order. The results show that *norm* consistently achieves almost the maximum effort reduction with an average difference of 0.1%. The unparameterized functions only work well for the high and low quality datasets, as expected, where *impact$^+$* works well for the former case, while *impact$^-$* works well for the latter. For the dataset with validity ratio 50%, *norm* achieves an additional 11.1% of automatization by using the parameterized ranking.

In general, the actual difference in performance achieved by the more precise parameterized ranking increases for a higher average maximum path length within connected decision space graphs. To see this, consider again the decision space shown in Table 3. It is clear that the distance between the highest ranked axioms for different ranking functions increases as the height of the presented tree becomes greater.

6.2. Evaluation of dynnorm

In order to evaluate our solution for situations where the validity ratio is unknown or only very rough estimates can be given

upfront, we analyzed the effectiveness of the dynamically learning ranking function *dynnorm*. To this end, we created the following annotation sets in addition to the datasets $L_1 - L_5$:

- small datasets S_1–S_5 with the size constantly growing from 29 to 102 axioms and validity ratios varying from 10% to 90%,
- medium-sized datasets M_1–M_5 with 500 axioms each and validity ratios varying from 10% to 91%.

Table 6 shows the results of the revision: the columns *optimal* and *random* are as described above, the column *norm* shows the results that we would obtain if we were to assume that the validity ratio is known and given as parameter to the *norm* ranking function, the columns *dynnorm$_{0.50}$*, *dynnorm$_{1.00}$* and *dynnorm$_{0.00}$* show the results for starting the revision with a validity ratio of 50%, 100%, and 0%, respectively, where over the course of the revision, we update the validity ratio estimate.

We observe, that, in case of small datasets (S_i), the deviation from *norm* (on average 5%) as well as the dependency of the results on the initial value of the validity ratio are clearly visible. However, the results of *dynnorm* are significantly better (45.0%) than those of a revision in random order. It is also interesting to observe that the average deviation from *norm* decreases with the size of a dataset (6.9%, 10.5%, 2.7%, 3.3%, 1.9% for S_1 to S_5, respectively) and that the probability of a strong deviation is lower for datasets with an extreme validity ratio (close to 100% or 0%).

For medium-sized and large datasets (M_i and L_i), the deviation from *norm* (on average 0.3% for both) as well as the dependency on the initial value of the validity ratio are significantly lower, as displayed in Fig. 2.

We conclude that

- ranking based on learning validity ratio is already useful for small datasets (30-100 axioms), and improves significantly with the growing size of the dataset under revision;
- in case of large datasets, the performance difference between the the results with a validity ratio known in advance and a

Table 6
Revision results for datasets S_1–S_5, M_1–M_5, and L_1–L_5.

	Validity ratio (%)	Optimal (%)	Norm (%)	$dynnorm_{0.50}$ (%)	$dynnorm_{1.00}$ (%)	$dynnorm_{0.00}$ (%)	Random (%)
S_1	90	72.4	72.4	58.6	72.4	65.5	40.8
S_2	77	68.6	65.7	57.1	62.9	48.6	38.2
S_3	48	65.1	65.1	65.1	60.3	61.9	22.0
S_4	25	68.3	68.3	64.6	63.4	67.1	37.6
S_5	10	72.5	72.5	71.6	67.6	72.5	29.2
M_1	91	66.4	66.0	66.2	66.4	65.6	40.8
M_2	77	60.0	60.0	59.6	59.8	59.2	38.2
M_3	44	40.8	40.6	40.4	40.6	40.4	22.0
M_4	25	60.0	60.0	59.6	59.2	59.8	37.6
M_5	10	53.2	53.0	52.8	52.8	53.2	29.2
L_1	90	65.6	65.4	65.4	65.4	65.3	41.7
L_2	76	59.8	59.8	59.8	59.8	59.9	35.8
L_3	50	47.8	47.6	47.4	47.2	47.3	24.4
L_4	25	59.9	59.8	59.8	59.8	59.8	37.6
L_5	10	63.9	63.9	63.9	63.8	63.9	40.3

learned validity ratio almost disappears, thereby making the assumption of known average validity ratio obsolete for axiom ranking.

6.3. Computational effort

During our experiments, we measured the average number of seconds after each expert decision required for the automatic evaluation and ranking as well as the average number of reasoning calls. If we compute the average values for the revision based on *dynnorm* ranking for all 15 datasets, the revision took 0.84 s (7.4 reasoning calls) after each expert decision. In the case of small datasets, partitioning yields an additional improvement by an order of magnitude in terms of reasoning calls. For medium-sized datasets without partitioning, the first step out of on average 153 evaluation steps took already 101101 reasoning calls (ca. 3 h) even when using decision spaces. Without decision spaces and partitioning, the required number of reasoning calls for the revision of the sets M_1–M_5 would be more than 500000, judging by the required reasoning calls to build the corresponding decision space in the worst case. For this reason, we did not try to run the experiment without partitioning for large datasets, which would require more than 50 million reasoning calls without decision in the worst case. In contrast to that, the average number of required reasoning calls for a complete revision of the sets M_1–M_5 with partitioning amounts to 3380. The revision of datasets L_1–L_5 with partitioning required overall on average 16175 reasoning calls, which corresponds to between 6 and 7 reasoning calls per evaluation decision. We can summarize the evaluation results as follows:

- The proposed reasoning-based support performs well in an interactive revision process with on average 0.84 seconds per expert decision.
- In particular in case of large datasets containing several partitions, partitioning significantly reduces the computational effort.
- The employment of decision spaces saves in our experiments on average 75% of reasoner calls. As measured in the case of small datasets, partitioning further intensifies the effect of decision spaces and we save even 80% of reasoner calls.

7. User front-end

Fig. 3 shows the user front-end of the *revision helper* tool. It allows the user to load the set $\mathcal{O}$ of axioms under revision and save or load an evaluation state for the currently loaded set $\mathcal{O}$. Thereby, the user can interrupt the revision at any time and proceed later on. If partitioning is activated, revision helper shows the partitions one after another and the revision of each partition is independent from the revision of all other partitions.

By default, revision helper initializes the set $\mathcal{O}^{\not\models}$ of undesired statements with the minimal set of statements expressing the inconsistency of the ontology or unsatisfiability of its classes. The set of desired statements $\mathcal{O}^{\models}$ can be initialized by loading an arbitrary ontology. A statement can be evaluated by choosing one of the values *Accept* and *Decline*, and it can be excluded from the revision process by choosing *Exclude*. The latter option should be used, if the meaning of a statement is not clear and the user cannot decide whether to accept or to decline it. After the statement has
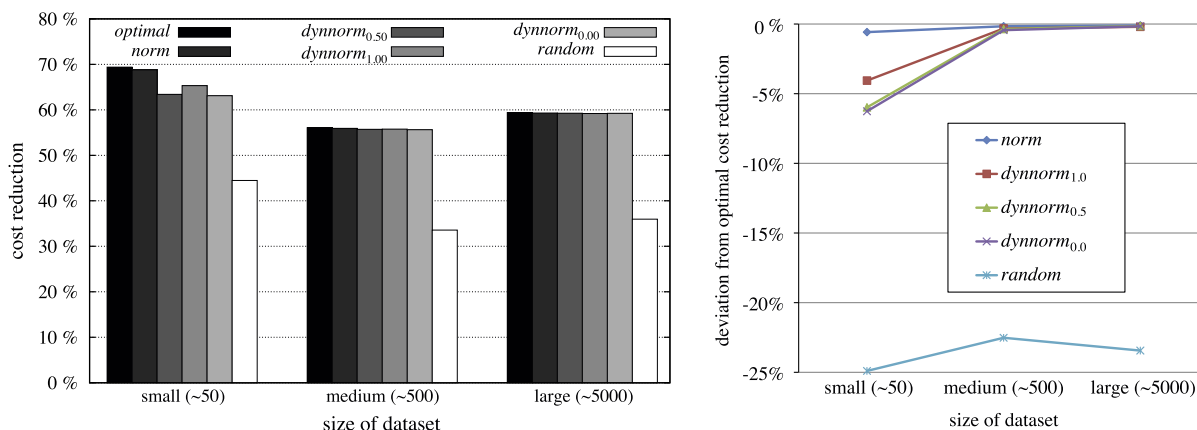

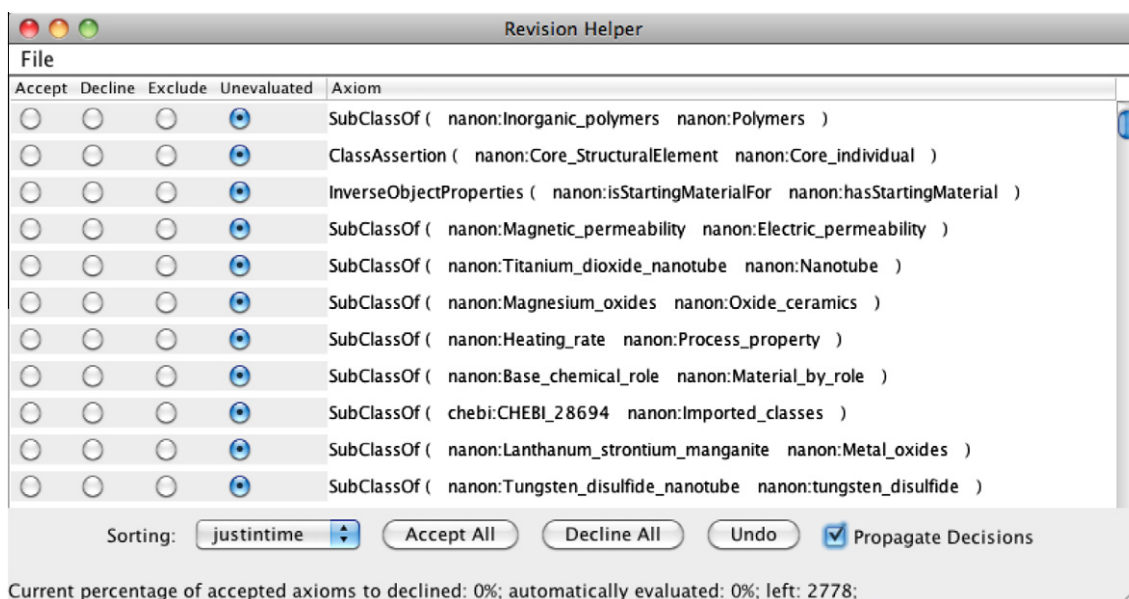**Fig. 2.** Effect of learning validity ratio for different data set sizes.

Fig. 3. Revision helper GUI.

been evaluated, it disappears from the revision list as well as all statements that could be evaluated automatically, unless the checkbox *Propagate Decisions* is deactivated. The ranking strategy used for sorting the statements can be selected or deactivated at any time and is taken into account after the next evaluation decision. At any stage of the revision, it is possible to export the current set $\mathcal{O}^{\vDash}$ of accepted statements as an ontology. For the export, we exclude, however, axioms with which $\mathcal{O}^{\vDash}$ has been initialized at the beginning of the revision.

8. Related work

In our previous work [15], we proposed an approach for determining a beneficial order of axiom evaluation under the assumption of a high validity ratio within the axiom set under investigation. The latter approach aims at reducing the manual effort of revision by eliminating the redundancy within the corresponding axiom set, which is the major factor leading to automatic axiom evaluation under the assumption of a high validity ratio.

In addition to our own work, we are aware of two approaches for supporting the revision of ontological data based on logical appropriateness: an approach by Meilicke et al. [16] and another one called *ContentMap* by Jiménez-Ruiz et al. [17]. Both approaches are applied in the context of mapping revision. An extension of *ContentMap* called *ContentCVS* [18] supports an integration of changes into an evolving ontology. In all of these approaches, dependencies between evaluation decisions are determined based on a set of logical criteria each of which is a subset of the criteria that can be derived from the notion of revision state consistency introduced in Def. 1.

In contrast to our approach, ContentMap and ContentCVS focus on the visualization of consequences and user guidance in case of difficult evaluation decisions. These approaches selectively materialize and visualize the logical consequences caused by the axioms under investigation and support the revision of those consequences. Subsequently, the approved and declined axioms are determined in correspondence with the revision of the consequences. The minimization of the manual and computational effort required for the revision is not considered. In contrast to our approach, which requires at most a polynomial number of entailment checks, ContentMap and ContentCVS require an exponential num-

ber of reasoning operations compared to the size of the ontology under revision. The reason for this is that ContentMap and Content-CVS determine the dependencies between the consequences by comparing their *justifications*, i.e., sets of axioms causing the entailment of the consequence.

Similarly to our approach, Meilicke et al. aim at reducing the manual effort of mapping revision. However, their results are difficult to generalize to the revision of ontologies, since the notion of impact is defined based on specific properties of mapping axioms. For every mapping axiom possible between the entities of the two mapped ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, they define the impact as the corresponding number of possible entailed and contradicting mapping axioms. The assumption is that the set of possible mapping axioms and the set of possible axioms in $\mathcal{O}_1$ and $\mathcal{O}_2$ are mostly disjoint, since axioms in $\mathcal{O}_1$ and $\mathcal{O}_2$ usually refer only to entities from the same ontology, while mapping axioms are assumed to map only entities from different ontologies. In case of ontology revision in general, no such natural distinction criteria for axioms under revision can be defined. Moreover, in contrast to our approach, Meilicke et al. abstract from the interactions between more than one mapping axiom.

Another strand of work is related to the overall motivation of enriching ontologies with additional expert-curated knowledge in a way that minimizes the workload of the human expert: based on the *attribute exploration* algorithm from formal concept analysis [19], several approaches have proposed structured interactive enumeration strategies of axioms of certain fragments of OWL which then are to be evaluated by an expert [20,21]. While similar in terms of the workflow, the major difference of these approaches to ours is that the axioms are not pre-specified but created on the fly and, therefore, the exploration may require (in the worst case exponentially) many human decisions.

9. Summary

In this paper, we proposed a methodology for supporting ontology revision based on logical criteria. We stated consistency criteria for revision states and introduced the notion of revision closure, based on which the revision of ontologies can be partially automated.

Even though a significant effort reduction can be achieved when axioms are chosen randomly for each expert decision, an evaluation of the axioms in an appropriate order usually yields a higher effort reduction. We introduced the notion of axiom impact, which can directly be used to define axiom ranking functions that work well for data of either high or low quality. In order to achieve significant improvements for data with an arbitrary validity ratio, we further refine the ranking functions to take into account the estimated validity ratio of the ontology under revision. We then show how the validity ratio can be learned on-the-fly over the course of the revision, which alleviates the user from having to provide such an estimate.

Moreover, we provide an efficient and elegant way of determining the revision closure and axiom impact by computing and updating structures called *decision spaces* which saved 75% of reasoner calls during our evaluation.

We present the implementation of the approach including an optimization based on partitioning, which significantly reduces the required computational effort. We evaluate our implementation in a revision of ontology-based annotations of scientific publications comprising over 25000 statements and show that

- On average, we were able to reduce the number of required evaluation decisions by 36% when the statements were reviewed in an arbitrary order, and by 55.4% when the unparameterized ranking techniques were used. The parametrized ranking technique almost achieved the maximum possible automatization (59.4% of evaluation decisions) thereby reducing the manual effort of revision by 59.3%. The gain of the parametrized compared to the unparameterized ranking functions is particularly important for datasets with a validity ratio close to 50% (we observed an up to 11.1% improvement), since for those datasets the potential of automatization cannot be fully exploited without parameterizing the ranking function.
- In case of large datasets with an unknown validity ratio, learning the validity ratio is particularly effective due to the law of large numbers. In our experiments, the proportion of automatically evaluated statements is nearly the same as in case where the validity ratio is known *a priori* and is used as a fixed parameter of *norm*, thereby making the assumption of known average validity ratio not necessary for axiom ranking.
- The proposed reasoning-based support is feasible for an interactive revision process requiring on average less than one second after each expert decision in our evaluation.

As part of our future work, we intend to study more general partitioning methods, e.g., [13], to increase the applicability of the partitioning optimization. Also the currently proposed partitioning could be further developed. The initially determined sets remain independent during the whole revision process. However, a single partition could potentially be further divided after some evaluation decisions, e.g., after rejecting an axiom a partition might fall apart into two partitions. We do not yet take such further refinements into account and simply stick to the originally defined partition, however it would be interesting to investigate whether such an additional partition refinement pays off. Another interesting approach in this direction would also be to study the effects of separating the ontology into parts that are not logically independent. In such a case, we might miss automatic decisions, but the potential performance gain, due to the reasoning with smaller subsets of the ontology, might compensate for this drawback.

References

[1] N. Nikitina, S. Rudolph, B. Glimm, Reasoning-supported interactive revision of knowledge bases, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), AAAI Press/IJCAI, 2011, pp. 1027–1032.

[2] N. Nikitina, B. Glimm, S. Rudolph, Wheat and chaff practically feasible interactive ontology revision, Proceedings of the 10th International Semantic Web Conference (ISWC 2011), vol. 7031, Springer-Verlag, 2011, pp. 487–503.

[3] C.E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: partial meet contraction and revision functions, Journal of Symbolic Logic 50 (1985) 510–530.

[4] M. Dalal, Investigations into a theory of knowledge base revision: preliminary report, in: Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-88), 1988, pp. 475–479.

[5] K. Satoh, Nonmonotonic reasoning by minimal belief revision, in: Proceedings of the International Conference on Fifth Generation Computer Systems, Springer, 1988, pp. 455–462.

[6] S. Schlobach, R. Cornet, Non-standard reasoning services for the debugging of description logic terminologies, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), Morgan Kaufmann, 2003, pp. 355–362.

[7] G. Qi, F. Yang, A survey of revision approaches in description logics, Proceedings of the 21st International Workshop on Description Logics (DL'08), vol. 353, CEUR, 2008.

[8] B. Motik, P.F. Patel-Schneider, B. Parsia, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax, W3C Recommendation, 2009, Available at <http://www.w3.org/TR/owl2-syntax/>.

[9] R. Shearer, I. Horrocks, Exploiting partial information in taxonomy construction, in: 8th International Semantic Web Conference (ISWC 2009), vol. 5823, 2009, pp. 569–584.

[10] OWL 2 Direct Semantics, OWL 2 Web Ontology Language Direct Semantics, W3C Recommendation, 2009, Available at <http://www.w3.org/TR/owl2-direct-semantics/>.

[11] B. Cuenca Grau, B. Motik, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web Ontology Language: Profiles, W3C Recommendation, 2009, Available at <http://www.w3.org/TR/owl2-profiles/>.

[12] Y. Kazakov, $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$, in: Proceedings of the 11th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2008), AAAI Press/The MIT Press, 2008, pp. 274–284.

[13] B. Konev, C. Lutz, D. Ponomaryov, F. Wolter, Decomposing description logic ontologies, in: Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR'10), 2010, pp. 236–246.

[14] W. OWL Working Group, OWL 2 Web Ontology Language: Document Overview, W3C Recommendation, 2009, Available at <http://www.w3.org/TR/owl2-overview/>.

[15] N. Nikitina, Semi-automatic revision of formalized knowledge, in: H. Coelho, R. Studer, M. Wooldridge (Eds.), ECAI Frontiers in Artificial Intelligence and Applications, vol. 215, IOS Press, 2010, pp. 1097–1098.

[16] C. Meilicke, H. Stuckenschmidt, A. Tamilin, Supporting manual mapping revision using logical reasoning, in: Proceedings of the 23rd Conference on Artificial Intelligence (AAAI 2008), AAAI Press, 2008, pp. 1213–1218.

[17] E. Jiménez-Ruiz, B.C. Grau, I. Horrocks, R.B. Llavori, Ontology integration using mappings: towards getting the right logical consequences, Proceedings of the 6th European Semantic Web Conference (ESWC 2009) of LNCS, vol. 5554, Springer-Verlag, 2009, pp. 173–187.

[18] E. Jiménez-Ruiz, B.C. Grau, I. Horrocks, R.B. Llavori, Building ontologies collaboratively using ContentCVS, Proceedings of the 22nd International Workshop on Description Logics (DL 2009) of CEUR Workshop Proceedings, vol. 477, CEUR-WS.org, 2009.

[19] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, 1997.

[20] S. Rudolph, Exploring relational structures via FLE, Conceptual Structures at Work: 12th International Conference on Conceptual Structures of LNCS, vol. 3127, Springer-Verlag, 2004, pp. 196–212.

[21] F. Baader, B. Ganter, B. Sertkaya, U. Sattler, Completing description logic knowledge bases using formal concept analysis, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), 2007, pp. 230–235.