Reasoning-Supported Interactive Revision of Knowledge Bases

Knowledge Representation, Reasoning and Logic::Description Logics and Ontologies
Web and Knowledge-based Information Systems::Ontologies

Abstract

Quality control is an essential task within ontology development projects especially when the knowledge formalization is partially automatized. In this paper, we propose a reasoning-based, interactive approach to support the revision of formalized knowledge. We state consistency criteria for revision states and introduce the notion of revision closure, based on which the revision of ontologies is partially automatized. 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. Finally, we develop the notion of *decision spaces*, which are structures for calculating and updating the revision closure and axiom impact. The use of decision spaces saves on average 75% of the costly reasoning operations during a revision.

1 Introduction

Manual knowledge formalization for real-world knowledge-intensive applications is a highly time-consuming task. An additional application of (semi-)automatic knowledge acquisition methods such as ontology learning or matching is, therefore, often considered to be 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.

In this paper, we present an approach that supports users in the process of such an inspection. In the following, we refer to such an inspection as ontology revision. Our goal is to provide a semi-automatic revision process in which a user inspects a set of candidate axioms and decides for each of them whether it is a desired logical consequence; based on this decision, we automatically discard or include yet unevaluated axioms depending on their logical relationships with the already evaluated axioms.

We illustrate the main challenges with an example in common First-Order Logic notation. Let us assume that we have already confirmed that the axioms

$$\forall x.(\text{Metal}(x) \to \text{Chemical\_Element}(x)) \tag{1}$$
$$\forall x.(\text{Chemical\_Element}(x) \to \text{Material}(x)) \tag{2}$$

belong to the desired consequences. Let us further assume that the following axioms are still to be evaluated:

$$\forall x.(\text{Copper}(x) \to \text{Material}(x)) \tag{3}$$
$$\forall x.(\text{Copper}(x) \to \text{Chemical\_Element}(x)) \tag{4}$$
$$\forall x.(\text{Copper}(x) \to \text{Metal}(x)) \tag{5}$$

If Axiom (3) is declined, i.e., it is an undesired consequence, we can immediately also decline Axioms (4) and (5) since accepting the axioms would implicitly lead to the undesired consequence (3). Similarly, if Axiom (5) is approved, Axioms (3) and (4) are implicit consequences and we can, therefore, already accept the two remaining axioms. Changing the order in which we inspect the axioms can, however, make further manual decisions necessary, e.g., if we start with declining Axiom (5), no automatic evaluation can be performed. It can be observed that

- a high grade of automation requires a good order for evaluating the axioms, and that
- approval and decline of an axiom has a different impact.

Which axioms have the biggest impact on decline or approval and which axioms can be automatically evaluated once a decision has been made can be determined with the help of algorithms for automated reasoning. Even for not very expressive knowledge representation formalisms, reasoning is an expensive task and in an interactive setting as described above, a crucial challenge is to minimize the amount of expensive reasoning tasks while maximizing the number of automated decisions. We transfer ideas developed for ontology classification [Shearer and Horrocks, 2009] to our problem of ontology revision thereby reducing the number of required reasoning tasks. For this, we introduce the notion of *decision spaces*, which exploit the characteristics of the logical entailment relation between axioms to maximize the amount of information gained by reasoning. We implemented the proposed strategy in a prototypical system. From our evaluation, it can be observed that, on the one hand, a considerable proportion of axioms can be evaluated automatically by our revision support, and, on the other hand, an application of decision spaces significantly reduces the number of required reasoning operations, resulting in a considerable performance gain.

The paper is organized as follows: In Section 2, we formalize the basic notions and ideas of reasoning-supported ontology revision. In Section 3, we define decision spaces, show how they can be updated in the process of the revision and how they can be used to determine a beneficial axiom order without additional reasoning operations. We evaluate our approach in Section 4. Finally, we discuss related approaches in Section 5 before we conclude in Section 6.

2 Revision of Knowledge Bases

The approach proposed here is applicable for any logic where taking all consequences is a closure operation, i.e., extensive ($\{\varphi\} \models \varphi$), monotone ($\Phi \models \varphi$ implies $\Phi \cup \Psi \models \varphi$), and idempotent ($\Phi \models \varphi$ and $\Phi \cup \{\varphi\} \models \psi$ imply $\Phi \models \psi$). Moreover, we presume the existence of a decision procedure for logical entailment.

The revision of a knowledge base $\mathcal{K}$ aims at a separation of its axioms (i.e., logical statements) into two disjoint sets: the set of intended consequences $\mathcal{K}^{\models}$ and the set of unintended consequences $\mathcal{K}^{\not\models}$. This motivates the following definitions.

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

Since we expect that the deductive closure of the intended consequences in $\mathcal{K}^{\models}$ must not contain unintended consequences, we introduce the notion of *consistency* for revision states. 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 and Closure) *A (complete or incomplete) revision state* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *is* consistent *if there is no* $\alpha \in \mathcal{K}^{\not\models}$ *such that* $\mathcal{K}^{\models} \models \alpha$*. The* revision closure $\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *of* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *is* $(\mathcal{K}, \mathcal{K}_c^{\models}, \mathcal{K}_c^{\not\models})$ *with* $\mathcal{K}_c^{\models} := \{\alpha \in \mathcal{K} \mid \mathcal{K}^{\models} \models \alpha\}$ *and* $\mathcal{K}_c^{\not\models} := \{\alpha \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha\} \models \beta$ *for some* $\beta \in \mathcal{K}^{\not\models}\}$.

We can show the following useful properties of the closure of consistent revision states:

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

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

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

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

Algorithm 1 Interactive Knowledge Base Revision

Input: $(\mathcal{K}, \mathcal{K}_0^{\models}, \mathcal{K}_0^{\not\models})$ a consistent revision state
Output: $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ a complete and consistent revision state

1: $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) \leftarrow \mathrm{clos}(\mathcal{K}, \mathcal{K}_0^{\models}, \mathcal{K}_0^{\not\models})$
2: **while** $\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models} \neq \mathcal{K}$ **do**
3:     choose $\alpha \in \mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})$
4:     **if** expert confirms α **then**
5:         $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) \leftarrow \mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models})$
6:     **else**
7:         $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) \leftarrow \mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\alpha\})$
8:     **end if**
9: **end while**

Instead of starting with empty sets for $\mathcal{K}_0^{\models}$ and $\mathcal{K}_0^{\not\models}$, we can initialize the latter sets with approved and declined axioms from a previous revision or add axioms of the knowledge base that is being developed to $\mathcal{K}_0^{\models}$. We can further initialize $\mathcal{K}_0^{\not\models}$ with axioms that express inconsistency and unsatisfiability of predicates (i.e. of classes or relations) in $\mathcal{K}$, which we assume to be unintended consequences.

In line 3, an axiom is chosen that is evaluated next. As motivated in the introduction, a random decision can have a detrimental effect on the amount of manual decisions. Ideally, we want to rank the axioms and choose one that allows for a high number of consequential automatic decisions. For this purpose, we introduce the following notion of *axiom impact*.

Definition 3 (Impact) *Let* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *be a consistent revision state with* $\alpha \in \mathcal{K}$ *and let* $?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) := |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})|$*. The* approval impact *of* α *is defined as:*

$$\textit{impact}^{+}(\alpha) = ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models}))$$

and the decline impact *as:*

$$\textit{impact}^{-}(\alpha) = ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\alpha\})).$$

The guaranteed impact *of* α *is:*

$$\textit{guaranteed}(\alpha) = \min(\textit{impact}^{+}(\alpha), \textit{impact}^{-}(\alpha))$$

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* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *with* $\mathcal{K}^{\not\models} \neq \emptyset$*, the according* decision space $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})} = (\mathcal{K}^{?}, E, C)$ *contains the set*

$$\mathcal{K}^{?} := \mathcal{K} \setminus (\{\alpha \mid \mathcal{K}^{\models} \models \alpha\} \cup \{\alpha \mid \mathcal{K}^{\models} \cup \{\alpha\} \models \beta, \beta \in \mathcal{K}^{\not\models}\})$$

of unevaluated axioms together with two binary relations E *(read: entails) and* C *(read: conflicts) defined by*

- $\alpha E \beta$ *iff* $\mathcal{K}^{\models} \cup \{\alpha\} \models \beta$
- $\alpha C \beta$ *iff* $\mathcal{K}^{\models} \cup \{\alpha, \beta\} \models \gamma$ *for some* $\gamma \in \mathcal{K}^{\not\models}$

The requirement that $\mathcal{K}^{\not\models} \neq \emptyset$ is without loss of generality since we can always add an axiom that expresses a contradiction (an inconsistency), which is clearly undesired. As a direct consequence of this definition, we have $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})} = \mathbb{D}_{\mathrm{clos}(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$. Also the following properties are immediate from the above definition:

Lemma 2 *Given* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})} = (\mathcal{K}^{?}, E, C)$ *for a revision state* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *with* $\mathcal{K}^{\not\models} \neq \emptyset$*, then*

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

On the other hand, the properties established in the preceding lemma are characteristic:[1]

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

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* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$ *isomorphic to* $(V, \underline{E}, \underline{C})$*.*

Proof. As a very basic formalism, we choose propositional logic as KR language. Let $\mathcal{K}$ contain one atomic proposition p_v for every $v \in V$, let $\mathcal{K}^{\models} = \{p_v \to p_{v'} \mid v\underline{E}v'\} \cup \{\neg p_v \vee \neg p_{v'} \mid v\underline{C}v'\}$ and let $\mathcal{K}^{\not\models} = \{false\}$. First observe that $\mathcal{K}^{?} = \mathcal{K}$. Next, we claim that the function $f : V \to \mathcal{K}$ with $v \mapsto p_v$ is an isomorphism between $(V, \underline{E}, \underline{C})$ and $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$. Clearly, f is a bijection. Moreover, $v\underline{E}v'$ implies $p_v E p_{v'}$ by modus ponens since $p_v \to p_{v'} \in \mathcal{K}^{\models}$. Likewise, $v\underline{C}v'$ implies $p_v C p_{v'}$ due to $\neg p_v \vee \neg p_{v'} \in \mathcal{K}^{\models}$. 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 easily verified that this interpretation is a model of $\mathcal{K}^{\models}$ and additionally satisfies p_v as well as $\neg p_{v'}$, hence $\mathcal{K}^{\models} \cup \{p_v\} \not\models p_{v'}$ and consequently $p_v E p_{v'}$ cannot hold, so we have a contradiction.

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 easily verified that this interpretation is a model of $\mathcal{K}^{\models}$ and additionally satisfies p_v as well as $p_{v'}$, hence $\mathcal{K}^{\models} \cup \{p_v, p_{v'}\} \not\models$ *false* and consequently $p_v C p_{v'}$ cannot hold, so we have a contradiction. □

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 ${\uparrow}\alpha = \{\beta \mid \alpha E \beta\}$ and ${\downarrow}\alpha = \{\beta \mid \beta E \alpha\}$. Moreover, we let $\wr\alpha = \{\beta \mid \alpha C \beta\}$.

Lemma 4 *Given* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})} = (\mathcal{K}^{?}, E, C)$ *for a revision state* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *such that* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) = \mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ *with* $\mathcal{K}^{\not\models} \neq \emptyset$ *and* $\alpha \in \mathcal{K}^{?}$*, then*

1. $\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models}) = (\mathcal{K}, \mathcal{K}^{\models} \cup {\uparrow}\alpha, \mathcal{K}^{\not\models} \cup \wr\alpha)$ *and*
2. $\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\alpha\}) = (\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup {\downarrow}\alpha)$.
3. *impact*$^{+}(\alpha) = |{\uparrow}\alpha| + |\wr\alpha|$
4. *impact*$^{-}(\alpha) = |{\downarrow}\alpha|$

Proof.

1. By definition of closures, we have
   $\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models})$
   $= (\mathcal{K}, \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha\} \models \beta\},$
   $\{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha, \beta\} \models \gamma \text{ for some } \gamma \in \mathcal{K}^{\not\models}\})$.
   By definition of the entails and conflicts relation
   $= (\mathcal{K}, \mathcal{K}^{\models} \cup \{\beta \in \mathcal{K}^{?} \mid \alpha E \beta\}, \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K}^{?} \mid \alpha C \beta\})$
   and by definition of ${\uparrow}\alpha$ and $\wr\alpha$
   $= (\mathcal{K}, \mathcal{K}^{\models} \cup {\uparrow}\alpha, \mathcal{K}^{\not\models} \cup \wr\alpha)$.

2. Since $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ is already closed, we have

$$
\begin{array}{ll}
 & \mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\alpha\}) \\
= & (\mathcal{K}, \mathcal{K}^{\models}, \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\beta\} \models \gamma \\
 & \quad \text{for some } \gamma \in (\mathcal{K}^{\not\models} \cup \{\alpha\})\}) \\
= & (\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K}^{?} \mid \mathcal{K}^{\models} \cup \{\beta\} \models \alpha\}).
\end{array}
$$

By definition of the conflicts relation, we have

$= \quad (\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K}^{?} \mid \alpha C \beta\}),$

which by definition of $\downarrow\!\alpha$ gives

$= \quad (\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \downarrow\!\alpha).$

3. By Definition 3 we have $\mathrm{impact}^{+}(\alpha)$

$= \quad ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models}))$

by Definition 2

$= \quad ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathcal{K}, \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha\} \models \beta\},$
$\quad \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha, \beta\} \models \gamma \text{ for some } \gamma \in \mathcal{K}^{\not\models}\})$

by the definition of $?(\cdot)$ (Definition 3)

$= \quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})| -$
$\quad |\mathcal{K} \setminus (\{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha\} \models \beta\} \cup$
$\quad \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\alpha, \beta\} \models \gamma \text{ for some } \gamma \in \mathcal{K}^{\not\models}\})|$

by definition of the entails and conflicts relations

$= \quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})| - |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \{\beta \in \mathcal{K}^{?} \mid \alpha E \beta\}$
$\quad \cup \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K}^{?} \mid \alpha C \beta\})|$

by definition of $\uparrow$ and $\wr$

$= \quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})| - |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \uparrow\!\alpha \cup \mathcal{K}^{\not\models} \cup \wr\alpha)|$

$= \quad |\mathcal{K}| - (|\mathcal{K}^{\models}| + |\mathcal{K}^{\not\models}|)$
$\quad -(|\mathcal{K}| - (|\mathcal{K}^{\models}| + |\uparrow\!\alpha| + |\mathcal{K}^{\not\models}| + |\wr\alpha|))$

$= \quad |\uparrow\!\alpha| + |\wr\alpha|$

4. By Definition 3 we have $\mathrm{impact}^{-}(\alpha)$

$= \quad ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\alpha\}))$

by Definition 2

$= \quad ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathcal{K}, \mathcal{K}^{\models},$
$\quad \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\beta\} \models \alpha\})$

by the definition of $?(\cdot)$ (Definition 3)

$= \quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})| -$
$\quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K} \mid \mathcal{K}^{\models} \cup \{\beta\} \models \alpha\})|$

by definition of the entails relation

$= \quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})|$
$\quad -|\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models} \cup \{\beta \in \mathcal{K}^{?} \mid \beta E \alpha\})|$

by definition of $\downarrow$

$= \quad |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})| - |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models} \cup \downarrow\!\alpha)|$

$= \quad |\mathcal{K}| - (|\mathcal{K}^{\models}| + |\mathcal{K}^{\not\models}|)$
$\quad -(|\mathcal{K}| - (|\mathcal{K}^{\models}| + |\mathcal{K}^{\not\models}| + |\downarrow\!\alpha|))$

$= \quad |\downarrow\!\alpha|$

$\square$

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 decision 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 to R9, which describe the connections 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.

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

An analysis of the dependencies between the rules R0 to 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 to R9 can be condensed into the following operations:

$$
\begin{array}{lcl}
E & \leftarrow & E^{*} \\
C & \leftarrow & E \circ (C \cup C^{-}) \circ E^{-} \\
\overline{C} & \leftarrow & E^{-} \circ (\overline{C} \cup Id \cup \overline{C}^{-}) \circ E \\
\overline{E} & \leftarrow & E^{-} \circ (\overline{C} \circ C \cup \overline{E}) \circ E^{-}
\end{array}
$$

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

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

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

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 relations E and C of a decision space by deriving all derivable (non-)entailments and (non-)conflicts from the above correspondences before executing entailment checks. We employ the following optimizations: We split computations into several subsequent ones where appropriate; 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., $(R \cup \{(\alpha, \beta)\})^{*} \setminus R^{*}$)); finally, we also avoid redundant computations and reduce the input sets for the composition of relations ($\circ$) by explicitly bookkeeping sets E',C',$\overline{C}'$, and $\overline{E}'$ containing only the instances newly added in the current step.

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

Algorithm 2 Decision Space Completion

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

1: $E \leftarrow E^*$
2: $C \leftarrow E \circ C \circ E^-$
3: $C \leftarrow C \cup C^-$
4: $\overline{C} \leftarrow E^- \circ \overline{C} \cup Id_{\mathcal{K}^?} \circ E$
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** $E \cup \overline{E} \neq \mathcal{K}^? \times \mathcal{K}^?$ **do**
9: pick one $(\alpha, \beta) \in \mathcal{K}^? \times \mathcal{K}^? \setminus (E \cup \overline{E})$
10: **if** $\mathcal{K}^{\models} \cup \{\alpha\} \models \beta$ **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** $C \cup \overline{C} \neq \mathcal{K}^? \times \mathcal{K}^?$ **do**
28: pick one $(\alpha, \beta) \in \mathcal{K}^? \times \mathcal{K}^? \setminus (C \cup \overline{C})$
29: **if** $\mathcal{K}^{\models} \cup \{\alpha, \beta\} \models \gamma$ for some $\gamma \in \mathcal{K}^{\not\models}$ **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**

Lemma 5 *Let $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ be a revision state with $\mathcal{K}^{\not\models} \neq \emptyset$ and $E, \overline{E}, C, \overline{C}$ (possibly empty) subsets of the entailment and conflicts relations. We denote the size $|\mathcal{K}|$ of $\mathcal{K}$ with n. Given $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ 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.*

Proof. We first note that $\mathcal{K}^?$ is bounded by n since $|\mathcal{K}^?| = |\mathcal{K}| - (|\mathcal{K}^{\models}| + |\mathcal{K}^{\not\models}|)$. 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 $\mathcal{K}$. 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 $\mathcal{K}$ 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 relation is again over axioms from $\mathcal{K}$ 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 the 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. $\square$

Lemma 6 *Let $\mathcal{L}$ be a logic where taking all consequences is a closure operation and such that there is a decision procedure for logical entailment of complexity $c(n)$ for n the size of the input to the entailment checking procedure. Let $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ be a revision state with $\mathcal{K}^{\not\models} \neq \emptyset$, $|\mathcal{K}| := n$ and where the axioms in $\mathcal{K}$ are expressible in $\mathcal{L}$, and let $E, \overline{E}, C, \overline{C}$ be (possibly empty) subsets of the entailment and conflicts relations. There is a polynomial p such that the runtime of Algorithm 2, given $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$ 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 $\alpha \in \mathcal{K}^?$, and hence α is added to the set $\mathcal{K}^{\models}$ of wanted consequences.

Lemma 7 *Let $\mathbb{D}_{(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})} = (\mathcal{K}^?, E, C)$, $\alpha \in \mathcal{K}^?$ and $\mathbb{D}_{(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models})} = (\mathcal{K}^?_{\text{new}}, E', C')$. Then*

- *$\mathcal{K}^?_{\text{new}} = \mathcal{K}^? \setminus (\uparrow\alpha \cup \wr\alpha)$,*

Algorithm 3 Decision Space Update on Approving α

Input: $\mathbb{D}_{(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})}$ a decision space, $\alpha \in \mathcal{K}^?$ an axiom
Output: $\mathbb{D}_{(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models})}$ the updated decision space

$\mathcal{K}^? \leftarrow \mathcal{K}^? \setminus (\uparrow\alpha \cup \wr\alpha)$
$E \leftarrow E \cap (\mathcal{K}^? \times \mathcal{K}^?)$
$C \leftarrow C \cap (\mathcal{K}^? \times \mathcal{K}^?)$
$\overline{C} \leftarrow E^- \circ E$
$\overline{E} \leftarrow E^- \circ \overline{C} \circ C \circ E^-$
execute lines 8–38 from Alg. 2

- $\beta E \gamma$ *implies* $\beta E' \gamma$ *for* $\beta, \gamma \in \mathcal{K}^?_{\text{new}}$, *and*
- $\beta C \gamma$ *implies* $\beta C' \gamma$ *for* $\beta, \gamma \in \mathcal{K}^?_{\text{new}}$.

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 takes advantage of these correspondences when fully determining the updated decision space.

Lemma 8 *Let* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$ *be a decision space,* $\alpha \in \mathcal{K}^?$ *an axiom. We denote the size* $|\mathcal{K}|$ *of* $\mathcal{K}$ *with* n. *Given* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$ *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. The executions in the first part of Algorithm 3 can be performed 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. □

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* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})} = (\mathcal{K}^?, E, C)$, $\alpha \in \mathcal{K}^?$ *and* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models}\cup\{\alpha\})} = (\mathcal{K}^?_{\text{new}}, E', C')$. *Then*

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

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 Alg. 1 after the accept (line 5) or decline revision step (line 7), respectively.

Lemma 10 *Let* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$ *be a decision space,* $\alpha \in \mathcal{K}^?$ *an axiom. We denote the size* $|\mathcal{K}|$ *of* $\mathcal{K}$ *with* n. *Given* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$ *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 executions in the first part of Algorithm 4 before the while loop 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. □

Algorithm 4 Decision Space Update on Declining α

Input: $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models})}$ a decision space, $\alpha \in \mathcal{K}^?$ an axiom
Output: $\mathbb{D}_{(\mathcal{K},\mathcal{K}^{\models},\mathcal{K}^{\not\models}\cup\{\alpha\})}$ the updated decision space

```
K? ← K? \ ↓α,
E ← E ∩ (K? × K?)
Ē ← Ē ∩ (K? × K?)
C ← C ∩ (K? × K?)
C̄ ← E⁻ ∘ E
while C ∪ C̄ ≠ K? × K? do
    pick one (β, γ) ∈ K? × K? \ (C ∪ C̄)
    if K^⊨ ∪ {β, γ} ⊨ α then
        C ← C ∪ (E ∘ {(β, γ), (γ, β)} ∘ E⁻)
    else
        C̄ ← C̄ ∪ (E⁻ ∘ {(β, γ), (γ, β)} ∘ E)
    end if
end while
```

4 Evaluation

For a first evaluation of the developed methodology, we choose a scenario motivated by ontology-supported literature search. The hand-crafted *NanOn* ontology models the scientific domain of nano technology, including substances, structures, procedures used in that domain. The ontology, denoted here with $\mathcal{O}$, is specified in the Web Ontology Language OWL DL [OWL Working Group, 2009] and comprises 2,289 logical axioms. The project associated to NanOn aims at developing techniques to automatically analyze scientific documents for the occurrence of NanOn concepts. When such concepts are found, the document is automatically annotated with NanOn concepts to facilitate topic-specific information retrieval on a fine-grained level. Since total accuracy of the automatically added annotations (which can be seen as logical axioms expressing factual knowledge) cannot be guaranteed, they need to be inspected by human experts, which provides a natural application scenario for our approach.

For our evaluation, we employed tools for automated textual analysis to produce a set of document annotations, the validity of which was then manually evaluated. This provided us with sets of valid and invalid annotation facts (denoted by $\mathcal{A}^+$ and $\mathcal{A}^-$, respectively). To investigate how the a priori quality of each axiom set influences the results, we created six distinct annotation sets S_1 to S_6 using different annotation methods. The different methods result in different *validity ratios* $|\mathcal{A}^+|/(|\mathcal{A}^+|+|\mathcal{A}^-|)$ of the datasets, where $|S|$ denotes the cardinality of a set S. The size of each set as well as the corresponding validity ratio in percent are shown in the headers of Table 1.

We then 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 concept unsatisfiability. We then 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.[2]

For each set, Table 1 shows the effects of the different choice functions $impact^+$, $guaranteed$, $impact^-$ by measuring the reduction of expert decisions compared to evaluating the whole set manually (1st column for each set), followed by the number of necessary reasoner calls with and without the use of decision spaces (2nd and 3rd column, respectively). As a baseline, we also include the reduction of expert decision when choosing axioms randomly. We did not use decision spaces for the calculation of the baseline, since axiom impact is not taken into account. The upper bound for the manual effort reduction was obtained by applying the "impact oracle" function defined by

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

	S_1 (54, 94%)			S_2 (60, 100%)		
$impact^+$	69%	4,677	36,773	83%	2,584	18,702
$guaranteed$	48%	11,860	51,677	65%	8,190	55,273
$impact^-$	9%	17,828	46,461	12%	20,739	67,625
upper bound	74%	4,110	11,399	83%	2,645	27,850
random	45%	-	1,291	60%	-	1,090

	S_3 (40, 45%)			S_4 (35, 48%)		
$impact^+$	20%	3,137	26,759	29%	2,198	15,601
$guaranteed$	43%	3,914	27,629	43%	3,137	18,367
$impact^-$	28%	9,947	46,461	31%	7,309	10,217
upper bound	48%	3,509	13,202	51%	2,177	7,002
random	31%	-	764	31%	-	534

	S_5 (26, 26%)			S_6 (72, 12%)		
$impact^+$	8%	1,778	11,443	13%	9,352	212,041
$guaranteed$	39%	1,290	6,647	54%	8,166	99,586
$impact^-$	54%	954	1,438	76%	6,797	16,922
upper bound	54%	801	1,989	76%	5,219	19,861
random	41%	-	212	57%	-	1,065

Table 1: Revision results for different axiom choosing strategies

The results of the evaluation show that:

- Decision spaces save on average 75% of reasoner calls, which leads to a considerable overall performance gain given that, on average, 88% of computation time in our experiments is spent on reasoning.
- Compared to an all manual revision, a significant effort reduction of on average 44% is already achieved when axioms are chosen randomly for each expert decision by automatically approving and declining axioms based on the computed revision closure. However it leaves space for improvement. The "impact oracle" manages to reduce the manual effort of revision on average by 64%.
- If the ratio of approved axioms is rather high or rather low, *impact*$^+$ or *impact*$^-$, respectively, perform best.
- If the ratios of approved and declined axioms are more or less equal, the guaranteed impact is the best choice.

Therefore, the appropriate axiom choosing strategy has to be selected based on the expected ratio of valid axioms. We see that an application of the most suitable axiom choosing strategy for each validity ratio, listed in grey rows, yields on average an effort reduction of 61%, which is 15% higher than the performance of *random* and only 3% less than the effort reduction achieved by the "impact oracle".

5 Related Work

We are aware of two approaches for supporting the revision of ontological data based on logical appropriateness: an approach by Meilicke et al. [2008] and another one called *ContentMap* by Jiménez-Ruiz et al. [2009b]. Both approaches are applied in the context of mapping revision. An extension of *ContentMap* called *ContentCVS* [Jiménez-Ruiz *et al.*, 2009a] 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, the focus of ContentMap and ContentCVS lies within 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 out of scope. In contrast to our approach, which requires at most a polynomial number of entailment checks, ContentMap and ContentCVS require an exponential number of reasoning operations compared to the size of the ontology under revision. The reason for this is that ContentMap and ContentCVS 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.

[2]http://www.hermit-reasoner.com

Another strand of work is related to the overall motivation of enriching knowledge bases 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 (FCA) [Ganter and Wille, 1997], several works have proposed structured interactive enumeration strategies of inclusion dependencies or axioms of certain fragments of description logics which then are to be evaluated by the expert [Rudolph, 2004; Baader *et al.*, 2007]. 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.

6 Conclusions and Future Work

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 automatized.

Even though a significant effort reduction can be achieved when axioms are chosen randomly for each expert decision, an evaluation of axioms in an appropriate order usually yields a higher effort reduction. We introduced the notion of axiom impact which is used to determine a beneficial order of evaluation. Depending on the expected ratio of approved axioms, $impact^{+}$, $impact^{-}$ or the guaranteed impact can be employed in order to achieve a higher effort reduction. In fact, in three out of six cases during the evaluation, the maximum possible effort reduction was achieved when employing the best suitable axiom choosing strategy.

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

In future, we would like to investigate how the axiom choosing strategy can be adjusted according to the current ratio of approved axioms. Another open question is how the axioms under investigation can be efficiently partitioned into logically independent sets which can be reviewed independently.

References

[Baader *et al.*, 2007] Franz Baader, Bernhard Ganter, Baris Sertkaya, and Ulrike Sattler. Completing description logic knowledge bases using formal concept analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 230–235, 2007.

[Ganter and Wille, 1997] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, 1997.

[Jiménez-Ruiz *et al.*, 2009a] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga Llavori. Building ontologies collaboratively using ContentCVS. In *Proceedings of the 22nd International Workshop on Description Logics (DL 2009)*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[Jiménez-Ruiz *et al.*, 2009b] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga Llavori. Ontology integration using mappings: Towards getting the right logical consequences. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, volume 5554 of *LNCS*, pages 173–187. Springer-Verlag, 2009.

[Meilicke *et al.*, 2008] Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Supporting manual mapping revision using logical reasoning. In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI 2008)*, pages 1213–1218. AAAI Press, 2008.

[OWL Working Group, 2009] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

[Rudolph, 2004] Sebastian Rudolph. Exploring relational structures via FLE. In *Conceptual Structures at Work: 12th International Conference on Conceptual Structures*, volume 3127 of *LNCS*, pages 196–212. Springer-Verlag, 2004.

[Shearer and Horrocks, 2009] Rob Shearer and Ian Horrocks. Exploiting partial information in taxonomy construction. In *8th International Semantic Web Conference (ISWC 2009)*, volume 5823, pages 569–584, 2009.