Reasoning-Supported Interactive Revision of Knowledge Bases

Nadeschda Nikitina[1], Sebastian Rudolph[1], and Birte Glimm[2]

[1] Karlsruhe Institute of Technology, Germany,
[1] {nikitina,rudolph}@kit.edu,
[2] The University of Oxford, Department of Computer Science, UK
[2] birte.glimm@cs.ox.ac.uk

Abstract. We propose a method for controlling the quality of (semi-)automatically acquired axioms. We combine the manual inspection of axioms with automatic evaluation decisions and propose *decision spaces* as a means to efficiently compute which decisions can be automatized and which axiom evaluation order is beneficial.

1 Introduction

Manual knowledge formalization for real-world knowledge-intensive applications is highly time-consuming. An application of (semi-)automatic knowledge acquisition methods such as ontology learning or matching is, therefore, often considered a reasonable way to reduce the cost of ontology development. Automatically acquired knowledge usually has to be manually inspected; either partially, to estimate the overall quality, or even fully, to maintain high quality standards.

So far, the knowledge representation community has been focusing on restoring the consistency of ontologies enriched with new axioms as done in various belief revision and repair approaches. Such approaches, however, are not directly suited in case a more restrictive quality control is required. We support an exhaustive manual inspection of newly acquired axioms before adding the selected ones into the ontology and call this process *interactive ontology revision*. Once a decision (add or not, i.e., accept or decline) has been made, we determine which other axioms can be evaluated automatically by exploiting logical dependencies between axioms.

We illustrate the main challenges with an example in which we have already confirmed that the axioms

$$\mathrm{Metal} \sqsubseteq \mathrm{Chemical\_Element} \tag{1}$$

$$\mathrm{Chemical\_Element} \sqsubseteq \mathrm{Material} \tag{2}$$

belong to the desired consequences, while the following axioms are still to be evaluated:

$$\mathrm{Copper} \sqsubseteq \mathrm{Material} \tag{3}$$

$$\mathrm{Copper} \sqsubseteq \mathrm{Chemical\_Element} \tag{4}$$

$$\mathrm{Copper} \sqsubseteq \mathrm{Metal} \tag{5}$$

If Axiom (3) is declined, 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, which can be approved automatically. If we start, however, with declining Axiom (5), no automatic evaluation can be performed. It can be observed that

– a high grade of automation requires a good evaluation order, and
– approval and decline decisions have a different impact.

Which axioms have the highest 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 it is crucial to minimize the number of reasoning tasks while maximizing the number of automated decisions. We reduce the number of reasoning tasks by transferring ideas for ontology classification [8] to our problem. 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. From the evaluation of our prototypical system, it can be observed that a considerable proportion of axioms can be evaluated automatically. Furthermore, decision spaces significantly reduce the number of required reasoning operations, resulting in a considerable performance gain.

In the next sction we formalize the basic notions and ideas; in Section 3, we define decision spaces, how they can be updated, and how they help to determine a beneficial axiom order. Our evaluation is presented in Section 4. Finally, we discuss related approaches in Section 5 before we conclude in Section 6. Further details and proofs can be found in a technical report [4]. The paper is an adaptation of our IJCAI'2011 paper [5].

2 Revision of Knowledge Bases

The approach proposed here is not only applicable to Description Logics, but to 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 wanted consequences $\mathcal{K}^{\models}$ and the set of unwanted 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* $\mathcal{K}^{\models} \subseteq \mathcal{K}, \emptyset \neq \mathcal{K}^{\not\models} \subseteq \mathcal{K}$*, 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 wanted consequences in $\mathcal{K}^{\models}$ must not contain unwanted consequences, we introduce the notion of *consistency* for 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**

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* $\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$.

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

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 unwanted 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* $?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) := |\mathcal{K} \setminus (\mathcal{K}^{\models} \cup \mathcal{K}^{\not\models})|$. *For an axiom* α,

- *the* approval impact *is:* $\mathit{impact}^{+}(\alpha) = ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models} \cup \{\alpha\}, \mathcal{K}^{\not\models}))$,
- *the* decline impact *is:* $\mathit{impact}^{-}(\alpha) = ?(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models}) - ?(\mathrm{clos}(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models} \cup \{\alpha\}))$,
- *the* guaranteed impact *is:* $\mathit{guaranteed}(\alpha) = \min(\mathit{impact}^{+}(\alpha), \mathit{impact}^{-}(\alpha))$.

The approval (decline) impact of an axiom α is determined by the number of automatically evaluated axioms in case α is approved (declined), while the guaranteed impact is the minimum of the two impact functions.

In the example from Section 1, Axioms (3), (4) and (5) have an approval impact of 0, 1, and 2, a decline impact of 2, 1, and 0, and a guaranteed impact of 0, 1, and 0, respectively. We show in the evaluation that the ratio of accepted axioms to all axioms that are to be evaluated can be used to determine which impact function is best.

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*

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

of unevaluated axioms together with two binary relations, E (entails) and C (conflicts) on $\mathcal{K}^{?}$*:*

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

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* $(\mathcal{K}, \mathcal{K}^{\models}, \mathcal{K}^{\not\models})$, $\mathcal{K}^{\not\models} \neq \emptyset$,

- 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.*

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

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* $(V, \underline{E}, \underline{C})$*.*

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

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. $\mathit{impact}^{+}(\alpha) = |\uparrow\alpha| + |\wr\alpha|$
4. $\mathit{impact}^{-}(\alpha) = |\downarrow\alpha|$

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:

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

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 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: $E' \leftarrow \mathtt{transupdatediff}(E, (\alpha, \beta))$
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**

Algorithm 3 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

1: $\mathcal{K}^? \leftarrow \mathcal{K}^? \setminus \downarrow\alpha$,
2: $E \leftarrow E \cap (\mathcal{K}^? \times \mathcal{K}^?)$
3: $\overline{E} \leftarrow \overline{E} \cap (\mathcal{K}^? \times \mathcal{K}^?)$
4: $C \leftarrow C \cap (\mathcal{K}^? \times \mathcal{K}^?)$
5: $\overline{C} \leftarrow E^- \circ E$
6: **while** $C \cup \overline{C} \neq \mathcal{K}^? \times \mathcal{K}^?$ **do**
7: pick one $(\beta, \gamma) \in \mathcal{K}^? \times \mathcal{K}^? \setminus (C \cup \overline{C})$
8: **if** $\mathcal{K}^{\models} \cup \{\beta, \gamma\} \models \alpha$ **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**

Algorithm 4 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

1: $\mathcal{K}^? \leftarrow \mathcal{K}^? \setminus (\uparrow\alpha \cup \wr\alpha)$
2: $E \leftarrow E \cap (\mathcal{K}^? \times \mathcal{K}^?)$
3: $C \leftarrow C \cap (\mathcal{K}^? \times \mathcal{K}^?)$
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 Alg. 2

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 computations 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.

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 5. *Let* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^\models,\mathcal{K}^\not\models)} = (\mathcal{K}^?, E, C)$*,* $\alpha \in \mathcal{K}^?$*,* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^\models \cup\{\alpha\},\mathcal{K}^\not\models)} = (\mathcal{K}^?_{\text{new}}, E', C')$*. Then*
- $\mathcal{K}^?_{\text{new}} = \mathcal{K}^? \setminus (\uparrow\alpha \cup \wr\alpha)$*,*
- $\beta E\gamma$ *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 4 takes advantage of these correspondences when fully determining the updated decision space.

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 6. *Let* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^\models,\mathcal{K}^\not\models)} = (\mathcal{K}^?, E, C)$*,* $\alpha \in \mathcal{K}^?$*,* $\mathbb{D}_{(\mathcal{K},\mathcal{K}^\models,\mathcal{K}^\not\models \cup\{\alpha\})} = (\mathcal{K}^?_{\text{new}}, E', C')$*. 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 3 implements the respective

Table 1. Characteristics of the evaluated datasets

dataset	size	validity ratio
S_1	54	94%
S_2	60	100%
S_3	40	45%

dataset	size	validity ratio
S_4	35	48%
S_5	26	26%
S_6	72	12%

decision space update, additionally exploiting that new conflicts can only arise from derivability of the newly declined axiom α.

Algorithms 4 and 3 have to be called in Alg. 1 after the accept (line 5) or decline revision step (line 7), respectively.

For n the number of involved axioms, Algorithms 2, 4, and 3 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, which then also applies to our algorithm.

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 [6] 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 followed by the corresponding validity ratio in percent are shown in 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 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

Table 2. Revision results for different axiom choosing strategies

	impact$^+$			*guaranteed*			*impact*$^-$			upper bound			rand
S_1	69%	4,677	36,773	48%	11,860	51,677	9%	17,828	46,461	74%	4,110	11,399	45%
S_2	83%	2,584	18,702	65%	8,190	55,273	12%	20,739	67,625	83%	2,645	27,850	60%
S_3	20%	3,137	26,759	43%	3,914	27,629	28%	9,947	46,461	48%	3,509	13,202	31%
S_4	29%	2,198	15,601	43%	3,137	18,367	31%	7,309	10,217	51%	2,177	7,002	31%
S_5	8%	1,778	11,443	39%	1,290	6,647	54%	954	1,438	54%	801	1,989	41%
S_6	13%	9,352	212,041	54%	8,166	99,586	76%	6,797	16,922	76%	5,219	19,861	57%

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 2 shows the effects of the different choice functions *impact*$^+$, *guaranteed* and *impact*$^-$ by measuring the reduction of expert decisions compared to evaluating the whole set manually (1st column for each axiom set and choice function), followed by the number of necessary reasoner calls when using decision spaces (2nd column for each axiom set and choice function) and the corresponding number of reasoner calls without the use of decision spaces (3rd column for each axiom set and choice function). As a baseline, we also include the reduction of expert decisions when choosing axioms randomly (last column). The upper bound for the manual effort reduction was obtained by applying the "impact oracle" function:

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

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 within the methods of the reasoner according to our profiling measurements. The experiments with the same datasets took on average 8 times longer without the application of decision spaces.
- 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, there is still some space for improvement, since 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.

From these observations we can conclude that 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".

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

5 Related Work

In our previous work [3], 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. Prior to the interactive revision, a minimal non-redundant subset of axioms under investigation is identified and then reviewed by the expert thereby not requiring the expensive computation of axiom impacts after each expert decision.

In addition to our own work, we are aware of two approaches for supporting the revision of ontological data based on logical appropriateness: Meilicke et al. [2] and Jiménez-Ruiz et al. [1] propose two approaches, both of which are applied in the context of mapping revision. In 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 Definition 1. Similarly to our approach, Meilicke et al. aim at reducing the manual effort of mapping revision by relying on a heuristic notion of impact. The approach is, however, difficult to generalize to the revision of ontologies since the notion of impact is based on the hypothetically possible number of mapping axioms for two ontologies O_1 and O_2 and further relies on the assumption that the set of possible mapping axioms is mostly disjoint from the axioms in $O_1 \cup O_2$. This assumption is justified in case of mapping revision, since axioms in O_1 (O_2) usually refer only to entities from O_1 (O_2), while mapping axioms link entities from O_1 and O_2. For interactive ontology revision in general, however, the axioms that are to be revised are typically not disjoint from the already evaluated axioms.

The focus of ContentMap [1] lies within the visualization of consequences and user guidance in case of difficult evaluation decisions, while the minimization of the manual and computational effort required for the revision is out of scope. ContentMap selectively materializes and visualizes the logical consequences caused by the axioms under investigation and supports the revision of those consequences. ContentMap requires an exponential number of reasoning operations in the size of the ontology under revision since dependencies between the consequences are determined by comparing their *justifications* (sets of axioms causing the entailment aka minAs). Our approach, however, requires at most a polynomial number of entailment checks.

Another strand of work starting from [7] 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), 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. 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 interactive 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, choosing 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 computing and updating structures called *decision spaces* which saved 75% of reasoner calls during our evaluation.

In our future work, we will 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 sets that can be reviewed independently.

Acknowledgments This work is supported by the German Federal Ministry of Education and Research (BMBF) under the SAW-project NanOn, by the German Research Foundation (DFG) under the project ExpresST, and by the EPSRC project HermiT: Reasoning with Large Ontologies.

References

1. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Llavori, R.B.: Ontology integration using mappings: Towards getting the right logical consequences. In: Proc. ESWC 2009. pp. 173–187 (2009)
2. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Supporting manual mapping revision using logical reasoning. In: Proc. AAAI 2008. pp. 1213–1218 (2008)
3. Nikitina, N.: Semi-automatic revision of formalized knowledge. In: Proc. ECAI 2010. pp. 1097–1098 (2010)
4. Nikitina, N., Rudolph, S., Glimm, B.: Reasoning-supported interactive revision of knowledge bases. Technical Report 3013, Institut AIFB, KIT, Karlsruhe (April 2011)
5. Nikitina, N., Rudolph, S., Glimm, B.: Reasoning-supported interactive revision of knowledge bases. In: Proc. IJCAI 2011 (2011), to Appear
6. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009), available at `http://www.w3.org/TR/owl2-overview/`
7. Rudolph, S.: Exploring relational structures via FLE. In: Proc. ICCS 2004. pp. 196–212. Springer (2004)
8. Shearer, R., Horrocks, I.: Exploiting partial information in taxonomy construction. In: Proc. ISWC 2009. pp. 569–584 (2009)