Model-based Revision Operators for Terminologies in Description Logics

Guilin Qi

Institute AIFB
University of Karlsruhe
Karlsruhe, 76128, Germany

Jianfeng Du

Institute of Software, Chinese Academy of Sciences &
Graduate University of the Chinese Academy of Sciences
Beijing, 100190, China

Abstract

The problem of revising an ontology consistently is closely related to the problem of belief revision which has been widely discussed in the literature. Some *syntax-based belief revision operators* have been adapted to revise ontologies in Description Logics (DLs). However, these operators remove the whole axioms to resolve logical contradictions and thus are not fine-grained. In this paper, we propose three model-based revision operators to revise *terminologies* in DLs. We show that one of them is more rational than others by comparing their logical properties. Therefore, we focus on this revision operator. We also consider the problem of computing the result of revision by our operator with the help of the notion of *concept forgetting*. Finally, we analyze the computational complexity of our revision operator.

1 Introduction

Next generation semantic applications are characterized by a large number of ontologies, some of them constantly evolving. When changing ontologies, we often confront the problem of dealing with inconsistencies [Haase and Stojanovic, 2005; Schlobach *et al.*, 2007; Qi *et al.*, 2008]. In [Qi *et al.*, 2008], two scenarios have been considered where we need to deal with this problem. The first one is the ontology learning scenario where inconsistencies occur during the process of learning expressive ontologies from text corpus incrementally and the second one is the ontology mapping scenario where erroneous mappings may result in an inconsistency and need to be repaired. Inconsistencies hamper the effective use of ontologies because answers derived with standard-reasoning are completely meaningless. Therefore, inconsistency handling is an important problem for ontology change.

Inconsistency handling has been considered as a central problem in many topics of knowledge representation, such as belief revision [Gärdenfors, 1988] which deals with the problem of accommodating newly received information consistently. The problem of revising an ontology consistently is closely related to the problem of belief revision. Many operators in belief revision have been adapted to deal with inconsistencies in description logic-based ontologies where de-

scription logics (DLs) are a family of important ontology representation languages (see [Halaschek-Wiener *et al.*, 2006; Qi *et al.*, 2006; Flouris *et al.*, 2006]). However, these operators are all syntax-dependent ones. That is, suppose there are two ontologies that are logically equivalent and we want to revise them with another ontology using these operators, then we may get two ontologies which are not logically equivalent. In belief revision, there is an important family of revision operators, called model-based revision operators, that are independent of the syntactical forms of the ontologies to be revised. A model-based revision operator in propositional logic is usually defined by the *symmetric difference set* between two interpretations which are a set of propositional variables. However, it is not trivial to adapt the model-based revision operators to DLs because DLs have their own features (see [Flouris *et al.*, 2005]). For example, in DLs, we can distinguish two kinds of logical contradictions: inconsistency and incoherence. An ontology is inconsistent iff it has no model, i.e., it is inconsistent in the first-order sense. An ontology is incoherent iff there exists some unsatisfiable concept (i.e, an unsatisfiable concept stands for the empty set).

In this paper, we propose three novel model-based revision operators to revise *terminologies* in DLs by adapting the well-known Dalal revision operator defined in propositional logic. In Section 3, we first define a revision operator by directly adapting the notions of *difference set between two interpretations* and *distance between two terminologies*. However, this revision operator cannot deal with incoherence. To solve the problem, we propose a modified distance between two terminologies. The revision operator defined by this new distance function is still problematic because it does not differentiate satisfiable concepts and unsatisfiable concepts. Therefore, we define a modified difference set between two interpretations. We also consider the problem of computing the result of revision by this operator with the help of the notion of *forgetting*. In Section 4, we show that this revision operator is better than others by comparing their logical properties. Finally, we consider some computational issues of this revision operator.

2 Preliminaries on Description Logics

We introduce some basic notions of Description Logics (DLs) (more detailed of DLs can be found in [Baader *et al.*, 2007]). In our work, we consider only terminological part of a DL-based ontology, i.e., the so-called *TBox* (or terminology)

which is used to express the intensional level of the ontology. A TBox $\mathcal{T}$ consists of concept axioms and role axioms (RBox). Concept axioms (or terminology axioms) have the form $C \sqsubseteq D$ where C and D are (possibly complex) concept descriptions, and role axioms are expressions of the form $R \sqsubseteq S$, where R and S are (possibly complex) role descriptions. The signature $\text{Sig}(\mathcal{T})$ of TBox $\mathcal{T}$ is the set of concept and role names occur in $\mathcal{T}$.

The semantics of DLs is defined via a model-theoretic semantics, which explicates the relationship between the language syntax and the model of a domain: An interpretation $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain set $\triangle^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$, which maps from concepts and roles to subsets of the domain and binary relations on the domain, respectively. Given an interpretation $\mathcal{I}$, we say that $\mathcal{I}$ satisfies a concept axiom $C \sqsubseteq D$ (respectively, a role inclusion axiom $R \sqsubseteq S$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (respectively, $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$). An interpretation $\mathcal{I}$ is called a *model* of a TBox $\mathcal{T}$, written $\mathcal{I} \models \mathcal{T}$, iff it satisfies each axiom in $\mathcal{T}$. We use $Mod(\mathcal{T})$ to denote all the models of a TBox $\mathcal{T}$. Two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are equivalent, written $\mathcal{T}_1 \equiv \mathcal{T}_2$, iff $Mod(\mathcal{T}_1) = Mod(\mathcal{T}_2)$. A named concept C in a terminology $\mathcal{T}$ is unsatisfiable iff, for each model $\mathcal{I}$ of $\mathcal{T}$, $C^{\mathcal{I}} = \emptyset$. A terminology $\mathcal{T}$ is inconsistent iff it does not have a model, and it is incoherent iff there exists an unsatisfiable named concept in $\mathcal{T}$. Incoherence is a kind of logical contradiction which has been widely discussed (see [Flouris *et al.*, 2006]). When there is a concept in a TBox, if the TBox is inconsistent, then it must be incoherent.

3 Model-based Revision Operators for Terminologies

In this paper, we assume that there is at least one concept in any TBox and each individual TBox is coherent (so it is consistent by the first assumption). Our revision operators are adapted from Dalal's operator [Dalal, 1988], which is an important model-based operator defined as follows: given two propositional formulas ϕ and ψ, we first calculate the distance between them as the minimal cardinality of the *difference sets* between models of ϕ and models of ψ, then the set of models of the result of revising ϕ by ψ consists of models of ψ that satisfies the following condition: there exists a model of ϕ such that the cardinality of the difference set between the two models is the same as the distance between ϕ and ψ. The difference set between two models consists of propositional variables that are interpreted differently by them.

3.1 Definitions

To adapt Dalal's revision operator to DLs, we need to define the "difference set" between two models. By treating each concept name as a propositional variable, we can define the difference between two models in DLs in a similar way as the difference set between two models in propositional logic. Suppose we want to revise a TBox $\mathcal{T}_1$ using another one $\mathcal{T}_2$. Following the idea of Dalal's revision operator, in our revision operator, we revise some models of $\mathcal{T}_1$ to make them as models of $\mathcal{T}_2$. However, we introduce a special treatment. Since logical errors in TBoxes are usually caused by incorrect concept definitions (see [Rector *et al.*, 2004]), to revise a

model $\mathcal{I}$ of $\mathcal{T}_1$, we only revise $A^{\mathcal{I}}$ for some concept names A, and keep $R^{\mathcal{I}}$ for all role names R intact. We will show that revising $A^{\mathcal{I}}$ for some concept names A is enough to turn a model of $\mathcal{T}_1$ into a model of $\mathcal{T}_2$ (Section 4). In order to ensure that $R^{\mathcal{I}}$ is not revised for role names R, the distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ should not be related to role names. This can be achieved by the following definition of a difference set.

Definition 1. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes, and CN and RN be respectively sets of concept names and role names in* $\text{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. *Let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ and $\mathcal{I}' = (\Delta, \cdot^{\mathcal{I}'})$ be models of $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively, which are defined on* $\text{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. *The* difference set *between $\mathcal{I}$ and $\mathcal{I}'$, written $diff(\mathcal{I}, \mathcal{I}')$, is defined as follows:*

$$diff(\mathcal{I}, \mathcal{I}') = \begin{cases} CN, & \text{if there exists } R \in RN, R^{\mathcal{I}} \neq R^{\mathcal{I}'}, \\ \{A \in CN | A^{\mathcal{I}} \neq A^{\mathcal{I}'}\}, & \text{otherwise.} \end{cases}$$

That is, the difference set between two models is the set of all concept names if they interpret a role name differently and the set of concept names that are interpreted differently by the models otherwise. Note that if $\mathcal{I}_1$ and $\mathcal{I}_2$ do not share a common domain, then they are not comparable.

We can define the distance between two TBoxes. Throughout the paper, we use $|S|$ to denote the cardinality of set S.

Definition 2. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes. The* distance *between $\mathcal{T}_1$ and $\mathcal{T}_2$, written $d(\mathcal{T}_1, \mathcal{T}_2)$, is defined as:*

$$d(\mathcal{T}_1, \mathcal{T}_2) = min_{\mathcal{I} \models \mathcal{T}_1, \mathcal{I}' \models \mathcal{T}_2} |diff(\mathcal{I}, \mathcal{I}')|.$$

The distance between two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ is the minimal cardinality of the difference sets between models of $\mathcal{T}_1$ and models of $\mathcal{T}_2$.

We define our first revision operator.

Definition 3. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes. A revision operator, written $\circ_M$, is defined in a model-theoretical way as follows:*

$$Mod(\mathcal{T}_1 \circ_M \mathcal{T}_2) = \{\mathcal{I} \models \mathcal{T}_2 \mid \exists \mathcal{I}' \models \mathcal{T}_1, \\ |diff(\mathcal{I}, \mathcal{I}')| = d(\mathcal{T}_1, \mathcal{T}_2)\}.$$

That is, the models of the result of our revision operator $\circ_M$ are the models of TBox $\mathcal{T}_2$ satisfying the condition that there exists a model of $\mathcal{T}_1$ such that the difference between them is equal to the distance between the two TBoxes.

It is easy to see that if $\mathcal{T}_1$ is consistent with $\mathcal{T}_2$, then $\mathcal{T}_1 \circ_M \mathcal{T}_2 \equiv \mathcal{T}_1 \cup \mathcal{T}_2$. From this property, we have that $\mathcal{T}_1 \circ_M \mathcal{T}_2$ is incoherent if $\mathcal{T}_1 \cup \mathcal{T}_2$ is incoherent but consistent. It has been pointed out incoherence will cause trivial subsumption relation on unsatisfiable concepts so that it should be resolved after revision (see [Schlobach *et al.*, 2007] and [Flouris *et al.*, 2006]). The problem of the revision operator is caused by the distance between two TBoxes. That is, when $\mathcal{T}_1 \cup \mathcal{T}_2$ is incoherent but consistent, their distance is 0. So any interpretation that is a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ is a model of $\mathcal{T}_1 \circ_M \mathcal{T}_2$. To solve this problem, we need to give a special treatment of unsatisfiable concepts in $\mathcal{T}_1 \cup \mathcal{T}_2$.

Definition 4. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes and CN be the set of concept names in $\text{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. The* modified distance *between $\mathcal{T}_1$ and $\mathcal{T}_2$, written $d'(\mathcal{T}_1, \mathcal{T}_2)$, is defined as:*

$$d'(\mathcal{T}_1, \mathcal{T}_2) = min_{\mathcal{I} \models \mathcal{T}_1, \mathcal{I}' \models \mathcal{T}_2, \forall A \in CN : A^{\mathcal{I}'} \neq \emptyset} |diff(\mathcal{I}, \mathcal{I}')|.$$

When model $\mathcal{I}'$ of $\mathcal{T}_2$ is used to define the new distance function, it cannot interpret any concept as an empty set. This will exclude the case that there exists model $\mathcal{I}$ of $\mathcal{T}_2$ such that both $\mathcal{I}$ and $\mathcal{I}'$ interpret an unsatisfiable concept as an empty set. Note that the existence of model $\mathcal{I}'$ of $\mathcal{T}_2$ that does not interpret any concept as an empty set is justified by the assumption that $\mathcal{T}_2$ is coherent. By replacing d in Definition 3 with d' we can get a new revision operator, which is denoted as $\circ_{M'}$. The following example shows that we do not have $\mathcal{T}_1 \circ_M \mathcal{T}_2 \equiv \mathcal{T}_1 \cup \mathcal{T}_2$ anymore even if $\mathcal{T}_1 \cup \mathcal{T}_2$ is incoherent but consistent.

Example 1. *Let $\mathcal{T}_1 = \{A \sqsubseteq B \sqcap D,\ A \sqsubseteq C, B \sqsubseteq C\}$ and $\mathcal{T}_2 = \{A \sqsubseteq \neg B, A \sqsubseteq \neg C\}$. A is an unsatisfiable concept in $\mathcal{T}_1 \cup \mathcal{T}_2$. Consider a model $\mathcal{I}$ of $\mathcal{T}_2$ and a model $\mathcal{I}'$ of $\mathcal{T}_1$ such that $\Delta^{\mathcal{I}'} = \{a,b,c,d\}$, $A^{\mathcal{I}'} = \{a\}$, $B^{\mathcal{I}'} = \{a,b,c\}$, $C^{\mathcal{I}'} = \{a,b,c\}$ and $D^{\mathcal{I}'} = \{a,c\}$, and $\Delta^{\mathcal{I}} = \{a,b,c,d\}$, $A^{\mathcal{I}} = \{d\}$, $B^{\mathcal{I}} = \{a,b,c\}$, $C^{\mathcal{I}} = \{a,b,c\}$ and $D^{\mathcal{I}} = \{a,c\}$. We have $diff_S(\mathcal{I},\mathcal{I}') = \{A\}$. Therefore $|diff(\mathcal{I},\mathcal{I}')| = 1$ and $d'(\mathcal{T}_1,\mathcal{T}_2) = 1$. We show that $Mod(\mathcal{T}_1 \circ_{M'} \mathcal{T}_2) = Mod(\mathcal{T}_2 \cup \{B \sqsubseteq C\})$. Suppose $\mathcal{I} \models \mathcal{T}_1 \circ_{M'} \mathcal{T}_2$, then $\mathcal{I} \models \mathcal{T}_2$ and there exists a model $\mathcal{I}'$ of $\mathcal{T}_1$ such that $|diff(\mathcal{I},\mathcal{I}')| = 1$. It is not difficult to see that we must have $diff(\mathcal{I},\mathcal{I}') = \{A\}$. Therefore, $B^{\mathcal{I}} = B^{\mathcal{I}'}$ and $C^{\mathcal{I}} = C^{\mathcal{I}'}$. Therefore, $\mathcal{I} \models B \sqsubseteq C$. Conversely, suppose $\mathcal{I} \models \mathcal{T}_2 \cup \{B \sqsubseteq C\}$. Then $\mathcal{I} \models \mathcal{T}_2$. Suppose $\mathcal{I}'$ is an interpretation such that $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$, $B^{\mathcal{I}'} = B^{\mathcal{I}}$ and $C^{\mathcal{I}} = C^{\mathcal{I}'}$, $A^{\mathcal{I}'} = \emptyset$ and $D^{\mathcal{I}'} = D^{\mathcal{I}}$. We have $diff(\mathcal{I},\mathcal{I}') = \{A\}$. Therefore, $\mathcal{I}$ is a model of $\mathcal{T}_1 \circ_{M'} \mathcal{T}_2$.*

From Example 1, to resolve an unsatisfiability, the revision operator $\circ_{M'}$ may remove all the unsatisfiable concepts in the original TBox. This is not the standard way to deal with unsatisfiability because it does not debug the ontology to find the cause of the contradiction (see [Schlobach *et al.*, 2007]). In Example 1, one of the reasons that concept A becomes unsatisfiable is that it is claimed to be a subconcept of both B and $\neg B$. To resolve an unsatisfiability, we should change those concepts or axioms that are involved in the contradiction and keep the unsatisfiable concept if possible.

The problem of the revision operator $\circ_{M'}$ is caused by the definition of difference set. That is, it gives the same priority to the satisfiable concepts and the unsatisfiable concepts in a difference set between two interpretations. To solve this problem, we introduce a notion of *stratified set* to define a new difference set. Given n sets of concept names $S_1, ..., S_n$, by $S = (S_1, ..., S_n)$ we denote a stratified set such that elements in S_i $(i = 1, ..., n)$ have the same priority but elements in S_j are preferred to those in S_k for any $j < k$. The cardinality of the stratified set $S = (S_1, ..., S_n)$ is defined as an ordered set of numbers $|S| = (|S_1|, ..., |S_n|)$. Let $S = (S_1, ..., S_n)$ and $S' = (S_1', ..., S_n')$, $|S| < |S'|$ iff there exists an $i \in \{1, ..., n\}$, such that $|S_i| < |S_i'|$ and $|S_j| = |S_j'|$ for any $j < i$, and $|S| = |S'|$ iff $|S_i| = |S_i'|$ for any $i \in \{1, ..., n\}$. We now define a stratified difference set between two interpretations. We assume that the set CN of concept names in signature $\mathrm{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$ has been stratified such that some concept names are more important than others. When defining the stratified difference set between interpretations $\mathcal{I}$ and $\mathcal{I}'$, we first give higher priority to con-

cept names in $diff(\mathcal{I},\mathcal{I}')$ that are unsatisfiable in $\mathcal{T}_1 \cup \mathcal{T}_2$ than those that are satisfiable in $\mathcal{T}_1 \cup \mathcal{T}_2$, then further stratify these two sets of concept names using the stratified set on CN. Formally, we have the following definition.

Definition 5. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes. Suppose $(S_1, ..., S_n)$ is a stratified set on all concept names in $\mathrm{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. Let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ and $\mathcal{I}' = (\Delta, \cdot^{\mathcal{I}'})$ be two interpretations. Let $U = \{A \in diff(\mathcal{I},\mathcal{I}') \mid A$ is unsatisfiable in $\mathcal{T}_1 \cup \mathcal{T}_2\}$ and $W = diff(\mathcal{I},\mathcal{I}') \setminus U$. The stratified difference set between $\mathcal{I}$ and $\mathcal{I}'$, written $diff_S(\mathcal{I},\mathcal{I}')$, is defined as follows: $diff_S(\mathcal{I},\mathcal{I}') = (U \cap S_1, ..., U \cap S_n, W \cap S_1, ..., W \cap S_n)$*

Similarly, we can define $d_S(\mathcal{T}_1,\mathcal{T}_2)$ through replacing $diff$ by $diff_S$ in Definition 4 and then define a new revision operator through replacing $diff$ by $diff_S$ and replacing d by d_S in Definition 3, which is denoted as $\circ_S$. We need some further explanations of the stratified set on concept names in $\mathrm{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$ in Definition 5. There are some benefits of defining the stratified difference set by this stratified set. First, based on different stratified sets, we can define different difference sets, thus different revision operators. We allow users to order the signature of the language and decide which concept names are more important than others. Therefore, our revision operator is more flexible than those defined previously. Second, in Section 5, we will provide a special stratified set on concept names in $\mathrm{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that the result of the revision operator defined by the stratified difference set can be computed in polynomial time for a lightweight DL. A naive stratified set is to give all concept names in $\mathrm{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$ the same priority. In this case, $n = 1$ in Definition 5 and $diff_S(\mathcal{I},\mathcal{I}') = (U, S_1 \setminus U)$. We illustrate the new revision operator using Example 1, by taking this naive stratified set.

Example 2. *Consider model $\mathcal{I}$ of $\mathcal{T}_2$ and model $\mathcal{I}'$ of $\mathcal{T}_1$ such that $\Delta^{\mathcal{I}'} = \{a,b,c,d\}$, $A^{\mathcal{I}'} = \{a\}$, $B^{\mathcal{I}'} = \{a,b,c\}$, $C^{\mathcal{I}'} = \{a,b,c\}$ and $D^{\mathcal{I}'} = \{a,c\}$, and $\Delta^{\mathcal{I}} = \{a,b,c,d\}$, $A^{\mathcal{I}} = \{a\}$, $B^{\mathcal{I}} = \{b,c\}$, $C^{\mathcal{I}} = \{b,c\}$ and $D^{\mathcal{I}} = \{a,c\}$. We have $diff_S(\mathcal{I},\mathcal{I}') = (\emptyset, \{B,C\})$. So $|diff_S(\mathcal{I},\mathcal{I}')| = (0,2)$. We must have $d(\mathcal{T}_1,\mathcal{T}_2) = (0,2)$ based on the following observations: (1) for any pair of interpretations $\mathcal{I}_1$ and $\mathcal{I}_1'$, we must have $A^{\mathcal{I}_1} = A^{\mathcal{I}_1'} \neq \emptyset$, (2) for any pair of interpretations $\mathcal{I}_1$ and $\mathcal{I}_1'$, if $|diff_S(\mathcal{I}_1,\mathcal{I}_1')| = (0,m)$, where m is an integer, then we must have $m \geq 2$. Therefore, a model $\mathcal{I}$ of $\mathcal{T}_2$ is a model of $\mathcal{T}_1 \circ_S \mathcal{T}_2$ iff there exists a model $\mathcal{I}'$ of $\mathcal{T}_1$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ and $|diff_S(\mathcal{I},\mathcal{I}')| = (0,2)$.*

3.2 Syntactical counterpart of revision operator $\circ_S$

We show that the result of our revision operator $\circ_S$ can be computed with the help of the notion of *concept forgetting*.

It has been shown in [Lang and Marquis, 2002] that Dalal's revision operator is a special case of a general framework based on *variable forgetting*, where a propositional variable forgotten in a propositional formula will result in another formula which is logically strongest consequence of the original formula that is independent of the variable. This inspires us to provide an approach for computing our revision operator $\circ_S$ syntactically by using the notion of *forgetting* in DLs.

We first introduce the relation $\sim_A$ and the notion of concept forgetting given in [Wang *et al.*, 2008]: let A be a concept name in a DL language $\mathcal{L}$, and $\mathcal{I}$ and $\mathcal{I}'$ interpretations of $\mathcal{L}$. We define $\mathcal{I} \sim_A \mathcal{I}'$ iff $\mathcal{I}$ and $\mathcal{I}'$ agree on all concept names and role names except possibly on A. The result of forgetting about A in $\mathcal{T}$, denoted as $\mathtt{forget}(\mathcal{T}, A)$, is defined in a model-theoretical way as follows: $\mathtt{forget}(\mathcal{T}, A)$ is a TBox on the signature $\mathtt{Sig}(\mathcal{T}) \setminus \{A\}$ and any interpretation $\mathcal{I}'$ is a model of $\mathtt{forget}(\mathcal{T}, A)$ iff there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $\mathcal{I} \sim_A \mathcal{I}'$. It has been shown in [Wang *et al.*, 2008] that when we forget a set $\mathbf{A} = \{A_1, ..., A_n\}$ of concept names, the order of concept forgetting will not influence the final result of forgetting. Therefore, we can define $\mathtt{forget}(\mathcal{T}, \mathbf{A}) = \mathtt{forget}(...(\mathtt{forget}(\mathcal{T}, A_1), ...), A_n)$. In [Wang *et al.*, 2008], an algorithm is given to compute the result of concept forgetting in *DL-Lite*, a family of DLs that provide tractable reasoning. However, for more expressive DLs, the result of concept forgetting may not be expressed in the same language [Konev *et al.*, 2008]. In the following, we define a revision operator by using the notion of forgetting and show that this operator corresponds to the revision operator $\circ_S$.

We define the notion of recovery set, which is a set of concept names in the original TBox that will be forgotten to restore coherence.

Definition 6. *Given two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, a recovery set of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$ is a set V of concept names in $\mathcal{T}_1 \cup \mathcal{T}_2$ such that $\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$ is coherent.*

We use $\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$ to denote all the recovery sets of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$. A trivial recovery set is the set CN of all concept names in $\mathcal{T}_1 \cup \mathcal{T}_2$ as $\mathtt{forget}(\mathcal{T}_1, CN) = \emptyset$ and $\mathcal{T}_2$ is assumed to be coherent. In Example 1, $\{A\}$ and $\{B, C\}$ are recovery sets of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$. Among all the recovery sets of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$, some are preferred to others. In the following, we define a preference relation on $\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$. We first stratify a recovery set by giving priority to unsatisfiable concepts in it and further stratify these unsatisfiable concepts and other concepts by a pre-defined stratified set. We then compare two recovery sets by comparing the cardinality of their stratified sets.

Definition 7. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes. Suppose $(S_1, ..., S_n)$ is a stratified set on all concept names in $\mathtt{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. A preference relation on $\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$, called a lexicographic relation and is denoted as $\preceq_{lex}$, is defined as follows: for any two recovery sets V_1 and V_2, let $U_i = \{A \in V_i \mid A$ is unsatisfiable in $\mathcal{T}_1 \cup \mathcal{T}_2\}$ and $W_i = V_i \setminus U_i$ for $i = 1, 2$. $V_1 \preceq_{lex} V_2$ iff $|(U_1 \cap S_1, ..., U_1 \cap S_n, W_1 \cap S_1, ..., W_1 \cap S_n)| \leq |(U_2 \cap S_1, ..., U_2 \cap S_n, W_2 \cap S_1, ..., W_2 \cap S_n)|$.*

We call a recovery set $V \in \mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$ *preferred* if for all recovery sets $V' \in \mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$, $V \preceq_{lex} V'$. By $Pre(\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2})$ we denote the set of all preferred recovery sets in $\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$.

Definition 8. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes. Suppose $(S_1, ..., S_n)$ is a stratified set on all concept names in $\mathtt{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. A forgetting-based revision operator, denoted as $\circ_f$, is defined as follows:*

$\mathcal{T}_1 \circ_f \mathcal{T}_2 = \{\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2 \mid V \in Pre(\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2})\}$,

where $Pre(\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2})$ is defined by $(S_1, ..., S_n)$.

$\mathcal{T}_1 \circ_f \mathcal{T}_2$ consists of the union of $\mathcal{T}_2$ and the knowledge bases[1] obtained from $\mathcal{T}_1$ by forgetting preferred recovery sets. Therefore, the result of revision is a set of knowledge bases. In [Meyer *et al.*, 2005], a set of knowledge bases is called a disjunctive knowledge base (DKB). Its semantics is given as follows: A DKB $\mathcal{B}$ is satisfied by an interpretation $\mathcal{I}$ iff $\mathcal{I}$ is a model of at least one of elements of $\mathcal{B}$.

Example 3. *(Example 1 continues) Suppose the stratified set on $\{A, B, C\}$ is $\{\{A, B, C\}\}$. There are several recovery sets of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$, such as $\{A\}$, $\{B, C\}$, $\{A, B, C\}$, etc. It is easy to check that $V = \{B, C\}$ is the only preferred recovery set. Therefore, $\mathcal{T}_1 \circ_f \mathcal{T}_2 = \{\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2\} = \{\{A \sqsubseteq D\} \cup \mathcal{T}_2\}$. That is, we get a unique TBox $\mathcal{T} = \{A \sqsubseteq D, A \sqsubseteq \neg B, A \sqsubseteq \neg C\}$ as the result of revision.*

One may notice that our revision operator drops more information than necessary to restore coherence because $B \sqsubseteq C$ can be added to $\mathcal{T}$ without causing a contradiction. This kind of problem can be fixed, for example, by taking another step to restore unnecessary removals by a syntax-based revision after we apply the forgetting-based revision operator. The detailed discussion of this problem will be left as future work.

We are able to show that revision operators $\circ_S$ and $\circ_f$ are semantically equivalent. Due to page limit, proofs of propositions in this paper are either omitted or sketched. Full proofs can be found in a technical report at http://www.aifb.uni-karlsruhe.de/WBS/gqi/papers/IJCAI09QD.pdf.

Proposition 1. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes. Suppose $(S_1, ..., S_n)$ is a stratified set on all concept names in $\mathtt{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. If both $\circ_S$ and $\circ_f$ are defined by $(S_1, ..., S_n)$, then we have $Mod(\mathcal{T}_1 \circ_S \mathcal{T}_2) = Mod(\mathcal{T}_1 \circ_f \mathcal{T}_2)$.*

Proof sketch. We first show that $Mod(\mathcal{T}_1 \circ_S \mathcal{T}_2) \subseteq Mod(\mathcal{T}_1 \circ_f \mathcal{T}_2)$.

Suppose $\mathcal{I} \models \mathcal{T}_1 \circ_S \mathcal{T}_2$, then there exists a model $\mathcal{I}'$ of $\mathcal{T}_2$ such that $diff_S(\mathcal{I}, \mathcal{I}') = d_S(\mathcal{T}_1, \mathcal{T}_2)$, $R^{\mathcal{I}} = R^{\mathcal{I}'}$ for any $R \in RN$, and $A^{\mathcal{I}'} \neq \emptyset$ for any $A \in CN$. We set V as the set of all the concept names in $diff_S(\mathcal{I}, \mathcal{I}')$. We first show that V is a recovery set, that is, $\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$ is coherent by showing that $\mathcal{I}'$ is model of $\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$ and using the fact that $A^{\mathcal{I}'} \neq \emptyset$ for any $A \in CN$. Next, we show that V is a preferred recovery set by reduction to absurdity by using the fact that if $\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$ is coherent, then there exists a model $\mathcal{I}$ of $\mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$ such that $A^{\mathcal{I}} \neq \emptyset$ for any unsatisfiable concept of $\mathcal{T}_1 \cup \mathcal{T}_2$ and there exists a model $\mathcal{I}'$ of $\mathcal{T}_1$ such that $\mathcal{I} \sim_V \mathcal{I}'$.

Second, we show that $Mod(\mathcal{T}_1 \circ_f \mathcal{T}_2) \subseteq Mod(\mathcal{T}_1 \circ_S \mathcal{T}_2)$.

Suppose $\mathcal{I} \models \mathcal{T}_1 \circ_f \mathcal{T}_2$. Then $\mathcal{I} \models \mathtt{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$. Since $\mathcal{I} \models \mathtt{forget}(\mathcal{T}_1, V)$, there exists $\mathcal{I}' \models \mathcal{T}_1$ such that $\mathcal{I} \sim_V \mathcal{I}'$. So $|diff(\mathcal{I}, \mathcal{I}')| = |V|$. We show that $d_S(\mathcal{T}_1, \mathcal{T}_2) = |(U \cap S_1, ..., U \cap S_n, W \cap S_1, ..., W \cap S_n)|$ by reduction to absurdity. Therefore, $|diff_S(\mathcal{I}, \mathcal{I}')| = d_S(\mathcal{T}_1, \mathcal{T}_2)$. Since $\mathcal{I} \models \mathcal{T}_2$ and $\mathcal{I}' \models \mathcal{T}_1$, $\mathcal{I}$ is a model of $\mathcal{T}_1 \circ_S \mathcal{T}_2$. $\qquad\square$

---

[1]The result of forgetting may not be a DL knowledge base anymore.

For a terminological axiom ϕ, $\mathcal{T}_1 \circ_f \mathcal{T}_2 \models \phi$ (resp. $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$) if ϕ is satisfied by every model in $Mod(\mathcal{T}_1 \circ_f \mathcal{T}_2)$ (resp. $Mod(\mathcal{T}_1 \circ_S \mathcal{T}_2)$). Proposition 1 tells that there exists an algorithm to decide if $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$, provided that there exist algorithms for computing the function forget, because $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$ iff $\mathcal{T}_1 \circ_f \mathcal{T}_2 \models \phi$ iff $\text{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2 \models \phi$ for all preferred recovery set $V \in Pre(\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2})$.

4 Logical properties

We consider postulates for revision operators in DLs given in [Qi *et al.*, 2006], which are reformulated from Katsuno and Mendelzon's postulates (KM postulates) in [Katsuno and Mendelzon, 1992].

(G1) $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) \subseteq Mod(\phi)$ for all $\phi \in \mathcal{T}_2$.
(G2) If $Mod(\mathcal{T}_1) \cap Mod(\mathcal{T}_2) \neq \emptyset$, then $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) = Mod(\mathcal{T}_1) \cap Mod(\mathcal{T}_2)$.
(G3) If $\mathcal{T}_2$ is consistent, then $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) \neq \emptyset$.
(G4) If $Mod(\mathcal{T}) = Mod(\mathcal{T}_1)$ and $Mod(\mathcal{T}') = Mod(\mathcal{T}_2)$, then $Mod(\mathcal{T} \circ \mathcal{T}') = Mod(\mathcal{T}_1 \circ \mathcal{T}_2)$.
(G5) $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) \cap Mod(\mathcal{T}_3) \subseteq Mod(\mathcal{T}_1 \circ (\mathcal{T}_2 \cup \mathcal{T}_3))$.
(G6) If $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) \cap Mod(\mathcal{T}_3)$ is not empty, then $Mod(\mathcal{T}_1 \circ (\mathcal{T}_2 \cup \mathcal{T}_3)) \subseteq Mod(\mathcal{T}_1 \circ \mathcal{T}_2) \cap Mod(\mathcal{T}_3)$.

(G1) guarantees that every axiom in the new TBox can be inferred from the result of revision. (G2) says that we do not change the original knowledge base if there is no conflict. (G3) is a condition preventing a revision from introducing unwarranted inconsistency. (G4) says the revision operator should be independent of the syntactical forms of knowledge bases. (G5) and (G6) together are used to ensure minimal change.

We are able to show that the operator $\circ_M$ satisfies all these postulates.

Proposition 2. *$\circ_M$ satisfies (G1)-(G6).*

Postulates (G1)-(G6) are reasonable when we only want to deal with inconsistency. However, (G2) is not a good postulate to capture a rational revision operator for terminologies because it infers that if a TBox is consistent with another one but their union is incoherent, then the result of revision is still incoherent. This is exactly the problem for the revision operator $\circ_M$ given by Definition 3. Therefore, we modify (G2) as follows:

(G2') For any TBox $\mathcal{T}_1$ and any coherent TBox $\mathcal{T}_2$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is coherent if and only if $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) = Mod(\mathcal{T}_1) \cap Mod(\mathcal{T}_2)$.

Postulate (G2') requires that if $\mathcal{T}_1 \cup \mathcal{T}_2$ is incoherent but $\mathcal{T}_2$ is coherent, then the result of revision should not be equivalent to their union. It is clear that $\circ_M$ does not satisfy this postulate.

We are able to show that our revision operators $\circ_{M'}$ and $\circ_S$ satisfy (G2') and many other postulates.

Proposition 3. *For any stratified set on all concept names in $\text{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$, $\circ_{M'}$ satisfies (G1),(G2') and (G3)-(G6). $\circ_S$ also satisfies (G1),(G2') and (G3)-(G6). But they do not satisfy (G2) in general.*

Proposition 3 shows that postulates (G1),(G2') and (G3)-(G6) do not differentiate operator $\circ_S$ and operator $\circ_{M'}$. Therefore, we propose a new postulate which is satisfied by $\circ_S$ but falsified by $\circ_{M'}$.

Unsatisfiability Repair Given two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, suppose there exists a set V of satisfiable concepts in $\mathcal{T}_1 \cup \mathcal{T}_2$ such that all the unsatisfiable concepts in $\mathcal{T}_1 \cup \mathcal{T}_2$ are satisfiable in $\text{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$. Then for any model $\mathcal{I} \in Mod(\mathcal{T}_1 \circ \mathcal{T}_2)$, there exists a model $\mathcal{I}' \in Mod(\mathcal{T}_1)$ such that $A^{\mathcal{I}} = A^{\mathcal{I}'}$ for all unsatisfiable concept A in $\mathcal{T}_1 \cup \mathcal{T}_2$.

This postulate says if we can resolve all unsatisfiable concepts by forgetting some satisfiable concepts, then none of the unsatisfiable concepts should be forgotten after revision. Example 1 shows that $\circ_{M'}$ does not satisfy this postulate.

Proposition 4. *For any stratified set on all concept names in $\text{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$, $\circ_S$ satisfies the postulate "Unsatisfiability Repair".*

Propositions 1-4 show that $\circ_S$ is better than other two when applied to deal with incoherence.

5 Computational Issues of Revision Operator $\circ_S$ in DL-Lite$_\sqcap$

We first analyze the computational complexity of our revision operator in a special DL, called DL-Lite$_\sqcap$. The language of DL-Lite$_\sqcap$ extends the core language for the DL-Lite family [Calvanese *et al.*, 2007] by allowing conjunctions of basic concepts in the left-hand side of inclusion axioms, where there exists an algorithm to compute the result of concept forgetting [Wang *et al.*, 2008]. The DLs of the DL-Lite family are tailed to capture conceptual modeling constructs, but still have low reasoning overheads. All reasoning tasks in the DLs of this family, such as concept subsumption and answering complex queries, are computationally tractable.

We show the time complexity of subsumption checking under our revision operator.

Proposition 5. *Given two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ in DL-Lite$_\sqcap$ and a concept subsumption axiom ϕ in DL-Lite$_\sqcap$, the problem of deciding if $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$ is Δ_2^p-complete.*

Proof sketch. (Hardness) We show this by a $\leq_m^p$-reduction of the following Δ_2^p-complete problem [Krentel, 1988]: Given a satisfiable clause set $C = \{C_l, ..., C_n\}$ on $X = \{x_1, ..., x_n\}$, decide whether the lexicographically maximum truth assignment $\Phi(X)$ on $(x_1, ..., x_n)$, which we denote by $\Phi_m(X)$, fulfills $\Phi_m(x_n) = \text{true}$. For $1 \leq j \leq m$, let $L(C_j)$ denote the set of literals in C_j. For l a literal occurring in C, let B_l denote the concept name A_i if l is x_i, or A_i' if l is $\neg x_i$.

Let $\mathcal{T}_1 = \{G_i \sqsubseteq A_i, G_i \sqsubseteq A_i', G_{i+n} \sqsubseteq A_i, G_{i+n} \sqsubseteq A_i'' \mid 1 \leq i \leq n\} \cup \{H_j \sqsubseteq B_l \mid 1 \leq j \leq m, l \in L(C_j)\}$ and $\mathcal{T}_2 = \{A_i \sqcap A_i' \sqsubseteq \neg G_i, A_i \sqcap A_i'' \sqsubseteq \neg G_{i+n} \mid 1 \leq i \leq n\} \cup \{\bigsqcap_{l \in L(C_j)} B_l \sqsubseteq \neg H_j \mid 1 \leq j \leq m\}$. We define a stratified set for $\circ_S$ as $S = (\{G_i \mid 1 \leq i \leq 2n\} \cup \{H_j \mid 1 \leq j \leq m\}, \{A_i, A_i' \mid 1 \leq i \leq n\}, A_1'', ..., A_n'')$. For a truth assignment $\Phi(X)$, let V_Φ denote $\{A_i \mid 1 \leq i \leq n, \Phi(x_i) = \text{true}\} \cup \{A_i', A_i'' \mid 1 \leq i \leq n, \Phi(x_i) = \text{false}\}$. Then V_Φ is a recovery set of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$ iff $\Phi(X)$ satisfies C. Note that for a preferred recovery set V of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$, since V does not contain any G_i or H_j, it must contain A_i or A_i', A_i'' for all $1 \leq i \leq n$; otherwise some G_i is unsatisfiable in $\text{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2$. Since S prefers recovery sets that do not contain A_1'' (and hence contain A_1) over recovery sets that

contain A_1'' etc., it is clear that for distinct truth assignments $\Phi(X)$ and $\Psi(X)$ satisfying C, we have $V_\Phi \preceq_{lex} V_\Psi$ iff $\Phi(X)$ is not lexicographically less than $\Psi(X)$ w.r.t. $(x_1, ..., x_n)$. It follows that $Pre(\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}) = \{V_{\Phi_m}\}$. Hence $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models G_n \sqsubseteq A_n'$ iff $\mathcal{T}_1 \circ_f \mathcal{T}_2 \models G_n \sqsubseteq A_n'$ (by Proposition 1) iff $A_n' \notin V_{\Phi_m}$ iff $\Phi_m(x_n) = \text{true}$. Since $\mathcal{T}_1, \mathcal{T}_2$ and S can be constructed in polynomial time, the hardness holds.

(Membership) Let the stratified set for $\circ_S$ be $(S_1, ..., S_n)$. We use the following algorithm to check if $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$.

1. $U := \{A \in \bigcup_{i=1}^n S_i \mid A \text{ is unsatisfiable in } \mathcal{T}_1 \cup \mathcal{T}_2\}$;
2. **for** $i := 1, ..., n$ **do** $T_i = U \cap S_i$; $T_{i+n} = S_i \setminus U$;
3. **for** $i := 1, ..., 2n$ **do**
4. **for** $j := 0, ..., |T_i|$ **do**
5.   Guess $(V_1, ..., V_i)$ where $V_k \subseteq T_k$ for all $1 \leq k \leq i$, and check if $|V_k| = m_k$ for $1 \leq k \leq i-1$, $|V_i| = j$ and $\bigcup_{k=1}^i V_k \cup \bigcup_{k=i+1}^{2n} T_k$ is a recovery set of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$; if so, set m_i as j and continue next i;
6. Guess a stratified set $(W_1, ..., W_n)$ where $W_i \subseteq S_i$ for all $1 \leq i \leq n$, and check if $\bigcup_{i=1}^n W_i$ is a recovery set of $\mathcal{T}_1$ w.r.t. $\mathcal{T}_2$, $\text{forget}(\mathcal{T}_1, \bigcup_{i=1}^n W_i) \cup \mathcal{T}_2 \not\models \phi$, and for all $1 \leq i \leq n$, $|W_i \cap U| = m_i$ and $|W_i \setminus U| = m_{i+n}$; if so, **return** false;
7. **return** true;

In the above algorithm, a preferred recovery set V is computed in lines 1–5 by a guess-and-test approach, and m_i is set as $|V \cap T_i|$ for all $1 \leq i \leq 2n$. Clearly $(m_1, ..., m_{2n})$ is unique for all preferred recovery sets. Note that U can be computed in polynomial time. Note also that the checking process in line 5 or line 6 can be accomplished in polynomial time, so the algorithm finishes after calling an NP oracle $O(1 + \sum_{i=1}^{2n}(|T_i| + 1)) = O(\sum_{i=1}^n |S_i|)$ times. That is, the algorithm works in Δ_2^p time. $\square$

The above proposition shows that subsumption checking under $\circ_S$ is in general intractable. However, in a special case where every concept name constitutes a stratum in the given stratified set for $\circ_S$, subsumption checking in DL-Lite$_\sqcap$ under $\circ_S$ is tractable (see Algorithm 1).

Algorithm 1. $\text{Checking}(\mathcal{T}_1, \mathcal{T}_2, (S_1, ..., S_n), \phi)$

Input: Two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, a stratified set $(S_1, ..., S_n)$ on all concept names in $\text{Sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$ such that $|S_i| = 1$ for all $1 \leq i \leq n$, and a concept subsumption axiom ϕ.

Output: The truth value of $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$, where $\circ_S$ is defined by $(S_1, ..., S_n)$.

1. $U := \{A \in \bigcup_{i=1}^n S_i \mid A \text{ is unsatisfiable in } \mathcal{T}_1 \cup \mathcal{T}_2\}$; $V := \emptyset$;
2. **for** $i := 1, ..., n$ **do** $T_i := S_i \cap U$; $T_{i+n} := S_i \setminus U$;
3. **for** $i := 1, ..., 2n$ with $T_i \neq \emptyset$ **do**
4. **if** $\text{forget}(\mathcal{T}_1, V \cup \bigcup_{k=i+1}^{2n} T_k) \cup \mathcal{T}_2$ is incoherent **then**
5. $V := V \cup T_i$;
6. **return** $\text{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2 \models \phi$;

Algorithm 1 first computes the unique preferred recovery set V in $\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$ (lines 1–5), then checks whether $\text{forget}(\mathcal{T}_1, V) \cup \mathcal{T}_2 \models \phi$ (line 6). By Proposition 1, the result of the above checking is exactly the truth value of $\mathcal{T}_1 \circ_S \mathcal{T}_2 \models \phi$. In the algorithm, T_i consists of at most one axiom for any $1 \leq i \leq 2n$. Since a preferred recovery set V should make $|(T_1 \cap V, ..., T_{2n} \cap V)|$ minimal among that of all recovery sets, $\mathcal{R}_{\mathcal{T}_1}^{\mathcal{T}_2}$ has only one preferred recovery set. It is computed by considering $T_1, ..., T_{2n}$ in turn to keep as many axioms with higher priority as possible while guaranteeing coherence. Let V be a partial preferred recovery set before considering T_i. Then T_i is appended to V if $\text{forget}(\mathcal{T}_1, V \cup \bigcup_{k=i+1}^{2n} T_k) \cup \mathcal{T}_2$ is incoherent, because otherwise coherence cannot be restored by considering $T_{i+1}, ..., T_{2n}$. Since the result of concept forgetting in DL-Lite$_\sqcap$ TBoxes can be computed in polynomial time [Wang *et al.*, 2008], as well as coherence checking and subsumption checking in DL-Lite$_\sqcap$ can be accomplished in polynomial time [Calvanese *et al.*, 2007], Algorithm 1 works in polynomial time.

6 Related Work

In [Flouris *et al.*, 2005], Flouris et al. generalize the well-known AGM (Alchourrón, Gärdenfors and Markinson) framework in order to apply the rationales behind the AGM framework to a wider class of logics. In [Flouris *et al.*, 2006], a framework for the distinction between incoherence and inconsistency of an ontology is proposed. A set of rational postulates for a revision operator in DLs is proposed. In [Qi *et al.*, 2006], reformulated AGM postulates for revision are adapted to DLs. A set of postulates is given in [Ribeiro and Wassermann, 2007] by adapting Hansson's postulates for a semi-revision operator (see [Hansson, 1999]).

Some concrete operators have been proposed. In [Haase *et al.*, 2005], an algorithm is given to determine consistent sub-ontologies by adding an axiom to an ontology. A revision operator is given in [Halaschek-Wiener *et al.*, 2006] based on Hansson's kernel operator. In [Qi *et al.*, 2006], the authors propose two revision operators that satisfy their postulates. All these revision operators deal with inconsistencies. Qi et al. in [Qi *et al.*, 2008] propose a general revision operator to deal with incoherence by adapting Hansson's kernel revision operator. However, this operator is not fine-grained in the sense that it removes a whole axiom from a TBox if it is selected by an *incision function*. Our revision operators are different from existing ones in that they follow another family of belief revision operators, i.e. model-based revision operators, so they are more fine-grained than existing ones.

Our work is different from existing revision approaches on prioritized knowledge bases, such as [Benferhat *et al.*, 2002], because our work is based on a preference over concept names.

7 Conclusion and Future Work

In this paper, we adapted the well-known Dalal revision operator for revising terminologies in description logics. We pointed out some pitfalls when we tried to adapt the notion of a difference set between two interpretations and the notion of a distance function between TBoxes. We defined three model-based revision operators successively and showed that one of them ($\circ_S$) is more rational than others using some examples. We then defined a revision operator syntactically using the notion of concept forgetting and showed that this operator is semantically equivalent to the revision operator

$\circ_S$. We considered logical properties of our revision operators with respect to postulates given in [Qi *et al.*, 2006] and by proposing some new postulates. We showed that the revision operator $\circ_S$ is more rational than others. Finally, we showed that subsumption checking in DL-Lite$_{core}$ under our revision operator $\circ_S$ is $P^{NP[O(\log n)]}$-complete and provided a polynomial time algorithm to compute the result of revision in a special case.

We consider only revision of terminologies in DLs. As a future work, we will extend our work to consider ontologies with ABoxes. This is very challenging because it has been shown in [Wang *et al.*, 2008] that even for DL-Lite languages, forgetting results in DL-Lite with ABoxes are not expressible in the same language. We will consider the following solutions. First, we do not require that the result of revision must be expressed in the same language as the original language for the ontologies under consideration. Second, we consider approximation of the result of revision in the original language, like the work done in [Giacomo *et al.*, 2007].

Acknowledgments

References

[Baader *et al.*, 2007] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and application*. Cambridge University Press, 2007.

[Benferhat *et al.*, 2002] Salem Benferhat, Didier Dubois, Henri Prade, and Mary-Anne Williams. A practical approach to revising prioritized knowledge bases. *Studia Logica*, 70(1):105–130, 2002.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[Dalal, 1988] M. Dalal. Investigations into a theory of knowledge base revision. In *Proc. of AAAI*, pages 475–479, 1988.

[Flouris *et al.*, 2005] G. Flouris, D. Plexousakis, and G. Antoniou. On applying the AGM theory to DLs and OWL. In *Proc. of ISWC*, pages 216–231, 2005.

[Flouris *et al.*, 2006] G. Flouris, Z. Huang, J.Z. Pan, D. Plexousakis, and H. Wache. Inconsistencies, negations and changes in ontologies. In *Proc. of AAAI*, pages 1295–1300, 2006.

[Gärdenfors, 1988] P. Gärdenfors. *Knowledge in Flux-Modeling the Dynamic of Epistemic States*. The MIT Press, Cambridge, Mass, 1988.

[Giacomo *et al.*, 2007] G. De Giacomo, M. Lenzerini, A. Poggi, and R. Rosati. On the approximation of instance level update and erasure in description logics. In *Proc. of AAAI*, pages 403–408, 2007.

[Haase and Stojanovic, 2005] P. Haase and L. Stojanovic. Consistent evolution of owl ontologies. In *Proc. of ESWC*, pages 182–197, 2005.

[Haase *et al.*, 2005] P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *Proc. of ISWC*, pages 353–367, 2005.

[Halaschek-Wiener *et al.*, 2006] C. Halaschek-Wiener, Y. Katz, and B. Parsia. Belief base revision for expressive description logics. In *Proc. of OWL-ED*. 2006.

[Hansson, 1999] S.O. Hansson. *A Textbook of Belief Dynamics: Theory Change and Database Updating*. Kluwer Academic Publishers, 1999.

[Katsuno and Mendelzon, 1992] H. Katsuno and A.O. Mendelzon. Propositional knowledge base revision and minimal change. *Artif. Intell.*, 52(3):263–294, 1992.

[Konev *et al.*, 2008] B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularisation. In H. Stuckenschmidt and S. Spaccapietra, eds., *Ontology Modularization,* Springer, 2008.

[Krentel, 1988] Mark W. Krentel. The complexity of optimization problems. *J. Comput. Syst. Sci.*, 36(3):490–509, 1988.

[Lang and Marquis, 2002] J. Lang and P. Marquis. Resolving inconsistencies by variable forgetting. In *Proc. of KR*, pages 239–250, 2002.

[Meyer *et al.*, 2005] T. Meyer, K. Lee, and R. Booth. Knowledge integration for description logics. In *Proc. of AAAI*, pages 645–650, 2005.

[Qi *et al.*, 2006] G. Qi, W. Liu, and D. Bell. Knowledge base revision in description logics. In *Proc. of JELIA*, pages 386–398, 2006.

[Qi *et al.*, 2008] G. Qi, P. Haase, Z. Huang, Q. Ji, J.Z. Pan, and J. Völker. A kernel revision operator for terminologies–algorithms and evaluation. In *Proc. of ISWC*, pages 419–434. 2008.

[Rector *et al.*, 2004] A.L. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *Proc. of EKAW*, pages 63–81, 2004.

[Ribeiro and Wassermann, 2007] M. Moretto Ribeiro and R. Wassermann. Base revision in description logics - preliminary results. In *Proc. of IWOD*, pages 69–82, 2007.

[Schlobach *et al.*, 2007] S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen. Debugging incoherent terminologies. *J. Autom. Reasoning*, 39(3):317–349, 2007.

[Wang *et al.*, 2008] Z. Wang, K. Wang, R. W. Topor, and J.Z. Pan. Forgetting concepts in dl-lite. In *Proc. of ESWC*, pages 245–257, 2008.